EDS84AVTCxx
13387462

# L-force *Drives*

**Software Manual**

# 8400


**E84AVTCxxxxx**

**Inverter Drives 8400 TopLine C**

Lenze

# Overview of technical documentation for Inverter Drives 8400

**Project planning, selection & ordering**

- ☐ 8400 hardware manual
- 🖹 Catalogue

**Mounting & wiring**

- 🖹 MA 8400 BaseLine/StateLine/HighLine/TopLine
- 🖹 MA for the communication module
- 🖹 MA for the extension module
- 🖹 MA for the safety module
- 🖹 MA for the accessories

**Parameterisation**

- 🖹 BA keypad
- ☐ SW 8400 BaseLine
- ☐ SW 8400 StateLine
- ☐ SW 8400 HighLine
- ☐ SW 8400 TopLine ← **This documentation**
- ☐ KHB for the communication module

**Drive commissioning**

- ☐ SW 8400 BaseLine/StateLine/HighLine/TopLine
  - → Chapter "Commissioning"
  - → Chapter "Diagnostics & error management"
- ☐ Remote maintenance manual

**Networking**

- ☐ KHB for the communication medium used

Legend:

- 🖹 Printed documentation
- ☐ Online documentation (PDF/Engineer online help)

Abbreviations used:

| | |
|---|---|
| BA | Operating Instructions |
| KHB | Communication manual |
| MA | Mounting instructions |
| SW | Software Manual |

# Contents

# 1 About this documentation

**Danger!**

The controller is a source of danger which may lead to death or severe injury of persons.

To protect yourself and others against these dangers, observe the safety instructions before switching on the controller.

Please read the safety instructions provided in the **8400 mounting instructions** and in the **8400 hardware manual**. Both documents are supplied with the controller.

This software manual contains information on the parameterisation of the 8400 TopLine C controller by means of the L-force »Engineer« and the X400 keypad.

The information provided in this software manual applies to the 8400 TopLine C controller with the following nameplate data:

| Product series | Type designation | from software version |
|---|---|---|
| 8400 TopLine C | E84AVTCxxxxx | 01.00 |

All screenshots provided in this documentation are application examples. Depending on the software version of the controller and the version of the installed »Engineer« software, the screenshots in this documentation may differ from the representation in the »Engineer«.

**Tip!**

Information and tools for Lenze products are provided in the download area at

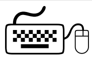
http://www.Lenze.com

## 1.1 Document history

| Version | | | Description |
|---|---|---|---|
| 2.0 | 08/2011 | TD05 | Extended by new functions for 8400 TopLine C V02.00.00 |
| 1.2 | 02/2011 | TD05 | Error corrections & supplements |
| 1.1 | 11/2010 | TD05 | First edition |

## 1.2 Conventions used

This Software Manual uses the following conventions to distinguish between different types of information:

| Type of information | Writing | Examples/notes |
|---|---|---|
| Spelling of numbers | | |
| Decimal separators | Point | The decimal point is generally used.<br>For example: 1234.56 |
| Text | | |
| Version info | Blue text colour | Information that is only valid for or from a certain software version of the controller is marked accordingly in this manual.<br>Example: This function extension is available from version 02.00.00! |
| Program name | » « | The Lenze »Engineer« PC software ... |
| Window | *italics* | The *Message* window ... / The *Options* dialog box... |
| Variable identifier | | Set *bEnable* to TRUE to... |
| Control element | **bold** | The **OK** button... / The **Copy** command... / The **Properties** tab... / The **Name** input field... |
| Sequence of menu commands | | If the execution of a function requires several commands, the individual commands are separated by an arrow: Select **File→Open** to... |
| Shortcut | **<bold>** | Press **<F1>** to open the online help. |
| | | If a command requires a combination of keys, a "+" is placed between the key symbols:<br>Use **<Shift>+<ESC>** to... |
| Hyperlink | underlined | Optically highlighted reference to another topic. In this documentation activated by mouse-click. |
| Icons | | |
| Page reference | (📖 26) | Optically highlighted reference to another page. In this documentation activated by mouse-click. |
| Step-by-step instructions |  | Step-by-step instructions are indicated by a pictograph. |

Information that is only valid for or from a certain software version of the controller is marked accordingly in this manual.

## 1.3 Terminology used

| Term | Meaning |
|---|---|
| »Engineer« | Lenze PC software which supports you in "engineering" (parameterisation, diagnostics and configuration) throughout the whole life cycle, i.e. from planning to maintenance of the commissioned machine. |
| Application block | Block for a technology application (e.g. actuating drive speed)<br>A technology application is a drive solution based on the experience and know-how of Lenze in which function blocks interconnected to a signal flow form the basis for implementing typical drive tasks. |
| ASM | Abbreviation for asynchronous motor |
| Code | Parameter used for controller parameterisation or monitoring.<br>The term is usually called "index". |
| Display code | Parameter that displays the current status or value of an input/output of a system block. |
| Emergency brake | The emergency brake serves to shutdown rotary or translatory masses in motion in emergency situations. Emergency situations are exceptional situations that only occur sporadically. |
| FB Editor | Function block editor<br>Graphical interconnection tool which is provided for FB interconnections in the »Engineer« on the **FB Editor** tab and by means of which the applications integrated in the 8400 HighLine controller can also be reconfigured and extended by individual functions. |
| Function block | General designation of a function block for free interconnection (from "HighLine" device version).<br>A function block can be compared with an integrated circuit that contains a certain control logic and delivers one or several values when being executed.<br>Each function block has a unique identifier (the instance name) and a processing number which defines the position at which the function block is calculated during the task cycle.<br>Example: "L_Arithmetik1" (function block for arithmetic operations) |
| Holding brake | The holding brake serves to statically hold e.g. a position during the downtimes of a robot/travelling/synchronous/hoist drive. |
| Keypad | The keypad is an alternative to the PC for the local operation, parameterisation, and diagnostics in a simple manner.<br>• Detailed type designation: X400 keypad<br>• Order designation: EZAEBK1001 |
| LA | Abbreviation for **L**enze **A**pplication block<br>Example: "LA_NCtrl" (block for the "Actuating drive speed" application) |
| Lenze setting | This setting is the default factory setting of the device. |
| LP | Abbreviation for **L**enze **P**ort block<br>Example: "LP_CanIn1" (CAN1 port block) |
| LS | Abbreviation for **L**enze **S**ystem block<br>Example: "LS_DigitalInput" (system block for digital input signals) |
| Port block | Block for implementing the process data transfer via a fieldbus |
| PSM | Abbreviation for permanently excited synchronous motor |
| QSP | Abbreviation for quick stop |
| SC | Abbreviation for **S**ervo **C**ontrol |
| Service brake | The service brake serves to shutdown rotary or translatory masses in motion in a controlled manner. The energy to be dissipated in this process is produced in the form of friction energy. Unlike emergency braking, this process is a regular and recurring operating mode. |
| SLPSM | Abbreviation for sensorless control of synchronous motors |
| SLVC | Abbreviation for **S**ensor**L**ess **V**ector **C**ontrol |

| Term | Meaning |
|---|---|
| Subcode | If a code contains several parameters, the individual parameters are stored under "subcodes".<br>This Manual uses a slash "/" as a separator between code and subcode (e.g. "C00118/3").<br>The term is usually called "subindex". |
| System block | System blocks provide interfaces to basic functions and to the hardware of the controller in the FB Editor of the »Engineer« (e.g. to the digital inputs). |
| USB diagnostic adapter | The USB diagnostic adapter is used for the operation, parameterisation, and diagnostics of the controller. Data are exchanged between the PC (USB connection) and the controller (diagnostic interface on the front) via the diagnostic adapter.<br>• Order designation: E94AZCUS |
| VFCplus | Abbreviation for **V**oltage **F**requency **C**ontrol |

## 1.4 Definition of the notes used

The following signal words and symbols are used in this Software Manual to indicate dangers and important information:

### Safety instructions

Layout of the safety instructions:

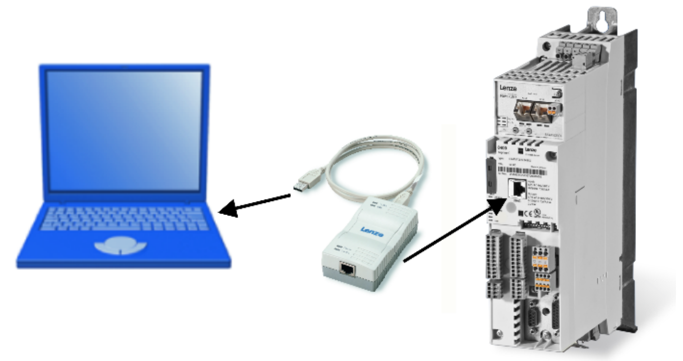
> **Pictograph and signal word!**
>
> (characterise the type and severity of danger)
>
> **Note**
>
> (describes the danger and informs how to prevent dangerous situations)

| Pictograph | Signal word | Meaning |
| --- | --- | --- |
| | **Danger!** | **Danger of personal injury through dangerous electrical voltage**<br>Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken. |
| | **Danger!** | **Danger of personal injury through a general source of danger**<br>Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken. |
| | **Stop!** | **Danger of property damage**<br>Reference to a possible danger that may result in property damage if the corresponding measures are not taken. |

### Application notes

| Pictograph | Signal word | Meaning |
| --- | --- | --- |
| | **Note!** | Important note to ensure trouble-free operation |
| | **Tip!** | Useful tip for easy handling |

# 2 Introduction: Parameterising the controller



[2-1] Exemplary constellation for parameterising the controller

Being a component of a machine which includes a speed-variable drive system, the controller needs to be adjusted to its drive task. The controller is adjusted by changing parameters which are saved in the memory module. These parameters can be accessed by keypad, by the L-force »Engineer« or by a master control via fieldbus communication:

- Simply plug in the X400 keypad onto the controller if you would like to use it.
- The USB diagnostic adapter, for instance, can be used for the communication between the PC (including the L-force »Engineer« software) and the controller, see illustration. The USB diagnostic adapter is the connection between the PC (free USB port) and the controller (X6 diagnostic interface).
- The "CAN on board" CAN interface and the MCI interface are provided for fieldbus communication and can be used for a communication module (e.g. PROFIBUS).

**Tip!**

How to obtain/update the L-force »Engineer« software:

- **Download from the Internet:**
  The full version of the »Engineer StateLevel« is provided free of charge. Current software can be found on the Internet in the "Services & Downloads" area under http://www.Lenze.com.
- **Requesting the CD**
  You can also request the L-force »Engineer« separately on CD free of charge at your Lenze representative. See the "About Lenze" area on our homepage for e.g. the corresponding German address.

## 2.1 General notes on parameters

All parameters for controller parameterising or monitoring are saved as so-called "codes".

- The codes are numbered and indicated by the prefix "C" before the code, e.g. "C00002".
- In addition, every code has a name and specific attributes:
  - Access type (read, write)
  - Data type
  - Limit values
  - Lenze setting (factory settings)
- For the sake of clarity, some codes contain "subcodes" for saving parameters. This Manual uses a slash "/" as a separator between code and subcode, e.g. C00118/3".
- According to their functionality, the parameters are divided into three groups:

| Parameter group | Examples |
|---|---|
| **Setting parameters**<br>Parameters for specifying setpoints and for setting device / monitoring functions. | C00005: Application selection<br>C00012: Accel. time - main setpoint<br>C00169: Logbook setting<br>C00604: Resp. to device overload (Ixt) |
| **Configuration parameters**<br>Parameters for configuring signal connections (function block interconnections) within the device, e.g. assignment of the digital input terminals to the control inputs of the technology application. | C00700: LA_NCtrl: Analog connection list<br>C00701: LA_NCtrl: Digital connection list<br>C00710: LA_TabPos: Analog connection list<br>C00711: LA_TabPos: Digital connection list |
| **Diagnostic/Display parameters**<br>Parameters for displaying device-internal process factors, current actual values, and status messages, e.g. for diagnostic purposes. These are read-only parameters. | C00052: Motor voltage<br>C00137: Device status<br>C00150: Status word<br>C00165: Error information |

**Tip!**

The terms "code" and "subcode" generally correspond to the terms "index" and "subindex" and "parameter" and "subparameter".

## 2.2 Handling the memory module

All parameters of the drive system are saved in the integrated memory module of the controller. These include

- the parameters of the controller
- the parameters of the communication module plugged into the MCI interface
- the parameters of the possibly existing safety module (device variant)

When handling the memory module, a distinction is drawn between the following scenarios:

### Delivery

- All devices are delivered with a plugged-on memory module.
- In the delivery state, the Lenze setting of the parameters has been saved to the memory module.
- The memory module can be preconfigured with customised data.
- The memory module is available as a spare part - without any data.

### During operation

**Note!**

Automatic saving is explicitly not supported because this significantly reduces the service life of the memory module.

- Plugging and unplugging the memory module results in a message.
- Full functionality of the memory module is even provided if the power supply has been switched off and only the electronic components of the controller are externally supplied by a 24 V DC voltage, e.g. via the X5/24E terminal.
- Parameter sets can be saved manually.
- Parameter sets can be loaded manually.

### Removing the memory module

- The memory module can be plugged and unplugged during operation. The behaviour of the device remains basically unchanged because all parameters are available in the RAM after the device has been started.
- Unplugging the memory module during operation results in an error message and should therefore be avoided.
- The device can also be parameterised when the memory module is unplugged. Of course, the parameter sets cannot be saved to the memory module then (cp. Saving the parameters to the memory module safe against mains failure (🕮 33)).

### Replacing the controller

- In the event of a device replacement, the entire parameter data of an axis can be copied to the replacement device by "taking along" the memory module, so that additional PC or keypad operations are not required.
- When replacing the controller, the versions of the old device and the new device are of importance. Before data are actually transferred, the versions are internally checked. Basically, the following applies:
  - Parameter sets of old devices with V 1.0 can be processed on new devices ≥ V 1.0 (downward compatibility).
  - Parameters of devices with higher versions are not supported on devices with lower versions. An error message will be issued if the parameter set versions of the two devices are not compatible.
  - Parameter sets of devices with versions that have less functions (e.g. 8400 StateLine) can be loaded into and executed on devices with versions that have more functions (e.g. 8400 HighLine). The reverse is not possible!

### Saving the parameters to the memory module safe against mains failure

Controller parameter changes via the »Engineer«, the keypad, or a master control via fieldbus communication will be lost after mains switching of the controller unless the settings have been explicitly saved to the memory module.

- Execute the "Save all parameter sets" device command (C00002/11) to save the current parameter settings to the memory module. (🕮 96)
- Using the keypad, you can also press function key SAVE to save the parameter settings.

**Note!**

When the device or the external 24 V DC voltage supply is switched on, all parameters are automatically loaded from the memory module into the main memory of the controller.

Please observe the following points during the storage process to avoid data inconsistencies which would result in an error when loading the parameters from the memory module:

- Do not switch off the supply voltage!
- Do not remove the memory module from the device!

### Parameter set transfer using the »Engineer«

When an online connection to the controller has been established, the following transfer functions can directly be executed via the *Toolbar* or the **Online** menu using the L-force »Engineer«:

| Symbol | Menu command | Shortcut |
|---|---|---|
|  | Download parameter set | <F5> |
|  | Read parameter set from device | <F7> |
|  | Save parameter set | |

**Tip!**

Detailed information on parameter set transfers using the »Engineer« can be found in the »Engineer« online help.

## 2.3 Handling the keypad

Use the keypad for quick and simple parameter setting and for displaying current actual values and device statuses via the respective display parameters. For this purpose, plug in the keypad onto the X6 diagnostic interface on the front of the standard device.

**Danger!**

In general, changing a parameter causes an immediate response in the controller!

- This may lead to undesirable behaviour on the motor shaft if the controller has been enabled!
- Setpoint sources, for instance, may switch over all of a sudden (e.g. when configuring the signal source for the main setpoint).

Certain device commands or settings which may cause critical states of drive behaviour constitute exceptions. Such parameter changes are only possible if the controller is inhibited. Otherwise, a corresponding error message will be issued.

**Note!**

Plugging and unplugging the keypad is possible during operation.

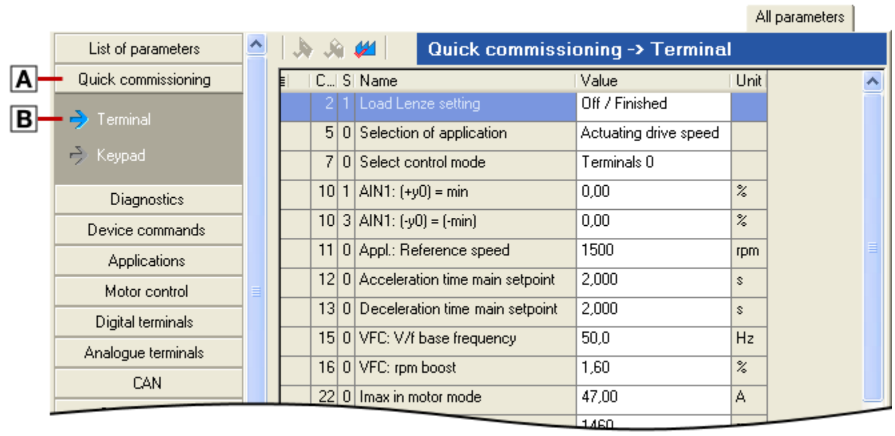
Detailed information on the keypad can be found in the operating instructions accompanying the keypad!

### Menu structure

In the keypad, the parameters are classified into various menus and submenus.

- The **USER menu** includes a selection of frequently used parameters (freely configurable in C00517).
- The **Code list** contains all parameters.
- The **Go to param** function enables you to reach the corresponding parameter directly.
- The **Logbook** logs all errors and their chronological history.

▶ All other menus and submenus in the keypad correspond precisely to the categories and subcategories which are displayed in the form of buttons on the left hand side of the **All parameters** tab in the »Engineer«:




A Category
B Subcategories

[2-2] **All parameters** tab in the »Engineer«

### General operation

1. Use the ⇧/⇩ navigation keys to select the desired menu.
   – Use the ⇦/⇨ navigation keys to reach a higher/lower menu level.
   – Use the ☰ function key to return to the main menu.
2. Use the ⇧/⇩ navigation keys to select the parameter to be set within a submenu.
3. Use the EDIT function key to switch over to the editing mode.
4. Use the navigation keys to set the desired value.
   – Use the ⇦/⇨ navigation keys to move the cursor to the left/right.
   – Use the ⇧/⇩ navigation keys to change the selected digit.
5. Use the OK function key to accept the change and to leave the editing mode.
   – Use the ESC function key to leave the editing mode without accepting the change.

**Tip!**

General information on setting parameters using the »Engineer« can be found in the »Engineer« online help.

## 2.4 Device access protection

Various tasks can be carried out via the functions of the device access protection:

- Password protection
  - Only authorised persons (who know the password) are permitted to read/change all parameters of the controller.
  - Unauthorised persons (who do not know the password) are only granted access to the max. 32 parameters of the user menu.
- Device personalisation
  - Only controllers and memory modules which are personalised with a specific binding ID can be used for the system.

**Note!**

If password protection/device personalisation is used:

- Let the end customer know that Lenze can only provide limited service in case of devices with access protection.
- Lenze cannot modify e.g. a replacement device via special access to make it work with a personalised memory module.
- The keypad does not support the alpha-numeric entry of a password, thus the keypad cannot be used for entry.

### 2.4.1 Password protection

If password protection has been activated, only a write/read access to the parameters of the user menu is possible. Possible configurations of various protective functions, personalised for every single communication channel, are being prepared.

**Short overview of the relevant parameters for password protection:**

| Parameter | Info | | Lenze setting |
|---|---|---|---|
| C00505/3 | Password<br>• The password can have a maximum length of 16 characters.<br>• The password may contain the following characters: lower case letters (a - z), upper case letters (A - Z), digits (0 - 9)<br>**Note:**<br>After the execution of one of the device commands listed below, this parameter provides the current password status: | | |
| | OFF | No password has been set, password protection is not active (Lenze delivery status). | |
| | ON | Password has been set, password protection is active.<br>• This status is also displayed after an unsuccessful check/deletion of the password due to an invalid entry. | |
| | OK | Password has been set, password protection is not active.<br>• Password protection is temporarily deactivated. | |
| **Device commands**<br>Before the following device commands are executed, enter the corresponding password in C00505/3. | | | |
| C00002/31 | Set password<br>▸ Activating password protection | | 0: Off / ready |
| C00002/32 | Check password<br>▸ Temporary deactivation of active password protection | | 0: Off / ready |
| C00002/33 | Delete password<br>▸ Deactivating password protection/changing the password | | 0: Off / ready |
| **Status displays** | | | |
| C00003 | Status of the last device command | | - |
| C00507/1 | Password protection - all communication channels<br>• Bit coded display of the active protective functions: | | - |
| | Bit 0 | Only access to user menu | |
| | Bit 1 | Parameter write protection | |
| | Bit 2 | Parameter read protection | |
| | Bit 3 ... 14 | Reserved | |
| | Bit 16 | Memory module binding on | |
| Highlighted in grey = display parameter | | | |

### Activating password protection

Password protection is activated by setting a password.

**How to set a password:**

1. Enter the desired password in C00505/3.
   - The password can have a maximum length of 16 characters.
   - The password may contain the following characters: lower case letters (a - z), upper case letters (A - Z), digits (0 - 9)
2. Execute "Set password" device command: C00002/31 = "1: On / start"
   - After successful execution, password status ON is displayed in C00505/3 and password protection takes immediate effect.

### Temporary deactivation of active password protection

Execute the "Check password" device command to deactivate the password protection temporarily to be able to carry out password protected functions.

▶ Password protection will be deactivated until
  – a valid password is entered and checked
    - or -
  – the external 24 V supply of the control electronics is switched off (< 19 V).

**How to temporarily deactivate active password protection:**

1. Enter the set password in C00505/3.
2. Execute "Check password" device command C00002/32 = "1: On / start"
   - After a successful check, password status OK is displayed in C00505/3.

### Deactivating password protection/changing the password

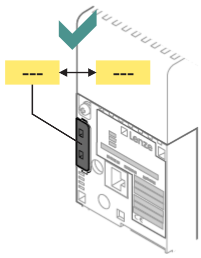
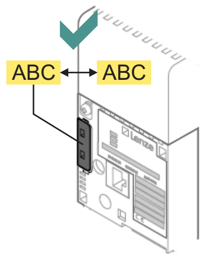
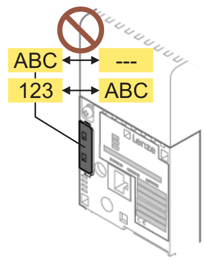
Password protection can simply be deactivated by deleting the set password. If you would like to change the set password, delete the set password first. Then set the desired new password.

**How to delete the set password:**

1. Enter the set password in C00505/3.
2. Execute "Delete password" device command C00002/33 = "1: On / start"
   - After a successful deletion, password status OFF is displayed in C00505/3.

### 2.4.2 Device personalisation

The device personalisation serves to connect the controller to the memory module by means of a binding ID. If the device personalisation is active, write/read actions between the controller and the memory module are only executed if both components have identical binding IDs.

| **Lenze delivery status:** | **Operation at the customer's site:** | **Impermissible replacement by the end user:** |
|---|---|---|
|  |  |  |
| No binding ID has been set. | Customer sets binding ID for device personalisation purposes. | If device personalisation is active: A replacement of the controller or the memory module results in an error message if the binding ID is incorrect or not available. |

If, for instance, a parameter set has been loaded from the memory module when the device personalisation was active, this parameter set cannot be saved to another memory module that has a different binding ID or no binding ID at all.

- Hence, a parameter set cannot be copied from a personalised memory module to a non-personalised memory module!

The following two types are distinguished during the check:

- If a different binding ID is detected during switch-on of the controller (during device initialisation):
  - The "Fault" error response is returned.
  - The "PS10: Invalid memory module binding" error message is entered into the logbook.
- If a different binding ID is detected during the execution of a device command for loading/saving the parameter set:
  - The loading/saving process is not carried out.
  - A corresponding status for the device command is output in C00003.

### Short overview of the relevant parameters for device personalisation:

| Parameter | Info | Lenze setting |
|---|---|---|
| C00505/2 | Binding ID<br>• The binding ID can have a maximum length of 16 characters.<br>• The binding ID may contain the following characters: lower case letters (a - z), upper case letters (A - Z), digits (0 - 9)<br>**Note:**<br>After the execution of one of the device commands listed below, this parameter provides the current binding ID status:<br>OFF: No binding ID has been set.<br>ON: Binding ID has been set. | |
| **Device commands**<br>Before the following device commands are executed, enter the corresponding binding ID in C00505/2. | | |
| C00002/29 | Set binding ID<br>▸ Activating device personalisation | 0: Off / ready |
| C00002/30 | Delete binding ID<br>▸ Deactivating the device personalisation/changing the binding ID | 0: Off / ready |
| **Status displays** | | |
| C00003 | Status of the last device command | - |
| C00507/1 | Password protection - all communication channels<br>• Bit coded display of the active protective functions:<br>Bit 0: Only access to user menu<br>Bit 1: Parameter write protection<br>Bit 2: Parameter read protection<br>Bit 3 ... 14: Reserved<br>Bit 16: Memory module binding on | - |

Highlighted in grey = display parameter

### Activating device personalisation

The device personalisation is activated by setting a binding ID.

**How to set a binding ID:**

1. Enter the desired binding ID in C00505/2.
   - The binding ID can have a maximum length of 16 characters.
   - The binding ID may contain the following characters:
     lower case letters (a - z), upper case letters (A - Z), digits (0 - 9)
2. Execute "Set binding ID" device command C00002/29 = "1: On / start"
   - After successful execution, status ON is displayed in C00505/2.

#### Deactivating the device personalisation/changing the binding ID

The device personalisation is simply deactivated by deleting the set binding ID. If you would like to change the set binding ID, delete the set binding ID first. Then set the desired new binding ID.

**How to delete the binding ID:**

1. Enter the set binding ID in C00505/2.
   - If the controller and the memory module have different binding IDs, enter the binding ID of the memory module to delete the binding IDs of both components.
2. Execute "Delete binding ID" device command C00002/30 = "1: On / start"
   - After a successful deletion, status OFF is displayed in C00505/2.

### 2.4.3 Unlocking the controller with a MasterPin

Every controller has an individual master password, the MasterPin. A controller which is locked due to password mechanism can reach the delivery status again if the MasterPin is entered.

**Stop!**

If the MasterPin is entered, the parameter set both in the controller and in the memory module is reset to the Lenze setting!
- Customised parameterisation will be permanently lost and must be recreated!
- The reset to the Lenze setting may lead to unexpected level changes at the I/O terminals (e.g. brake control)!

**How to re-establish the delivery status:**

1. Inhibit the controller if it is enabled, e.g. via the C00002/16 device command.
2. Enter the MasterPin in C00505/1.
   - The last six digits of the serial number of the memory module represent the MasterPin.
3. Execute "Check MasterPin" device command C00002/28 = "1: On / start"

# 3 Commissioning

**Danger!**

**Uncontrolled motor movements can occur**

Under certain conditions the motor may rotate after mains connection.

**Possible consequences:**

- Persons in the vicinity of the machine or plant risk getting hurt.
- Unexpected starting action may damage the machine or plant.

**Protective measures:**

- Commissioning with external 24 V supply and without mains voltage. In this case, the controller can only be parameterised and diagnosed during commissioning.
- Ensure that setpoints are not active.

**Tip!**

- Information on some of the operating statuses can quickly be obtained via the LED status displays on the front of the controller. (📖 582)
- **Check firmware:** Particularly with regard to the use of older controllers (e.g. if the customer is using one from stock) it makes sense to check the software (firmware) version. The software version of the controller can be seen on the nameplate in the "HW/SW" line and can be determined by reading out code C00099.
- **Restore delivery status:** Set code C00002/1 to "1: On / start" to reset all parameter settings of the device to the Lenze setting. This leaves you with a defined device configuration. ▶ Load Lenze setting (📖 94)

## 3.1 Selection of the right commissioning tool

There are two possibilities for commissioning the 8400 TopLine controller:

- Commissioning via keypad (diagnosis terminal)
  - If you only have a few parameters to adapt.
  - For test / demonstration purposes.
- Commissioning via PC / »Engineer«
  - In general, commissioning via the »Engineer« is suitable for every drive task, particularly for those with higher requirements or more comprehensive parameterisation.
  - Via the »Engineer« all parameters of the 8400 TopLine controller can easily be accessed which makes commissioning highly flexible.
  - Moreover, the »Engineer« provides comprehensive diagnostics options and an integrated online help which can be called via function key **[F1]**.

The following chapters describe the commissioning of the available technology applications with the »Engineer«.

Information on the commissioning with the keypad (diagnosis terminal) is provided in the **8400 hardware manual**.

- The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 drive controller.

## 3.2 Safety instructions with regard to commissioning

### General safety instructions

To avoid injury to persons or damage to material assets,

- check before connecting the mains voltage
  - the wiring for completeness, short circuit, and earth fault
  - the "emergency stop" function of the entire system
  - that the motor circuit configuration (star/delta) is adapted to the output voltage of the controller
  - the in-phase connection of the motor
- check the setting of the most important drive parameters before enabling the controller:
  - the V/f rated frequency must be adapted to the motor circuit configuration!
  - the drive parameters relevant for your application must be set correctly!
  - the configuration of the I/O terminals must be adapted to the wiring!
- ensure that there are no active speed setpoints before enabling the controller.

### Safety instructions with regard to motor operation

**Danger!**

- Continuous operation of self-ventilated motors at small field frequencies and rated motor currents is not permissible for thermal reasons!
  - In the Lenze setting, the Motor temperature monitoring (PTC) is activated. (📖 278)
  - Activate the Brake resistor monitoring (I2xt) if necessary. (📖 279)
- C00015 must be used to select 87 Hz operation if a delta-connected asynchronous motor (nameplate data: 400 V Y /230 V △) is to be operated in conjunction with a drive controller for a mains voltage of 400 V.

## 3.3 Notes on motor control

In the Lenze setting, the V/f characteristic control (VFCplus) as motor control is set in C00006 with a linear characteristic.

- V/f characteristic control (VFCplus) is a motor control mode for classic frequency inverter applications on the basis of a simple and robust control procedure for the operation of machines with a linear or quadratic load torque characteristic (e.g. fans).
- The presettings of the parameters ensure that the controller is ready for operation right away and the motors works adequately without further parameterisation if a controller and a 50 Hz asynchronous machine with matching performances are assigned.

**Note!**

Check the nameplate data against the motor data set in the controller. Further information is provided in the chapter "Motor selection/Motor data". (📖 121)

**Recommendations for the following application cases:**

- If the controller and motor differ greatly in terms of performance:
  Set the Imax limit (in motor mode) in C00022 to 2x rated motor current.
- If a high starting torque is required:
  When the motor is idling, set a value for Vmin boost in C00016 which ensures that the rated motor current flows at a field frequency of f = 3 Hz (display in C00058).
- For noise optimisation:
  In C00018, set a switching frequency of "16 kHz var./drive-opt.".
- If a high torque must be provided at small speeds without feedback:
  Select "Sensorless vector control (SLVC) as motor control mode in C00006.

**Related topics:**

- Motor control (MCTRL) (📖 120)

## 3.4 Preconditions for commissioning with the »Engineer«

For commissioning, you need

- a PC that satisfies the following requirements:
  - processor with 1.4 GHz or higher
  - at least 512 MB RAM and 650 MB free hard disc space
  - Microsoft® Windows® 2000 operating system (from service pack 2 onwards) or Windows® XP
- the Lenze »Engineer« PC software
- a connection to the controller, e.g. via a USB diagnostic adapter:
  - connect the USB diagnostic adapter to the X6 diagnostic interface.
  - establish a connection between the USB diagnostic adapter and the PC via a free USB port.

**Tip!**

How to obtain/update the L-force »Engineer« software:

- **Download from the Internet:**
  The full version of the »Engineer StateLevel« is provided free of charge. Current software can be found on the Internet in the "Services & Downloads" area under http://www.Lenze.com.
- **Requesting the CD**
  You can also request the L-force »Engineer« separately on CD free of charge at your Lenze representative. See the "About Lenze" area on our homepage for e.g. the corresponding German address.

## 3.5 Trouble-shooting during commissioning

With the »Engineer« trouble during commissioning can be detected and eliminated conveniently. Proceed as follows:

- Check whether error messages are displayed in the »Engineer«.
  - On the **Diagnostics** tab, relevant actual states of the controller and pending error messages are displayed in a well-arranged visualisation.
- Check the input terminals for their corresponding setpoints.
  - On the **Terminal assignment** tab, the current input and output signals are displayed.
- Check the signal flow of the application.
  - For this purpose, click the **Signal flow** button on the **Application parameter** tab. The displayed signal flow shows active setpoints and their further processing.

**Related topics:**

- Diagnostics & error management (581)
- LED status displays (582)
- Error messages of the operating system (602)

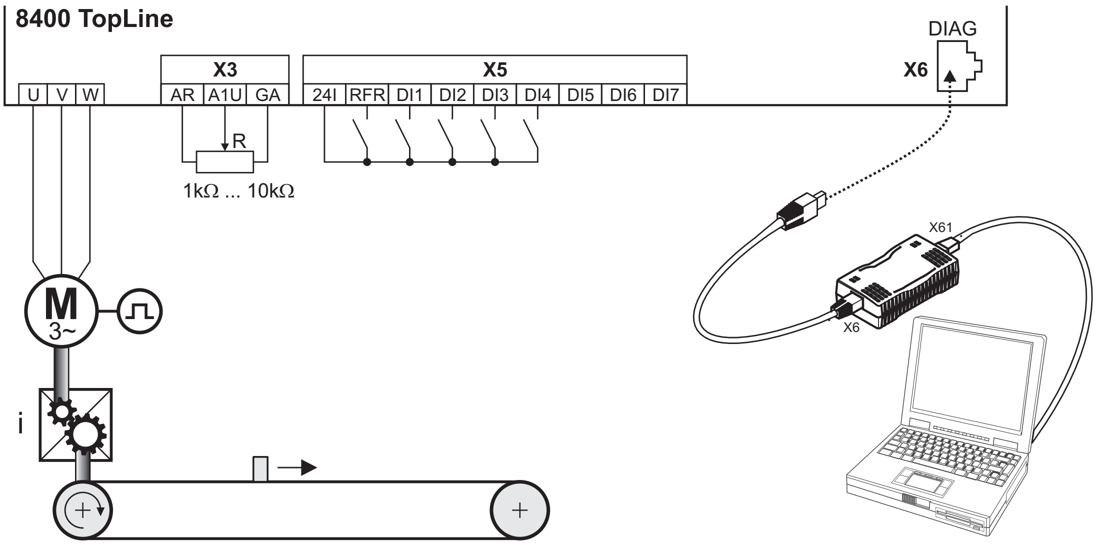
## 3.6 Commissioning of the "Actuating drive speed" technology application

**Note!**

Take all the necessary safety precautions before you carry out the following commissioning steps and switch the device on!

▶ Safety instructions with regard to commissioning (📖 45)

### System constellation




| | |
|---|---|
| X3/AR | Reference voltage (10 V) for analog signals |
| X3/A1U | Speed setpoint selection (slider of setpoint potentiometer R)<br>• Scaling: 10 V ≡ 100 % ≡ 1500 rpm[1] (for a 4-pole motor) |
| X3/GA | Ground potential (GND) for analog signals |
| X5/RFR | Controller enable |
| X6 | Diagnostic interface for connecting the USB diagnostic adapter |

[3-1] Block diagram for wiring the commissioning example for the "Actuating drive speed" application

### Commissioning steps

Find a description of the commissioning steps of the "Actuating drive speed" technology application below.

Please observe the sequence of the steps in the following chapters and follow them through carefully. This will help you to commission your controller quickly and as safely as possible:

▶ Prepare controller for commissioning (📖 50)
▶ Creating an »Engineer« project & going online (📖 51)
▶ Parameterising the motor control (📖 52)
▶ Parameterising the application (📖 53)

▶ Saving parameter settings safe against mains failure (📖 55)
▶ Enable controller and test application (📖 55)

### 3.6.1 Prepare controller for commissioning

1. Power connection wiring
   – Refer to the mounting instructions supplied with the drive controller to find help on how to correctly design the power connections to match the requirements of your device.
2. Wire the control connections
   – The assignment for your digital inputs should correspond to one of the preconfigured control modes (C00007) for terminal control:

| Control mode | Assignment of the digital terminals DI1 | DI2 | DI3 | DI4 |
|---|---|---|---|---|
| Terminals 0 | JOG 1/3 | JOG 2/3 | DCB | Cw/Ccw |
| Terminals 2 | JOG 1/3 | JOG 2/3 | QSP | Cw/Ccw |
| Terminals 11 | Cw/Ccw | DCB | MPotUp | MPotDown |
| Terminal 16 | JOG 1/3 | JOG 2/3 | Cw/QSP | Ccw/QSP |

**Abbreviations used:**

| | |
|---|---|
| JOG | Selection of fixed setpoints 1 ... 3 parameterised in C00039/1...3 |
| DCB | Manual DC-injection braking |
| Cw/Ccw | CW/CCW rotation |
| QSP | Quick stop |
| MPotUp | Motor potentiometer: Increase speed |
| MPotDown | Motor potentiometer: Reduce speed |
| Cw/QSP | Fail-safe selection of the direction of rotation in connection with quick stop |
| Ccw/QSP | |

3. Inhibit controller: Set terminal X5/RFR to LOW level or open contact.
4. Connect USB diagnostic adapter.
5. Switch on voltage supply of the controller.
   – Without motor operation: Connect external 24 V supply.
   – With motor operation: Connect mains voltage.

If the green "DRV-RDY" LED is blinking and the red "DRV-ERR" LED is off, the controller is ready for operation and commissioning can proceed.

**Related topics:**

▶ Automatic restart after mains connection/fault... (📖 112)
▶ LED status displays (📖 582)

### 3.6.2 Creating an »Engineer« project & going online

You can find detailed information on the general use of the »Engineer« in the online help which you can call with **[F1]**.

- The chapter "Working with projects" describes, among other things, all options of the *Start-up wizard* which are available to create a new »Engineer« project.

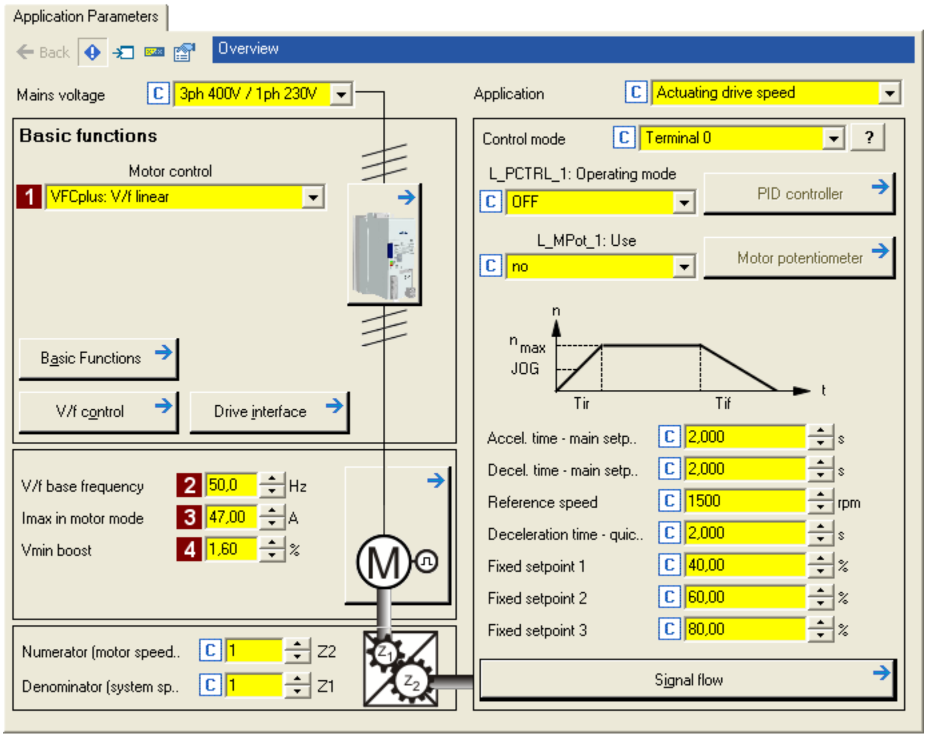
The following steps serve to describe a general method for creating a project with the **Select component from catalogue** option. For this purpose, individual components (controller, motor, etc.) are selected from selection lists.

1. Start the »Engineer«.
2. Create a new project with the *Start-up wizard* and the **Select component from catalogue** option:
   - In the **Component** step, select the 8400 TopLine controller.
   - In the **Device modules** step, select the available communication module.
   - In the **Application** step, select the "Actuating drive speed" application. (The application can also be selected any time afterwards via the **Application parameter** tab or C00005.)
   - In the **Other components** step, select other components (motor / gearbox) to be added to the project.
3. Go online.
   - After a connection to the controller has been established, the following status is displayed in the *Status line*:

     ONLINE
4. Transfer parameter set to the device.
   - This command serves to overwrite the current parameter settings in the controller with the parameter settings of the »Engineer« project.

### 3.6.3 Parameterising the motor control

1. Go to *Workspace* and change to the **Application parameters** tab.
   – The motor control parameters, among other things, can be found on the left:




2. In the **1** **Motor control** list field (C00006), select the desired motor control.
3. Adapt the motor control parameters:

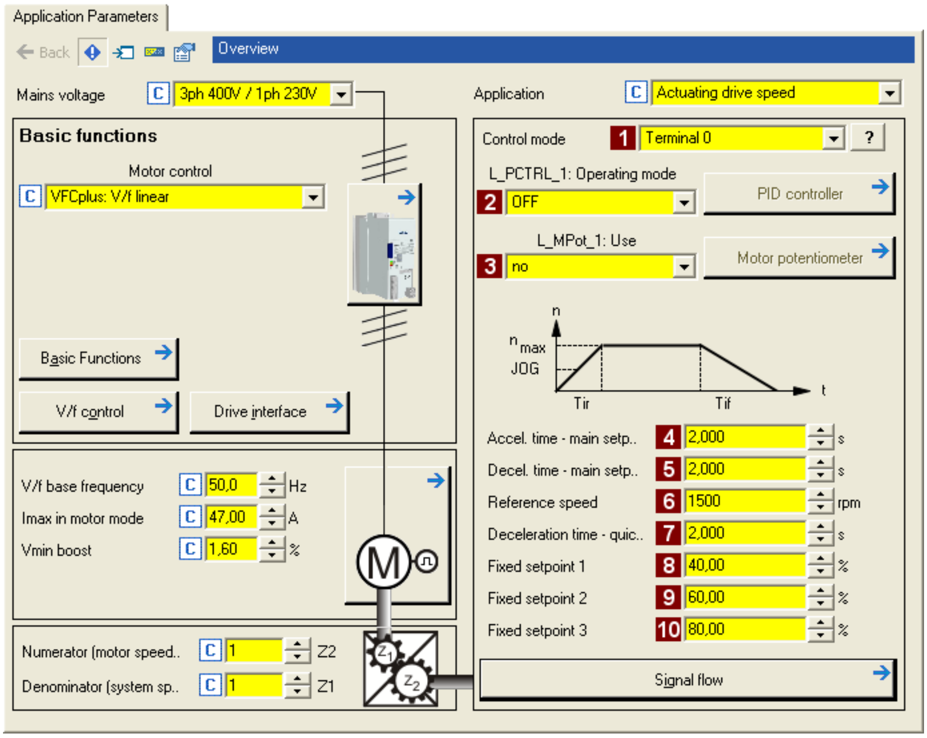
| Parameter | | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 2 | **V/f base frequency** (C00015) | 50.0 | Hz | ▸ Adapting the V/f base frequency (🕮 154) |
| 3 | **Imax in motor mode** (C00022) | 47.00 | A | ▸ Optimising the Imax controller (🕮 158) |
| 4 | **Vmin boost** (C00016) | 1.60 | % | ▸ Adapting the Vmin boost (🕮 156) |

**Related topics:**

▸ Notes on motor control (🕮 46)
▸ Motor control (MCTRL) (🕮 120)

## 3.6.4 Parameterising the application

The application parameters can be found on the right side of the **Application parameter** tab:




1. In the **1** **Control mode** list field (C00007), select the control mode suitable for the wiring of the terminals.
   – The corresponding wiring diagram is displayed in a pop-up window if you click the **?** button right to the list field.
   – For a detailed description, see the chapter "Terminal assignment of the control modes". (📖 394)
2. Optional: Use a process controller.
   – For this purpose, select the desired operating mode in the **2** **L_PCTRL_1: Operating mode** list field (C00242).
   – For a detailed description see the L_PCTRL_1 function block. (📖 1337)
   – Go to the parameterisation dialog of the process controller via the **Process controller** button.
3. Optional: Use a motor potentiometer.
   – For this purpose, select "1: On" in the **3** **L_MPot_1: Use** list field (C00806).
   – For a detailed description see the L_MPot_1 function block. (📖 1269)
   – Go to the parameterisation dialog of the motor potentiometer via the **Motor potentiometer** button.

4. Adapt the application parameters:

| Parameter | | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| 4 | Accel. time - main setpoint (C00012) | 2.000 | s | The setpoint is led via a ramp function generator with linear characteristic. The ramp function generator converts setpoint step-changes at the input into a ramp. ▸ L_NSet_1 (📖 1287) |
| 5 | Decel. time - main setpoint (C00013) | 2.000 | s | |
| 6 | Reference speed (C00011) | 1500 | rpm | All speed setpoint selections are provided in % and always refer to the reference speed set in C00011. The motor reference speed is indicated on the motor nameplate. |
| 7 | Decel. time - quick stop (C00105) | 2.000 | s | If quick stop is requested, motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0). ▸ Activate/Deactivate quick stop (📖 98) |
| 8 | Fixed setpoint 1 (C00039/1) | 40.00 | % | A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via the digital DI1 and DI2 inputs. • Fixed setpoints are selected in [%] based on the reference speed (C00011). ▸ L_NSet_1 (📖 1287) |
| 9 | Fixed setpoint 2 (C00039/2) | 60.00 | % | |
| 10 | Fixed setpoint 3 (C00039/3) | 80.00 | % | |

**Tip!**

- Click the **Signal flow** button to go down one dialog level to the signal flow of the application with further possible parameter settings. See chapter "Basic signal flow". (📖 384)
- The preconfigured I/O connection in the selected control mode can be changed via configuration parameters. See chapter "User-defined terminal assignment". (📖 370)

**More detailed information on the technology application:**

▸ TA "Actuating drive speed" (📖 383)

▸ Internal interfaces | application block "LA_NCtrl" (📖 386)

▸ Process data assignment for fieldbus communication (📖 403)

▸ Terminal assignment of the control modes (📖 394)

▸ Setting parameters (short overview) (📖 405)

▸ Configuration parameters (📖 407)

### 3.6.5 Saving parameter settings safe against mains failure

The parameter set must be saved to the device safe against mains failure to prevent parameter settings becoming lost due to mains switching.

▶ Save parameter set.

### 3.6.6 Enable controller and test application

**Stop!**

Before stipulating a speed setpoint, check whether the brake in the form of a holding brake on the motor shaft has been released!

**Note!**

If the controller is enabled and the "Inhibit at power-on" auto-start option is activated in C00142 (Lenze setting) when the mains is connected, the controller remains in the "ReadyToSwitchOn" state.

To be able to change to the "SwitchedOn" status, the controller enable must be deactivated first: Set terminal X5/RFR to LOW level.

If the controller is in the "SwitchedOn" status:

1. Enable controller: Set terminal X5/RFR to HIGH level or close contact.
   – If there is no other active source for the controller inhibit, the controller changes from the "SwitchedOn" status to the "OperationEnabled" status.
   – The **Diagnostics** tab and C00158 display all active sources for the controller inhibit.
2. Select speed setpoint.
   – In the "Terminal 0" control mode by selecting a voltage at the analog input via the setpoint potentiometer or by selecting a fixed setpoint via the digital DI1/DI2 inputs:

| DI1 | DI2 | Speed selection |
|---|---|---|
| **LOW** | **LOW** | The setpoint speed is selected via analog input 1<br>• Scaling: 10 V ≡ 100 % ≡ reference speed (C00011) |
| HIGH | LOW | Fixed setpoint 1 (C00039/1) is used as setpoint speed.<br>• Lenze setting: 40 % of the reference speed (C00011) |
| LOW | HIGH | Fixed setpoint 2 (C00039/2) is used as setpoint speed.<br>• Lenze setting: 60 % of the reference speed (C00011) |
| HIGH | HIGH | Fixed setpoint 3 (C00039/3) is used as setpoint speed.<br>• Lenze setting: 80 % of the reference speed (C00011) |

**Note!**

Observe the actual speed value (display in C00051) as well as the LED status displays. (📖 582)

**Tip!**

Other control functions in the "Terminal 0" control mode:

- DI3: HIGH level ≡ Request DC-injection braking
- DI4: HIGH level ≡ Request a change of direction of rotation

**Related topics:**

▸ "Inhibit at power-on" auto-start option (📖 112)
▸ Trouble-shooting during commissioning (📖 48)
▸ Diagnostics & error management (📖 581)

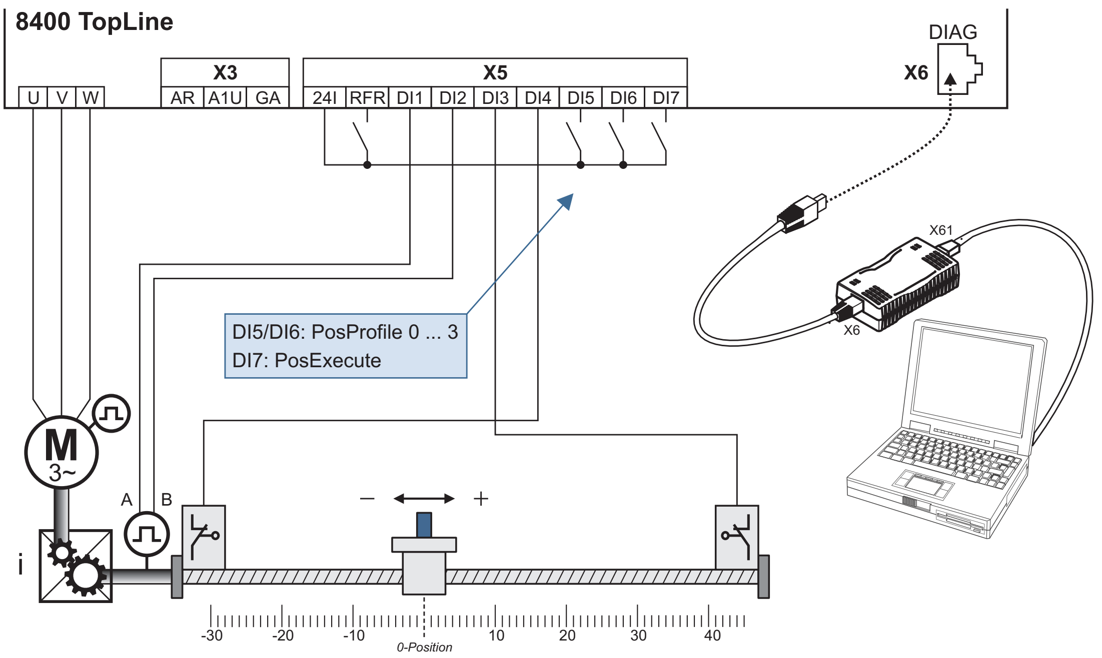
## 3.7 Commissioning of the "Table positioning" technology application

**Note!**

Take all the necessary safety precautions before you carry out the following commissioning steps and switch the device on!

▶ Safety instructions with regard to commissioning (📖 45)

### System constellation




| | |
|---|---|
| X5/RFR | Controller enable |
| X5/DI1 | Position encoder connection, track A |
| X5/DI2 | Position encoder connection, track B |
| X5/DI3 | Hardware limit switch connection in positive direction |
| X5/DI4 | Hardware limit switch connection in negative direction |
| X5/DI5/DI6 | Profile selection and change of operating mode at the same time:<br>0 \| 0 = Profile selection 0 and request for "Speed follower" operating mode<br>0 \| 1 = Profile selection 1 and request for "Homing" operating mode<br>1 \| 0 = Profile selection 2 and request for "Manual jog" operating mode<br>1 \| 1 = Profile selection 3 and request for "Positioning" operating mode |
| X5/DI7 | Start of profile generation |
| X6 | Diagnostic interface for connecting the USB diagnostic adapter |

[3-2] Block diagram for wiring the commissioning example for the "Table positioning" application

**Commissioning steps**

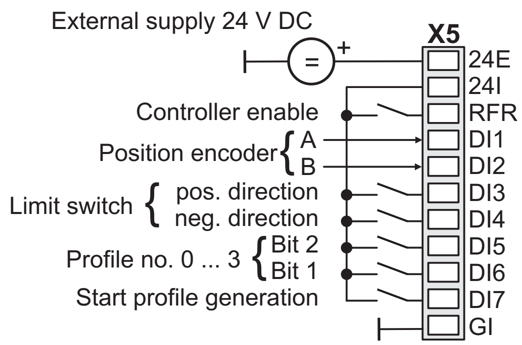
Below find a description of the commissioning steps of the "Table positioning" application shown in illustration [3-2].

Please observe the sequence of the steps in the following chapters and follow them through carefully. This will help you to commission your controller quickly and as safely as possible:


- Prepare controller for commissioning (59)
- Creating an »Engineer« project & going online (60)
- Parameterising the motor control (61)
- Parameterising the application (62)
- Saving parameter settings safe against mains failure (70)
- Enable controller and test application (70)

### 3.7.1 Prepare controller for commissioning

1. Power connection wiring
   – Refer to the mounting instructions supplied with the drive controller to find help on how to correctly design the power connections to match the requirements of your device.
2. Wire the control connections
   – The system constellation including position encoder and hardware limit switches shown in illustration [3-2] requires wiring according to control mode "Terminals 0":




3. Inhibit controller: Set terminal X5/RFR to LOW level or open contact.
4. Connect USB diagnostic adapter.
5. Switch on voltage supply of the controller.
   – Without motor operation: Connect external 24 V supply.
   – With motor operation: Connect mains voltage.

If the green "DRV-RDY" LED is blinking and the red "DRV-ERR" LED is off, the controller is ready for operation and commissioning can proceed.

**Related topics:**

▶ Automatic restart after mains connection/fault... (📖 112)
▶ LED status displays (📖 582)

### 3.7.2 Creating an »Engineer« project & going online

You can find detailed information on the general use of the »Engineer« in the online help which you can call with **[F1]**.

- The chapter "Working with projects" describes, among other things, all options of the *Start-up wizard* which are available to create a new »Engineer« project.

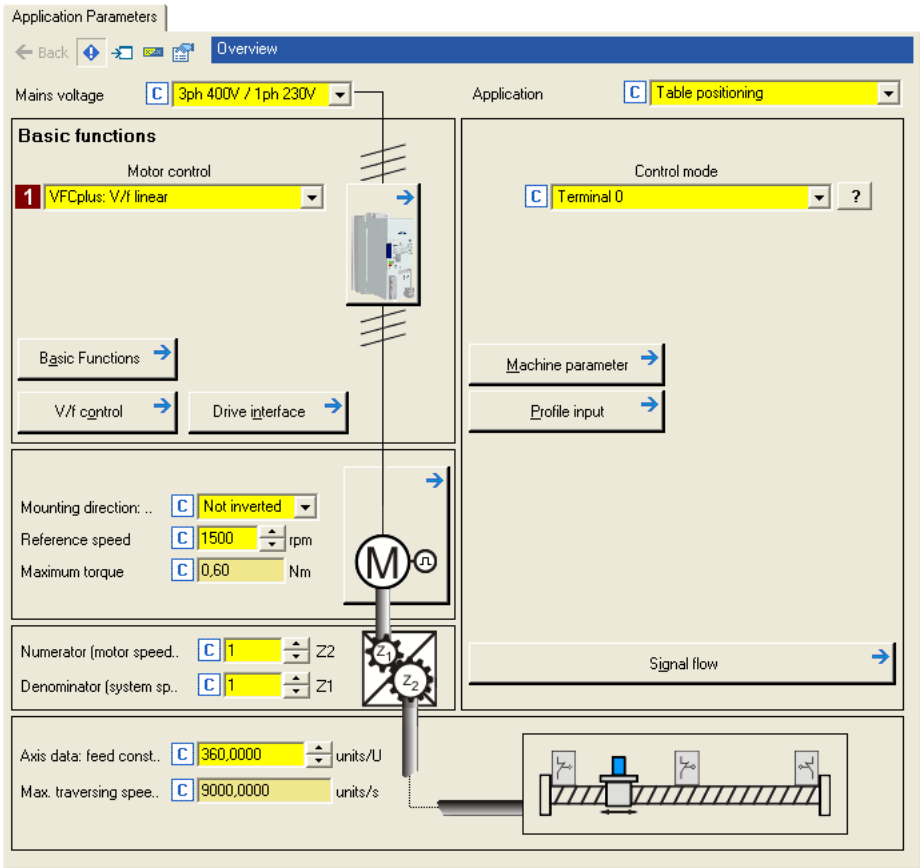
The following steps serve to describe a general method for creating a project with the **Select component from catalogue** option. For this purpose, individual components (controller, motor, etc.) are selected from selection lists.

1. Start the »Engineer«.
2. Create a new project with the *Start-up wizard* and the **Select component from catalogue** option:
   - In the **Component** step, select the 8400 TopLine controller.
   - In the **Device modules** step, select the available communication module.
   - In the **Application** step, select the "Table positioning" application. (The application can also be selected any time afterwards via the **Application parameter** tab or C00005.)
   - In the **Other components** step, select other components (motor / gearbox) to be added to the project.
3. Go online.
   - After a connection to the controller has been established, the following status is displayed in the *Status line*:


4. Transfer parameter set to the device.
   - This command serves to overwrite the current parameter settings in the controller with the parameter settings of the »Engineer« project.

### 3.7.3 Parameterising the motor control

1. Go to *Workspace* and change to the **Application parameters** tab.
   – The motor control parameters, among other things, can be found on the left:




2. In the **1** **Motor control** list field (C00006), select the desired motor control.

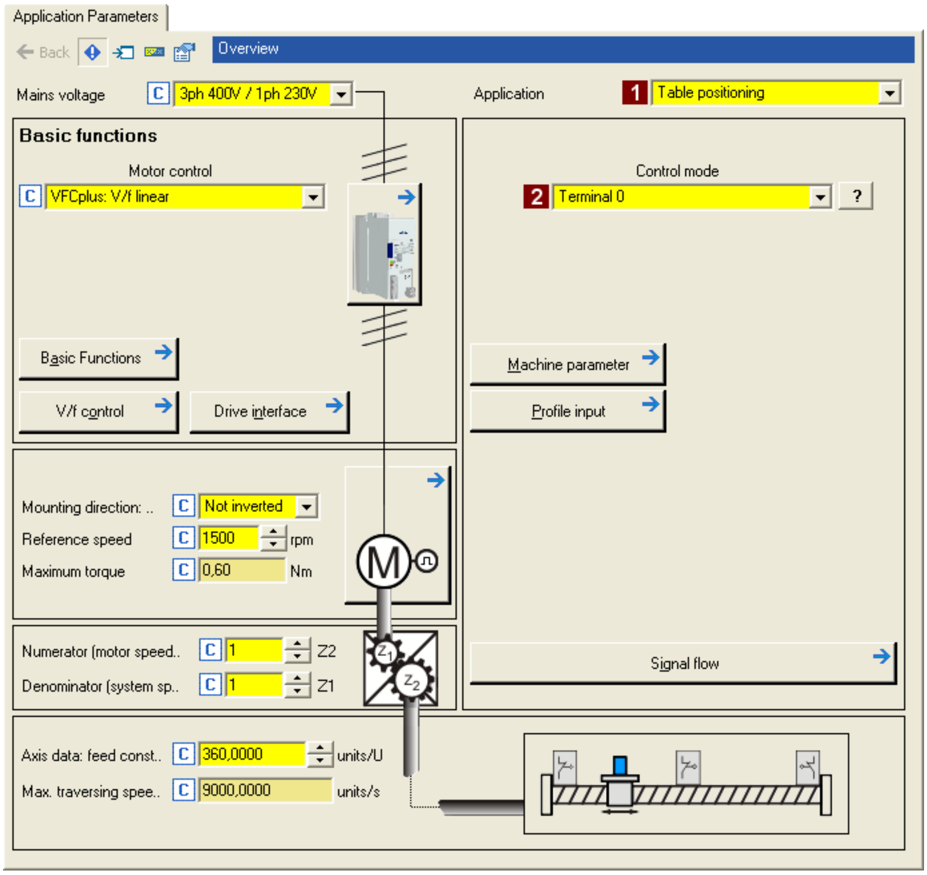
**Related topics:**

▸ Notes on motor control (📖 46)

▸ Motor control (MCTRL) (📖 120)

### 3.7.4 Parameterising the application

The application parameters can be found on the right side of the **Application parameter** tab:




1. Select "Table positioning" in the **1** **Application** list field (C00005) (if you have not already done so while creating the project).
   – After the "Table positioning" application is selected, the contents of the tab change, e.g. the **Machine parameters** and **Profile entry** buttons are shown now.
2. In the **2** **Control mode** list field (C00007), the preset "Terminals 0" control mode is already suitable for the system constellation shown in illustration [3-2] and need not be changed.
   – The corresponding wiring diagram is displayed in a pop-up window if you click the ? button right to the list field.
   – For a detailed description, see the chapter "Terminal assignment of the control modes". (📖 394)

### 3.7.4.1 Set machine parameters

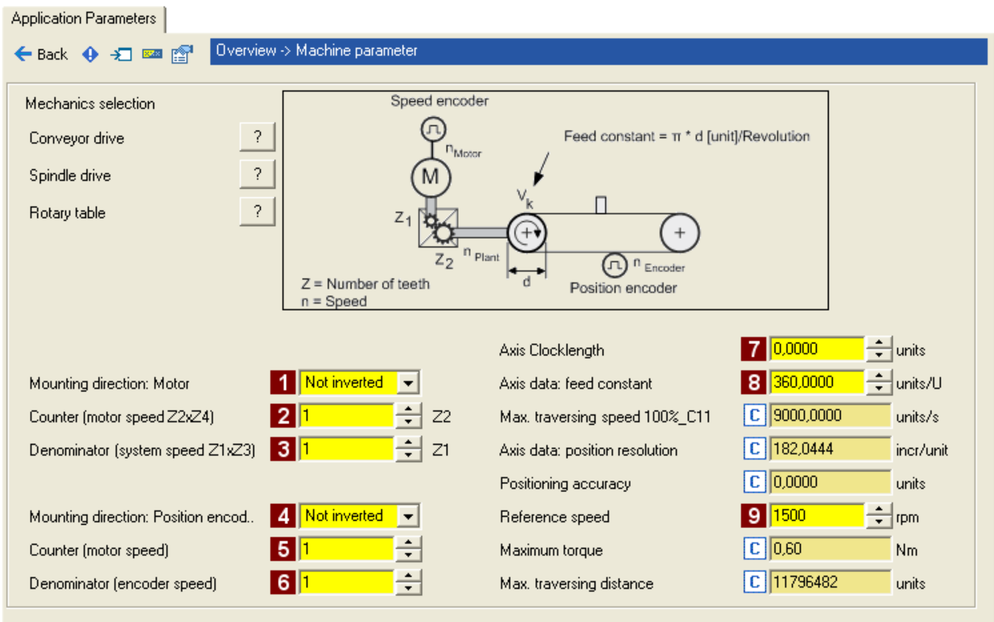
**Note!**

Setting the machine parameters is a basic prerequisite for the operating modes "Homing", "Manual jog" and "Positioning".

The more precisely the machine parameters are set, the better the results of positioning!

Detailed information on the machine parameters is provided in chapter "Basic drive functions" in subchapter "Machine parameters". (📖 498)

1. Go to the right side of the **Application parameter** tab and click the **Machine parameter** button to change to the *Overview* ➜ *Machine parameter* dialog level:




2. Set the machine parameters according to the system constellation at hand.

| | Parameter | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| 1 | **Mounting direction: Motor** (C01206/1) | not inverted | | Inversion if motor mounting is mirrored. |
| 2 | **Numerator (Z2)** (C01202/1) | 1 | | Gearbox factor - motor<br>• Entry of the gearbox factor as numerator/denominator ratio (numerator = motor speed and denominator = output speed of gearbox) or from the number of teeth of the gearbox arrangement. |
| 3 | **Denominator (Z1)** (C01202/2) | 1 | | |
| 4 | **Mounting direction: Position encoder** (C01206/2) | not inverted | | Inversion if position encoder mounting is mirrored. |

| Parameter | | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| 5 | **Numerator** (C01203/1) | 1 | | Gearbox factor - position encoder<br>• Entry of the gearbox factor as numerator/denominator ratio, with numerator = motor speed and denominator = position encoder speed. |
| 6 | **Denominator** (C01203/2) | 1 | | |
| 7 | **Axis cycle** (C01201/1) | 0.0000 | units | Cycle for Modulo measuring system<br>• The Modulo system is activated by setting a cycle (C01201/1) > 0 units.<br>• When the cycle (C01201/1) is set to 0 units (Lenze setting), the traversing range is unlimited (classical measuring system). |
| 8 | **Feed constant** (C01204) | 360.0000 | units/rev. | The feed constant corresponds to the machine movement for one revolution of the gearbox output shaft.<br>• The value is entered in application units referred to one revolution. |
| 9 | **Reference speed** (C00011) | 1500 | rpm | All speed setpoint selections are provided in % and always refer to the reference speed set in C00011.<br>The motor reference speed is indicated on the motor nameplate. |

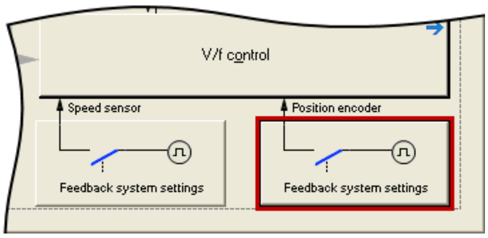
3. After setting the machine parameters, click the **Back** button to change to the *Overview* dialog level.

### 3.7.4.2 Set the position encoder

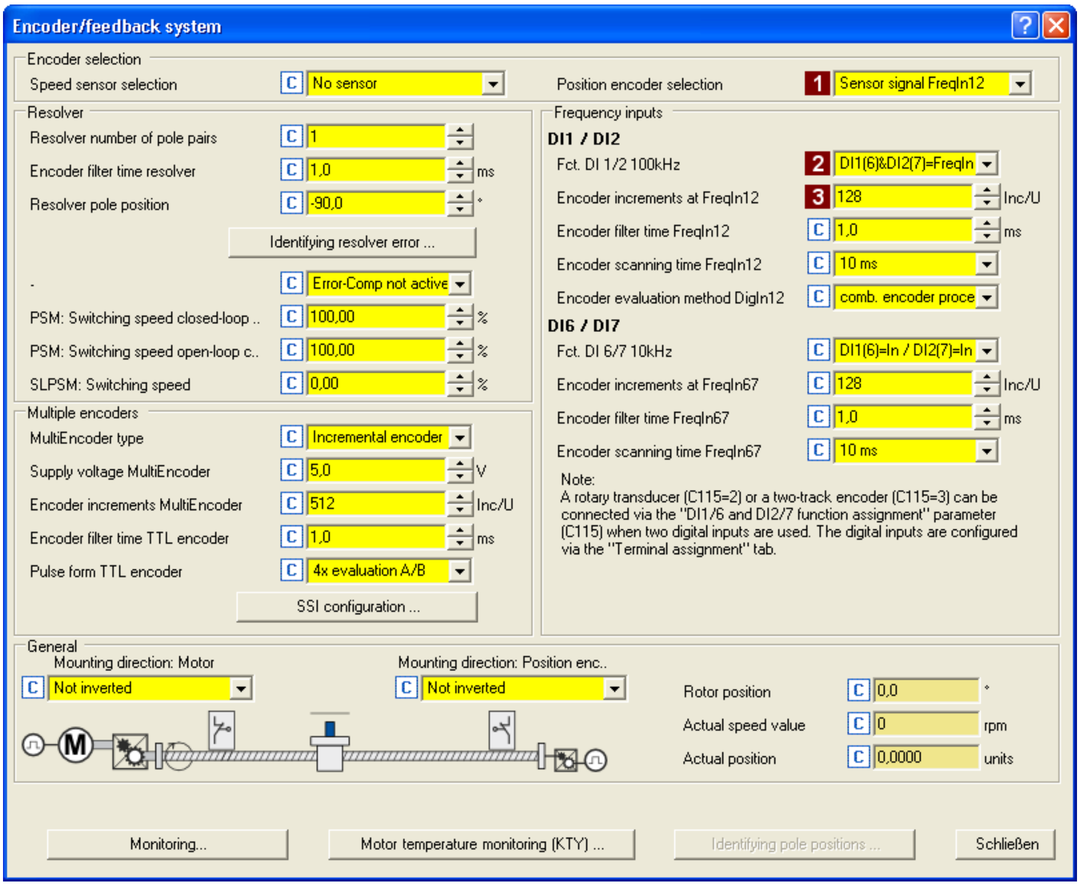
In the Lenze setting, the digital input terminals are configured as "normal" digital inputs. Since in this system constellation DI1 and DI2 are used to connect a two-track position encoder, the function assignment of these two inputs must be changed accordingly.

Detailed information on how to use a resolver/encoder as motor or position encoder can be found in the main chapter "Encoder/feedback system". (📖 293)

1. Go to the right side of the **Application parameter** tab and click the **Signal flow** button to change to the *Overview* → *Signal flow* dialog level.
2. Click the **Feedback system settings** button in the signal flow:



The *Encoder/Feedback system* parameterisation dialog will be shown:



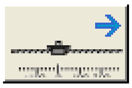

3. In the **1** **Position encoder selection** list field (C00490), select "Encoder signal FreqIn12".
4. In the **2** **Fct. DI 1/2 100kHz** list field (C00115/1), select "DI1(6)&DI2(7)=FreqIn (2-track)".
   – This selection ensures that the digital DI1 and DI2 input terminals are configured as frequency inputs.
5. Set the number of position encoder increments in the **3** **Number of encoder increments at FreqIn12** input field (C00420/1).
6. Click the **Close** button to close the parameterisation dialog again.

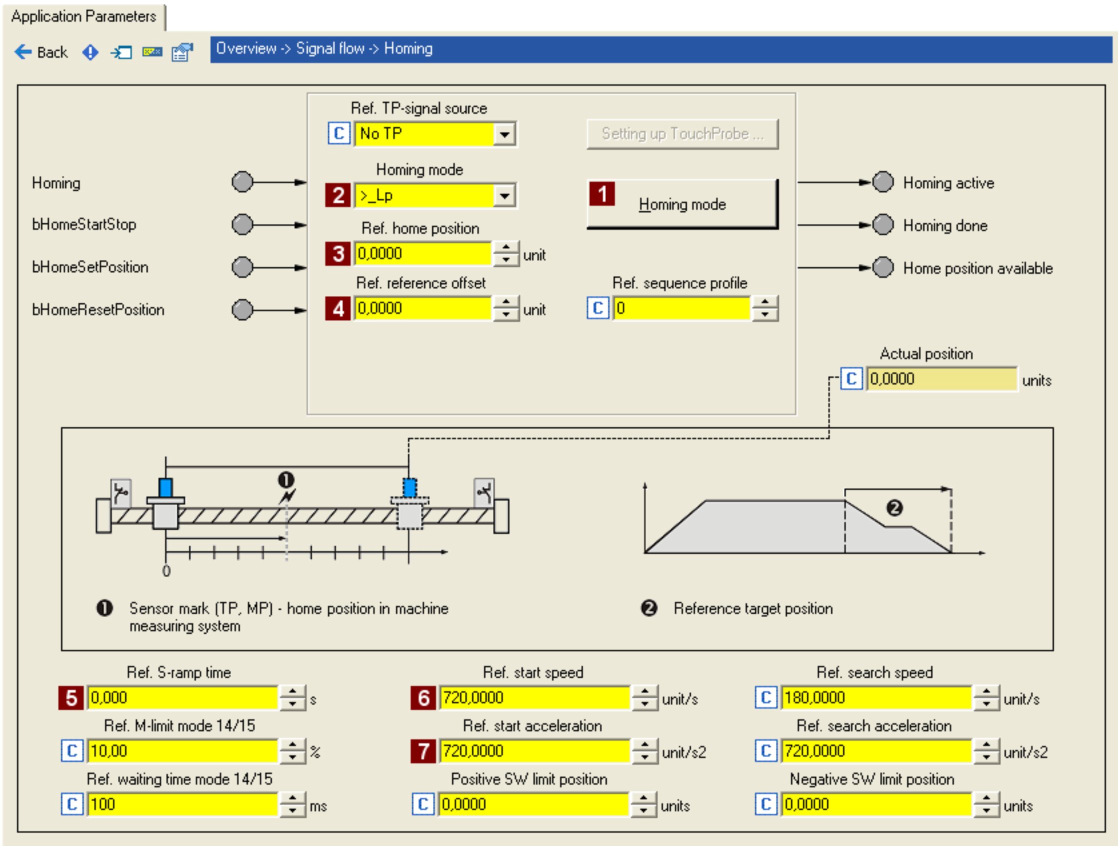
**Related topics:**


- Digital input terminals (📖 329)
- Using DI1(6) and DI2(7) as frequency inputs (📖 333)
- Encoder/feedback system (📖 293)

### 3.7.4.3 Set homing

Detailed information on the "Homing" operating mode is provided in chapter "Basic drive functions" in subchapter "Homing". (📖 517)

1. Click the  button (in the "MotionControlKernel" block) in the signal flow to change to the *Overview* ➞ *Signal flow* ➞ *Homing* dialog level:




2. Select the homing mode and hence the type of homing via the **1** **Homing mode** button or directly in the **2** **Homing mode** list field (C01221).
   – A graphic display of all possible homing modes can be seen via the **Homing mode** button.
   – In this system constellation without touch probe, the preset ">_Lp" mode can be used: movement in positive direction, positive edge of the limit switch sets reference.

3. Set the position to be used for loading the set position and the actual position after homing has finished in the **3** **Homing home position** input field (C01227/2).

4. Recommendation: Set an offset in the 4 **Homing home offset** input field (C01227/1) to prevent the machine from stopping at an activated limit switch.
   – Background: The load machine can also leave the travel range limit switch. There follows a return to the home position that was set with the positive edge of the travel range limit switch.
5. Set suitable profile parameters for the homing:

| | Parameter | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| 5 | **S-ramp time** (C01226/1) | 0.000 | s | S-ramp time for reference search/homing.<br>• Setting "0 s" ≡ no rounding |
| 6 | **Start speed** (C01224/1) | 720.0000 | unit/s | Starting speed for approaching the pre-stop signals. |
| 7 | **Start acceleration** (C01225/1) | 720.0000 | unit/$s^2$ | Starting acceleration for the starting speed ramps. |

6. After setting the parameters for the homing, click the **Back** button to change back to the *Overview* → *Signal flow* dialog level.

### 3.7.4.4 Enter one or more profiles

You can find detailed information on entering profiles in the chapter entitled "Basic drive functions", subchapter "Profile entry". (📖 545)

1. Click the **Profile entry** button (in the "MotionControlKernel" block) in the signal flow to change to the *Overview* ➞ *Signal flow* ➞ *Profile entry* dialog level:


**Note!**

In the Lenze setting, the profiles are assigned to certain operating modes, i.e. the selection of a profile also results in a change of operating mode:

- If profile 0 is stipulated: Activation of "Speed follower" operating mode
- If profile 1 is stipulated: Activation of "Homing" operating mode
- If profile 2 is stipulated: Activation of "Manual jog" operating mode
- If profile 3 ... 15 is stipulated: Activation of "Positioning" operating mode

2. Select number 3 in the 1 **Profile number** list field to enter a positioning profile which can be selected via digital inputs DI5 and DI6.

3. Set the profile parameters:

| Parameter | | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| 2 | **Profile x: Mode** (C01300/x) | absolute (shortest way)t | | Selection of the way in which positioning is to be carried out. ▶ Positioning modes (📖 548) |
| 3 | **Profile x: Position** (C01301/x) | 360.0000 | unit | Target position or distance to be traversed. |
| 4 | **Profile x: Speed** (C01302/x) | 360.0000 | unit/s | Maximum speed at which the target is to be approached. |
| 5 | **Profile x: Acceleration** (C01303/x) | 720.0000 | unit/$s^2$ | Stipulation of the change in speed at which maximum acceleration is to take place. |
| 6 | **Profile x: Deceleration** (C01304/x) | 720.0000 | unit/$s^2$ | Stipulation of the change in speed at which maximum deceleration to a standstill is to take place. |
| 7 | **Profile x: S-ramp time** (C01306/x) | 0.000 | s | Due to stipulation of an S-ramp time for a profile, the profile is executed with S-shaped ramps, i.e. acceleration and braking processes are initiated smoothly in order to reduce jerk and thus the stress on the drive components. |
| 8 | **Profile x: Final speed** (C01305/x) | 0.0000 | unit/s | This specifies the speed at which the drive is to start the next profile after reaching the target position. |
| 9 | **Profile x: Sequence profile** (C01307/x) | 0 | | Optional: Sequence profile for profile linkage / following block control.<br>After execution of the profile (target position reached), the set following (subsequent) profile is started automatically. In this way, profile chains can be stipulated without additional control processes. |
| 10 | **Profile x: TP profile** (C01308/x) | 0 | | Optional: Profile number of the profile (1 ... 15) that is to be executed after a touch probe has been detected.<br>• If "0" is set, there will be no profile stepping through touch probe.<br>• Only relevant for positioning modes with touch-probe. |
| 11 | **Profile x: TP source** (C01308/x) | TP-DigIn3 | | Optional: Selection of the signal source for touch probe detection.<br>• Only relevant for positioning modes with touch-probe. |

4. After entering the profile, click the **Back** button to change back to the *Overview* → *Signal flow* dialog level.

**Related topics:**

▶ Touch probe detection (📖 362)
▶ Operating mode change with profile number (📖 495)
▶ Positioning (📖 541)

**More detailed information on the technology application:**

- TA "Table positioning" (📖 410)
- Basic signal flow (📖 411)
- Internal interfaces | Application block "LA_TabPos" (📖 414)
- Process data assignment for fieldbus communication (📖 434)
- Terminal assignment of the control modes (📖 424)
- Setting parameters (short overview) (📖 436)
- Configuration parameters (📖 436)

### 3.7.5 Saving parameter settings safe against mains failure

The parameter set must be saved to the device safe against mains failure to prevent parameter settings becoming lost due to mains switching.

- Save parameter set.

### 3.7.6 Enable controller and test application

**Stop!**

Before stipulating a speed setpoint, check whether the brake in the form of a holding brake on the motor shaft has been released!

**Note!**

If the controller is enabled and the "Inhibit at power-on" auto-start option is activated in C00142 (Lenze setting) when the mains is connected, the controller remains in the "ReadyToSwitchOn" state.

To be able to change to the "SwitchedOn" status, the controller enable must be deactivated first: Set terminal X5/RFR to LOW level.

If the controller is in the "SwitchedOn" status:

1. Enable controller: Set terminal X5/RFR to HIGH level or close contact.
   - If there is no other active source for the controller inhibit, the controller changes from the "SwitchedOn" status to the "OperationEnabled" status.
   - The **Diagnostics** tab and C00158 display all active sources for the controller inhibit.
2. Select the respective control signals via the digital inputs (see sections below).

**Note!**

Observe the actual speed value (display in C00051) as well as the LED status displays. (📖 582)

**Homing**

1. Requesting the "Homing" operating mode:
   – Set digital input DI5 to LOW level and digital input DI6 to HIGH level to select profile 1 and request the "Homing" operating mode at the same time.
2. Start homing:
   – Set digital input DI7 to HIGH level ("Alternative function PosExecute").
3. Conclude homing when homing position has been reached:
   – Reset digital input DI7 to LOW level.

**Positioning**

1. Requesting the "Positioning" operating mode:
   – Set digital inputs DI5 and DI6 to HIGH level to select profile 3 and request the "Positioning" operating mode at the same time.
2. Start positioning:
   – Set digital input DI7 to HIGH level ("Alternative function PosExecute").
3. Conclude positioning when the target position has been reached:
   – Reset digital input DI7 to LOW level.

**Tip!**

The active operating mode (C01243) in the signal flow can also be controlled by means of the setting of the switch in the "Motion Control Kernel" block.

**Related topics:**


- "Inhibit at power-on" auto-start option (📖 112)
- Trouble-shooting during commissioning (📖 48)
- Diagnostics & error management (📖 581)

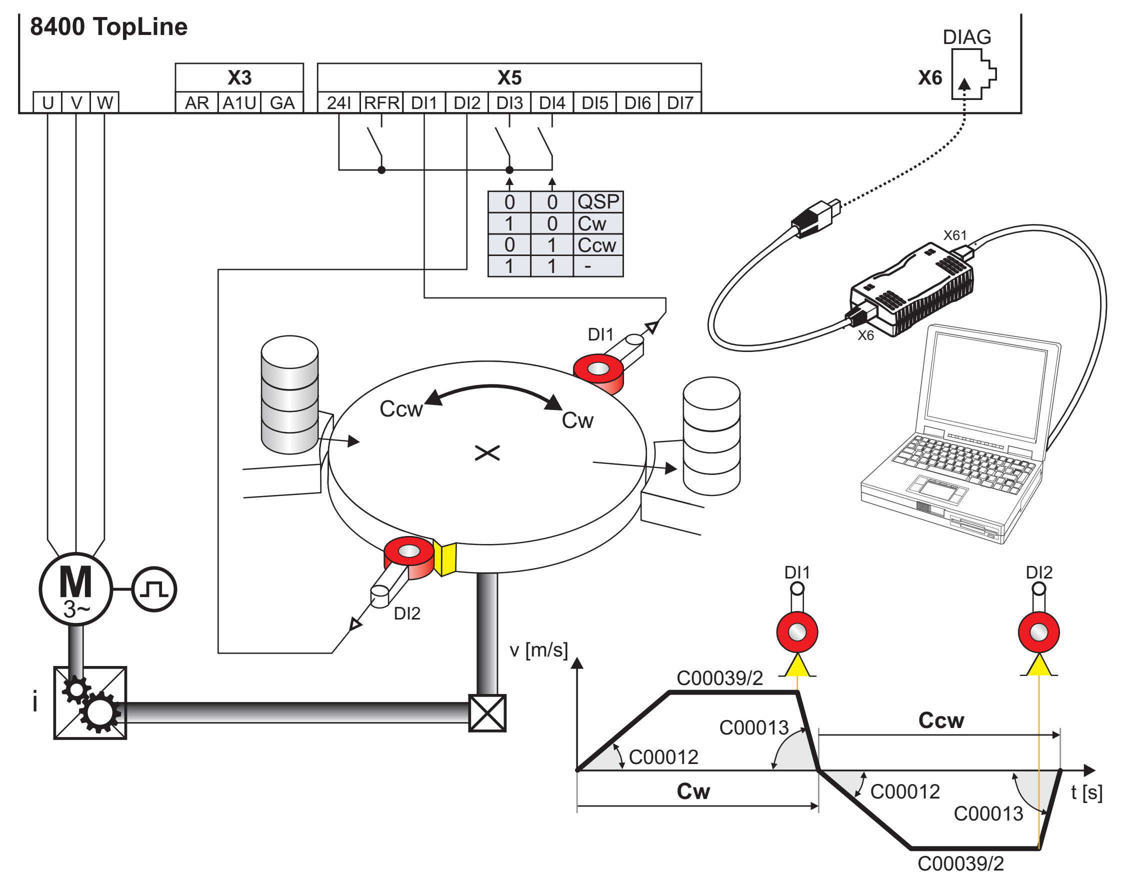
## 3.8 Commissioning of the "Switch-off positioning" technology application

**Note!**

Take all the necessary safety precautions before you carry out the following commissioning steps and switch the device on!

▶ Safety instructions with regard to commissioning (📖 45)

### System constellation




| | |
|---|---|
| X4/RFR | Controller enable |
| X5/DI1 | Initiator connection for stop function 1 |
| X5/DI2 | Initiator connection for stop function 2 |
| X5/DI3 | Quick stop for CW rotation / selection of switch-off position 1 |
| X5/DI4 | Quick stop for CCW rotation / selection of switch-off position 2 |
| X6 | Diagnostic interface for connecting the USB diagnostic adapter |

[3-3] Block diagram for wiring of the commissioning example for the "Switch-off positioning" application

### Functional principle of a switch-off positioning without pre-switch off

In case of the switch-off positioning without pre-switch off shown above, it makes sense to use the "Terminals 2" control mode:

1. Set DI3 to HIGH level to activate CW rotation.
2. The drive accelerates along the acceleration ramp (C00012) up to the traversing speed set in C00039/2.
3. After reaching the DI1 contact, the drive comes to a stop with quick stop (QSP) in the target position.
4. Reset DI3 to LOW level and set DI4 to HIGH level to activate CCW rotation now.
5. The drive is accelerated along the acceleration ramp (C00012) up to the traversing speed set in C00039/2.
6. After reaching the DI2 contact, the drive comes to a stop in the initial position with quick stop (QSP).

**Tip!**

- In order to avoid positioning inaccuracy due to signal propagation delays, the initiators can be directly evaluated by the drive controller. Limit switch evaluation can be configured in the drive controller. In code C00488/x you can change the method of detecting position signals from level evaluation to edge evaluation.
- In order to prevent unintended movements of the load in the target position, the use of a holding brake is recommended as an alternative to DC-injection braking (limited torque).
- The device terminals and their function assignment do not appear in the FB Editor. The assignment of (hardware) terminals to (software) functions is explained in the chapter "Terminal assignment of the control modes". (📖 451)

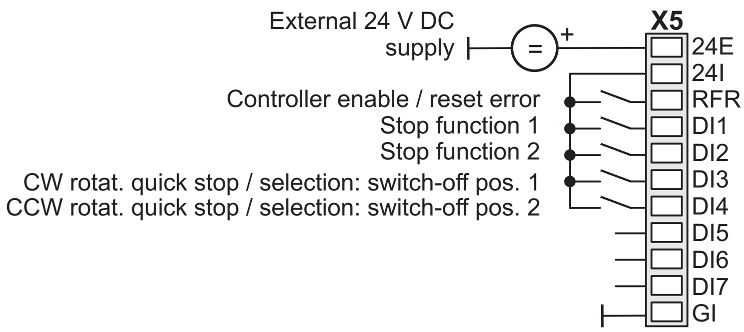
### Commissioning steps

As shown in illustration [3-3], below find a description of the commissioning steps of the "Switch-off positioning" application without pre-switch off.

Please observe the sequence of the steps in the following chapters and follow them through carefully. This will help you to commission your controller quickly and as safely as possible:

### 3.8.1 Prepare controller for commissioning

1. Power connection wiring
   - Refer to the mounting instructions supplied with the drive controller to find help on how to correctly design the power connections to match the requirements of your device.
2. Wire the control connections
   - In case of the application shown in illustration [3-3], switch-off positioning without pre-switch off, wiring according to the "Terminals 2" control mode makes sense:




3. Inhibit controller: Set terminal X5/RFR to LOW level or open contact.
4. Connect USB diagnostic adapter.
5. Switch on voltage supply of the controller.
   - Without motor operation: Connect external 24 V supply.
   - With motor operation: Connect mains voltage.

If the green "DRV-RDY" LED is blinking and the red "DRV-ERR" LED is off, the controller is ready for operation and commissioning can proceed.

**Related topics:**

- Automatic restart after mains connection/fault... (📖 112)
- LED status displays (📖 582)

### 3.8.2 Creating an »Engineer« project & going online

You can find detailed information on the general use of the »Engineer« in the online help which you can call with **[F1]**.

- The chapter "Working with projects" describes, among other things, all options of the *Start-up wizard* which are available to create a new »Engineer« project.

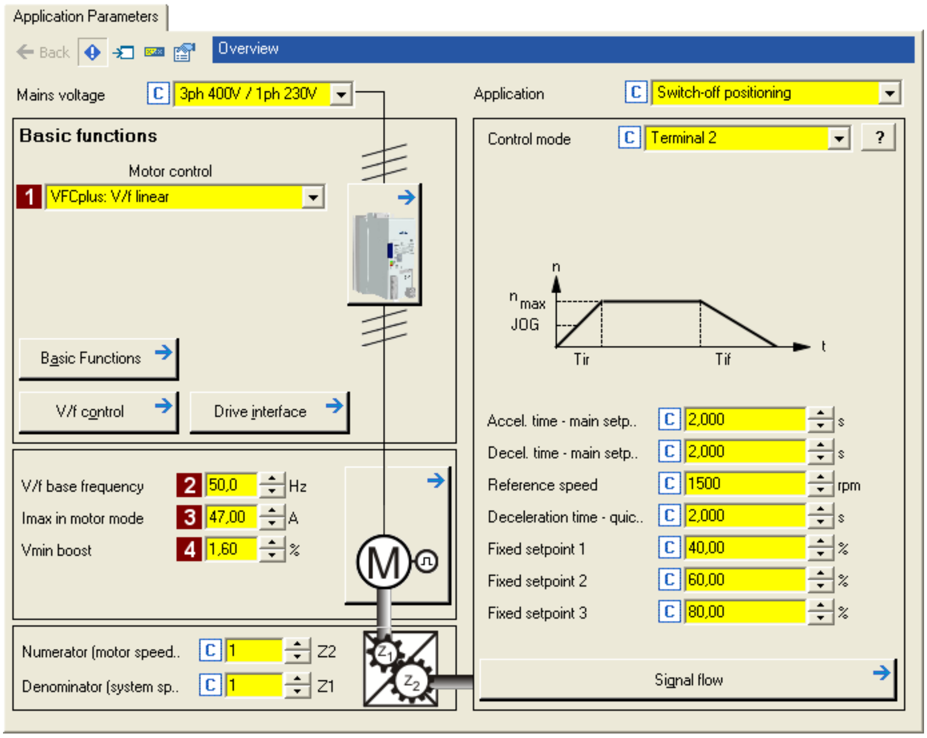
The following steps serve to describe a general method for creating a project with the **Select component from catalogue** option. For this purpose, individual components (controller, motor, etc.) are selected from selection lists.

1. Start the »Engineer«.
2. Create a new project with the *Start-up wizard* and the **Select component from catalogue** option:
   - In the **Component** step, select the 8400 TopLine controller.
   - In the **Device modules** step, select the available communication module.
   - In the **Application** step, select the "Switch-off positioning" application. (The application can also be selected any time afterwards via the **Application parameter** tab or C00005.)
   - In the **Other components** step, select other components (motor / gearbox) to be added to the project.
3. Go online.
   - After a connection to the controller has been established, the following status is displayed in the *Status line*:


4. Transfer parameter set to the device.
   - This command serves to overwrite the current parameter settings in the controller with the parameter settings of the »Engineer« project.

### 3.8.3 Parameterising the motor control

1. Go to *Workspace* and change to the **Application parameters** tab.
   – The motor control parameters, among other things, can be found on the left:




2. In the **1** **Motor control** list field (C00006), select the desired motor control.
3. Adapt the motor control parameters:

| Parameter | | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| **2** | **V/f base frequency** (C00015) | 50.0 | Hz | ▸ Adapting the V/f base frequency (📖 154) |
| **3** | **Imax in motor mode** (C00022) | 47.00 | A | ▸ Optimising the Imax controller (📖 158) |
| **4** | **Vmin boost** (C00016) | 1.60 | % | ▸ Adapting the Vmin boost (📖 156) |

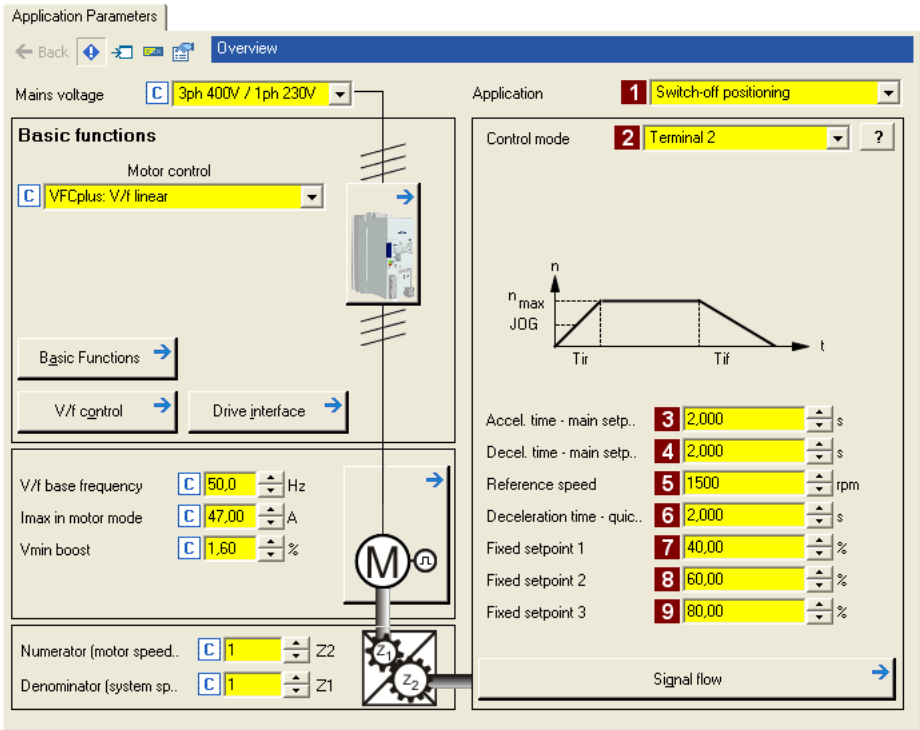
**Related topics:**

▸ Notes on motor control (📖 46)

▸ Motor control (MCTRL) (📖 120)

### 3.8.4 Parameterising the application

The application parameters can be found on the right side of the **Application parameter** tab:




1. In the 1 **Application** list field (C00005), select the "Switch-off positioning" application (if you have not already done so while creating the project).
   - After the "Switch-off positioning" application is selected, the contents of the tab change, e.g. the **Process controller** and **Motor potentiometer** buttons are not shown any more.
2. In the 2 **Control mode** list field (C00007) and in case of illustration [3-3], for the shown switch-off positioning without pre-switch off the "Terminals 2" control mode must be selected.
   - The corresponding wiring diagram is displayed in a pop-up window if you click the ? button right to the list field.
   - For a detailed description, see the chapter "Terminal assignment of the control modes". (📖 394)

3. Adapt the application parameters:

| | Parameter | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| 3 | **Accel. time - main setpoint** (C00012) | 2.000 | s | The setpoint is led via a ramp function generator with linear characteristic. The ramp function generator converts setpoint step-changes at the input into a ramp. Note: These settings only apply if no other ramp times have been selected at the L_NSet FB! |
| 4 | **Decel. time - main setpoint** (C00013) | 2.000 | s | |
| 5 | **Reference speed** (C00011) | 1500 | rpm | All speed setpoint selections are provided in % and always refer to the reference speed set in C00011. The motor reference speed is indicated on the motor nameplate. |
| 6 | **Decel. time - quick stop** (C00105) | 2.000 | s | If quick stop is requested, motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0). ▸ Activate/Deactivate quick stop (📖 98) |
| 7 | **Fixed setpoint 1** (C00039/1) | 40.00 | % | Fixed setpoints are selected in [%] based on the reference speed (C00011). **Fixed setpoint 2 must be less than fixed setpoint 3!** Otherwise, the drive will start at a low speed and accelerate after the pre-switch off. |
| 8 | **Fixed setpoint 2** (C00039/2) | 60.00 | % | |
| 9 | **Fixed setpoint 3** (C00039/3) | 80.00 | % | |

**Tip!**

- Click the **Signal flow** button to go down one dialog level to the signal flow of the application with further possible parameter settings. See chapter "Basic signal flow". (📖 442)
- The preconfigured I/O connection in the selected control mode can be changed via configuration parameters. See chapter "User-defined terminal assignment". (📖 370)
- Low-jerk traversing profiles can be implemented by means of S-shaped ramps.
- In the case of high breakaway torques combined with horizontal motion sequences, "Sensorless vector control (SLVC)" can be used as motor control (C00006).
- For reversal of rotation direction (bidirectional motion), comprehensive configuration options are available in the drive controller (e.g. by means of the L_DFlipFlop function block).

**More detailed information on the technology application:**

▸ TA "Switch-off positioning" (📖 440)

▸ Internal interfaces | application block "LA_SwitchPos" (📖 443)

▸ Process data assignment for fieldbus communication (📖 460)

▸ Terminal assignment of the control modes (📖 451)

▸ Setting parameters (short overview) (📖 462)

▸ Configuration parameters (📖 464)

### 3.8.5 Saving parameter settings safe against mains failure

The parameter set must be saved to the device safe against mains failure to prevent parameter settings becoming lost due to mains switching.

▶ Save parameter set.

### 3.8.6 Enable controller and test application

**Stop!**

Before stipulating a speed setpoint, check whether the brake in the form of a holding brake on the motor shaft has been released!

**Note!**

If the controller is enabled and the "Inhibit at power-on" auto-start option is activated in C00142 (Lenze setting) when the mains is connected, the controller remains in the "ReadyToSwitchOn" state.

To be able to change to the "SwitchedOn" status, the controller enable must be deactivated first: set terminal X5/RFR to LOW level.

If the controller is in the "SwitchedOn" status:

1. Enable controller: Set terminal X5/RFR to HIGH level or close contact.
   - If there is no other active source for the controller inhibit, the controller changes from the "SwitchedOn" status to the "OperationEnabled" status.
   - The **Diagnostics** tab and C00158 display all active sources for the controller inhibit.
2. Select the respective control signals via the digital inputs.

**Note!**

Observe the actual speed value (display in C00051) as well as the LED status displays. (📖 582)

**Related topics:**

- "Inhibit at power-on" auto-start option (📖 112)
- Trouble-shooting during commissioning (📖 48)
- Diagnostics & error management (📖 581)

## 3.9 PC manual control

For test and demonstration purposes, PC manual control can be used to manually control various drive functions via the »Engineer« when an online connection has been established.

**Supported drive functions:**

- Speed control (follow a speed setpoint)
- Activate/Deactivate quick stop
- Set/reset home position
- Manual jog
- Positioning (relative or absolute)

**Other control functions:**

- Reset error messages
- Set digital/analog outputs (in preparation)

**Diagnostic functions:**

- Display of the actual speed value and motor current (in a temporal characteristic)
- Display of the current device status
- Display of the status determining error
- Display of the status of the digital/analog inputs (in preparation)

### 3.9.1 Activating PC manual control

**Stop!**

PC manual control must be explicitly activated by the user.

If PC manual control is activated, the controller is inhibited via device command (C00002/16) first.

**Note!**

**For activated PC manual control:**

The online connection between the PC and the controller is monitored by the controller.

- If the online connection is interrupted for longer than the set timeout (Lenze setting: 2 s):
  - Error response "Fault" is triggered, i.e. the motor becomes torqueless and is coasting unless it already is at standstill.
  - The "Ck16: Time overflow manual control" error message is entered into the logbook.

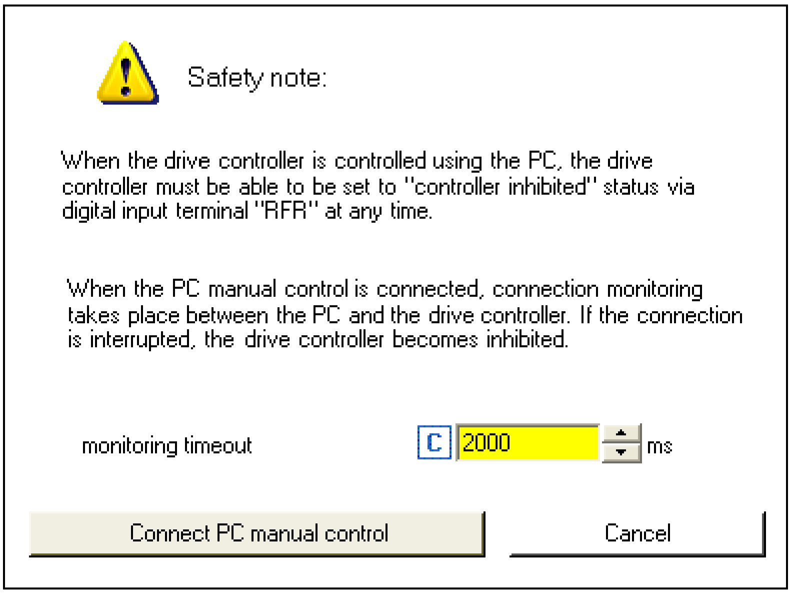
PC manual control provides the **Motion Control Kernel** and the motor interface with all required control signals and setpoint signals.

- The existing application (function block interconnection) is now decoupled from these interfaces but will be processed as before and remains unchanged.
- It does not matter what type of motor control is set in C00006.

**How to activate PC manual control:**

1. If an online connection to the controller has not been established yet:

   Go online.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Go to the *Overview* dialog level and click the "**PC manual control**" button.
   - The following safety note is displayed first:

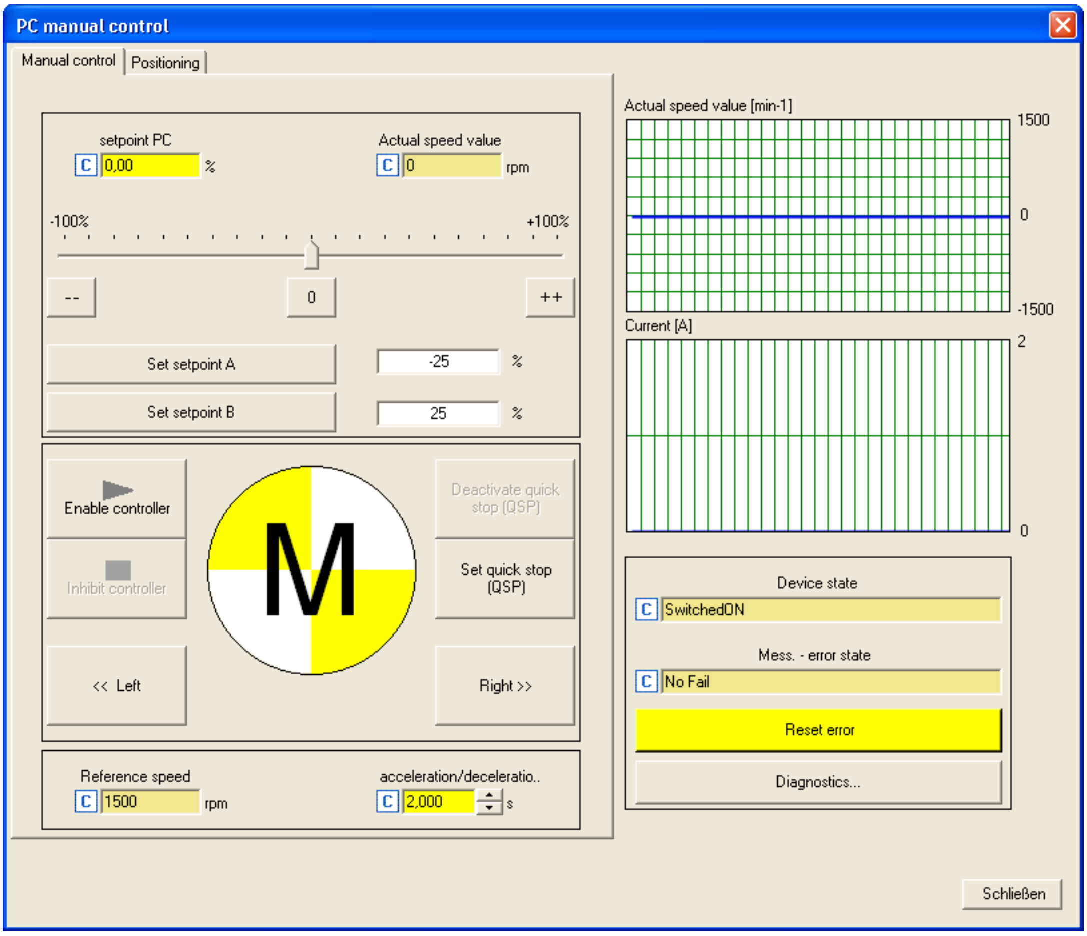



   - Click the **Cancel** button to abort the action and close the dialog box.
   - The **Timeout monitoring** input field serves to adapt the timeout for monitoring the connection between the PC and the controller.

4. To acknowledge the note and activate PC manual control:
   Click the **Activate PC manual control** button.
   - The controller is inhibited via device command (C00002/16).
   - The *PC manual control* operator dialog is displayed.

### PC manual control operator dialog

On the left-hand side, the *PC manual control* operator dialog includes several tabs which serve to select various control functions. On the right-hand side, setpoint and status displays are provided for diagnostic purposes:

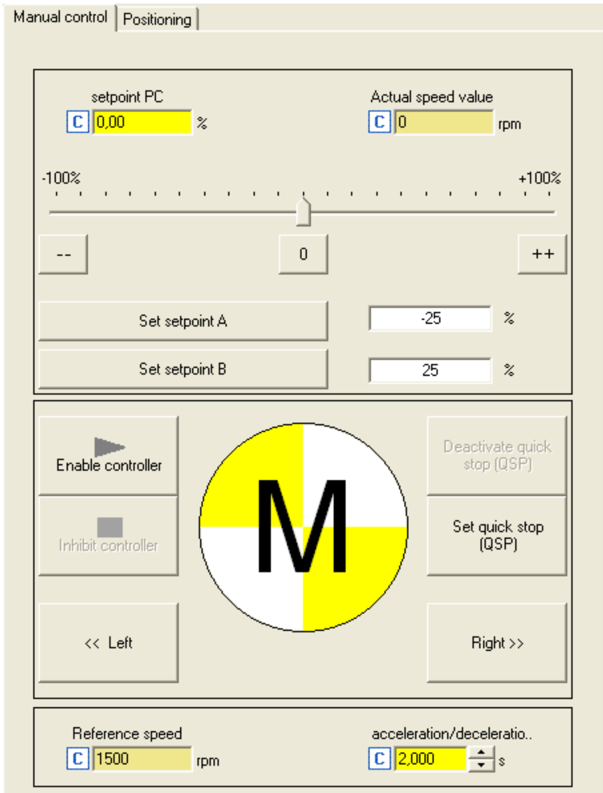



> **Note!**
>
> PC manual control can be exited any time by clicking the **Close** button.
>
> If you exit PC manual control or change to another tab, the controller is inhibited via device command (C00002/16), i.e. the motor becomes torqueless and is coasting unless it already is at standstill.

The execution of the various functions is described in the following chapters.

### 3.9.2 Speed control

Via the **Speed control** tab, simply make the drive rotate in the "Speed follower" operating mode without the need to set control parameters or feedback systems:

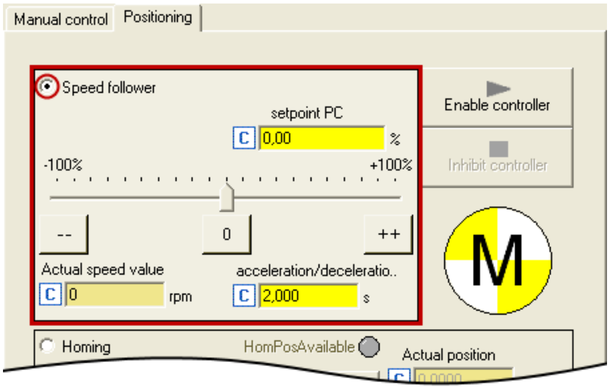



**How to make the motor rotate in its most basic way:**

1. Set the desired speed setpoint in [%] based on the reference speed, e.g. directly in the **Setpoint PC** input field or via the slider.
   - Via the **--** / **0** / **++** buttons, the currently set speed setpoint can be reduced/increased in steps of 10 percent or set to zero.
   - Via the **Set setpoint A/B** buttons, the speed setpoint can be set to a previously set constant value A/B.
2. To start the speed follower:
   Enable the controller via the **Enable controller** button.
   - Please observe that the controller will not be enabled if other sources of controller inhibit (e.g. terminal RFR) are active.
   - The enabled controller is now following the selected speed setpoint.
   - To avoid shocks or overload in case of great setpoint changes, the speed setpoint follows a linear ramp generator with adjustable acceleration/deceleration time.
   - Via the **Inhibit controller** button, the controller can be inhibited again, i.e. the motor becomes torqueless and is coasting unless it already is at standstill.

**Further functions:**

- If the **Set quick stop (QSP)** button is clicked, the motor is braked to a standstill within the deceleration time parameterised in C00105.
  – Via the **Deactivate quick stop (QSP)** button, the quick stop can be deactivated.
- Via the **<< CCW** and **CW >>** buttons, the direction of rotation can be changed.
- The "Speed follower" operating mode can also be activated without the functions described before via the second **Positioning** tab:

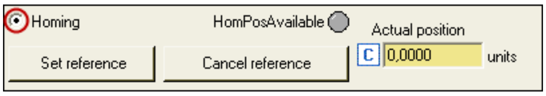


### 3.9.3 Set/reset home position

The measuring system in the machine is selected by means of homing and the 0 position is set within the possible physical travel range.

Detailed information on the "Homing" operating mode is provided in chapter "Basic drive functions" in subchapter "Homing". (📖 517)

The change to the "Homing" mode can be made by selecting the correspondent option on the **Positioning** tab:




**How to set the reference manually with a stopped motor:**

Click the **Set reference** button.

The current actual position in the machine measuring system now corresponds to the home position set in C01227/2 (Lenze setting: 0.0000 units).

## Stop!

If you set the reference with the PC manual control, the drive is referenced.

**If another home position is required for normal operation:**

Click the **Cancel reference** button and reset the status signals *HomPosAvailable* and *HomePosDone* in the MCK status word in order that no mechanical problems can occur during normal operation.

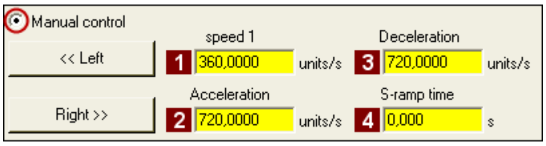
- Setpoints and actual positions remain untouched until a renewed reference setting or homing.

### 3.9.4 Manual jog

In the "Manual jog" mode, the drive can be traversed manually in a clockwise or counter-clockwise direction.

> You can find detailed information on the "Manual jog" mode in the chapter entitled "Basic drive functions" in the subchapter "Manual jog". (📖 533)

The change to the "Manual jog" mode can be made by selecting the correspondent option on the **Positioning** tab:




▶ The profile parameters saved in the parameter set for the basic "manual control" function are used:

| | Parameter | Lenze setting | | Info |
|---|---|---|---|---|
| | | Value | Unit | |
| 1 | **Speed 1** (C01231/1) | 360.0000 | units/s | Speed for manual jog |
| 2 | **Acceleration** (C01232/1) | 720.0000 | units/s² | For accelerating and decelerating, different values can be set so that smooth starting and quick stopping of the drive can be implemented. |
| 3 | **Deceleration** (C01232/2) | 720.0000 | units/s² | |
| 4 | **S-ramp time** (C01233/1) | 0.000 | s | In order to reduce jerking, the two ramps can be set in such a way that they are s-shaped. This is done by entering a relative S-ramp time. |

**How to traverse the drive manually in inching mode:**

1. If the controller is still inhibited, enable the controller via the **Enable controller** button.
   - Please observe that the controller will not be enabled if other sources of controller inhibit (e.g. terminal RFR) are active.

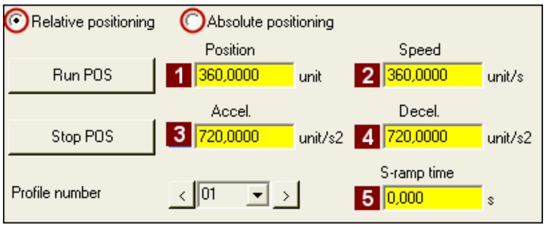
   If the controller is in the "OperationEnabled" status:
2. Click **<< Left** or **Right >>** (and keep it pressed) to traverse the drive in the corresponding direction of rotation.

### 3.9.5 Positioning (relative or absolute)

Positioning means that a workpiece/tool or material is moved from a starting position to a defined destination.

You can find detailed information on the "Positioning" mode in the chapter entitled "Basic drive functions" in the subchapter "Positioning". (📖 541)

The change to the "Positioning" operating mode and the simultaneous selection of the positioning mode (relative or absolute) is made by selecting the corresponding option on the **Positioning** tab:




▶ The profile parameters saved in the parameter set for the basic "Positioning" function are used:

| | Parameter | Lenze setting Value | Unit | Info |
|---|---|---|---|---|
| 1 | **Position** (C01301/1...15) | 360.0000 | units | Target position or distance to be traversed.<br>• When "Relative positioning" has been selected: Distance to starting position (current position).<br>• When "Absolute positioning" has been selected: Distance to defined zero position. |
| 2 | **Speed** (C01302/1...15) | 360.0000 | units/s | Maximum speed at which the target is to be approached. |
| 3 | **Acceleration** (C01303/1...15) | 720.0000 | units/$s^2$ | Stipulation of the change in speed at which maximum acceleration is to take place. |
| 4 | **Deceleration** (C01304/1...15) | 720.0000 | units/$s^2$ | Stipulation of the change in speed at which maximum deceleration to a standstill is to take place. |
| 5 | **S-ramp time** (C01306/1...15) | 0.000 | s | By defining an S-ramp time, a profile is travelled with S-shaped ramps. |

**Note!**

In order that no complex travel motions occur, no forwarding to following profile set in the profile data takes place with PC manual control!

You can find detailed information on entering profiles in the chapter entitled "Basic drive functions", subchapter "Profile entry". (📖 545)

**How to execute a positioning:**

1. Select the profile to be travelled (1 ... 15) in the **Profile number** list field.
2. If the controller is still inhibited, enable the controller via the **Enable controller** button.
   - Please observe that the controller will not be enabled if other sources of controller inhibit (e.g. terminal RFR) are active.

   If the controller is in the "OperationEnabled" status:
3. Click the **Run POS** button to start the positioning process.
   - Click the **Stop POS** button to stop an active positioning process anytime.

# 4 Device control (DCTRL)

This chapter provides information on internal device control as well as the device commands which can be executed via the subcodes of C00002.

- The device control causes the controller to take defined device statuses.
- The device control provides a multitude of status information in many ways:
  - Visually via the LED status displays on the front of the controller. (📖 583)
  - As text messages in the Logbook. (📖 590)
  - As process signals via the outputs of the LS_DriveInterface system block. (📖 115)
  - Via diagnostic / display parameters which are included in the »Engineer« parameter list as well as in the **Diagnostics** category in the keypad.

**Note!**

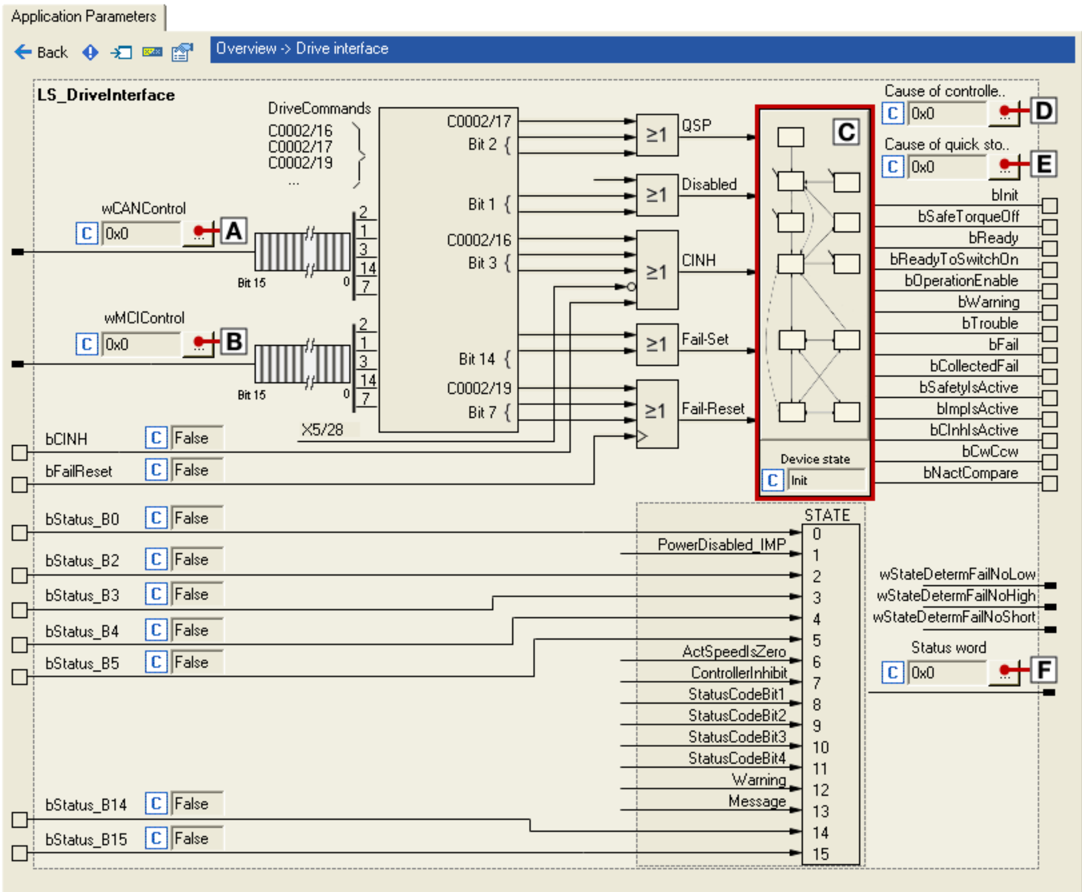
The device statuses of the controller are based on the operating statuses of the CiA402 standard. ▶ Device state machine and device statuses (📖 101)

### How to get to the parameterisation dialog of the device control:

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Go to the *Overview* dialog level and click the **Drive interface** button.

### Parameterisation dialog in the »Engineer«

The parameterisation dialog shows the input / output signals and the internal signal flow of the LS_DriveInterface system block which displays the device control in the function block editor:




| Range / Meaning | | Display parameter |
|---|---|---|
| A | Display of the control word via system bus (CAN) | C00136/2 |
| B | Display of the control word via communication module (e.g. PROFIBUS) | C00136/1 |
| C | Display of the internal state machine and the current device status | C00137 |
| D | Display of all active sources of a controller inhibit | C00158 |
| E | Display of all active sources of a quick stop | C00159 |
| F | Display of the status word of the device control | C00150 |

## 4.1 Device commands (C00002/x)

This chapter describes the device commands which are provided in the subcodes of C00002 and can be carried out using the keypad or, alternatively, the »Engineer« when an online connection has been established.

The device commands serve, among other things, to directly control the controller, to organise parameter sets, and to call diagnostic services.

Regarding the execution of the device commands, a distinction is drawn between:

- Device commands which have an immediate effect on control (e.g. "Activate quick stop")
  - After being called in C00002/x, these device commands provide static status information ("On" or "Off").
- Device commands with longer execution duration (several seconds)
  - After being called in C00002/x, these device commands provide dynamic status information ("Work in progress 20%" → "Work in progress 40%", etc.).
  - The execution of the device command has not finished successfully until the "Off / ready" status information is provided in C00002/x.
  - In the event of an error, the "Action cancelled" status information is provided in C00002/x . In this case, further details can be obtained from the status of the device command executed last which is displayed in C00003.

**Stop!**

Before the supply voltage is switched off after a device command has been transmitted via C00002/x, the device command must be checked for successful completion on the basis of the status information provided in C00002/x!

- This is of particular importance for device commands which save data to the memory module of the device. Incomplete storage processes may lead to data inconsistencies in the memory module.

**Note!**

- Before activating device commands by a master control, wait for the "Ready" signal of the controller.
- The device will reject a write process to C00002/x if the value is >1 and issue an error message.
- C00003 displays the status of the device command that was executed last.

### Activate device command

When an online connection has been established, simply use the »Engineer« to activate a device command by selecting the corresponding option from the **Parameters** tab in C00002/x ("0: off" or "1: On / start").

- Alternatively, the device command can also be activated via e.g. keypad or through a master control by writing to C00002/x.
- Some of the frequently used device commands (such as "Save parameter set") can also be executed via the *Toolbar* icons of the »Engineer« when an online connection has been established:

| Symbol | Function |
| --- | --- |
|  | Enable controller |
|  | Inhibit controller |
|  | Save parameter set (for 8400: Save all parameter sets) |

**Note!**

Device commands that can be executed via the *Toolbar* of the »Engineer« always affect the element currently selected in the *Project view* including all subelements!

- If no controller but e.g. a system module is selected in the *Project view* instead, the corresponding device command will be activated in all lower-level controllers that have an online connection to the »Engineer«.

Before the desired action is carried out, a confirmation prompt appears first, asking whether the action is really to be carried out.

### Short overview of device commands

Device commands described in this chapter:

| C00002 Subcode: | Device command | Controller inhibit required | Status information |
|---|---|---|---|
| 1 | Load Lenze setting | ● | dynamic |
| 6 | Load all parameter sets | ● | dynamic |
| 11 | Save all parameter sets | | dynamic |
| 16 | Enable/Inhibit controller | | static |
| 17 | Activate/Deactivate quick stop | | static |
| 19 | Reset error | | static |
| 21 | Delete logbook | | static |
| 27 | Device search function | | static |

Device commands described in other chapters:

| C00002 Subcode: | Device command | Controller inhibit required | Status information |
|---|---|---|---|
| 23 | Motor parameter identification<br>▸ Automatic motor data identification | ● | dynamic |
| 25 | Identify resolver error<br>▸ Optimising resolver behaviour | ● | dynamic |
| 26 | CAN reset node<br>▸ Reinitialising the CANopen system bus interface | | static |
| 28 | Check MasterPin<br>▸ Unlocking the controller with a MasterPin | ● | static |
| 29 | Set binding ID<br>▸ Device personalisation | | static |
| 30 | Delete binding ID<br>▸ Device personalisation | | static |
| 31 | Set password<br>▸ Password protection | | static |
| 32 | Check password<br>▸ Password protection | | static |
| 33 | Delete password<br>▸ Password protection | | static |
| 34 | Identify pole position (360°)<br>▸ Pole position identification (PPI) | ● | dynamic |

### 4.1.1 Load Lenze setting

The C00002/1 = "1: On / start" device command resets the parameters to the Lenze setting which are saved in the controller Firmware.

- Can only be executed if the controller is inhibited; otherwise, the feedback C00002/1 = "6: No access - controller inhibit" will be returned.
- All parameter changes which have been carried out after the last time the parameter set was saved will be lost!
- This device command has an effect on the settings of the parameters of the operating system, application and module.

**How to load the Lenze setting:**

1. If the controller is enabled, it must be inhibited, e.g. by executing the "Enable/Inhibit controller" device command (C00002/16 = "0: Off / ready").
2. Execute the "Load Lenze setting" device command:
   C00002/1 = "1: On / start"

   The loading process may take a couple of seconds. After the device command has been called, C00002/1 returns dynamic status information ("Work in progress 20%" ➞ "Work in progress 40 %" ➞ "Work in progress 60 %", etc.).

### 4.1.2 Load all parameter sets

The C00002/6 = "1: On / start" device command reloads all parameter settings from the memory module to the controller.

- Can only be executed if the controller is inhibited; otherwise, the feedback C00002/6 = "6: No access - controller inhibit" will be returned.
- All parameter changes which have been carried out after the last time the parameter set was saved will be lost!
- This device command has an effect on the settings of the parameters of the operating system, application and module.

**Note!**

The controller is currently provided with one data record for all parameters, i.e. every parameter has a value. Several data records per controller are in preparation.

For up to 32 freely selectable parameters, the basic function Parameter change-over provides a change-over between four sets with different parameter values. (📖 734)

**How to load the parameter settings from the memory module:**

1. If the controller is enabled, it must be inhibited, e.g. by executing the "Enable/Inhibit controller" device command (C00002/16 = "0: Off / ready").
2. Execute the "Load all parameter sets" device command:
   C00002/6 = "1: On / start"

   The loading process may take a couple of seconds. After the device command has been called, C00002/6 returns dynamic status information ("Work in progress 20 %" ➞ "Work in progress 40 %" ➞ "Work in progress 60 %", etc.).

### 4.1.3 Save all parameter sets

If parameter settings are changed in the controller, those changes will be lost after mains switching of the controller unless the settings have been saved explicitly.

The C00002/11 = "1: On / start" device command saves the current parameter settings safe against mains failure to the memory module of the controller.

**Note!**

When the device is switched on, all parameters are automatically loaded from the memory module to the main memory of the controller.

Observe the following to avoid data inconsistencies which cause errors when the parameters are loaded from the memory module:

During the storage process:

- Do not switch off the supply voltage!
- Do not remove the memory module from the device!

The controller is currently provided with one data record for all parameters, i.e. every parameter has a value. Several data records per controller are in preparation.

**How to save the parameter settings to the memory module:**

Execute the "Save all parameter sets" device command:
C00002/11 = "1: On / start"

The storage process may take a couple of seconds. After the device command has been called in C00002/11, dynamic status information ("Work in progress 20%" ➞ "Work in progress 40%" ➞ "Work in progress 60%", etc.) is returned.

**Tip!**

- This device command can also be activated via the *toolbar* icon.
- The "Load Lenze setting" device command (C00002/1 = "1: On / start") resets the parameter settings to the delivery status of the device.

### 4.1.4 Enable/Inhibit controller

The C00002/16 = "1: On / start" device command enables the controller, provided that no other source of a controller inhibit is active.

The C00002/16 = "0: Off / ready" device command inhibits the controller again, i.e. the power output stages in the controller are inhibited and the speed/current controllers of the motor control are reset.

- The motor becomes torqueless and coasts, if it has not yet been in standstill.
- When the controller is inhibited, the status output *bCInhActive* of the LS_DriveInterface system block is set to TRUE.
- When the controller inhibit request is reset, the drive synchronises to the actual speed. For this purpose,
  - If the flying restart circuit is activated in C00990, the flying restart function parameterised in C00991 is used for the synchronisation to the rotary or standing drive. ▸ Flying restart function (📖 247)
  - In the case of an operation with feedback, the actual speed is read out by the encoder system.
  - In the case of a sensorless vector control (SLVC), the actual speed from the motor model of the motor control is used for the synchronisation.
- C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit:

| Bit | Cause/Source of controller inhibit |
|---|---|
| Bit 0 | Terminal controller enable |
| Bit 1 | CAN control word |
| Bit 2 | MCI control word |
| Bit 3 | SwitchOn |
| Bit 4 | Application (LS_DriveInterface system block: *bCInh* input) |
| **Bit 5** | **Device command** (C00002/16) |
| Bit 6 | Error with a "Fault"/"Trouble" error response or system error, respectively |
| Bit 7 | Internal signal |
| Bit 8 | Reserved |
| Bit 9 | Reserved |
| Bit 10 | AutoStartLock |
| Bit 11 | Motor parameter identification |
| Bit 12 | Automatic brake operation |
| Bit 13 | DCB-IMP |
| Bit 14 | Reserved |
| Bit 15 | Reserved |

**Tip!**

The controller can also be enabled or inhibited via the and *toolbar* icons.

### 4.1.5 Activate/Deactivate quick stop

The C00002/17 = "1: On / start" device command activates the quick stop function, i.e. the motor control is separated from the setpoint selection, and within the deceleration time parameterised inC00105 the motor is brought to a standstill ($n_{act} = 0$).

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00105 | Decel. time - quick stop | 2.000 | s |

- The motor is kept at a standstill during closed-loop operation.
- A pulse inhibit (CINH) is set if the auto DCB function has been activated via C00019.
- In C00104, the position control can be activated during ramp-down and/or at standstill.

The C00002/17 = "0: Off / ready" device command deactivates the quick stop again, provided that no other source of a quick stop is active.

- C00159 provides a bit coded representation of all active sources/triggers of a quick stop:

| Bit | Cause/Source of controller inhibit |
|---|---|
| Bit 0 | Reserved |
| Bit 1 | CAN control word (bit 2) |
| Bit 2 | MCI control word (bit 2) |
| Bit 3 | Reserved |
| Bit 4 | Application (LS_MotorInterface system block: *bQspOn* input) |
| **Bit 5** | **Device command** (C00002/17) |
| Bit 6 | Device error with "TroubleQSP" error response |
| Bit 7 | Internal signal |
| Bit 8 | Reserved |
| Bit 9 | Reserved |
| Bit 10 | Operating system |
| Bit 11 | Reserved |
| Bit 12 | MCK (System block LS_MotionControlKernel: Input *bQspOn*) |
| Bit 13 | Reserved |
| Bit 14 | Reserved |
| Bit 15 | Reserved |

### 4.1.6 Reset error

The C00002/19 = "1: On / start" device command acknowledges an existing error message if the error cause has been eliminated and thus the error is not pending anymore.

- After the reset (acknowledgement) of the current error, further errors may be pending which must also be reset.
- The status-determining error is displayed in C00168.
- The current error is displayed in C00170.

**Tip!**

An error message can also be acknowledged by activating the **Reset error** button in the **Diagnostics** tab.

Detailed information on error messages can be found in the "Diagnostics & error management" chapter. (581)

### 4.1.7 Delete logbook

The C00002/21 = "1: On / start" device command deletes all logbook entries.

**Tip!**

Click the **Logbook** button in the **Diagnostics** tab to display the logbook in the »Engineer«.

In the *Logbook* dialog box, it is also possible to delete all logbook entries by clicking the **Delete** button.

Detailed information on the logbook can be found in the "Diagnostics & error management" chapter. (581)

### 4.1.8 Device search function

In some applications where controllers are installed in control cabinets or are positioned in a spacious plant, it is often difficult to locate a device connected online for e.g. maintenance work. There is an established online connection with the device but you do not know where the controller is located physically.

The C00002/27 = "1: On / start" device command serves to carry out an "optical location":

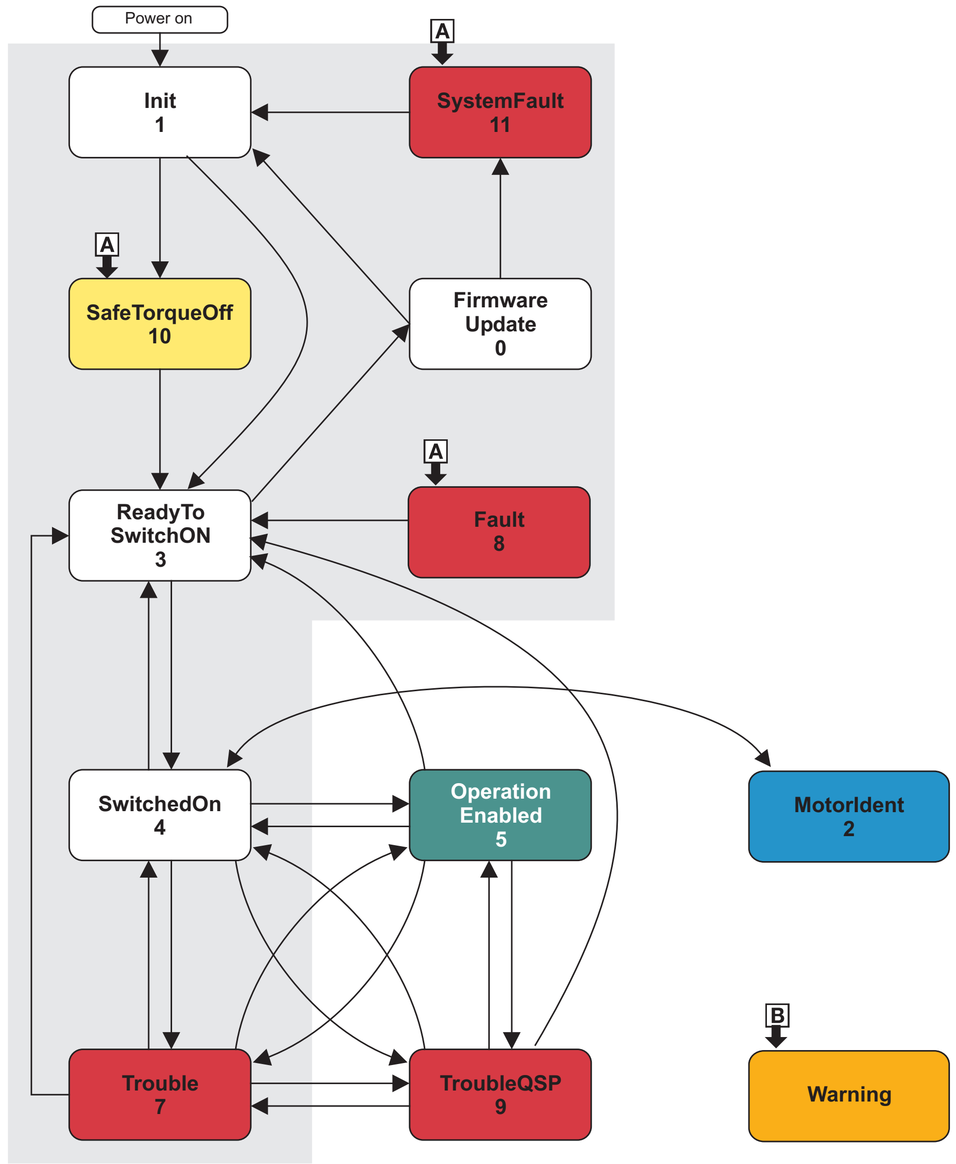
- For the time set in C00181/1, all four status LEDs at the front of the controller flash. Afterwards, the function is turned off automatically.
- If the device command is executed again within the set time period, the duration is extended accordingly.
- The setting C00002/27 = "0: Off / ready" serves to abort or switch off the function.
- Adjustable time period: 0 ... 6000 s (Lenze setting: 5 s)

**Tip!**

The device search function can also be activated via the *toolbar* icon.

## 4.2 Device state machine and device statuses

The behaviour of the controller is mainly determined by the current device status within the device state machine. Which device status is active and device status is next depends on certain control signals (e.g. for controller inhibit and quick stop) and status parameters.




Pulse inhibit (grey field)

A Can be reached from all statuses.

B "Warning" contradicts the definition of a device status. In fact, it is a message which is to call attention to the device status the warning exists for.
"Warning" can occur in parallel to other device statuses.

[4-1] Device state machine

- The arrows between the device statuses mark possible status changes.
- The digits stand for the status ID (see table below).

- The change from one status to the other is carried out within a 1-ms cycle. If within this time there are several requests for status changes, the status with the higher priority is processed first (see table below).
- C00137 displays the current device state.
- C00150 (status word) provides a bit coded representation of the current device state via bits 8 ... 11 (see table below).

| ID | Device status (Display in C00137) | Priority | Status bits (Display in C00150) | | | | Meaning |
|---|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 | |
| 0 | FirmwareUpdate | - | 0 | 0 | 0 | 0 | Firmware update function is active |
| 1 | Init | - | 0 | 0 | 0 | 1 | Initialisation is active |
| 2 | MotorIdent | - | 0 | 0 | 1 | 0 | Motor parameter identification is active |
| 3 | ReadyToSwitchOn | Prio 5 | 0 | 0 | 1 | 1 | Device is ready to start |
| 4 | SwitchedOn | Prio 4 | 0 | 1 | 0 | 0 | Device is switched on |
| 5 | OperationEnabled | Prio 1 | 0 | 1 | 0 | 1 | Operation |
| 6 | - | - | 0 | 1 | 1 | 0 | - |
| 7 | Trouble | Prio 3 | 0 | 1 | 1 | 1 | Trouble is active |
| 8 | Fault | Prio 7 | 1 | 0 | 0 | 0 | Fault is active |
| 9 | TroubleQSP | Prio 2 | 1 | 0 | 0 | 1 | TroubleQSP is active |
| 10 | SafeTorqueOff | Prio 6 | 1 | 0 | 1 | 0 | Safe torque off is active |
| 11 | SystemFault | Prio 8 | 1 | 0 | 1 | 1 | System fault is active |

[4-1] Device statuses, priorities, and meaning of the status bits in the status word

### 4.2.1 FirmwareUpdate

**Note!**

This function may only be executed by qualified Lenze personnel!

## 4.2.2 Init

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Off | Off | Init | 0 | 0 | 0 | 1 |

The controller is in this status immediately after switching on its 24 V supply voltage.

In the "Init" status, the operating system is initialised and all device components (communication module, memory module, power section, etc.) are identified. When identifying the power section, it is checked first if it is switched on or if the required voltage lies within the tolerance zone, respectively.

- The inverter is inhibited, i.e. the motor terminals (U, V, W) of the inverter are deenergised.
- The digital and analog inputs are not yet evaluated at this time.
- The bus systems (CAN, PROFIBUS etc.) do not work yet, i.e. communication is not possible.
- The application is not yet processed.
- The monitoring functions are not yet active.
- The controller cannot be parameterised yet and no device commands can be carried out yet.

**Note!**

If the 24V voltage supply is in the valid range (>19V) and the initialisation is finished, the device changes automatically to the "ReadyToSwitchOn" status.

If only the 24V voltage supply is available during the mains connection, the error message "LU: Undervoltage in the DC bus" is also entered into the logbook of the drive controller.

### 4.2.3 MotorIdent

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | Off | MotorIdent | 0 | 0 | 1 | 0 |

The controller has been provided with the "Motor parameter identification" function for automatic identification of the motor parameters. If the motor parameter identification is active, the controller is in the "MotorIdent" device status.

The "MotorIdent" device status can only be reached from the "SwitchedOn" device status, i.e. the controller must be inhibited first so that motor parameter identification can be started via the "Motor parameter identification" device command (C00002/23).

**Stop!**

While the motor parameters are being detected,

- the controller does not respond to setpoint changes or control processes (e.g. speed setpoints, quick stop, torque limitations),
- the application remains active,
- all system interfaces (IO, bus systems, etc.) remain active,
- error monitoring remains active,
- the inverter is controlled independently of the setpoint sources.

After the motor parameter identification is completed, the status changes back to "SwitchedOn".

**Tip!**

Detailed information on motor parameter identification can be found in the "Automatic motor data identification" subchapter on motor control. (126)

### 4.2.4 SafeTorqueOff

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | Off | SafeTorqueOff | 1 | 0 | 1 | 0 |

**Note!**

This device status is only possible in connection with an integrated safety system and if a power section supply is available!

#### Integrated safety systems with Inverter Drives 8400

Drive controller of the 8400 series can be equipped with the integrated "Safe torque off (STO)" safety system.

The integrated safety system is applicable on machines for the protection of persons.

The drive function is still carried out by the drive controller. The safety system provides safe inputs. If the safety system is activated, it executes control functions according to EN 60204-1 directly in the drive controller in case of errors.

#### Safety status

If the drive controller is switched off by the safety system, the device changes to the "SafeTorqueOff" status.

If the safety system deactivates the "Safe torque off (STO)" request, the device changes to the "ReadyToSwitchOn" status.

Detailed information on the integrated safety system can be found in the hardware manual!

**The hardware manual contains important notes on the safety system which must be observed!**

- The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 drive controller.

### 4.2.5 ReadyToSwitchOn

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| | Off | ReadyToSwitchOn | 0 | 0 | 1 | 1 |

The controller is in this device status directly after the initialisation has been completed!

- The bus systems are running and the terminals and encoders are evaluated.
- The monitoring modes are active.
- The controller can be parameterised.
- The application is basically executable.

**Note!**

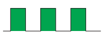
- The "ReadyToSwitchOn" status is not only activated after the mains connection, but also after the deactivation of "Trouble", "Fault" or "SafeTorqueOff".
- If C00142 activates the autostart option "Inhibit at power-on" (Lenze setting), explicit deactivation of the controller inhibit after mains connection is always required for the controller to change from the "ReadyToSwitchOn" status to the "SwitchedOn" status.
- If only the 24V voltage supply is available during the mains connection, the error message "LU: Undervoltage in the DC bus" is entered into the logbook of the drive controller and the drive controller remains in the "ReadyToSwitchOn" status.

**Danger!**

If the "Inhibit at power-on" auto-start option has been deactivated in C00142, the "ReadyToSwitchOn" status switches directly to the SwitchedOn" status after mains connection.

▸ Automatic restart after mains connection/fault... (📖 112)

### 4.2.6 SwitchedOn

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|  | Off | SwitchedON | 0 | 1 | 0 | 0 |

The drive is in this device status if the DC bus voltage is applied and the controller is still inhibited by the user (controller inhibit).

- The bus systems are running and the terminals and encoders are evaluated.
- The monitoring modes are active.
- The application is basically executable.

If the controller inhibit is deactivated, the devices changes to the "OperationEnabled" status and the motor follows the setpoint defined by the active application.

**Tip!**

C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit.

Depending on certain conditions, a status change takes place based on the "SwitchedOn" device status:

| Change condition | Changeover to the device status |
|---|---|
| Control bit "EnableOperation" of all control channels = "1"<br>**AND** terminal RFR = HIGH level (controller enable) | OperationEnabled |
| "SwitchOn" control bit of a control channel = "0". | ReadyToSwitchOn |
| Motor parameter identification requested. | MotorIdent |
| Undervoltage in the DC bus. | Trouble/Fault (depending on C00600/1) |
| Error with error response "Trouble" occurs. | Trouble |
| Error with error response "TroubleQSP" occurs. | TroubleQSP |

**Related topics:**

▸ wCANControl/wMCIControl control words (📖 118)

### 4.2.7 OperationEnabled

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| (green) | Off | OperationEnabled | 0 | 1 | 0 | 1 |

The drive controller is in this device status if the controller inhibit is deactivated and no error ("Trouble" or "TroubleQSP") has occurred.

If the operation is enabled and the magnetisation in case of servo control (SC) and sensorless vector control (SLVC) has been completed, the motor follows the setpoint determined by the active application.

Depending on certain conditions, a status change takes place based on the "OperationEnabled" device status.

| Change condition | Changeover to the device status |
|---|---|
| Control bit "EnableOperation" of a control channel = "0"<br>**OR** terminal RFR = LOW level (controller inhibit). | SwitchedOn |
| "SwitchOn" control bit of a control channel = "0". | ReadyToSwitchOn |
| Undervoltage in the DC bus. | Trouble/Fault (depending on C00600/1) |
| Error with error response "Trouble" occurs. | Trouble |
| Error with error response "TroubleQSP" occurs. | TroubleQSP |

**Related topics:**

▸ wCANControl/wMCIControl control words (🕮 118)

### 4.2.8 TroubleQSP

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
|  |  | TroubleQSP | 1 | 0 | 0 | 1 |

This device status will be active as soon as a monitoring mode responds, the error response "TroubleQSP" has been parameterised for.

- The drive is decelerated to standstill with torque within the deceleration time parameterised for quick stop independently of the defined setpoint and can be kept there.
- The device status can only be abandoned by acknowledging the error if the error cause is removed.
- When the controller is inhibited, it is possible to jump to the "SwitchedOn" status even during the error status since the controller inhibit function has a higher priority. As long as the error is pending and has not been acknowledged, the status is changed back to the "TroubleQSP" status when the controller is enabled afterwards.

Depending on certain conditions a status change takes place based on the "TroubleQSP" device status.

| Change condition | Changeover to the device status |
|---|---|
| "SwitchOn" control bit of a control channel = "0". | ReadyToSwitchOn |
| Control bit "EnableOperation" of all control channels = "1"<br>**AND** terminal RFR = HIGH level (controller enable)<br>**AND** error is reset by the control bit "ResetFault"<br>**AND** no more errors are pending. | OperationEnabled |
| Control bit "EnableOperation" of a control channel = "0"<br>**OR** terminal RFR = LOW level (controller inhibit)<br>**AND** error is reset by the control bit "ResetFault"<br>**AND** no more errors are pending. | SwitchedOn |
| A message is active in the system. | Trouble |

**Related topics:**

- wCANControl/wMCIControl control words (📖 118)
- Basics on error handling in the controller (📖 581)
- Error messages of the operating system (📖 602)

### 4.2.9 Trouble

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Off | | Trouble | 0 | 1 | 1 | 1 |

This device status becomes active as soon as a monitoring mode responds for which the error response "Trouble" has been parameterised.

- The motor has no torque (is coasting) due to the inhibit of the inverter.
- The "Trouble" device status is automatically abandoned if the error cause has been removed.

**Note!**

If the "Inhibit at trouble" auto-start option has been activated in C00142, explicit deactivation of the controller inhibit is required before this status can be abandoned.

Depending on certain conditions a status change takes place based on the "Trouble" device status.

| Change condition | Changeover to the device status |
|---|---|
| The error cause is no longer active. | ReadyToSwitchOn |
| Control bit "EnableOperation" of all control channels = "1"<br>**AND** terminal RFR = HIGH level (controller enable)<br>**AND** the message has been cancelled. | OperationEnabled |
| Control bit "EnableOperation" of a control channel = "0"<br>**OR** terminal RFR = LOW level (controller inhibit)<br>**AND** the message has been cancelled. | SwitchedOn |
| In the system, there is an error configured on "TroubleQSP".<br>**AND** the message has been cancelled. | TroubleQSP |

**Related topics:**

- wCANControl/wMCIControl control words (📖 118)
- Basics on error handling in the controller (📖 581)
- Error messages of the operating system (📖 602)

### 4.2.10 Fault

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Off | | Fault | 1 | 0 | 0 | 0 |

This device status will be active as soon as a monitoring mode responds, the error response "Fault" has been parameterised for.

- The motor has no torque (is coasting) due to the inhibit of the inverter.
- The error must explicitly be reset ("acknowledged") in order to exit the device status, e.g. by the device command "Reset error" or via the control bit "ResetFault" in the control word *wCanControl* or *wMCIControl*.

**Note!**

If an undervoltage in the DC bus of the drive controller occurs (error message "LU"), the device changes to the "Trouble" status.

An additional error of higher priority leads the device into the "Fault" status.

According to the Device state machine, the device changes to the "ReadyToSwitchOn" status after acknowledging the error although the undervoltage is still available!

If the "Inhibit at fault" auto-start option has been activated in C00142, explicit deactivation of the controller inhibit is required before the status can be abandoned.

**Related topics:**

- wCANControl/wMCIControl control words (118)
- Basics on error handling in the controller (581)
- Error messages of the operating system (602)

### 4.2.11 SystemFault

| LED DRIVE READY | LED DRIVE ERROR | Display in C00137 | Display in status word 1 (C00150) | | | |
|---|---|---|---|---|---|---|
| | | | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| Off | | SystemFault | 1 | 0 | 1 | 1 |

This device status becomes active if a system fault occurs.

- The device status can only be abandoned by
  - mains switching or
  - a system restart *(in preparation)*.

## 4.3 Automatic restart after mains connection/fault...

### .../Error/undervoltage/loading of the Lenze setting

In C00142, the starting performance of the controller after mains connection, undervoltage, loading of the Lenze setting as well as a "Trouble" or a "Fault" reset can be parameterised individually:

| Auto-start option (C00142) | | Lenze setting |
|---|---|---|
| Bit 0 | "Inhibit at power-on" auto-start option | 1 ≡ Inhibit is active |
| Bit 1 | Inhibit at trouble | 0 ≡ Inhibit is not active |
| Bit 2 | Inhibit at fault | 0 ≡ Inhibit is not active |
| Bit 3 | Inhibit at undervoltage | 1 ≡ Inhibit is active |
| Bit 4 | Auto-start option "Inhibit at Lenze setting" | 1 ≡ Inhibit is active |
| Bit 5 | Reserved | 0 |
| Bit 6 | | |
| Bit 7 | | |

**Note!**

In the Lenze setting , automatic restart after mains connection, undervoltage, and loading of the Lenze setting is inhibited.

### 4.3.1 "Inhibit at power-on" auto-start option

The auto-start option "Inhibit at power-on" prevents the change to the "SwitchedOn" status after mains connection if the controller is already enabled at mains connection.

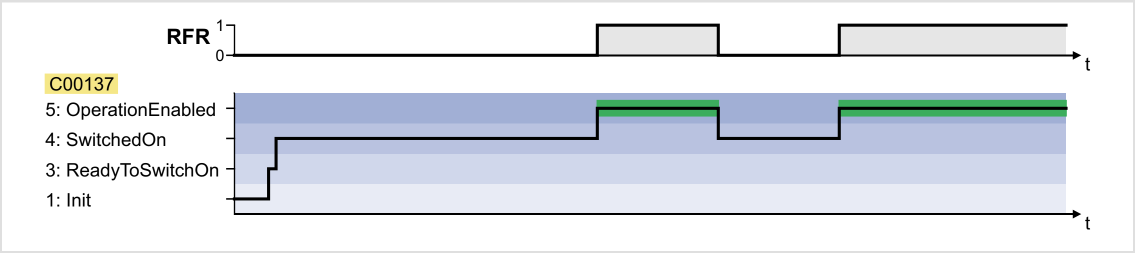
**Danger!**

If the "Inhibit at power-on" auto-start option has been deactivated in C00142, (bit 0 = 0), the motor can directly start to run if the controller is enabled after mains connection!

The following three cases describe the behaviour of the controller after mains connection depending on whether the controller is enabled and the set auto-start option. Here, it is assumed that after mains connection, no errors and trouble occur in the controller and the "EnableOperation" control bit in the *wDriveControl* is set to "1".

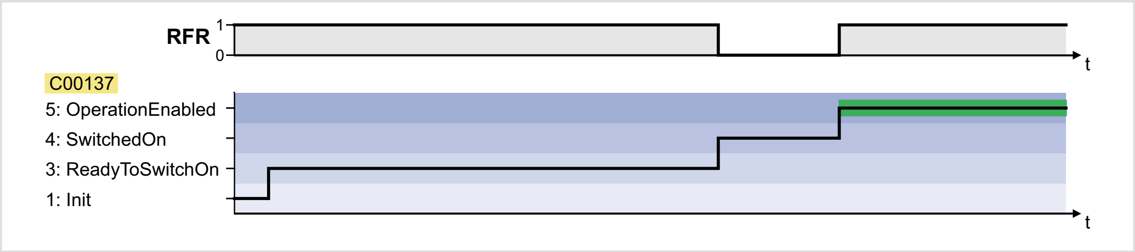
#### Case 1: No controller enable at mains connection

If the controller is not enabled at mains connection, the controller remains in the "SwitchedOn" status. Only with the controller enable, the status changes to the "OperationEnabled" status, independent of the set auto-start option:



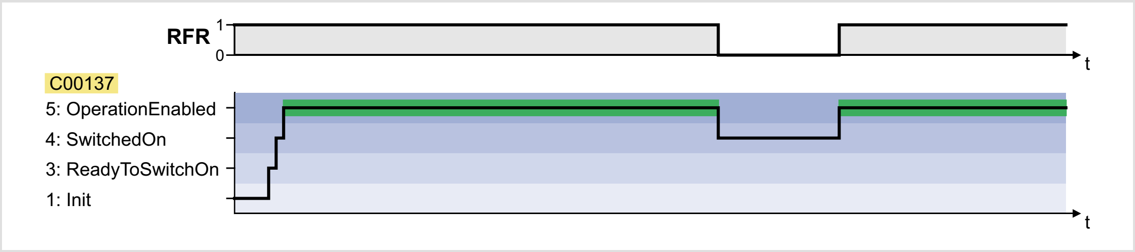

#### Case 2: Controller enable at mains connection and "Inhibit at power-on" activated

If the controller is enabled at mains connection and the auto-start option "Inhibit at power-on" is activated, the controller remains in the "ReadyToSwitchOn" status. For changing to the "SwitchedOn" status, the controller enable must first be deactivated. Only when the controller is enabled again afterwards, the status changes to "OperationEnabled":




#### Case 3: Controller enable at mains connection and "Inhibit at power-on" deactivated

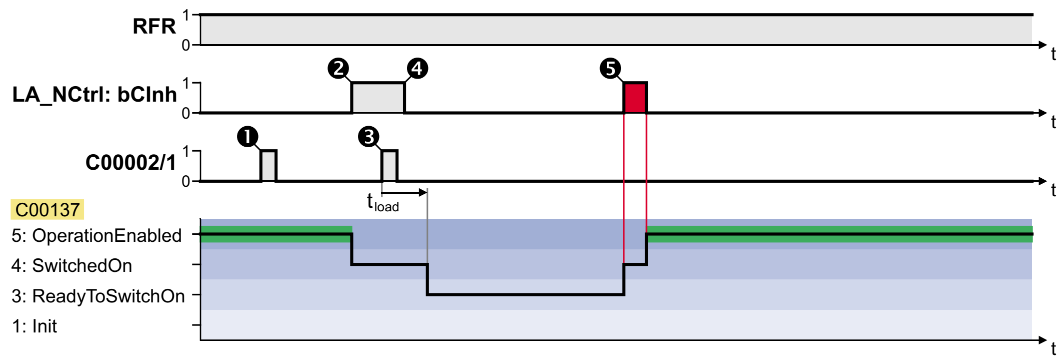
If the "Inhibit at power-on" auto-start option is deactivated in C00142 (bit 0 = 0), the status first changes from "ReadyToSwitchOn" to "SwitchedOn" and then to "OperationEnabled" after mains connection with enabled controller:



## 4.3.2 Auto-start option "Inhibit at Lenze setting"

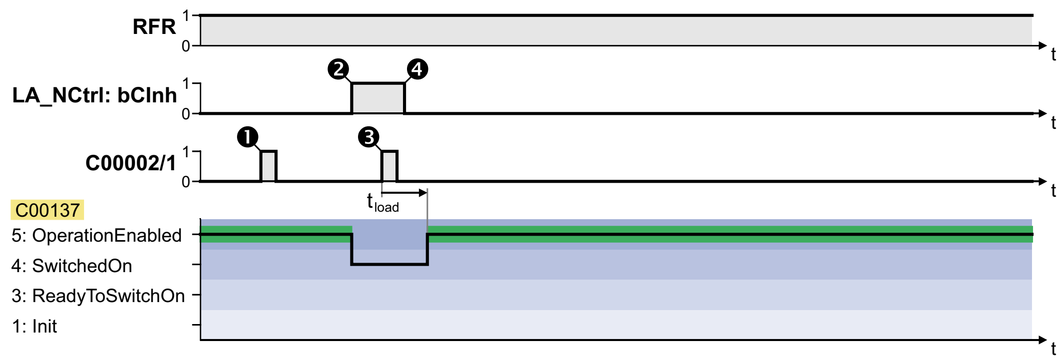
The "Inhibit at Lenze setting" auto-start option configurable via bit 4 of C00142 prevents the change to the "SwitchedOn" status after the Lenze setting has been loaded and the controller is enabled.

For a change to the "SwitchedOn" status, the controller enable must first be deactivated after the Lenze setting has been loaded. Only if the controller is enabled again afterwards, the status changes to "OperationEnabled":




❶ Loading of the Lenze setting not possible since controller inhibit not set.
❷ Controller inhibit via signal change by the user
❸ Loading of the Lenze setting via device command C00002/1 = "1"
❹ Controller enable via signal change by the user
❺ Controller inhibit and subsequent controller enable via signal change by the user
$t_{load}$ = time required for loading the Lenze setting

[4-2] Example 1: Behaviour with activated auto-start option "Inhibit at Lenze setting" (C00142: Bit 4 = "1")

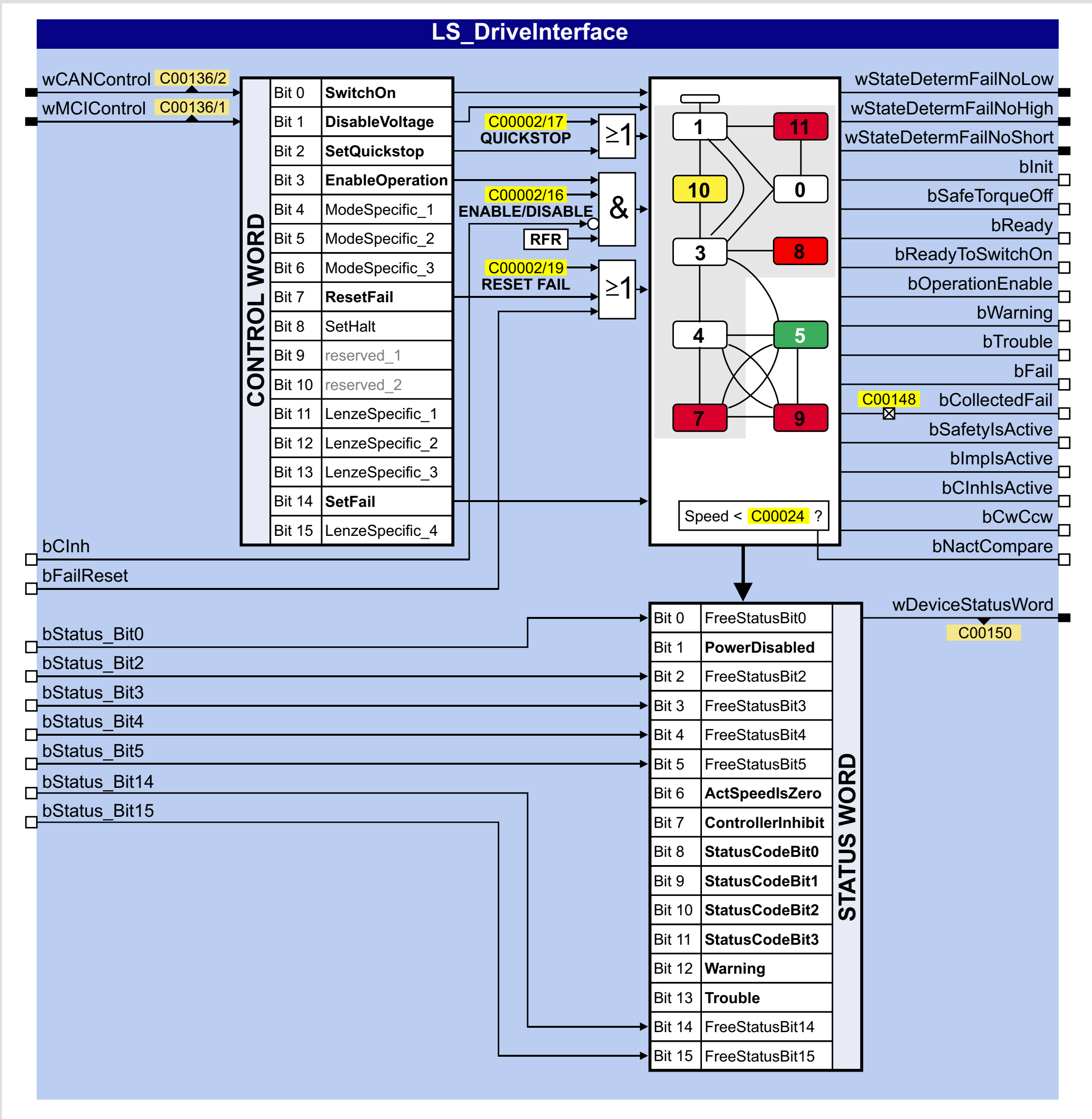



❶ Loading of the Lenze setting not possible since controller inhibit not set.
❷ Controller inhibit via signal change by the user
❸ Loading of the Lenze setting via device command C00002/1 = "1"
❹ Controller enable via signal change by the user
$t_{load}$ = time required for loading the Lenze setting

[4-3] Example 2: Behaviour with deactivated auto-start option "Inhibit at Lenze setting" (C00142: Bit 4 = "0")

## 4.4 Internal interfaces | "LS_DriveInterface" system block

The **LS_DriveInterface** system block displays the device control in the FB Editor.



### Inputs

| Identifier<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| wCANControl<br>C00136/2 \| WORD | Control word via system bus (CAN)<br>• The controller controlled by a master control (e.g. IPC) receives its control word by the CANopen system bus interface. The process data word is provided at this input by the upstream port block LP_CanIn1.<br>• For a detailed description of the individual control bits, see chapter "wCANControl/wMCIControl control words". (118) | |
| wMCIControl<br>C00136/1 \| WORD | Control word via communication module (e.g. PROFIBUS)<br>• The controller controlled by a master control (e.g. IPC) receives its control word by a plugged-in communication module. The process data word is provided at this input by the upstream port block LP_MciIn1.<br>• For a detailed description of the individual control bits, see chapter "wCANControl/wMCIControl control words". (118) | |
| bCInh<br>C00833/36 \| BOOL | ▸ Enable/Inhibit controller (97) | |
| | FALSE | Enable controller: The controller switches to the "OperationEnabled" device status if no other source of a controller inhibit is active.<br>• C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit. |
| | TRUE | Inhibit controller (controller inhibit): The controller switches to the "SwitchedOn" device status. |
| bFailReset<br>C00833/37 \| BOOL | ▸ Reset error messages (606) | |
| | FALSE↗TRUE | The current error is reset. |
| bStatus_Bit0<br>bStatus_Bit2<br>bStatus_Bit3<br>bStatus_Bit4<br>bStatus_Bit5<br>bStatus_Bit14<br>bStatus_Bit15<br>C00833/38 ... 44 \| BOOL | Freely assignable bits in the status word of the controller.<br>• You can use these bits for returning information to the master control (e.g. IPC). | |

### Outputs

| Identifier<br>DIS code \| data type | Value/meaning |
|---|---|
| wDeviceStatusWord<br>C00150 \| WORD | Status word of the controller (based on DSP-402)<br>• The status word contains all information relevant for controlling the controller.<br>• The status word is sent as a process data word to the master control via a port block:<br>– Port block LP_CanOut1 when the CANopen system bus interface is used or<br>– Port block LP_MciOut when a plugged-in communication module is used (e.g. PROFIBUS).<br>• For a detailed description of each status bit see chapter "wDeviceStatusWord status word". (119) |
| wStateDetermFailNoLow<br>WORD | Display of the status determining error (32-bit error number, Low-Word)<br>• If the "Use I16BitFailNo" (bit 15 = "1") option is activated in C00148, the short 16-bit error number (*wStateDetermFailNoShort*) is provided via this output as well.<br>– In this case, the *wStateDetermFailNoHigh* output is "0".<br>– Advantage: The bus transfer of the error numbers is possible via a data word without changing the interconnection of the technology application. |
| wStateDetermFailNoHigh<br>WORD | Display of the status determining error (32-bit error number, High-Word) |
| wStateDetermFailNoShort<br>WORD | Display of the status determining error (16-bit error number) |

| Identifier<br>DIS code \| data type | | Value/meaning |
|---|---|---|
| bInit<br>BOOL | TRUE | "Init" device status is active |
| bSafeTorqueOff<br>BOOL | TRUE | "SafeTorqueOff" device status is active |
| bReady<br>BOOL | TRUE | "SwitchedOn" device status is active |
| bReadyToSwitchOn<br>BOOL | TRUE | "ReadyToSwitchOn" device status is active |
| bOperationEnable<br>BOOL | TRUE | "OperationEnabled" device status is active |
| bWarning<br>BOOL | TRUE | A warning is indicated |
| bTrouble<br>BOOL | TRUE | "Trouble" device state is active |
| bFail<br>BOOL | TRUE | "Fault" device state is active |
| bCollectedFail<br>BOOL | TRUE | Group error: A device status according to the group error configuration in C00148 has occurred, the drive is not able to follow the setpoint selection. |
| bSafetyIsActive<br>BOOL | TRUE | In preparation |
| bImpIsActive<br>BOOL | TRUE | Pulse inhibit is active |
| bCInhIsActive<br>BOOL | TRUE | Controller inhibit is active |
| bCwCcw<br>BOOL | FALSE | Motor rotates in CW direction |
| | TRUE | Motor rotates in CCW direction |
| bNactCompare<br>BOOL | TRUE | During open-loop operation:<br>Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>Actual speed value < Comparison value (C00024) |

### Option "Lock bFail at TroubleQSP"

The TroubleQSP device status gets active as soon as a monitoring function responds that has been parameterised for the "TroubleQSP" error response. Since the *bFail* status output is not set in this case, it is not recognisable after pulse inhibit due to e.g. automatic brake operation (as well for a higher-level control), why the drive is standing und does not start when the setpoint is selected. Only after an error reset, a setpoint is accepted again.

From version 02.00.00: If the "Lock bFail at TroubleQSP" option is activated (bit 14 = "1") in C00148, the *bFail* status output is also set to TRUE if the device status is TroubleQSP.

### 4.4.1 wCANControl/wMCIControl control words

The controller is controlled by a master control (e.g. IPC) via the *wCanControl* or *wMCIControl* control word, respectively.

- *wCANControl*: Control word via system bus (CAN)
  - The process data word is provided at the *wCanControl* input via the upstream LP_CanIn1 port block.
  - Display parameter: C00136/2
- *wMCIControl*: Control word via a plugged-in communication module (e.g. PROFIBUS)
  - The process data word is provided at the *wMCIControl* input via the upstream LP_MciIn1 port block.
  - Display parameter: C00136/1
- The bit assignment for the *wCanControl*/*wMCIControl* control words can be seen from the table below.

**Note!**

The assignment of bits 11 ... 13 and bit 15 depends on the technology application selected in C00005!

- See description of the corresponding technology application.

| Bit | Name | Function |
|---|---|---|
| Bit 0 | SwitchOn | 1 ≡ Change to the "SwitchedOn" device status<br>• This bit must be set in the CAN **AND** MCI control word in order that the drive changes to the "SwitchedOn" state after main connection. In order to reach the "ReadyToSwitchOn" state, it is sufficient to set the bit to 0 in one of the two control words. |
| Bit 1 | DisableVoltage | 1 ≡ Inhibit inverter control (IMP - pulse inhibit) |
| Bit 2 | SetQuickStop | Activate quick stop (QSP)<br>Activate/Deactivate quick stop (98) |
| Bit 3 | EnableOperation | 1 ≡ Enable controller (RFR)<br>• This bit must be set in CAN **AND** in the MCI control word, otherwise the controller will be inhibited. |
| Bit 4 | ModeSpecific_1 | Reserved (currently not assigned) |
| Bit 5 | ModeSpecific_2 | |
| Bit 6 | ModeSpecific_3 | |
| Bit 7 | ResetFault | 1 ≡ Reset fault (trip reset)<br>• Acknowledge fault message (if the error cause has been eliminated). |
| Bit 8 | SetHalt | 1 ≡ Activate stop function<br>• Stop drive via stopping ramp (in preparation). |
| Bit 9 | reserved_1 | Reserved (currently not assigned) |
| Bit 10 | reserved_2 | |
| Bit 11 | LenzeSpecific_1 | Assignment depends on the selected technology application<br>• See description of the corresponding technology application. |
| Bit 12 | LenzeSpecific_2 | |
| Bit 13 | LenzeSpecific_3 | |
| Bit 14 | SetFail | 1 ≡ Set error (trip set) |
| Bit 15 | LenzeSpecific_4 | Assignment depends on the selected technology application<br>• See description of the corresponding technology application. |

**Tip!**

If a bus control is not wanted (e.g. in case of control via terminals):

Connect both control word inputs with the *wDriveCtrl* output signal of the LS_ParFix system block. This output signal has the fixed value "9", which corresponds to the following assignment:

- Bit 0, SwitchOn = 1
- Bit 3, EnableOperation = 1
- All others: 0

### 4.4.2 wDeviceStatusWord status word

The *wDeviceStatusWord* status word provided by the control system contains all information relevant for controlling the controller.

▶ The status word is sent as a process data word to the master control via a port block:
  – The **LP_CanOut1** port block if "CAN on board" is used or
  – the **LP_MciOut1** port block if a plugged-in communication module is used (e.g. PROFIBUS).

▶ Display parameter: C00150

▶ The bit assignment of the *wDeviceStatusWord* status word can be seen from the table below.

| Bit | Name | Status |
|---|---|---|
| Bit 0 | FreeStatusBit0 | Free status bit 0 |
| Bit 1 | PowerDisabled | 1 ≡ Inverter control inhibited (pulse inhibit is active) |
| Bit 2 | FreeStatusBit2 | Free status bit 2 (not assigned, freely assignable) |
| Bit 3 | FreeStatusBit3 | Free status bit 3 (not assigned, freely assignable) |
| Bit 4 | FreeStatusBit4 | Free status bit 4 (not assigned, freely assignable) |
| Bit 5 | FreeStatusBit5 | Free status bit 5 (not assigned, freely assignable) |
| Bit 6 | ActSpeedIsZero | During open-loop operation:<br>1 ≡ Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>1 ≡ Actual speed value < Comparison value (C00024) |
| Bit 7 | ControllerInhibit | 1 ≡ Controller inhibited (controller inhibit is active) |
| Bit 8 | StatusCodeBit0 | Bit coded display of the active device status<br>▸ Device state machine and device statuses (see table [4-1]) |
| Bit 9 | StatusCodeBit1 | |
| Bit 10 | StatusCodeBit2 | |
| Bit 11 | StatusCodeBit3 | |
| Bit 12 | Warning | 1 ≡ A warning is indicated |
| Bit 13 | Trouble | 1 ≡ Controller is in the "Trouble" device status<br>• E.g. if an overvoltage has occurred. |
| Bit 14 | FreeStatusBit14 | Free status bit 14 (not assigned, freely assignable) |
| Bit 15 | FreeStatusBit15 | Free status bit 15 (not assigned, freely assignable) |

# 5 Motor control (MCTRL)

This chapter provides information on the parameter setting of the controller's internal motor control. The 8400 TopLine controller both supports synchronous and asynchronous motors.

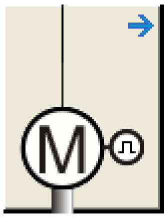
**Topics:**

## 5.1 Motor selection/Motor data

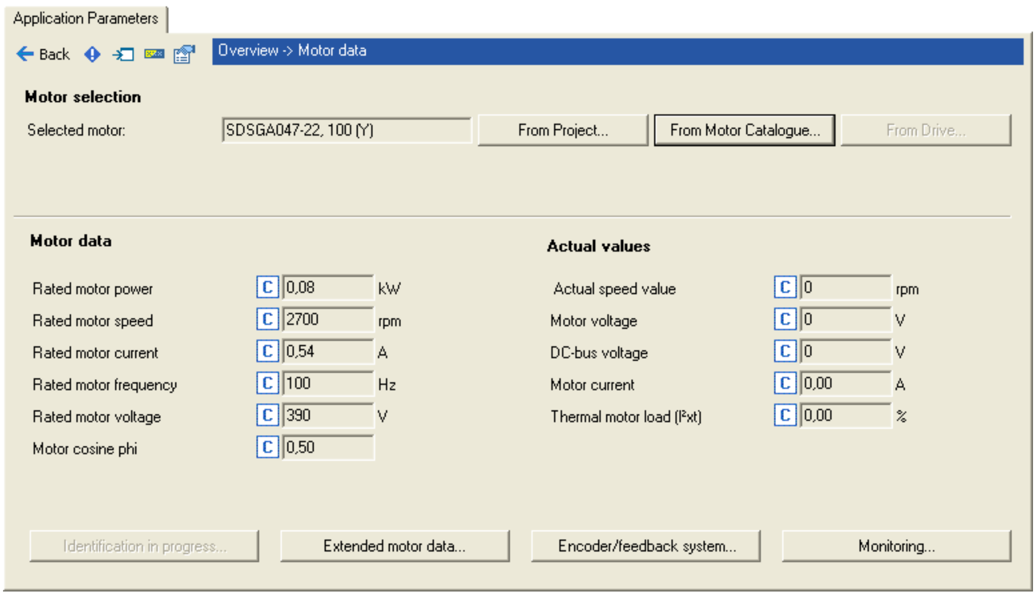
The motor data term comprises all parameters that only depend on the motor and that only characterise the electrical behaviour of the machine. The motor data are independent of the application in which the controller and the motor are used.

**Proceed as follows to open the dialog for parameterising the motor data:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Go to the *Overview* dialog level and click the following button:




**Parameterisation dialog in the »Engineer«**




- Via the **From Motor Catalogue** button, the motor catalogue can be opened to select another motor. ▸ Selecting a motor from the motor catalogue in the »Engineer« (📖 124)
- Via the **From drive...** button, the motor data set in the controller can be copied to the »Engineer« when an online connection has been established.
- When an online connection has been established to the controller, the **Identification in progress...** button serves to automatically identify different motor data. ▸ Automatic motor data identification (📖 126)
- The **Encoder/feedback system...** serves to get to the settings for the encoder/feedback system, if available. ▸ Encoder/feedback system (📖 293)

**Note!**

Sensorless vector control (SLVC) and sensorless control for synchronous motors (SLPSM) in particular requires the motor data parameters to be set. The motor data comprise the data of the motor nameplate and the data of the motor equivalent circuit.

If the motor has been selected via the motor catalogue of the »Engineer« or the motor data have been adapted offline using the »Engineer«, all motor data must then be copied to the controller and saved power-failure-proof to the memory module (device command: C00002/11) when an online connection has been established.

If the motor has a rated motor frequency with decimal position (e.g. motor nameplate data "23.7 Hz"), the following motor nameplate data must be increased by the factor 10:

- C00089: Rated motor frequency
- C00081: Rated motor power
- C00087: Rated motor speed
- C00090: Rated motor voltage

With a rated motor frequency of "23.7 Hz", for instance 237 Hz must be set in C00089.

### Motor data

In the parameterisation dialog, the data of the motor nameplate for the selected motor are displayed under "Motor data".

| Parameter | Info |
|---|---|
| C00081 | Rated motor power |
| C00087 | Rated motor speed |
| C00088 | Rated motor current |
| C00089 | Rated motor frequency |
| C00090 | Rated motor voltage |
| C00091 | Motor cos φ |

### Actual values

When an online connection to the controller has been established, the following actual values are displayed in the parameterisation dialog under "Actual values":

| Parameter | Info |
|---|---|
| C00051 | Actual speed value |
| C00052 | Motor voltage |
| C00053 | DC-bus voltage |
| C00054 | Motor current |
| C00066 | Thermal motor load (I2xt) |
| Highlighted in grey = display parameter | |

### Adapting motor data manually

If a third party manufacturer's motor is used, the displayed motor data can exactly be adapted to the real motor by clicking the **From project...** button and selecting the "Own motor settings" entry from the **Motor selection** dialog box afterwards. For this purpose, the data of the motor nameplate and the equivalent circuit diagram must be available.

**Tip!**

For a better concentricity factor, we recommend to perform motor parameter identification of the third party manufacturer's motor first. The motor parameters can be manually adapted afterwards.

Improving the concentricity factor includes

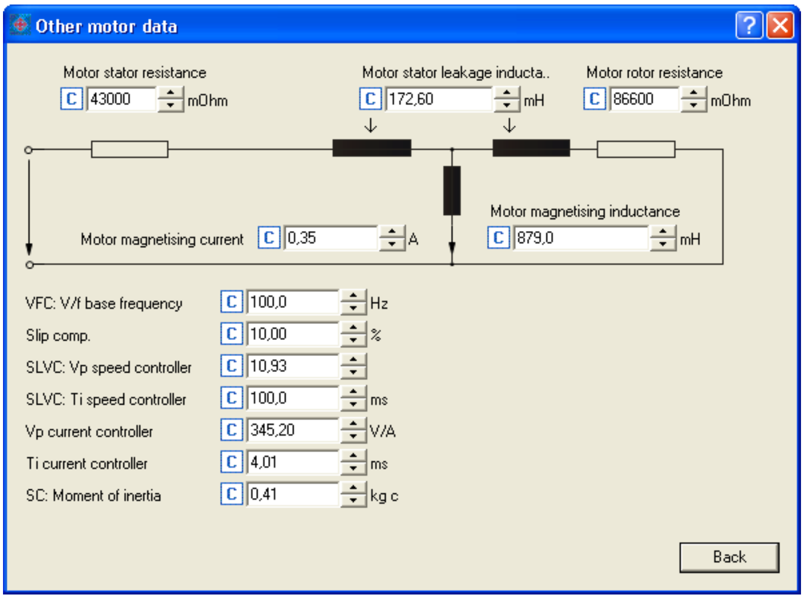
- the adjustment of the inverter error characteristic to the drive system and
- the knowledge of the motor cable resistance.

Both factors are determined in the course of motor parameter identification.

▸ Automatic motor data identification (📖 126)

### Other motor data

Click the **Other motor data...** button and go to the *Other motor data* dialog box including the motor equivalent circuit (in the following for an asynchronous motor):

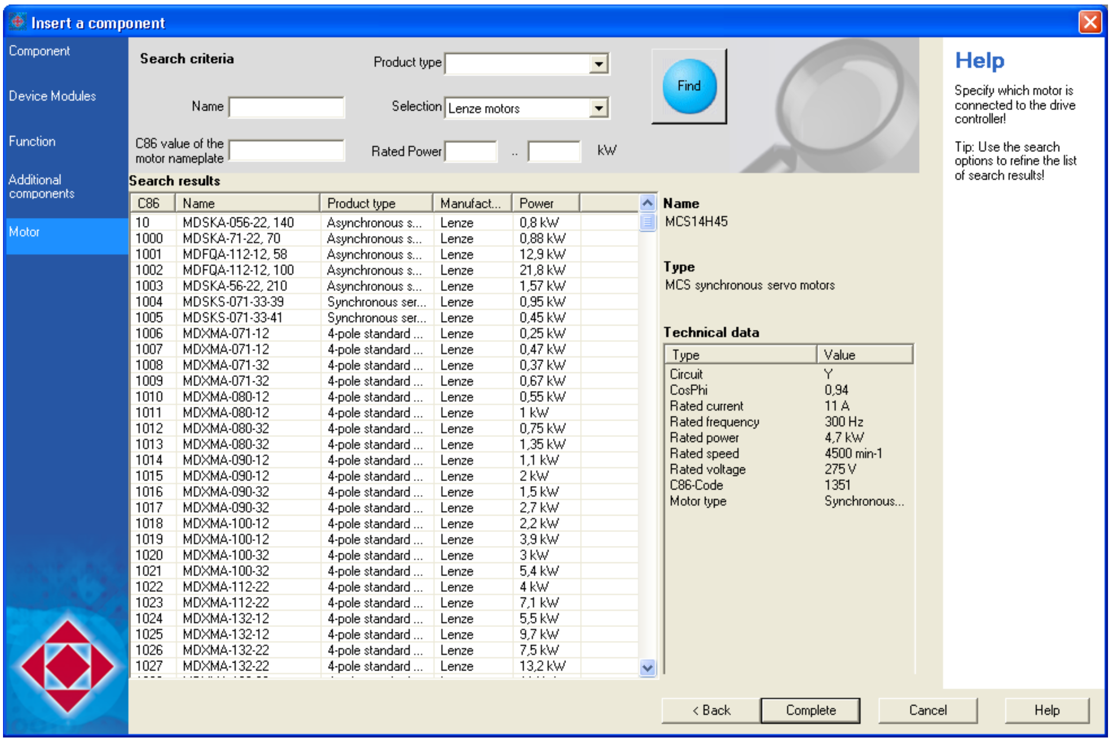



| Parameter | Info | ASM | PSM |
|---|---|---|---|
| C00084 | Motor stator resistance | ● | ● |
| C00085 | Motor stator leakage inductance | ● | ● |
| C00082 | Motor rotor resistance | ● | |
| C00095 | Motor magnetising current | ● | |
| C00092 | Motor magnetising inductance | ● | |
| C00015 | VFCplus: V/f base frequency | ● | |
| C00021 | Slip compensation | ● | |

| Parameter | Info | ASM | PSM |
|---|---|---|---|
| C00070 | SLVC: Vp speed controller | ● | |
| C00071 | SLVC: Ti speed controller | ● | |
| C00075 | Vp current controller | ● | ● |
| C00076 | Ti current controller | ● | ● |
| C00273 | Moment of inertia | ● | ● |

### 5.1.1 Selecting a motor from the motor catalogue in the »Engineer«

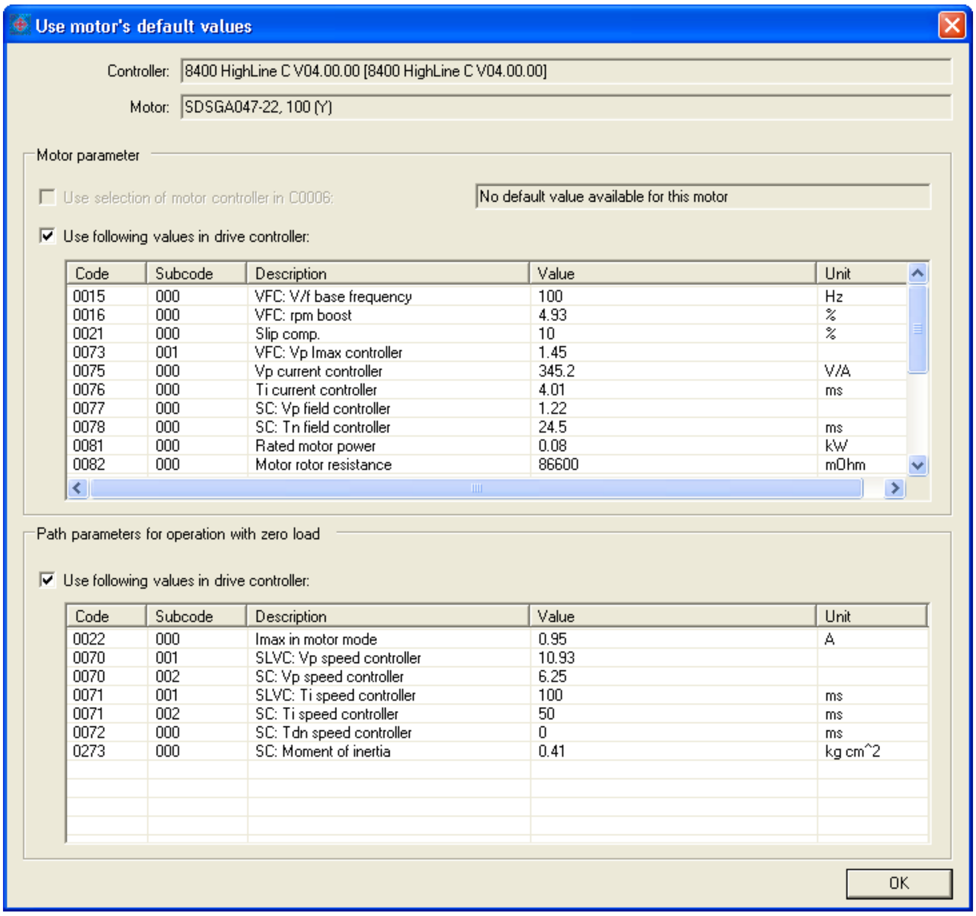
If a checkmark is set in the **Motor** control field in the "Other components" dialog when the controller is inserted into the project, the motor for the controller can be selected from the motor catalogue in another dialog:




- Alternatively, the motor can be inserted into the project at a later time via the **Insert a component** command.
- Go to the **Application parameters** tab in the *Overview* → *Motor data* dialog level and click the **From motor catalogue...** button to also reach the motor catalogue for the selection of another motor.

### Accepting the default values of the motor

If a motor is selected from the motor catalogue at a later time, the *Use motor's default values* dialog box is displayed afterwards which includes all motor data of the selected motor. Please select here which of the default values are to be copied to the controller:




- The listed motor parameters are already optimally preset for the selected Lenze motor. An adaptation is not required.
- The "plant parameter" term comprises all parameters that result from the combination of motor and load. These characterise the transfer behaviour of the entire controlled system.
  - The plant parameters depend on the application in which the controller and motor are used.
  - When a Lenze motor is selected in the »Engineer«, plant parameters are suggested for this motor for a load-free operation.

**Tip!**

If a third party manufacturer's motor is used, select a Lenze motor from the motor catalogue first which is similar in terms of current, voltage and speed rating. Adapt the preselected motor data exactly to the real motor afterwards.

### 5.1.2 Automatic motor data identification

Via the motor parameter identification, the inverter characteristic, the influences of the motor cable, and the motor parameters listed in the table below can be identified automatically:

| Parameter | Info | ASM | PSM |
|---|---|---|---|
| C00015 | V/f base frequency | ● | |
| C00016 | $V_{min}$ boost | ● | |
| C00021 | Slip compensation | ● | |
| C00082 | Motor rotor resistance | ● | |
| C00083 | Motor rotor time constant | ● | |
| C00084 | Motor stator resistance | ● | ● |
| C00085 | Motor stator leakage inductance | ● | ● |
| C00092 | Motor magnetising inductance | ● | |
| C00095 | Motor magnetising current | ● | |

For the execution of the motor parameter identification, two modes are available in C02867/1:

- "1: Basic identification"
  - Only for asynchronous motors
  - Duration approx. 30 s
- "2: extended identification"
  - Stands out due to increased accuracy of the determined motor parameters.
  - Also supports synchronous motors and asynchronous motors with a power of more than 11 kW.
  - Duration approx. 80 s

**Tip!**

In the Lenze setting, a setting of "0: automatic" is selected in C02867/1. This setting ensures that the controller automatically selects the optimum procedure for motor parameter identification.

**Danger!**

During motor parameter identification, the motor is energised via the outputs U, V and W of the controller!

Observe the corresponding safety instructions!

**Stop!**

- If motor parameter identification is aborted, unstable drive behaviour may be the result!
- The following applies to the motor parameter identification of synchronous motors:
  - Only the extended motor parameter identification can be used.
  - During the motor parameter identification, the shaft of the synchronous motor mus rotate freely (must not be locked).
  - During motor parameter identification, rotations take place.
- For asynchronous motors as of a power of 11 kW, the following applies:
  - Only the extended motor parameter identification can be used.
- When the extended motor parameter identification is started, it is decided based on the motor control selected in C00006 whether an asynchronous motor or a synchronous motor is to be identified.
  - Thus, set a suitable motor control for the motor in C00006 before starting the extended motor parameter identification! ▶ Selecting the control mode (📖 131)

**Note!**

- We strongly recommend motor parameter identification before the initial commissioning of the sensorless vector control (SLVC) and the sensorless control for synchronous motors (SLPSM).
- The motor parameter identification must be carried out when the motor is cold!
- The load machine may remain connected. Holding brakes, if present, may remain in the braking position.
- With an idling motor, a small angular offset may occur at the motor shaft.
- The amplitude of the rated motor current (C00088) is injected to identify the stator resistance. If the rated motor current amounts to less than 60 % of the rated inverter current, at least 60 % of the rated inverter current will be injected to ensure sufficient motor parameter identification accuracy.

### Preconditions

The motor parameters listed in the table below are excluded from automatic identification and must therefore be adapted to the used motor before motor parameter identification is carried out (see motor nameplate).

| Parameter | Info |
|---|---|
| C00081 | Rated motor power |
| C00087 | Rated motor speed |
| C00088 | Rated motor current |
| C00089 | Rated motor frequency |
| C00090 | Rated motor voltage |
| C00091 | Motor cos φ |

Furthermore, the available motor cable must be specified in terms of length and cross-section:

| Parameter | Info |
|---|---|
| C00915 | Motor cable length |
| C00916 | Motor cable cross-section |

### Sequence of the motor parameter identification

1. The motor stator resistance (C00084) is measured.
2. The inverter error characteristic is measured.
3. The motor stator leakage inductance (C00085) is measured

   For asynchronous motors only:
4. The motor magnetising inductance (C00092) and the motor rotor resistance (C00082) are measured.
5. The motor magnetising current (C00095) is measured.
6. The V/f base frequency (C00015) is calculated.
7. Slip compensation (C00021) is calculated.
8. $V_{min}$ boost (C00016) is detected.

### Optimising motor parameter identification

For the measurement of the required variables, the motor is energised via the controller terminals U, V and W during the motor parameter identification.

The corresponding current controller can be set via the following parameters:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00075 | Vp current controller | 7.00 | V/A |
| C00076 | Ti current controller | 10.61 | ms |

In the Lenze setting, the current controller is preset in such a way that an optimum controller behaviour is obtained for an asynchronous motor with power adaptation to the controller.

**Note!**

The motor parameter identification can be aborted in the following cases:

- If a special motor (e.g. mid-frequency motor) or a servo motor is used.
- If there is a large deviation between inverter and motor power.

In this case we recommend (with the simple motor parameter identification):

- to reduce the P component Vp of the current controller (C00075) e.g. by halving.
- to increase the time constant Ti of the current controller (C00076) e.g. by doubling.

With the extended motor parameter identification, the current controller parameters are determined automatically. If the identification is aborted all the same, the current controller parameters set in C00075 and C00076 can be used by parameterising C02866 to "1".

Another cause for the abort of the motor parameter identification could be the implausibility of the entered nameplate data, e.g. the entry P = 0 kW for the motor power.

**How to carry out automatic motor parameter identification:**

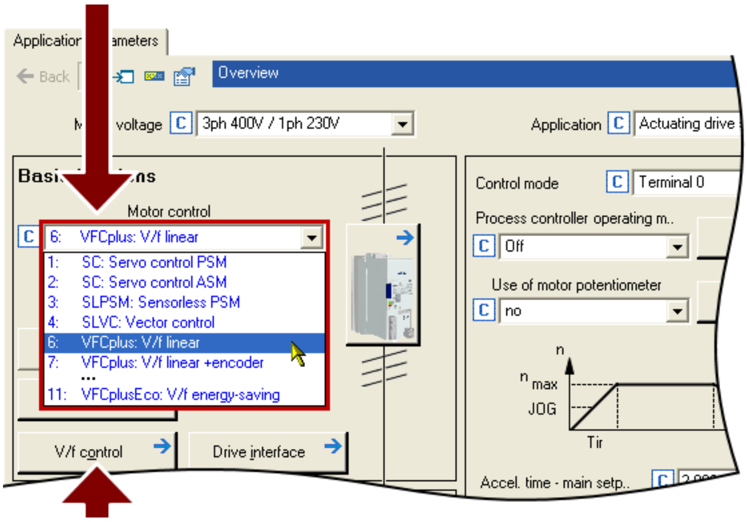
1. Inhibit the controller if it is enabled, e.g. via the C00002/16 device command or a LOW signal at the X5/RFR terminal.
   - For executing the motor parameter identification, the controller must be in the "SwitchedOn" status.
2. Wait until the drive is at standstill.
3. Ensure that a motor control suitable for the motor is set in C00006.
4. Transfer the nameplate data to the following codes:
   - C00081: Rated motor power
   - C00087: Rated motor speed
   - C00088: Rated motor current (according to the connection method Y/Δ)
   - C00089: Rated motor frequency (according to the connection method Y/Δ)
   - C00090: Rated motor voltage (according to the connection method Y/Δ)
   - C00091: Motor cos φ
5. Defining the motor cable length and motor cable cross-section:
   - C00915: Motor cable length
   - C00916: Motor cable cross-section

   The resulting motor cable resistance is displayed in C00917.
6. Activate motor parameter identification via the C00002/23 = "1: On / start" device command.
7. Enable the controller again.
   - The controller changes to the "MotorIdent" device status.
   - Motor parameter identification starts.
   - The progress of the identification run can be seen in C00002/23.
   - The identification is completed if the "0: Off / ready" message is displayed in C00002/23.
   - After successful identification, it changes back to the "SwitchedOn" device status.
8. Inhibit controller again.

## 5.2 Selecting the control mode

The 8400 TopLine controller supports various modes for motor control (open loop or closed loop).

- V/f characteristic control (VFCplus) with linear characteristic for asynchronous motors is preset.
- The control mode can be selected in the »Engineer« on the **Application parameter** tab via the **Motor control** (C00006) list field:




- A click on the **Motor control...** button leads you to the parameterisation dialog of the selected motor control. (The button is labelled according to the selected motor control.)

**Tip!**

In order to make the selection of the motor control easier, we provide a selection help with recommendations and alternatives for standard applications in the subchapter entitled "Selection help". (📖 135)

The following section briefly describe the control modes. A reference to more details can be found at the end of each section.

### V/f characteristic control (VFCplus)

The V/f characteristic control (VFCplus) is a motor control mode for standard frequency inverter applications based on a simple and robust control process which is suitable for the operation of asynchronous motors with linear or square-law load torque characteristic (e.g. fans). Furthermore, this motor control mode is also suitable for group drives and special motors. Due to the low parameterisation effort, commissioning of such applications is fast and easy.

The $V_{min}$-boost (C00016) and slip compensation (C00021) required for optimising the drive behaviour are dimensioned for asynchronous motors with power adaptations to the inverter in the Lenze setting.

▸ V/f characteristic control (VFCplus) (📖 147)

### V/f characteristic control - energy-saving (VFCplusEco)

In contrast to the V/f characteristic control mode (VFCplus), this motor control mode uses a cosφ control in partial load operational range to automatically reduce the power loss in the asynchronous motor (energy optimisation).

The motor data required for the cosφ control and the $V_{min}$ boost (C00016) and slip compensation (C00021) required for optimising the drive behaviour are dimensioned for asynchronous motors with power adaptations to the inverter in the Lenze setting.

The required motor data (motor rotor resistance, motor stator resistance, motor stator leakage inductance and mutual motor inductance) only affect the extent of energy optimisation but not the stability.

In case of applications with dynamically very high sudden load variations from the unloaded operation, this motor control mode should not be used since a motor stalling cannot be excluded.

Energy optimisation for dynamic applications is not possible with this motor control mode.

▸ V/f characteristic control - energy-saving (VFCplusEco) (📖 166)

### V/f control (VFCplus + encoder)

The V/f control can be selected for operating asynchronous motors with speed feedback. With this motor control, a slip regulator can be additionally parameterised which adjusts the actual speed value dynamically to the speed setpoint.

▸ V/f control (VFCplus + encoder) (📖 176)

#### Sensorless vector control (SLVC)

Sensorless (field-oriented) vector control for asynchronous motors is based on a decoupled, separate control for the torque-producing and the field-producing current component. In addition, the actual speed is reconstructed by means of a motor model so that a speed sensor is not required.

In comparison to the V/f characteristic control without feedback, the following can be achieved by means of sensorless vector control SLVC:

- A higher maximum torque throughout the entire speed range
- A higher speed accuracy
- A higher concentricity factor
- A higher level of efficiency
- The implementation of torque-actuated operation with speed limitation
- The limitation of the maximum torque in motor and generator mode for speed-actuated operation

**Tip!**

If a high torque without feedback is to be provided at small speeds, we recommend the "Sensorless vector control" motor control mode.

▸ Sensorless vector control (SLVC) (📖 182)

#### Sensorless control for synchronous motors (SLPSM)

This sensorless control enables an encoderless control of synchronous motors. The process is based on field-oriented control within a higher speed range (e.g. > 10 % of the rated motor speed). The actual speed value and rotor position are reconstructed via a motor model.

Standard applications for this control type are pumps and fans, horizontal materials handling and simple positioning technology.

▸ Sensorless control for synchronous motors (SLPSM) (📖 198)

#### Servo control (SC)

Field-oriented servo control (SC) is based on a decoupled, separate control of the torque-producing and the field-producing current component. The motor control is based on a field-oriented, cascaded controller structure with feedback function and enables dynamic and stable operation in all of the four quadrants.

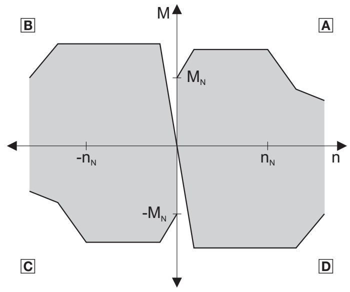
The servo control can be used for synchronous motor (PSM) and asynchronous motors (ASM) and basically offers the same advantages as the sensorless vector control (SLVC).

▸ Servo control (SC) (📖 214)

### Speed feedback

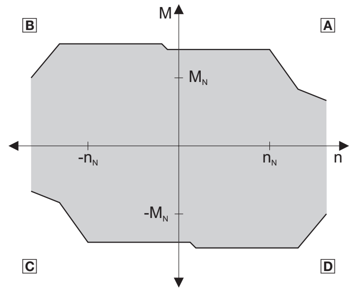
As shown in the following graphics, the drive systems with feedback have, independently of the motor control, more advantages than systems without feedback.

**V/f characteristic control (VFCplus)**




9300vec092

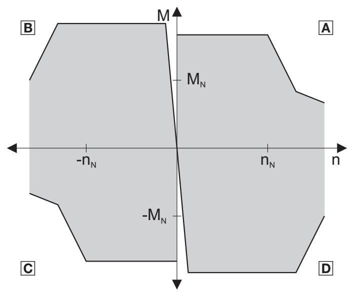
**V/f control (VFCplus + encoder)**




9300vec093

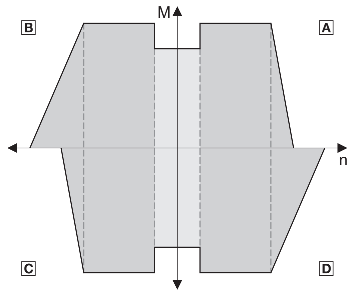
A Operation in motor mode (CW rotation), B Operation in generator mode (CCW rotation),
C Operation in motor mode (CCW rotation), D Operation in generator mode (CW rotation)

**Sensorless vector control (SLVC)**



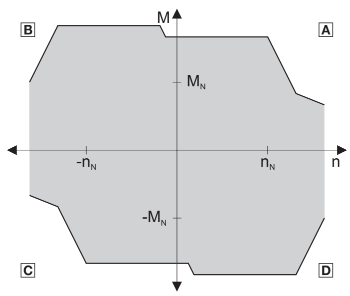

9300vec095

**Sensorless control for synchronous motors (SLPSM)**




A Operation in motor mode (CW rotation), B Operation in generator mode (CCW rotation),
C Operation in motor mode (CCW rotation), D Operation in generator mode (CW rotation)

**Servo control (SC)**




9300vec094

A Operation in motor mode (CW rotation), B Operation in generator mode (CCW rotation),
C Operation in motor mode (CCW rotation), D Operation in generator mode (CW rotation)

### 5.2.1 Selection help

To ease the selection of the motor control mode, the two following tables contain recommendations and alternatives to standard applications.

| Application | recommended | Alternatively |
|---|---|---|
| **Single drives** | | |
| With constant load | VFCplus: V/f linear | SLVC or SLPSM |
| With extremely alternating loads | VFCplus: V/f linear | SLVC |
| With high starting duty | SLVC | VFCplus: V/f linear |
| Torque limitation | SLVC | SLPSM |
| With torque limitation (power control) | VFCplus: V/f linear | SLPSM |
| Three-phase reluctance motor | VFCplus: V/f linear | - |
| Three-phase sliding rotor motor | VFCplus: V/f linear | - |
| Three-phase AC motors with permanently assigned frequency/voltage characteristic | VFCplus: V/f linear | - |
| Pump and fan drives with quadratic load characteristic | VFCplusEco | SLVC or SLPSM |
| Simple hoists | VFCplus: V/f linear | - |
| **Group drives** *(several motors connected to controller)* | | |
| Identical motors and loads | VFCplus: V/f linear | - |
| Different motors and/or alternating loads | VFCplus: V/f linear | - |

[5-1] Standard applications without speed feedback

| Application | recommended | Alternatively |
|---|---|---|
| **Single drives** | | |
| With constant load | SC | VFCplus / SLVC |
| With extremely alternating loads | SC | VFCplus / SLVC |
| With high starting duty | SC | VFCplus / SLVC |
| With speed control (speed feedback) | SC | VFCplus |
| With high dynamic performance e.g. for positioning and infeed drives | SC | - |
| Torque limitation | SC | SLVC |
| With torque limitation (power control) | - | - |
| Winder with dancer position control | SC | VFCplus |
| Unwinder with dancer position control | SC | VFCplus |
| Three-phase reluctance motor | - | - |
| Three-phase sliding rotor motor | - | - |
| Three-phase AC motors with permanently assigned frequency/voltage characteristic | - | - |
| Pump and fan drives with quadratic load characteristic | - | - |
| Simple hoists | VFCplus: V/f linear | - |
| **Group drives** *(several motors connected to controller)* | | |
| Identical motors and loads | VFCplus | - |
| Different motors and/or alternating loads | - | VFCplus |

[5-2] Standard applications with speed feedback

## 5.3 Defining current and speed limits

### Limitation of the speed setpoint

Parameterising the reference speed in C00011 means that the drive must rotate at the set speed if a speed setpoint of 100% is specified.

All speed setpoint selections are provided in % and always refer to the reference speed set in C00011.

**Tip!**

For reasons of achievable resolution and the accuracy involved, the reference speed should be geared to the speed range required for the respective application.

Lenze recommendation: Reference speed (C00011) = 1500 ... 3000 rpm

Irrespective of the selected motor control, there are more limitation options:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00909/1 | Max. positive speed | 120 | % |
| C00909/2 | Max. negative speed | 120 | % |
| C00910/1 | Max. positive output frequency | 1000 | Hz |
| C00910/2 | Max. negative output frequency | 1000 | Hz |

**Note!**

In the torque-controlled operation (*bTorquemodeOn* = TRUE), the limitation of the speed setpoint does not have any effect! In this case, a permissible speed range can be defined via speed limitation (*nSpeedHighLimit* and *nSpeedLowLimit*).

### Current limitation in motor and generator mode

In the various motor control modes, the controller is provided with functions which determine the dynamic behaviour under load and counteract exceedance of the maximum current in motor or generator mode.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00022 | Imax in motor mode | 47.00 | A |
| C00023 | Imax in generator mode<br>• 100 % ≡ Imax in motor mode (C00022) | 100 | % |

The current limits must be selected depending on

- the permissible maximum current of the motor → recommendation: $I(Mot)_N$ < 1.5 ... 2.0
- the permissible maximum current of the inverter
- the torque in motor/generator mode required for the application

**Note!**

**Highly dynamic applications**
(that have e.g. too short acceleration/deceleration times or excessively changing loads)

The overcurrent disconnection may respond (fault message OC1 or OC11) if the setting of the maximum current in motor mode in C00022 approximately corresponds to the maximum permissible value of the respective inverter.

Remedies:

- Increase of the acceleration and deceleration ramp times
- Reduction of the maximum current in motor mode (C00022)
- Reduction of the maximum current in generator mode (C00023)
- Adaptation of the indirect peak current limitation (procedure depends on the selected motor control mode, see below)
- Reduction of the reset time of the current limiting controller (C00074/1)

**Influencing the torque in motor/generator mode**

The torque in motor and generator mode can be limited via the *nTorqueMotLim* and *nTorqueGenLim* process signal inputs.

- If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.
- If sensorless vector control (SLVC), sensorless control for synchronous motors (SLPSM) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.

If keypad control is selected, the *nTorqueMotLim* and *nTorqueGenLim* process signals can be parameterised via C00728/1...2.

**How to adapt the peak current limitation:**

V/f characteristic control (VFCplus):

- Reduce the slip compensation with C00021.

V/f control (VFCplus + encoder):

- Reduce the slip limitation to twice the rated motor slip with C00971.
- Reduce the $V_{min}$ boost in C00016.

Sensorless vector control (SLVC):

- Reduce the slip compensation with C00021.
- Reduce the limitation of the torque in motor mode via *nTorqueMotLimit_a* (C00728/1) and the limitation of the torque in generator mode via *nTorqueGenLimit_a* (C00728/2).

Servo control (SC):

- Reduce the jerk limitation with C00274.
- Reduce the limitation of the torque in motor mode via *nTorqueMotLimit_a* (C00728/1) and the limitation of the torque in generator mode via *nTorqueGenLimit_a* (C00728/2).

## 5.4 Pole position identification (PPI)

**Note!**

Only required:
- For servo control with synchronous motor of a third-party manufacturer.
- For servo control with synchronous motor and use of incremental encoders (TTL or sin/cos encoders as well as multi-pole pair resolvers).
- After changes of the motor feedback system, e.g. encoder exchange.

For the sensorless control of synchronous motors (SLPSM), a pole position identification is not required.

For the control of permanent-magnet synchronous machines, the pole position – the angle between the motor phase U and the field axis of the rotor – must be known.

- For Lenze motors with absolute value encoder or resolver, the pole position is already set correctly in C00926/1...2.
- When incremental encoders (TTL or sin/cos encoders) are used, a pole position identification (PPI) is always required after mains switching, even with Lenze motors.
- The controller can also evaluate multi-pole-pair resolvers.
  - When the number of motor pole pairs is an integer multiple of the number of pole pairs of the resolver, a pole position identification must only be executed once.
  - When the number of motor pole pairs is no integer multiple of the number of pole pairs of the resolver, a pole position identification must be executed after every mains switching.
- The device command "Identify pole position (360°)" serves to detect the pole position for the motor encoder currently activated in C00495 (see the following instructions).

### 5.4.1 Pole position identification 360°

#### Procedure for "pole position identification 360°"

If all conditions are met, the motor is energised with a direct current corresponding to the lower of the following two values:

$$\sqrt{2} \cdot \text{Rated device current}$$
or
$$\sqrt{2} \cdot \text{Rated motor current}$$

- The rotor is aligned through the current flow. This is absolutely necessary for the procedure.
- To ensure that the torque-neutral axis is not accidentally energised and the rotor stops, a 45° current vector is (electrically) generated for a short instant and then (electrically) switched back to 0° (≡ phase U).
  - Then a DC current of the above-mentioned value could be measured in this motor phase.
- If a resolver or an optical encoder without absolute track is used, the difference between the preselected current angle and the mechanical rotor angle is determined. After this, the current vector is (electrically) turned by another 22.5° and the difference between current angle and rotor angle is determined once again.
  - The procedure is repeated 16 times. This corresponds to one electrical revolution. The machine rotates by 360° (mech.)/pole pair number.
  - Take the average value of the 16 measurements to compensate for asymmetries.

#### Adjustment of the pole position identification 360°

The pole position identification can be adjusted to the respective machine and the prevailing moments of inertia by means of the parameters described below:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00644/1 | PLI 360° traversing direction | right rotating field | |
| C00645/2 | PLI 360° max. error tolerance | 0.0 | ° |
| C00646/2 | PLI 360° current amplitude | 100 | % |
| C00647/2 | PLI 360° ramp time | 100 | % |

- The current amplitude can be adjusted proportionally in C00646/2.
  - For large machines and high mass inertia values or for linear direct drives, the current amplitude usually has to be increased.
  - The Lenze setting "100 %" corresponds to the smaller of the two following values:

$$\sqrt{2} \cdot \text{Rated device current}$$
or
$$\sqrt{2} \cdot \text{Rated motor current}$$

**Stop!**

If there is no temperature monitoring in the motor and/or the I2xt motor monitoring and the maximum current monitoring are not parameterised correctly, the motor may be permanently damaged when the current amplitude is set too high (e.g. to the maximum value)!

- Motor load monitoring (I2xt) (📖 276)
- Maximum current monitoring (📖 283)

**Note!**

If the current amplitude is set to > 100 % in C00646/2, the device utilisation (Ixt) monitoring and/or one of the motor monitoring functions may respond and cause the abort of the pole position identification.

- The ramp time can be adjusted proportionally in C00647/2.
  - For large machines and high mass inertia values, the ramp time usually has to be increased.
  - For small machines, a reduction of the ramp time can speed up the pole position identification process.
- In some situations it may be helpful to reverse the travel direction in C00644/2 for the pole position identification (e.g. for linear motor at the end stop).
- The "pole position identification 360°" procedure comprises a plausibility check. If the rotor position determined via the encoder system does not correspond to the controlled output position:
  - the pole position identification procedure is aborted.
  - the response parameterised in C00643/1 (Lenze setting: "Fault") is activated.
  - the error message "ID5: Pole position identification error" is entered into the logbook.
- The preset fault tolerance for the plausibility check can be changed in C00645/2.

### Execute pole position identification 360°

**Danger!**

The machine must not be braked or blocked during the pole position identification! For this reason, the pole position identification is not permitted for hanging loads!

During the pole position identification the rotor aligns itself. The motor shaft moves by max. one electrical revolution which causes the corresponding movement of the connected mechanical components!

**Stop!**

Check the correct parameterisation of the max. motor current monitoring before carrying out the pole position identification to prevent the motor from being permanently damaged.

▸ Maximum current monitoring (📖 283)

**Note!**

By means of controller inhibit, the procedure started can be cancelled anytime, if required, without carrying out a change in C00926.

If the pole position identification is aborted, the response parameterised in C00643/1 is activated (Lenze setting: "Fault").

- If this behaviour is not wanted, deactivate the monitoring by selecting "0: No response" in C00643/1.

### Preconditions for the execution

- A synchronous motor has been selected.
- No other identification is active.
- No error has occurred.

**How to execute the pole position identification:**

1. Inhibit the controller if it is enabled, e.g. via the C00002/16 device command or a LOW signal at the X5/RFR terminal.
2. Start the pole position identification with the device command C00002/34 = "1: On / start".
3. Enable the controller again.
   - The pole position identification starts.
   - The progress of the identification run can be seen in C00002/34.
   - The identification is completed if the "0: Off / ready" message is displayed in C00002/34.
4. Inhibit controller again.

**After successful completion...**

...the controller is inhibited automatically and the pole position determined for the activated feedback system is set in the corresponding subcode of C00926.

▶ For a permanent acceptance of the identified pole position, the parameter set must be saved (C00002/11 = "1: On / start").

▶ The next controller inhibit and subsequent controller enable serve to cancel the controller inhibit automatically set by the procedure (e.g. by first executing the device command C00002/16 = "0: Off / ready" and then executing the device command C00002/16 = "1: On / start").

**In the event of an error**

If an error occurs during the procedure or the pulse inhibit gets active (e.g. due to short-time undervoltage), the procedure is terminated with controller inhibit without making a change in C00926.

If the machine was braked or blocked during the procedure, this will be recognised at the end of the measurement and no change is made in C00926.

▶ If the pole position identification is aborted:
   – the response parameterised in C00643/1 (Lenze setting: "Fault") is activated.
   – the error message "ID5: Pole position identification error" is entered into the logbook.

### 5.4.2 Pole position identification without motion

This function extension is available from version 02.00.00!

**Note!**

Function only possible with:

- Servo control (SC) for synchronous motors
- Sensorless control for synchronous motors (SLPSM)

From version 02.00.00, a pole position identification without motion is also possible in case of servo control and sensorless control.

- For sensorless control of synchronous motors (SLPSM), this function is already activated in the Lenze setting, i.e. with every controller enable the rotor displacement angle is identified and thus jerks in the machine after controller enable can be avoided.
  - In order to achieve the same behaviour as before, set bit 0 to "0" in C02874.
- For servo control, this function can be activated for various events in C02874 via the bits 1 ... 3:
  - Bit 1: Pole position identification after mains connection
  - Bit 2: Pole position identification after controller enable
  - Bit 3: Pole position identification after error acknowledgement

**Note!**

- The "Pole position identification without motion" cannot completely replace the "pole position identification 360°"!
  - The electrical rotor displacement angle can only be electrically identified with an error of up to 10°. This inaccuracy can cause a worse torque accuracy and a worse energy efficiency.
- The identification takes motor-dependent 1 ... 15 ms. The setpoint enabling of the device is reduced by this time.
- In the Lenze setting, the function is preset so that in most of the cases no further settings have to be made.

**Tip!**

In case of servo control (SC), it is generally sufficient to activate this function only once after mains connection of the device (C02874: Bit 1 = "1").

Typical application case: A speed sensor with an unknown pole position is used (e.g. encoder). The pole position identification 360°, however, cannot be used since, e.g. the motor is locked, the application does not permit it or the identification takes too long.

### Short overview of the relevant parameters:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02874 | PLI without motion | 0x1 (bit coded) | |
| C02872 | PLI without motion: Adaptation of time duration | 0 | |
| C02875 | PLI without motion: Adaptation of ident angle | 0 | ° |
| C02870 | PLI without motion: Optimisation factor | - | % |
| C02871 | PLI without motion: Running time | - | ms |
| C02873 | PPI without motion: Identified rotor displacement angle | - | ° |
| Highlighted in grey = display parameter | | | |

**Note!**

In case of synchronous motors with a stator time constant < 1 ms, the pole position identification is not executed since the resulting test current pulse could exceed the permissible motor current.

- This, however, only affects very few synchronous motors with very low power (e.g. Lenze motor MDSKS-020-13-300 with rated power of 40 W).
- A non-executed pole position identification can be recognised by the display C02870 = 0 % and C02871 = 0 ms.

▶ The stator time constants can be calculated based on the following formula:

$$T_s[\mathrm{ms}] = \frac{L_{ss}[\mathrm{mH}]}{R_s[\Omega]}$$

$T_S$ = stator time constant
$L_{ss}$ = Motor stator leakage inductance (C00085)
$R_s$ = Motor stator resistance (C00084)

### Optimising the pole position identification

**Stop!**

In case of a too high setting in C02872, an impermissible motor current may occur during pole position identification. In this case, the "Fault" error response occurs and the "ID5: Pole position identification error" error message entered in the logbook.

In case of aconsiderably too high setting in C02872:

- The following current monitoring functions can be triggered:
  - OC7: Motor overcurrent
  - OC11: Active clamp operation
  - OC1: Power section - short circuit
- The optimisation degree "0 %" is displayed in C02870.
- The time "0 ms" is displayed in C02871.

**How to optimise the pole position identification without motion:**

1. For optimisation, enable the controller for various rotor displacement angles.
2. After every controller enable, check the optimisation degree displayed in C02870.

   The pole position identification is set optimally if an optimisation degree in the range 70 ... 130 % is displayed in C02870 after every controller enable.
3. When the optimisation degree is > 130 %:
   Reduce the setting in C02872 step by step an execute controller enable for various rotor displacement angles until an optimisation degree < 130 % is displayed.
4. When the optimisation degree is < 70 %:
   Increase setting in C02872 step by step and execute controller enable for various rotor displacement angles until an optimisation degree > 70 % is displayed.
5. Optional: C02875 serves to increase or reduce the identified electrical rotor displacement angle. This can prevent e.g. a reversing of the motor due to the accuracy of identification if applications require it.

## 5.5 V/f characteristic control (VFCplus)

In case of the V/f characteristic control (VFCplus), the motor voltage of the inverter is determined by means of a linear or quadratic characteristic depending on the field frequency or motor speed to be generated. The voltage follows a preselected characteristic.

**Stop!**

- The V/f characteristic control is only suitable for asynchronous motors.
- The following must be observed when operating drives with quadratic V/f characteristic:
  - Please always check whether the corresponding drive is suitable for operation with a quadratic V/f characteristic!
  - If you pump or fan drive is not suitable for operation with a square-law V/f characteristic, we recommend using the energy-saving V/f characteristic control (VFCplusEco). Alternatively, you can use the V/f characteristic control with linear V/f characteristic or the sensorless vector control (SLVC) or servo control (SC).
- For adjustment, observe the thermal performance of the connected asynchronous motor at low output frequencies.
  - Usually, standard asynchronous motors with insulation class B can be operated for a short time with their rated current in the frequency range 0 Hz ... 25 Hz.
  - Contact the motor manufacturer to get the exact setting values for the max. permissible motor current of self-ventilated motors in the lower speed range.
  - If you select square-law V/f characteristics, we recommend setting a lower $V_{min}$ or using the energy-saving V/f characteristic control (VFCplusEco).
- The nameplate data of the motor (at least rated speed and rated frequency) must be entered if, instead of a standard motor, an asynchronous motor is used with the following values:
  - rated frequency $\neq$ 50 Hz (star) or
  - rated frequency $\neq$ 87 Hz (delta) or
  - number of pole pairs $\neq$ 2

**Note!**

When the auto DCB threshold (C00019) is set > 0 rpm, there is no torque at the motor shaft in the lower speed range!

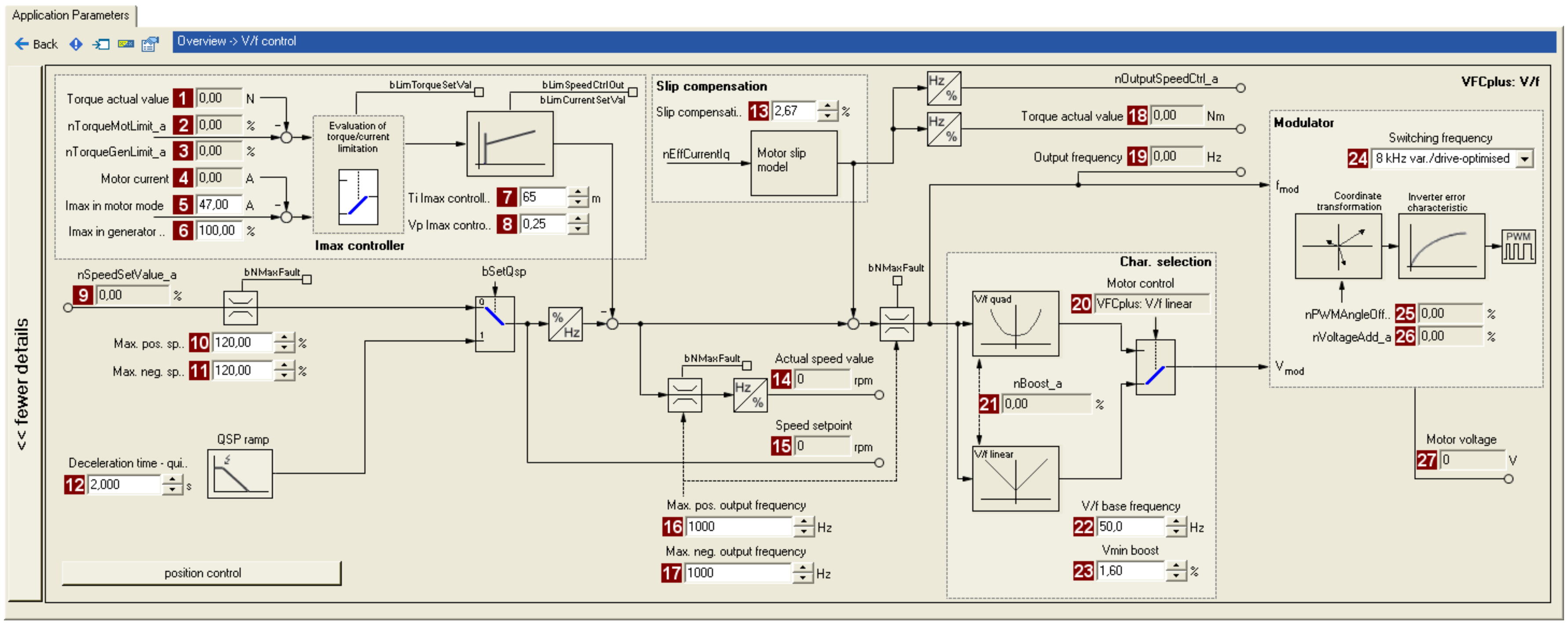
▶ Automatic DC-injection braking (Auto-DCB) (📖 251)

### 5.5.1 Parameterisation dialog/signal flow

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Select the motor control from the *Overview* dialog level in the **Motor control** list field:
   - "6: VFCplus: V/f linear" for linear characteristic or
   - "8: VFCplus: V/f quadr" for square-law characteristic

   More available V/f characteristic control modes:
   - "10: VFCplus: V/f definable".
     With this motor control, the V/f characteristic can be freely defined. ▶ Defining a user-defined V/f characteristic (📖 162)
   - "11: VFCplusEco: V/f energy-saving".
     With this motor control, the motor is always operated in an optimal efficiency range via a cosφ control and the resulting voltage reduction (reduced copper losses in the asynchronous motor). ▶ V/f characteristic control - energy-saving (VFCplusEco) (📖 166)
4. Click the **Motor control V/f** button to change to the *Overview* → *Motor control V/f* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed.




| | Parameter | Info |
|---|---|---|
| 1 | C00056/2 | Actual torque value |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00054 | Motor current |
| 5 | C00022 | Imax in motor mode |
| 6 | C00023 | Imax in generator mode |
| 7 | C00074 | Ti Imax controller |
| 8 | C00073 | Vp Imax controller |
| 9 | C00830/22 | Speed setpoint |
| 10 | C00909/1 | Max. pos. speed |
| 11 | C00909/2 | Max. neg. speed |
| 12 | C00105 | Decel. time - quick stop |

| | Parameter | Info |
|---|---|---|
| 13 | C00021 | Slip compensation |
| 14 | C00051 | Actual speed value |
| 15 | C00050 | Speed setpoint |
| 16 | C00910/1 | Max. pos. output frequency |
| 17 | C00910/2 | Max. neg. output frequency |

| | Parameter | Info |
|---|---|---|
| 18 | C00056/2 | Actual torque value |
| 19 | C00058 | Output frequency |
| 20 | C00006 | Motor control |
| 21 | C00830/26 | MCTRL: nBoost_a |
| 22 | C00015 | V/f base frequency |
| 23 | C00016 | Vmin boost |
| 24 | C00018 | Switching frequency |
| 25 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 26 | C00830/31 | MCTRL: nVoltageAdd_a |
| 27 | C00052 | Motor voltage |

### 5.5.2 Basic settings

The "Initial commissioning steps" listed in the table below are sufficient for a simple characteristic control.

- Detailed information on the individual steps can be found in the following subchapters.

| Initial commissioning steps | |
|---|---|
| 1. | Defining the V/f characteristic shape. (📖 151) |
| 2. | Defining current limits (Imax controller). (📖 152) |

**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in the chapter "Optimising the control mode". (📖 153)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 240)

### 5.5.2.1 Defining the V/f characteristic shape

In principle, four different characteristic shapes can be stipulated:

1. **Linear V/f characteristic**:
   For drives for a constant, speed-independent load torque.
2. **Quadratic V/f characteristic**:
   For drives with a load torque curve which is quadratic or in relation to speed. Quadratic V/f characteristics are preferred in the case of centrifugal pumps and fan drives.
3. **Freely definable V/f characteristic**:
   For drives that require adaptation of the magnetising current by means of the output speed. The freely definable V/f characteristic can be used e.g. for operation in conjunction with special machines such as reluctance motors in order to suppress oscillations at the machine or to optimise energy consumption.


[5-3] Principle of a linear V/f characteristic (on the left) and a quadratic V/f characteristic (on the right)

4. **Linear V/f characteristic with voltage reduction**:
   For drives which often work in partial load operation, the energy-saving V/f characteristic control (VFCplusEco) offers the opportunity to reduce the voltage at low load in order to save energy. At higher loads, the voltage reduction is cancelled and a linear characteristic is caused.

The V/f characteristic shape is defined by selecting the corresponding motor control mode in C00006:

| V/f characteristic shape | Motor control to be selected (C00006) |
|---|---|
| Linear V/f characteristic | 6: VFCplus: V/f linear |
| Square-law V/f characteristic | 8: VFCplus: V/f quadr |
| User-definable V/f characteristic | 10: VFCplus: V/f definable |
| Linear V/f characteristic with voltage reduction | 11: VFCplusEco: V/f energy-saving |

**Tip!**

- You can find detailed information on freely definable V/f characteristics in the subchapter entitled "Defining a user-defined V/f characteristic". (📖 162)
- You can find detailed information on the linear V/f characteristic with voltage reduction in the chapter entitled "V/f characteristic control - energy-saving (VFCplusEco)". (📖 166)

### 5.5.2.2 Defining current limits (Imax controller)

The V/f characteristic control (VFCplus) and the V/f control (VFCplus + encoder) operating modes are provided with a current limitation control which is decisive for the dynamic behaviour under load and counteracts exceedance of the maximum current in motor or generator mode. This current limitation control is called $I_{max}$ control.

- The efficiency (motor current) measured by the $I_{max}$ control is compared with the current limit value for motor load set in C00022 and the current limit value for generator load set in C00023.
- If the current limit values are exceeded, the controller changes its dynamic behaviour.

#### Motor overload during acceleration

The controller prolongs the acceleration ramp to keep the current on or below the current limit.

#### Generator overload during deceleration

The controller prolongs the deceleration ramp to keep the current on or below the current limit.

#### Increasing load with constant speed

- If the motor current limit value is reached:
  - The controller reduces the effective speed setpoint until a stable working point is set or an effective speed setpoint of 0 rpm is reached.
  - If the load is reduced, the controller increases the effective speed setpoint until the setpoint speed is reached or the load reaches the current limit value again.
- When the generator current limit value is reached:
  - The controller increases the effective speed setpoint until a stable working point is set or the maximally permissible speed (C00909) or output frequency is reached (C000910).
  - If the load is reduced, the controller reduces the effective speed setpoint until the setpoint speed is reached or the load reaches the current limit value again.
- If a sudden load is built up at the motor shaft (e.g. drive is blocked), the overcurrent disconnection may respond (fault message OC1 or OC11).

### 5.5.3 Optimising the control mode

The V/f characteristic control (VFCplus) is generally ready for operation. It can be adapted subsequently by adapting the characteristic and/or the drive behaviour.

#### Adapting characteristic

For the linear and quadratic characteristic, it is also possible to match its curve to different load profiles or motors by adapting the V/f base frequency (C00015) and the $V_{min}$ boost (C00016).

▸ Adapting the V/f base frequency (📖 154)

▸ Adapting the Vmin boost (📖 156)

#### Freely defining the characteristic

The V/f characteristic can also be defined freely if the linear and quadratic characteristics are not suitable.

▸ Defining a user-defined V/f characteristic (📖 162)

#### Adapting drive behaviour

- Limitation of the maximum current by a current limitation controller (e.g. to prevent the motor from stalling or to limit to the maximally permissible motor current). ▸ Optimising the Imax controller (📖 158)
- Adaptation of the field frequency by a load-dependent slip compensation (improved speed accuracy for systems without feedback)
- Adaptation of the controller parameters of the slip regulator if V/f control (VFCplus + encoder) is selected. ▸ Parameterising the slip regulator (📖 179)

### 5.5.3.1 Adapting the V/f base frequency

The V/f base frequency (C00015) determines the slope of the V/f characteristic and has considerable influence on the current, torque, and power performance of the motor.

- The setting in C00015 applies to all permitted mains voltages.
- Mains fluctuations or fluctuations of the DC-bus voltage (operation in generator mode) do not need to be considered when the V/f base frequency is set. They are automatically compensated for by the internal mains voltage compensation of the device.
- Depending on the setting in C00015, it may be required to adapt the reference speed (C00011) to traverse the entire speed range of the motor.
- The V/f base frequency is automatically calculated from the stored motor nameplate data by the motor parameter identification:

$$\mathrm{C00015}\ [\mathrm{Hz}] = \frac{U_{FI}\ [\mathrm{V}]}{U_{Ratedmot}\ [\mathrm{V}]} \cdot f_{Rated}\ \mathrm{Hz}$$

$U_{FU}$: Mains voltage 400 V or 230 V
$U_{ratedmot}$: Rated motor voltage depending on the connection method
$f_{rated}$: Rated motor frequency

[5-4] Calculation of the V/f base frequency

#### Typical values of the V/f base frequency

| Drive controller with 400 V mains connection | | | |
|---|---|---|---|
| **Motor voltage [V]** | **Motor frequency [Hz]** | **Motor connection** | **V/f base frequency (C00015)** |
| 230 / 400 | 50 | Y | 50 Hz |
| 220 / 380 | 50 | Y | 52.6 Hz |
| 280 / 480 | 60 | Y | 50 Hz |
| 400 / 690<br>400 | 50<br>50 | △ | 50 Hz |
| 230 / 400<br>280 / 480<br>400 | 50<br>60<br>87 | △ | 87 Hz |
| 220 / 380 | 50 | △ | 90.9 Hz |

| Drive controller with 230 V mains connection | | | |
|---|---|---|---|
| **Motor voltage [V]** | **Motor frequency [Hz]** | **Motor connection** | **V/f base frequency (C00015)** |
| 230 | 50 | △ | 50 Hz |
| 220 / 380 | 50 | △ | 52.3 Hz |

**Note!**

**87-Hz operation**

4-pole asynchronous motors which are designed for a rated frequency of f = 50 Hz in star connection can be operated in delta connection when being constantly excited up to f = 87 Hz.

- Advantages:
  - Higher speed-setting range
  - 73% higher power output in case of standard motors
- Motor current and motor power increase by the factor $\sqrt{3}$.
- The field weakening range starts above 87 Hz.
- Generally, this process can also be used with motors which have different numbers of pole pairs. In case of 2-pole asynchronous motors, the mechanical limit speed must be maintained.

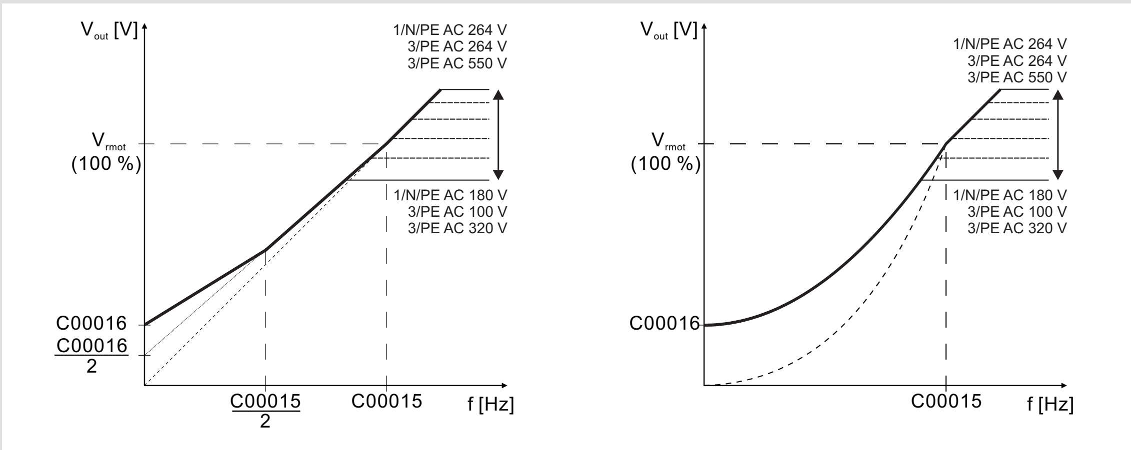
### 5.5.3.2 Adapting the Vmin boost

The $V_{min}$ boost (C00016) of the motor voltage serves to select a load independent magnetising current which is required for asynchronous motors. The torque behaviour of the motor can be optimised by adapting the setting in C00016.

**Note!**

The $V_{min}$ boost has an effect on output frequencies below the V/f base frequency (C00015).

The general linear and quadratic V/f characteristics are shown in the illustrations below. The illustrations show the impacts of the parameters used to adapt the characteristic shape.




[5-5] Representation of the linear V/f characteristic (on the left) and quadratic V/f characteristic (on the right)

**How to set the $V_{min}$ boost:**

1. Operate motor in idle state at approx. 6 % of the rated motor speed.
2. Increase $V_{min}$ boost (C00016) until the following motor current is reached:

   Motor in short-time operation up to 0.5 $n_{rated}$

   - for self-ventilated motors: $I_{motor} \approx I_{rated\ motor}$
   - for forced ventilated motors: $I_{motor} \approx I_{rated\ motor}$

   Motor in continuous operation up to 0.5 $n_{rated}$

   - for self-ventilated motors: $I_{motor} \approx 0.8\ I_{rated\ motor}$
   - for forced ventilated motors: $I_{motor} \approx I_{rated\ motor}$

**Note!**

$V_{min}$ boost is automatically calculated by the motor parameter identification using the data specified on the motor nameplate so that a no-load current of approx. 0.8 $I_{rated\ motor}$ results at the slip frequency of the machine.

**V/f control (VFCplus + encoder)**

If V/f control (VFCplus + encoder) is selected, we recommend a decidedly lower $V_{min}$ boost:

- In this case, select a $V_{min}$ boost which ensures that approx. 50 % of the rated motor current flows at slip frequency when the motor is idling.

### 5.5.3.3 Optimising the Imax controller

Using the Lenze setting of the current limitation controller, the drive is stable:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00073/1 | VFC: Vp Imax controller | 0.25 | |
| C00074/1 | VFC: Ti Imax controller | 65 | ms |

Most applications do not require optimisation.

The setting of the current limitation controller must be adapted if

- power control including great moments of inertia is performed.
  - Recommendation: Increase of the reset time Ti (C00074/1) of the $I_{max}$ controller.
- vibrations occur in the V/f control (VFCplus + encoder) mode during the intervention of the current limitation controller.
  - Recommendation: Increase of the reset time Ti (C00074/1) of the $I_{max}$ controller.
- overcurrent errors (e.g. OC3) occur due to load impulses or too high acceleration ramps.
  - Recommendation: Reduction of the gain Vp (C00073/1) and reset time Ti (C00074/1) of the $I_{max}$ controller.

### 5.5.3.4 Optimising the stalling behaviour

Motor stalling due to a torque overload in the field weakening range is prevented in all characteristic-based motor control types (VFCplus) by means of an inverter-internal stalling current monitoring. In the field weakening range, hence at frequencies above the base frequency, it reduces the maximum current to prevent the motor from stalling. The reduction depends on the current field frequency, the base frequency, the DC-bus voltage and the maximum current (C00022). Generally it applies that a higher field frequency causes a stronger limitation of the maximum current.

The behaviour in the field weakening range can be adapted via the override point of field weakening (C00080). This parameter serves to shift the frequency-dependent maximum current characteristic:

- C00080 > 0 Hz:
  - The maximum current characteristic is shifted by the entered frequency to higher field frequencies.
  - The maximally permissible current and the maximum torque increase in the field weakening range.
  - The risk of motor stalling increases.
- C00080 < 0 Hz:
  - The maximum current characteristic is shifted by the entered frequency to lower field frequencies.
  - The maximally permissible current and the maximum torque are reduced in the field weakening range.
  - The risk of motor stalling is reduced.

**Note!**

We recommend to keep the Lenze setting (0 Hz).

### 5.5.3.5 Torque limitation

The previous chapter, "Optimising the Imax controller", describes how the drive can be protected from overload. During commissioning, these settings are carried out once and remain unchanged afterwards. However, it is often necessary to limit the torque to a lower value for plant or process reasons.

▶ To avoid overload in the drive train, the torque in motor mode can be limited via the *nTorqueMotLimit_a* process input signal, and the torque in generator mode can be limited via the *nTorqueGenLimit_a* process input signal:

| **Identifier** DIS code \| data type | **Information/possible settings** |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

**Note!**

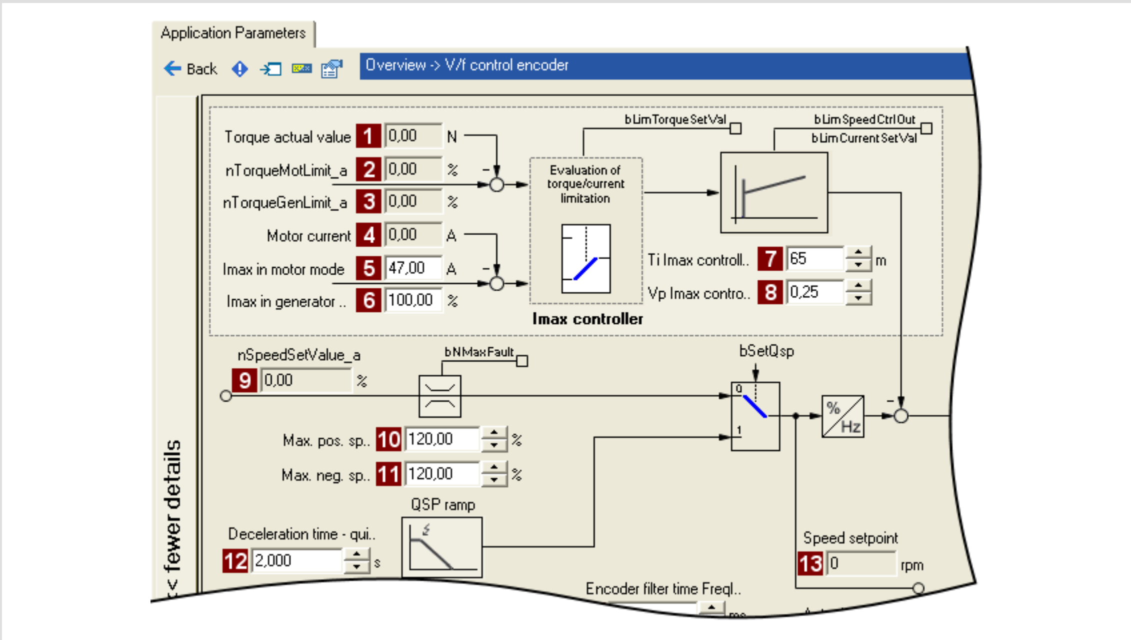
- The actual torque (C00056/2) is directly calculated from the current slip speed of the machine. This requires correct entry of the motor data. ▸ Motor selection/Motor data (🕮 121)
- To avoid instabilities during operation with active slip compensation, the torque limit values are internally processed as absolute values.
- If slip compensation is deactivated (C00021 = 0), indirect torque limitation (differential signal between the apparent motor current and *nTorqueMotLimit_a* or *nTorqueGenLimit_a*). Above the no-load current of the motor, the accuracy of the indirect torque limitation is limited.

### V/f characteristic control (VFC)

The accuracy of the torque limitation is limited because the actual torque (C00056/2) is only calculated from the slip speed measured indirectly via the motor current.

### V/f control (VFC + encoder)

The slip speed of the motor is available at the slip controller output. This leads to a high accuracy for the actual torque (C00056/2) and the torque limitation.




[5-6] Extract from the signal flow of the V/f control (VFC + encoder)

| | Parameter | Info |
|---|---|---|
| 1 | C00056/2 | Actual torque value |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00054 | Motor current |
| 5 | C00022 | Imax in motor mode |
| 6 | C00023 | Imax in generator mode |
| 7 | C00074 | Ti Imax controller |
| 8 | C00073 | Vp Imax controller |

| | Parameter | Info |
|---|---|---|
| 9 | C00830/22 | MCTRL: nSpeedSetValue_a |
| 10 | C00909/1 | Max. pos. speed |
| 11 | C00909/2 | Max. neg. speed |
| 12 | C00105 | Decel. time - quick stop |
| 13 | C00050 | Speed setpoint |

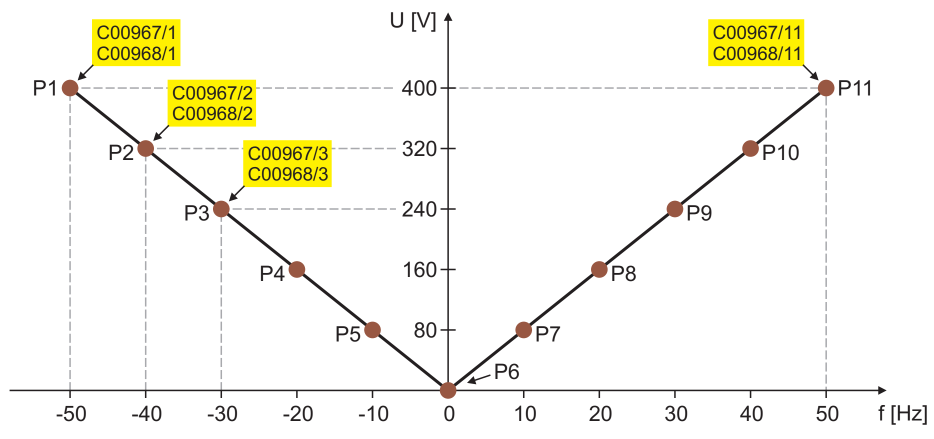
### 5.5.3.6 Defining a user-defined V/f characteristic

For individual adaptation of the motor magnetisation to the actual application, the motor control "10: VFCplus: V/f definable" with a freely definable characteristic can be selected in C00006 as an alternative if the linear and quadratic characteristics are not suitable.

**Note!**

The V/f base frequency (C00015) and the $V_{min}$ boost (C00016) no longer exert an influence if this motor control is chosen.

- The 11 grid points (voltage/frequency values) of the characteristic are stipulated by means of the 11 subcodes of C00967 and C00968.
  - It is necessary to set all 11 grid points by means of corresponding subcodes.
  - If fewer grid points (voltage/frequency values) are needed, this can be achieved indirectly by ascribing the same voltage and frequency values to consecutive grid points.
    Example: C00967/**3** = C00967/**4** and C00968/**3** = C00968/**4**
  - The grid points can be specified in any sequence. Internally, they are automatically ordered from the minimum to the maximum frequency value.
  - Above the maximum and below the minimum frequency, the previous rise is continued until the maximum output voltage.
- In the Lenze setting, the 11 grid points represent a linear characteristic.
  - 3-phase devices: Output voltage 400 V at f = 50 Hz
  - 1-phase devices: Output voltage 230 V at f = 50 Hz




| | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| V | 400 V | 320 V | 240 V | 160 V | 80 V | 0 V | 80 V | 160 V | 240 V | 320 V | 400 V |
| f | -50 Hz | -40 Hz | -30 Hz | -20 Hz | -10 Hz | 0 Hz | 10 Hz | 20 Hz | 30 Hz | 40 Hz | 50 Hz |

[5-7] Freely definable characteristic (Lenze setting for 3-phase devices)

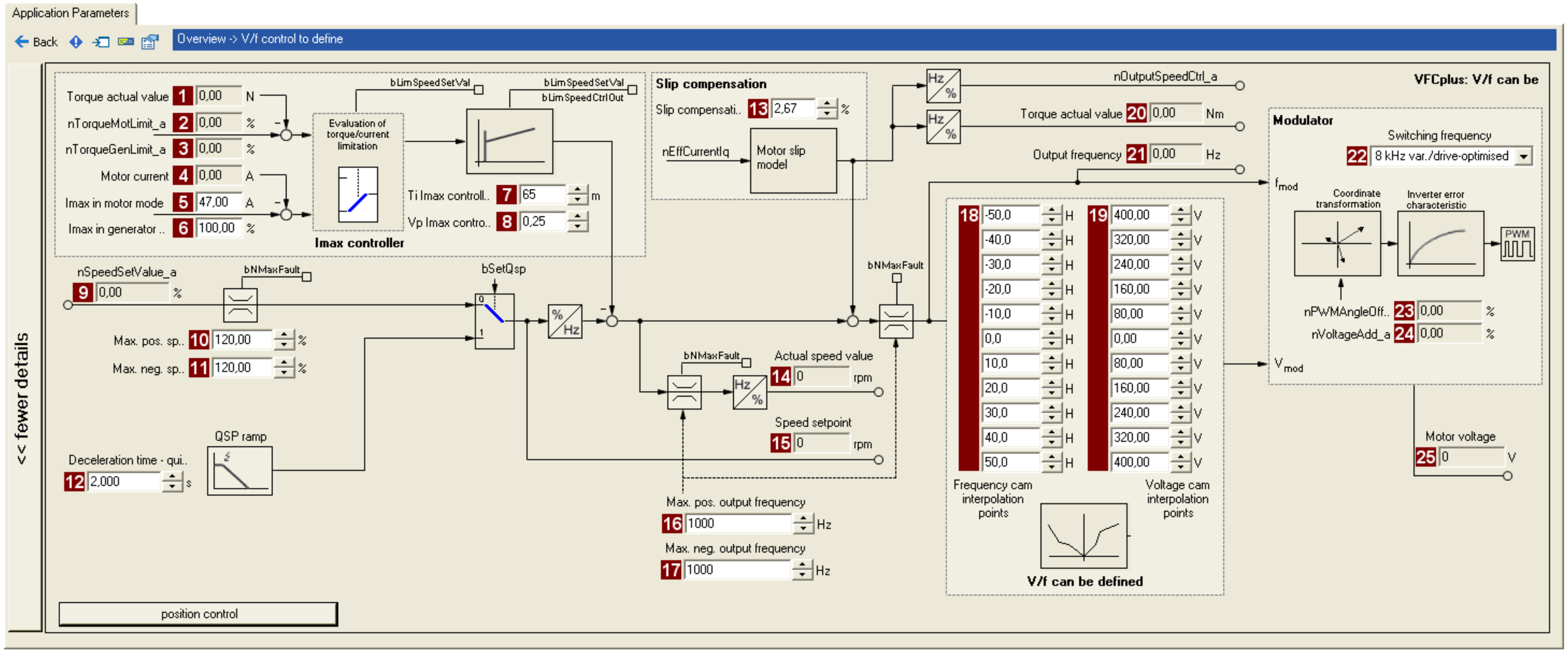
**Tip!**

Cases of application for this function:

- Operation of reluctance motors or synchronous motors during controlled acceleration (reduction of natural frequencies caused by wrong excitation).
- Adaptation of the voltage requirement for the motor, depending on specific load conditions.

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Select the motor control "10: VFCplus: V/f definable" from the *Overview* dialog box in the **Motor control** list field:
4. Click the **Motor control V/f definable** button to change to the *Overview* → *Motor control V/f* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed.




| Parameter | | Info |
|---|---|---|
| 1 | C00056/2 | Actual torque value |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00054 | Motor current |
| 5 | C00022 | Imax in motor mode |
| 6 | C00023 | Imax in generator mode |
| 7 | C00074 | Ti Imax controller |
| 8 | C00073 | Vp Imax controller |
| 9 | C00830/22 | Speed setpoint |
| 10 | C00909/1 | Max. pos. speed |
| 11 | C00909/2 | Max. neg. speed |
| 12 | C00105 | Decel. time - quick stop |

| Parameter | | Info |
|---|---|---|
| 13 | C00021 | Slip compensation |
| 14 | C00051 | Actual speed value |
| 15 | C00050 | Speed setpoint |
| 16 | C00910/1 | Max. pos. output frequency |
| 17 | C00910/2 | Max. neg. output frequency |

| Parameter | | Info |
|---|---|---|
| 18 | C00967/x | Frequency interpol. points |
| 19 | C00968/x | Voltage interpol. points |
| 20 | C00056/2 | Actual torque value |
| 21 | C00058 | Output frequency |
| 22 | C00018 | Switching frequency |
| 23 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 24 | C00830/31 | MCTRL: nVoltageAdd_a |
| 25 | C00052 | Motor voltage |

### 5.5.4 Remedies for undesired drive behaviour

| Drive behaviour | Remedy |
|---|---|
| Inadequate smooth running at low speeds, especially in the case of operation with a long motor cable | ▶ Automatic motor data identification (📖 126) |
| Problems in case of high starting duty (great mass inertia) | ▶ Adapting the Vmin boost (📖 156) |
| Drive does not follow the speed setpoint. | The current controller intervenes in the set field frequency to limit the controller output current to the maximum current (C0022, C0023). Therefore:<br>• Prolong acceleration/deceleration times:<br>C00012: Accel. time - main setpoint<br>C00013: Decel. time - main setpoint<br>• Consider a sufficient magnetising time of the motor. Depending on the motor power, the magnetising time amounts to 0.1 ... 0.2 s.<br>• Increase the maximally permissible current:<br>C00022: Imax in motor mode<br>C00023: Imax in generator mode) |
| For operation without speed feedback (C00006 = 6): Insufficient speed constancy at high load (setpoint and motor speed are not proportional anymore) | • Increase slip compensation (C00021).<br>Important: Unstable drive due to overcompensation!<br>• With cyclic load impulses (e. g. centrifugal pump), a smooth motor characteristic is achieved by smaller values in C00021 (possibly negative values).<br>Note: The slip compensation is only active for operation without speed feedback. |
| "Clamp operation active" error message (OC11): Controller cannot follow dynamic processes, i.e. too short acceleration/deceleration times in terms of load ratios. | • Increase the gain of the $I_{max}$ controller (C00073/1)<br>• Reduce the reset time of the $I_{max}$ controller (C00074/1)<br>• Prolong the acceleration time (C00012)<br>• Prolong the deceleration time (C00013) |
| Motor stalling in the field weakening range (adaptation especially required for small machines) | • Reduce the override point of field weakening (C00080)<br>• If motor power < inverter power:<br>Set C00022 to $I_{max} = 2\ I_{rated\ motor}$<br>• Reduce dynamic performance of setpoint generation |

## 5.6 V/f characteristic control - energy-saving (VFCplusEco)

With the energy-saving V/f characteristic control mode (VFCplusEco), the motor voltage of the inverter is detected by means of a linear characteristic depending on the field frequency to be created or the motor speed. Moreover, a cosφ control and the resulting voltage reduction causes the motor to be always operated in the optimum efficiency range (reduction of copper losses in the asynchronous motor).

- Hence, these are the advantages of this motor control mode:
  - Good robustness
  - Easy parameter setting
  - High energy efficiency (lower heating of the motor in partial load operational range)
  - Same speed accuracy and maximum torques as with VFCplus
- Predestinated application areas of this motor control mode are materials handling technology and pump and fan systems.
- This motor control mode serves to improve efficiency of standard asynchronous motors with efficiency class IE1 (standard IEC 60034-30 2008) in the range 0 ... $M_{efficiency\_max}$ between 0 ... 20 % (Ø 5 ... 10 %).
  - For asynchronous motors with energy efficiency class IE2 the potential for efficiency improvement is reduced to approx. 0 ... 15 %.
  - Description of $M_{efficiency\_max}$: Indicates the torque [%] of $M_{rated\_motor}$, where the motor has the max. efficiency.)
- In case of asynchronous motors with a higher energy efficiency class (IE2 and IE3), the absolute energy saving of the motor control mode is lower due to improved efficiency of the machine. However, energy saving is still achieved in a higher load range.
- $M_{efficiency\_max}$ is performance-related and listed in the following table for some power values of the energy efficiency class IE1 and IE2:

| | $M_{Efficiency\_max}$ (related to $M_{rated\_motor}$) | |
|---|---|---|
| **Power** | **IE1** | **IE2** |
| 0.25 kW | 75 % | |
| 0.75 kW | 65 % | 75 % |
| 2.2 kW | 55 % | 85 % |
| 7.5 kW | 30 % | 45 % |
| 22 kW | 23 % | |
| 45 kW | 21 % | |

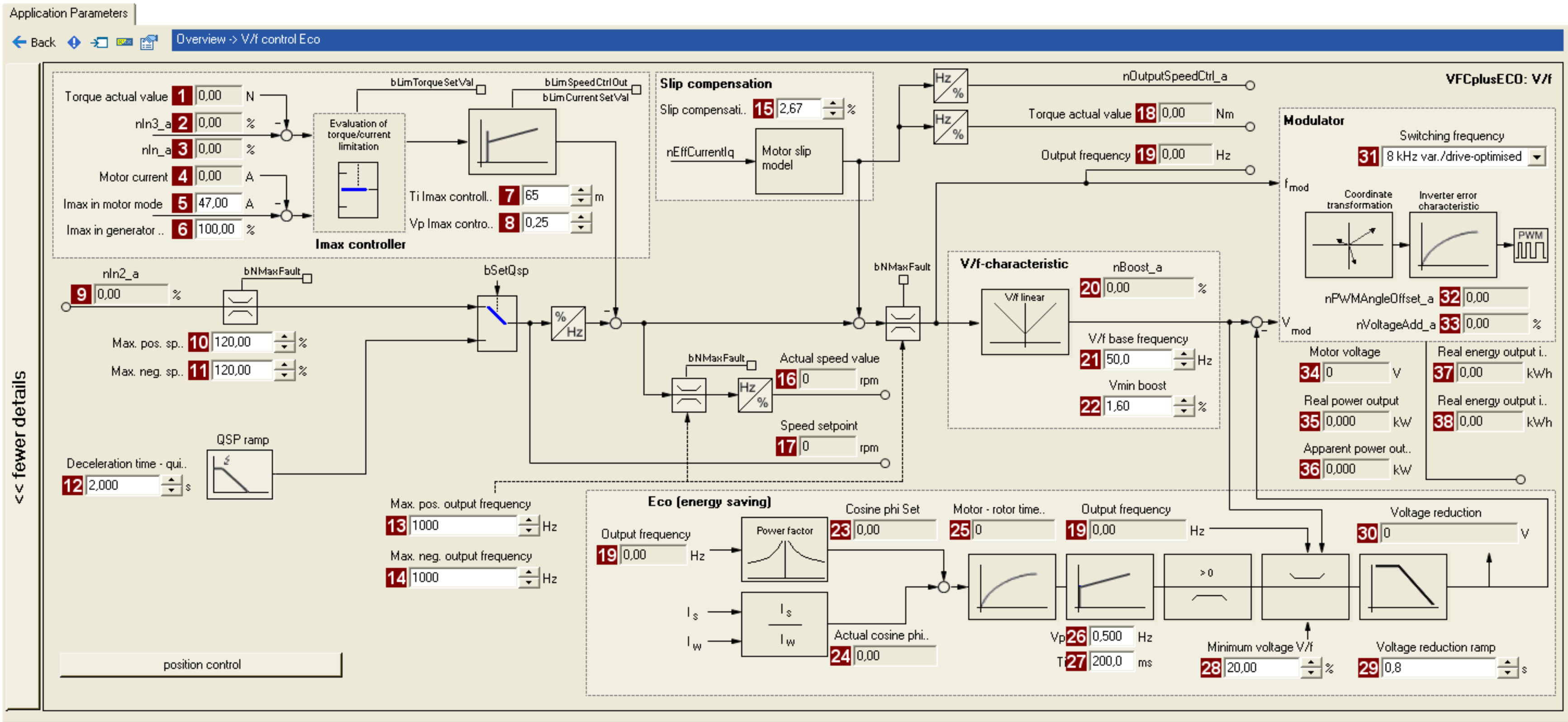
**Stop!**

- For adjustment, observe the thermal performance of the connected asynchronous motor at low output frequencies.
  – Usually, standard asynchronous motors with insulation class B can be operated for a short time with their rated current in the frequency range 0 Hz ... 25 Hz.
  – Contact the motor manufacturer to get the exact setting values for the max. permissible motor current of self-ventilated motors in the lower speed range.
- The nameplate data of the motor (at least rated speed and rated frequency) must be entered if, instead of a standard motor, an asynchronous motor is used with the following values:
  – rated frequency ≠ 50 Hz (star) or
  – rated frequency ≠ 87 Hz (delta) or
  – number of pole pairs ≠ 2

### 5.6.1 Parameterisation dialog/signal flow

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Select the motor control"11: VFCplusEco: V/f energy-saving" from the *Overview* dialog box in the **Motor control** 11list field:
4. Click the **Motor control V/f Eco** button to change to the *Overview* ➞ *Motor control V/f* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed.

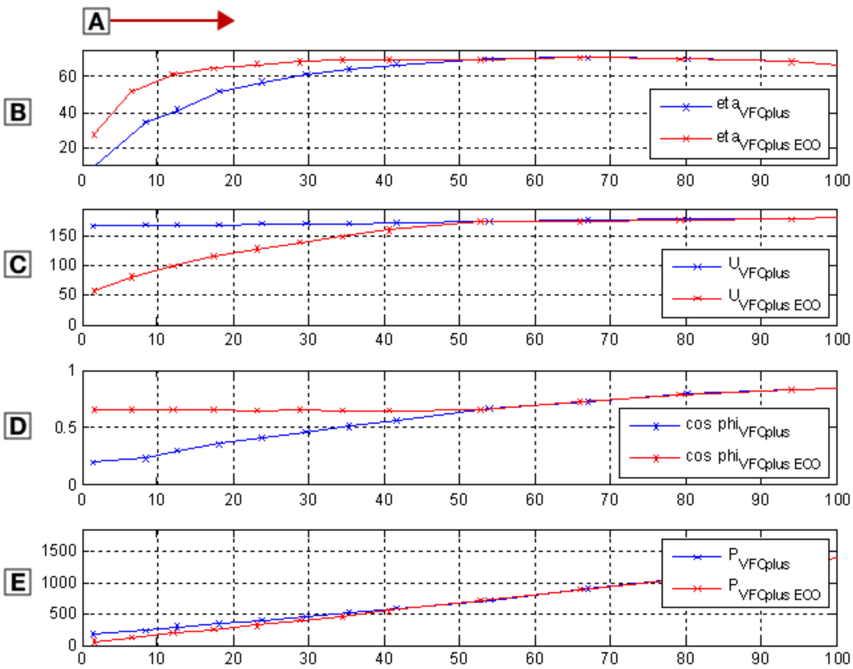



| | Parameter | Info | | Parameter | Info | | Parameter | Info |
|---|---|---|---|---|---|---|---|---|
| 1 | C00056/2 | Actual torque value | 13 | C00910/1 | Max. pos. output frequency | 26 | C00975 | VFC-ECO: Vp |
| 2 | C00830/4 | Limitation of torque in motor mode | 14 | C00910/2 | Max. neg. output frequency | 27 | C00976 | VFC-ECO: Ti |
| 3 | C00830/5 | Limitation of torque in generator mode | 15 | C00021 | Slip compensation | 28 | C00977 | VFC-ECO: Minimum voltage V/f |
| 4 | C00054 | Motor current | 16 | C00051 | Actual speed value | 29 | C00982 | VFC-ECO: Voltage reduction ramp |
| 5 | C00022 | Imax in motor mode | 17 | C00050 | Speed setpoint | 30 | C00978 | VFC-ECO: Voltage reduction |
| 6 | C00023 | Imax in generator mode | 18 | C00056/2 | Actual torque value | 31 | C00018 | Switching frequency |
| 7 | C00074 | Ti Imax controller | 19 | C00058 | Output frequency | 32 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 8 | C00073 | Vp Imax controller | 20 | C00830/26 | MCTRL: nBoost_a | 33 | C00830/31 | MCTRL: nVoltageAdd_a |
| 9 | C00830/3 | Speed setpoint | 21 | C00015 | V/f base frequency | 34 | C00052 | Motor voltage |
| 10 | C00909/1 | Max. pos. speed | 22 | C00016 | Vmin boost | 35 | C00980/1 | Active output power |
| 11 | C00909/2 | Max. neg. speed | 23 | C00979/2 | Cosine phi set | 36 | C00980/2 | Apparent output power |
| 12 | C00105 | Decel. time - quick stop | 24 | C00979/1 | Cosine phi act | 37 | C00981/1 | Output energy in motor mode |
| | | | 25 | C00083 | Motor rotor time constant | 38 | C00981/2 | Output energy in generator mode |

### 5.6.2 Comparison of VFCplusEco - VFCplus

The following characteristics show the impact of the energy-saving V/f characteristic control (VFCplusEco) compared to the standard V/f characteristic control (VFCplus).

▶ The characteristics were recorded with a standard asynchronous motor 2.2 kW with energy efficiency class IE1 at speed = 600 rpm.




A scaled torque in [%]
B efficiency in [%]
C motor voltage in [V]
D power factor
E electrical power in [W]

[5-8] Comparison of VFCplusEco - VFCplus

### 5.6.3 Basic settings

The "Initial commissioning steps" listed in the table below are sufficient for the V/f characteristic control - energy-saving (VFCplusECo).

▶ Detailed information on the individual steps can be found in the following subchapters.

<table>
<tr><th colspan="3">Initial commissioning steps</th></tr>
<tr><td>1.</td><td colspan="2">Determine the motor control:<br>C00006 = "11: VFCplusEco: V/f energy-saving"</td></tr>
<tr><td rowspan="3">2.</td><td colspan="2">The required motor data are pre-initialised depending on the device and thus, they do not need to be entered directly.In order to achieve a high energy optimisation, these motor data can be entered (see the following section).</td></tr>
<tr><td colspan="2">Set the motor selection/motor data<br>• When selecting and parameterising the motor, the motor nameplate data and the equivalent circuit diagram data are relevant. Detailed information can be found in the "Motor selection/Motor data" chapter. (📖 121)<br>Depending on the motor manufacturer, proceed as follows:</td></tr>
<tr><td>Lenze motor:<br>Selecting a motor from the motor catalogue in the »Engineer«<br>- or -<br>1. Set the motor nameplate data<br>2. Automatic motor data identification</td><td>Third party manufacturer's motor:<br>1. Set the motor nameplate data<br>2. Automatic motor data identification or set known equivalent circuit diagram manually:<br>C00084: Motor stator resistance<br>C00085: Motor stator leakage inductance<br>C00092: Motor magnetising inductance</td></tr>
<tr><td>3.</td><td colspan="2">Defining current limits (Imax controller). (📖 152)</td></tr>
</table>

**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in the chapter "Optimising the control mode". (📖 171)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 240)

### 5.6.4 Optimising the control mode

The V/f characteristic control - energy-saving (VFCplus) is generally ready for operation. It can be adapted subsequently by adapting the characteristic and/or the drive behaviour.

#### Adapting characteristic

For the linear characteristic as part of the V/f characteristic control - energy-saving (VFCplusEco), it is also possible (like in case of the standard V/f characteristic control) to match its curve to different load profiles or motors by adapting the V/f base frequency (C00015) and the $V_{min}$ boost (C00016).

**Note!**

For an adaptation of the $V_{min}$ boost, the V/f characteristic control - energy-saving (VFCplusEco) must not be set. For this purpose, set the V/f characteristic control (VFCplus).

Adapting the V/f base frequency (📖 154)

Adapting the Vmin boost (📖 156)

#### Adapting drive behaviour

- Limitation of the maximum current by a current limitation controller (e.g. to prevent the motor from stalling or to limit to the maximally permissible motor current). Optimising the Imax controller (📖 158)
- Adaptation of the field frequency by a load-dependent slip compensation (improved speed accuracy for systems without feedback).
- Improving the behaviour at high dynamic load changes. (📖 172)
- Adapting the slope limitation for lowering the Eco function. (📖 173)
- Optimising the cos/phi controller. (📖 173)

#### Torque limitation

Limit the torque to a lower value. Torque limitation (📖 160)

### 5.6.4.1 Improving the behaviour at high dynamic load changes

Due to the voltage reduction executed via the cosφ control, the motor may stall in the Lenze setting at high dynamic load torque changes. This is caused by the flux reduction and the connected reduction of the stalling torque of the motor current:

$$M_{Max(t)} = M_{Stalling} \cdot \frac{U^2_{Motor(t)}}{(U_{Motor(t)} - U_{Reduction})^2} \qquad \text{with } M_{Stalling} = 1.6 \ldots 2.5 \cdot M_{Rated\_motor}$$

$V_{Motor}$ = display in C00052
$V_{Reduction}$ = display in C00978

It generally applies that when the output voltage is divided in halves, the maximum torque is approx. reduced by the factor 4. A reduction by the factor 3 reduces the torque to approx. 15 %.

The minimum voltage and thus the maximum influence access of the Eco function on the output voltage can be defined in C00977. With full influence of the Eco function, the following stalling torque can be ensured depending on the setting in C00977:

| Minimum voltage V/f (C00977) | Maximum torque |
|---|---|
| 100 % | 160 % ... 250 % $M_{rated}$ |
| 70 % | 80 % ... 130 % $M_{rated}$ |
| 50 % | 40 % ... 70 % $M_{rated}$ |
| 20 % | 15 % ... 50 % $M_{rated}$ |

An adaptation of the minimum voltage V/f (C00977) improves the stability in case of load impulses.

- In the Lenze setting, the minimum voltage V/f is set to 20 % for the highest energy optimisation. This setting serves to respond to load torques if these amount to approx. 25 % of the rated torque or occur with low dynamics.
- An increase of the minimum voltage V/f to 70 % permits to apply a dynamic load impulse from 0 to 100 % rated motor torque without the motor stalling. This reduces the energy optimisation to be achieved by approx. 75 %.
- A further increase of the stability at still higher dynamic load impulses can be achieved by a further increase of the minimum voltage V/f, but means a further loss in energy optimisation.

**Note!**

The energy optimisation can be switched off by setting the minimum voltage V/f (C00977) to 100 %. Then, the behaviour corresponds to the V/f characteristic control (VFCplus) with linear characteristic.

In case of applications with very high dynamic sudden load variations from the unloaded operation, this motor control mode should not be used or the energy optimisation should be switched off, since a motor stalling cannot be excluded.

### 5.6.4.2 Adapting the slope limitation for lowering the Eco function

The ramp set in C00982 for voltage reduction serves as slope limitation in order to prevent that voltage is suddenly applied to the motor when the Eco function is deactivated. Otherwise, the overvoltage limitation (Imax, Clamp) would be activated.

- This ramp is, depending on the device, pre-initialised to approx. the triple rotor time constant. An adaptation of this parameter is not required.

When the Eco function is switched off, a quick reaction (high dynamic performance) is required, but with a low current overshoot and a small torque jump. Thus, the Lenze setting of C00982 is a compromise regarding the switch-off of the Eco function (voltage reduction = 0).

- To increase the dynamics when switching off the Eco function:
  Reduce → setting in C00982.
  (Current compensation actions increase when the Eco function is switched off.)
- In order to reduce current compensation actions when switching off the Eco function:
  Increase → setting in C00982.
  (The dynamics when switching off the eco function is reduced)

### 5.6.4.3 Optimising the cos/phi controller

With the Lenze setting, the cosφ controller is set such that usually no adaptation is required for all power ratings and application cases.

| Behaviour | Remedy/recommendation |
|---|---|
| The cosφ actual value (C00979/1) varies greatly. | Reduce gain Vp (C00975) and reset time Ti (C00976). |
| The cosφ actual value (C00979/1) is permanently lower than the cosφ setpoint (C00979/2). | Increase gain Vp (C00975) and reset time Ti (C00976). |

### 5.6.5 Remedies for undesired drive behaviour

| Drive behaviour | Remedy |
|---|---|
| Inadequate smooth running at low speeds, especially in the case of operation with a long motor cable | ▸ Automatic motor data identification (🕮 126)<br>Reduce the influence of the Eco function by increasing the minimum voltage V/f (C00977). |
| Problems in case of high starting duty (great mass inertia) | 1. Set motor control VFCplus with linear characteristic (C00006 = 6).<br>2. Adapting the Vmin boost. (🕮 156)<br>3. Again set motor control VFCplusEco (C00006 = 11). |
| Drive does not follow the speed setpoint | The current controller intervenes in the set field frequency to limit the controller output current to the maximum current (C0022, C0023). Therefore:<br>• Prolong acceleration/deceleration times:<br>C00012: Accel. time - main setpoint<br>C00013: Decel. time - main setpoint<br>• Consider a sufficient magnetising time of the motor. Depending on the motor power, the magnetising time amounts to 0.1 ... 0.2 s.<br>• Increase the maximally permissible current:<br>C00022: Imax in motor mode<br>C00023: Imax in generator mode<br>• Make adaptations for the Eco function:<br>– Improving the behaviour at high dynamic load changes. (🕮 172)<br>– Adapting the slope limitation for lowering the Eco function. (🕮 173)<br>– Optimising the cos/phi controller. (🕮 173) |
| Insufficient speed constancy at high load (setpoint and motor speed are not proportional anymore) | • Increase slip compensation (C00021).<br>Important: Unstable drive due to overcompensation!<br>• With cyclic load impulses (e. g. centrifugal pump), a smooth motor characteristic is achieved by smaller values in C00021 (possibly negative values).<br>Note: The slip compensation is only active for operation without speed feedback. |
| "Clamp operation active" error message (OC11):<br>Controller cannot follow dynamic processes, i.e. too short acceleration/deceleration times in terms of load ratios. | • Increase the gain of the $I_{max}$ controller (C00073)<br>• Reduce the reset time of the $I_{max}$ controller (C00074)<br>• Prolong the acceleration time (C00012)<br>• Prolong the deceleration time (C00013)<br>• Make adaptations for the Eco function:<br>– Improving the behaviour at high dynamic load changes. (🕮 172)<br>– Adapting the slope limitation for lowering the Eco function. (🕮 173) |
| Motor stalling in the field weakening range (adaptation especially required for small machines) | • If motor power < inverter power:<br>Set C00022 to $I_{max}$ = 2 $I_{rated\ motor}$<br>• Reduce dynamic performance of setpoint generation<br>• Make adaptations for the Eco function:<br>– Improving the behaviour at high dynamic load changes. (🕮 172)<br>– Adapting the slope limitation for lowering the Eco function. (🕮 173) |
| Speed variations in no-load operation for speeds > 1/3 rated speed. | Minimise speed oscillations with oscillation damping (C00234). |

| Drive behaviour | Remedy |
| --- | --- |
| Speed variations in no-load operation and with load for speeds > rated speed. | Minimise speed oscillations with increasing the oscillation damping field weakening (C00236).<br>Caution: If C00236 is increased, the maximum output voltage of the device is reduced! |
| Output voltage is too low. There is a too low maximum torque in the high field weakening range. | Reduction of the oscillation damping field weakening (C00236).<br>Caution: When C00236 = 0, oscillation damping field weakening is inactive. Thus, a maximum output voltage is available but the tendency to speed oscillations in the field weakening range at no-load operation and with load increases. |

## 5.7 V/f control (VFCplus + encoder)

The previously described V/f characteristic control (VFCplus) can be operated with a feedback of speed. This bears the following advantages:

- Stationary accuracy of speed
- Low parameterisation effort compared to sensorless vector control (SLVC)
- Improved dynamics compared to V/f characteristic control without feedback or to sensorless vector control (SLVC).
- Suitability for group drives

The descriptions in chapter "V/f characteristic control (VFCplus)" also apply for the V/f control. (📖 147)

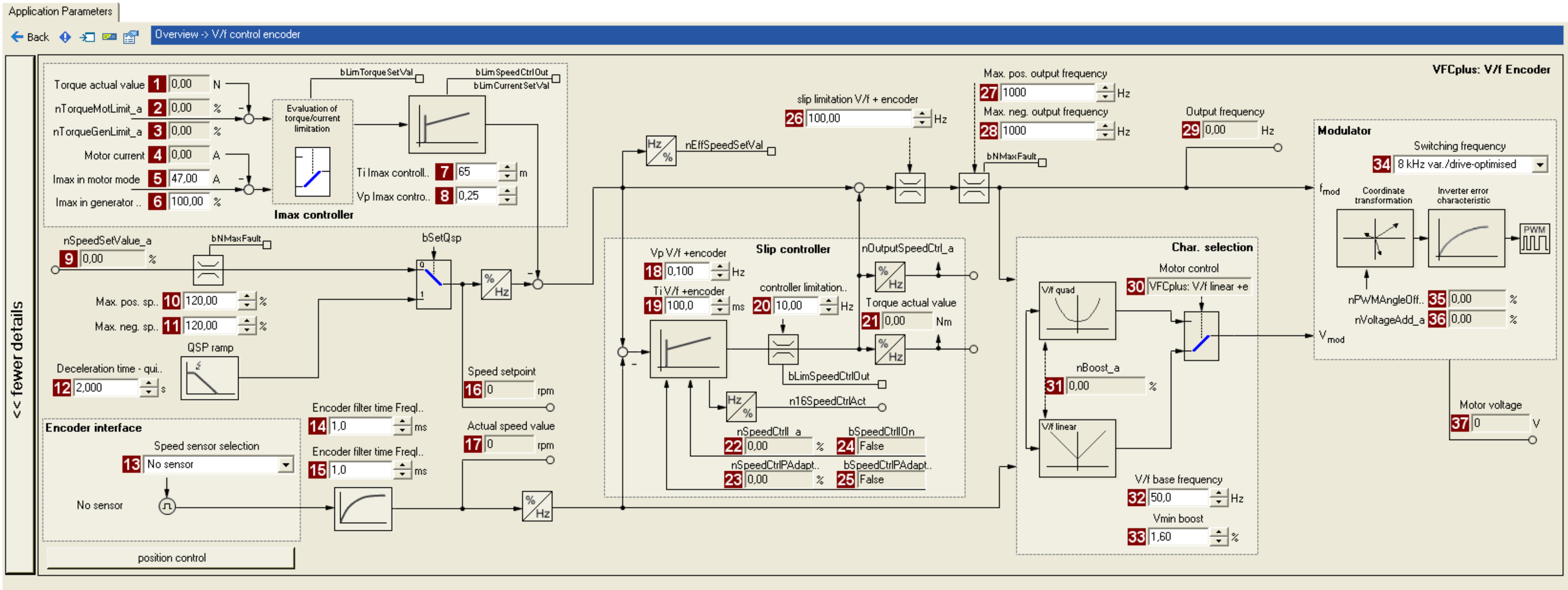
**Note!**

As the slip is calculated in the feedback V/f operation and injected through the slip regulator, the slip compensation (C00021) is deactivated with V/f control.

### 5.7.1 Parameterisation dialog/signal flow

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Select the motor control from the *Overview* dialog level in the **Motor control** (C00006) list field:
   - "7: VFCplus: V/f linear +encoder" for linear characteristic or
   - "9: VFCplus: V/f quadr +encoder" for square-law characteristic
4. Click the **Motor control V/f encoder** button to change to the *Overview* → *Motor control V/f* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed, as shown in the following subchapter.




| | Parameter | Info |
|---|---|---|
| 1 | C00056/2 | Actual torque value |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00054 | Motor current |
| 5 | C00022 | Imax in motor mode |
| 6 | C00023 | Imax in generator mode |
| 7 | C00074 | Ti Imax controller |
| 8 | C00073 | Vp Imax controller |
| 9 | C00830/22 | Speed setpoint |
| 10 | C00909/1 | Max. pos. speed |
| 11 | C00909/2 | Max. neg. speed |
| 12 | C00105 | Decel. time - quick stop |
| 13 | C00495 | Speed sensor selection |
| 14 | C00497/2 | Encoder filter time FreqIn67 |
| 15 | C00497/1 | Encoder filter time FreqIn12 |
| 16 | C00050 | Speed setpoint |
| 17 | C00051 | Actual speed value |
| 18 | C00972 | Vp Vf+encoder |
| 19 | C00973 | Ti Vf+encoder |
| 20 | C00971/1 | Controller limitation Vf+encoder |
| 21 | C00056/2 | Actual torque value |
| 22 | C00830/24 | MCTRL: nSpeedCtrll_a |
| 23 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 24 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 25 | C00833/69 | MCTRL: bSpeedCtrlPAdaptOn |
| 26 | C00971/2 | Slip limitation Vf+encoder |
| 27 | C00910/1 | Max. pos. output frequency |
| 28 | C00910/2 | Max. neg. output frequency |
| 29 | C00058 | Output frequency |
| 30 | C00006 | Motor control |
| 31 | C00830/26 | MCTRL: nBoost_a |
| 32 | C00015 | V/f base frequency |
| 33 | C00016 | Vmin boost |
| 34 | C00018 | Switching frequency |
| 35 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 36 | C00830/31 | MCTRL: nVoltageAdd_a |
| 37 | C00052 | Motor voltage |

### 5.7.2 Basic settings

In order to protect the drive system, carry out the commissioning of the V/f control and the slip regulator in several steps.

▶ Detailed information on the single steps can be found in the following subchapters or in the corresponding subchapters for V/f characteristic control.

| Initial commissioning steps | |
|---|---|
| 1. | Define the V/f characteristic:<br>• C00006 = 7: Linear characteristic<br>• C00006 = 9: Square-law characteristic |
| 2. | Defining current limits (Imax controller). (📖 152) |
| 3. | Parameterise the encoder/feedback system.<br>▸ Encoder/feedback system (📖 293) |
| 4. | If special motors with a rated frequency other than 50 Hz or with a number of pole pairs ≠ 2 are used, set the motor parameters according to the motor nameplate.<br>▸ Motor selection/Motor data (📖 121) |
| 5. | Define speed setpoint (e.g. 20 % of the rated speed) and enable controller. |
| 6. | Check whether the actual speed value (C00051) ≈ speed setpoint (C00050) and then inhibit the controller again.<br>• In case of a sign reversal between actual value and setpoint, check the connection or setting of the encoder.<br>• In case the actual value differs considerably from the setpoint (factor 2), set the motor parameters according to motor nameplate. Then repeat step 5. |
| 7. | For protecting the drive, reduce slip regulator limitation in C00971/1.<br>• e.g. reduction to half the slip frequency (≈ 2 Hz) |
| 8. | Define speed setpoint (e.g. 20 % of the rated speed) and enable controller. |
| 9. | In case of a semi-stable operational performance, reduce the reset time (C00972) or the proportional gain (C00973) of the slip regulator until a stable operation has been achieved.<br>▸ Parameterising the slip regulator (📖 179) |
| 10. | In a final step, increase the slip regulator limitation again in C00971/1.<br>• e.g. increase to twice the slip frequency |

**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in the "Optimising the control mode" chapter for the V/f characteristic control (VFCplus). (📖 153)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 240)

### 5.7.2.1 Parameterising the slip regulator

The slip regulator is designed as a PI controller. In order to improve the response to setpoint changes, the setpoint speed or setpoint frequency is added to the output (correcting variable) of the slip regulator as feedforward control value.

- In contrast to the conventional speed controller, the slip regulator only regulates the slip.
- In the Lenze setting, the slip regulator features a configuration with a goof robustness and moderate dynamics.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00971/1 | VFC: Controller limitation V/f +encoder | 10.00 | Hz |
| C00971/2 | VFC: Slip limitation V/f +encoder | 100.00 | Hz |
| C00972 | VFC: Vp V/f +encoder | 0.100 | Hz/Hz |
| C00973 | VFC: Ti V/f +encoder | 100.0 | ms |

#### Slip regulator gain Vp

The setting range of the slip regulator gain Vp (C00972) which leads to a stable operational performance, mainly depends on the resolution of the speed sensor. There is a direct relationship between encoder resolution and gain:

- The higher the encoder resolution, the higher the gain can be set.

The following table provides maximum and recommended slip regulator gains for encoder with standard encoder increments:

| Encoder increment [Increments/revolution] | Slip regulator gain Vp | |
|---|---|---|
| | maximum | recommended |
| 8 | 0,09 | 0,06 |
| 64 | 0,52 | 0,31 |
| 100 | 0,79 | 0,47 |
| 120 | 0,94 | 0,57 |
| 128 | 1,00 | 0,60 |
| 256 | 1,29 | 0,77 |
| 386 | 1,63 | 0,98 |
| 512 | 1,97 | 1,18 |
| 640 | 2,31 | 1,38 |
| 768 | 2,65 | 1,59 |
| 896 | 2,99 | 1,79 |
| 1014 | 3,33 | 2,00 |
| 1536 | 4,69 | 2,81 |
| 2048 | 6,05 | 3,63 |
| 3072 | 8,77 | 5,26 |
| 4096 | 11,49 | 6,90 |

[5-1] Slip regulator gain Vp based on the encoder increment

**How to adapt the slip regulator gain to the operating conditions:**

1. Adapt the slip regulator gain (C00972) to the encoder increment according to table [5-1].
2. Set controller limitation (C00971/1) to half the slip frequency (≈ 2 Hz).
3. Select speed setpoint (e.g. 20 % of the rated speed).
4. Enable controller.
5. Increase slip regulator gain (C00972) until the drive becomes semi-stable.
   - This can be recognised by motor noises or "humming" of the motor or by a noise on the actual speed signal.
6. Reduce slip regulator gain (C00972) until the drive runs stable again (no motor "humming").
7. Reduce slip regulator gain (C00972) to approx. half the value.
   - For lower encoder resolutions, another reduction of the slip regulator gain for lower speeds (speed setpoint ≈ 0) may become necessary.
   - It is recommended to check as a final step the behaviour at setpoint speed = 0 and further reduce the slip regulator gain in case of irregular running.
8. Increase controller limitation (C00971/1) again (e.g. to twice the slip frequency).

### Slip regulator time constant Ti

**How to set the slip regulator time constant:**

1. Set controller limitation (C00971/1) to half the slip frequency (≈ 2 Hz).
2. Select speed setpoint (e.g. 20 % of the rated speed).
3. Enable controller.
4. Reduce slip regulator time constant (C00973) until the drive becomes semi-stable.
   - This can be recognised by engine noises or motor "oscillating" or by oscillation on the actual speed signal.
5. Increase slip regulator time constant (C00973) until the drive runs stable again (no motor "oscillation").
6. Increase the slip regulator time constant (C00973) to approx. twice the value.
7. Increase controller limitation (C00971/1) again (e.g. to twice the slip frequency).

### Controller limitation

Max. intervention of the controller is limited by the controller limitation (C00971/1).

- The controller can be limited depending on the application.
- We recommend to limit the max. intervention to twice the rated slip of the motor.
- The rated slip is calculated as follows:

$$f_{Slip_{Rated}}[Hz] = f_{Rated}[Hz] - \left(\frac{n_{Motor_{Rated}}[rpm]}{60} \cdot p_{Number\ of\ pole\ pairs}\right)$$

[5-9] Calculation of the rated slip

**Note!**

The setting C00971/1 = 0 Hz deactivates the slip regulator. In this case, the structure of the V/f control corresponds to the structure of the V/f characteristic control without feedback.

### Slip limitation

In addition to limiting the slip regulator, the field frequency to be injected can also be limited by another limiting element, the slip limitation (C00971/2).

- A slip limitation to, for instance, double the rated slip of the motor prevents the motor from stalling in very dynamic processes.
- Motor stalling is caused by:
  - High overcurrent at very steep speed ramps
  - very fast speed changes due to load, e.g. abrupt stopping of the drive due to an encounter with a stop or a load that is not moving.

## 5.8 Sensorless vector control (SLVC)

Sensorless vector control (SLVC) is based on a better motor current control according to a field-oriented control mode by Lenze.

### Stop!

- The sensorless vector control (SLVC) is only suitable for asynchronous motors.
- The connected motor must not be more than two power classes smaller than the motor assigned to the controller.
- Operation of the sensorless vector control (SLVC) is only permissible for one single drive!
- Operation of the sensorless vector control (SLVC) is not permissible for hoists!
- The Lenze setting permits the operation of a power-adapted motor. Optimal operation is only possible if either:
  - the motor is selected via the Lenze motor catalogue
  - the motor nameplate data are entered and motor parameter identification is carried out afterwards
    *- or -*
  - the nameplate data and equivalent circuit data of the motor (motor leakage inductance and mutual motor inductance, slip compensation and motor stator resistance) are entered manually.
- When you enter the motor nameplate data, take into account the phase connection implemented for the motor (star or delta connection). Only enter the data applying to the selected connection type.
  - In this context, also observe the instructions in the chapter entitled "Adapting the V/f base frequency" relating to V/f characteristic control. (📖 154)

### Note!

Optimal operation of the sensorless vector control (SLVC) can be achieved from a minimum speed of approx. 0.5-fold slip speed. At lower speed values below the 0.5-fold slip speed, the maximum torque is reduced.

The maximum field frequency with this motor control mode is 650 Hz.

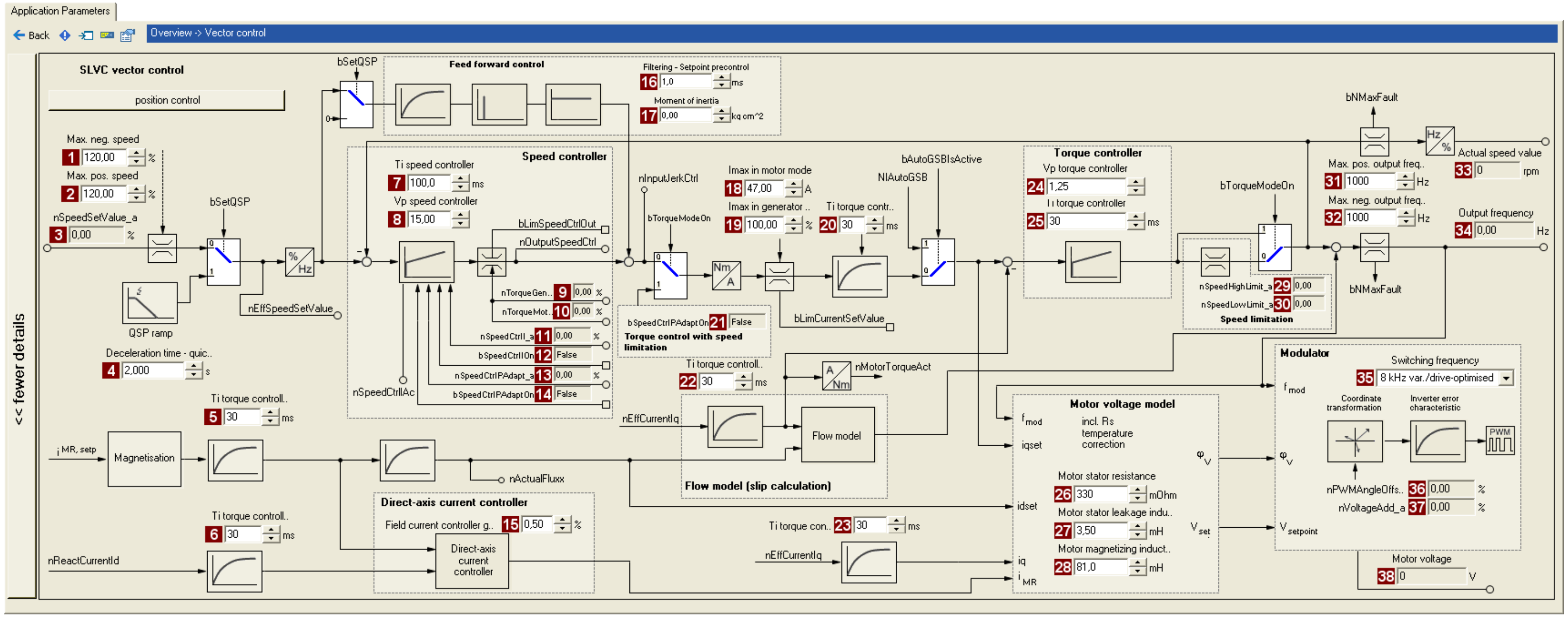
In comparison to the V/f characteristic control without feedback, the following can be achieved by means of sensorless vector control SLVC:

- A higher maximum torque throughout the entire speed range
- A higher speed accuracy
- A higher concentricity factor
- A higher level of efficiency
- The implementation of torque-actuated operation with speed limitation
- The limitation of the maximum torque in motor and generator mode for speed-actuated operation

### 5.8.1 Parameterisation dialog/signal flow

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Select the motor control "4: SLVC: Vector control" from the *Overview* dialog level in the **Motor control** list field (C00006):
4. Click the **Motor control vector** button to change to the *Overview* → *Motor control vector* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed, as shown in the following subchapter.




| Parameter | | Info |
|---|---|---|
| 1 | C00909/2 | Max. neg. speed |
| 2 | C00909/1 | Max. pos. speed |
| 3 | C00830/22 | Speed setpoint |
| 4 | C00105 | Decel. time - quick stop |
| 5 | C00074/2 | SLVC: Ti torque controller |
| 6 | C00074/2 | SLVC: Ti torque controller |
| 7 | C00071/1 | SLVC: Ti speed controller |
| 8 | C00070/1 | SLVC: Vp speed controller |
| 9 | C00830/28 | Limitation of torque in generator mode |
| 10 | C00830/29 | Limitation of torque in motor mode |
| 11 | C00830/24 | MCTRL: nSpeedCtrlI_a |
| 12 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 13 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 14 | C00833/69 | MCTRL: bSpeedCtrlPAdaptOn |
| 15 | C00985 | SLVC: Field current controller gain |

| Parameter | | Info |
|---|---|---|
| 16 | C00275 | Setpoint feedforward control filtering |
| 17 | C00273 | Moment of inertia |
| 18 | C00022 | Imax in motor mode |
| 19 | C00023 | Imax in generator mode |
| 20 | C00074/2 | SLVC: Ti torque controller |
| 21 | C00833/69 | MCTRL: bSpeedCtrlPAdaptOn |
| 22 | C00074/2 | SLVC: Ti torque controller |
| 23 | C00074/2 | SLVC: Ti torque controller |

| Parameter | | Info |
|---|---|---|
| 24 | C00073/2 | SLVC: Vp torque controller |
| 25 | C00074/2 | SLVC: Ti torque controller |
| 26 | C00084 | Motor stator resistance |
| 27 | C00085 | Motor stator leakage inductance |
| 28 | C00092 | Motor magnetising inductance |
| 29 | C00830/88 | MCTRL: nSpeedHighLimit_a |
| 30 | C00830/23 | MCTRL: nSpeedLowLimit_a |
| 31 | C00910/1 | Max. pos. output frequency |
| 32 | C00910/2 | Max. neg. output frequency |
| 33 | C00051 | Actual speed value |
| 34 | C00058 | Output frequency |
| 35 | C00018 | Switching frequency |
| 36 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 37 | C00830/31 | MCTRL: nVoltageAdd_a |
| 38 | C00052 | Motor voltage |

### 5.8.2 Types of control

The sensorless vector control can be operated in two different modes:

- Speed control with torque limitation (*bTorquemodeOn* = FALSE)
- Torque control with speed limitation (*bTorquemodeOn* = TRUE)

#### 5.8.2.1 Speed control with torque limitation

A speed setpoint is selected and the drive system is operated in a speed-controlled manner.

The operational performance can be adapted in the following ways:

- Overload limitation in the drive train
  - The torque is limited via the torque setpoint.
  - The torque setpoint is identical to the value at the output of the speed controller, *nOutputSpeedCtrl*.
  - To avoid overload in the drive train, the torque in motor mode can be limited via the *nTorqueMotLimit_a* process input signal, and the torque in generator mode can be limited via the *nTorqueGenLimit_a* process input signal:

| **Identifier**<br>DIS code \| data type | **Information/possible settings** |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

**Note!**

To avoid instabilities during operation, the torque limit values are internally processed as absolute values.

- Motor current limitation
  - A cross current setpoint is calculated from the torque setpoint which is limited depending on the magnetising current, the max. current in motor mode (C00022), and the max. current in generator mode (C00023).
  - Here, the total current injected into the motor does not exceed the max. currents in motor and generator mode.
- Slip compensation (📖 255)
  - Using a slip model, the slip of the machine is reconstructed.
  - The slip compensation (C00021) acts as the influencing parameter.

### 5.8.2.2 Torque control with speed limitation

For torque-controlled operation, a torque setpoint is defined in the drive system. Unlike the Speed control with torque limitation, this type of control has a deactivated speed controller and torque limitation.

- The torque setpoint is calculated directly from *nTorqueSetValue_a*.
- The speed is defined by the process.
- Due to its limitation, the speed-controlled drive can only rotate within a speed range whose positive speed is limited by *nSpeedHighLimit_a* and whose negative speed is limited by *nSpeedLowLimit_a*.

| **Identifier**<br>DIS code \| data type | **Information/possible settings** |
|---|---|
| nTorqueSetValue_a<br>C00830/27 \| INT | Torque setpoint / additive torque<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| nSpeedHighLimit_a<br>C00830/88 \| INT | Upper speed limit for speed limitation<br>(only for torque-controlled operation)<br>• Scaling: 16384 ≡ 100 % rated speed (C00011) |
| nSpeedLowLimit_a<br>C00830/23 \| INT | Lower speed limit for speed limitation<br>(only for torque-controlled operation)<br>• Scaling: 16384 ≡ 100 % rated speed (C00011) |

### 5.8.3 Basic settings

The following "Initial commissioning steps" must be performed to commission the sensorless vector control:

| Initial commissioning steps | | |
|---|---|---|
| 1. | Determine the motor control:<br>C00006 = "4: SLVC: Vector control" | |
| 2. | Set the motor selection/motor data<br>• When selecting and parameterising the motor, the motor nameplate data and the equivalent circuit diagram data are relevant. Detailed information can be found in the "Motor selection/Motor data" chapter. (📖 121)<br>Depending on the motor manufacturer, proceed as follows: | |
| | **Lenze motor:**<br>Selecting a motor from the motor catalogue in the »Engineer«<br>- or -<br>1. Set the motor nameplate data<br>2. Automatic motor data identification | **Third party manufacturer's motor:**<br>1. Set the motor nameplate data<br>2. Automatic motor data identification or set known equivalent circuit diagram data manually:<br>C00082: Motor rotor resistance<br>C00084: Motor stator resistance<br>C00085: Motor stator leakage inductance<br>C00092: Motor magnetising inductance<br>C00095: Motor magnetising current |
| 3. | Define the type of control:<br>*bTorquemodeOn* = FALSE: Speed control with torque limitation<br>*bTorquemodeOn* = TRUE: Torque control with speed limitation | |
| 4. | Set the slip compensation (C00021). ▸ Slip compensation (📖 255) | |

**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in the chapter "Optimising the control mode". (📖 188)

We recommend to use the flying restart function for connecting/synchronising the inverter to an already rotating drive system. ▸ Flying restart function (📖 247)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 240)

## 5.8.4 Optimising the control mode

### 5.8.4.1 Optimising the starting performance after a controller enable

After the controller has been enabled, the starting action of the motor is delayed due to the magnetisation of the motor. Under consideration of the motor rotor time constant (C00083), the time delay is calculated as follows:

Magnetisation = 1.5 * motor rotor time constant

If this delay cannot be tolerated for specific applications, the motor must always be operated in an energised condition. For this, select one of the following options:

**Procedure without setting a controller inhibit**

1. Deactivate the auto DCB function with C00019 = 0.
2. Do not activate the controller inhibit. Instead, stop the drive by selecting a setpoint of 0 or by activating the quick stop function.

**Procedure with setting a controller inhibit due to application requirements**

1. Deactivate the auto DCB function with C00019 = 0.
2. Enter a greater value for the motor rotor resistance (max. factor 2!) to reduce the magnetisation time in C00082.

**Note!**

During the starting action, a jerk may occur in the machine due to the temporarily increased motor current!

### 5.8.4.2 Optimise speed controller

The speed controller is designed as a PI controller.

- In the Lenze setting, the configuration of the speed controller provides robustness and moderate dynamics.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00070/1 | SLVC: Vp speed controller | 15.00 | |
| C00071/1 | SLVC: Ti speed controller | 100.0 | ms |

#### Speed controller gain Vp

The gain Vp (C00070/1) of the speed controller is defined in a scaled representation which enables a comparable parameterisation almost independent of the power of the motor or inverter. Here, the speed input difference of the controller is scaled to the rated motor speed whereas the output torque refers to the rated motor torque. A gain of 10 means that a speed difference of 1 % is gained through the P component with 10 % torque.

If the rated data of the motor and the mass inertia of the drive system are known, we recommend the following setting:

$$V_P \approx 1.5 \ldots 3 \cdot \frac{T_M[s]}{0.01[s]}$$

$$T_M[s] = \frac{2 \cdot \pi \cdot n_N[rpm]}{M_N[Nm] \cdot 60} \cdot J_{Drive,\,total}[kgm^2]$$

$$M_N[Nm] = \frac{P_N[W] \cdot 60}{2 \cdot \pi \cdot n_N[rpm]}$$

$V_P$ = Gain of the speed controller (C00070/1)
$T_M$ = Time constant for the acceleration of the motor
$M_N$ = Rated motor torque
$n_N$ = Rated motor speed
$J_{drive,\,total}$ = Total moment of inertia of the drive

[5-10] Recommendation for the setting of the gain of the speed controller

**Tip!**

Values recommended by Lenze for the setting of the (proportional) gain:

- For drive systems without feedback: Vp = 6 ... 25
- For drive systems with a good disturbance behaviour: Vp > 15
  In this case, we recommend the optimisation of the dynamic performance of the torque controller.

**Speed controller reset time Ti**

Apart from setting the P component, C00071/1 provides the possibility to take influence on the I component of the PI controller.

**Tip!**

Value range recommended by Lenze for the setting of the reset time:

Ti = 20 ms ... 150 ms

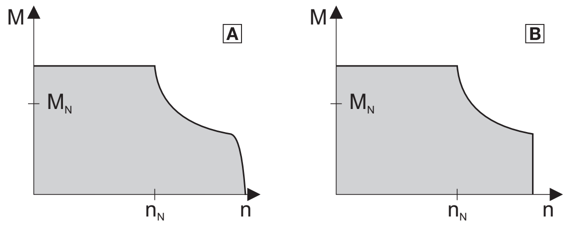
### 5.8.4.3 Optimising dynamic performance and field weakening behaviour

In the Lenze setting, the torque controller has been preset in such a way that robust and stable operation with a moderate dynamic response is enabled over the entire speed range. Retrospective optimisation of the controller parameters is not necessary.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00073/2 | SLVC: Vp torque controller | 1.25 | |
| C00074/2 | SLVC: Ti torque controller | 30 | ms |

A greater dynamic performance of the sensorless vector control can be achieved by reducing time constant Ti of the speed controller (C00074/2).

A greater dynamic performance of the field weakening function can be achieved by setting a time constant ≤ 15 ms. This means for actual speeds above rated speed a better torque-speed-characteristic in the field weakening range:




A Torque controller time constant Ti > 15 ms
B Torque controller time constant Ti ≤ 15 ms

[5-11] Speed / torque characteristic diagram in the field weakening range

- For Ti > 15 ms (see A), the actual speed value slightly drops in the field weakening range if the load torque increases in the motor mode.
- For Ti ≤ 15 ms (see B), the speed remains stable in the field weakening range if the torque is within the M/n characteristic field highlighted in grey.

**Tip!**

For applications with high dynamic performance and speed/torque accuracy requirements in the field weakening range, we recommend a time constant Ti ≤ 15 ms.

In this case, the maximum torque should be limited via the *nTorqueMotLimit_a* and *nTorqueGenLimit_a* process input signals to 1.5 x $M_N$ to ensure stable operation in the field weakening range.

#### 5.8.4.4 Optimising the stalling behaviour

Motor stalling due to a torque overload in the field weakening range in sensorless vector control is prevented by means of an inverter-internal stalling current monitoring. In the field weakening range, hence at frequencies above the base frequency, it reduces the maximum current to prevent the motor from stalling. The reduction depends on the current field frequency, the base frequency, the DC-bus voltage and the maximum current (C00022). Generally it applies that a higher field frequency causes a stronger limitation of the maximum current.

The field weakening behaviour of the sensorless vector control depends on the setting of the reset time Ti of the torque controller (C00074/2).

**The following applies to the reset time Ti (C00074/2) > 15 ms:**

The behaviour in the field weakening range can be adapted via the override point of field weakening (C00080). This parameter serves to shift the frequency-dependent maximum current characteristic:

- C00080 > 0 Hz:
  - The maximum current characteristic is shifted by the entered frequency to higher field frequencies.
  - The maximally permissible current and the maximum torque increase in the field weakening range.
  - The risk of motor stalling increases.
- C00080 < 0 Hz:
  - The maximum current characteristic is shifted by the entered frequency to lower field frequencies.
  - The maximally permissible current and the maximum torque are reduced in the field weakening range.
  - The risk of motor stalling is reduced.

**Note!**

We recommend to keep the Lenze setting (0 Hz).

**The following applies to the reset time Ti (C00074/2) <= 15 ms:**

The reduction of the magnetising current in the field weakening range can be adapted via the override point of field weakening (C00080):

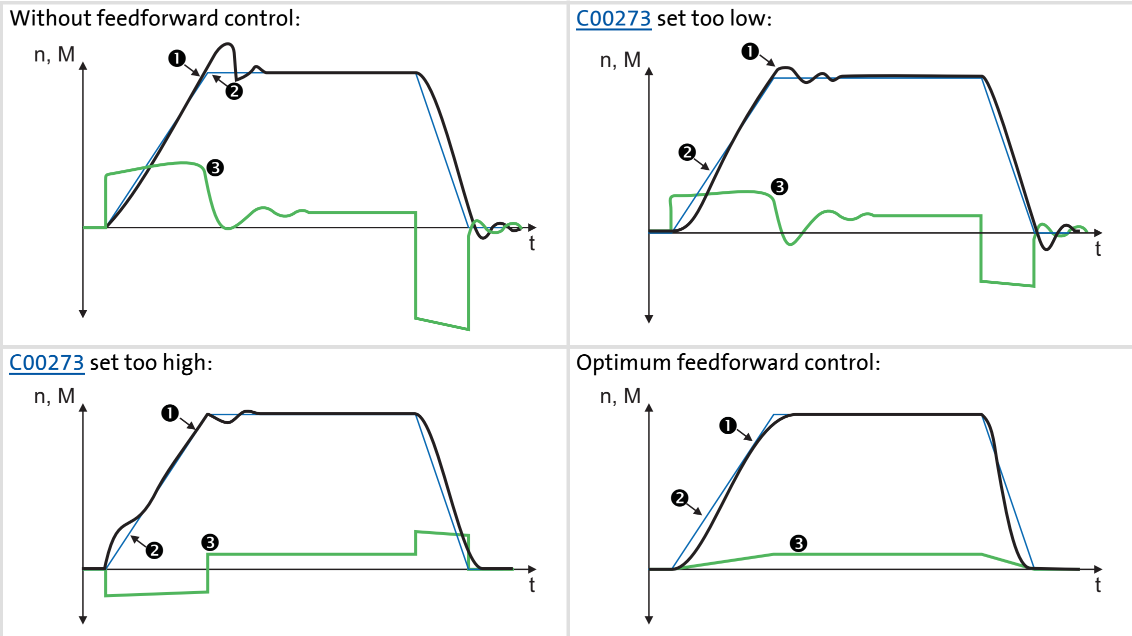
- C00080 > 0 Hz:
  The reduction of the magnetising current is shifted to higher field frequencies. Here, there is a risk of the motor being magnetised too much and having too little voltage reserve for the torque-creating current.
- C00080 < 0 Hz:
  The reduction of the magnetising current is shifted to lower field frequencies.

**Note!**

A function for enabling a stable operation can only be implemented to a limited extent with a reset time Ti <= 15 ms For applications with speeds above the 2-fold rated speed, we recommend a reset time Ti (C00074/2) > 15 ms.

### 5.8.4.5 Optimise response to setpoint changes and determine mass inertia

Setting the total moment of inertia under C00273 provides the optimum torque feedforward control. Depending on the application, an adjustment of the setting under C00273 may be necessary to optimise the response to position/speed setpoint changes by means of the torque feedforward control.




❶ Actual speed value (*nMotorSpeedAct_a* signal)
❷ Speed setpoint (*nSpeedSetValue_a* signal)
❸ Speed controller output (*nOutputSpeedCtrl_a* signal)

[5-12] Typical signal characteristics for different settings of the load moment of inertia

**How to optimise the torque feedforward control:**

1. Run a typical speed profile and record the inputs and outputs of the speed controller with the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
     *nOutputSpeedCtrl_a* (speed controller output)
2. Estimate the moment of inertia and set it in C00273 in relation to the motor end (i.e. with account being taken of the gearbox factors).
3. Repeat the data logger recording (see step 1).

   Now the data logger should show that part of the required torque is generated by the feedforward control and the speed controller output signal (*nOutputSpeedCtrl_a*) is correspondingly smaller. The resulting following error decreases.
4. Change the setting in C00273 and repeat the data logger recording until the intended response to setpoint changes is reached.
   - The optimisation could aim at the speed controller being completely relieved (see signal characteristics in Fig. [5-12]).
5. Save the parameter set (device command: C00002/11).

### 5.8.4.6 Slip calculation from motor equivalent circuit diagram data

This function extension is available from version 02.00.00!

In order to achieve a better speed stability and torque accuracy, the slip calculation can be either derived from the motor nameplate data (e.g. rated motor speed) or the motor equivalent circuit diagram data (stator resistance, rotor resistance etc.).

The data to be used for sensorless vector control is selected via bit 0 in C02879/1:

| Setting | | Info |
|---|---|---|
| Bit 0 | SLVC | In case of sensorless vector control:<br>• "0" ≡ Slip calculation from motor nameplate data (Lenze setting)<br>• "1" ≡ Slip calculation from motor equivalent circuit diagram data |
| Bit 1 | SC_ASM | In case of servo control for asynchronous motors:<br>• "0" ≡ Slip calculation from motor nameplate data<br>• "1" ≡ Slip calculation from motor equivalent circuit diagram data (Lenze setting) |
| Bit 2 ... 7 | Reserved | |

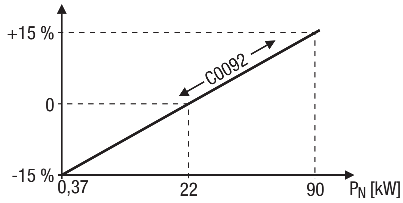
**Note!**

In order that the slip can be calculated from the motor equivalent circuit diagram data, the equivalent circuit data (stator resistance, rotor resistance etc.) must be known as exactly as possible.

- Selecting a motor in the »Engineer« motor catalogue loads the exact motor equivalent circuit diagram data.
- When the motor nameplate data is entered manually and the motor equivalent circuit diagram data is then detected via the motor parameter identification, the "extended identification" (C02867/1 = 2) must be used.
  ▶ Automatic motor data identification (📖 126)

In the slip calculation from the motor equivalent circuit diagram data, the slip compensation (C00021) has no influence anymore.

### 5.8.5 Remedies for undesired drive behaviour

<table>
<tr><th>Drive behaviour</th><th>Remedy</th></tr>
<tr><td>Deviation between no-load current and magnetising current or bad speed or torque accuracy.</td><td>Adapt the motor magnetising inductance (C00092) for no-load operation.<ul><li>If the no-load current is greater than the magnetising current (C00095) at 0.5-fold rated motor speed, the magnetising inductance must be reduced until the no-load current and the magnetising current have the same values.</li><li>Otherwise, the magnetising inductance must be increased.</li></ul>Tendency of the correction of C00092:<br>
<br>PN: Rated motor power</td></tr>
<tr><td>Insufficient speed constancy at high load: Setpoint and motor speed are not proportional anymore.<br>Caution: Overcompensation of the settings mentioned under " Remedy" may result in unstable behaviour!</td><td>Via the slip compensation (C00021), the speed stability under high loads can be affected:<ul><li>If $n_{act} > n_{slip}$, reduce the value in C00021</li><li>If $n_{act} < n_{slip}$, increase the value in C00021</li></ul></td></tr>
<tr><td>Unstable control with higher speeds.</td><td><ul><li>Check the setting of the magnetising inductance (C00092) by comparing the current consumption in no-load operation with the rated magnetising current (C00095).</li><li>Optimise oscillation damping (C00234).</li></ul></td></tr>
<tr><td>"Short circuit" (OC1) or "Clamp operation active" (OC11) error messages at short acceleration time (C00012) in proportion to the load (controller cannot follow the dynamic processes).</td><td><ul><li>Increase the gain of the torque controller (C00073/2).</li><li>Reduce the reset time of the torque controller (C00074/2).</li><li>Increase the acceleration (C00012)/deceleration ( C00013) time.</li></ul></td></tr>
<tr><td>Mechanical resonance at certain speeds.</td><td>The L_NSet_1 function block masks out those speed ranges that include resonance.</td></tr>
<tr><td>Speed variations in no-load operation for speeds > 1/3 rated speed.</td><td>Minimise speed oscillations with oscillation damping (C00234).</td></tr>
<tr><td>Drive runs unstable.</td><td rowspan="2">Check set motor data (nameplate data and equivalent circuit diagram data).<br>▸ Motor selection/Motor data (📖 121)</td></tr>
<tr><td>Setpoint speed and actual speed differ strongly.</td></tr>
<tr><td>The torque required is not generated at standstill.</td><td rowspan="3">Increase motor magnetising current (C00095).</td></tr>
<tr><td>Current overshoots occur when heavy loads are accelerated from standstill (OC1 or OC11 error).</td></tr>
<tr><td>The machine runs uneven.</td></tr>
</table>

## 5.9 Sensorless control for synchronous motors (SLPSM)

The sensorless control for synchronous motors is based on a decoupled and separated control of the torque-creating and field-creating current share of synchronous motors. In contrast to the servo control, the actual speed value and the rotor position are reconstructed via a motor model.

**Stop!**

- The sensorless control for synchronous motors is only possible up to a maximum output frequency of 650 Hz!
  – Depending on the number of motor pole pairs, the reference speed (C00011) may only be selected that high that the output frequency displayed in C00059 is lower than 650 Hz.
- We recommend to select a power-adapted combination of inverter and motor.
- The Lenze setting permits the operation of a power-adapted motor. Optimal operation is only possible if either:
  – the motor is selected via the Lenze motor catalogue
  – the motor nameplate data are entered and motor parameter identification is carried out afterwards
  *- or -*
  – the nameplate data and equivalent circuit data of the motor (motor leakage inductance and motor stator resistance) are entered manually.
- When you enter the motor nameplate data, take into account the phase connection implemented for the motor (star or delta connection). Only enter the data applying to the selected connection type.
- In order to protect the motor (e.g. from demagnetisation), we recommend setting the ultimate motor current in C00939. This ensures motor protection even with an unstable operation. ▶ Maximum current monitoring (📖 283)
- Controller enable is only possible if the motor is at standstill.
  – When the controller is enabled, a jerk may occur due to an angle jump since the rotor displacement angle is not known after controller enable. For some applications, this jerk in the machine is not acceptable.
  – From version 02.00.00, the rotor displacement angle is identified with every controller enable in the Lenze setting, and thus a jerk in the machine after controller enable can be avoided. ▶ Pole position identification without motion (📖 144)
  – A flying restart circuit for synchronising to rotating motors is in preparation.

**Stop!**

- The injection of a constant current may cause an unwanted heating of the motor at controlled operation.
  - We recommend using a temperature feedback via KTY, PTC or thermal contact.
    ▶ Motor temperature monitoring (KTY) (📖 323)
    ▶ Motor temperature monitoring (PTC) (📖 278)

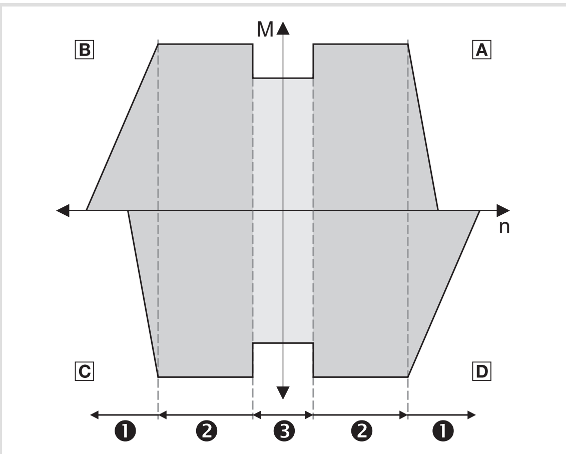
**Note!**

Currently, the sensorless control does not contain a flying restart function that enables a synchronisation of the controller to a rotating machine.

- Thus, we recommend taking measures for preventing overvoltages at operation in generator mode (e.g. brake resistor).
- By any means, the delay time for the "DC-bus overvoltage" error trigger in C00601/1 must be set to 0 s.

The motor model-based speed monitoring requires a rotating machine. Thus, the operational performance of the sensorless control for synchronous motors is divided into two categories:

1. Open-loop controlled operation ($|n_{setpoint}| < n_{C00996}$)
   - In the range of low speeds, the speed of a synchronous motor is not possible. Thus, only an adjustable and constant current is injected that enables an acceleration.
2. Closed-loop controlled operation ($|n_{setpoint}| > n_{C00996}$)
   - In this range, the rotor flux position and the speed are reconstructed via an observer. The control is carried out field-oriented. Only the current is injected that is needed for the required torque.




Ⓐ Operation in motor mode (CW rotation)
Ⓑ Operation in generator mode (CCW rotation)
Ⓒ Operation in motor mode (CCW rotation)
Ⓓ Operation in generator mode (CW rotation)

❶ Field weakening (in preparation)
❷ Closed-loop controlled operation
❸ Open-loop controlled operation

[5-13] Operating ranges of the sensorless control for synchronous motors

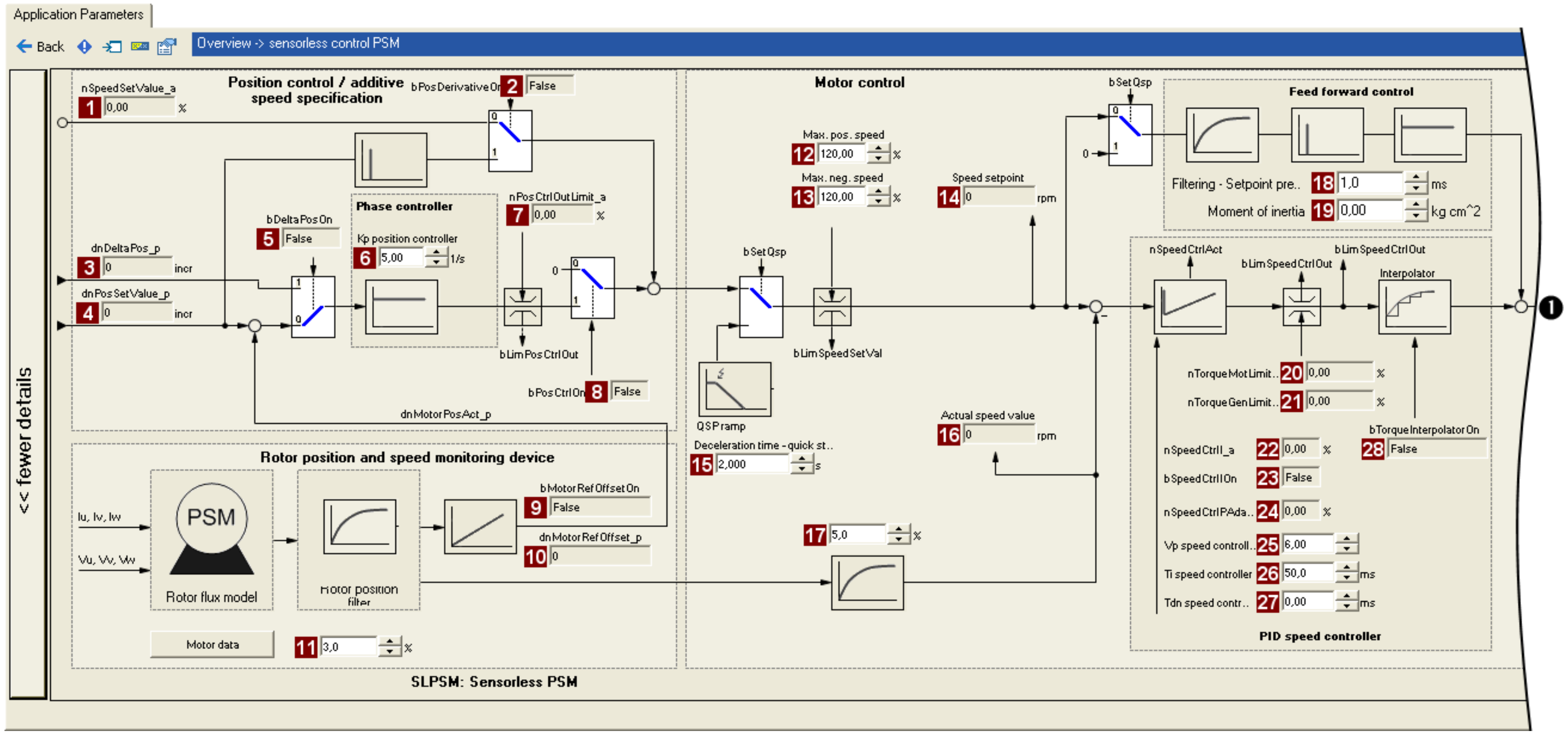
The sensorless control for synchronous motors has similar advantages for the closed-loop controlled operating range and the servo control (SC) for synchronous motors. Compared to asynchronous motors, there are the following advantages:

- Higher power density of the motor
- Higher efficiency
- Limitation of the maximum torque in motor mode and generator mode in closed-loop operating range
- Implementation of simple positioning

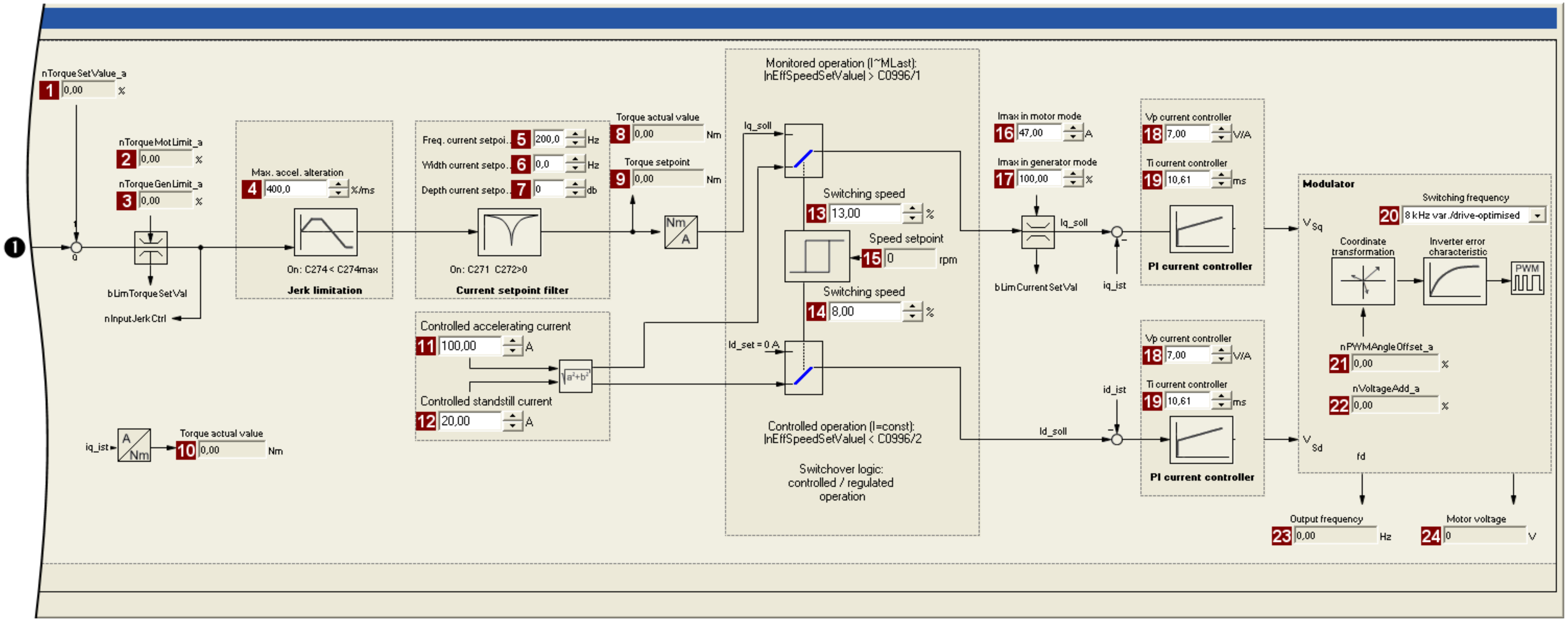
### 5.9.1 Parameterisation dialog/signal flow

**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Select the motor control "3: SLPSM: Sensorless PSM" from the *Overview* dialog level in the **Motor control** list field:
4. Click the **Motor control servo SLPSM** button to change to the *Overview* → *Motor control vector* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed.




| | Parameter | Info |
|---|---|---|
| 1 | C00830/22 | Speed setpoint |
| 2 | C00833/67 | MCTRL: bPosDerivativeOn |
| 3 | C00834/4 | MCTRL: dnDeltaPos_p |
| 4 | C00834/5 | MCTRL: dnPosSetValue_p |
| 5 | C00833/35 | MCTRL: bDeltaPosOn |
| 6 | C00254 | Kp position controller |
| 7 | C00830/21 | MCTRL: nPosCtrlOutLimit_a |
| 8 | C00833/27 | MCTRL: bPosCtrlOn |
| 9 | C00833/68 | MCTRL: bMotorRefOffsetOn |
| 10 | C00834/6 | MCTRL: dnMotorRefOffset_p |
| 11 | C00998/1 | SLPSM: Filter time rotor position |
| 12 | C00909/1 | Max. pos. speed |
| 13 | C00909/2 | Max. neg. speed |
| 14 | C00050 | Speed setpoint |
| 15 | C00105 | Decel. time - quick stop |
| 16 | C00051 | Actual speed value |
| 17 | C00998/2 | SLPSM: Filter time actual speed value |
| 18 | C00275 | Setpoint feedforward control filtering |
| 19 | C00273 | Moment of inertia |
| 20 | C00830/29 | Limitation of torque in motor mode |
| 21 | C00830/28 | Limitation of torque in generator mode |
| 22 | C00830/24 | MCTRL: nSpeedCtrlI_a |
| 23 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 24 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 25 | C00070/3 | SLPSM: Vp speed controller |
| 26 | C00071/3 | SLPSM: Ti speed controller |
| 27 | C00072 | SC: Tdn speed controller |
| 28 | C00833/29 | MCTRL: bTorqueInterpolatorOn |




| Parameter | | Info |
|---|---|---|
| 1 | C00830/27 | MCTRL: nTorqueSetValue_a |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00274 | SC: Max. change in acceleration |
| 5 | C00270 | SC: Freq. current setpoint filter |
| 6 | C00271 | SC: Current setpoint filter width |
| 7 | C00272 | SC: Current setpoint filter depth |
| 8 | C00056/2 | Actual torque value |
| 9 | C00056/1 | Torque setpoint |
| 10 | C00056/2 | Actual torque value |
| 11 | C00995/1 | SLPSM: Open-loop controlled accelerating current |
| 12 | C00995/2 | SLPSM: Open-loop controlled standstill current |
| 13 | C00996/1 | SLPSM: Closed-loop controlled switching speed |
| 14 | C00996/2 | SLPSM: Open-loop controlled switching speed |
| 15 | C00050 | Speed setpoint |
| 16 | C00022 | Imax in motor mode |
| 17 | C00023 | Imax in generator mode |
| 18 | C00075 | Vp current controller |
| 19 | C00076 | Ti current controller |
| 20 | C00018 | Switching frequency |
| 21 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 22 | C00830/31 | MCTRL: nVoltageAdd_a |
| 23 | C00058 | Output frequency |
| 24 | C00052 | Motor voltage |

## 5.9.2 Types of control

Sensorless control for synchronous motors can only be executed in the "Speed control with torque limitation" (*bTorquemodeOn* = FALSE) mode.

### Speed control with torque limitation

A speed setpoint is selected and the drive system is operated in a speed-controlled manner. For adapting the operational performance, the overload in the drive train can be limited:

- The torque is limited via the torque setpoint.
- The torque setpoint is identical to the value at the output of the speed controller, *nOutputSpeedCtrl*.
- To avoid overload in the drive train, the torque in motor mode can be limited via the *nTorqueMotLimit_a* process input signal, and the torque in generator mode can be limited via the *nTorqueGenLimit_a* process input signal:

| **Identifier**<br>DIS code \| data type | **Information/possible settings** |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

**Stop!**

Torque limitation is only active in the closed-loop controlled operation ($|n_{Setpoint}| > n_{C00996}$)!

- It must be prevented that the actual speed value is braked into the non-observable area due to the torque limitation!

**Note!**

To avoid instabilities during operation, the torque limit values are internally processed as absolute values.

### 5.9.3 Basic settings

The following "Initial commissioning steps" must be performed to commission the sensorless control for synchronous motors:

<table>
<tr><th colspan="3">Initial commissioning steps</th></tr>
<tr><td>1.</td><td colspan="2">Select motor control:<br>C00006 = "3: SLPSM: Sensorless PSM"</td></tr>
<tr><td>2.</td><td colspan="2">Set the motor selection/motor data<br>• When selecting and parameterising the motor, the motor nameplate data and the equivalent circuit diagram data are relevant. Detailed information can be found in the "Motor selection/Motor data" chapter. (📖 121)<br>Depending on the motor manufacturer, proceed as follows:</td></tr>
<tr><td></td><td>Lenze motor:<br>Selecting a motor from the motor catalogue in the »Engineer«<br>- or -<br>1. Set the motor nameplate data<br>2. Automatic motor data identification</td><td>Third party manufacturer's motor:<br>1. Set the motor nameplate data<br>2. Automatic motor data identification or set known equivalent circuit diagram manually:<br>C00084: Motor stator resistance<br>C00085: Motor stator leakage inductance</td></tr>
<tr><td>3.</td><td colspan="2">Set speed switching thresholds between open-loop and closed-loop controlled operation:<br>• Set transition speed from open-loop to closed-loop operation in C00996/1 in [%] with regard to the rated motor speed (C00087).<br>• Set transition speed from closed-loop to open-loop operation in C00996/2 in [%] with regard to the rated motor speed (C00087).<br>Tip!<br>• With voltage-adjusted motors, a speed switching threshold of 10 % is recommended.<br>• As a rule of thumb, the speed switching threshold should be selected as follows:<br>$$C00996/1...2\ [\%] = \frac{U_{Rated,\,motor}[V]}{U_{Rated,\,FI}[V]} \cdot 10$$</td></tr>
<tr><td>4.</td><td colspan="2">Set open-loop accelerating current in C00995/1 in [%] with regard to the rated motor current (C00088).<br>• This value defines the height of the current that is injected during the acceleration process.<br>• The accelerating current must be dimensioned so that the required torque in the lower speed range can always be reached (acceleration torque + load torque):<br>$$C00995/1\ [\%] = \frac{M_{Meax}[Nm]}{M_{Rated}[Nm]} \cdot I_{Rated,\,motor}[A] \cdot 1.3$$</td></tr>
<tr><td>5.</td><td colspan="2">Set open-loop steady-state current in C00995/2 in [%] with regard to the rated motor current (C00088).<br>• This value defines the height of the current for processes without acceleration (e.g. standstill or constant setpoint speed ).</td></tr>
<tr><td>6.</td><td colspan="2">For improving the operating characteristics:<br>If required, adapt the filter time for reconstructing the rotor position and the actual speed value through the motor model in C00998/1 and C00998/2.<br>• We recommend using the Lenze setting:<br>Filter time rotor position (C00998/1) = 3 ms<br>Filter time actual speed value (C00998/2) = 5 ms<br>• Deviant from this, the following value range can be used:<br>Filter time rotor position (C00998/1) = 2 ... 5 ms<br>Filter time actual speed value (C00998/2) = 3 ... 8 ms</td></tr>
<tr><td>7.</td><td colspan="2">For protecting the motor from demagnetisation:<br>Set the ultimate current in C00939.</td></tr>
</table>

**Note!**

The Lenze settings of the current controller are predefined for a power-adapted motor. For an optimal drive behaviour of a synchronous motor, we recommend to adapt the controller settings.

**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in the "Optimising the control mode" chapter.

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (240)

### 5.9.4 Optimising the control mode

The "optimisation steps" given in the table below serve to further optimise the control behaviour of the sensorless control for synchronous motors and adjust it to the concrete application.

▶ Detailed information on the individual steps can be found in the following subchapters.

Generally, the following optimisation steps are recommended:

| Optimisation steps | |
|---|---|
| 1. | Optimise current controller. (📖 207)<br>• The current controller should always be optimised if a motor of a third-party manufacturer with unknown motor data is used! |
| 2. | Optimise speed controller. (📖 208)<br>• The setting of the speed controller must be adapted depending on the mechanical path. |
| 3. | Optimise response to setpoint changes and determine mass inertia. (📖 212)<br>• For an optimal reference behaviour, the total moment of inertia can be used to make a feedforward control of the speed setpoint. |

**Note!**

**Current setpoint filter (band-stop filter) / jerk limitation**

The use of the functions is only recommended in exceptional cases.

The functions are described in the "Servo control (SC)" chapter:

- Setting the current setpoint filter (band-stop filter) (📖 231)
- Adapting the max. acceleration change (jerk limitation) (📖 232)

### 5.9.4.1 Optimise current controller

**Note!**

An optimisation of the current controller should generally be carried out unless a power-adapted standard motor is used or the motor has been selected from the motor catalogue of the »Engineer«!

An optimisation of the current controller is sensible since the two control parameters gain (C00075) and reset time (C00076) depend on the required maximum current and the set switching frequency.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00075 | Vp current controller | 7.00 | V/A |
| C00076 | Ti current controller | 10.61 | ms |

▶ Gain and reset time can be calculated as per the following formulae:

$$V_p = \frac{L_{ss}[H]}{T_E[s]}$$

$$T_i = \frac{L_{ss}[H]}{R_s[\Omega]}$$

$V_p$ = Current controller gain (C00075)
$T_i$ = Current controller reset time (C00076)
$L_{ss}$ = Motor stator leakage inductance (C00085)
$R_s$ = Motor stator resistance (C00084)
$T_E$ = Equivalent time constant (= 500 µs)

### 5.9.4.2 Optimise speed controller

The speed controller is in the form of a PID controller with an additional differential speed-setpoint gain. For optimum behaviour, the PID speed controller has to be optimised and the overall mass inertia of the drive train has to be determined.

- In the Lenze setting, the configuration of the speed controller provides robustness and moderate dynamics.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00070/3 | SLPSM: Vp speed controller | 3.00 | |
| C00071/3 | SLPSM: Ti speed controller | 100.0 | ms |
| C00072 | SC: Tdn speed controller | 0.00 | ms |

#### Speed controller gain Vp

The gain Vp (C00070/3) of the speed controller is defined in a scaled representation which enables a comparable parameterisation almost independent of the power of the motor or inverter. Here, the speed input difference of the controller is scaled to the rated motor speed whereas the output torque refers to the rated motor torque. A gain of 10 means that a speed difference of 1 % is gained through the P component with 10 % torque.

If the rated data of the motor and the mass inertia of the drive system are known, we recommend the following setting:

$$V_P \approx 0.2 \ldots 0.5 \cdot \frac{T_M[s]}{0.01[s]}$$

$$T_M[s] = \frac{2 \cdot \pi \cdot n_N[rpm]}{M_N[Nm] \cdot 60} \cdot J_{Drive,\,total}[kgm^2]$$

$$M_N[Nm] = \frac{P_N[W] \cdot 60}{2 \cdot \pi \cdot n_N[rpm]}$$

$V_P$ = Gain of the speed controller (C00070/3)
$T_M$ = Time constant for the acceleration of the motor
$M_N$ = Rated motor torque
$n_N$ = Rated motor speed
$J_{drive,\,total}$ = Total moment of inertia of the drive

[5-14] Recommendation for the setting of the gain of the speed controller

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

1. Specify speed setpoint.
   – A small speed just above the switching threshold is recommended in the closed-loop controlled operation.
2. Increase Vp (C00070/3) until the drive starts to oscillate (observe engine noise).
3. Reduce Vp (C00070/3) until the drive runs stable again.
4. Reduce Vp (C00070/3) to approx. half the value.
5. Afterwards check results of the optimisation in the entire speed range (one-time passing through of the speed range).

**Tip!**

Values recommended by Lenze for the setting of the (proportional) gain:

- For drive systems without feedback: Vp = 2 ... 8
- For drive systems with a good disturbance behaviour: Vp > 6

### Speed controller reset time Ti

Apart from setting the P component, C00071/3 provides the possibility to take influence on the I component of the PI controller.

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

1. Specify speed setpoint.
2. Reduce Ti (C00071/3) until the drive starts to oscillate (observe engine noise).
3. Increase Ti (C00071/3) until the drive runs stable again.
4. Increase Ti (C00071/3) to approx. twice the value.

**Tip!**

Value range recommended by Lenze for the setting of the reset time:

Ti = 20 ms ... 150 ms

#### Using the ramp response for setting the speed controller

If the mechanical components cannot be operated at the stability limit, the ramp response can also be used for setting the speed controller.

**Stop!**

If the controller parameters are preset unfavourably, the control can tend to heavy overshoots up to instability!

- Following and speed errors can adopt very high values.
- If the mechanics are sensitive, the corresponding monitoring functions are to be activated.

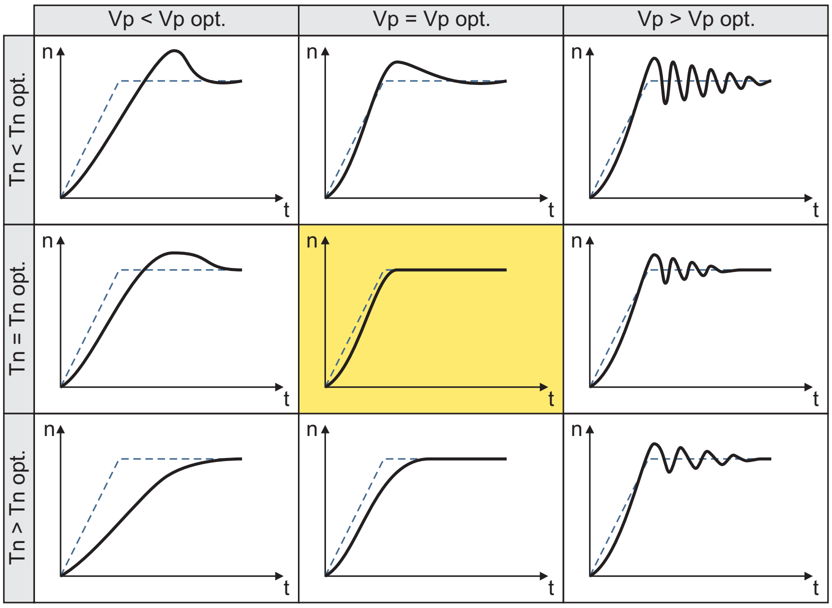
**Note!**

For an optimal setting, we recommend to determine the mass inertia (optimal response to setpoint changes) first.

▶ Optimise response to setpoint changes and determine mass inertia (📖 212)

**How to optimise the speed controller setting by means of the ramp response:**

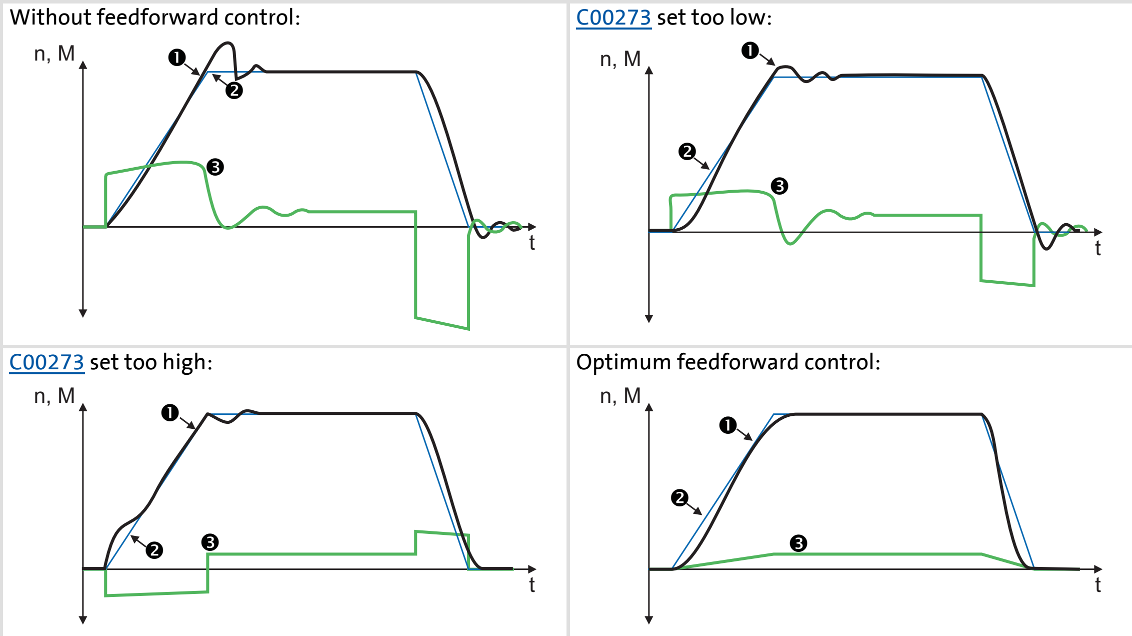
1. Run a typical speed profile and record the ramp response of the speed using the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
2. Evaluate the ramp response:




   - Solid line = ramp response (actual speed value)
   - Dash line = speed setpoint
3. Change gain Vp in C00070/3 and reset time Tn in C00071/3.
4. Repeat steps 1 ... 3 until the optimum ramp response is reached.

### 5.9.4.3 Optimise response to setpoint changes and determine mass inertia

Setting the total moment of inertia under C00273 provides the optimum torque feedforward control. Depending on the application, an adjustment of the setting under C00273 may be necessary to optimise the response to position/speed setpoint changes by means of the torque feedforward control.




❶ Actual speed value (signal *nMotorSpeedAct*)
❷ Speed setpoint (signal *nMotorSpeedSet*)
❸ Speed controller output (signal *nOutputSpeedCtrl*)

[5-15] Typical signal characteristics for different settings of the load moment of inertia

**How to optimise the torque feedforward control:**

1. Run a typical speed profile and record the inputs and outputs of the speed controller with the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
     *nOutputSpeedCtrl_a* (speed controller output)
2. Estimate the moment of inertia and set it in C00273 in relation to the motor end (i.e. with account being taken of the gearbox factors).
3. Repeat the data logger recording (see step 1).

   Now the data logger should show that part of the required torque is generated by the feedforward control and the speed controller output signal (*nOutputSpeedCtrl_a*) is correspondingly smaller. The resulting following error decreases.
4. Change the setting in C00273 and repeat the data logger recording until the intended response to setpoint changes is reached.
   - The optimisation could aim at the speed controller being completely relieved (see signal characteristics in Fig. [5-15]).
5. Save the parameter set (device command: C00002/11).

## 5.10 Servo control (SC)

Field-oriented servo control (SC) is based on a decoupled, separate control of the torque-producing and the field-producing current component. The motor control is based on a field-oriented, cascaded controller structure with feedback function and enables dynamic and stable operation in all of the four quadrants. It can be used for synchronous motors (PSM) and asynchronous motors (ASM).

### Stop!

- We recommend to select a power-adapted combination of inverter and motor.
- The Lenze setting permits the operation of a power-adapted motor. Optimal operation is only possible if either:
  - the motor is selected via the Lenze motor catalogue
  - the motor nameplate data are entered and motor parameter identification is carried out afterwards
    *- or -*
  - the nameplate data and equivalent circuit data of the motor are entered manually.
- When you enter the motor nameplate data, take into account the phase connection implemented for the motor (star or delta connection). Only enter the data applying to the selected connection type.

### Note!

- The servo control required a speed feedback!
  - See chapter entitled "Encoder/feedback system". (📖 293)
- For closed-loop control of a synchronous motor (PSM), the pole position of the motor must be known!
  - See chapter entitled "Pole position identification (PPI)". (📖 139)
- For closed-loop control of an asynchronous motor (ASM), the maximum current (C00022) should be higher than the magnetising current as otherwise the motor does not generate a torque.

Generally, the servo control offers the same advantages as the sensorless vector control (SLVC), i.e. compared to the V/f characteristic control, the servo control (SC) can be used to achieve

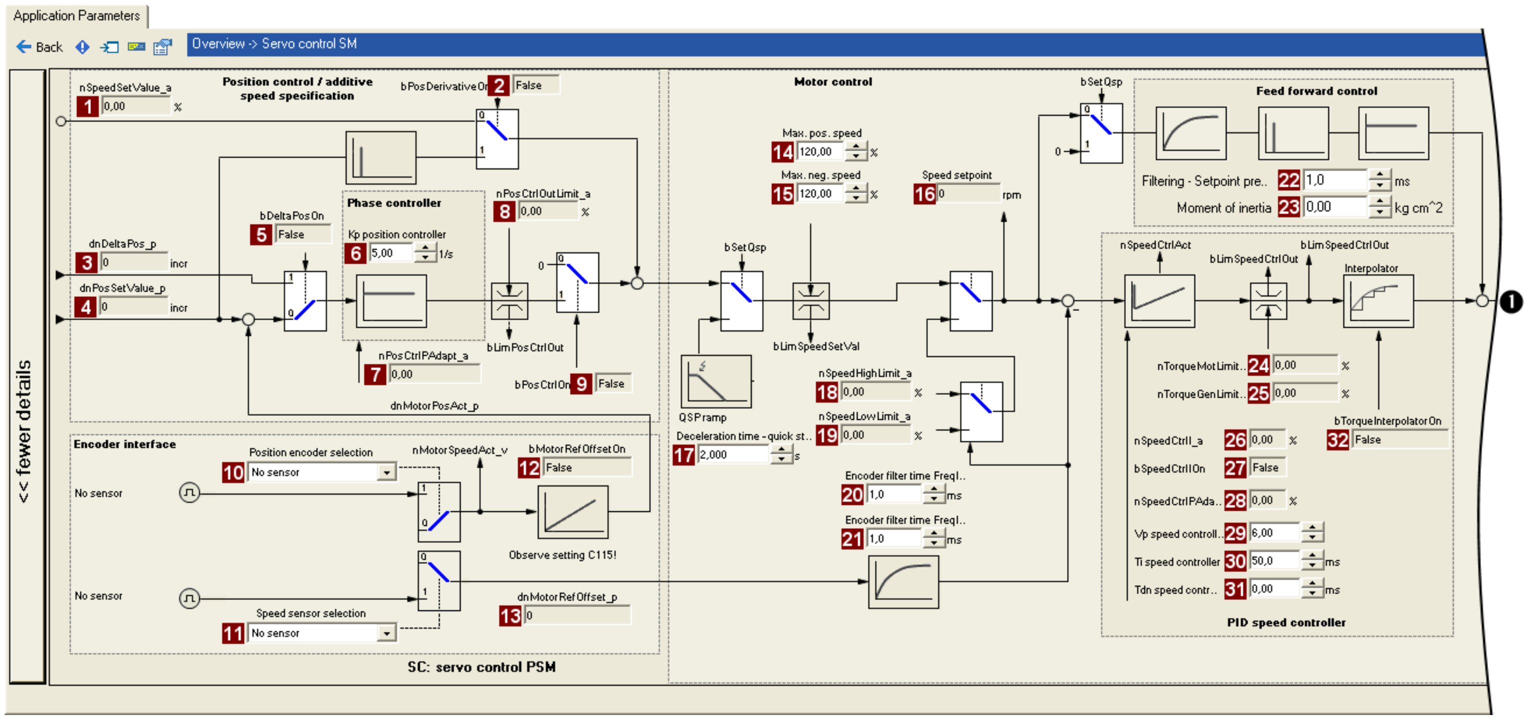
- A higher maximum torque throughout the entire speed range
- A higher speed accuracy
- A higher concentricity factor
- A higher level of efficiency
- The implementation of torque-actuated operation with speed limitation
- The limitation of the maximum torque in motor and generator mode for speed-actuated operation

### 5.10.1 Parameterisation dialog/signal flow

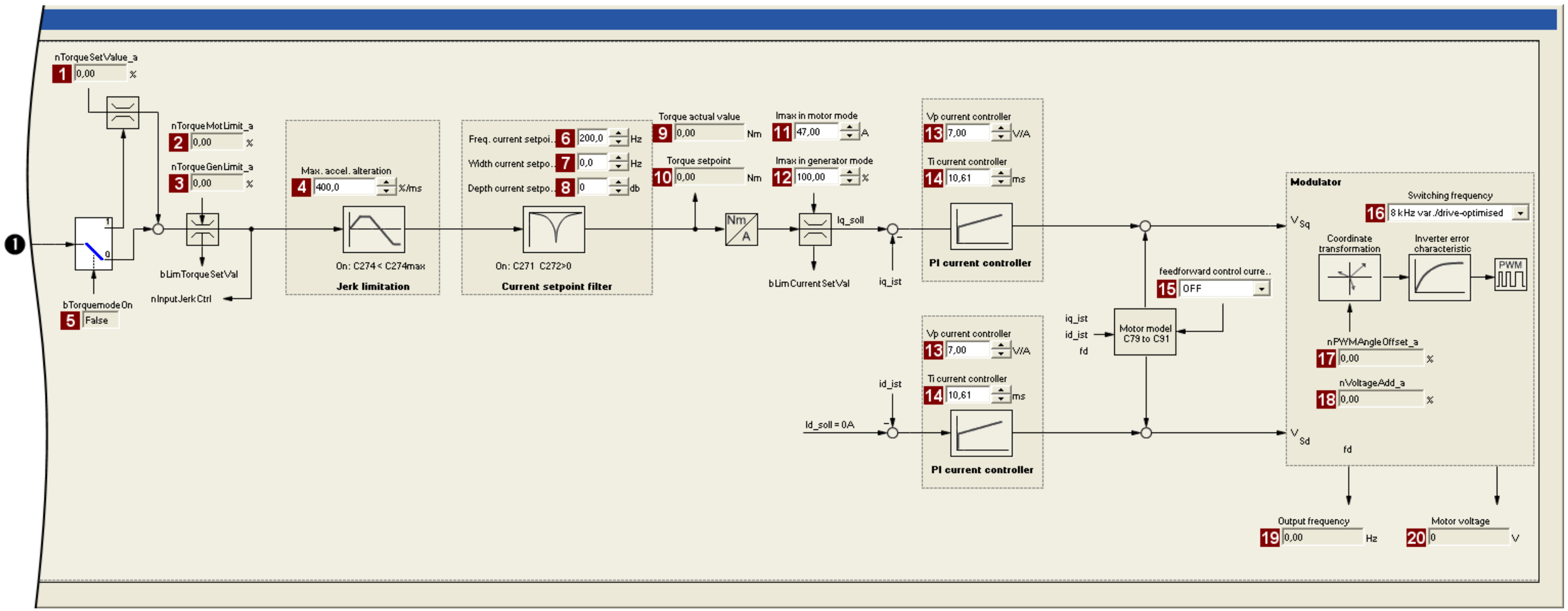
**Proceed as follows to open the dialog for parameterising the motor control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Select the motor control from the *Overview* dialog level in the **Motor control** (C00006) list field:
   - "1: SC: Servo control PSM" for synchronous motor
     - or -
   - "2: SC: Servo control ASM" for asynchronous motor
4. Click the **Motor control servo** button to change to the *Overview* → *Motor control vector* dialog box.
   - This dialog level only shows a simplified signal flow with the most important parameters.
   - When you click the **>>More details** button in the left-most position, a signal flow with more details/parameters is displayed, as shown in the following subchapter.

## Signal flow for servo control for synchronous motor (PSM):




| | Parameter | Info |
|---|---|---|
| 1 | C00830/22 | Speed setpoint |
| 2 | C00833/67 | MCTRL: bPosDerivativeOn |
| 3 | C00834/4 | MCTRL: dnDeltaPos_p |
| 4 | C00834/5 | MCTRL: dnPosSetValue_p |
| 5 | C00833/35 | MCTRL: bDeltaPosOn |
| 6 | C00254 | Kp position controller |
| 7 | C00830/20 | MCTRL: nPosCtrlPAdapt_a |
| 8 | C00830/21 | MCTRL: nPosCtrlOutLimit_a |
| 9 | C00833/27 | MCTRL: bPosCtrlOn |
| 10 | C00490 | Position encoder selection |
| 11 | C00495 | Speed sensor selection |
| 12 | C00833/68 | MCTRL: bMotorRefOffsetOn |
| 13 | C00834/6 | MCTRL: dnMotorRefOffset_p |
| 14 | C00909/1 | Max. pos. speed |
| 15 | C00909/2 | Max. neg. speed |
| 16 | C00050 | Speed setpoint |
| 17 | C00105 | Decel. time - quick stop |
| 18 | C00830/88 | MCTRL: nSpeedHighLimit_a |
| 19 | C00830/23 | MCTRL: nSpeedLowLimit_a |
| 20 | C00497/1 | Encoder filter time FreqIn12 |
| 21 | C00497/2 | Encoder filter time FreqIn67 |
| 22 | C00275 | Setpoint feedforward control filtering |
| 23 | C00273 | Moment of inertia |
| 24 | C00830/29 | Limitation of torque in motor mode |
| 25 | C00830/28 | Limitation of torque in generator mode |
| 26 | C00830/24 | MCTRL: nSpeedCtrlI_a |
| 27 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 28 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 29 | C00070/2 | SC: Vp speed controller |
| 30 | C00071/2 | SC: Ti speed controller |
| 31 | C00072 | SC: Tdn speed controller |
| 32 | C00833/29 | MCTRL: bTorqueInterpolatorOn |

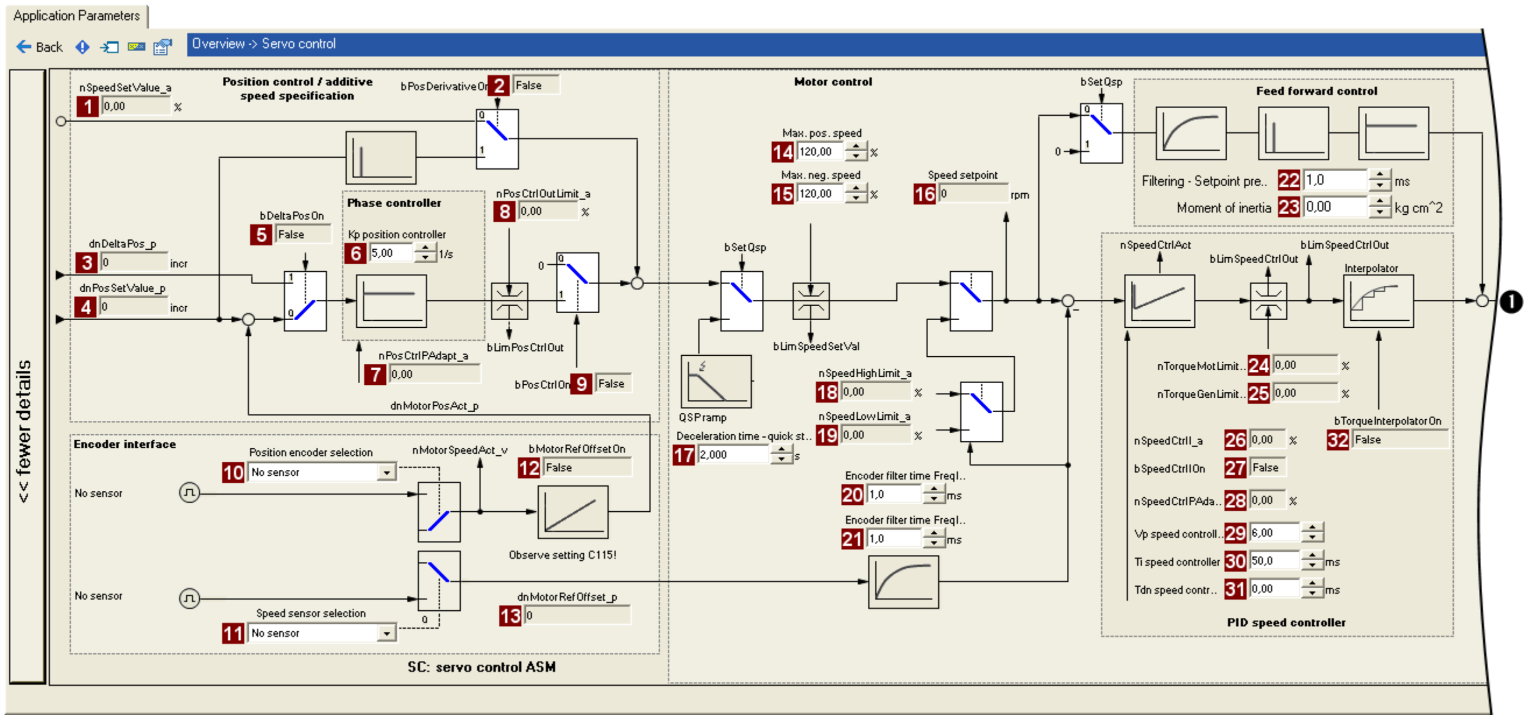



| Parameter | | Info |
|---|---|---|
| 1 | C00830/27 | MCTRL: nTorqueSetValue_a |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00274 | SC: Max. change in acceleration |
| 5 | C00833/30 | MCTRL: bTorqueModeOn |
| 6 | C00280 | SC: Filter time const. DC detection |
| 7 | C00052 | Motor voltage |
| 8 | C00577 | SC: Vp field weakening controller |
| 9 | C00578 | SC: Tn field weakening controller |
| 10 | C00576 | SC: Field feedforward control |

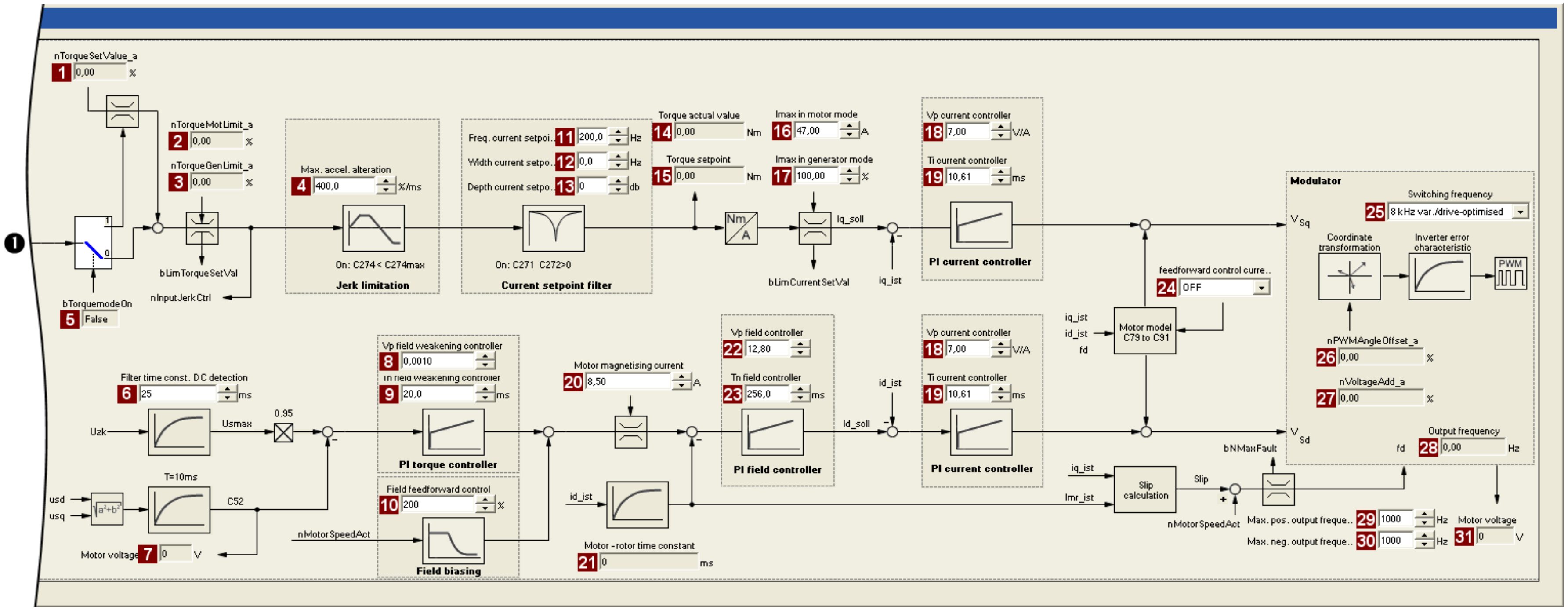
| Parameter | | Info |
|---|---|---|
| 6 | C00270 | SC: Freq. current setpoint filter |
| 7 | C00271 | SC: Current setpoint filter width |
| 8 | C00272 | SC: Current setpoint filter depth |
| 9 | C00056/2 | Actual torque value |
| 10 | C00056/1 | Torque setpoint |
| 11 | C00022 | Imax in motor mode |
| 12 | C00023 | Imax in generator mode |
| 13 | C00075 | Vp current controller |
| 14 | C00076 | Ti current controller |

| Parameter | | Info |
|---|---|---|
| 15 | C00079/1 | SC: Current controller - feedforward control |
| 16 | C00018 | Switching frequency |
| 17 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 18 | C00830/31 | MCTRL: nVoltageAdd_a |
| 19 | C00058 | Output frequency |
| 20 | C00052 | Motor voltage |

## Signal flow for servo control for asynchronous motor (ASM):




| | Parameter | Info |
|---|---|---|
| 1 | C00830/22 | Speed setpoint |
| 2 | C00833/67 | MCTRL: bPosDerivativeOn |
| 3 | C00834/4 | MCTRL: dnDeltaPos_p |
| 4 | C00834/5 | MCTRL: dnPosSetValue_p |
| 5 | C00833/35 | MCTRL: bDeltaPosOn |
| 6 | C00254 | Kp position controller |
| 7 | C00830/20 | MCTRL: nPosCtrlPAdapt_a |
| 8 | C00830/21 | MCTRL: nPosCtrlOutLimit_a |
| 9 | C00833/27 | MCTRL: bPosCtrlOn |
| 10 | C00490 | Position encoder selection |
| 11 | C00495 | Speed sensor selection |
| 12 | C00833/68 | MCTRL: bMotorRefOffsetOn |
| 13 | C00834/6 | MCTRL: dnMotorRefOffset_p |
| 14 | C00909/1 | Max. pos. speed |
| 15 | C00909/2 | Max. neg. speed |
| 16 | C00050 | Speed setpoint |
| 17 | C00105 | Decel. time - quick stop |
| 18 | C00830/88 | MCTRL: nSpeedHighLimit_a |
| 19 | C00830/23 | MCTRL: nSpeedLowLimit_a |
| 20 | C00497/1 | Encoder filter time FreqIn12 |
| 21 | C00497/2 | Encoder filter time FreqIn67 |
| 22 | C00275 | Setpoint feedforward control filtering |
| 23 | C00273 | Moment of inertia |
| 24 | C00830/29 | Limitation of torque in motor mode |
| 25 | C00830/28 | Limitation of torque in generator mode |
| 26 | C00830/24 | MCTRL: nSpeedCtrlI_a |
| 27 | C00833/31 | MCTRL: bSpeedCtrlIOn |
| 28 | C00830/25 | MCTRL: nSpeedCtrlPAdapt_a |
| 29 | C00070/2 | SC: Vp speed controller |
| 30 | C00071/2 | SC: Ti speed controller |
| 31 | C00072 | SC: Tdn speed controller |
| 32 | C00833/29 | MCTRL: bTorqueInterpolatorOn |




| Parameter | | Info |
|---|---|---|
| 1 | C00830/27 | MCTRL: nTorqueSetValue_a |
| 2 | C00830/29 | Limitation of torque in motor mode |
| 3 | C00830/28 | Limitation of torque in generator mode |
| 4 | C00274 | SC: Max. change in acceleration |
| 5 | C00833/30 | MCTRL: bTorqueModeOn |
| 6 | C00280 | SC: Filter time const. DC detection |
| 7 | C00052 | Motor voltage |
| 8 | C00577 | SC: Vp field weakening controller |
| 9 | C00578 | SC: Tn field weakening controller |
| 10 | C00576 | SC: Field feedforward control |
| 11 | C00270 | SC: Freq. current setpoint filter |
| 12 | C00271 | SC: Current setpoint filter width |
| 13 | C00272 | SC: Current setpoint filter depth |
| 14 | C00056/2 | Actual torque value |
| 15 | C00056/1 | Torque setpoint |
| 16 | C00022 | Imax in motor mode |
| 17 | C00023 | Imax in generator mode |
| 18 | C00075 | Vp current controller |
| 19 | C00076 | Ti current controller |
| 20 | C00095 | Motor magnetising current |
| 21 | C00083 | Motor rotor time constant |
| 22 | C00077 | SC: Vp field controller |
| 23 | C00078 | SC: Tn field controller |
| 24 | C00079/1 | SC: Current controller - feedforward control |
| 25 | C00018 | Switching frequency |
| 26 | C00830/32 | MCTRL: nPWMAngleOffset_a |
| 27 | C00830/31 | MCTRL: nVoltageAdd_a |
| 28 | C00058 | Output frequency |
| 29 | C00910/1 | Max. pos. output frequency |
| 30 | C00910/2 | Max. neg. output frequency |
| 31 | C00052 | Motor voltage |

## 5.10.2 Types of control

The servo control can be operated in two different modes:

- Speed control with torque limitation (*bTorquemodeOn* = FALSE)
- Torque control with speed limitation (*bTorquemodeOn* = TRUE)

**Tip!**

A position-controlled application requires a speed control with torque limitation (*bTorquemodeOn* = FALSE).

### 5.10.2.1 Speed control with torque limitation

A speed setpoint is selected and the drive system is operated in a speed-controlled manner.

The operational performance can be adapted in the following ways:

- Overload limitation in the drive train
  - The torque is limited via the torque setpoint.
  - The torque setpoint is identical to the value at the output of the speed controller, *nOutputSpeedCtrl*.
  - To avoid overload in the drive train, the torque in motor mode can be limited via the *nTorqueMotLimit_a* process input signal, and the torque in generator mode can be limited via the *nTorqueGenLimit_a* process input signal:

| Identifier<br>DIS code \| data type | Information/possible settings |
|---|---|
| nTorqueMotLimit_a<br>C00830/29 \| INT | Torque limitation in motor mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: 0 ... +199.99 %<br>• If keypad control is performed: Parameterisable via C00728/1. |
| nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in generator mode<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>• Setting range: -199.99 ... 0 %<br>• If keypad control is performed: Parameterisable via C00728/2. |

**Note!**

To avoid instabilities during operation, the torque limit values are internally processed as absolute values.

- Motor current limitation
  - A cross current setpoint is calculated from the torque setpoint which is limited depending on the magnetising current, the max. current in motor mode (C00022), and the max. current in generator mode (C00023).
  - Here, the total current injected into the motor does not exceed the max. currents in motor and generator mode.

### 5.10.2.2 Torque control with speed limitation

For torque-controlled operation, a torque setpoint is defined in the drive system.

- The torque setpoint is calculated directly from *nTorqueSetValue_a*.
- The speed is defined by the process.
- Due to its limitation, the speed-controlled drive can only rotate within a speed range whose positive speed is limited by the value of *nSpeedHighLimit_a* and whose negative speed is limited by the value of *nSpeedLowLimit_a*.

| **Identifier**<br>DIS code \| data type | **Information/possible settings** |
|---|---|
| nTorqueSetValue_a<br>C00830/27 \| INT | Torque setpoint / additive torque<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| nSpeedHighLimit_a<br>C00830/88 \| INT | Upper speed limit for speed limitation<br>(only for torque-controlled operation)<br>• Scaling: 16384 ≡ 100 % rated speed (C00011) |
| nSpeedLowLimit_a<br>C00830/23 \| INT | Lower speed limit for speed limitation<br>(only for torque-controlled operation)<br>• Scaling: 16384 ≡ 100 % rated speed (C00011) |

### 5.10.3 Basic settings

The following "Initial commissioning steps" must be performed to commission the servo control:

| Initial commissioning steps | | |
|---|---|---|
| 1. | Determine the motor control:<br>• For the closed-loop control of a synchronous motor (PSM):<br>C00006 = "1: SC: Servo control PSM"<br>• For the closed-loop control of an asynchronous motor (ASM):<br>C00006 = "2: SC: Servo control ASM" | |
| 2. | Set the motor selection/motor data<br>• When selecting and parameterising the motor, the motor nameplate data and the equivalent circuit diagram data are relevant. Detailed information can be found in the "Motor selection/Motor data" chapter. (🕮 121)<br>Depending on the motor manufacturer, proceed as follows: | |
| | **Lenze motor:**<br>Selecting a motor from the motor catalogue in the »Engineer«<br>- or -<br>1. Set the motor nameplate data<br>2. Automatic motor data identification | **Third party manufacturer's motor:**<br>1. Set the motor nameplate data<br>2. Automatic motor data identification or set known equivalent circuit diagram data manually:<br>C00082: Motor rotor resistance*<br>C00084: Motor stator resistance<br>C00085: Motor stator leakage inductance<br>C00092: Motor magnetising inductance*<br>C00095: Motor magnetising current*<br>* Setting only required for asynchronous motors. |
| 3. | Define the type of control:<br>*bTorquemodeOn* = FALSE: Speed control with torque limitation<br>*bTorquemodeOn* = TRUE: Torque control with speed limitation | |
| 4. | Parameterise the encoder/feedback system. ▸ Encoder/feedback system (🕮 293) | |
| 5. | Only with servo control for synchronous motors (PSM):<br>Detect pole position of the motor. ▸ Pole position identification (PPI) (🕮 139) | |

**Note!**

The Lenze settings of the controller are predefined for a power-adapted standard asynchronous motor. For an optimal drive behaviour, we recommend to adapt the controller settings.

**Tip!**

Information on the optimisation of the control mode and the adaptation to the real application is provided in the chapter "Optimising the control mode". (🕮 223)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (🕮 240)

### 5.10.4 Optimising the control mode

The "optimisation steps" given in the table below serve to further optimise the control behaviour of the servo control and adjust it to the concrete application.

- Detailed information on the individual steps can be found in the following subchapters.

Generally, the following optimisation steps are recommended:

| Optimisation steps | |
|---|---|
| 1. | Optimise current controller. (▯ 224)<br>• The current controller should always be optimised if a motor of a third-party manufacturer with unknown motor data is used! |
| 2. | Optimise speed controller. (▯ 225)<br>• The setting of the speed controller must be adapted depending on the mechanical path. |
| 3. | Optimise response to setpoint changes and determine mass inertia. (▯ 229)<br>• For an optimal reference behaviour, the total moment of inertia can be used to make a feedforward control of the speed setpoint. |

Special cases may require further optimisation steps:

| Optimisation steps | |
|---|---|
| 1. | Setting the current setpoint filter (band-stop filter). (▯ 231)<br>• In order to suppress or damp (mechanical) resonant frequencies, a current setpoint filter is integrated in the speed control loop which is switched off in the default setting but can be parameterised accordingly, if required.<br>Then readjust the speed controller: Optimise speed controller. (▯ 225) |
| 2. | Adapting the max. acceleration change (jerk limitation). (▯ 232) |
| 3. | Only with servo control for asynchronous motors (ASM):<br>Optimising the behaviour of the asynchronous motor in the field weakening range. (▯ 235)<br>(For synchronous motors, this function is in preparation) |

**Tip!**

In order to traverse a typical speed profile for optimisation of motor control, you can also use the basic function "Manual jog" with appropriately adapted manual jog parameters if this basic function is supported by the selected technology application. (▯ 533)

### 5.10.4.1 Optimise current controller

**Note!**

An optimisation of the current controller should generally be carried out unless a power-adapted standard asynchronous motor is used or the motor has been selected from the motor catalogue of the »Engineer«!

An optimisation of the current controller is sensible since the two control parameters gain (C00075) and reset time (C00076) depend on the required maximum current and the set switching frequency.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00075 | Vp current controller | 7.00 | V/A |
| C00076 | Ti current controller | 10.61 | ms |

▶ Gain and reset time can be calculated as per the following formulae:

$$V_p = \frac{L_{ss}[H]}{T_E[s]}$$

$$T_i = \frac{L_{ss}[H]}{R_s[\Omega]}$$

$V_p$ = Current controller gain (C00075)
$T_i$ = Current controller reset time (C00076)
$L_{ss}$ = Motor stator leakage inductance (C00085)
$R_s$ = Motor stator resistance (C00084)
$T_E$ = Equivalent time constant (= 500 µs)

### 5.10.4.2 Optimise speed controller

The speed controller is in the form of a PID controller with an additional differential speed-setpoint gain. For optimum behaviour, the PID speed controller has to be optimised and the overall mass inertia of the drive train has to be determined.

- In the Lenze setting, the configuration of the speed controller provides robustness and moderate dynamics.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00070/2 | SC: Vp speed controller | 6.00 | |
| C00071/2 | SC: Ti speed controller | 50.0 | ms |
| C00072 | SC: Tdn speed controller | 0.00 | ms |

#### Speed controller gain Vp

The gain Vp (C00070/2) of the speed controller is defined in a scaled representation which enables a comparable parameterisation almost independent of the power of the motor or inverter. Here, the speed input difference of the controller is scaled to the rated motor speed whereas the output torque refers to the rated motor torque. A gain of 10 means that a speed difference of 1 % is gained through the P component with 10 % torque.

If the rated data of the motor and the mass inertia of the drive system are known, we recommend the following setting:

$$V_P \approx 1.5 \ldots 3 \cdot \frac{T_M[s]}{0.01[s]}$$

$$T_M[s] = \frac{2 \cdot \pi \cdot n_N[rpm]}{M_N[Nm] \cdot 60} \cdot J_{Drive,\,total}[kgm^2]$$

$$M_N[Nm] = \frac{P_N[W] \cdot 60}{2 \cdot \pi \cdot n_N[rpm]}$$

$V_P$ = Gain of the speed controller (C00070/1)
$T_M$ = Time constant for the acceleration of the motor
$M_N$ = Rated motor torque
$n_N$ = Rated motor speed
$J_{drive,\,total}$ = Total moment of inertia of the drive

[5-16] Recommendation for the setting of the gain of the speed controller

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

1. Specify speed setpoint.
2. Increase Vp (C00070/2) until the drive is unstable (observe motor noise).
3. Reduce Vp (C00070/2) until the drive runs stable again.
4. Reduce Vp (C00070/2) to approx. half the value.

**Tip!**

Values recommended by Lenze for the setting of the (proportional) gain:

- For drive systems without feedback: Vp = 6 ... 20
- For drive systems with a good disturbance behaviour: Vp > 12

### Speed controller reset time Ti

Apart from setting the P component, C00071/2 provides the possibility to take influence on the I component of the PI controller.

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

1. Specify speed setpoint.
2. Reduce Ti (C00071/2) until the drive is unstable (observe motor noise).
3. Increase Ti (C00071/2) until the drive runs stable again.
4. Increase Ti (C00071/2) to approx. twice the value.

**Tip!**

Value range recommended by Lenze for the setting of the reset time:

Ti = 20 ms ... 150 ms

### Differential time constant Tdn (rate time)

The differential time constant Tdn of the speed controller can be set in C00072.

If the mass inertia of the drive is unknown, the optimisation can be achieved as follows:

▶ Increase Tdn (C00072) during operation until optimal control mode is reached.

#### Using the ramp response for setting the speed controller

If the mechanical components cannot be operated at the stability limit, the ramp response can also be used for setting the speed controller.

**Stop!**

If the controller parameters are preset unfavourably, the control can tend to heavy overshoots up to instability!

- Following and speed errors can adopt very high values.
- If the mechanics are sensitive, the corresponding monitoring functions are to be activated.

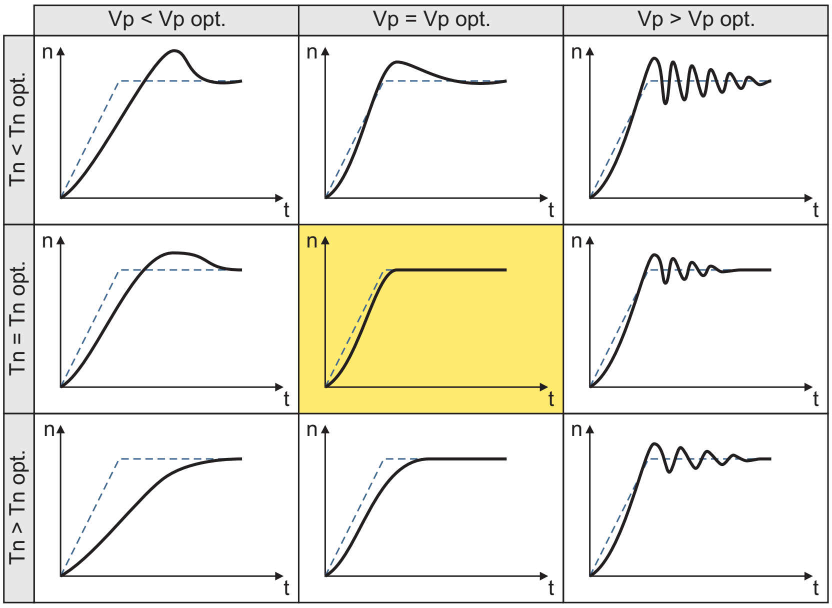
**Note!**

For an optimal setting, we recommend to determine the mass inertia (optimal response to setpoint changes) first.

▸ Optimise response to setpoint changes and determine mass inertia (229)

**How to optimise the speed controller setting by means of the ramp response:**

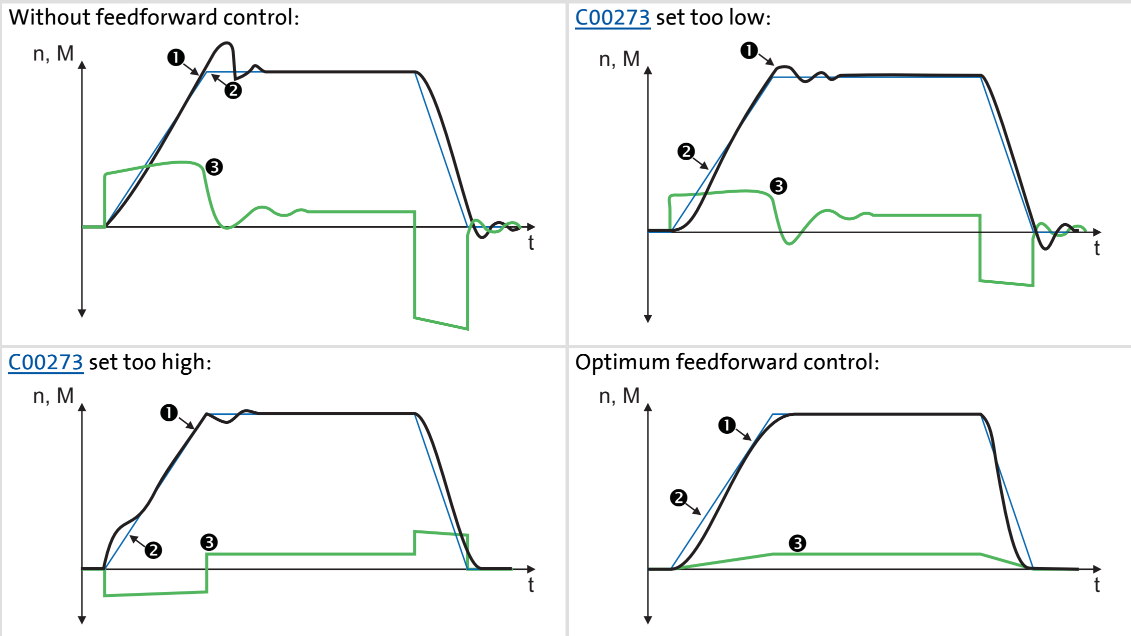
1. Run a typical speed profile and record the ramp response of the speed using the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
2. Evaluate the ramp response:




   - Solid line = ramp response (actual speed value)
   - Dash line = speed setpoint
3. Change gain Vp in C00070/2 and reset time Tn in C00071/2.
4. Repeat steps 1 ... 3 until the optimum ramp response is reached.

### 5.10.4.3 Optimise response to setpoint changes and determine mass inertia

Setting the total moment of inertia under C00273 provides the optimum torque feedforward control. Depending on the application, an adjustment of the setting under C00273 may be necessary to optimise the response to position/speed setpoint changes by means of the torque feedforward control.




❶ Actual speed value (signal *nMotorSpeedAct*)
❷ Speed setpoint (signal *nMotorSpeedSet*)
❸ Speed controller output (signal *nOutputSpeedCtrl*)

[5-17] Typical signal characteristics for different settings of the load moment of inertia

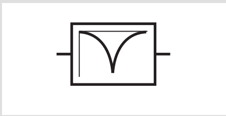
**How to optimise the torque feedforward control:**

1. Run a typical speed profile and record the inputs and outputs of the speed controller with the data logger.
   - Motor control variables to be recorded:
     *nSpeedSetValue_a* (speed setpoint)
     *nMotorSpeedAct_a* (actual speed value)
     *nOutputSpeedCtrl_a* (speed controller output)
2. Estimate the moment of inertia and set it in C00273 in relation to the motor end (i.e. with account being taken of the gearbox factors).
3. Repeat the data logger recording (see step 1).

   Now the data logger should show that part of the required torque is generated by the feedforward control and the speed controller output signal (*nOutputSpeedCtrl_a*) is correspondingly smaller. The resulting following error decreases.
4. Change the setting in C00273 and repeat the data logger recording until the intended response to setpoint changes is reached.
   - The optimisation could aim at the speed controller being completely relieved (see signal characteristics in Fig. [5-17]).
5. Save the parameter set (device command: C00002/11).

### 5.10.4.4 Setting the current setpoint filter (band-stop filter)

Due to the high dynamic performance/limit frequency of the closed current control loop, mechanical natural frequencies can be activated which may lead to an unstable speed control loop.



To mask out or at least damp these resonant frequencies, a so-called current setpoint filter is integrated into the speed control loop of the controller.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00270 | SC: Freq. current setpoint filter | 200.0 | Hz |
| C00271 | SC: Current setpoint filter width | 0.0 | Hz |
| C00272 | SC: Current setpoint filter depth | 0 | dB |

▶ In the default setting of 0 db of the filter depth (C00272), the current setpoint filter is switched off.

#### Setting of the current setpoint filter

Since the frequency response of the speed controlled system is only rarely known to such an extent that the current setpoint filter can be adjusted to the controlled system in the run-up, the following example describes how to set the current setpoint filter.

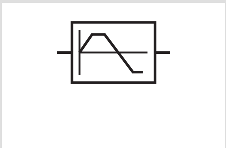
**How to set the current setpoint filter:**

1. Optimise current controller (📖 224).
2. Optimise speed controller (📖 225)
3. Measure the oscillation frequency (observe current or speed).
4. Set the measured oscillation frequency in C00270 as filter frequency.
5. Set "25%" of the filter frequency in C00271 as filter width.
   - Example: Filter frequency = 200 Hz ➞ filter width = 50 Hz.
6. Set "40 dB" in C00272 as filter depth.
   - If the filter depth is set to "0 dB" (default setting), the filter is not active.

**Note!**

Readjust the speed controller after setting the current setpoint filter.
▸ Optimise speed controller. (📖 225)

### 5.10.4.5 Adapting the max. acceleration change (jerk limitation)



Via the max. acceleration change that can be set in C00274, the change of the setpoint torque can be limited for jerk limitation. Hence, sudden torque step changes can be avoided. The entire speed characteristic is smoothed.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00274 | SC: Max. change in acceleration | 400.0 | %/ms |

In the default setting of 400 %/ms of the max. acceleration change (C00274), jerk limitation is switched off.

The setting defines the permissible maximum torque change per ms (based on the rated motor torque).

**Note!**

Only activate this jerk limitation for speed-controlled applications!

If table positioning or a free function block interconnection with a positioning operating mode is selected, jerk limitation must be switched off.

- Here, jerk limitation is provided for in the travel profile generator. Setting this jerk limitation in the motor control would lead to following errors!

### 5.10.4.6 Slip calculation from motor equivalent circuit diagram data

This function extension is available from version 02.00.00!

In order to achieve a better speed stability and torque accuracy, the slip calculation can be either derived from the motor nameplate data (e.g. rated motor speed) or the motor equivalent circuit diagram data (stator resistance, rotor resistance etc.).

The data to be used for servo control (for asynchronous motors) is selected via bit 1 in C02879/1:

| Setting | | Info |
|---|---|---|
| Bit 0 | SLVC | In case of sensorless vector control:<br>• "0" ≡ Slip calculation from motor nameplate data (Lenze setting)<br>• "1" ≡ Slip calculation from motor equivalent circuit diagram data |
| Bit 1 | SC_ASM | In case of servo control for asynchronous motors:<br>• "0" ≡ Slip calculation from motor nameplate data<br>• "1" ≡ Slip calculation from motor equivalent circuit diagram data (Lenze setting) |
| Bit 2 ... 7 | Reserved | |

**Note!**

In order that the slip can be calculated from the motor equivalent circuit diagram data, the equivalent circuit data (stator resistance, rotor resistance etc.) must be known as exactly as possible.

- Selecting a motor in the »Engineer« motor catalogue loads the exact motor equivalent circuit diagram data.
- When the motor nameplate data is entered manually and the motor equivalent circuit diagram data is then detected via the motor parameter identification, the "extended identification" (C02867/1 = 2) must be used.
  ▶ Automatic motor data identification (🕮 126)

### 5.10.4.7 Temperature compensation within the motor control

This function extension is available from version 02.00.00!

A temperature compensation over the detected motor temperature (display in C00063/1) serves to compensate inaccuracies in the output torque within the motor control in case of temperature changes of the asynchronous or synchronous motor.

**Note!**

In the Lenze setting, the temperature compensation within the motor control is activated. The temperature compensation, however, is only active with speed encoder selection "3: Multi encoder" or "4: Resolver" in C00495 as well as error-free KTY temperature detection (display in C00063/1 ≠ 255 °C).

The temperature compensation is switched on/off bit-coded in C02878/1:

| Setting | | Info |
|---|---|---|
| Bit 0 | SC_PSM | KTY motor temperature compensation with servo control for synchronous motors:<br>• "0" ≡ Off<br>• "1" ≡ On (Lenze setting) |
| Bit 1 | SC_ASM | KTY motor temperature compensation with servo control for asynchronous motors:<br>• "0" ≡ Off<br>• "1" ≡ On (Lenze setting) |
| Bit 2 ... 7 | Reserved | |

#### General information on temperature behaviour of the asynchronous and synchronous motor

In case of the **asynchronous motor**, the nameplate data always refer to rated data, i.e. on the rated load and thus on the rated temperature of the motor. For the 8400 device series, the equivalent circuit data for stator resistance (C00084) and rotor resistance (C00082) are always given for a cold asynchronous motor (20 °C) and converted to star connection. In case of a cold motor (20 °C), the real slip frequency thus corresponds to the calculated setpoint slip. In case of a warm motor, the real slip frequency is higher. This becomes especially apparent at high torques as an inaccuracy of the torque over the motor temperature.

In case of the **synchronous motor**. the field strength generated by the permanent magnet in the rotor is temperature-dependent. With a rated motor temperature, a rated field strength is generated. A negative temperature coefficient of -0.11 %/°C means that with a cold motor at the same setpoint torque a higher output torque is output than with a warm motor.

These errors must be compensated by means of the temperature compensation for asynchronous and synchronous motors in order to have a constant output torque over the motor temperature independent of the motor temperature.

#### 5.10.4.8 Optimising the behaviour of the asynchronous motor in the field weakening range

The behaviour of the asynchronous motor in the field weakening range is influenced by:

- the field controller
- the field weakening controller
- the field feedforward control unit
- adaptive adaptation of the P component of the field and speed controllers
- An additional limitation of the I component of the controller when the voltage limit is reached

**Note!**

In general, these settings are pre-optimised so that further optimisation is not required.

Optimisation for special motors (e.g. mid-frequency motors) or for standard asynchronous motors whose power is not adapted can be carried out according to the algorithms shown in the following sections.

### Field controller

The field controller is designed as a PI controller.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00077 | SC: Vp field controller | 12.80 | |
| C00078 | SC: Tn field controller | 256.0 | ms |

The gain Vp (C00077) of the field controller can be calculated using the motor rotor time constant and the equivalent time constant of the current-controlled motor:

$$V_{PFeld} = \frac{T_R[s]}{a^2 \cdot T_{Replacement}[ms]} \quad [\%]$$

$V_{PField}$: Gain of the field controller (C00077)
$T_R$: Motor rotor time constant (C00083)
$T_{Equivalent}$: Motor equivalent time constant (approx. 2.5 ms)
a: Measure for damping (z. B. a = 2)

If the rated data of the motor and the mass inertia of the drive system are known, we recommend the following setting:

$$V_{PFeld} \approx \frac{T_R[s]}{4 \cdot T_{Replacement}[ms]} = \frac{T_R[s]}{0.01[s]}$$

$$T_{NFeld}[ms] = T_R[ms]$$

$V_{PField}$: Gain of the field controller (C00077)
$T_R$: Motor rotor time constant (C00083)
$T_{Equi}$: Motor equivalent time constant
$T_{NField}$: Time constant of the field controller (C00078)

[5-18] Recommendation for the setting of the gain and the time constant of the field controller

**Tip!**

The motor rotor time constant depends on the motor rotor resistance, the magnetising inductance, and the leakage inductance.

- For an optimal calculation, we recommend to select the motor from the motor catalogue of the »Engineer« first if a Lenze motor is used. ▶ Selecting a motor from the motor catalogue in the »Engineer« (📖 124)
- If a third party manufacturer's motor is used, motor parameter identification must be carried out previously. ▶ Automatic motor data identification (📖 126)

### Field weakening controller

The field weakening controller serves to adapt the magnetising current when the maximum control voltage has been reached so that in steady operation approximately 95 % of the maximally possible control voltage is output. Thus, there is a voltage reserve for dynamic load or speed variations.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00577 | SC: Vp field weakening controller | 0.0010 | |
| C00578 | SC: Tn field weakening controller | 20.0 | ms |

$$V_{P,\,FS} \approx \frac{2}{V_{Path,\,FS}}$$

$$V_{Path,\,FS} = \frac{1-\sigma}{\sigma} \cdot \frac{p \cdot 2 \cdot \pi \cdot n_N}{60} \cdot \sigma \cdot (L_h + L_{ss})^2$$

$$\sigma = 1 - \frac{(L_h)^2}{(L_h + L_{ss})^2}$$

$V_{P,FW}$: Gain of the field weakening controller (C00577)
$L_m$: Mutual motor inductance (C00092)
$L_{sl}$: Motor stator leakage inductance (C00085)
p = number of pole pairs

[5-19] Recommendation for setting the gain of the field controller (for Tn ≈ 10 ... 30 ms)

**Tip!**

If the field weakening controller is parameterised unfavourably, vibrations occur in the magnetising current, in the direct-axis current and in the cross current, in the torque and in the speed in the field weakening range. This is also audible by an increased motor noise (humming). The vibrations can be dampened by a decrease of the proportional gain of the field weakening controller.

Procedure:

- Operate the drive with max. required speed in the field weakening operation.
- Reduce Vp (C00577) of the field weakening controller until the vibration response cannot be detected anymore.

### Field feedforward control

To have enough voltage reserve available for dynamic acceleration processes, a timely weakening of the field is required. The field weakening is controlled by the field feedforward control.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00576 | SC: Field feedforward control | 200.0 | % |

The field feedforward control reciprocally reduces the magnetising current from the V/f base frequency (C00015) on. The starting point of the reduction can be shifted to low frequencies via the field feedforward control (C00576). Thus, more voltage reserve is available for acceleration processes.

The field feedforward control (C00576) must be specified in %, based on the rated slip of the machine.

**Tip!**

Generally, the Lenze setting is sufficient for most applications.

- We recommend to increase the field feedforward control for applications with very dynamic acceleration processes in the field weakening range.
- Reduce the field feedforward control for very slow applications if necessary.

### Adaptive adaptation of the P component of the field and speed controllers

In the field weakening range, the properties of the drive change due to the reduction of the magnetisation and the voltage limitation of the inverter. To be able to continue to provide a stable, well dampened drive behaviour, the servo control has an automatic adaptation of the P component of the field weakening controller and speed controller.

In the Lenze setting, this function is activated. Depending on the application, this function can be deactivated via C00079/2.

| Parameter | Info | Lenze setting |
|---|---|---|
| C00079/2 | SC: Adaptive field weakening controller | 1: On |

**Tip!**

Lenze recommends to always activate the adaptation of the field weakening and speed controller.

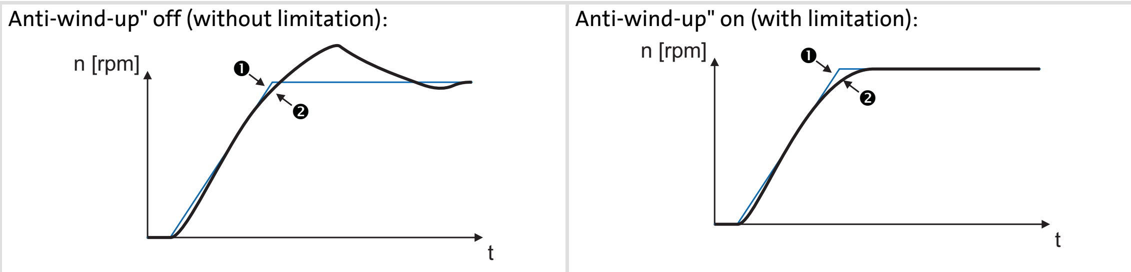
**Additional limitation of the I component of the speed controller when the voltage limit is reached (anti-wind-up)**

In dynamic acceleration processes in the field weakening range, the acceleration torque is limited due to the limitation of the maximum output voltage of the inverter. Acceleration ramps that are set too high (C00012) may lead to undesired charging of the integrator of the speed controller, which result in very strong overshoots of the actual speed value when the setpoint is reached.

The so-called anti-wind-up function serves to intelligently limit the integrator part in dynamic acceleration processes in the field weakening range so that an overshoot of the actual speed value can be prevented when the setpoint speed is reached.

This function is deactivated in the Lenze setting because the robustness of the drive may decrease in the field weakening range. Possible consequences are slight speed variations. If required by the application, the function can be activated via C00079/3.

| Parameter | Info | Lenze setting |
|---|---|---|
| C00079/3 | SC: n-Ctrl Anti-Wind-Up | 0: Off |




❶ Speed setpoint
❷ Actual speed value

[5-20] Typical signal characteristics for switched-on/off anti-wind-up function

**Tip!**

Lenze recommends to activate the additional limitation of the speed controller's I component on reaching the voltage limit (C00079 = 1) if dynamic acceleration processes in the field weakening range are wanted.

The function should be deactivated again if slight speed variations occur in the field weakening range.

## 5.11 Parameterisable additional functions

### 5.11.1 Selection of switching frequency

The switching frequency of the inverter that can be selected in C00018 influences the smooth running performance and the noise generation in the connected motor as well as the power losses in the controller.

The lower the switching frequency the higher the concentricity factor, the smaller the losses, and the higher the noise generation.

**Stop!**

If operated at a switching frequency of 16 kHz, the output current of the controller must not exceed the current limit values specified in the technical data!

▸ Defining current and speed limits (📖 136)

**Note!**

- Operate mid-frequency motors only at a switching frequency of 8 kHz or 16 kHz (var./drive-opt.).
- If operated at a switching frequency of 16 kHz, the Ixt evaluation (C00064) is considered including the required derating to 67 % of the rated device current at switching frequencies of 2.4 and 8 kHz.

#### Settable switching frequencies

| Selection in C00018 | | | |
|---|---|---|---|
| 1 | 4 kHz var./drive-optimised | 21 | 8 kHz var./drive-opt./4 kHz min |
| **2** | **8 kHz var./drive-optimised** | 22 | 16 kHz var./drive-opt./4 kHz min |
| 3 | 16 kHz var./drive-optimised | 23 | 16 kHz var./drive-opt./8 kHz min |
| 5 | 2 kHz constant/drive-optimised | 31 | 8 kHz var./min. Pv/4 kHz min |
| 6 | 4 kHz constant/drive-optimised | 32 | 16 kHz var./min. Pv/4 kHz min |
| 7 | 8 kHz constant/drive-optimised | 33 | 16 kHz var./min. Pv/8 kHz min |
| 8 | 16 kHz constant/drive-optimised | Abbreviations used:<br>• "var.": Adaptation of the switching frequency depending on the current<br>• "drive-opt.": drive-optimised modulation ("sine/delta modulation")<br>• "fixed": fixed switching frequencies<br>• "min. Pv": additional reduction of power loss | |
| 11 | 4 kHz var./min. Pv | | |
| 12 | 8 kHz var./min. Pv | | |
| 13 | 16 kHz var./min. Pv | | |
| 15 | 2 kHz constant/min. Pv | | |
| 16 | 4 kHz constant/min. Pv | | |
| 17 | 8 kHz constant/min. Pv | | |
| 18 | 16 kHz constant/min. Pv | | |

**Tip!**

The Lenze setting C00018 = 2 (8 kHz var./drive-opt.) is the optimal value for standard applications.

### Lowering the switching frequency due to high heatsink temperatures

Exceeding the maximally permissible heatsink temperature would lead to an inhibited drive due to the "Overtemperature" error and a torquelessly coasting motor. Therefore, if the Lenze setting is selected, the switching frequency is reduced to the next frequency below when the heatsink temperature has risen to 5 °C below the maximally permissible temperature. After the heatsink has cooled down, the controller automatically switches to the next frequency above until the set switching frequency is reached.

Switching frequency reduction due to high heatsink temperature can be deactivated via C00144. If the switching frequency reduction is deactivated, the "OH1: Heatsink overtemperature" error message will be issued when the maximally permissible heatsink temperature is reached. An "Error" response is the result and the motor is coasting.

| Parameter | Info | Lenze setting |
|---|---|---|
| C00144 | Switching frequency reduction (temp.) | 1: On |

### Lowering of the switching frequency depending on the output current

"Variable" switching frequencies can be selected for the controller in C00018, where the controller automatically lowers the switching frequency depending on the controller output current. The modulation mode will not be changed.

The switching thresholds are given in the rated data in the **8400 hardware manual**.

- The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 drive controller.

When a "fixed" switching frequency is selected, no switching frequency changeover takes place. In case of fixed frequencies, the controller output current is limited to the permissible value of the corresponding switching frequency. In case of larger load impulses, the overcurrent interruption may be activated, to which the controller responds with "Error".

### Limiting the maximum output frequency

**Note!**

If the servo control mode (SC) is selected, the drive assumes the 'maximum current limitation' state if the maximum output frequency (C00910) has been reached.

▶ Defining current and speed limits (📖 136)

The maximum output frequency (C00910) of the controller is not limited depending on the switching frequency. Therefore, adapt the maximum output frequency according to our recommendation:

$$\text{Maximum output frequency} \leq \frac{1}{8}\text{Switching frequency}$$

- At a switching frequency of 4 kHz, for instance, 500 Hz for the maximum output frequency should not be exceeded.

Carry out further measures:

- If required, deactivate the switching frequency changeover by the heatsink temperature via C00144.
- If required, ensure that the changeover threshold of the controller output current to the next switching frequency below will not be exceeded. If required, select a constant switching frequency in C00018.

### Display of the current switching frequency

The current switching frequency applied in the controller is displayed in C00725.

### Operation at an ambient temperature of 45°C

The controller is designed so that operation at an ambient temperature of 45° C without derating is permissible at a switching frequency of 4 kHz.

### 5.11.2 Operation with increased rated power

Under the operating conditions described here and under continuous operation, the controller can be operated with a higher power motor (increased rated power). The remaining overload capacity of the drive system (for 60 s/3 s) is reduced accordingly to approx. 120 %/160 %.

Typical applications stand out due to low dynamic requirements, e.g. pumps and fans, general horizontal materials handling technology and line drives.

The controllers that can be operated at increased rated power are listed in the rated data in the **8400 hardware manual**.

- The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 drive controller.

**Stop!**

Operation at increased rated power is only permitted ...

- with the controllers listed in the **8400 hardware manual** for this type of operation in the stated mains voltage range.
- at switching frequencies of 2 kHz and 4 kHz.
- at a max. ambient temperature of 40 °C.
- with the types of installation stated in the **8400 hardware manual**.
- with the fuses, cable cross-sections, mains chokes, and filters as required in the **8400 hardware manual** for this operation.
- after parameterisation according to the specifications below.

#### Required parameterisation

Operation at increased rated power requires the following settings to be made particularly for the V/f characteristic control (VFCplus), but also for all other types of control:

| Parameter | Info | Required setting |
|---|---|---|
| C00016 | VFC: Vmin boost | adapt to motor (reduce) |
| C00018 | Switching frequency | 1: 4 kHz var./drive-opt. |
| C00021 | Slip compensation | adapt to motor |
| C00120 | Setting of motor overload ($I^2xt$) | adapt to motor |
| C00123 | Device utilisat. threshold (Ixt) | 120 % |
| C00173 | Mains voltage | see hardware manual → Rated data |

All other types of control require the following settings in addition:

| Parameter | Info | Required setting |
|---|---|---|
| C00022 | Imax in motor mode | higher than rated motor current (max. 160 % rated motor current) |
| C00081 | Rated motor power | adapt motor data (see motor nameplate), then carry out identification run ▸ Automatic motor data identification (📖 126) |
| C00087 | Rated motor speed | |
| C00088 | Rated motor current | |
| C00089 | Rated motor frequency | |
| C00090 | Rated motor voltage | |
| C00091 | Motor cos φ | |

### 5.11.3 Correction of the stator leakage inductance...

**...and the current controller parameters by means of the saturation characteristic**

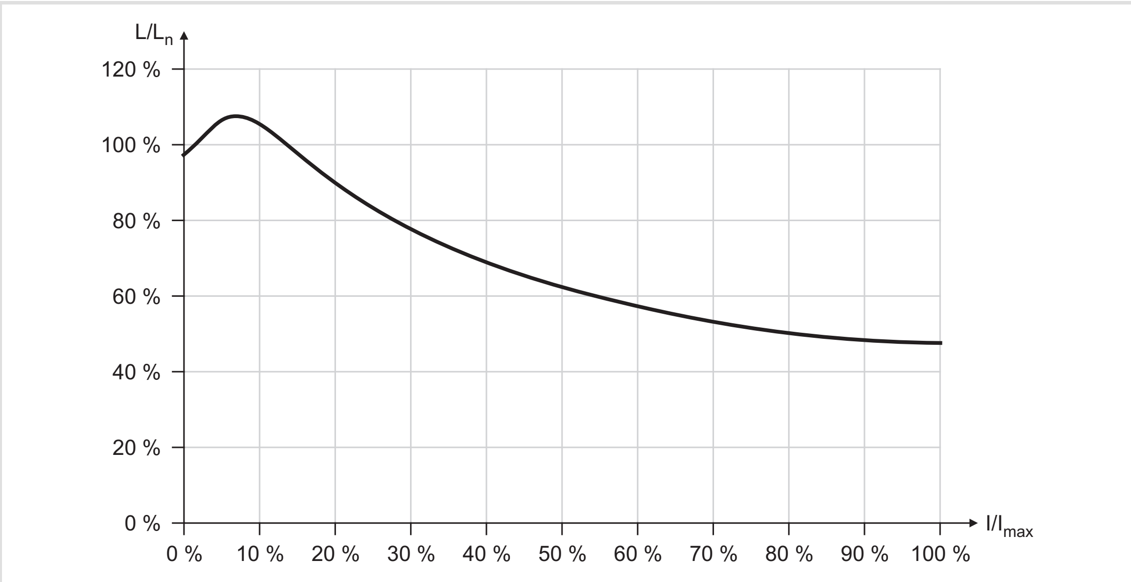
**Note!**

Function only possible with:
- Servo control (SC) for synchronous motors
- Sensorless control for synchronous motors (SLPSM)

The current controller must be adjusted to the electrical characteristics of the motor stator resistance (C00084) and stator leakage inductance (C00085). In case of modern motors, the stator leakage inductance changes with the height of the current so that a new current controller setting is required for each current height.

When the motor is operated with very low and very high currents (e.g. in *Pick and place* applications), it is not always possible to achieve a satisfactory current controller setting for all operating points. For this purpose, the correction of the stator leakage inductance and current controller parameters is now possible via an adjustable saturation characteristic that can be set in C02853 (17 interpolation points).

The following picture shows a typical saturation characteristic of an MCS motor:




[5-21] Saturation characteristic: Inductance referring to the inductance for rated current

- When a Lenze motor is selected from the »Engineer« motor catalogue, the saturation characteristic will also be loaded and switched on if required.
- The correction by means of this saturation characteristic can be switched on/off via C02859.
- If instabilities of the current controller occur when a third-party motor is used at high currents, ask the motor manufacturer whether the stator leakage inductance changes with the current height. If required, the saturation characteristic of this motor must be set and switched on.

**Note!**

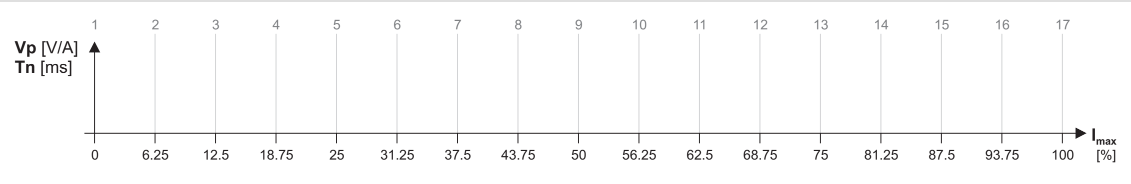
The saturation characteristic is not only used for the correction of the current controller but also influences the current controller feedforward control (C00079/1).

### Short overview of the relevant parameters:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02853/1...17 | Lss saturation characteristic | 100 | % |
| C02855 | Imax Lss saturation characteristic | 3000.0 | A |
| C02859 | Activate Lss saturation charact. | 0: Off | |

### Distribution of the interpolation points

- The saturation characteristic is defined by 17 interpolation points which are distributed linearly on the x axis.
- Interpolation point 17 represents 100 % of the maximum motor current in the process (C02855).




[5-22] Saturation characteristic: Distribution of the interpolation points

### 5.11.4 Flying restart function

The flying restart circuit for asynchronous motors uses a simple motor model which requires knowledge of the motor stator resistance RS and the rated motor current.

**Note!**

- Currently, the flying restart circuit is only available for asynchronous motors. (A flying restart circuit for synchronous motors is in preparation.)
- For a correct functioning of the flying restart circuit, we recommend to perform a parameter identification first. ▶ Automatic motor data identification (📖 126)
- The flying restart function works safely and reliably for drives with great centrifugal masses.
- Do not use the flying restart function if several motors with different centrifugal masses are connected to a controller.
- After the controller is enabled, the motor can start for a short time or reverse when machines with low friction and low mass inertia are used.
- The flying restart function serves to identify max. field frequencies up to ±200 Hz.
- When power-adapted standard asynchronous motors are used (rated motor power approximately corresponds to the rated inverter power), a motor parameter identification is not required.
- On drive systems with feedback, you do not need to use the flying restart function because the synchronisation to the speed detected by the feedback is always carried out in a jerk-free manner.

**Tip!**

In association with the flying restart function, we recommend information provided in this documentation on the following topic:
▶ Automatic DC-injection braking (Auto-DCB) (📖 251)

#### General information

This function serves to activate a mode which is used to "catch" a coasting motor during operation without speed feedback. This means that the synchronicity between controller and motor is to be adjusted in such a way that a jerk-free transition to the rotating machines is achieved in the instant of connection.

The drive controller determines the synchronicity by identifying the synchronous field frequency.

### Duration

The "catching" process is completed after approx. 0.5 ... 1.5 seconds. The duration is influenced by the starting value. If the field frequency is not known, we recommend a fixed starting value of 10 Hz (or -10 Hz with systems rotating in negative direction).

### Short overview of the relevant parameters:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00990 | Flying restart fct.: Activate | Off | |
| C00991 | Flying restart fct.: Process | -n...+n \| Start: +10 Hz | |
| C00992 | Flying restart fct.: Start frequency | 5 | Hz |
| C00993 | Flying restart fct: Int. time | 300 | ms |
| C00994 | Flying restart fct.: Current | 25.00 | % |

### How to parameterise the flying restart function:

1. Activate the flying restart circuit by selecting "1: On" in C00990.
   - Every time the controller is enabled, a synchronisation to the rotating or standing drive is carried out.

   When the Lenze setting is used, most applications do not require additional controller settings.

   If additional settings are necessary, proceed as follows:
2. Define the process and hence the speed range/rotational frequency range in C00991 which is to be examined by the flying restart circuit:
   - positive speed range (n ≥ 0 rpm$^{-}$)
   - negative speed range (n ≤ 0 rpm$^{-}$)
   - total speed range
3. Define the starting frequency.

   The starting frequency which defines the starting point of the flying restart function is 10 or -10 Hz for processes 0 ... 3 and has been pre-optimised for standard motors.

   If process 4 is selected in C00991, an arbitrary starting frequency can be defined via C00992. This is especially recommended for motors with higher rated frequencies.
   - We recommend to define a starting frequency of approximately 20 % of the rated motor frequency to enable a safe and fast connection to standing drive systems.
   - For systems with a known search speed (e.g. torque-controlled drive systems which are to synchronise to a defined speed) the starting value can be adapted to reduce the flying restart time.

4. Set the flying restart current in C00994.

   We recommend setting a flying restart current of 10 % ... 25 % of the rated motor current.
   - During a flying restart process, a current is injected into the motor to identify the speed.
   - Reducing the current causes a reduction of the motor torque during the flying restart process. A short-time starting action or reversing of the motor is prevented with low flying restart currents.
   - An increase of the current improves the robustness of the flying restart function.

**Tip!**

**Use of motors with higher rated frequencies**

For trouble-free operation, we recommend to manually enter a starting frequency of 20 % of the rated motor frequency in C00992 as well as to accelerate the flying restart process (see above) and to use a lower flying restart current (10 % of the rated motor current) if motors with higher rated frequencies are used.

**Optimisation of the flying restart time**

The duration of the flying restart process can be influenced via the setting of the integration time (C00993). A reduction of the integration time causes an accelerated flying restart function and thus a reduced flying restart time.
- We recommend not to change the Lenze setting of the integration time.
- When special motor are used (e.g. multi-pole motors or ASM servo motors), a reduced integration time may improve the flying restart behaviour.

**Optimising the current controller if the behaviour is unstable**

During the execution of flying restart function, peak currents/torques are avoided by controlling the current amplitude.

Gain (C00075) and reset time (C00076) of the current controller can be adapted to improve the jerk-free/torque-free connection of the inverter to the supply of the rotating motor.
- We recommend not to change the Lenze setting of the current controller.
- If the behaviour of the current controller is unstable, gain and reset time can be calculated as per the following formulae:

$$V_p = \frac{L_{ss}[H]}{T_E[s]}$$

$$T_i = \frac{L_{ss}[H]}{R_s[\Omega]}$$

$V_p$ = Current controller gain (C00075)
$T_i$ = Current controller reset time (C00076)
$L_{ss}$ = Motor stator leakage inductance (C00085)
$R_s$ = Motor stator resistance (C00084)
$T_E$ = Equivalent time constant (= 500 µs)

[5-23] Formulae for the calculation of the gain and reset time of the current controller

### 5.11.5 DC-injection braking

**Danger!**

The DC-injection braking or auto DC-injection braking function cannot be used with sensorless control for synchronous motors (SLPSM) and servo control (SC) mode.

Holding braking is not possible when this braking mode is used!

- For low-wear control of a holding brake, use the basic function "Holding brake control". (563)

DC-injection braking allows the drive to be quickly braked to a standstill without the need to use an external brake resistor.

- The braking current is set in C00036.
- The maximum braking torque to be generated by the DC braking current is approx. 20 ... 30 % of the rated motor torque. It is lower than that for braking in generator mode with an external brake resistor.
- Automatic DC-injection braking (auto DCB) improves the starting performance of the motor when operated without speed feedback.

**Tip!**

DC-injection braking has the advantage that it is possible to influence the braking time by changing the motor current or the braking torque..

**Short overview of the relevant parameters:**

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00019 | Auto DCB: Threshold<br>• Operating threshold for activating DC-injection braking | 3 | rpm |
| C00036 | DC braking: Current<br>• Braking current in [%] based on rated device current (C00098) | 50 | % |
| C00106 | Auto DCB: Hold time | 0.500 | s |
| C00107 | DC braking: Hold time | 999.000 | s |
| C00701/4 | LA_NCtrl: bSetDCBrake<br>• Selection of the signal source for activating DC-injection braking | Dependent on the selected control mode | |

**Method**

DC-injection braking can be carried out in two ways with different types of activation:

- Manual DC-injection braking (DCB) (📖 251)
- Automatic DC-injection braking (Auto-DCB) (📖 251)

### 5.11.5.1 Manual DC-injection braking (DCB)

DC-injection braking can be activated manually for the two technology applications "Actuating drive speed" and "Switch-off positioning" by connecting the *bSetDCBrake* input of the **LA_NCtrl** or **LA_SwitchPos** application block to a digital signal source (e.g. via the digital signal source *bCtrl1_B3* of the port block LP_CANIn1).

- For HIGH-active inputs, DC-injection braking is active as long as the signal is at HIGH level.
- After the hold time (C00107) has expired, the controller sets the pulse inhibit (CINH).

### 5.11.5.2 Automatic DC-injection braking (Auto-DCB)

"Automatic DC-injection braking" (referred to in the following as "auto DCB") can be used if there is a requirement that the drive be isolated from the supply at n ≈ 0.

**Note!**

**Deactivate automatic DC-injection braking when a holding brake is used!**

- For this purpose, go to C00019 and set the auto DCB threshold to "0".
- Background: Controller inhibit is already activated by the Holding brake control. (📖 563)

**Function**

For understanding the auto DCB function, it is necessary to distinguish between three different types of operation:

A. The drive has been enabled and, in the course of operation, the speed setpoint falls below the auto DCB threshold.
   - In case of operation without speed feedback, a braking current (C00036) is injected. After the auto DCB hold time (C00106) has expired, the motor is deenergised via the auto DCB function, i.e. a controller inhibit (CINH) is set.
   - In case of operation with speed feedback, the motor is deenergised via the auto DCB function after the auto DCB hold time (C00106) has expired, i.e. a controller inhibit (CINH) is set.
     The braking current which can be parameterised in C00036 does not have any effect during operation with speed feedback.

B. When the controller is enabled, the drive is at standstill (n = 0).
   If the enabled drive is to start, the speed setpoint passed via the acceleration ramp must exceed the auto DCB threshold (C00019). Below this threshold, the motor will not be energised.

C. When the controller is enabled, the motor (still) rotates at a speed which is above the auto DCB threshold. If the speed setpoint reached via the acceleration ramp exceeds the auto DCB threshold (C00019), the motor will be energised and the following action will take place:
   – During operation without speed feedback, the drive is "caught".
     ▶ Flying restart function (📖 247)
   – During operation with speed feedback, the drive synchronises to the current actual speed value.

**Tip!**

We recommend to deactivate the auto DCB function during operation with speed feedback via a setting of C00019 = 0.

### Auto DCB function during operation with speed feedback

**Stop!**

If the DC-injection braking operation is too long and the braking current or braking voltage is too high, the connected motor may overheat.

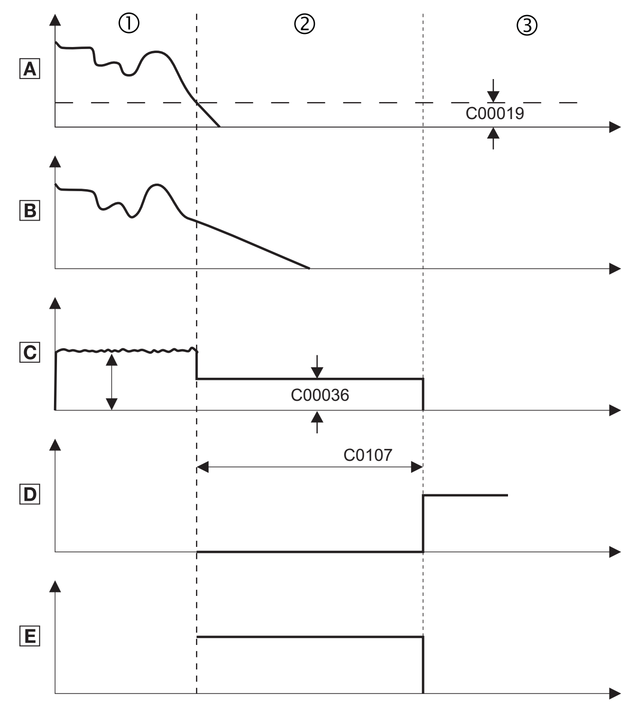
If you want to use the auto DCB function contrary to our recommendation (see above), the auto DCB threshold must not fall below the following values depending on the number of encoder increments (C00420):

| Number of encoder increments C00420 | Auto DCB threshold C00019 |
|---|---|
| 8 | 16 |
| 16 | 8 |
| 32 | 4 |
| 64 | 2 |
| > 128 | No restrictions |

### How to set the automatic DC-injection braking

1. Set a hold time in C00106 > 0 s.
   - Automatic DC-injection braking is active for the time set.
   - In case of operation without speed feedback, the braking current set in C00036 is injected.
   - After the set hold time has expired, the controller sets a pulse inhibit.
2. Set the operating threshold in C00019.
   - The operating threshold can serve to set a dead band in the setpoint. If DC-injection braking is not to be active then, C00106 must be set to a value of "0".

#### Explanation of the automatic DC-injection braking function by means of two examples



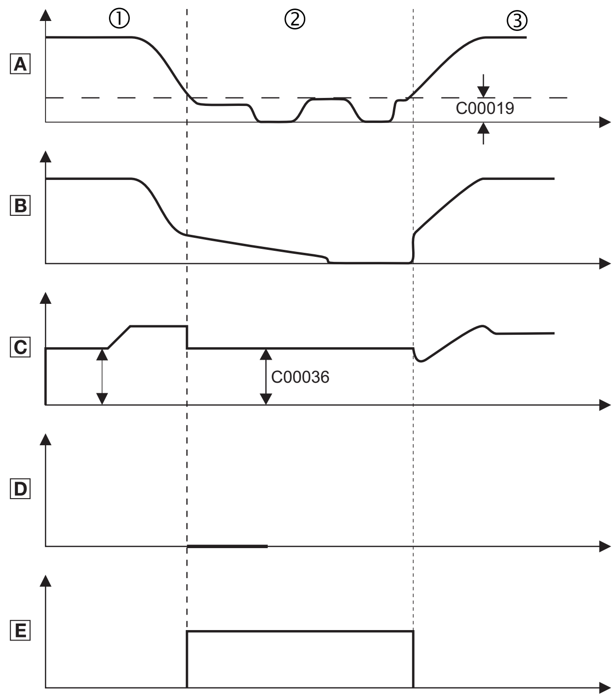

Ⓐ Speed setpoint
Ⓑ Actual speed value of the motor
Ⓒ Output current of the controller
Ⓓ Pulse inhibit
Ⓔ DC-injection braking is active

① The motor rotates at a specified speed. The current adjusts itself to the load, see Ⓒ.

② The DC braking current set in C00036 is injected.

③ After the hold time (C00106) has expired, a pulse inhibit is set.

[5-24] Example 1: Signal characteristic for automatic DC-injection braking of a drive without speed feedback




A Speed setpoint
B Actual speed value of the motor
C Output current of the controller
D Pulse inhibit
E DC-injection braking is active

① The motor rotates at the selected speed. The resulting current depends on the load, see F.

② The DC braking current set in C00036 is injected.

③ The actual speed value of the motor follows the speed setpoint. The resulting current depends on the load.

[5-25] Example 2: Signal characteristic for automatic DC-injection braking of a drive with speed feedback

### 5.11.6 Slip compensation

**Note!**

Slip compensation is only active with the following motor control modes:
- V/f characteristic control (VFCplus) (📖 147)
- Sensorless vector control (SLVC) (📖 182)

Under load, the speed of an asynchronous motor decreases. This load-dependent speed drop is called slip. The slip can partly be compensated for by the setting in C00021.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00021 | Slip compensation | 2.11 | % |

- The setting of C00021 can be done automatically in the course of motor parameter identification. ▸ Automatic motor data identification (📖 126)
- The setting must be made manually if the motor parameter identification cannot be called up.

**How to set the slip compensation manually:**

1. Set rated motor current (C00088) and rated motor frequency (C00089).
2. Calculate the slip compensation according to motor nameplate data:

$$s = \frac{n_{rsyn} - n_r}{n_{rsyn}} \cdot 100\%$$

$$n_{rsyn} = \frac{f_r \cdot 60}{p}$$

$s$ Slip constant (C00021) [%]
$n_{rsyn}$ Synchronous motor speed [rpm]
$n_r$ Rated motor speed according to the motor nameplate [rpm]
$f_r$ Rated motor frequency according to the motor nameplate [Hz]
$p$ Number of motor pole pairs (1, 2, 3 ...)

3. Transfer the calculated slip constant s to C00021.
4. Correct the setting in C00021 while the drive is running until the load-dependent speed drop does not occur anymore between idling and maximum load of the motor in the desired speed range.

**Tip!**

The following guide value applies to a correctly set slip compensation:

- Deviation from the rated motor speed ≤ 1% for the speed range of 10 % ... 100 % of the rated motor speed and loads ≤ rated motor torque.
- Greater deviations are possible in the field weakening range.
- If C00021 is set too high, the drive may get unstable.
- Negative slip (C00021 < 0) with V/f characteristic control results in "smoother" drive behaviour at heavy load impulses or applications requiring a significant speed drop under load.

### 5.11.7 Oscillation damping

Mechanical oscillations are undesirable effects in every process and they may have an adverse effect on the single system components and/or the production output.

Mechanical oscillations in the form of speed oscillations are suppressed by the oscillation damping function.

Mechanical oscillations may occur:

- In the voltage range (output voltage is lower than max. voltage)
  - Here, the oscillations occur in no-load operation.
  - Here, speeds of 40 ... 80 % of the rated speed are typical.
  - See subchapter "Oscillation damping voltage range". (📖 257)
- In the field weakening range (output voltage has reached maximum voltage)
  - Here, the oscillations occur in no-load operation and with load.
  - Here, speeds higher than the rated speed are typical, especially when the output frequency is close to the mains frequency.
  - See subchapter "Oscillation damping in the field weakening range". (📖 258)

**Note!**

With servo control (SC), the Oscillation damping voltage range has no influence.

Mechanical natural frequencies can be suppressed or at least dampened in the speed control loop of the servo control by means of a current setpoint filter. ▶ Setting the current setpoint filter (band-stop filter) (📖 231)

Oscillation dampening, especially in the field weakening range, is also possible by activating the current controller feedforward control (C00079/1).

### 5.11.7.1 Oscillation damping voltage range

The oscillation damping voltage range is successfully used with

- unloaded motors (no-load oscillations)
- motors whose rated power deviates from the rated power of the controller.
  - e.g. during operation at high switching frequency including the power derating involved.
- operation with higher-pole motors
- operation with special motors
- compensation of resonance in the drive
  - At an output frequency of approx. 20 ... 40 Hz, some asynchronous motors can show resonance which causes current and speed variations and thus destabilise the running operation.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00234 | Oscillation damping influence | 5.00 | % |
| C00235 | Oscillation damping filter time | 32 | ms |

**Note!**

Compensate the resonance during operation with feedback (closed loop, feedback of $n_{act}$) via the parameters of the slip regulator.

▶ Parameterising the slip regulator (📖 179)

**How to eliminate speed oscillations in no-load operation at speeds with 40 ... 80 % of the rated speed:**

1. Approach the area where the speed oscillations occur.
2. Reduce the speed oscillations by changing C00234 step by step (increment 1 %).
   - The filter time oscillation damping (C00235) should not be changed.
3. These can be indicators for smooth running:
   - Constant motor current characteristic
   - Reduction of the mechanical oscillations in the bearing seat

### 5.11.7.2 Oscillation damping in the field weakening range

When the max. possible output voltage (full modulation) has been reached, a voltage dip in the DC bus causes a voltage fluctuation in the motor. With load and during no-load operation this voltage fluctuation can cause mechanical oscillations.

The "oscillation damping field weakening" adjustable in C00236 serves to limit the maximum output voltage. This can be used to always compensate voltage dips in the DC bus to the output voltage (constant output voltage). This serves to prevent mechanical oscillations due to these voltage dips.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00236 | Oscillation damping field weakening<br>• Setting "0" ≡ 100 % output voltage can be reached | 14 | |

- With the Lenze setting of C00236 the limitation of the output voltage is set so that voltage dips in the DC bus in the output voltage for the single-phase and three-phase devices can largely be compensated so that no speed oscillations may be expected. Thus, an adaptation of C00236 is not required in the majority of cases.
- Maximum output voltage to be reached with Lenze setting of C00236:
  - Single-phase devices: 98.2 %
  - Three-phase devices: 99.7 %

**Note!**

The limitation of the output voltage via C00236 in the extreme field weakening range (high speeds) causes a reduction of the max. possible output torque (stalling torque).

- If the output torque to be reached in the extreme field weakening range is not sufficient (motor is stalling too early), reduce the setting in C00236.

With servo control (SC), the Lenze setting of C00236 should not be reduced. Otherwise the field weakening control could not work optimally anymore (the behaviour at high speeds may get worse). When servo control (SC) has been selected, the oscillation damping in C00236 should never be set to "0".

**How to eliminate speed oscillations in the field weakening range:**

1. Approach the area where the speed oscillations occur.
2. Reduce the speed oscillations by changing C00236 step by step (increment 1).
3. These can be indicators for smooth running:
   - Constant motor current characteristic
   - Reduction of the mechanical oscillations in the bearing seat

### 5.11.8 Phase sequence reversal for correcting misconnected UVW motor phases

If the motor phases are misconnected at the inverter output (e.g. phase u takes the place of phase v), the motor will rotate in the wrong direction.

To correct such misconnected motor phases, the rotating field of the controller's output can be reversed by selecting "1: Inverted" in C00905. In this case, a phase will be reversed at the output of the inverter.

This function does not have any effect on setpoints and actual values, i.e. the polarity of the speed setpoint/actual speed value, actual torque, output frequency, and AngleOffset do not change.

**Tip!**

Cases of application for this function:

- Phase sequence reversal in case of misconnected motor phases.
- Setting of the correctly signed direction of rotation for inversely mounted motors.

### 5.11.9 Field weakening for synchronous motors

This function extension is available from version 02.00.00!

**Note!**

Function only possible with:

- Servo control (SC)
- Sensorless control for synchronous motors (SLPSM) (from version 10.00.00)

The energy efficiency of a synchronous motor decreases with the field weakening operation. If a high energy efficiency is required, keep the field weakening switched off or restrict the field weakening operation via C00938.

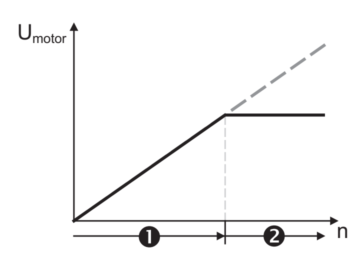
**Stop!**

In the field weakening operation, a current is injected into the synchronous motor even in idle state which can rise to maximum current (C00022).

Ensure that this no-load current does not cause the motor to be heated impermissibly!

- The use of a temperature feedback via KTY is recommended. ▸ Motor temperature monitoring (KTY) (📖 323)

If required, field weakening for synchronous motors can be switched on in C00079/4.



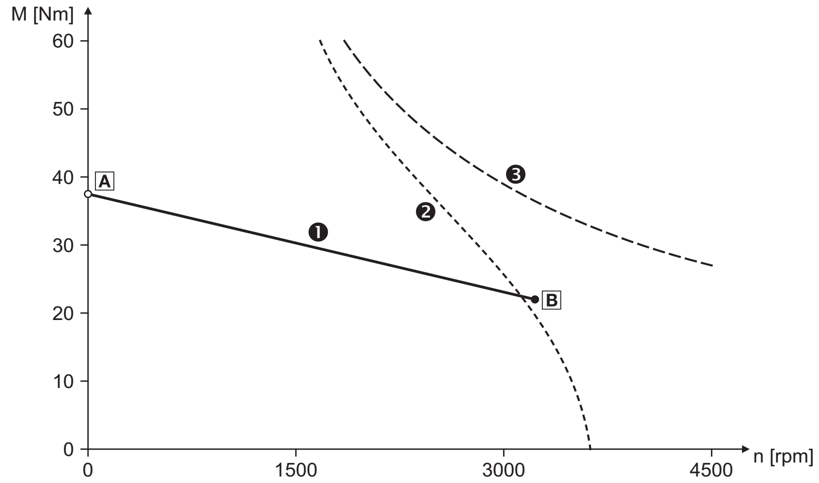

❶ Armature control range
❷ Field weakening range

- ▶ When field weakening is switched on, the motor magnetising current is increased from 0 A to the maximally effective magnetising current via an internal control loop when the voltage limit is reached.
- ▶ As a result, a higher speed can be reached at the same motor voltage or DC-bus voltage.

[5-26] Voltage/speed characteristic with switched-on field weakening

$$n_{max} = n_{nenn\_mot} \cdot \frac{800V}{\sqrt{2} \cdot U_{nenn\_mot}}$$

[5-27] Calculation of the maximally reachable speed with switched-on field weakening




A Standstill torque
B Rated torque
❶ Characteristic for continuous operation
❷ Limit characteristic at a mains voltage of 400 V without field weakening
❸ Limit characteristic with field weakening

[5-28] Speed/torque characteristics of a synchronous servo motor with field weakening

### Short overview of the relevant parameters:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00079/4 | Field weakening | 0: Off | |
| C00938 | Limitation of maximally effective field-producing motor current<br>• With regard to rated motor current (C00088) | 100 | % |
| C00937/1 | Maximally effective field-producing motor current | - | A |
| Highlighted in grey = display parameter | | | |

- The maximally effective field-producing motor current is calculated based on the motor data set in C00085, C00089 and C00098. Then, the value is internally limited to 98 % of the set maximum current (C00022 or maximally permissible current for the permanent switching frequency set in C00018).
- C00938 serves to limit the maximally effective field-producing motor current as well.
  - In the Lenze setting, the field-producing motor current is limited to a rated motor current (C00088) of 100 %. Hence, the maximum speed in the field weakening operation is limited and at the same time heating of the motor in the field weakening operation and idle state is limited.
  - If a higher speed for the field weakening operation is required or the current in the field weakening operation is to be limited (e.g. since no motor temperature detection is available and/or heating in the field weakening operation is to be limited), the value must be increased or reduced accordingly in C00938.

- In C000937/1, the actually used maximally effective field-producing motor current is displayed.
  - With switched-on and active field weakening: 0.00 A ... -x.xx A
  - With sensorless control for synchronous motors (SLPSM), the injected current is displayed in open-loop controlled operation: 0.00 A ... +x.xx A
  - If neither field weakening nor open-loop controlled operation are active,e "0.00 A" is displayed.

### Note!

**If a Lenze motor is used:**

The controller is automatically parameterised so that field weakening operates optimally and the maximally permissible speed is monitored.

### Stop!

**If an OEM motor is used:**

If pulse inhibit is set in the controller, the DC bus is loaded with the voltage that corresponds to the current speed of the machine.

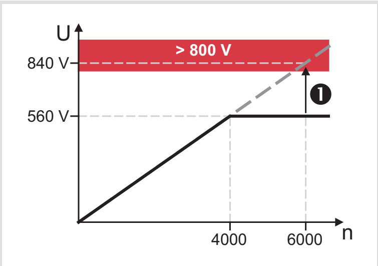
Since with switched-on field weakening higher speeds can be achieved at a correspondingly higher rotor voltage of the motor, the DC bus can be loaded to a voltage higher than the set DC-bus voltage in case of pulse inhibit and a currently high motor speed and even exceed the maximally permissible voltage of 800 V!

For device protection, either use a brake chopper or parameterise the motor speed monitoring via C00965 in such a way that only a maximum speed is possible which would be also reachable without field weakening with a DC-bus voltage of = 800 V. ▸ Motor speed monitoring (📖 284)

### Example: Voltage increase in the DC bus when field weakening is switched off

(For instance by an active setting of the controller inhibit or by tripping a fault or error at high motor speed.)

| Field weakening | Speed n | Motor voltage peak value |
|---|---|---|
| Switched off | 4000 rpm | 560 V |
| | 5700 rpm | 800 V |
| | 6000 rpm | 840 V |
| Switched-on | 6000 rpm | 560 V |




- If pulse inhibit occurs at 6000 rpm and switched-on field weakening, the DC bus is loaded to more than 800 V (❶).
- A speed limitation to 5700 rpm is required since this speed causes a DC-bus voltage of 800 V if field weakening is switched off.

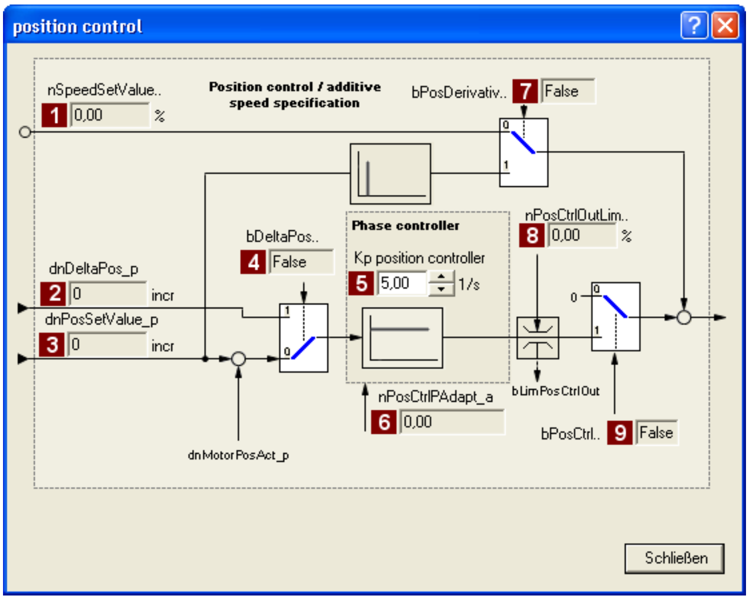
[5-29] Example: Possible DC-bus voltage > 800 V if field weakening gets lost

## 5.12 Position control/additive speed specification

In the Lenze setting, the position control is only active with TA "Table positioning".

**Proceed as follows to open the dialog for parameterising the position control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Go to the *Overview* dialog level and click the **Motor control...** button to change to the *Overview* → *motor control...* dialog level.
4. Click the **Position control** button in the displayed signal flow.




| | Parameter | Info | |
|---|---|---|---|
| 1 | C00830/22 | MCTRL: nSpeedSetValue_a | Speed setpoint |
| 2 | C00834/4 | MCTRL: dnDeltaPos_p | Position difference (following error input) |
| 3 | C00834/5 | MCTRL: dnPosSetValue_p | Absolute position setpoint |
| 4 | C00833/35 | MCTRL: bDeltaPosOn | TRUE = Position difference is active as setpoint selection |
| 5 | C00254 | Kp position controller | Gain for following error compensation |
| 6 | C00830/20 | MCTRL: nPosCtrlPAdapt_a | Adaptation of the position controller gain |
| 7 | C00833/67 | MCTRL: bPosDerivativeOn | TRUE = Setpoint for the speed controller is created from the position setpoint |
| 8 | C00830/21 | MCTRL: nPosCtrlOutLimit_a | Limitation of the position controller output |
| 9 | C00833/27 | MCTRL: bPosCtrlOn | TRUE = Position/angle control active |

## 5.13 Braking operation/brake energy management

When electric motors are braked, the kinetic energy of the drive train is fed back into the DC circuit regeneratively. This energy leads to an increase in the DC bus voltage. In order to avoid overvoltage in the DC bus, several different strategies can be used:

- Use of a brake resistor
- Stopping of the ramp function generator if brake chopper threshold exceeded (RFG_Stop)
- Use of the "Inverter motor brake" function
- Combination of the above named options

In the case of inverters with a 3-phase supply, the following is also possible:

- Coupling of the inverters in a DC-bus connection
- Recovery of regenerative energy with a regenerative module

**Stop!**

If the connected brake resistor is smaller than required, the brake chopper can be destroyed!

Appropriate protective measures are provided in the subchapter "Avoiding thermal overload of the brake resistor". (📖 273)

**Note!**

- We recommend to use the brake chopper (brake transistor) which is integrated into the controller for the braking operation, regardless of the selected motor mode.
  - Connect the required brake resistor to the $R_{B1}$ and $R_{B2}$ terminals of the controller.
- For a DC-bus connection with other devices, we recommend to connect the regenerative power supply module to terminals +UG and –UG.
- If none of these measures is taken, e.g. the overvoltage deactivation ("OU") may respond in case of low deceleration times during regenerative operation.
  ▸ Error messages of the operating system (📖 602)

To install the regenerative module, follow the instructions in the **8400 hardware manual**.

- The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 drive controller.

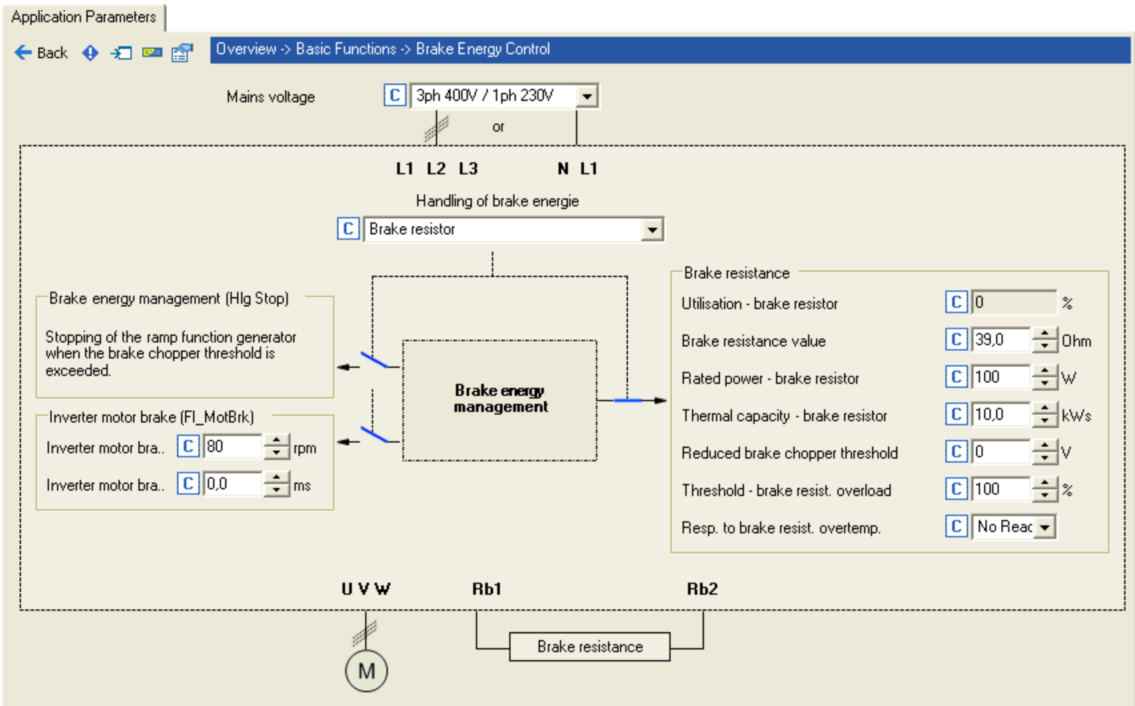
**Tip!**

In C00175, a ramp function generator stop (FB L_NSet_1) can be set for instances when the brake resistor is controlled. This prevents overvoltage deactivation in the case of short deceleration times.

▸ Selecting the response to an increase of the DC-bus voltage (📖 268)

**Proceed as follows to open the dialog for parameterising the brake energy management:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Go to the *Overview* dialog level and click the "basic functions" button.
4. Go to the *Overview* ➞ *basic functions* dialog box and click the **Brake energy management** button.




### Short overview of the relevant parameters:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00173 | Mains voltage | 3ph 400 V / 1ph 230 V | |
| C00175 | Brake energy management | R_Brake (brake resistance) | |
| Highlighted in grey = display parameter | | | |

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **Brake resistor** | | | |
| C00133 | Brake resistor utilisation | - | % |
| C00129 | Brake resistance value | 39.0 | Ohm |
| C00130 | Rated brake resistor power | 100 | W |
| C00131 | Thermal capacity - brake resistor | 10.0 | kWs |
| C00174 | Reduced brake chopper threshold | 0 | V |
| C00572 | Brake resistor overload threshold | 100 | % |
| C00574 | Resp. to overtemp. brake resistor | No response | |
| **Inverter motor brake** | | | |
| C00987 | Inverter motor brake: nAdd | 80 | rpm |
| C00988 | Inverter motor brake: PT1 filter time | 0.0 | ms |
| Highlighted in grey = display parameter | | | |

## 5.13.1 Setting the voltage source for braking operation

The voltage threshold for braking operation is set via the mains voltage (C00173) and the reduced brake chopper threshold (C00174). When this "brake chopper threshold" is exceeded, the response selected in C00175 takes place in the DC bus. The selected function (e.g. use of a brake resistor) serves to dissipate energy in the DC bus and reduce the DC-bus voltage.

- The "brake chopper threshold" is preset as follows so that it is higher than the specified mains voltage (C00173):

| | Mains voltage | | Brake chopper threshold | |
|---|---|---|---|---|
| C00173 | 1-phase | 3-phase | 1-phase | 3-phase |
| 0 | 1ph 230V | 3ph 400V | DC380V | DC725V |
| 1 | 1ph 230V | 3ph 440V | DC380V | DC735V |
| 2 | 1ph 230V | 3ph 480V | DC380V | DC775V |
| 3 | 1ph 230V | 3ph 500V | DC380V | DC790V |
| 4 | 1ph 115V | 3ph 400V | DC205V | DC725V |

- This brake chopper threshold can be reduced by 0 ... 150 V by means of C00174.

**Stop!**

The brake chopper threshold resulting from C00173 and C00174 must not exceed the stabilised DC-bus voltage!

Example:

- A 400 V device has a maximum mains voltage of 420 V AC.
  - Maximum stationary DC-bus voltage: 420 V AC * 1.414 = 594 V DC
  - C00173 has been set with the selection "0" for 400 V AC mains.
- This means that C00174 can be set to a maximum of 131 V DC (725 V DC - 594 V DC).

### 5.13.2 Selecting the response to an increase of the DC-bus voltage

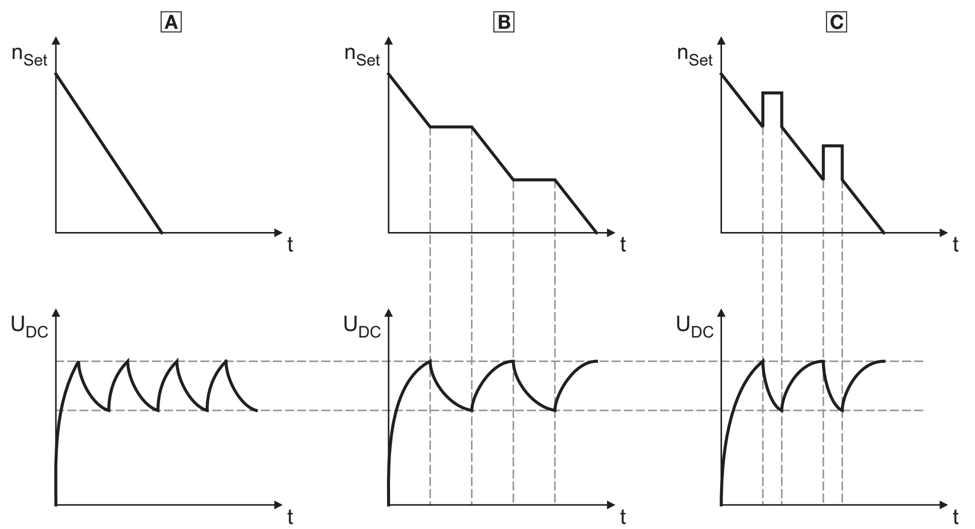
If the brake chopper threshold resulting from C00173 and C00174 is exceeded in the DC bus, the reaction selected in C00175 takes place (use of the brake resistor and/or stop of the ramp function generator and/or inverter motor brake).

- Optimum following of the actual speed value until the speed setpoint is reached (e.g. the motor is stopped rapidly) is always achieved with the help of a brake resistor.
- Stopping the ramp function generator enables smoother deceleration with lower torque oscillation..
- The inverter-motor brake is available for selection in C00175. This function enables rapid braking without a brake resistor. Torque oscillations can occur due to the traversing dynamics. ▶ Inverter motor brake (📖 270)

**Stop!**

- The two braking procedures "Stopping of the ramp function generator" and "Inverter motor brake" can only be used for speed-controlled applications without the influence of a position controller!
- When the "inverter motor brake" function is used, the Motor load monitoring (I2xt) is not adapted. If it is braked too frequently, there is a risk of the motor being thermally overloaded or the motor overload monitoring does not work properly!
- The "inverter motor brake" function must not be used with vertical conveyors (hoists) or with active loads!

The way in which the different braking procedures work is demonstrated schematically in the following illustration:




A Use of the brake resistor (C00175 = "0: Brake resistor")
B Stopping of the ramp function generator (C00175 = "1: HlgStop")
C Inverter motor brake (C00175 = "3: FI_MotBrk + RfgStop)

[5-30] Graph of the effective speed setpoint and the DC bus voltage during braking

**Tip!**

Independent of the selected motor control, all procedures given in C00175 can be used.

The actual speed value can optimally follow the speed setpoint when a brake resistor is used.

If it is possible to dispense with exact adherence to the deceleration ramp in simple applications, selection of a braking method without an external brake resistor enables costs to be reduced due to the avoidance of having to use a brake resistor .

With the "inverter motor brake" function, an effective braking torque of 10 ... 20 % of the rated motor torque can be achieved.

A combination of all three braking procedures is also possible, e.g. for emergency braking if the brake resistor fails
(C00175 = "4: Brake resistor + FI_MotBrk + RfgStop").

### 5.13.2.1 Inverter motor brake

With this braking method, which can be selected as an alternative in C00175, the regenerative energy in the motor is converted as a result of dynamic acceleration/deceleration with down-ramping of the ramp function generator..

**Stop!**

- This braking method only works without intervention of a position controller in the case of speed-controlled applications!
- When the "inverter motor brake" function is used, the Motor load monitoring (I2xt) is not adapted. If it is braked too frequently, there is a risk of the motor being thermally overloaded or the motor overload monitoring does not work properly!
- The "inverter motor brake" function must not be used with vertical conveyors (hoists) or with active loads!

**Tip!**

If no brake resistor is used, the DC injection brake can also be used for a braking process in addition to the "inverter motor brake" and "Stopping of the ramp function generator". DC-injection braking (250)

In applications with high mass inertia and long braking times (> 2 s), we recommend the use of the DC injection brake.

- The DC injection brake provides for an oscillation-minimised braking. The braking process generally takes more time than the "inverter motor brake" function with an optimised setting. Moreover, the function is only recommended for braking to a standstill.

In the following cases we recommend the "inverter motor brake" function:

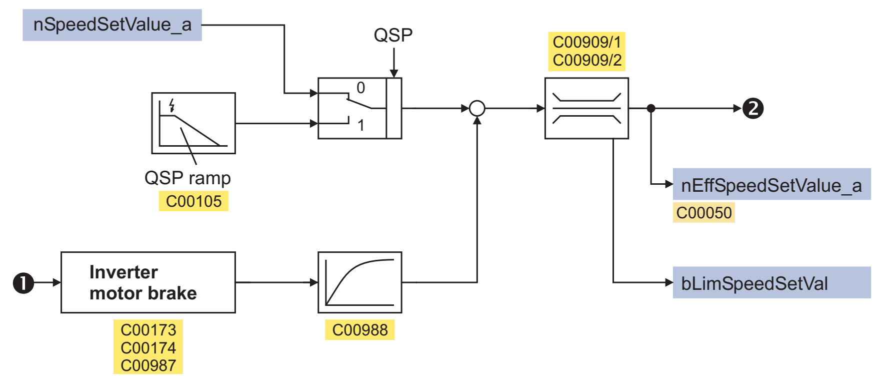
- With servo control (SC).
- For all applications that do not require braking to a standstill (e.g.braking to a lower speed setpoint) or the braking process can be interrupted by selecting a new speed setpoint.
- For applications with low mass inertias and a short braking time (< 1 s).
- For all applications where braking should be as quick as possible.

### Operating mode of the inverter motor brake

The ramp function generator is stopped during acceleration. The speed set in C00987 is added to the speed setpoint by means of a hysteresis-type 2-point DC bus voltage controller, whereby the sign of the current actual speed is taken into account. In addition, the ramp function generator is stopped during overvoltage.

If the DC bus voltage falls below a defined DC bus voltage potential of the hysteresis controller, the added speed is subtracted again and the ramp function generator is activated again.

The energy is converted into heat in the motor due to alternating instances of acceleration and deceleration as a result of this switching operation.




❶ DC-bus voltage
❷ Speed setpoint for motor control

[5-31] Signal flow of the "Inverter motor brake" function

- In case of an asynchronous motor, the additive speed setpoint (C00987) should be 1 ... 4 times the slip of the machine:

$$\text{C00987 [rpm]} = 1 \ldots 4 \cdot (n_{Sync}[\text{rpm}] - n_{Rated}[\text{rpm}])$$

$$n_{Sync}[\text{rpm}] = \frac{f_{Rated}\text{Hz} \cdot 60}{p}$$

p = number of pole pairs
$n_{Rat}$ = Rated speed of the motor
$f_{Rat}$ = Rated frequency of the motor
$n_{Sync}$ = Synchronous speed of the motor

[5-32] Formula for calculating the additive speed setpoint for an asynchronous motor

- In case of a synchronous motor, the additive speed setpoint (C00987) should be 5 ... 20 % of the rated machine speed.

### Short overview of the relevant parameters:

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C00173 | Mains voltage | 3ph 400V / 1ph 230V | |
| C00174 | Reduc. brake chopper threshold | 0 | V |
| C00175 | Resp. to brake resistor control | Brake resistor | |
| C00987 | Inverter motor brake: nAdd<br>• Speed lift which is connected in pulses to the brake rampe when the motor is braked. | 80 | rpm |
| C00988 | Inverter motor brake: PT1 filter time<br>• PT1 filter time for smoothing the speed lift which is added in pulses. | 0.0 | ms |

## Note!

When the "inverter motor brake" function is used, torque oscillations occur which have may have a negative effect on the service life of the components of the mechanical drive train (e.g. gearbox).

- The extent of the occurring oscillations depends on the drive train (mass inertia, natural frequencies, etc.) and the function setting.
- We recommend optimising the "inverter motor brake" function for an oscillation-free operation as described in the following. Usually, this setting does not cause any torque oscillations which affect the service life of the gearbox.
- The settings of implementing a maximum acceleration ramp are only recommended if the inverter motor brake is used infrequently (e.g. in case of quick stop).

**How to set the "inverter motor brake" function for an oscillation-reduced operation:**

For V/f characteristic open-loop control/closed-loop control (VFCplus):

- Set reduced brake chopper threshold (C00174) to approx. 70 V.
- Set additive speed (C00987) to rated slip speed.
- Adapt the deceleration ramp so that the deceleration time is slightly below (10 ... 30 %) the deceleration time that can be realised with the inverter motor brake.

For sensorless vector control (SLVC) and servo control (SC):

- Set reduced brake chopper threshold (C00174) to approx. 50 V.
- Set additive speed (C00987) to 1 ... 2-fold rated slip speed.
- Adapt the deceleration ramp so that the deceleration time is slightly below (10 ... 30 %) the deceleration time that can be realised with the inverter motor brake.

**How to set the "inverter motor brake" function for a maximum acceleration ramp:**

For V/f characteristic open-loop control/closed-loop control (VFCplus):

- Set reduced brake chopper threshold (C00174) to approx. 70 V.
- Set additive speed (C00987) to 1,5 ... 2,5-fold rated slip speed.
- Adapt the deceleration ramp so that the deceleration time is slightly below (10 ... 30 %) the deceleration time that can be realised with the inverter motor brake.

For sensorless vector control (SLVC) and servo control (SC):

- Set reduced brake chopper threshold (C00174) to approx. 70 V.
- Set additive speed (C00987) to 2 ... 4-fold rated slip speed.
- Adapt the deceleration ramp so that the deceleration time is slightly below (10 ... 30 %) the deceleration time that can be realised with the inverter motor brake.

### 5.13.3 Avoiding thermal overload of the brake resistor

- Parameterisation of an error response in C00574 and evaluation of the parameterised error message within the application or within the machine control system.
  - See chapter entitled "Brake resistor monitoring (I2xt)". (279)
- External interconnection using the thermal contact on the brake resistor (e.g. supply interruption via the mains contactor and activation of the mechanical brakes).

## 5.14 Monitoring

Many monitoring functions that are integrated into the controller can detect errors and thus protect the device/motor from damage or overload.

- Detailed information on the individual monitoring functions can be found in the following subchapters.

| Monitoring | Response | | Error message (with activated monitoring) |
|---|---|---|---|
| | Lenze setting | Configuration | |
| Device overload monitoring (Ixt) | Warning | C00604 | OC5 |
| Motor load monitoring (I2xt) | Warning | C00606 | OC6 |
| Motor temperature monitoring (PTC) | Fault | C00585 | OH3 |
| Brake resistor monitoring (I2xt) | No Reaction | C00574 | OC12 |
| Motor phase failure monitoring | No Reaction | C00597 | LP1 |
| Motor phse error monitoring before operation | | C02844/2 | |
| Mains phase failure monitoring | Warning | C00565 | Su02 |
| Maximum current monitoring | No Reaction | C00609 | OC7 |
| Maximum torque monitoring | No Reaction | C00608 | OT1 |
| Motor speed monitoring | Fault | - | OS2 |

### Parameterisable responses

If a monitoring function trips, the response set via the corresponding parameter is carried out. The following responses can be selected:

- "No response": Response/monitoring is deactivated.
- "Fault": Change of the operating status by a pulse inhibit of the power output stage.
- "Warning": Operating status of the controller remains unchanged. Only a message is entered into the logbook of the controller.

### Related topics:

- Device state machine and device statuses (📖 101)
- Diagnostics & error management (📖 581)
- Basics on error handling in the controller (📖 581)
- Error messages of the operating system (📖 602)

### 5.14.1 Device overload monitoring (Ixt)

C00064/1...3 displays the device utilisation (ixt) in [%] in different time intervals:

| Parameter | Info |
|---|---|
| C00064/1 | Device utilisation (Ixt)<br>• Maximum value of pulse utilisation (C00064/2) and permanent utilisation (C00064/3). |
| C00064/2 | Device utilisation (Ixt) 15s<br>• Pulse utilisation over the last 15 seconds (only for loads >160 %). |
| C00064/3 | Device utilisation (Ixt) 3 min<br>• Permanent utilisation over the last 3 minutes. |
| Highlighted in grey = display parameter | |

- If the device utilisation reaches the switch-off threshold set in C00123:
  - The error response set in C00604 will be carried out (Lenze setting: "Warning").
  - The "OC5: Ixt overload" error message will be entered into the logbook.
  - The *bMctrlIxtOverload* status output of the LS_DeviceMonitor system block will be set to TRUE.
- A setting of C00604 = "0: No Reaction" deactivates the monitoring.

### 5.14.2 Motor load monitoring (I2xt)

The Inverter Drives 8400 are provided with a simple, sensorless, thermal $I^2xt$ motor monitoring of self-ventilated standard motors which is based on a mathematical model.

- C00066 displays the calculated motor load in [%].
- If the calculated motor load reaches the motor load setting (C00120):
  - The error response set in C00606 will be carried out (Lenze setting: "Warning").
  - The "OC6: I2xt motor overload" error message will be entered into the logbook.
  - The *bMctrlI2xtOverload* status output of the LS_DeviceMonitor system block will be set to TRUE.
- A setting of C00606 = "0: No Reaction" deactivates the monitoring.

**Stop!**

The $I^2xt$ motor monitoring does not present full motor protection! As the motor utilisation calculated in the thermal motor model is lost after mains switching, for instance the following operating states cannot be measured correctly:

- Restarting (after mains switching) of a motor that is already very hot.
- Change of the cooling conditions (e.g. cooling air flow interrupted or too warm).

A full motor protection requires additional measures as e.g. the evaluation of temperature sensors that are located directly in the winding or the use of thermal contacts.

#### Adjustment of the motor utilisation meter

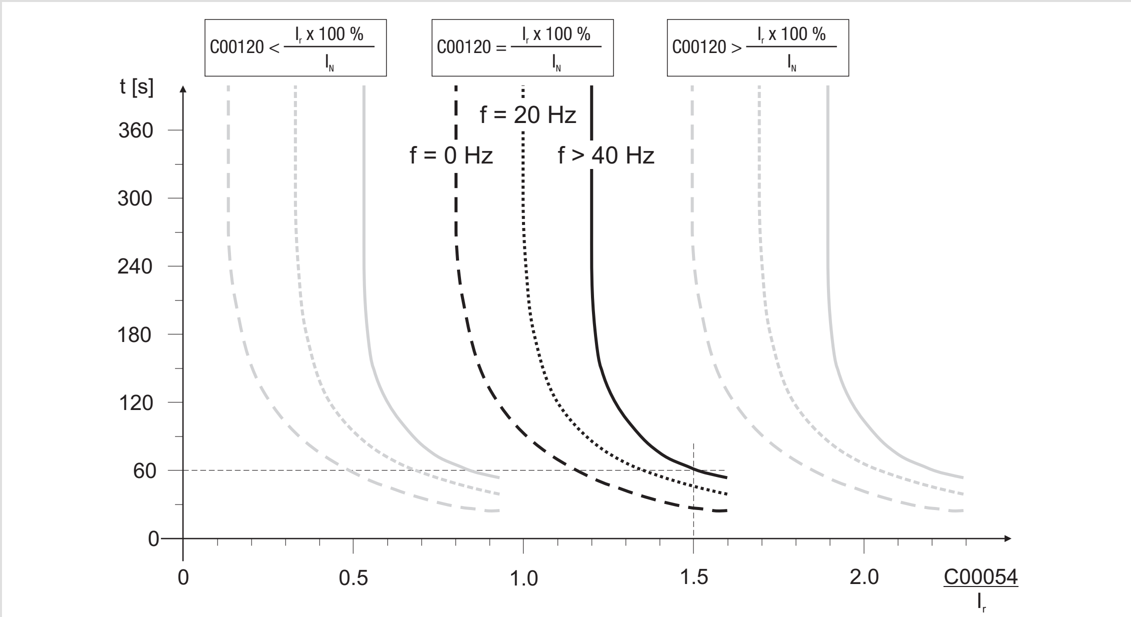
The motor utilisation meter for indicating the motor load in C00066 begins to count when the apparent motor current (C00054) is greater than the motor overload setting (C00120).

C00120 is to be set as follows:

$$C00120 = \frac{I_r}{I_N} \cdot 100\%$$

$I_r$ : Rated motor current (see nameplate of motor)
$I_N$: Rated controller current at a switching frequency of f = 8 kHz

- If you reduce C00120 starting from the calculated value, the motor utilisation meter will already be counted up before the rated overload threshold is reached.
- If you increase C00120 starting from the calculated value, the motor utilisation meter will not be counted up until the rated overload threshold is reached.




f: Output frequency
t: Release time
$I_N$: Rated current of drive controller at a switching frequency of f = 8 kHz
$I_r$ : Rated motor current (see nameplate of motor)
C00054: Apparent motor current
C00120: Motor overload setting

[5-33] Tripping characteristic of the $I^2$xt monitoring

Example:

C00120 = $I_r$ /$I_N$ x 100 %

C00054 = 150 % rated motor current

- After approx. 60 seconds, C00066 has reached the final value (100 %) at output frequencies f > 40 Hz.
- The controller outputs the "OC6: I2xt overload motor" error message and triggers the response set in C00606 (default setting: "Warning").

**Tip!**

- If forced ventilated motors are used, a premature response of the overload threshold can be avoided by deactivating this function if necessary (C00606 = "0: No Reaction").
- The current limits set in C00022 and C00023 influence the $I^2$xt calculation only in an indirect way. However, the operation of the motor at maximum possible load can be averted. Defining current and speed limits (136)

### 5.14.3 Motor temperature monitoring (PTC)

For detecting and monitoring of the motor temperature, a PTC thermistor (DIN 44081/DIN 44082) or a thermal contact (NC contact) can be connected to the terminals X106/T1 and X106/T2.

**Stop!**

- The controller can only evaluate one PTC thermistor!
  Do not connect several PTC thermistors in series or parallel.
- If several motors are operated on one controller, use thermal contacts (NC contacts) connected in series.
- To achieve full motor protection, an additional temperature monitoring with separate evaluation must be installed.

**Note!**

- In the Lenze setting (C00585 = "1: Fault"), motor temperature monitoring is activated!
- There is a wire jumper between the terminals X106/T1 and X106/T2 by default.
- Lenze three-phase AC motors are provided with a thermal contact on delivery.

- If 1.6 kΩ < R < 4 kΩ at the terminals X106/T1 and X106/T2, the monitoring will respond, see functional test below.
- If the monitoring responds:
  - The error response set in C00585 is activated (Lenze setting: "Fault").
  - The "OH3: Motor temperature (X106) tripped" error message is entered into the Logbook.
  - The *bMctrlMotorPtc* status output of the LS_DeviceMonitor system block is set to TRUE.
- A setting of C00585 = "0: No Reaction" deactivates the monitoring.

**Tip!**

We recommend to always activate the PTC input when using motors which are equipped with PTC thermistors or thermostats. This prevents the motor from being destroyed by overheating.

#### Functional test

Connect a fixed resistor to the PTC input:

- R > 4 kΩ : Fault message must be activated.
- R < 1 kΩ : Fault message must not be activated.

### 5.14.4 Brake resistor monitoring (I2xt)

Due to the converted braking power, the brake resistor is thermally stressed and can even be thermally destroyed by excessive braking power.

The monitoring of the $I^2xt$ utilisation of the controller serves to protect the brake resistor. It acts in proportion to the converted braking power.

**Danger!**

In the Lenze setting (C00574 = "0: No Reaction") the response of the monitoring function does not stop the braking process!

In particular for applications such as hoists or applications with a DC-bus connection, it must be checked if a stopping of the braking process due to a setting of C00574 = "1: Fault" is permissible.

**Stop!**

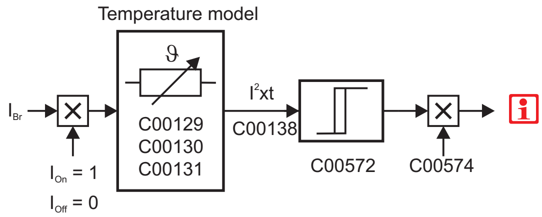
Implement appropriate protective measures against thermal overload of the brake resistor!

Examples:

- Parameterisation of an error response in C00574 and evaluation of the parameterised error message within the application or the machine control system.
- Interruption of the mains supply by means of the temperature contact at the brake resistor and a simultaneous activation of the mechanical brake.

- If the $I^2xt$ utilisation reaches the switch-off threshold set in C00572:
  - The error response set in C00574 will take place.
  - The "OC12: I2xt brake resistor overload" error message is entered into the logbook.
  - The *bMctrlBrakeChopper* status output of the LS_DeviceMonitor system block will be set to TRUE.
- If the system is dimensioned correctly, the monitoring should not be activated. If individual pieces of rated data of the actually connected brake resistor are not known, they have to be identified.
- If the DC-bus voltage exceeds the overvoltage threshold due to a braking energy that is too high, the monitoring for overvoltage in the DC bus is activated ("OU: DC-bus overvoltage" error message).
- Apart from the threshold of the $I^2xt$ utilisation that can be set in C00572, there is the switching threshold of the brake transistor which results from the mains voltage (C00173) and the reduced brake chopper threshold (C00174).

### Temperature model




Error message: "OC12: I2xt overload brake resistor"

[5-34] Signal flow for monitoring the brake resistor

The monitoring function calculates the braking current $I_{Br}$ from the current DC-bus voltage $U_{DC\_act}$ and the brake resistance parameterised in C00129:

$$I_{Br} = \frac{U_{DC\_act}}{C00129}$$

**Note!**

The monitoring function can also be triggered due to a value entered in C00129 although a brake resistor is not even connected.

- During the calculation, the thermal utilisation of the brake resistor on the basis of the following parameters is taken into consideration:
  - Resistance value (C00129)
  - Continuous power (C00130)
  - Thermal capacity (C00131)
- In the Lenze setting these parameters are preset with the corresponding power-adapted Lenze brake resistor.
- C00133 indicates the calculated utilisation of the brake resistor in [%].
  - A utilisation of 100 % corresponds to the continuous power of the brake resistor depending on the maximally permissible temperature limit.

**Related topics:**

▶ Braking operation/brake energy management (📖 265)

### 5.14.5 Motor phase failure monitoring

**Note!**

In the Lenze setting (C00597 = "0: No response"), the motor phase failure monitoring is not activated!

In order to safely detect the failure of a motor phase, a certain motor current must flow for the current sensor system. Thus, the response set in C00597 (Lenze setting: "No Reaction") is caused after a delay time of maximally 2 s after controller enable if a current-carrying motor phase U, V, W fails or if motor connection is missing. If the current threshold value set in C00599 is already exceeded within the delay time, the motor phase failure monitoring starts from this point in time.

The monitoring mode checks the current flow for each motor phase as a function of the commutation angle. Monitoring is activated if a commutation angle of approx. 140° is covered without the current set in C00599 being exceeded. Monitoring is activated at an output frequency of 0 Hz if none of the three motor phases reaches the threshold value set in C00599.

- If the motor phase failure monitoring is tripped:
  - The response set in C00597 will take place.
  - The error message "LP1: Motor phase failure" is entered into the logbook.
  - The *bMctrlMotorPhaseFault* status output of the LS_DeviceMonitor system block is set to TRUE.
- The motor phase failure monitoring is inactive if
  - a controller inhibit is set,
  - connection to a rotating machine is carried out (flying restart circuit or connection to actual speed value),
  - an error is pending due to a DC-bus overvoltage ("OU"),
  - motor parameter identification is carried out,
  - DC-injection braking is active.

### 5.14.6 Motor phse error monitoring before operation

This function extension is available from version 02.00.00!

This extended motor phase failure monitoring can both detect a phase failure on the basis of test signals and check for the existence of the motor.

- The "motor phase error monitoring before operation" is only directly active after controller enable if
  - an error response is set in C00597 **AND**
  - the motor phase error monitoring is switched on (C2866/2 = "1: Yes").

- The following parameters show the cause of the motor phase failure:
  - C00561/3: Motor phase U
  - C00561/4: Motor phase V
  - C00561/5: Motor phase W

**Note!**

The motor phase error monitoring before operation must not be connected to a rotating or coasting machine (high compensation currents and effect of the DC injection braking).

- In case of motor control with feedback, no motor phase error monitoring is executed if the actual speed value is > 10 rpm.
- In case of motor control without feedback, the user must ensure that the motor phase error monitoring will only be executed if the speed is 0.

If the motor is at quick stop and the brake is applied, no motor phase error monitoring is executed when quick stop is deactivated (same with "0" speed and applied brake).

If the rated current of the connected motor is lower than 10 % of the rated device current, the motor phase error monitoring can be activated although no motor phase error has occurred. In this case, the motor phase error monitoring must be switched off before operation (C2866/2 = "0: No").

**With automatic brake control:**

In case of automatic brake control, the brake will only be released if no motor phase failure exists and the magnetisation of the field-oriented control types is completed.

**With manual brake control:**

In case of manual brake control and forced release of the brake, the brake will be controlled directly as before.

The user himself must ensure that the brake will only be opened if all of the following conditions are met:

- Motor phase failure monitoring (C00597) and motor phase error monitoring before operation (C2866/2) are active.
- The controller is enabled (controller enable).
- The *bMctrlMotorPhaseFault* status output of the SB LS_DeviceMonitor is set to FALSE.
- A delay of the pulse enable in the device after controller enable for releasing the brake has been considered (opening the brake after controller enable must be executed with a delay).

### 5.14.7 Mains phase failure monitoring

**Stop!**

Under load, the mains input of a three-phase controller can be destroyed if the device is only supplied by two phases (e.g. if a mains phase fails).

The drive controller has a simple mains-phase failure detection function with which a mains phase failure can be detected under load.

- In the case of power-adapted machines, approx. 50 % of the rated motor power must be exceeded so that a main-phase failure can be detected.
- If the mains phase failure monitoring is tripped:
  - The error response set in C00565 will be carried out (Lenze setting: "Warning").
  - The "Su02: One mains phase is missing" error message will be entered into the logbook.
  - The *bMctrlMainsFault* status output of the LS_DeviceMonitor system block will be set to TRUE.

### 5.14.8 Maximum current monitoring

The ultimate motor current to be parameterised in C00939 is a limit value to protect the motor from destruction, influence of the rated data and demagnetisation.

- This limit value must not be travelled cyclically in the drive process.
- If the instantaneous value of the motor current exceeds the limit value set in C00939, the error response "Fault" occurs to protect the motor and the error message "OC7: Motor overcurrent" is entered into the logbook.
- The maximum currents to be parameterised in C00022 and C00023 should have a sufficient distance to this limit value.

**Note!**

If a Lenze motor is selected from the catalogue whose plant parameters are transferred into the controller, the settings in C00022 and C00023 will automatically be adapted to the selected motor.

### 5.14.9 Maximum torque monitoring

**Note!**

In the Lenze setting (C00608 = "0: No response"), the maximum torque monitoring is not activated!

If the maximum possible torque C00057 is reached at the motor shaft, the response set in C00608 will be carried out (Lenze setting: "0: No response").

If the activated monitoring is tripped:

- The "OT1: Maximum torque reached" error message is entered into the logbook.
- The *bMctrlTorqueMax* status output of the LS_DeviceMonitor system block will be set to TRUE.

### 5.14.10 Motor speed monitoring

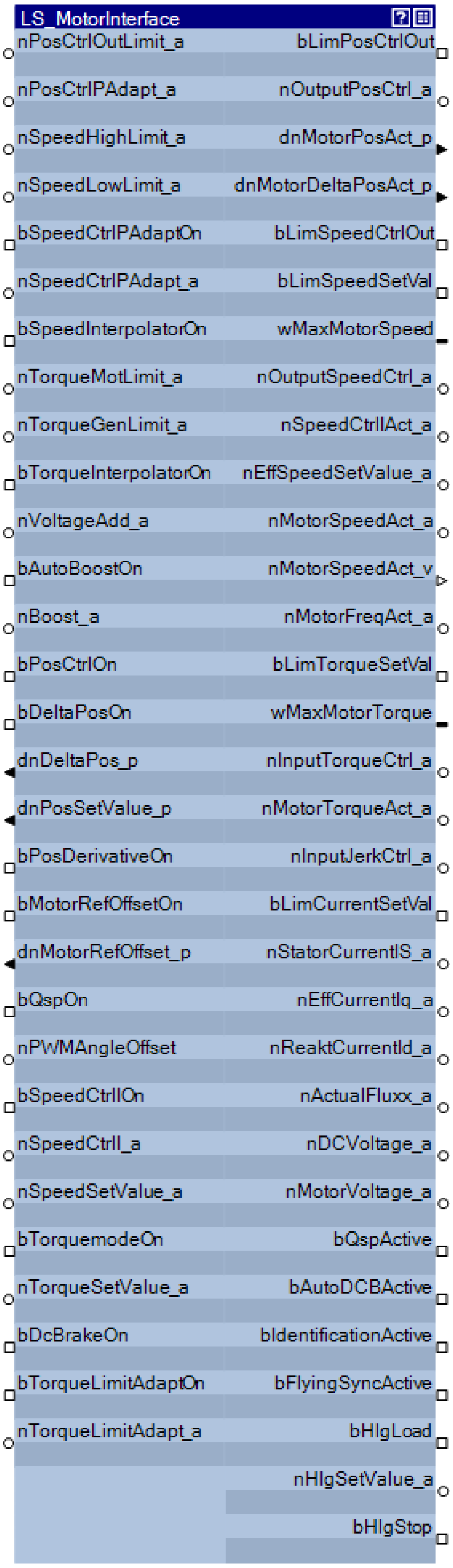
This function extension is available from version 02.00.00!

If the drive reaches the maximally permissible motor speed (C00965):

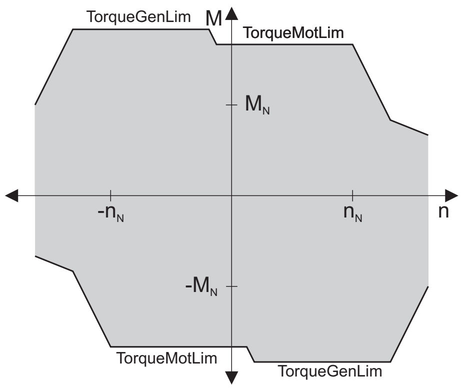
- The "Fault" error response takes place, i.e. the motor is shut down immediately.
- The error message "OS2: Max. motor speed reached" is entered into the logbook.

## 5.15 Internal interfaces | system block "LS_MotorInterface"

The **LS_MotorInterface** system block provides the internal interfaces to the driving machine in the function block editor.



### Inputs

| Identifier<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| nPosCtrlOutLimit_a<br>C00830/21 \| INT | Limitation of the position controller output<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nPosCtrlPAdapt_a<br>C00830/20 \| INT | Adaptation of the position controller gain<br>• Scaling: 16384 ≡ 100 % Vp (C00254) | |
| nSpeedHighLimit_a<br>C00830/88 \| INT | Upper speed limit for the speed limitation<br>• During torque-controlled operation only (*bTorquemodeOn* = TRUE)<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nSpeedLowLimit_a<br>C00830/23 \| INT | Lower speed limit for speed limitation<br>• During torque-controlled operation only (*bTorquemodeOn* = TRUE)<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| bSpeedCtrlPAdaptOn<br>C00833/69 \| BOOL | Adaptive adjustment of the speed controller gain | |
| | FALSE | Deactivate adaptive adaptation. |
| | TRUE | Activate adaptive adaptation. |
| nSpeedCtrlPAdapt_a<br>C00830/25 \| INT | Adaptive adjustment of the speed controller gain<br>• Scaling: 16384 ≡ 100 % Vp (C00070) | |
| bSpeedInterpolatorOn<br>C00833/28 \| BOOL | Speed setpoint interpolation | |
| | FALSE | Deactivate interpolation |
| | TRUE | Activate interpolation |
| nTorqueMotLimit_a<br>C00830/29 \| INT<br>nTorqueGenLimit_a<br>C00830/28 \| INT | Torque limitation in motor mode and in generator mode<br>• The drive cannot output a higher torque in motor/generator mode than set here.<br>• The applied values (any polarity) are internally interpreted as absolute values.<br>• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.<br>• If sensorless vector control (SLVC) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| | Torque limits in motor and generator mode:<br><br> | |
| bTorqueInterpolatorOn<br>C00833/29 \| BOOL | Torque setpoint interpolation | |
| | FALSE | Deactivate interpolation |
| | TRUE | Activate interpolation |

<table>
<tr><th>Identifier<br>DIS code | data type</th><th colspan="2">Information/possible settings</th></tr>
<tr><td rowspan="2">nVoltageAdd_a<br>C00830/31 | INT</td><td colspan="2">Additive voltage impression<br>• An additional setpoint for the motor voltage can be specified via this process input.<br>• If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time.<br>• If the value is negative, the voltage is reduced.<br>• Scaling: 16384 ≡ 1000 V</td></tr>
<tr><td colspan="2">Stop!<br>Values selected too high may cause the motor to heat up due to the resulting current!</td></tr>
<tr><td rowspan="3">bAutoBoostOn<br>C00833/32 | BOOL</td><td colspan="2">AutoBoost function<br>• Motor voltage boost during the starting torque, controlled by process signals from the function block interconnection.</td></tr>
<tr><td>FALSE</td><td>Deactivate function</td></tr>
<tr><td>TRUE</td><td>Activate function</td></tr>
<tr><td rowspan="2">nBoost_a<br>C00830/26 | INT</td><td colspan="2">Additional setpoint for the motor voltage at speed = 0<br>• The entire voltage-frequency characteristic is provided with an offset.<br>• Scaling: 16384 ≡ 1000 V</td></tr>
<tr><td colspan="2">Stop!<br>Values selected too high may cause the motor to heat up due to the resulting current!</td></tr>
<tr><td rowspan="3">bPosCtrlOn<br>C00833/27 | BOOL</td><td colspan="2">Position/angle control</td></tr>
<tr><td>FALSE</td><td>Deactivate position/angle control.</td></tr>
<tr><td>TRUE</td><td>Activate position/angle control.</td></tr>
<tr><td rowspan="3">bDeltaPosOn<br>C00833/35 | BOOL</td><td colspan="2">Activate position difference as setpoint selection<br>• In order to position the motor shaft, the position control function can work within the motor control function with the absolute position setpoint dnPosSetValue_p or alternatively with the speed setpoint nSpeedSetValue_a and the position difference dnDeltaPos_p.</td></tr>
<tr><td>FALSE</td><td>Positioning with position setpoint dnPosSetValue_p.</td></tr>
<tr><td>TRUE</td><td>Positioning with speed setpoint nSpeedSetValue_a and position difference dnDeltaPos_p.</td></tr>
<tr><td>dnDeltaPos_p<br>C00834/4 | DINT</td><td colspan="2">Position difference (following error input)<br>• Difference between setpoint position and actual position in [increments]<br>• Is used for position control if bDeltaPosOn = TRUE.<br>• Scaling: 65535 ≡ 1 revolution</td></tr>
<tr><td>dnPosSetValue_p<br>C00834/5 | DINT</td><td colspan="2">Absolute position setpoint in [increments]<br>• Is used for position control if bDeltaPosOn = FALSE.<br>• Scaling: 65535 ≡ 1 revolution</td></tr>
<tr><td rowspan="2">bPosDerivativeOn<br>C00833/67 |BOOL</td><td colspan="2">Create a setpoint for the speed controller from the position setpoint<br>• For highly dynamic control systems, the setpoint for the speed controller can be created from the absolute position setpoint dnPosSetValue_p instead of the speed setpoint nSpeedSetValue_a ▸ Position control/additive speed specification</td></tr>
<tr><td>TRUE</td><td>Create a speed setpoint from the position setpoint.<br>• The absolute position setpoint dnPosSetValue_p is differentiated and a speed value is created which is the setpoint for the speed controller.<br>• Internal limitation of 65536 increments/ms.</td></tr>
<tr><td rowspan="2">bMotorRefOffsetOn<br>C00833/68 | BOOL</td><td colspan="2">Set home position ("referencing on the fly")</td></tr>
<tr><td>TRUE</td><td>Set home position to value dnMotorRefOffset_p.</td></tr>
<tr><td>dnMotorRefOffset_p<br>C00834/6 | DINT</td><td colspan="2">Home position in [increments]<br>• Scaling: 65535 ≡ 1 revolution</td></tr>
</table>

| Identifier<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| bQspOn<br>C00833/33 \| BOOL | Quick stop | |
| | FALSE | Deactivate quick stop |
| | TRUE | Activate quick stop |
| nPWMAngleOffset_a<br>C00830/32 \| INT | Angle step change of output voltage phasor<br>• Scaling: 65535 ≡ 1 revolution | |
| bSpeedCtrlIOn<br>C00833/31 \| BOOL | Directly set the I-component of speed controller<br>• In order to statically specify a minimum torque, e.g. when a load is being lifted. | |
| | TRUE | Se the I-component of the speed controller to the value *nSpeedCtrlI_a*. |
| nSpeedCtrlI_a<br>C00830/24 \| INT | Value of the speed controller integrator<br>• Scaling depends on the selected motor control:<br>–V/f control (VFCplus + encoder):<br>16384 ≡ 100 % reference speed (C00011)<br>–Servo control (SC) or vector control (SLVC):<br>16384 ≡ 100 % $M_{max}$ (C00057) | |
| nSpeedSetValue_a<br>C00830/22 \| INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| bTorquemodeOn<br>C00833/30 \| BOOL | Selection: Speed/Torque control | |
| | FALSE | Speed control with torque limitation |
| | TRUE | Torque control with speed limitation |
| nTorqueSetValue_a<br>C00830/27 \| INT | Torque setpoint / additive torque<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| bDcBrakeOn<br>C00833/34 \| BOOL | Activate DC injection brake | |
| | FALSE | Deactivate DC-injection braking |
| | TRUE | Activate DC-injection braking |
| bTorqueLimitAdaptOn<br>C00833/98 \| BOOL | Adaptation of torque limitation | |
| | TRUE | Activate adaptation of torque limitation. |
| nTorqueLimitAdapt_a<br>C00830/70 \| INT | Value for adaptation of torque limitation<br>• Scaling: 16384 ≡ 100 % *nTorqueMotLimit_a* and. *nTorqueGenLimit_a* | |

## Outputs

| Identifier<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| bLimPosCtrlOut<br>BOOL | "Position controller output inside the limitation" status signal | |
| | TRUE | The position controller output is internally limited |
| nOutputPosCtrl_a<br>INT | Position controller output<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| dnMotorPosAct_p<br>DINT | Current position of the motor shaft in [increments] | |
| dnMotorDeltaPosAct_p<br>DINT | Current following error in [increments]<br>• Following error = Difference between set position and actual position | |
| bLimSpeedCtrlOut<br>BOOL | "Speed controller or manipulating variable of the slip regulator inside the limitation" status signal | |
| | TRUE | The speed controller output is internally limited |
| bLimSpeedSetVal<br>BOOL | "Reduction or increase of the setpoint speed active" status signal | |
| | TRUE | Reduction or increase of the setpoint speed by the $I_{max}$ controller is active |
| wMaxMotorSpeed<br>C00011 \| BOOL | Reference speed (C00011) | |

| Identifier<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| nOutputSpeedCtrl_a<br>INT | Speed or slip controller output<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nSpeedCtrlIAct_a<br>INT | Current value of speed controller integrator<br>• Scaling depends on the selected motor control:<br>– V/f control (VFCplus + encoder):<br>16384 ≡ 100 % reference speed (C00011)<br>– Servo control (SC) or vector control (SLVC):<br>16384 ≡ 100 % $M_{max}$ (C00057) | |
| nEffSpeedSetValue_a<br>INT | Effective speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorSpeedAct_a<br>C00051 \| INT | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorSpeedAct_v<br>INT | Actual speed value<br>• Scaling: 65535 ≡ 1 revolution | |
| nMotorFreqAct_a<br>C00058 \| INT | Current field frequency | |
| bLimTorqueSetVal<br>BOOL | "Setpoint torque inside the limitation" status signal | |
| | TRUE | The setpoint torque is internally limited |
| wMaxMotorTorque<br>C00057 | Maximum motor torque<br>• *wMaxMotorTorque* = 10 * $M_{max}$ (C00057) | |
| nInputTorqueCtrl_a<br>INT | Input value of the torque control (torque setpoint)<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| nMotorTorqueAct_a<br>C00056/2 \| INT | Actual torque<br>• In the "VFC (+encoder)" motor control mode, this value is determined from the current motor current and only approximately corresponds to the actual torque value..<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| nInputJerkCtrl_a<br>INT | Input value of the jerk limitation<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| bLimCurrentSetVal<br>BOOL | "Current setpoint inside the limitation" status signal | |
| | TRUE | The current setpoint is internally limited |
| nStatorCurrentIS_a<br>INT | Current stator current/effective motor current<br>• Scaling: 16384 ≡ 100 % $I_{max\_mot}$ (C00022) | |
| nEffCurrentIq_a<br>INT | Current torque-producing cross current<br>• Scaling: 16384 ≡ 100 % $I_{max\_mot}$ (C00022) | |
| nReaktCurrentId_a<br>INT | Current field-producing direct-axis current<br>• Scaling: 16384 ≡ 100 % $I_{max\_mot}$ (C00022) | |
| nActualFluxx_a<br>INT | Current magnetising current<br>• Scaling: 16384 ≡ 100 % $I_{max\_mot}$ (C00022) | |
| nDCVoltage_a<br>INT | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V | |
| nMotorVoltage_a<br>INT | Current motor voltage/inverter output voltage<br>• Scaling: 16384 ≡ 1000 V | |
| bQspActive<br>BOOL | "Quick stop active" status signal | |
| | TRUE | Quick stop is active |
| bAutoDCBActive<br>BOOL | "Automatic DC-injection braking active" status signal<br>▸ DC-injection braking (📖 250) | |
| | TRUE | Automatic DC-injection braking is active |
| bIdentificationActive<br>BOOL | "Motor parameter identification active" status signal<br>▸ Automatic motor data identification (📖 126) | |
| | TRUE | Motor parameter identification is active |

| **Identifier**<br>DIS code \| data type | **Value/meaning** | |
|---|---|---|
| bFlyingSyncActive<br>BOOL | "Flying restart function active" status signal<br>▸ Flying restart function (📖 247) | |
| | TRUE | Flying restart function is active |
| bHlgLoad<br>BOOL | Control signal for an additional loading function of the ramp function generator<br>• → L_NSet_1.*bExternalCINH*<br>• To enable the ramp function generator to follow automatically when the controller is inhibited, for jerk-free setpoint connection. | |
| | TRUE | Set the ramp function generator to a setpoint of *nHlgSetValue_a* |
| nHlgSetValue_a<br>INT | Setpoint for an additional loading function of the ramp function generator<br>• → L_NSet_1.*nCInhVal_a*<br>• For speed-controlled drive tasks, the current actual speed value (e.g. in case of an active pulse inhibit, flying restart function, controller inhibit) is provided at this output.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| bHlgStop<br>BOOL | Control signal for stopping the ramp function generator (L_NSet_1) | |
| | TRUE | Stop the ramp function generator |

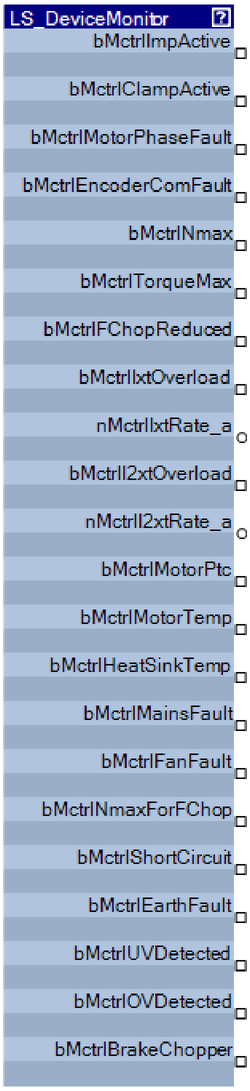
## 5.16 Internal status signals | system block "LS_DeviceMonitor"

The **LS_DeviceMonitor** system block provides the status signals of the motor control in the function block editor.

**Note!**

The **LS_DeviceMonitor** system block can only be inserted on the application level.

If status signals of the motor control function are to be output via digital outputs or example, you can use the free *bFreeOut1* ... *bFreeOut8* outputs of the application block to transfer the desired status signals from the application level to the I/O level. On the I/O level, you can then establish the logical link to the digital output terminals.



## Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bMctrlImpActive<br>BOOL | TRUE | Pulse inhibit is active |
| bMctrlClampActive<br>BOOL | TRUE | Clamp current limitation is active |
| bMctrlMotorPhaseFault<br>BOOL | TRUE | Motor phase fault has been detected |
| bMctrlEncoderComFault<br>BOOL | TRUE | Encoder error has been detected |
| bMctrlNmax<br>BOOL | TRUE | Max. speed limitation is active |
| bMctrlTorqueMax<br>BOOL | TRUE | Max. torque limitation is active |
| bMctrlFChopReduced<br>BOOL | TRUE | PWM frequency reduction is active |
| bMctrlIxtOverload<br>BOOL | TRUE | Device utilisation (Ixt) ≥ device utilisation threshold (C00123)<br>• Lenze setting: C00123 = 100 % |
| nMctrlIxtRate_a<br>INT | Current device utilisation (Ixt)<br>• Scaling: 16384 ≡ 100 % | |
| bMctrlI2xtOverload<br>BOOL | TRUE | Thermal motor overload ($I^2xt$) ≥ motor overload setting (C00120)<br>• Lenze setting: C00120 = 100 %. |
| nMctrlI2xtRate_a<br>INT | Current thermal motor load (I2xt)<br>• Scaling: 16384 ≡ 100 % | |
| bMctrlMotorPTC<br>BOOL | TRUE | Temperature monitoring: An error has been detected |
| bMctrlMotorTemp<br>BOOL | TRUE | Thermal motor overload |
| bMctrlHeatSinkTemp<br>BOOL | TRUE | Thermal inverter overload |
| bMctrlMainsFault<br>BOOL | TRUE | Mains phase failure/Mains failure |
| bMctrlFanFault<br>BOOL | TRUE | Fan monitoring: An error has been detected |
| bMctrlNmaxForFChop<br>BOOL | TRUE | The maximum field frequency for the respective switching frequency has been exceeded. |
| bMctrlShortCircuit<br>BOOL | TRUE | Motor short circuit has been detected |
| bMctrlEarthFault<br>BOOL | TRUE | Earth fault has been detected |
| bMctrlUVDetected<br>BOOL | TRUE | An undervoltage has been detected |
| bMctrlOVDetected<br>BOOL | TRUE | An overvoltage has been detected |
| bMctrlBrakeChopper<br>BOOL | TRUE | Brake chopper error |

# 6 Encoder/feedback system

The 8400 TopLine uses various encoder/feedback systems for the motor controls with speed feedback and the position control:

- Resolver at X7
- TTL encoder at X8
- HTL encoder at the digital DI1/DI2 or DI6/DI7 input terminals

**Danger!**

- If the encoder/resolver is used as motor encoder:
  Safe operation of the motor is no longer ensured in the event of an error!
- If servo control (SC) or V/f control (VFCplus + encoder) are used: For safety reasons, always select "Fault" as a response for the (open-circuit) monitoring of the encoder!
  - C00586: Resp. open circuit HTL encoder
  - C00603/1: Resp. open circuit MultiEncoder
  - C00603/2: Resp. open circuit resolver
- If an HTL encoder is used at the digital input terminals:
  Observe the maximum input frequencies of the digital inputs!
  - DI1/DI2: max. 200 kHz
  - DI6/DI7: max. 10 kHz
- To avoid interference when using an encoder, only use shielded motor and encoder cables!

**Note!**

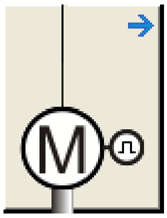
Speed feedback is essential for the following motor control modes with feedback:

- Servo control (SC)
- V/f control (VFCplus + encoder)

## 6.1 Parameterisation dialog/signal flow

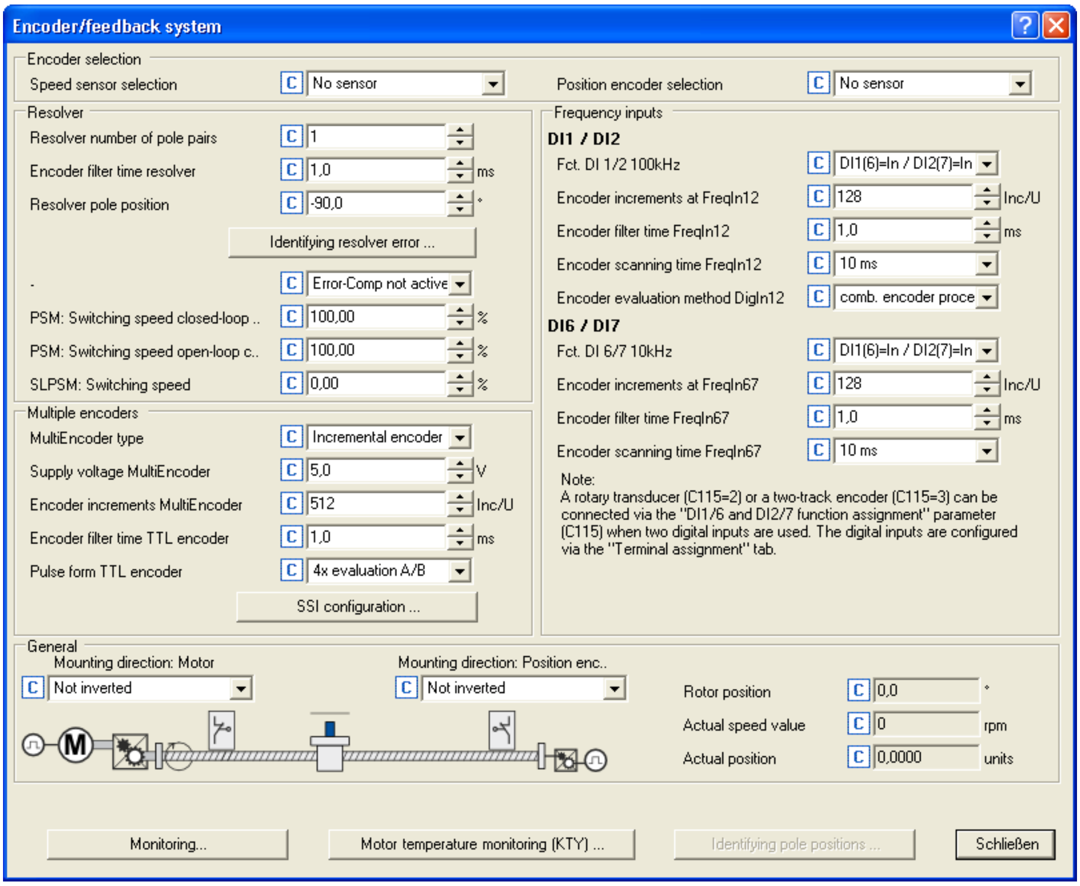
**How to get to the parameterisation dialog of the encoder/feedback system:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Go to the *Overview* dialog level and click the following button:



4. Go to the *Overview* ➞ *Motor data* dialog level and click the **Encoder/Feedback system...** button.

### Parameterisation dialog in the »Engineer«



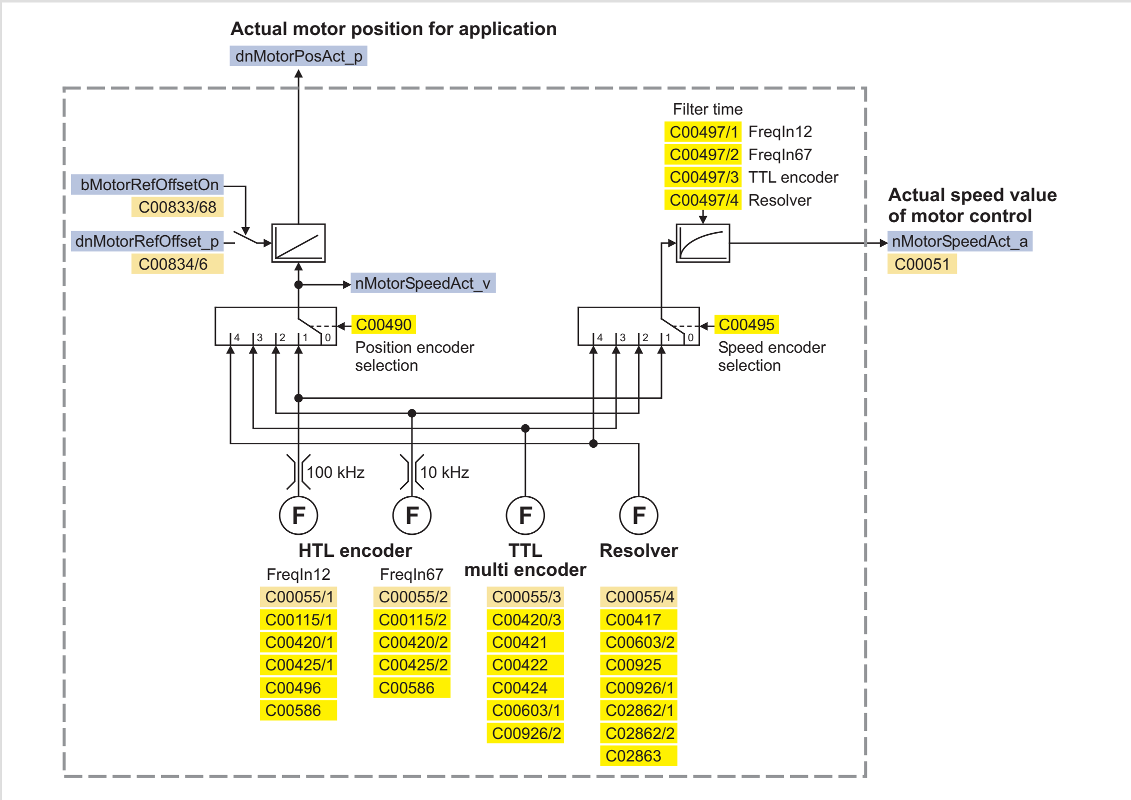
### Short overview of the relevant parameters:

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| **Encoder selection / General settings** | | | |
| C00495 | Speed sensor selection<br>• Source of feedback signal for speed control. | No sensor | |
| C00490 | Position encoder selection<br>• Source of feedback signal for position control. | No sensor | |
| C00254 | Kp position controller | 5.00 | 1/s |
| C01206/1 | Mounting direction: Motor | not inverted | |
| C01206/2 | Mounting direction: Position encoder | not inverted | |
| C00927 | Rotor position | - | ° |
| C00051 | MCTRL: Actual speed value | - | rpm |
| C01210/3 | MCK: Actual position | - | units |
| **Resolver settings** | | | |
| C00925 | Resolver - number of pole pairs | 1 | |
| C00497/4 | Encoder filter time resolver | 2.0 | ms |
| C00926/1 | Resolver pole position | -90.0 | ° |
| C00002/25 | Identify resolver error | 0: Off / ready | |
| C00417 | Activate resolver error comp. | 0: Activate resolver error compensation | |
| C02862/1 | Resolver: Cos gain | 100.00 | % |
| C02862/2 | Resolver: Sin gain | 100.00 | % |
| C02863 | Resolver: Phase error | 0.00 | % |
| C00055/4 | Actual value - resolver | - | rpm |
| **TTL Multi-Encoder settings** | | | |
| C00422 | Encoder type | 0: Incremental encoder (TTL) | |
| C00421 | Encoder voltage | 5.0 | V |
| C00420/3 | TTL encoder | 512 | Incr./rev. |
| C00497/3 | Encoder filter time TTL encoder | 1.0 | ms |
| C00424 | Pulse form TTL encoder | 0: 4x evaluation A/B | |
| C00926/2 | Encoder pole position | 0.0 | ° |
| C00055/3 | Actual value - TTL-Multi-Encoder | - | rpm |
| **Settings for HTL encoder at DI1/DI2** | | | |
| C00115/1 | Fct. DI 1/2 100kHz<br>• Function of the digital inputs DI1 and DI2 | DI1(6)=In / DI2(7)=In | |
| C00420/1 | Encoder increments at FreqIn12<br>• If the digital inputs DI1 and DI2 are used as encoder inputs. | 128 | Incr./rev. |
| C00497/1 | Encoder filter time FreqIn12<br>• If the digital inputs DI1 and DI2 are used as encoder inputs. | 1.0 | ms |
| C00425/1 | Encoder scanning time FreqIn12<br>• If the digital inputs DI1 and DI2 are used as encoder inputs. | 10 | ms |
| C00496 | ▸ Encoder evaluation method (📖 316) | Combined encoder method | |
| C00055/1 | Actual value - HTL encoder FreqIn12 | - | rpm |

Highlighted in grey = display parameter

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **Settings for HTL encoder at DI6/DI7** | | | |
| C00115/2 | Fct. DI 6/7 10kHz<br>• Function of the digital inputs DI6 and DI7 | DI1(6)=In / DI2(7)=In | |
| C00420/2 | Encoder increments at FreqIn67<br>• If the digital inputs DI6 and DI7 are used as encoder inputs. | 128 | Incr./rev. |
| C00497/2 | Encoder filter time FreqIn67<br>• If the digital inputs DI6 and DI7 are used as encoder inputs. | 1.0 | ms |
| C00425/2 | Encoder sample time FreqIn67<br>• If the digital inputs DI6 and DI7 are used as encoder inputs. | 10 | ms |
| C00055/2 | Actual value - HTL encoder FreqIn67 | - | rpm |
| **Monitoring** | | | |
| C00498 | Open-circuit monitoring | 0: Speed encoder and position encoder | |
| C00586 | Resp. open circuit HTL encoder | 1: Fault | |
| C00603/1 | Resp. to MultiEncoder open circuit | 0: No Reaction | |
| C00603/2 | Resp. to resolver open circuit | 0: No Reaction | |
| C00603/3 | Resp. to encoder angular drift monitoring | 0: No Reaction | |
| C00603/4 | Resp. to MultiEncoder comm. error | 1: Fault | |
| C00579 | Resp. max. speed/output freq. reached | 0: No Reaction | |
| C00607 | Resp. to max. freq. feedb. DIG12/67 | 1: Fault | |
| Highlighted in grey = display parameter | | | |

## Signal flow




[6-1] Signal flow - encoder interface

## Generation of the actual speed value

...depending on the encoder selection and mounting direction:

| Speed sensor (C00495) | Position encoder (C00490) | Motor mounting direction (C01206/1) | Direction of rotation of motor shaft (at setpoint = Cw) | Actual speed value (nAct_v) |
|---|---|---|---|---|
| ☐ | ☐ | not inverted | Cw | ΔSetPos |
| | | inverted | Ccw | |
| ☒ | ☐ | not inverted | Cw | C00495 |
| | | inverted | Ccw | |
| ☒ | ☒ | not inverted | Cw | |
| | | inverted | Ccw | |
| ☐ | ☒ | not inverted | Cw | C00490 → ΔActPos |
| | | inverted | Ccw | |
| ☐ no encoder | ☒ Encoder set | | | |

## 6.2 Resolver at X7

**Danger!**

- If the resolver is used as motor encoder:
  Safe operation of the motor is no longer ensured in the event of an error!
- If servo control (SC) or V/f control (VFCplus + encoder) are used: For safety reasons, always select "Fault" in C00603/2 as a response for the (open-circuit) monitoring of the encoder!

### 6.2.1 Parameterising the resolver

**Note!**

If a Lenze motor with resolver is used, the resolver is the only item that can be selected as speed sensor in C00495. Further resolver parameter setting is not required.

Moreover, Lenze synchronous machines do not require rotor position adjustment because the resolvers of the Lenze synchronous motors are pre-adjusted.

The following steps are required for commissioning a resolver:

1. If number of resolver pole pairs ≠ 1:
   Set number of pole pairs (C00925).
2. Select resolver as speed sensor (C00495) or/and as position encoder (C00490).
3. Adjust filter time of the actual speed value (C00497/4) if necessary.
   - Lenze setting: 2 ms
   - Lenze recommend a filter time between 1 ... 2 ms.
4. If the resolver is used as motor encoder and synchronous machine:
   Perform rotor position adjustment.
5. For the detection and monitoring of the motor temperature via the resolver cable:
   - Select the thermal sensor of the resolver in C01190/1.
   - Parameterise the monitoring:
     ▸ Motor temperature monitoring (KTY) (📖 323)
6. Parameterise open-circuit monitoring.
   ▸ Resolver/multi-encoder open-circuit monitoring (📖 320)

### 6.2.2 Optimising resolver behaviour

Due to mounting and production tolerances as well as resolver material property leakage, errors may occur which, among other things, result in speed-dependent vibration of the actual speed. These errors are called resolver errors. Resolver errors typically occur in the form of the 1st and 2nd harmonic. They have two different causes:

1. The inductances of the sine and cosine track of the resolver have slightly different values.
2. Sine and cosine track do not magnetise orthogonally to each other.

Resolver errors due to cause 1 can be corrected by adjusting the gains of the digital/analog converters which feed the resolver tracks. In the Lenze setting, the gains of both resolver tracks are preset with identical values.

Resolver errors due to cause 2 can be compensated for by a slight correction of the angle via which both resolver tracks are fed relative to one another.

When the "Resolver error identification" device command (C00002/25) is executed, the gain of the resolver signals and the angular drift of both resolver tracks is corrected to minimise the resolver error.

- Select a speed-controlled operating mode (e.g. servo control) for your machine while you perform a resolver error identification run. During the identification run, speed must be constant and greater than 500 rpm.
- After a successful resolver error identification run, the resolver automatically uses the following resolver error parameters which have been identified during the procedure:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02862/1 | Resolver: Cos gain | 100.00 | % |
| C02862/2 | Resolver: Sin gain | 100.00 | % |
| C02863 | Resolver: Phase error | 0.00 | % |

- The detected gain can have values from 80 ... 120 %. It makes sense to adjust only one of the two gains during a resolver error compensation. The other one remains at 100 % (Lenze setting).
- Save the parameter set afterwards to accept the identified resolver error parameters permanently (device command C00002/11).
- If the resolver error compensation is deactivated (C00417 = "1: Resolver error comp. deact."), the resolver will resume work with the Lenze setting. However, the identified resolver error parameters remain saved.

**How to perform a resolver error identification run:**

1. Select a motor control mode with speed feedback in C00006 and commission it.
   - Selecting the control mode
   - Avoid a speed controller setting that is too tight (motor humming).
   - Set the speed ramps as well as the speed setpoint (> 500 rpm).
2. Inhibit the controller, e.g. via the C00002/16 device command or via a LOW signal at terminal X5/RFR.
   - The controller must be in the "SwitchedOn" status for the resolver error identification run to be performed.
3. Activate resolver error identification via the C00002/25 device command = "1: On / start".
4. Enable the controller again.
   - The controller changes to the "MotorIdent" device status.
   - The resolver error identification run starts.
   - The progress of the identification run can be seen in C00002/25.
   - During the resolver error identification run, the direction of rotation of the motor is not important.
   - We recommend to traverse at constant setpoint speed (|nset| > 500 rpm) until resolver error identification has finished.
   - If the application offers only limited space for traversing, reversing is also possible during active resolver error identification. In this case, ensure that the controller is not inhibited and that the speed profile is traversed for at least 1 second at constant setpoint speed (|nset| > 500 rpm).
   - The identification is completed if the "0: Off / ready" message is displayed in C00002/25.
   - After successful identification, it changes back to the "SwitchedOn" device status.
5. After the resolver identification run has been completed successfully:
   - It is possible to inhibit the controller again.
   - Resolver error compensation is activated (C00417 = "0: Resolver error comp. act.").
6. In order to accept the identified resolver error parameters:

   Save parameter set.

**Tip!**

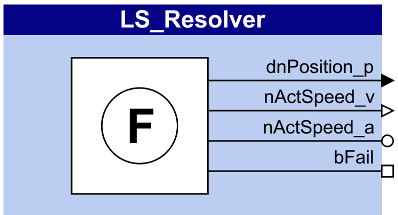
Resolver error identification can be aborted any time by setting a controller inhibit.

**Possible causes for resolver error identification failure:**

- A controller inhibit was set during resolver error identification.
- A time-out occurred while the algorithm was processed.
  - This error may occur if more than 60 seconds pass by between the setting of the "Resolver error identification" device command and the enable of the controller.
- The setpoint speed was too small (|nset| < 500 rpm).
- The setpoint speed was not traversed for at least 1 second.

### 6.2.3 Internal interfaces | "LS_Resolver" system block

The **LS_Resolver** system block provides the application with speed, position and error information from the resolver in the form of process signals.




#### Outputs

| Identifier | Data type | Value/meaning |
|---|---|---|
| dnPosition_p | DINT | Actual encoder position in [increments] |
| nActSpeed_v | INT | Actual speed in [increments/ms] |
| nActSpeed_a | INT | Actual speed in [%]<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| bFail | BOOL | "Error" status<br>TRUE: Encoder error (e.g. open circuit) |

## 6.3 Multi-Encoder at X8

**Danger!**

- If the resolver is used as motor encoder:
  Safe operation of the motor is no longer ensured in the event of an error!
- If servo control (SC) or V/f control (VFCplus + encoder) are used: For safety reasons, always select "Fault" (Lenze setting) in C00603 1 as a response for the (open-circuit) monitoring of the encoder!

**Stop!**

Before connecting an encoder, ensure that the encoder supply voltage has the correct setting in C00421:

- Lenze setting: 5 V (setting range: 5 ... 12 V)

If the set supply voltage exceeds the permissible supply voltage of the connected encoder, the encoder may be destroyed!

**Note!**

The encoder position can be saved safe against mains failure in the controller.

- For this purpose, bit 0 must be set in C02652. The actual position of the motor control is then saved in the controller (not in the memory module) and thus remains known to the drive control after mains switching.
- In case of a device replacement, this information gets lost and the home position might be approached or set again once. A device replacement can be recognised by using the SB LS_RetainData and a suitable function block interconnection.

Regarding the position resolution, higher-level applications refer to the resolution of the encoder that is activated for position control.

The interface of the Multi-Encoder offers various ways in which the encoders can be used / evaluated:

- TTL incremental encoder
- Sin/cos encoder
- Absolute value encoder (sin/cos encoder with hiperface)
- SSI encoder
- Digital frequency coupling – (in preparation)

### Required settings

The table below lists the required settings for the various encoder types as a function of the cable length.

- The given values apply when Lenze system cables are used under typical ambient temperatures.
- Other cables and cable cross-sections or extreme ambient temperatures may require metrologically detected adaptations.
- In case of a higher supply power of encoders, e.g. with laser-based length measuring systems, an external supply of the encoders must be provided.

| **Encoder type**<br>Product key | **Rated voltage** $V_{rated}$ | **Encoder number of increments** C00420/3 | **Voltage setting in** C00421 with a cable length of | | | |
|---|---|---|---|---|---|---|
| | | | 0 - 30 m | 30 - 70 m | 70 - 100 m | 100 - 150 m |
| **TTL incremental encoder** | | | | | | |
| C00422 ="0: Incremental encoder (TTL)" | | | | | | |
| IG2048-5V-T | 5 V ±5 % | 2048 | 5.0 V | 5.1 V | 5.2 V | 5.3 V |
| IG4096-5V-T | | 4096 | | | | |
| IK2048-5V-T | | 2048 | | | | |
| IK4096-5V-T | | 4096 | | | | |
| **Sin/cos encoder** | | | | | | |
| C00422 ="1: Sine/cosine encoder" | | | | | | |
| IG1024-5V-V | 5 V ±5 % | 1024 | 5.0 V | 5.1 V | 5.2 V | 5.3 V |
| IG2048-5V-S | | 2048 | | | | |
| **Absolute value encoder (sin/cos encoder with hiperface)** | | | | | | |
| C00422 ="2: Absolute value encoder (hiperface)" | | | | | | |
| AM1024-8V-H | 8 V<br>(7 ... 12 V) | 1024 | 8.0 V | | | |
| **SSI encoder** | | | | | | |
| C00422 ="4: Absolute value encoder (SSI)"<br>• All SSI encoders which use the Stegmann SSI protocol are supported.<br>• For configuration, see subchapter "SSI encoder". (📖 307) | | | | | | |

### 6.3.1 Parameterising Multi-Encoders

The following steps are required for commissioning an encoder:

1. Select encoder type (C00422).
2. Set encoder voltage (C00421).
3. Set number of encoder increments (C00420/3).
4. Adjust filter time of the actual speed value (C00497/3) if necessary.
   – Lenze setting: 1 ms
   – Lenze recommend a filter time between 1 ... 2 ms.
5. If the encoder is used as motor encoder and synchronous machine:
   Perform rotor position adjustment.
6. For the detection and monitoring of the motor temperature via the encoder cable:
   – Select the thermal sensor of the encoder in C01190/2.
   – Parameterise the monitoring:
     ▶ Motor temperature monitoring (KTY) via encoder (📖 325)
7. Parameterise open-circuit monitoring.
   ▶ Resolver/multi-encoder open-circuit monitoring (📖 320)
8. When a TTL-incremental encoder is used:
   No further settings required. The TTL incremental encoder is now ready for operation.
9. When another encoder is used:
   Perfom the commissioning steps specified in the subchapter for the respective encoder.

#### 6.3.1.1 TTL incremental encoder

For operating TTL incremental encoders, only observe the information on parameter setting and commissioning in chapter [6.3.1].

#### 6.3.1.2 Sin/cos encoder

The use of sin/cos encoders enables a fine interpolation of the position within one encoder increment due to the analog representation of the track signals A and B. This serves to considerably increase the resolution of the position information of a sin/cos encoder compared to a conventional TTL incremental encoder.

### 6.3.1.3 Absolute value encoder (sin/cos encoder with hiperface)

Absolute value encoders are especially suited for synchronous motors or positioning modes where only a one-time homing is to be executed. The analog evaluation of the sin/cos tracks enables a high resolution.

- With regard to the saving of the position information, it is distinguished between singleturn and multiturn encoders:
  - Singleturn: Saving within one revolution
  - Multiturn: Saving within a number of revolutions

**Note!**

The selection of an error response in C00603/4 serves to optionally monitor the communication between absolute value encoder and controller.

If the monitoring is tripped:

- The error response set in C00603/4 is triggered.
- The "SD7: Encoder communication error" error message is entered into the logbook.

**Short overview of the relevant parameters:**

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00492 | Hiperface: Detected TypCode | - | |
| C00493 | Hiperface: TypCode | 0 | |
| C00494/1 | Hiperface: Number of revolutions | 0 | |
| C00494/2 | Hiperface: Steps per revolution | 0 | |

Highlighted in grey = display parameter

**How to parameterise the absolute value encoder:**

1. As encoder type (C00422), select "2: Absolute value encoder (hiperface)".
2. Automatic detection/manual setting of encoder type:

   **Automatic readout of the type code**

   If the encoder has already been connected and read out, the type code is displayed in C00492.

   - Automatic readout takes place if C00494/1..2 or C00422 change (change to selection "2: Absolute value encoder (hiperface)") and if the device is switched on after a waiting time of approx. 1 s has expired.
   - An overview of encoder types that are detected and set automatically by the controller can be found in the table below.
   - If the value 0xFF is read out as type code, it is about an encoder with extended nameplate (ET). This will be set automatically as well.
   - The values read out of the encoder are displayed in C00494/1..2 and cannot be overwritten.

   **Encoders not detected automatically**

   If the used encoder is not included in the table (unknown type code), the value 0x00 is displayed in C00492.

   - In this case, the number of steps per revolution must be set in C00494/1 and the number of revolutions in C00494/2 for a multiturn encoder.
3. Set the number of sine/cosine periods per revolutions as "increment" of the encoder in C00420/3.
   - This value can be obtained from the encoder data sheet.
   - This point must be executed last since it initiates a renewed readout of the encoder.

### Encoders detected automatically

| Type code | Bits Singleturn | Bits Multiturn |
|---|---|---|
| 0x02 | 14 | 0 |
| 0x07 | 14 | 11 |
| 0x22 | 15 | 0 |
| 0x27 | 15 | 12 |
| 0x32 | 12 | 0 |
| 0x37 | 12 | 9 |
| 0x42 | 9 | 0 |
| 0x47 | 9 | 6 |

### 6.3.1.4 SSI encoder

All SSI encoders which use the Stegmann SSI protocol are supported.

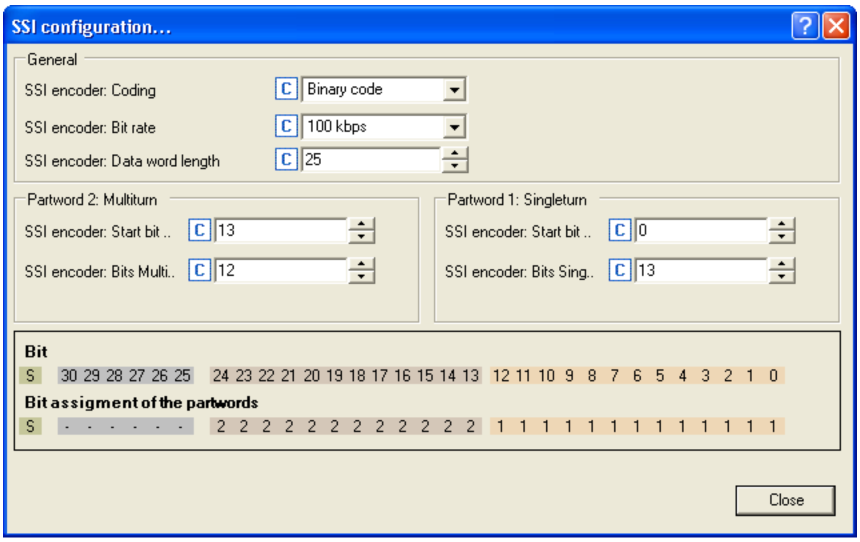
- Supported bit rates for SSI communication: 100 ... 1000 kbps
- Supported data word widths: 1 ... 31 bits (effective)
- Supported output code of the SSI encoder: Gray or binary
- The SSI encoder can be used as position encoder or master encoder with a minimum cycle time of 1 ms.
- The SSI encoder can be supplied via X8 up to a maximum voltage of 12 V and a maximum current of 0.25 A.
- The received SSI data words are provided to the application via the LS_MultiEncoder system block for further processing in the function block editor.

**Note!**

For some SSi encoder types it is common that despite a high data transfer rate an slow position detection (e.g. every 5, 10 or 20 ms) takes place. In this case, the position control, working in a 1-ms cycle, only functions with low gain and thus low dynamics since the values of the SSI encoder are only updated every x ms.

#### Parameterisation dialog in the »Engineer«

Confirm the **SSI configuration...** button in the *Encoder/Feedback system* dialog box to go to the parameterisation dialog for configuring the SSI encoder:



### Short overview of the relevant parameters:

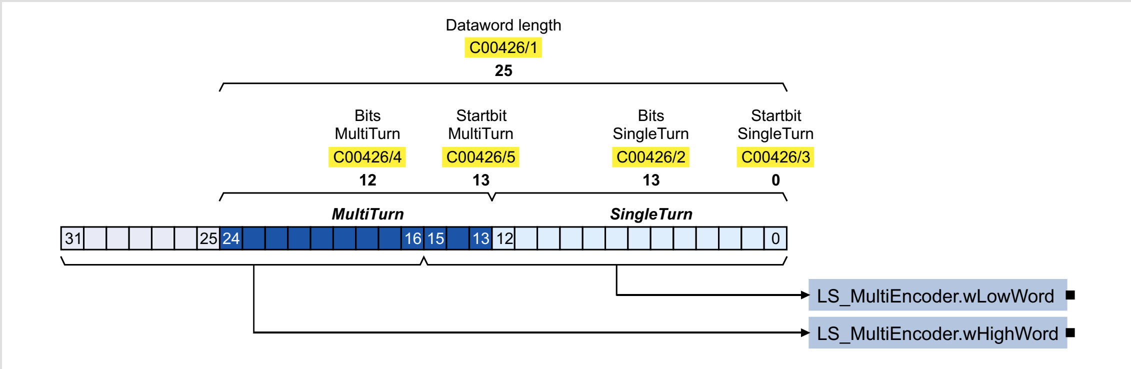
| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| **General** | | | |
| C00428 | SSI encoder: Coding | 0: Binary code | |
| C00427 | SSI encoder: Bit rate | 1: 100 kbps | |
| C01110 | LS_MultiEncoder: Solid measure | 0: rotatively unipolar | |
| C01119/1 | LS_MultiEncoder: Current position | - | units |
| C01119/2 | LS_MultiEncoder: Maximum travel distance | - | units |
| **Data word structure** | | | |
| C00426/1 | SSI encoder: Data word length | 25 | |
| C00426/2 | SSI encoder: Bits SingleTurn | 13 | |
| C00426/3 | SSI encoder: Start bit Singleturn | 0 | |
| C00426/4 | SSI encoder: Bits Multiturn | 12 | |
| C00426/5 | SSI encoder: Start bit Multiturn | 13 | |
| C00426/6 | SSI encoder: Status bit | 0 | |
| C00426/7 | SSI encoder: Left shift of raw value | 0 | |
| **Further settings** | | | |
| C01111/1 | LS_MultiEncoder: Difference - traverse path | 1 | |
| C01111/2 | LS_MultiEncoder: Difference - encoder path | 1 | |
| C01112/1 | LS_MultiEncoder: Position offset | 0.0000 | |
| C01112/2 | LS_MultiEncoder: Data area min | -214748.3647 | |
| C01112/3 | LS_MultiEncoder: Data area max | 214748.3647 | |
| Highlighted in grey = display parameter | | | |

### How to parameterise the SSI encoder:

1. Set the encoder voltage (C00421) of the used SSI encoder.
2. As encoder type (C00422), select "4: Absolute value encoder (SSI)".
3. Set the bit rate (C00427) for SSI communication.
   - For SSI protocols, the permissible baud rate is reduced if the cable length is increased. A safe bit rate must be set, depending on the length of the used encoder cable and the electromagnetic interference level.
4. Set the partitioning of the SSI data word according to the used SSI encoder in C00426/1...5 (see example below).
5. If an SSI encoder with Gray coding is used:
   Select "1: Gray code" in C00428 to convert from Gray to binary.
   - In the Lenze setting, "0: Binary code", conversion is not performed, i.e. an SSI encoder with binary coding is expected.
6. If a linear SSI encoder is used instead of a rotative encoder or the encoder value is to be interpreted with or without sign:
   Set the corresponding setting in C01110.

### Configuration example 1: Rotative encoder "BaumerIVO G0M2H Z04"

Coding: 8192x4096 Gray code

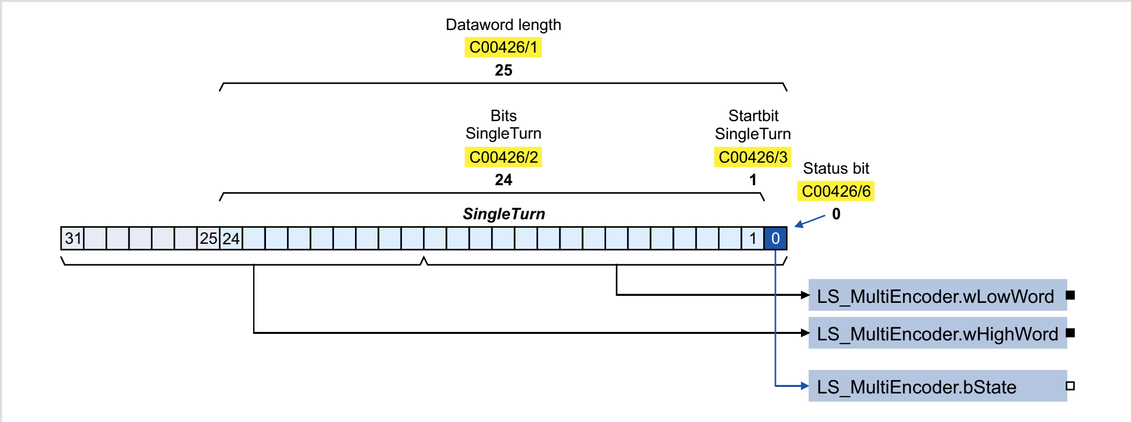



| Parameter | Info | Setting for example 1 |
|---|---|---|
| C00426/1 | SSI encoder: Data word length | 25 |
| C00426/2 | SSI encoder: Bits SingleTurn | 13 |
| C00426/3 | SSI encoder: Start bit Singleturn | 0 |
| C00426/4 | SSI encoder: Bits Multiturn | 12 |
| C00426/5 | SSI encoder: Start bit Multiturn | 13 |
| C00426/6 | SSI encoder: Status bit | 0 |
| C00428 | SSI encoder: Coding | 1: Gray code |
| C01110 | LS_MultiEncoder: Solid measure | 0: rotatively unipolar |

- In "Machine parameters", the feed constant and the gearbox factors for motor and position encoder must be set correctly. (📖 498)
- One encoder revolution is displayed with 65536 increments.

### Configuration example 2: Linear encoder "Sick DME 5000-111"

Coding: Gray code with 25 bits of data word length
(Bit 0 = status bit, bit 1 ...24 = position, resolution 1 bit = 0.1 mm)




| Parameter | Info | Setting for example 2 |
|---|---|---|
| C00426/1 | SSI encoder: Data word length | 25 |
| C00426/2 | SSI encoder: Bits SingleTurn | 24 |
| C00426/3 | SSI encoder: Start bit Singleturn | 1 |
| C00426/4 | SSI encoder: Bits Multiturn | 0 |
| C00426/5 | SSI encoder: Start bit Multiturn | 0 |
| C00426/6 | SSI encoder: Status bit | 0 |
| C00428 | SSI encoder: Coding | 1: Gray code |
| C01110 | LS_MultiEncoder: Solid measure | 1: linearly unipolar |

In "Machine parameters", the feed constant and the gearbox factors for motors and position encoders must be set as follows:

- The position encoder gearbox factor must be set to 1:1.
- When the motor gearbox factor is set to 1:1, the path covered at one motor revolution has to be entered as feed constant in C01204 in [units].
- In case of a different motor gearbox factor, the path covered at one defined "drive roll" revolution has to be entered as feed constant in C01204 in [units].
- The feed of the mechanics entered in C01204 is displayed with 65536 increments.

Scaling of the position value

In order to scale the position change provided by the position encoder at this feed to 65536 as well, enter the traverse path in [units] in C1111/1 and the corresponding difference of the raw value provided by the encoder in C1111/2.

- Only the bits that contain the position data may be taken from the data word for the raw value. If e.g. the bit 0 contains status information, the raw value must be shifted to the right by this bit since only then the SLB of the position value is positioned correctly.

Scaling example:

1 [unit] = 1 mm

Feed constant (C01204) = 4 mm/revolution

Encoder resolution = 50 µm → raw value difference = 20 [digit]/1 [unit]

→ Difference - traverse path (C1111/1) = 1 [unit]

→ Difference - encoder value (C1111/2) = 20 [digit]

Thus, at a feed of the mechanics of 4 mm, the encoder provides a position difference of 65536 increments.

$$\text{Position} = \text{Raw value} \cdot \frac{\text{Encoder feed constant}}{65536}$$

$$\text{with encoder feed constant} = \frac{\text{C01111/1} \cdot 65536^2}{\text{C01111/2} \cdot \text{C01204}}$$

$$\text{Position} = \text{Raw value} \cdot \frac{\text{C01111/1} \cdot 65536}{\text{C01111/2} \cdot \text{C01204}}$$

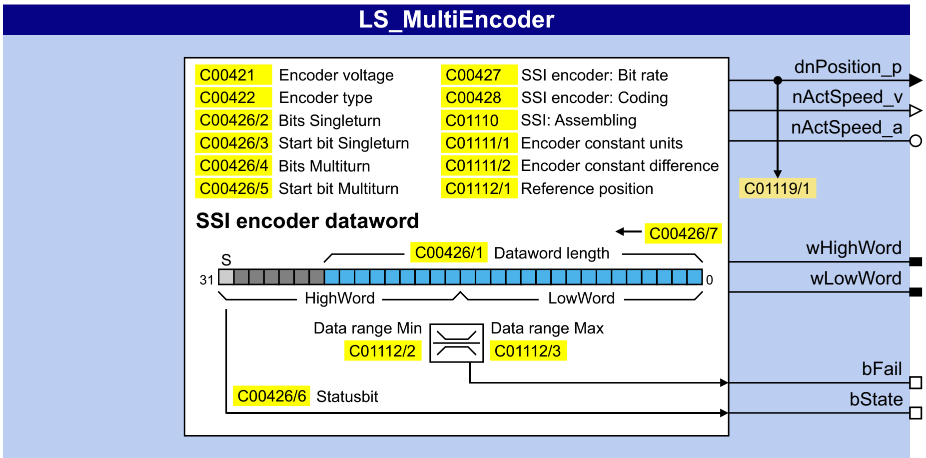
**Left shift of raw value**

The data read-in by the encoder is processed via the settings of the start bits for singleturn and multiturn and the number of bits in such a way that a position signal suitable for the controller is created that can be fed into the function block interconnection.

For evaluating status information that is transmitted by the encoders as well, it is sometimes required to adjust the output at the "raw value" outputs (*wHighWord* and *wLowWord*). For this purpose, a left shift can be set in C00426/7.

## 6.3.2 Internal interfaces | "LS_MultiEncoder" system block

The **LS_MultiEncoder** system block provides the application with speed, position and error information from the encoder in the form of process signals.




### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnPosition_p<br>DINT | Actual position value of the encoder in [increments] |
| nActSpeed_v<br>INT | Actual speed of the encoder in [increments/ms] |
| nActSpeed_a<br>INT | Actual speed of the encoder in [%]<br>• 100 % ≡ reference speed (C00011) |
| wHighWord<br>wLowWord<br>WORD | 32-Bit data telegram of the encoder (raw data)<br>▶ SSI encoder (📖 307) |
| bFail<br>BOOL | Configurable status signal "invalid encoder value"<br>TRUE: The encoder value is outside the value range set in C01112/2 and C01112/3.<br>▶ Recognition of invalid encoder values |
| bState<br>(from version 02.00.00)<br>BOOL | Status bit of the encoder<br>• The bit position of the status bit within the 32-bit data telegram can be set in C00426/6. |

### 6.3.2.1 Recognition of invalid encoder values

For detecting invalid encoder values, a value range can be defined via the parameters C01112/2 and C01112/3. When the encoder signal leaves the set value range, the *bFail* status signal is set to TRUE. Monitoring is only active if the *bSetRef* input is set to TRUE.

- **Example 1: SSI laser distance meter**
  The *bFail* status signal can be used to detect an interruption of the laser beam or a dirty mirror.
- **Example 2: SSI bar code scanner**
  In an application, an SSI bar code scanner is used to detect the workpiece ID and the workpiece position. A setting of sensible limit values serves to use the *bFail* status signal to determine whether the ID or the position of the SSI bar code scanner has been detected.

### 6.3.2.2 Setting of a static position offset

If, for example, the encoder cannot be mounted mechanically in order that the supplied position matches the mechanics or if for other reasons, a static position offset is required, this value can be set in C01112/1.

## 6.4 HTL encoder at DI1/DI2 or DI6/DI7

### 6.4.1 Parameterising digital inputs as encoder inputs

The function of the digital inputs DI1/DI2 and DI6/DI7 is defined via C00115/1...2.

To be able to use the digital inputs as encoder inputs, select 2, 3, or 4 (Lenze recommendation: 2) in C00115/1 or C00115/2, depending on the input terminals used.

| Selection in C00115/1...2 | Function |
|---|---|
| 2: DI1(6)&DI2(7)=FreqIn (2-track) | DI1/6 and DI2/7 = 2-track frequency input<br>• Permits a two-track evaluation of the encoder including correct detection of the direction of rotation. |
| 3: DI1(6)=FreqIn / DI2(7)=Direction | DI1/6 = 1-track frequency input<br>DI2/7 = specification of direction |
| 4: DI1(6)=CountIn / DI2(7)=In | DI1/6 = counter input<br>DI2/7 = digital input |

**Danger!**

- For single-track evaluation, make sure that the sign is correctly specified. Otherwise, the motor may overspeed.
- If an HTL encoder is used at the digital input terminals:
  Observe the maximum input frequencies of the digital inputs!
  – DI1/DI2: max. 200 kHz
  – DI6/DI7: max. 10 kHz

**Note!**

If the digital inputs are parameterised as encoder inputs, the corresponding output signals (*bIn1*/*bIn2* and *bIn6*/*bIn7*) at the LS_DigitalInput system block are automatically set to FALSE.

Wiring diagram, assignment and electrical data of the digital input terminals can be found in the **8400 hardware manual** in the chapter "Technical data".

- The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 drive controller.

**General procedure**

(if the encoder is connected to the digital inputs DI1 and DI2)

1. Define the function of the digital inputs DI1 and DI2 in C00115/1.
2. Set the encoder increments in C00420/1.
3. Select "1: Encoder signal FreqIn12" in C00495/1.
4. Adapt the filter time of the speed measurement in C00497/1.
5. In the case of encoders with a very low resolution (number of increments < 120 increments):
   Change the encoder evaluation procedure in C00496 if necessary.

**Related topics:**

▶ Digital input terminals (📖 329)
▶ Using DI1(6) and DI2(7) as frequency inputs (📖 333)

### 6.4.2 Encoder evaluation method

Depending on the encoder used, the following table specifies which evaluation method should be selected in C00496:

| Selection in C00496 | Encoder evaluation method |
|---|---|
| 0: High-resolution encoder | High-precision procedure for high-resolution encoders (>=512 increments)<br>• Method for speed measurement with automatic scan time setting (0.5 ... 500 ms).<br>• Evaluation with automatic scan time minimisation for an optimum dynamic performance.<br>• Particularly suited for high-resolution encoders (≥ 1024 inc) with good signal quality, i.e.<br>– good scanning ratio 1:1<br>– exactly 90°-phase offset between track A and B (error ≤ ± 10°)<br>• Not suited for encoders with poor signal quality.<br>• Wiring according to EMC (e.g. motor and encoder cable shielding) is required! |
| 1: Low-resolution encoder (StateLine) | High-precision procedure for low-resolution encoders (<=128 increments)<br>• Exact method for speed measurement with automatic scan time setting (0.5 ... 500 ms) for low-resolution encoders in the range of 4 .... 128 increments.<br>• Evaluation with automatic scan time minimisation for an optimum dynamic performance.<br>• Method is also suited for encoders with poor signal quality, e.g. for encoders with high error rate in scanning ratio and phase offset.<br>• This method requires an equidistant period length per encoder increment.<br>• Wiring according to EMC (e.g. motor and encoder cable shielding) is required! |
| 2: Comb. encoder method (Lenze setting) | Combination of the first two procedures as a function of the speed (recommended procedure)<br>• For a high-precision speed measurement suited for encoders with an arbitrary number of increments (4 ... 1024 increments).<br>• Low input frequencies at the encoder inputs:<br>The method is used for low-resolution encoders.<br>• High input frequencies at the encoder inputs:<br>The method is used for high-resolution encoders.<br>• This method is suited for encoders with average to good signal quality.<br>• Evaluation with automatic scan time minimisation for an optimum dynamic performance.<br>• This method requires an equidistant period length per encoder increment.<br>• Wiring according to EMC (e.g. motor and encoder cable shielding) is required! |
| 3: Edge-counting procedure | Simple edge counting procedure with adjustable scanning time (C00425)<br>• Speed measurement by means of the edges of tracks A and B measured per scanning interval.<br>• Integrated correction algorithm for EMC interference.<br>• Limited suitability for systems with unshielded encoder and/or motor cable.<br>• Limited suitability for encoders with poor signal quality, i.e. high error rate in scanning ratio and phase offset. |

**Tip!**

- We recommend to use the preset combined encoder method (C00496 = 2).
- Use one of the first three procedures (C00496 = 0, 1, or 2) for dynamic applications (e.g. operating mode: servo control).
- For dynamic speed control or positioning processes, use an HTL encoder with 1024 increments.

### 6.4.3 HTL encoder at DI1/DI2

**Note!**

At the digital terminals DI1 and DI2, only encoders with HTL level can be used.

In spite of the selected operating mode without encoder feedback, the actual speed value (C00051) is calculated if an encoder is connected and "1: Encoder signal FrqIn12" is selected in C00495.

**Low speeds (except for edge counting)**

For the first three methods (C00496 = 0, 1, or 2), the minimum speed that can be measured depends on the encoder resolution.

The quantisation error

- is independent of the encoder resolution,
- exclusively depends on the encoder quality (encoder errors).
- at least amounts to 0.5 rpm.

Internal arithmetic operations automatically maintain the minimally required value of the scanning time in order to achieve maximum dynamics.

| Encoder resolution (Number of increments) | Min. measurable speed in [rpm] |
|---|---|
| 8 | 16 |
| 16 | 8 |
| 32 | 4 |
| 64 | 2 |
| 128 | 1 |
| 256 | 0.5 |
| ≥ 512 | 0.25 |

### Low speeds with edge counting

The minimum speed that can be measured and the quantisation error of speed measurement in the edge-counting procedure (C00496 = 3) depend on the scanning time that can be set in C00425/1 and the encoder resolution.

Depending on accuracy and the requirements with regard to the dynamic performance, the respective scanning time must be selected and set in C00425/1:

| Encoder resolution (Number of increments) | Scanning time [ms] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 5 | 10 | 20 | 50 | 100 | 200 | 500 | 1000 |
| | Min. measurable speed in [rpm] | | | | | | | | | |
| **8** | 1875 | 938 | 375 | 188 | 93.8 | 37.5 | 18.8 | 9.4 | 3.8 | 1.9 |
| **16** | 938 | 469 | 188 | 94 | 46.9 | 18.8 | 9.4 | 4.7 | 1.9 | 0.9 |
| **32** | 469 | 234 | 94 | 46.9 | 23.4 | 9.4 | 4.7 | 2.3 | 0.9 | 0.5 |
| **64** | 234 | 117 | 46.9 | 23.4 | 11.7 | 4.7 | 2.3 | 1.2 | 0.5 | 0.2 |
| **128** | 117 | 58.6 | 23.4 | 11.7 | 5.9 | 2.3 | 1.2 | 0.6 | 0.2 | 0.12 |
| **256** | 58.6 | 29.3 | 11.7 | 5.9 | 2.9 | 1.2 | 0.6 | 0.3 | 0.12 | 0.06 |
| **512** | 29.3 | 14.6 | 5.9 | 2.9 | 1.5 | 0.6 | 0.3 | 0.15 | 0.06 | 0.03 |
| **1024** | 14.6 | 7.3 | 2.9 | 1.5 | 0.7 | 0.3 | 0.15 | 0.07 | 0.03 | 0.01 |

### 6.4.4 HTL encoder at DI6/DI7

**Note!**

Single-track evaluation of the digital DI6/DI7 terminals as speed feedback (C0115/2 = 1 or 3) is not possible. Hence, a single-track encoder cannot be used for speed control at the digital DI6 terminal!

#### Low speeds with edge counting

The speed measurement is evaluated at the digital terminals DI6/DI7 with the edge-counting procedure and a fixed scanning time that can be set in C00425/2.

The minimum speed that can be measured and the quantisation error of speed measurement in the edge-counting procedure depend on the scanning time that can be set in C00425/2 and the encoder resolution.

Depending on accuracy and the requirements with regard to the dynamic performance, the respective scanning time must be selected and set in C00425/2:

| Encoder resolution (Number of increments) | Scanning time [ms] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 5 | 10 | 20 | 50 | 100 | 200 | 500 | 1000 |
| | Min. measurable speed in [rpm] | | | | | | | | | |
| **8** | 1875 | 938 | 375 | 188 | 93.8 | 37.5 | 18.8 | 9.4 | 3.8 | 1.9 |
| **16** | 938 | 469 | 188 | 94 | 46.9 | 18.8 | 9.4 | 4.7 | 1.9 | 0.9 |
| **32** | 469 | 234 | 94 | 46.9 | 23.4 | 9.4 | 4.7 | 2.3 | 0.9 | 0.5 |
| **64** | 234 | 117 | 46.9 | 23.4 | 11.7 | 4.7 | 2.3 | 1.2 | 0.5 | 0.2 |
| **128** | 117 | 58.6 | 23.4 | 11.7 | 5.9 | 2.3 | 1.2 | 0.6 | 0.2 | 0.12 |
| **256** | 58.6 | 29.3 | 11.7 | 5.9 | 2.9 | 1.2 | 0.6 | 0.3 | 0.12 | 0.06 |
| **512** | 29.3 | 14.6 | 5.9 | 2.9 | 1.5 | 0.6 | 0.3 | 0.15 | 0.06 | 0.03 |
| **1024** | 14.6 | 7.3 | 2.9 | 1.5 | 0.7 | 0.3 | 0.15 | 0.07 | 0.03 | 0.01 |

#### Maximum speeds with edge counting

Due to the lower input frequency (max. 10 kHz) compared to the terminals DI1/DI2, operation with maximum speed at the terminals DI6/DI7 is limited. ▶ Digital input terminals (📖 329)

| Encoder resolution (Number of increments) | Max. measurable speed in [rpm] |
|---|---|
| **8** | No restrictions |
| **16** | 37500 |
| **32** | 18750 |
| **64** | 9375 |
| **128** | 4688 |
| **256** | 2344 |
| **512** | 1172 |
| **1024** | 586 |

## 6.5 Monitoring

### 6.5.1 Resolver/multi-encoder open-circuit monitoring

Open-circuit monitoring is available for the resolver cable and the encoder cable.

**Danger!**

- For safety reasons, always select "Fault" as a response for the (open-circuit) monitoring of the resolver/encoder!
- To avoid interference when using an encoder, only use shielded motor and encoder cables!

The encoders to be monitored are selected in C00498:

| Selection | Function |
|---|---|
| 0: Speed encoder and position encoder | Open-circuit monitoring is active for the speed encoder selected in C00495 and the position encoder selected in C00490.<br>• Depending on whether a speed encoder or position encoder has been selected, the monitoring mode for the resolver and/or encoder is activated.<br>• If no speed encoder and position encoder have been selected, open-circuit monitoring is deactivated. |
| 1: Resolver only | Open-circuit monitoring is only active for the resolver, independent on the selection of the speed encoder and position encoder.<br>• Open-circuit monitoring for the encoder is deactivated. |
| 2: Encoder only | Open-circuit monitoring is only active for the encoder, independent on the selection of the speed encoder and position encoder.<br>• Open-circuit monitoring for the resolver is deactivated. |
| 3: Resolver and encoder | Open-circuit monitoring is active for the resolver and encoder, independent on the selection of the speed encoder and position encoder. |

**Tip!**

We recommend the Lenze setting, "0: Speed sensor and position encoder". Further possible settings make sense if the respective encoder is not used as speed sensor or position encoder but for other tasks in the application. Moreover, a selective switch-off of the monitoring is possible if other settings are made.

### When does the open-circuit monitoring system respond?

| Resolver | Multi encoder |
|---|---|
| • In case of open circuit in the encoder cable.<br>• If the resolver impedance is too high.<br>• In case of EMC interferences. | • In case of open circuit in the encoder cable. |

### Response to open circuit

| Resolver | Multi encoder |
|---|---|
| The error response set in C00603/2 is triggered (Lenze setting: "0: No Reaction").<br>Logbook entry: "SD2: Open circuit - resolver" | The error response set in C00603/1 is triggered (Lenze setting: "0: No Reaction").<br>Logbook entry: "SD4: Open circuit - MultiEncoder" |

### 6.5.2 Open-circuit monitoring HTL encoder

Open-circuit monitoring is available for the HTL encoder cable.

**Danger!**

For safety reasons, always select "Fault" (Lenze setting) as a response for the (open-circuit) monitoring of the HTL encoder!

**Note!**

If you do not use an HTL encoder, deactivate the monitoring (C00586 = "0: No Reaction").

**When does the open-circuit monitoring system respond?**

The open-circuit monitoring will trigger if

- an open circuit occurs in the encoder cable.
- extreme overload (e.g. a blocked motor shaft) occurs in the starting phase of the motor.
- highly dynamic reversion of the motor occurs.

**Which measured values lead to an actuation of the open-circuit monitoring system?**

The following measured values checked for plausibility lead to an actuation of the open-circuit monitoring system:

1. If the total deviation between actual speed and setpoint speed is higher than f = 40 Hz for a time > 0.1 s .
2. If the actual speed detected is f = 0 Hz or n = 0 rpm and the $I_{max}$ controller or the torque limitation for servo control (SC) is active for t ≥ 0.1 s.
3. If the sign of the injected frequency and the actual speed is not the same, the $I_{max}$ controller is active and this status is active for 0.1 s. Usually this is the case when A/B tracks are reversed.

**Response to open circuit**

- If the open-circuit monitoring is tripped:
  - The error response set in C00586 is activated (Lenze setting: "Fault").
  - The "SD3: Open circuit - feedback system" is entered into the Logbook.
  - The *bMctrlEncoderComFault* status output of the LS_DeviceMonitor SB is set to TRUE.
- A setting of C00586 = "0: No Reaction" deactivates the monitoring.

### 6.5.3 Motor temperature monitoring (KTY)

The motor temperature can also be detected and monitored via the resolver cable and/or the encoder cable.

- The feedback system to be used for the motor temperature is selected in C01193.
- The respective motor temperature monitoring includes an early warning stage as well as an open-circuit monitoring for the thermal sensor.

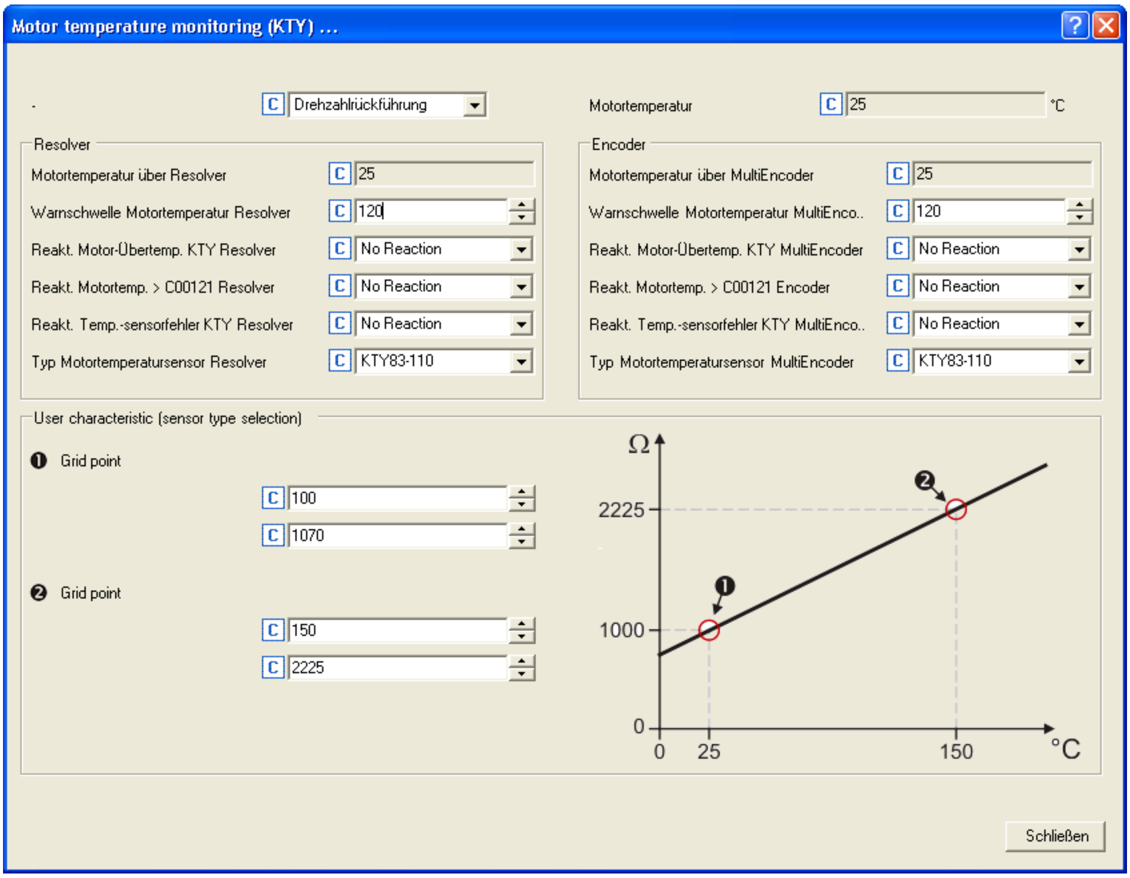
**Note!**

In the Lenze setting of C00583/1...6, the motor temperature monitoring functions are switched on! These monitoring functions are only active with speed encoder selection "3: Multi encoder" or "4: Resolver" in C00495.

From version V11.00.00, the servo control (SC) the temperature compensation within the motor control is activated in the Lenze setting of C02878/1. However, temperature compensation is only active with speed sensor selection "3: Multi encoder" or "4: Resolver" in C00495 as well as error-free KTY temperature detection (display in C00063/1 ≠ 255 °C).

#### Parameterisation dialog in the »Engineer«

Click the **Motor temperature monitoring (KTY) ...** button in the *Encoder/feedback system* dialog box to access the parameterisation dialog:



### Short overview of the relevant parameters:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01193 | Feedback system motor temperature | 0: Speed feedback | |
| C00063/1 | Motor temperature | - | °C |
| **Resolver** | | | |
| C00063/2 | Motor temperature via resolver | - | °C |
| C00121/1 | Warning threshold motor temp. to resolver | 120 | °C |
| C00583/1 | Motor overtemperature KTY resolver | 1: Fault | |
| C00583/3 | Motor temp. > C00121 resolver | 5: Warning | |
| C00583/5 | Thermal sensor motor KTY resolver | 1: Fault | |
| C01190/1 | Thermal motor sensor to resolver | 0: KTY83-110 | °C |
| **Encoder** | | | |
| C00063/2 | Motor temperature via encoder | - | °C |
| C00121/2 | Warning threshold motor temp. to encoder | 120 | °C |
| C00583/2 | Motor overtemperature KTY encoder | 1: Fault | |
| C00583/4 | Motor temp. > C00121 encoder | 5: Warning | |
| C00583/6 | Thermal sensor motor KTY encoder | 1: Fault | |
| C01190/2 | Thermal motor sensor to encoder | 0: KTY83-110 | °C |
| **User characteristic** | | | |
| C01191/1 | PTC characteristic: Temperature 1 | 100 | °C |
| C01191/2 | PTC characteristic: Temperature 2 | 150 | °C |
| C01192/1 | PTC characteristic: Resistance 1 | 1070 | Ohm |
| C01192/2 | PTC characteristic: Resistance 2 | 2225 | Ohm |
| Highlighted in grey = display parameter | | | |

### Motor temperature monitoring (KTY) via resolver

- If the winding temperature detected via the motor temperature sensor exceeds the limit value set in C00121/1 (Lenze setting: 120 °):
  - An advance warning in the form of the error response set in C00583/3 is issued.
  - The "OH7: Motor temperature - resolver > C121" error message is entered into the logbook.
- If the fixed limit value of 150 °C is exceeded:
  - The error response set in C00583/1 is triggered.
  - The "OH9: Motor temperature - resolver" error message is entered into the logbook.
  - The *bFail* status output of the LS_Resolver SB is set to TRUE.
- If open circuit is detected for the motor temperature sensor:
  - The error response set in C00583/5 is triggered.
  - The "SD6: Thermal detector error - resolver" error message is entered into the logbook.
  - The *bFail* status output of the LS_Resolver SB is set to TRUE.

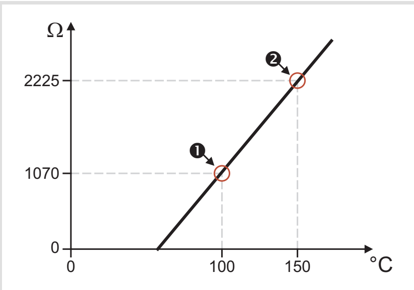
### Motor temperature monitoring (KTY) via encoder

- If the winding temperature detected via the motor temperature sensor exceeds the limit value set in C00121/2 (Lenze setting: 120 °):
  - An advance warning in the form of the error response set in C00583/4 is issued.
  - The "OH6: Motor temperature - MultiEncoder > C121" error message is entered into the logbook.
- If the fixed limit value of 150 °C is exceeded:
  - The error response set in C00583/2 is triggered.
  - The "OH12: Motor overtemperature - MultiEncoder" error message is entered into the logbook.
  - The *bFail* status output of the LS_MultiEncoder SB is set to TRUE.
- If open circuit is detected for the motor temperature sensor:
  - The error response set in C00583/6 is triggered.
  - The "SD12: Thermal detector error - MultiEncoder" error message is entered into the logbook.
  - The *bFail* status output of the LS_MultiEncoder SB is set to TRUE.

### Specific characteristic for the motor temperature sensor

A specific characteristic can be defined and activated for the motor temperature sensor if required.

- The specific characteristic is defined on the basis of two grid points which must be set in C01191 and C01192. Those two points define a line which is extrapolated to the right and to the left.
- The specific characteristic can be activated by selecting "1: Spec. characteristic" in C01190/1 for the resolver and in C01190/2 for the encoder.
- In the Lenze setting, the specific characteristic is defined as follows:




- Grid point ❶
  - C01191/1 = 100 °C
  - C01192/1 = 1070 Ω
- Grid point ❷
  - C01191/2 = 150 °C
  - C01192/2 = 2225 Ω

[6-2] Lenze setting of the specific characteristic

**Note!**

If a motor is selected from the motor catalogue, parameters C01190, C01191, and C01192 are overwritten!

### 6.5.4 Encoder - angular drift monitoring

This function extension is available from version 02.00.00!

**Note!**

In the Lenze setting (C00603/3 = "0: No response"), the angular drift monitoring of the encoder is not activated!

The encoder angular drift monitoring monitors a possible deviation of the real encoder angle from the angle calculated by counting increments in the encoder evaluation.

- If a deviation higher than 45° (electrical) is recognised when monitoring is activated:
  - The error response set in C00603/3 is triggered.
  - The "SD8: Encoder angular drift monit." error message is entered into the logbook.
  - The "Reference known" status of the "Homing" basic drive function is reset (if this status was set before)

**Tip!**

A deviation may occur, for instance, by incorrect parameter setting of the encoder increments, by lines in the form of interferences caused by EMC or loss of lines caused by EMC.

The encoder angular drift monitoring for encoders with and without absolute information is implemented by two different principles which are explained in detail in the following subchapters.

#### 6.5.4.1 Angular drift monitoring for encoders without absolute information

For an encoder without absolute information, the number of incoming encoder lines between two zero pulses (one revolution) is monitored. This value must equal the encoder increments set in C00420.

**Note!**

After mains switching, monitoring is only active after second incoming zero pulse since the first line difference to be used can only be calculated with the second and first zero pulse.

When the motor (and thus the encoder) is replaced, it is very likely that a angular drift error occurs within the first revolution after acknowledging the encoder error since the monitoring function cannot recognise that the encoder has been replaced.

### 6.5.4.2 Angular drift monitoring for encoders with absolute information

For an encoder with absolute information, cyclical communication with the encoder takes place and the angle is read out digitally. This angle is compared to the angle from the encoder evaluation.

**Note!**

If monitoring is deactivated (C00603/3 ="0: No Reaction"), no cyclical communication with the encoder takes place and thus no communication errors with the encoder can occur.

If monitoring is activated, it is only executed for speeds lower than 100 rpm due to runtimes for communication.

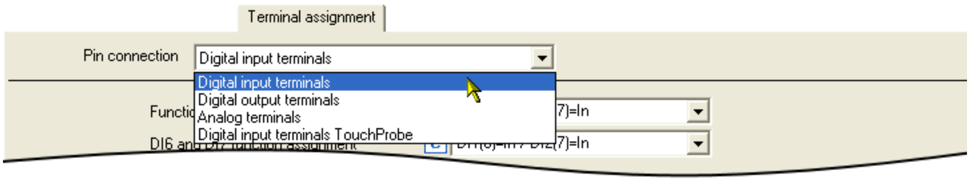
- If increments get lost at higher speeds, this deviation can only be recognised when the speed falls below 100 rpm for at least 80 ms.

After each detected encoder angular drift error, a renewed read-out of the position is tripped automatically and this angle is written into the encoder evaluation. This makes it possible to acknowledge the error. In case of synchronous machines, the pole position is corrected simultaneously.

# 7 I/O terminals

This chapter provides information on the function, possible parameter settings, and technical data of the input/output terminals of the controller.

In the »Engineer«, the input and output terminals are parameterised on the **Terminal assignment** tab. To do this, go to the **Control terminals** list field and select the terminals that you wish to parameterise:




You can find further information in the respective subchapter:

- Digital input terminals (📖 329)
- Digital output terminals (📖 349)
- Analog terminals (📖 352)
- Touch probe detection (📖 362)

**Note!**

The input and output terminals of the drive controller have already been functionally assigned in the default setting ("Lenze setting"). The preconfigured assignment depends on the technology application selected in C00005 and the control mode selected in C00007:

- **TA "Actuating drive speed":** Terminal assignment of the control modes (📖 394)
- **TA "Table positioning":** Terminal assignment of the control modes (📖 424)
- **TA "Switch-off positioning":** Terminal assignment of the control modes (📖 451)

Wiring diagram, assignment and electrical data of the input and output terminals can be found in the **8400 hardware manual** in the chapter "Technical data".

- The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 drive controller.

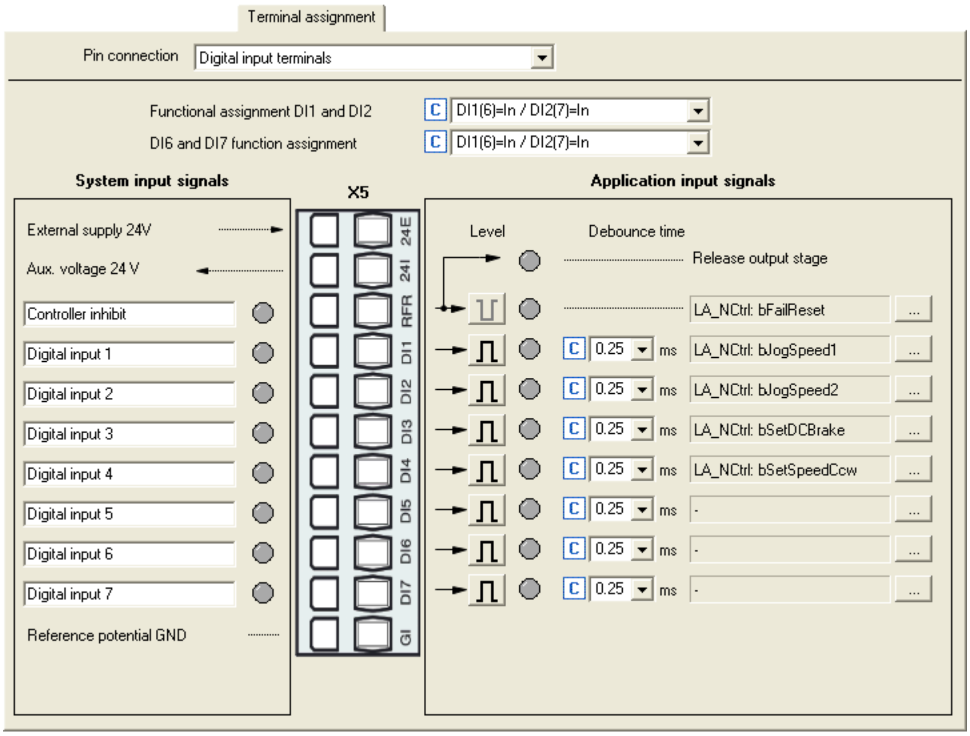
**Tip!**

How you can alter the preconfigured assignment of the input and output terminals is described in the chapter entitled "User-defined terminal assignment". (📖 370)

## 7.1 Digital input terminals

The drive controller has 7 parameterisable input terminals (DI1 ... DI7) for detecting digital signals. The RFR control input for enabling the controller is permanently connected to the device control unit.

### Parameterisation dialog in the »Engineer«:




| Button | Function |
|---|---|
| ⎍ | Indicates the polarity of the input is HIGH active.<br>The polarity can be changed from HIGH active to LOW active by clicking this button. |
| ⊔ | Indicates that the polarity of the input is LOW active.<br>The polarity can be changed from LOW active to HIGH active by clicking this button. |
| ... | Open the parameterising dialog for assigning application inputs to the digital input.<br>▸ Changing the terminal assignment with the »Engineer« (🕮 374) |

### Short overview of parameters for the digital input terminals:

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C00115/1 | Fct. DI 1/2 200kHz<br>▸ Change function assignment (📖 331) | 0: DI1(6)=In / DI2(7)=In | |
| C00115/2 | Fct. DI 6/7 10kHz<br>▸ Change function assignment (📖 331) | 0: DI1(6)=In / DI2(7)=In | |
| **Digital inputs DI1 ... DI7** | | | |
| C00114 | DigInX: Inversion | Bit coded | |
| C02830/1...7 | DI1...DI7: Debounce time | 1: 0.25 | |
| C00443/1 | DIx: Terminal level | - | |
| C00443/2 | DIx: Output level | - | |
| Highlighted in grey = display parameter | | | |

### Related topics:

▸ Touch probe detection (📖 362)
▸ User-defined terminal assignment (📖 370)

### 7.1.1 Change function assignment

The internal processing function of the digital input terminals DI1/DI2 und DI6/DI7 can be reconfigured in C00115 if necessary. In this way, these input terminals can alternatively be used as frequency or counting inputs in order to implement the following functions:

- Detection of the input frequency
- Detection and processing of two unipolar input frequencies to one bipolar frequency
- Counting of input pulses
- Evaluation of the speed feedback (HTL encoder) for the motor control (speed-controlled operation)

| C00115/1: Function assignment of DI1 and DI2<br>C00115/2: Function assignment of DI6 and DI7 | | Function assignment<br>DI1 / DI6 | <br>DI2 / DI7 |
|---|---|---|---|
| 0 | **DI1(6)=In / DI2(7)=In** | **Digital input** | **Digital input** |
| 1 | DI1(6)=FreqIn / DI2(7)=In | Frequency input | Digital input |
| 2 | DI1(6)&DI2(7)=FreqIn (2-track) | Frequency input (2-track) | |
| 3 | DI1(6)=FreqIn / DI2(7)=direction | Frequency input (speed) | Frequency input (direction) |
| 4 | DI1(6)=CountIn / DI2(7)=In | Count input | Digital input |

**Note!**

- In the Lenze setting of C00115, the digital input terminals DI1/DI2 and DI6/DI7 have been configured as "normal" digital inputs.
- The digital input terminals DI3 ... DI5 are basically designed as "normal" digital inputs.
- Very high pulse frequencies can be measured at the DI1/DI2 and DI6/DI7 input terminals if the latter have been configured as frequency or counting inputs in C00115. Scanning is then carried out within less than µs instead of the otherwise usual scanning rate of 1 kHz (1 ms).

You can find detailed information on the respective function assignment in the following subchapters:

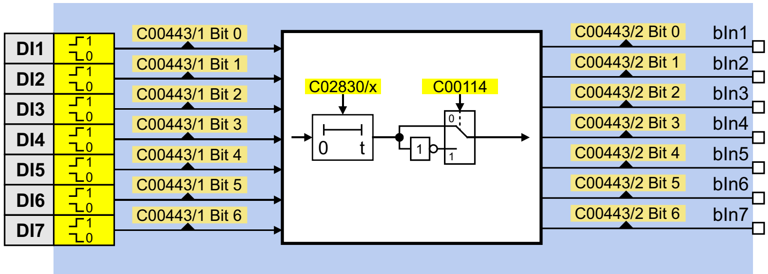
- Using DI1(6) and DI2(7) as digital inputs (📖 332)
- Using DI1(6) and DI2(7) as frequency inputs (📖 333)
- Using DI1(6) as counting input (📖 338)

### 7.1.1.1 Using DI1(6) and DI2(7) as digital inputs

**Function assignment 0: DI1(6)=In / DI2(7)=In**

With this setting in C00115 , the digital input terminals have been configured as "normal" digital inputs.

- For each digital input, the debounce time (C02830/1...7) and the terminal polarity (C00114) can be set individually.
- The current terminal level at the input of the internal processing function is shown in C00443/1 in bit-coded form.
- The output level for the application is shown in C00443/2 in bit-coded form.




**Internal interfaces to the application**

- Relevant outputs at the LS_DigitalInput system block:

| Output<br>DIS code \| data type | Value/meaning |
|---|---|
| bIn1 ... bIn7<br>C00443/2 \| BOOL | Digital input DI1 ... DI7 |

**Related topics:**

- Using DI1(6) and DI2(7) as frequency inputs (📖 333)
- Using DI1(6) as counting input (📖 338)
- Internal interfaces | System block "LS_DigitalInput" (📖 342)

### 7.1.1.2 Using DI1(6) and DI2(7) as frequency inputs

**General information on using the input terminals as frequency inputs**

The frequency inputs serve to detect HTL encoders with any number of increments and single-track and two-track signals. Single-track signals can be evaluated with or without rotation signal.

**Note!**

- Make sure that, when motor control with speed feedback is in use, the maximum input frequency of the respective input terminal is not exceeded.
  - DI1/DI2: $f_{max}$ = 100 kHz (from version 02.00.00: $f_{max}$ = 200 kHz)
  - DI6/DI7: $f_{max}$ = 10 kHz
- If the encoder signal is used as an actual speed value:
  Number of encoder pulses / revolution ≤ **8192!**

Example of DI6/DI7 (in accordance with the preceding note):

- Number of encoder increments: 512 pulses / motor revolution
- Reference speed (C00011): 1500 rpm
- Speed setpoint: 100 %

$$\text{Input frequency} = \frac{1500\text{ rpm}}{60\text{ s}} \times 512\text{ pulse} = 12800\text{ pulse/s} = 12.8\text{ kHz}$$

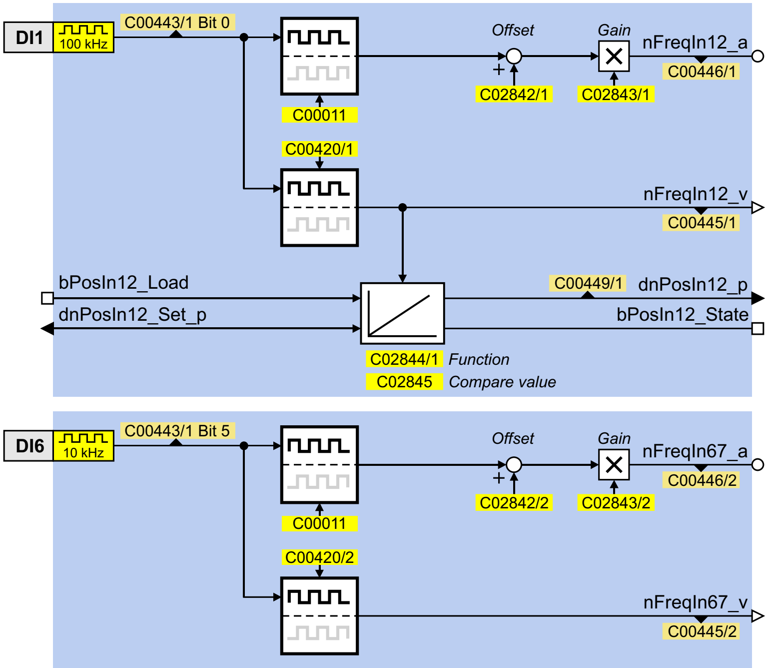
- Result: The speed or the number of increments is too high!

**Tip!**

The LS_DigitalInput system block can also provide the encoder position. Detailed information on this topic is provided in chapter "Output of the encoder position of the DI1/DI2 frequency input". (🕮 345)

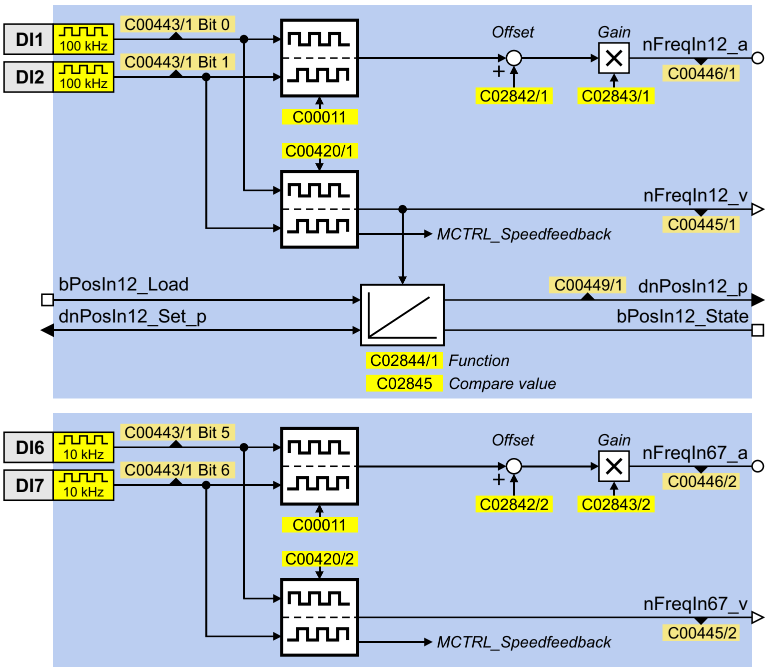
### Function assignment 1: DI1(6)=FreqIn / DI2(7)=In

This setting in C00115 configures the input terminal DI1 or DI6 as frequency input. The input terminal DI2 or DI7 remains configured as "normal" digital input..



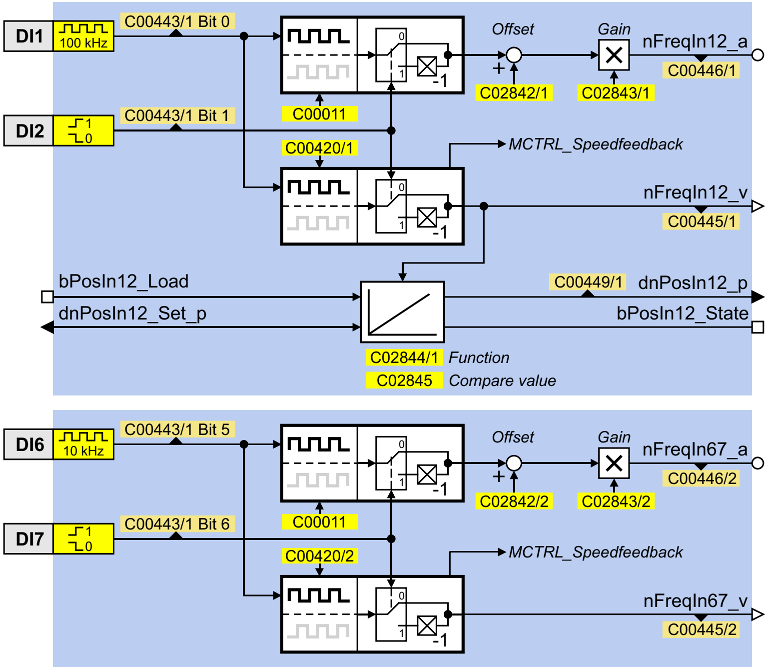

### Function assignment 2: DI1(6)&DI2(7)=FreqIn (2-track)

This setting in C00115 can be used to connect a two-track encoder to the DI1/DI2 or DI6/DI7 terminals.



#### Function assignment 3: DI1(6)=FreqIn / DI2(7)=Direction

This setting in C00115 can be used to connect a single-track encoder to terminals DI1/DI2 or DI6/DI7. For this purpose, the rotation speed is evaluated via terminal DI1(6) and the direction of rotation of the encoder (LOW level ≡ CW direction of rotation) is evaluated via the DI(7) terminal.



### Short overview of the parameters for the frequency inputs:

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C00011 | Appl.: Reference speed | 1500 | rpm |
| **Frequency input DI1/DI2** | | | |
| C00115/1 | Fct. DI 1/2 200kHz | 0: DI1(6)=In / DI2(7)=In | |
| C00420/1 | Encoder increments at FreqIn12 | 128 | Incr./rev. |
| C02842/1 | FreqIn12: Offset | 0.00 | % |
| C02843/1 | FreqIn12: Gain | 100.00 | % |
| C02844/1 | PosIn12: Function | Loading with level | |
| C02845 | PosIn12: Comparison value | 0 | |
| C00443/1 | DIx: Terminal level | - | |
| C00445/1 | FreqIn12_nOut_v | - | Incr/ms |
| C00446/1 | FreqIn12_nOut_a | - | % |
| C00449/1 | FreqIn12_dnOut_p | - | Incr |
| **Frequency input DI6/DI7** | | | |
| C00115/2 | Fct. DI 6/7 10kHz | 0: DI1(6)=In / DI2(7)=In | |
| C00420/2 | Encoder increments at FreqIn67 | 128 | Incr./rev. |
| C02842/2 | FreqIn67: Offset | 0.00 | % |
| C02843/2 | FreqIn67: Gain | 100.00 | % |
| C00443/1 | DIx: Terminal level | - | |
| C00445/2 | FreqIn67_nOut_v | - | Incr/ms |
| C00446/2 | FreqIn67_nOut_a | - | % |
| Highlighted in grey = display parameter | | | |

### Internal interfaces to the application

▶ Relevant inputs at LS_DigitalInput system block:

| Input<br>Data type | Information/possible settings | |
|---|---|---|
| **Frequency input DI1/DI2** | | |
| bPosIn12_Load<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnPosIn12_Set_p* and *bPosIn12_State* is reset to FALSE. |
| dnPosIn12_Set_p<br>DINT | Starting value for angle integrator | |

▶ Relevant outputs at the LS_DigitalInput system block:

| Output<br>Data type | Value/meaning | |
|---|---|---|
| **Frequency input DI1/DI2** | | |
| nFreqIn12_a<br>C00446/1 \| INT | Output frequency as scaled analog signal in [%] | |
| nFreqIn12_v<br>C00445/1 \| INT | Output frequency as speed signal in [inc/ms] | |
| dnPosIn12_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bPosIn12_State*) | |
| bPosIn12_State<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bPosIn12_Load*. | |
| | TRUE | Overflow has occurred or distance is processed. |
| **Frequency input DI6/DI7** | | |
| nFreqIn67_a<br>C00446/2 \| INT | Output frequency as scaled analog signal in [%] | |
| nFreqIn67_v<br>C00445/2 \| INT | Output frequency as speed signal in [inc/ms] | |

### Related topics:

▸ Output of the encoder position of the DI1/DI2 frequency input (📖 345)
▸ Using DI1(6) and DI2(7) as digital inputs (📖 332)
▸ Using DI1(6) as counting input (📖 338)
▸ Internal interfaces | System block "LS_DigitalInput" (📖 342)

### 7.1.1.3 Using DI1(6) as counting input

**General information on use as a counting input**

The counting input is used for counting fast edges. A 32-bit counter counts from a parameterisable starting value up to a parameterisable comparison value and then outputs a corresponding status signal.

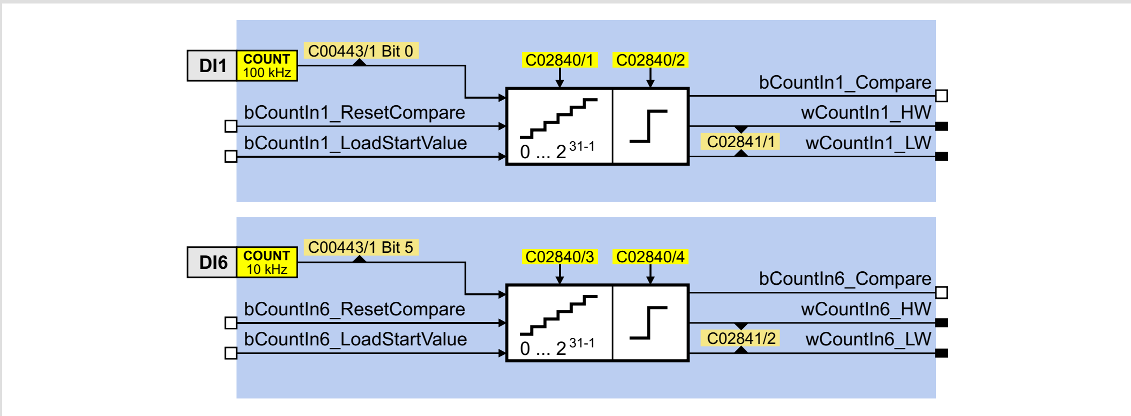
▶ Possible counting range: 0 ... $2^{31}$ - 1 (0 ... 2147483647)

**Note!**

- The starting value must be have been set so that it is smaller than the comparison value. Otherwise, the counter will be kept at the starting value because the condition "Count value ≥ Comparison value" has been satisfied.
- Note the maximum input frequency of the respective input terminal:
  - DI1: $f_{max}$ = 200 kHz
  - DI6: $f_{max}$ = 10 kHz

**Function assignment 4: DI1(6)=CountIn / DI2(7)=In**

This setting in C00115 configures the input terminal DI1 or DI6 as counting input. The input terminal DI2 or DI7 remains configured as "normal" digital input..



### Short overview of parameters for the counting inputs:

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| **Counting input DI1** | | | |
| C00115/1 | Fct. DI 1/2 200kHz | 0: DI1(6)=In / DI2(7)=In | |
| C00621/3 | LS_DigitalInput: bCountIn1_Reset | 0: Not connected | |
| C00621/4 | LS_DigitalInput: bCountIn1_LoadStartValue | 0: Not connected | |
| C02840/1 | CountIn1: Starting value | 0 | incr |
| C02840/2 | CountIn1: Comparison value | 65535 | incr |
| C02841/1 | CountIn1: Counter content | - | incr |
| C00443/1 | DIx: Terminal level | - | |
| **Counting input DI6** | | | |
| C00115/2 | Fct. DI 6/7 10kHz | 0: DI1(6)=In / DI2(7)=In | |
| C00621/97 | LS_DigitalInput: bCountIn6_Reset | 0: Not connected | |
| C00621/98 | LS_DigitalInput: bCountIn6_LoadStartValue | 0: Not connected | |
| C02840/3 | CountIn6: Starting value | 0 | incr |
| C02840/4 | CountIn6: Comparison value | 65535 | incr |
| C02841/2 | CountIn6: Counter content | - | incr |
| C00443/1 | DIx: Terminal level | - | |
| Highlighted in grey = display parameter | | | |

### Internal interfaces to the application

▶ Relevant inputs at the LS_DigitalInput system block:

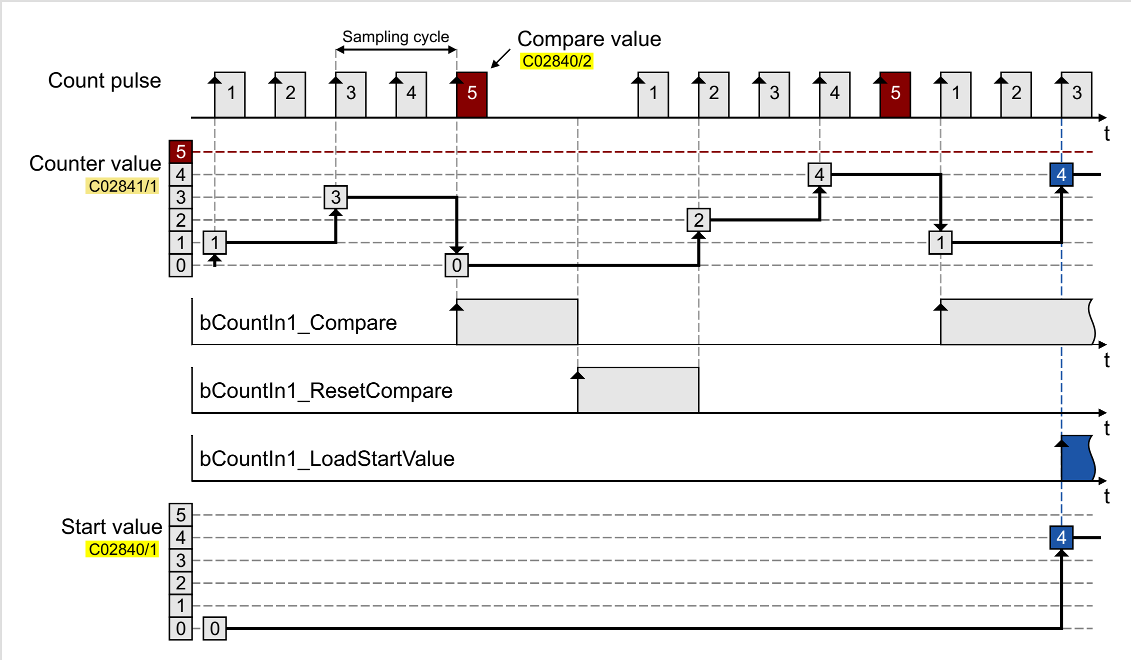
| Identifier<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| **Counting input DI1** | | |
| bCountIn1_ResetCompare<br>BOOL | Reset status signal "Comparison value reached" | |
| | FALSE↗TRUE | The output *bCountIn1_Compare* is reset to FALSE. |
| bCountIn1_LoadStartValue<br>BOOL | Load starting value into counter | |
| | FALSE↗TRUE | The starting value set in C02840/1 is accepted as the current count value. |
| **Counting input DI6** | | |
| bCountIn6_ResetCompare<br>BOOL | Reset status signal "Comparison value reached" | |
| | FALSE↗TRUE | The output *bCountIn6_Compare* is reset to FALSE. |
| bCountIn6_LoadStartValue<br>BOOL | Load starting value into counter | |
| | FALSE↗TRUE | The starting value set in C02840/3 is accepted as the current count value. |

▶ Relevant outputs at the LS_DigitalInput system block:

| Identifier<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| **Counting input DI1** | | |
| bCountIn1_Compare<br>BOOL | Status signal "Comparison value reached" | |
| | FALSE | Current count value < comparison value (C02840/2) |
| | TRUE | Current count value ≥ comparison value (C02840/2) |
| wCountIn1_HW<br>wCountIn1_LW<br>C02841/1 \| WORD | Current count value<br>• Output as High and Low word (without sign)<br>• Possible counting range: 0 ... $2^{31}$ - 1 | |
| **Counting input DI6** | | |
| bCountIn6_Compare<br>BOOL | Status signal "Comparison value reached" | |
| | FALSE | Current count value < comparison value (C02840/4) |
| | TRUE | Current count value ≥ comparison value (C02840/4) |
| wCountIn6_HW<br>wCountIn6_LW<br>C02841/2 \| WORD | Current count value<br>• Output as High and Low word (without sign)<br>• Possible counting range: 0 ... $2^{31}$ - 1 | |

### Counting behaviour

The following temporal characteristic shows the counting process depending on the signals of the interfaces described before:




[7-1] Transient characteristic of a quick counter block, sampling cycle = 1 ms

- The counter starts with the parameterised starting value.
- If the comparison value is reached or exceeded:
  - The counter jumps back to its starting value.
  - The output *bCount1(6)_Compare* is set to TRUE.
- If there is a FALSE-TRUE edge is at the input *bCountIn1(6)_ResetCompare*, the output *bCountIn1(6)_Compare* can be reset to FALSE.
- If there is a FALSE-TRUE edge at the input *bCountIn1(6)_LoadStartValue*, the current counter content can be reset to the parameterised starting value.

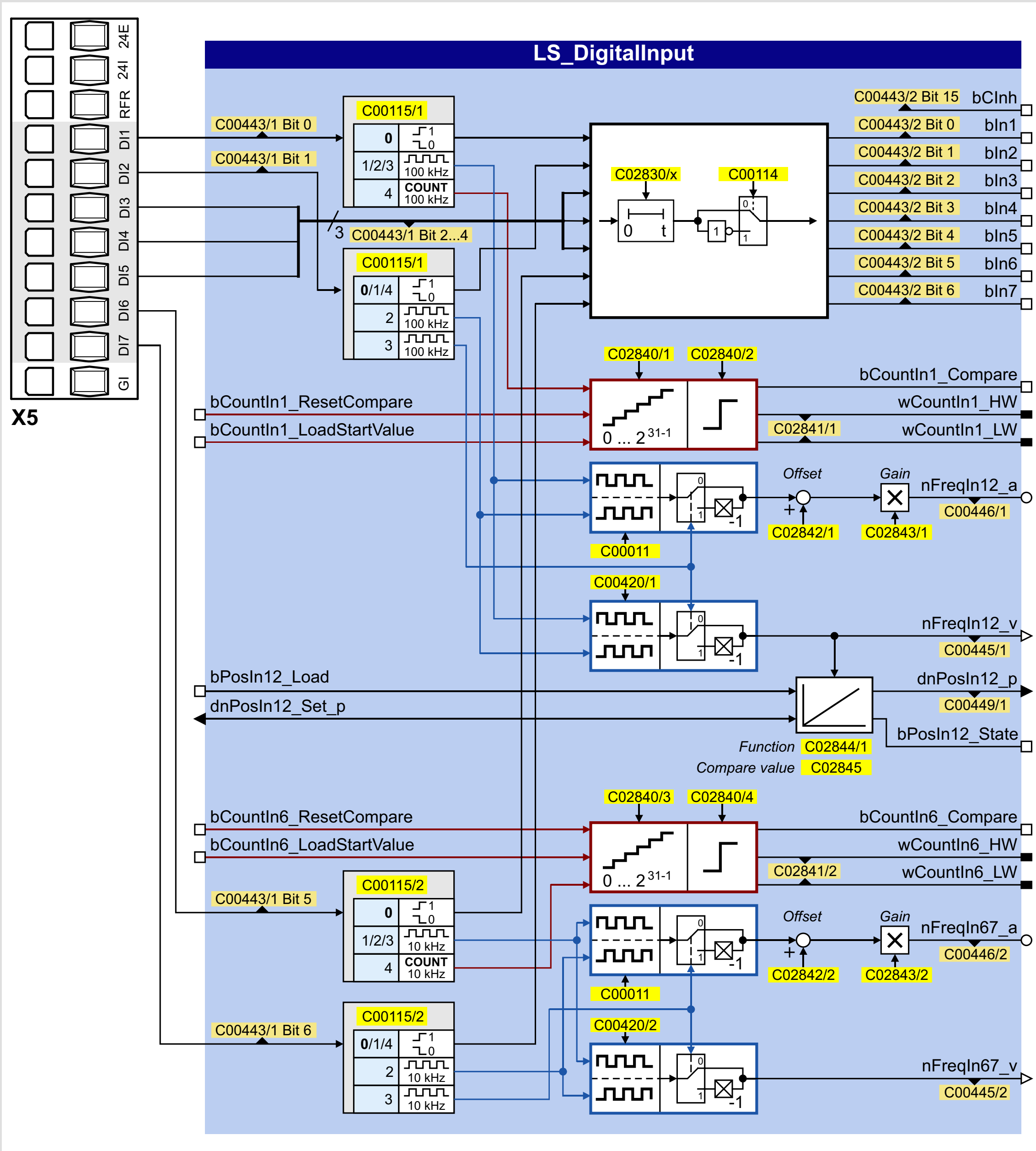
**Related topics:**

- Using DI1(6) and DI2(7) as digital inputs (332)
- Using DI1(6) and DI2(7) as frequency inputs (333)
- Internal interfaces | System block "LS_DigitalInput" (342)

### 7.1.2 Internal interfaces | System block "LS_DigitalInput"

The system block **LS_DigitalInput** maps the digital input terminals in the FB editor.

- The internal processing function of the digital DI1/2 and DI6/7 input terminals can be reconfigured inn C00115 if necessary. These input terminals can then be alternatively used as frequency inputs or counting inputs.
- The DI3 ... DI5 input terminals are basically designed as "normal" digital inputs.



### Inputs

| Identifier<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| **Counting input DI1** | ▸ Using DI1(6) as counting input | |
| bCountIn1_ResetCompare<br>BOOL | Reset status signal "Comparison value reached" | |
| | FALSE↗TRUE | The output *bCountIn1_Compare* is reset to FALSE. |
| bCountIn1_LoadStartValue<br>BOOL | Load starting value into counter | |
| | FALSE↗TRUE | The starting value set in C02840/1 is accepted as the current count value. |
| **Frequency input DI1/DI2** | ▸ Output of the encoder position of the DI1/DI2 frequency input | |
| bPosIn12_Load<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnPosIn12_Set_p* and *bPosIn12_State* is reset to FALSE. |
| dnPosIn12_Set_p<br>DINT | Starting value for angle integrator | |
| **Counting input DI6** | ▸ Using DI1(6) as counting input | |
| bCountIn6_ResetCompare<br>BOOL | Reset status signal "Comparison value reached" | |
| | FALSE↗TRUE | The output *bCountIn6_Compare* is reset to FALSE. |
| bCountIn6_LoadStartValue<br>BOOL | Load starting value into counter | |
| | FALSE↗TRUE | The starting value set in C02840/3 is accepted as the current count value. |

### Outputs

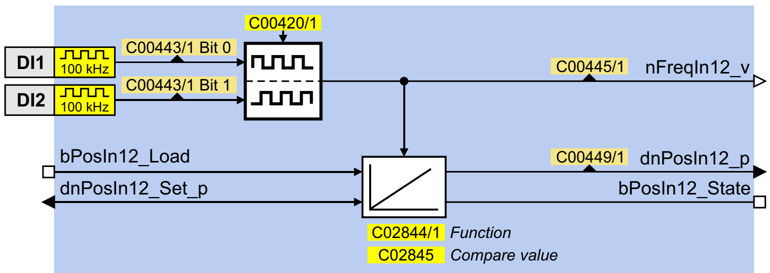
| Identifier<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| bCInh<br>C00443/2 \| BOOL | RFR digital input (controller enable) | |
| **Digital inputs DI1 ... DI7** | ▸ Using DI1(6) and DI2(7) as digital inputs | |
| bIn1 ... bIn7<br>C00443/2 \| BOOL | Digital input DI1 ... DI7 | |
| **Counting input DI1** | ▸ Using DI1(6) as counting input | |
| bCountIn1_Compare<br>BOOL | Status signal "Comparison value reached" | |
| | FALSE | Current count value < comparison value (C02840/2) |
| | TRUE | Current count value ≥ comparison value (C02840/2) |
| wCountIn1_HW<br>wCountIn1_LW<br>C02841/1 \| WORD | Current count value<br>• Output as High and Low word (without sign)<br>• Possible counting range: 0 ... $2^{31}$ - 1 | |
| **Frequency input DI1/DI2** | ▸ Using DI1(6) and DI2(7) as frequency inputs | |
| nFreqIn12_a<br>C00446/1 \| INT | Output frequency as scaled analog signal in [%] | |
| nFreqIn12_v<br>C00445/1 \| INT | Output frequency as speed signal in [inc/ms] | |
| dnPosIn12_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bPosIn12_State*) | |
| bPosIn12_State<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bPosIn12_Load*. | |
| | TRUE | Overflow has occurred or distance is processed. |

| Identifier<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| **Counting input DI6** | Using DI1(6) as counting input | |
| bCountIn1_Compare<br>BOOL | Status signal "Comparison value reached" | |
| | FALSE | Current count value < comparison value (C02840/4) |
| | TRUE | Current count value ≥ comparison value (C02840/4) |
| wCountIn6_HW<br>wCountIn6_LW<br>C02841/2 \| WORD | Current count value<br>• Output as High and Low word (without sign)<br>• Possible counting range: 0 ... $2^{31}$ - 1 | |
| **Frequency input DI6/DI7** | Using DI1(6) and DI2(7) as frequency inputs | |
| nFreqIn67_a<br>C00446/2 \| INT | Output frequency as scaled analog signal in [%] | |
| nFreqIn67_v<br>C00445/2 \| INT | Output frequency as speed signal in [inc/ms] | |

### 7.1.2.1 Output of the encoder position of the DI1/DI2 frequency input

The LS_DigitalInput system block includes an integrator for providing the encoder position.

- The integrator can take max. ±32000 encoder revolutions.
- The starting position can be loaded via inputs.
- The internal function can be set via parameters.
- In addition to the encoder position, the "Overflow occurred/distance processed" status signal is provided.




#### Inputs

| Identifier<br>DIS code \| data type | Information/possible settings | |
|---|---|---|
| bPosIn12_Load<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnPosIn12_Set_p* and *bPosIn12_State* is reset to FALSE. |
| dnPosIn12_Set_p<br>DINT | Starting value for angle integrator | |

#### Outputs

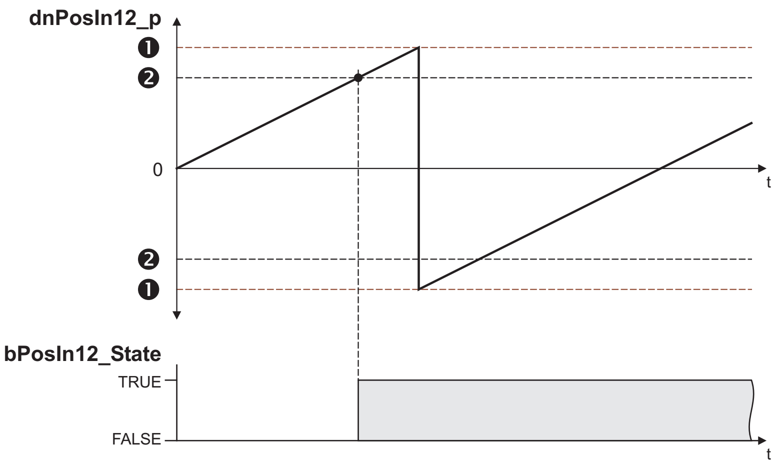
| Identifier<br>DIS code \| data type | Value/meaning | |
|---|---|---|
| dnPosIn12_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bPosIn12_State*) | |
| bPosIn12_State<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bPosIn12_Load*. | |
| | TRUE | Overflow has occurred or distance is processed. |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C02844/1 | | | | Function |
| | 0 | Loading with level | | Load integrator with TRUE level at the *bPosIn12_Load* input (Lenze setting). |
| | 1 | Loading with edge | | Load integrator with FALSE/TRUE edge at the *bPosIn12_Load* input. |
| | 2 | Loading with level + reset | | Load integrator when reaching the comparison value or with TRUE level at the *bPosIn12_Load* input. |
| C02845 | 0 | | 2000000000 | Comparison value<br>• Is valid for both the positive and the negative value range.<br>• Lenze setting: 0 |

### Function at constant input value

Selection: C02844/1 = "0: Loading with level" or "1: Loading with edge"




❶ Overflow at ±2147483648 incr. (±32767 encoder revolutions)
❷ C02845: Comparison value (applies to both positive and negative value range)

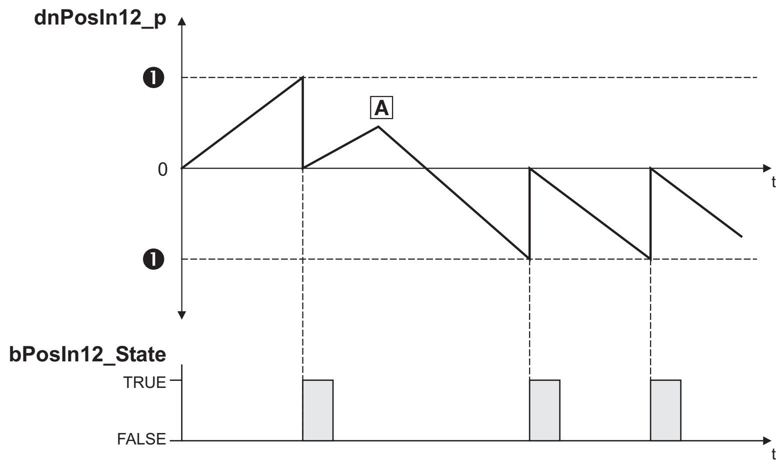
[7-2] Switching performance if the overflow is in the positive direction

- If "0: Loading with level" is selected in C02844/1, the *bPosIn12_Load* input is status-controlled: In case of a TRUE signal, the integrator is loaded with the value at *dnPosIn12_Set_p* and the *bPosIn12_State* output is set to FALSE.
- If "1: Loading with edge" is selected in C02844/1, the *bPosIn12_Load* input is edge-controlled: In case of a FALSE/TRUE edge, the integrator is loaded with the value at *dnPosIn12_Set_p* and then immediately continues to integrate, the *bPosIn12_State* output is set to FALSE.
- A positive *nFreqIn12_v* signal is incremented (the counter content is increased with every cycle).
- A negative *nFreqIn12_v* signal is decremented (the counter content is reduced with every cycle).

- *dnPosIn12_p* provides the counter content of the bipolar integrator.
  - If the counter content exceeds a value of +32767 encoder revolutions (corresponds to +2147483647 incr.), an overflow occurs and the counting process continues at a value of -32768 encoder revolutions.
  - If the counter content falls below a value of -32768 encoder revolutions (corresponds to -2147483648 incr.), an overflow occurs and the counting process starts at a value of +32767 encoder revolutions.
- *bPosIn12_State* is set to TRUE if the comparison value set in C02845 has been reached.

### Function at input value with sign reversal

Selection: C02844/1 = "2: Loading with level + reset"




A Sign reversal of the value at *nFreqIn12_v*
❶ C02845: Comparison value (applies to both positive and negative value range)

[7-3] Switching performance if the input signal changes signs

- If "2: Loading with level + reset" is selected in C02844/1, the *bPosIn12_Load* input is status-controlled: In case of a TRUE signal, the integrator is loaded with the value at *dnPosIn12_Set_p* and the *bPosIn12_State* output is set to FALSE.
- A positive *nFreqIn12_v* signal is incremented (the counter content is increased with every cycle).
- A negative *nFreqIn12_v* signal is decremented (the counter content is reduced with every cycle).
- *dnPosIn12_p* provides the counter content of the bipolar integrator.
  - If the positive counter content is higher than the comparison value set in C02845, the comparison value will be subtracted from the counter content, and *bPosIn12_State* will be set to TRUE for one task cycle.
  - If the negative counter content is lower than the comparison value set in C02845, the comparison value will be added to the counter content, and *bPosIn12_State* will be set to TRUE for one task cycle.

### Calculation of the output signal

The output value at *dnPosIn12_p* is calculated as per the formula below:

$$\text{dnPosIn12\_p [incr.]} = \text{nFreqIn12\_v [rpm]} \cdot \text{t [s]} \cdot \text{65535 [incr./rev.]}$$

t = integration time
16384 ≡ 15000 rpm
1 ≡ 1 incr.

### Example

You want to determine the counter content of the integrator at a certain speed at the input and a certain integration time **t**.

Given values:

- *nFreqIn12_v* = 1000 rpm ≈ integer value 1092
- Integration time t = 10 s
- Starting value of the integrator = 0

Solution:

- Conversion of the *nFreqIn12_v* input signal:

$$1000\ \text{rpm} = \frac{1000\ \text{rev.}}{60\ \text{s}}$$

- Calculation of the output value:

$$\text{dnPosIn12\_p} = \frac{1000\ \text{rev.}}{60\ \text{s}} \cdot 10\ \text{s} \cdot \frac{65535\ \text{incr.}}{\text{Rev.}} = 10922666\ \text{incr.}$$

## 7.2 Digital output terminals

The drive controller has

- three parameterisable output terminals (DO1 ... DO3) for outputting digital signals
- a relay output (terminal strip X101),
- a (high-current) output for controlling a brake (terminal strip X107).

**Note!**

Initialisation behaviour:

- After mains switching up to the start of the application, the digital outputs remain set to FALSE.

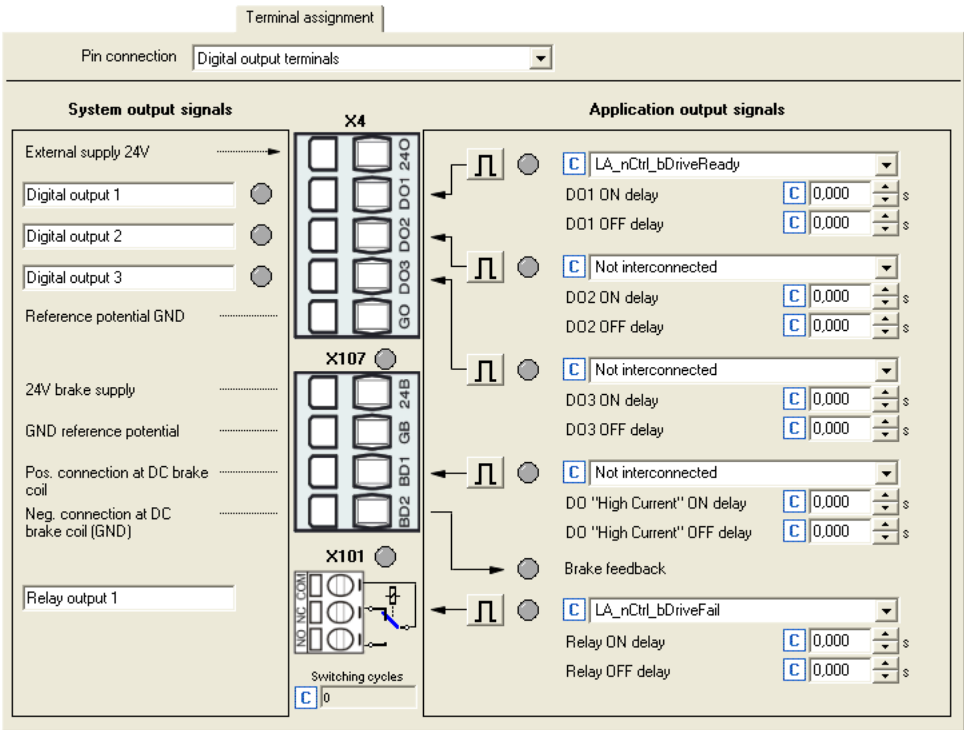
Exception handling:

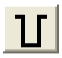
- In case of a critical exception in the application (e.g. reset), the digital outputs are set to FALSE considering the terminal polarity parameterised in C00118.

Switching cycle diagnostics of the relay:

- A reference for evaluating the wear limit can be obtained via the number of switching cycles of the relay displayed in C00177/2.

**Parameterisation dialog in the »Engineer«:**



| Button | Function |
|---|---|
|  | Indicates that the polarity of the output is HIGH active.<br>The polarity can be changed from HIGH active to LOW active by clicking this button. |
|  | Indicates that the polarity of the output is LOW active.<br>The polarity can be changed from LOW active to HIGH active by clicking this button. |

### Short overview of parameters for the digital output terminals:

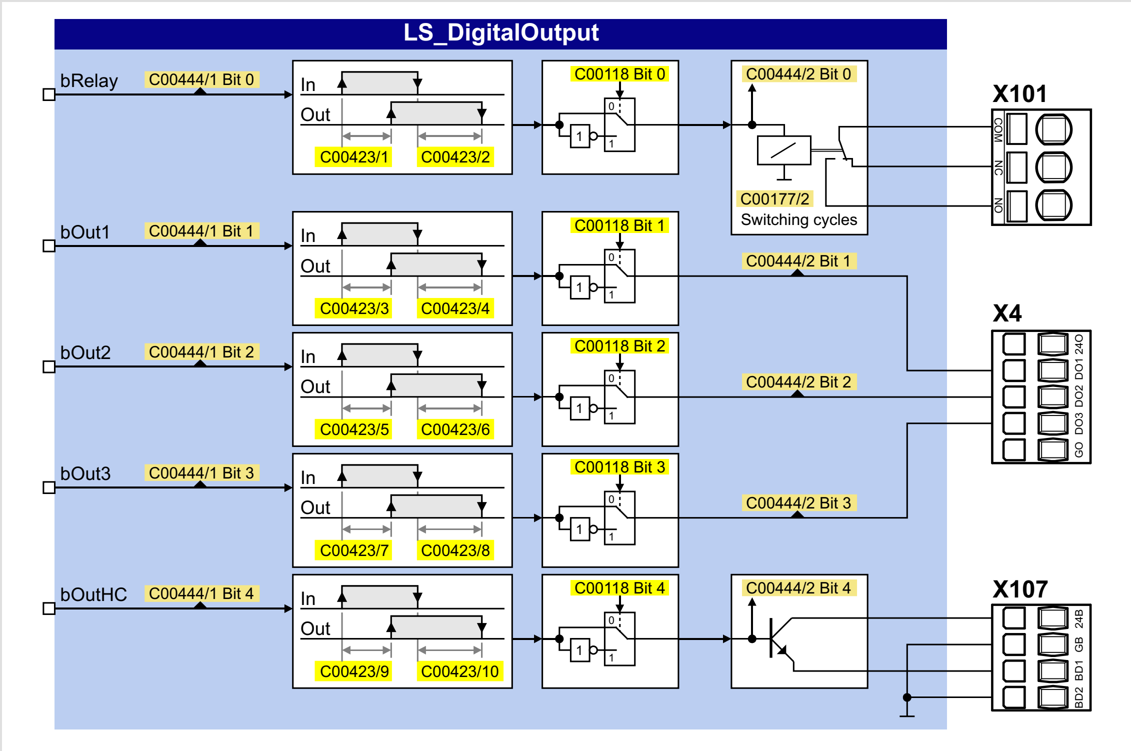
| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **Digital outputs DO1/DO2** | | | |
| C00118 | DigOutX: Inversion | Bit coded | |
| C00423/3 | DO1 ON delay | 0.000 | s |
| C00423/4 | DO1 OFF delay | 0.000 | s |
| C00423/5 | DO2 ON delay | 0.000 | s |
| C00423/6 | DO2 OFF delay | 0.000 | s |
| C00423/7 | DO3 ON delay | 0.000 | s |
| C00423/8 | DO3 OFF delay | 0.000 | s |
| C00444/1 | DOx: Input level | - | |
| C00444/2 | DOx: Terminal level | - | |
| **High current output** | | | |
| C00423/9 | DO "High Current" ON delay | 0.000 | s |
| C00423/10 | DO "High Current" OFF delay | 0.000 | s |
| C00117 | Status of brake output BD | - | |
| **Relay output** | | | |
| C00423/1 | Relay ON delay | 0.000 | s |
| C00423/2 | Relay OFF delay | 0.000 | s |
| **Digital outputs - terminal configuration** | | | |
| C00621/1 | LS_DigitalOutput:bRelay | 1001: LA_nCtrl_bDriveFail | |
| C00621/2 | LS_DigitalOutput:bOut1 | 1000: LA_nCtrl_bDriveReady | |
| C00621/99 | LS_DigitalOutput: bOut2 | 0: Not connected | |
| C00621/100 | LS_DigitalOutput: bOut3 | 0: Not connected | |
| C00621/101 | LS_DigitalOutput: bOut HighCurrent | 0: Not connected | |
| Highlighted in grey = display parameter | | | |

### Related topics:

- Configuring exception handling of the output terminals (📖 369)
- User-defined terminal assignment (📖 370)

## 7.2.1 Internal interfaces | System block "LS_DigitalOutput"

The **LS_DigitalOutput** system block maps the digital output terminals in the FB editor.




| Input<br>DIS code \| data type | Information/possible settings |
|---|---|
| bRelay<br>C00444/1 \| BOOL | Relay output, potential-free two-way switch |
| bOut1 ... bOut3<br>C00444/1 \| BOOL | Digital output DO1 ... DO3 |
| bOutHC<br>C00444/1 \| BOOL | Output for brake control |

## 7.3 Analog terminals

The analog input terminals together with the analog output terminals are located on the X3 plug connector.

### Analog input terminals

The drive controller has four analog input terminals for detecting two current signals and two voltage signals:

- Voltage signals in the ± 10 V range
  The voltage signal can be e.g. an analog speed setpoint or the signal of an external sensor (temperature, pressure, etc.).
- Current signals in the 0/+ 4 ... + 20 mA range
  For open-circuit monitoring, the current signal can be evaluated with regard to "Life Zero" or "Dead Zero":
  - 0 ... 20 mA, without open-circuit monitoring
  - 4 ... 20 mA, with open-circuit monitoring

**Note!**

To avoid undefined states, free input terminals of the controller must be assigned as well, e.g. by applying 0 V to the terminal.

### Analog output terminals

The drive controller has four analog output terminals.

- Two output terminals output an analog current signal (O1I, O2I)
- Two output terminals output an analog voltage signal (O1U, O2U)

**Note!**

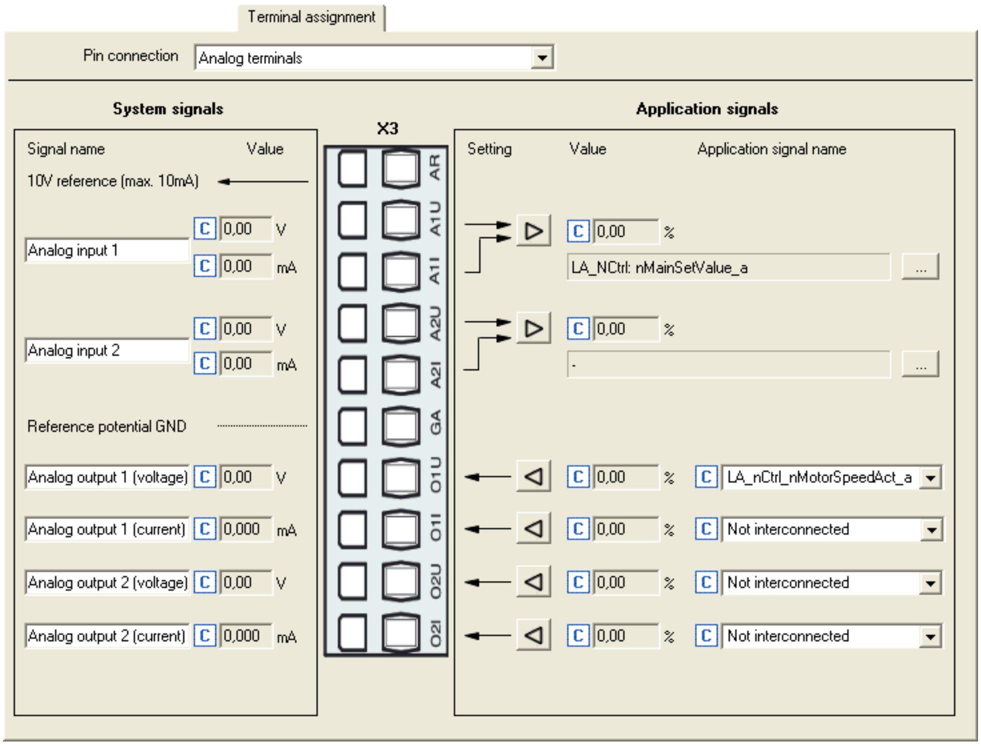
Initialisation behaviour:

- After mains switching up to the start of the application, the analog outputs remain set to 0 V.

Exception handling:

- In case of a critical exception in the application (e.g. reset), the analog outputs are set to 0 V.

### Parameterisation dialog in the »Engineer«:




| Button | Function |
|---|---|
| ▷ | Parameterising analog input (🕮 355) |
| ◁ | Parameterising analog output (🕮 359) |
| ... | Open the parameterising dialog for assigning application inputs to the analog input.<br>▸ Changing the terminal assignment with the »Engineer« (🕮 374) |

### Short overview of parameters for the analog terminals:

| Parameter | Info | Lenze setting<br>Value | Unit |
|---|---|---|---|
| **Analog input 1** | | | |
| C00028/1 | AIN1: Input voltage | - | V |
| C00029/1 | AIN1: Input current | - | mA |
| C00033/1 | AIN1: Output value (to application) | - | % |
| **Analog input 2** | | | |
| C00028/2 | AIN2: Input voltage | - | V |
| C00029/2 | AIN2: Input current | - | mA |
| C00033/2 | AIN2: Output value (to application) | - | % |
| Highlighted in grey = display parameter | | | |

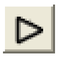
| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| **Analog output 1** | | | |
| C00439/1 | O1U: Input value (from application) | - | % |
| C00439/3 | O1I: Input value (from application) | - | % |
| C00436/1 | O1U: Voltage | - | V |
| C00437/1 | O1I: Current | - | mA |
| **Analog output 2** | | | |
| C00439/2 | O2U: Input value (from application) | - | % |
| C00439/4 | O2I: Input value (from application) | - | % |
| C00436/2 | O2U: Voltage | - | V |
| C00437/2 | O2I: Current | - | mA |
| **Analog outputs - terminal assignment** | | | |
| C00620/1 | LS_AnalogOutput: nOut1_a (V) | 1003: LA_nCtrl_nMotorSpeedAct_a | |
| C00620/39 | LS_AnalogOutput: nOut1_a (I) | 0: Not connected | |
| C00620/38 | LS_AnalogOutput: nOut2_a (V) | 0: Not connected | |
| C00620/40 | LS_AnalogOutput: nOut2_a (I) | 0: Not connected | |

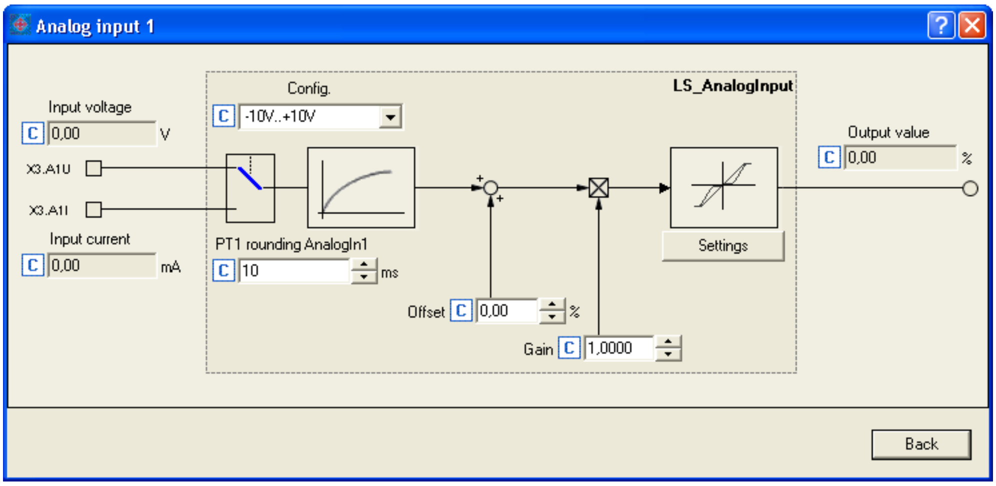
Highlighted in grey = display parameter

### Related topics:

- Configuring exception handling of the output terminals (369)
- User-defined terminal assignment (370)

## 7.3.1 Parameterising analog input

By clicking on the  button on the **Terminal assignment** tab, you reach the parameterising dialog for the corresponding analog input:




### Short overview of parameters for the analog inputs:

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| **Analog input 1** | | | |
| C00034/1 | AIN1: Config. | 0: -10V..+10V | |
| C00026/1 | AIN1: Offset | 0.00 | % |
| C00027/1 | AIN1: Gain | 1.0000 | |
| C00028/1 | AIN1: Input voltage | - | V |
| C00029/1 | AIN1: Input current | - | mA |
| C00033/1 | AIN1: Output value (to application) | - | % |
| C00440/1 | PT1 rounding AnalogIn1 | 10 | ms |
| C00598/1 | Resp. to open circuit AIN1 | 3: TroubleQuickStop | |
| **Analog input 2** | | | |
| C00034/2 | AIN2: Config. | 0: -10V..+10V | |
| C00026/2 | AIN2: Offset | 0.00 | % |
| C00027/2 | AIN2: Gain | 1.0000 | |
| C00028/2 | AIN2: Input voltage | - | V |
| C00029/2 | AIN2: Input current | - | mA |
| C00033/2 | AIN2: Output value (to application) | - | % |
| C00440/2 | PT1 rounding AnalogIn2 | 10 | ms |
| C00598/2 | Resp. to open circuit AIN2 | 3: TroubleQuickStop | |
| Highlighted in grey = display parameter | | | |

### Using current input A1I/A2I

In the Lenze setting, voltage signals in the range of ± 10 V are evaluated via the A1U and A2U input terminals. If current signals are detected via the A1I or A2I input terminals instead, the selection "1: 0...20mA" or "2: 4...20mA" is to be set in C00034.

**Tip!**

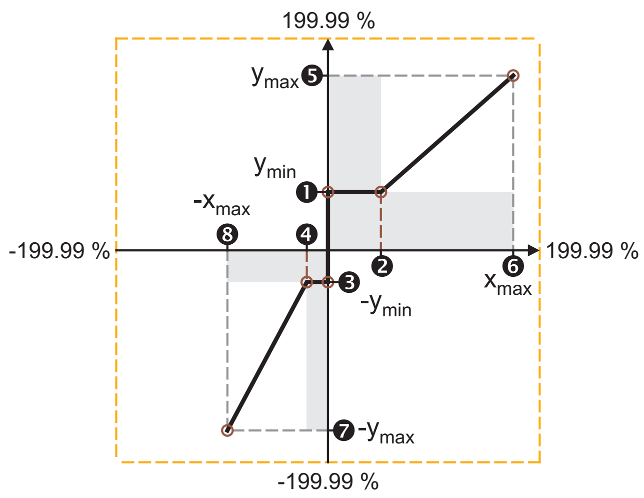
By selecting "2: 4...20mA", you can implement a 4 ...20 mA current loop, e.g. for stipulation of the speed setpoint.

### Open-circuit monitoring

In the case of configuration as a 4 ... 20 mA current loop, the fault response set in C00598 takes place in the event of a wire breakage (Lenze setting: "TroubleQuickStop").

### 7.3.1.1 Signal adaptation by means of characteristic

According to the illustration below, an individual characteristic can be parameterised for the analog inputs via the subcodes of C00010 and C00020 to provide different slopes and a dead band. Here, the input signal corresponds to the X axis and the output signal corresponds to the Y axis:



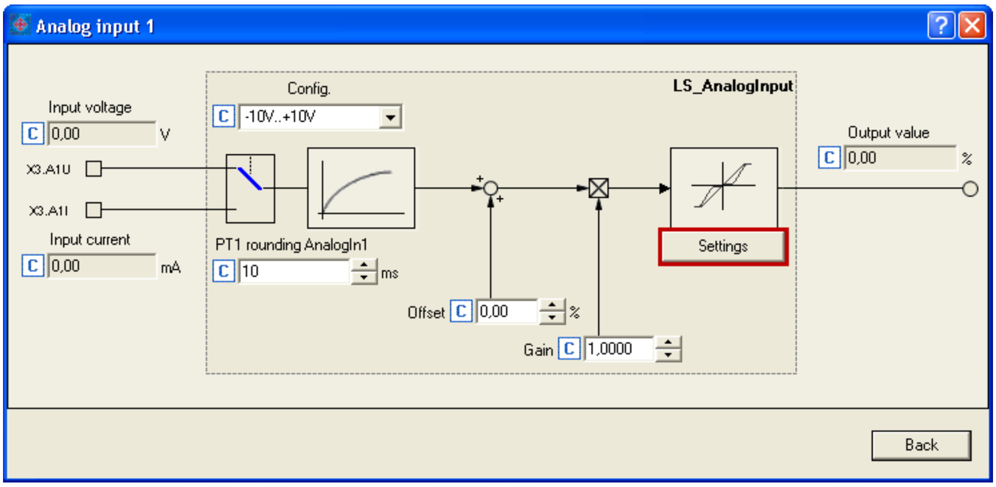

[7-4] Characteristic for analog inputs

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00010/1 | ❶ AIN1: (+y0) = min | 0.00 | % |
| C00010/2 | ❷ AIN1: (+x0) = Dead band | 1.00 | % |
| C00010/3 | ❸ AIN1: (-y0) = (-min) | 0.00 | % |
| C00010/4 | ❹ AIN1: (-x0) = (-Dead band) | 1.00 | % |
| C00010/5 | ❺ AIN1: (+ymax) | 199.99 | % |
| C00010/6 | ❻ AIN1: (+xmax) | 199.99 | % |
| C00010/7 | ❼ AIN1: (-ymax) | 199.99 | % |
| C00010/8 | ❽ AIN1: (-xmax) | 199.99 | % |
| C00020/1 | ❶ AIN2: (+y0) = min | 0.00 | % |
| C00020/2 | ❷ AIN2: (+x0) = Dead band | 1.00 | % |
| C00020/3 | ❸ AIN2: (-y0) = (-min) | 0.00 | % |
| C00020/4 | ❹ AIN2: (-x0) = (-Dead band) | 1.00 | % |
| C00020/5 | ❺ AIN2: (+ymax) | 199.99 | % |
| C00020/6 | ❻ AIN2: (+xmax) | 199.99 | % |
| C00020/7 | ❼ AIN2: (-ymax) | 199.99 | % |
| C00020/8 | ❽ AIN2: (-xmax) | 199.99 | % |

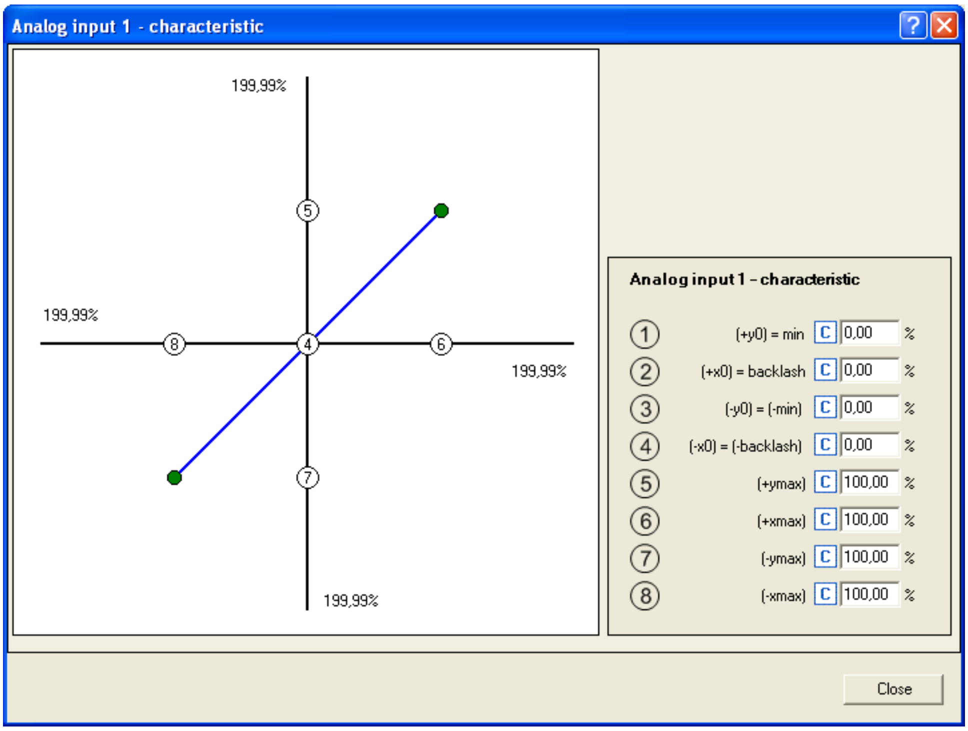
In the »Engineer«, there is a parameterising dialog for entering the characteristic. This dialog also displays the set characteristic graphically.

**Proceed as follows to open the dialog for parameterising the characteristic:**

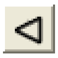
1. Go to the **Terminal assignment** tab and select the "Analog terminals" entry in the **Control connections** list field.
2. Click on the ▷ button for the analog input in order to open the *Analog input* dialog.

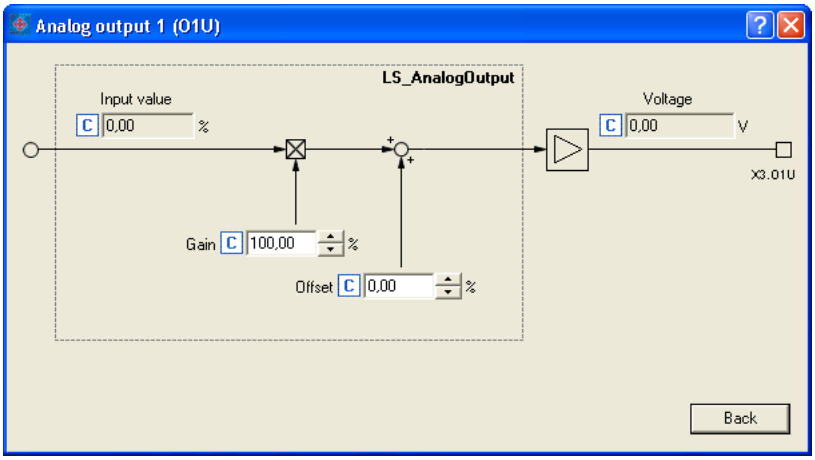



3. Click on the **Settings** button in order to open the *Analog input - Characteristic* dialog box:



### 7.3.2 Parameterising analog output

By clicking on the  button on the **Terminal assignment** tab, you can open the parameterising dialog for the corresponding analog output (here: O1U):

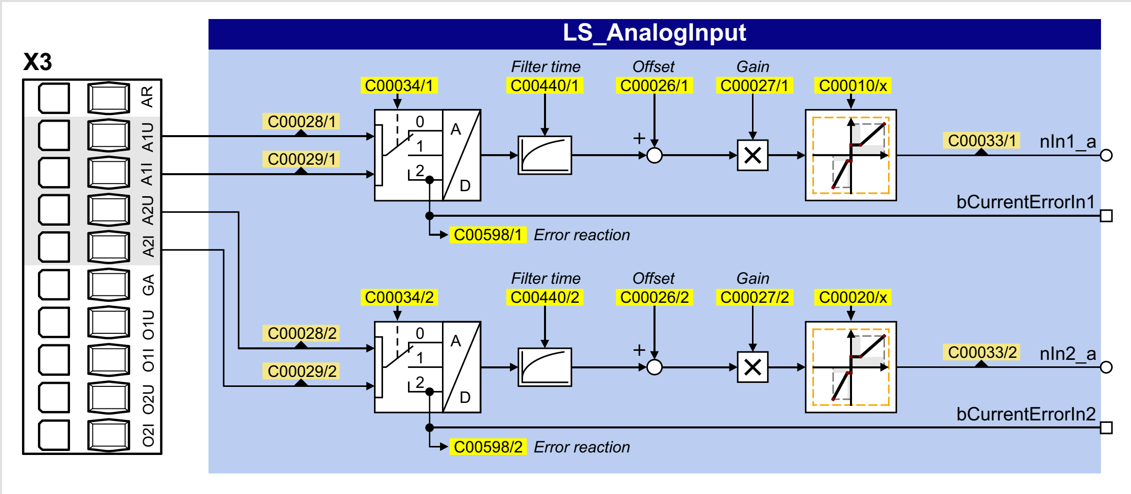



**Short overview of parameters for the analog outputs:**

| Parameter | Info | Lenze setting<br>Value | <br>Unit |
|---|---|---|---|
| **Analog output 1** | | | |
| C00434/1 | O1U: Gain | 100.00 | % |
| C00435/1 | O1U: Offset | 0.00 | % |
| C00439/1 | O1U: Input value (from application) | - | % |
| C00439/3 | O1I: Input value (from application) | - | % |
| C00436/1 | O1U: Voltage | - | V |
| C00437/1 | O1I: Current | - | mA |
| **Analog output 2** | | | |
| C00434/2 | O2U: Gain | 100.00 | % |
| C00435/2 | O2U: Offset | 0.00 | % |
| C00439/2 | O2U: Input value (from application) | - | % |
| C00439/4 | O2I: Input value (from application) | - | % |
| C00436/2 | O2U: Voltage | - | V |
| C00437/2 | O2I: Current | - | mA |
| Highlighted in grey = display parameter | | | |

### 7.3.3 Internal interfaces | System block "LS_AnalogInput"

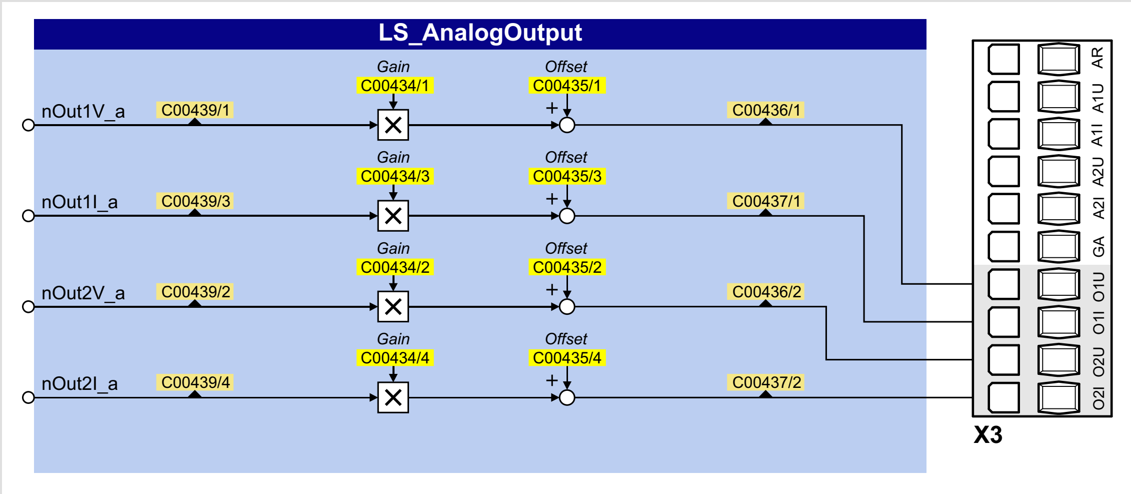
The **LS_AnalogInput** system block maps the analog inputs in the FB editor.




| **Output**<br>DIS code \| data type | **Value/meaning** | |
|---|---|---|
| nIn1_a<br>C00033/1 \| INT | Analog input 1<br>• Scaling:<br>$\pm 2^{14} \equiv \pm 10$ V for use as voltage input<br>$+2^{14} \equiv +20$ mA for use as current input | |
| bCurrentErrorIn1<br>BOOL | Status signal "Current input error"<br>• Only when analog input 1 is used as current input.<br>• Application: Cable-breakage monitoring of the 4 ...20 mA circuit. | |
| | TRUE | $\lvert I_{AIN1} \rvert < 4$ mA |
| nIn2_a<br>C00033/2 \| INT | Analog input 2<br>• Scaling:<br>$\pm 2^{14} \equiv \pm 10$ V for use as voltage input<br>$+2^{14} \equiv +20$ mA for use as current input | |
| bCurrentErrorIn2<br>BOOL | Status signal "Current input error"<br>• Only when analog input 2 is used as current input.<br>• Application: Cable-breakage monitoring of the 4 ...20 mA circuit. | |
| | TRUE | $\lvert I_{AIN2} \rvert < 4$ mA |

### 7.3.4 Internal interfaces | System block "LS_AnalogOutput"

The **LS_AnalogInput** system block maps the analog outputs in the FB editor.




| Input<br>DIS code \| data type | Information/possible settings |
|---|---|
| nOut1V_a<br>C00439/1 \| INT | Analog output 1 (voltage)<br>• Scaling: $2^{14} \equiv 16384 \equiv 10$ V |
| nOut1I_a<br>C00439/3 \| INT | Analog output 1 (current)<br>• Scaling: $2^{14} \equiv 16384 \equiv 20$ mA |
| nOut2V_a<br>C00439/2 \| INT | Analog output 2 (voltage)<br>• Scaling: $2^{14} \equiv 16384 \equiv 10$ V |
| nOut2I_a<br>C00439/4 \| INT | Analog output 2 (current)<br>• Scaling: $2^{14} \equiv 16384 \equiv 20$ mA |

## 7.4 Touch probe detection

"Touch probe" (TP) in general means quick detection of a position by a quick sensor if the pulse duration of the sensor signal is too short to be detected by a "normal" digital input (scanning time: 1 ms).

Moreover, touch probe requires exact detection of the position. Here, the difference in position between the 1 ms position detection and the touch probe signal is considered as correction value.

**Applications:**

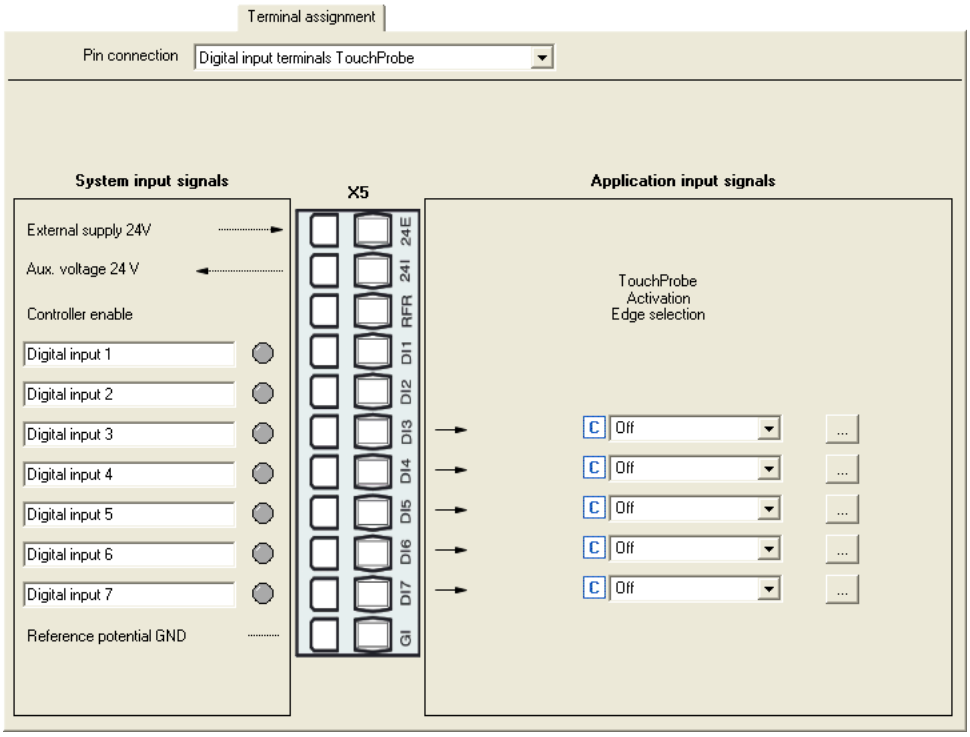
- Precise approach of a position at a previously detected signal mark
- High-precision traversing of a section starting at a previously detected signal mark
- Safe detection of signal marks with very short signal edges
- Measurement of a high-precision position at a previously detected signal mark
- Homing to touch probe (homing signal)
- Relative residual path positioning (traversing of a section starting at signal mark)
- Absolute positioning to target position, activated by a safely detected signal mark
- Position measurement of a distance between 2 signal marks

### Overview of touch probe signal sources

The 8400 TopLine provides the following signal sources for touch probe detection which can be configured independently:

| Signal source | Edge sensitivity | Applications |
|---|---|---|
| Digital input DI3<br>...<br>Digital input DI7 | Rising edge, falling edge, rising and falling edge (parameterisable) | Homing<br>Residual path positioning<br>Position measurement<br>Freely interconnectable |
| Zero pulse - resolver* | Rising edge | Homing |
| Zero pulse - Multi-Encoder* | Rising edge | Homing |
| * in preparation | | |

### Parameterisation dialog in the »Engineer«:



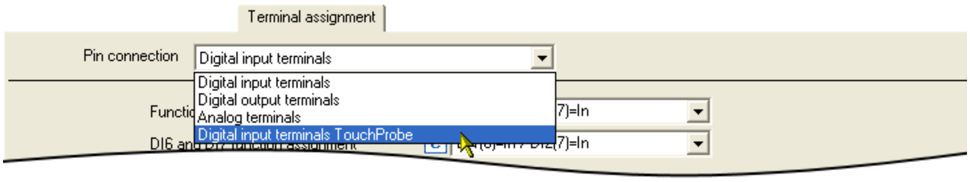

| Button | Function |
|---|---|
| ... | Open the parameterisation dialog for the selected TP source.<br>▸ Parameter setting (📖 364) |

| Parameter | Info | Lenze setting Value \| Unit |
|---|---|---|
| C02810/3...7 | TPDigIn3 ... TPDigIn7: Edge selection | 0: Off |

### 7.4.1 Parameter setting

Proceed as follows to open the parameterisation dialog for setting a TP signal source:

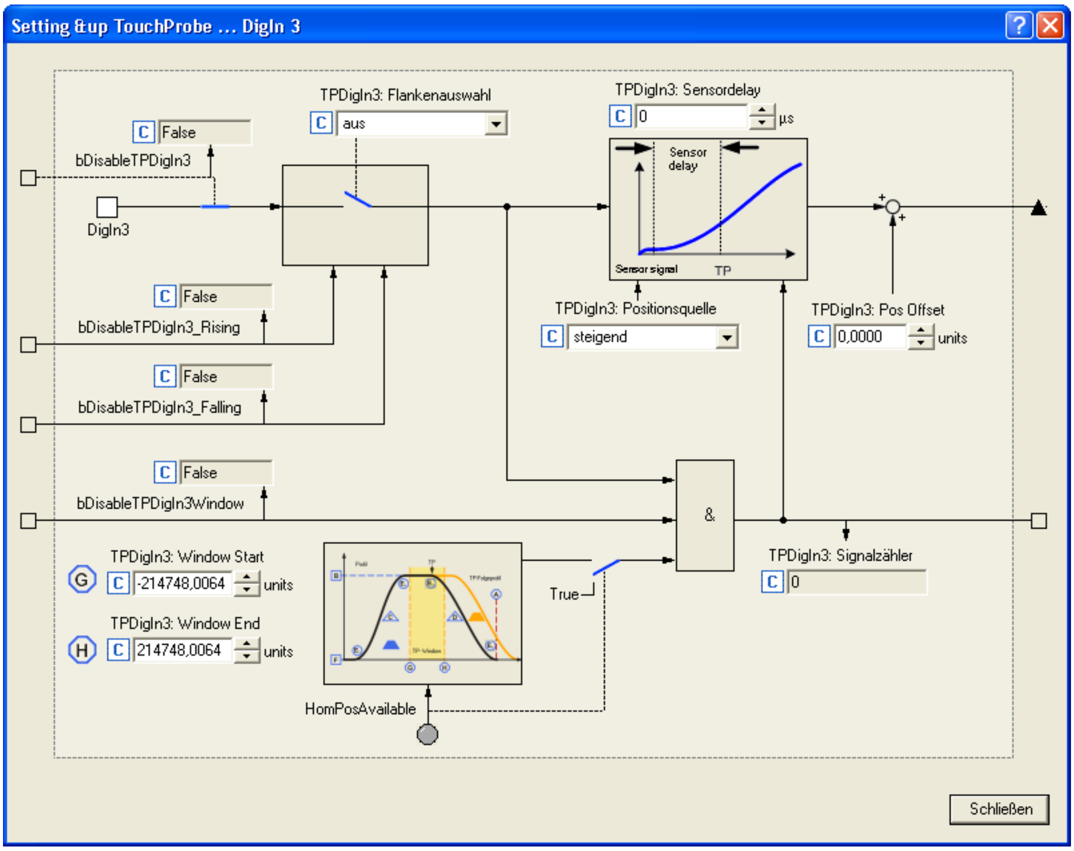
1. Go to the **Terminal Assignment** tab and select "Digital input terminals TouchProbe" in the **Control connections** list field:




2. Click the ... button for the digital input that is to be set for the touch probe detection.

**Tip!**

If the "Table positioning" technology application is used, the following parameterisation dialog can be opened for the selected TP source in the parameterisation dialogs for the Profile entry and the basic "Homing" function via the **Set up touch probe...** button:



**Short overview of the relevant parameters:**

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02810/3...7 | TPDigIn3 ... TPDigIn7: Edge selection | 0: Off | |
| C02811/3...7 | TPDigIn3 ... TPDigIn7: Sensor delay | 0 | µs |
| C02812/3...7 | TPDigIn3 ... TPDigIn7: Pos Offset | 0.0000 | units |
| C02813/1...3 | TPDigIn3 ... TPDigIn5: Window start | 0.0000 | units |
| C02814/1...3 | TPDigIn3 ... TPDigIn5: Window end | 0.0000 | units |
| C02815/3...7 | TPDigIn3 ... TPDigIn7: Position source | 0: Position encoder actual value | |
| C02816/3...7 | TPDigIn3 ... TPDigIn7: Signal counter | - | |
| Highlighted in grey = display parameter | | | |

### Edge selection

Select which edge the corresponding input is to respond to.

- In the case of signal sources DI3 ... DI5, the edge sensitivity can also be dynamically changed via inputs at the LS_TouchProbe SB.

### Sensor delay

This setting serves to compensate for a sensor delay of the touch probe sensor, if any.

- Typical values for laser photoelectric barriers are e.g. 300 µs.
- Delay of the digital inputs for the 8400 device series:
  - 5 µs for a rising edge
  - 25 µs for a falling edge
- Internal automatic compensation for device-internal signal delay due to encoder zero pulse.

### Pos Offset

Use this setting to add an offset to the position value measured by touch probe. This may be required if the touch probe sensor has assumed a disadvantageous position on the machine. By adding an offset the touch probe sensor can be moved to a position which is more convenient to the application.

#### Window start / end

Via the two parameters Window start (C02813/x) and Window end (C02814/x), acceptance windows can be set for the DI3 ... DI5 signal sources in which the touch probe signal is accepted.

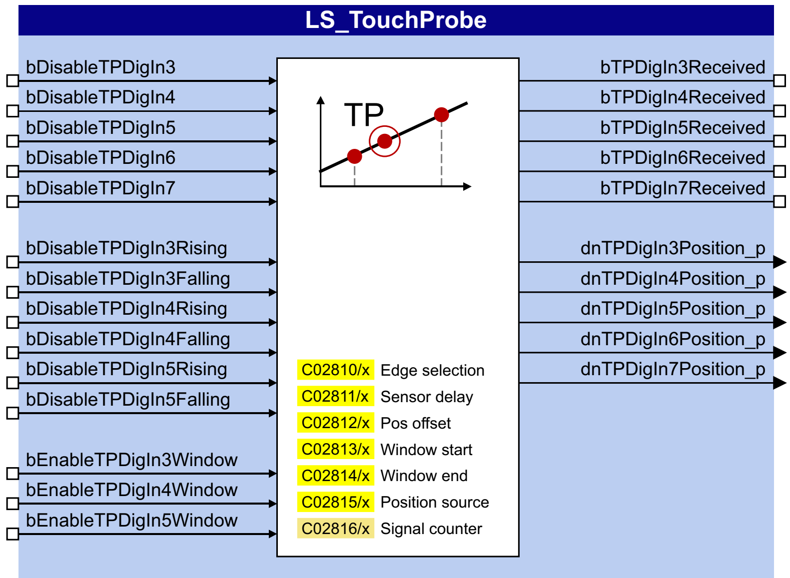
- The acceptance windows can be dynamically activated via the inputs at the LS_TouchProbe SB.
- If the actual position is outside the acceptance window, touch probe is automatically deactivated.
- If both window limits are set to "0", the acceptance window does not have any effect.
- To properly use this function, the drive needs to know the home position (zero position).

#### Position source

Selection of the position signal source to be measured with touch probe. This usually is the actual position of the motor / encoder position encoder.

### 7.4.2 Internal interfaces | System block "LS_TouchProbe"

The **LS_TouchProbe** system block provides the internal interfaces for touch probe detection in the function block editor:




#### Inputs

| Input<br>Data type | Information/possible settings | |
|---|---|---|
| bDisableTPDigIn3...7<br>BOOL | DI3 ... DI7: Dynamically deactivate TP function | |
| | TRUE | TP function is deactivated. |
| bDisableTPDigIn3...5Rising<br>BOOL | DI3 ... DI5: Dynamically deactivate detection of rising edges | |
| | TRUE | Detection of rising edges is deactivated. |
| bDisableTPDigIn3...5Falling<br>BOOL | DI3 ... DI5: Dynamically deactivate detection of falling edges | |
| | TRUE | Detection of falling edges is deactivated. |
| bEnableTPDigIn3...5Window<br>BOOL | DI3 ... DI5: Activate acceptance window | |
| | TRUE | Acceptance window function is active:<br>• If the actual position is outside the acceptance window, whose starting position is set in C02813/x and whose end position is set in C02814/x, touch probe is automatically deactivated. |

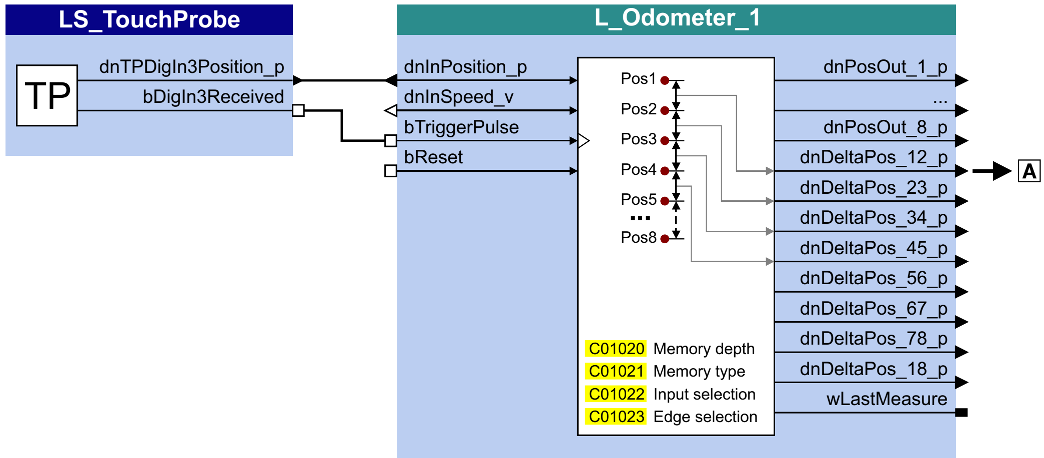
#### Outputs

| Output<br>Data type | Value/meaning | |
|---|---|---|
| bTPDigIn3...7Received<br>BOOL | TRUE | DI3 ... DI7: Touch probe received.<br>• The signal is only pending for one task cycle (1 ms). |
| dnTPDigIn3...7Position_p<br>DINT | DI3 ... DI7: Position measured by touch probe in [units] | |

### 7.4.2.1 Application example: "Position measurement"

The touch probe function can be combined with the L_Odometer FB for a position measurement. This FB is able to save position signals to a ring buffer and detect differences between two position signals.

In the wiring below, digital input DI3 is used to connect the touch probe sensor. For the sake of clarity, irrelevant inputs and outputs of the LS_TouchProbe SB are masked out.




A Delta position between two touch probes

[7-5] Wiring for position measurement

## 7.5 Configuring exception handling of the output terminals

Exception handling for the analog and digital output terminals in the event of an error can be set via decoupling configuration and decoupling values.

- Bit coded selection is carried out in C00441 for the analog output terminals, defining the events that will trigger decoupling.
- Bit coded selection is carried out in C00447 for the digital output terminals, defining the events that will trigger decoupling.

| Bit | Event |
|---|---|
| Bit 0 ☐ | SafeTorqueOff |
| Bit 1 ☐ | ReadyToSwitchOn |
| Bit 2 ☐ | SwitchedOn |
| Bit 3 ☐ | Reserved |
| Bit 4 ☐ | Trouble |
| Bit 5 ☐ | Fault |
| Bit 6 ☐ | Reserved |
| Bit 7 ☐ | Reserved |
| Bit 8 ☐ | Reserved |
| Bit 9 ☐ | Fail CAN_Management |
| Bit 10 ☐ | Reserved |
| Bit 11 ☐ | Reserved |
| Bit 12 ☐ | Reserved |
| Bit 13 ☐ | Reserved |
| Bit 14 ☐ | Reserved |
| Bit 15 ☐ | Reserved |

Finally, the following parameters define the value/status that the output terminals are to have when they are decoupled:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00442/1 | AOut1_U: Decoupling value | 0.00 | % |
| C00442/2 | AOut2_U: Decoupling value | 0.00 | % |
| C00442/3 | AOut1_I: Decoupling value | 0.00 | % |
| C00442/4 | AOut2_I: Decoupling value | 0.00 | % |
| C00448 | DigOut decoupling value | Bit coded | |

### Related topics:

- Configuring exception handling of the CAN PDOs (📖 666)

## 7.6 User-defined terminal assignment

In order to individually adapt the preconfigured assignment of the input/output terminals to your application, you can choose one of the following procedures:

A. In the »Engineer«:
   - Change the terminal assignment on the **Terminal assignment** tab.
   - Change the signal assignment on the **Application Parameters** tab, on the dialog level *Overview* → *Signal flow*.
   - Change the interconnections in the FB editor (on the I/O level).

B. In the »Engineer« or with the keypad:
   - Change the parameters for signal configuration in the parameters list.

**Note!**

If you change the preconfigured assignment of the input/output terminals, the terminal assignment will be a user-defined one. In C00007, control mode "0: Interconnection changed" will be shown.

**Tip!**

First of all, select a Lenze configuration useful for the purpose at hand by going to C00005 and selecting a technology application that matches your drive task and then going to C00007 and selecting an appropriate control mode. You will then have an application for which there is a signal flow, logical block links and terminal assignment.

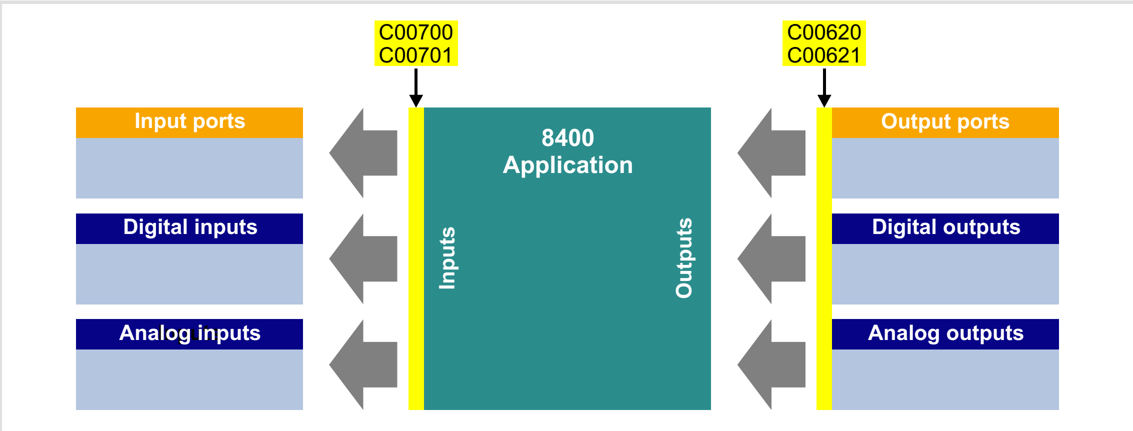
We recommend using the »Engineer« for the implementation of comprehensive user-defined drive solutions.

### 7.6.1 Source-destination principle

The I/O configuration of the input and output signals is carried out according to the source/destination principle:

- A connection always has a direction and therefore always has a source and a target.
- The inputs signals of the technology application are logically linked to the outputs of system blocks which represent the device input terminals.
- The inputs of system blocks that represent the device output terminals are logically linked to output signals of the technology application.

The following graphic illustrates the source/destination principle:




[7-6] Source-destination principle

Note the following:

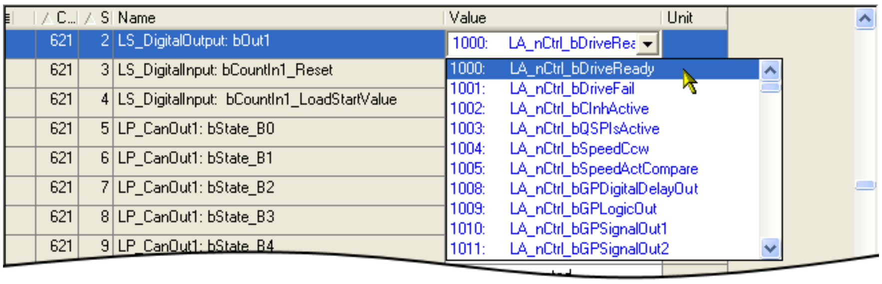
- An equipment input terminal can be logically linked to several inputs of the application block.
- Every input of the application block can only be logically linked to one input signal.
- An output of the application block can be logically linked to several device output terminals.

## 7.6.2 Changing the terminal assignment with the keypad

You can reconfigure the preconfigured terminal assignment with the keypad (and with the »Engineer«) by means of so-called configuration parameters.

- Each configuration parameter represents a signal input of a system block or application block.
- Each configuration parameter contains a selection list with output signals of the same type of data.
- Logical linking is thus carried out by selecting the output signal for the corresponding signal input.

In the following example, digital output 1 (**LS_DigitalOutput**.*bOut1* input) is logically linked to the status signal "Drive ready" (*LA_nCtrl_bDriveReady* output signal):




### Configuration parameters for the analog and digital output terminals

The preconfigured assignment of the analog and digital output terminals can be altered by means of the subcodes of C00620 and C00621:

| Parameter | Info | Lenze setting<br>Value Unit |
|---|---|---|
| **Analog outputs - terminal assignment** | | |
| C00620/1 | LS_AnalogOutput: nOut1_a (V) | 1003: LA_nCtrl_nMotorSpeedAct_a |
| C00620/39 | LS_AnalogOutput: nOut1_a (I) | 0: Not connected |
| C00620/38 | LS_AnalogOutput: nOut2_a (V) | 0: Not connected |
| C00620/40 | LS_AnalogOutput: nOut2_a (I) | 0: Not connected |
| **Digital outputs - terminal assignment** | | |
| C00621/1 | LS_DigitalOutput:bRelay | 1001: LA_nCtrl_bDriveFail |
| C00621/2 | LS_DigitalOutput:bOut1 | 1000: LA_nCtrl_bDriveReady |
| C00621/99 | LS_DigitalOutput: bOut2 | 0: Not connected |
| C00621/100 | LS_DigitalOutput: bOut3 | 0: Not connected |
| C00621/101 | LS_DigitalOutput: bOut HighCurrent | 0: Not connected |

Other subcodes (not shown here) allow the configuration of input signals of different system blocks and port blocks.

### Configuration parameters for the inputs of the technology application

The following parameters can be used to change the preconfigured assignment of the application inputs:

| Parameter | Info |
|---|---|
| **TA "Actuating drive speed":** Configuration parameters (📖 407) | |
| C00700/x | Analog connection list |
| C00701/x | Digital connection list |
| **TA "Table positioning":** Configuration parameters (📖 436) | |
| C00710/x | Analog connection list |
| C00711/x | Digital connection list |
| **TA "Switch-off positioning":** Configuration parameters (📖 464) | |
| C00760/x | Analog connection list |
| C00761/x | Digital connection list |

### Example

Task: Starting from the preset technology application "Actuating drive speed" and the "Terminals 0" control mode, the DI2 digital input is to be used for choosing an alternative acceleration/deceleration time for the main setpoint instead of for choosing the fixed setpoint 2/3. To do this, the DI2 digital input is not to be linked to the *bJogSpeed2* input but to the *bJogRamp1* input of the application module.

Procedure:

1. Use the keypad to go to the menu level **Applications → Actuating drive speed (conf.)**. This menu level contains all the configuration parameters of the "Actuating drive speed" technology application". ▸ Configuration parameters (📖 407)
2. Navigate to the configuration parameter LA_NCtrl: bJogSpeed2 (C00701/10) which represents the logical signal link of the application input *bJogSpeed2*.
3. Change the setting of C00701/10:
   Change selection "16001: DigIn_bIn2" in selection "0: Not interconnected".
4. Navigate to the configuration parameter LA_NCtrl: bJogRamp1 (C00701/13) which represents the logical signal link of the application input *bJogRamp1*.
5. Change the setting of C00701/13:
   Change selection "0: Not interconnected" in selection "16001: DigIn_bIn2".

**Tip!**

The example shows that, for each input of the application block, the associated configuration parameter (C00700/x or C00701/x) is only allowed to contain one source that you enter.

### 7.6.3 Changing the terminal assignment with the »Engineer«

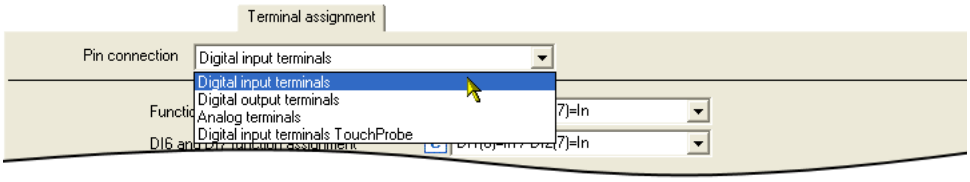
Whereas the configuration parameters referred to have to be parameterised with the keypad, implementation in the »Engineer« is much easier due to the availability of the corresponding dialogs. The following task illustrates the respective procedure.

Task: Starting from the preset technology application "Actuating drive speed" and the "Terminals 0" control mode, the DI2 digital input is to be used for choosing an alternative acceleration/deceleration time for the main setpoint instead of for choosing the fixed setpoint 2/3. To do this, the DI2 digital input is not to be linked to the *bJogSpeed2* input but to the *bJogRamp1* input of the application module.

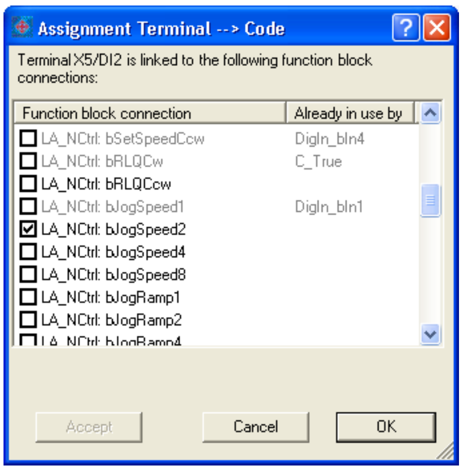
#### Possibility 1: Change terminal assignment by means of the Terminal Assignment tab

Procedure:

1. Go to the **Terminal Assignment** tab and select "Digital input terminals" in the **Control connections** list field:




2. Click on the ... button for the DI2 terminal in order to open the dialog box *Assignment Terminal --> Function block*.
   – In the list field, all block inputs that are currently logically linked to digital input DI2 are marked with a checkmark:

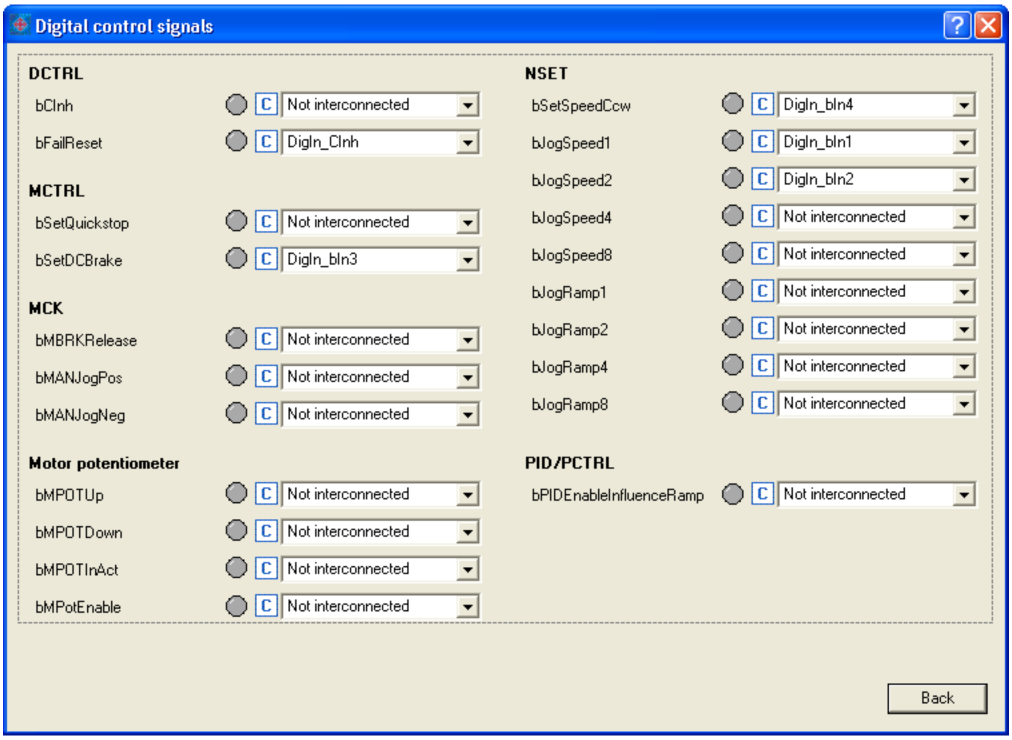



3. Remove checkmark for the connection **LA_NCtrl: bJogSpeed2** in order to cancel the existing logical link.
4. Set checkmark for connection **LA_NCtrl: bJogRamp1** in order to logically link this application input to digital input DI2.

**Possibility 2: Change terminal assignment by means of the signal flow shown**

Procedure:

1. Go to the **Application parameters** tab.
2. Go to the **Application Parameters** tab and click on the **Signal flow** button in order to change to the dialog level *Overview* ➞ *Signal flow*.
3. On the dialog level *Overview* ➞ *Signal flow*, click on the **Digital control signals** button in order to open the *Digital control signals* dialog box:

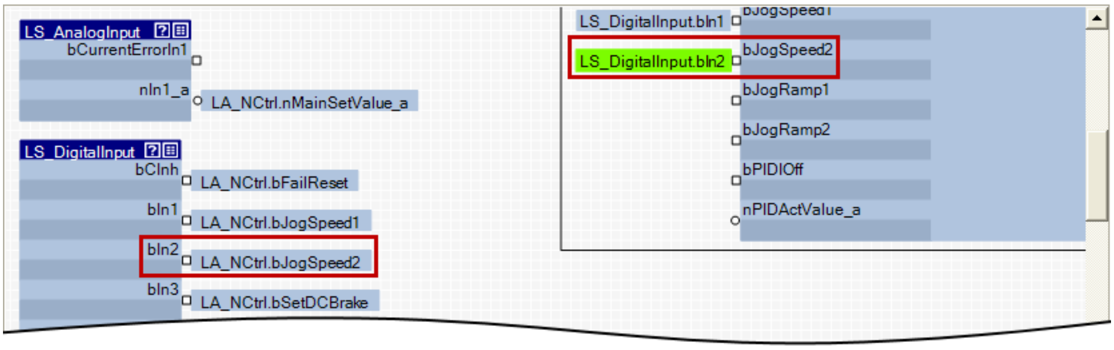



4. In the **bJogSpeed2** list field, set the selection "0: Not interconnected".
5. In the **bJogRamp1** list field, set the selection "16001: DigIn_bIn2".
6. Click on the **Back** button in order to close the dialog box again.

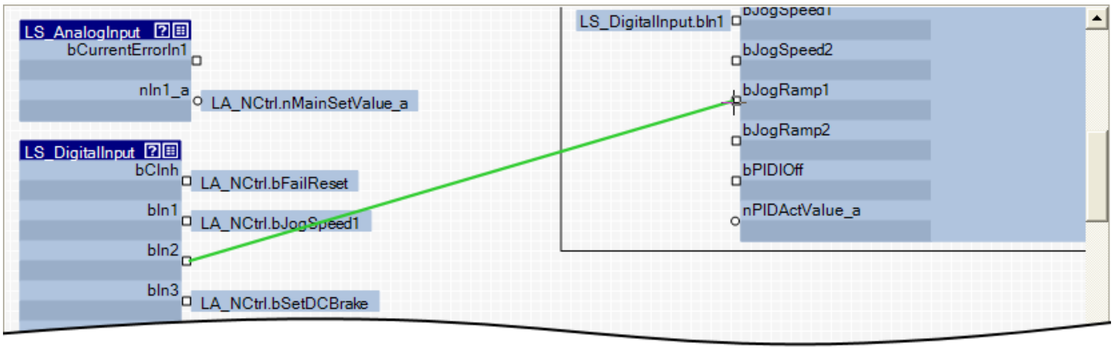
### Possibility 3: Change terminal assignment with the FB editor

Procedure:

1. Go to the **FB Editor** tab.
2. Delete the existing interconnection from **LS_DigitalInput**.*bIn2* to **LA_NCtrl**.*bJogSpeed2*:




3. Establish a new interconnection from **LS_DigitalInput**.*bIn2* to **LA_NCtrl**.*bJogRamp1*:




**Tip!**

You can find detailed information on how to use the FB editor of the »Engineer« in the main chapter entitled "Working with the FB Editor". (📖 1058)

# 8 Technology applications

A technology application is a drive solution equipped with Lenze's experience and know-how in which function and system blocks interconnected to a signal flow clearly show the basis for implementing typical drive tasks.

This chapter describes the handling and the functional range of the technology applications available for the 8400 HighLine drive controller.

**Note!**

Please note that the StateLine and HighLine device types differ with regard to the number, functional range, and flexibility of the technology applications offered.

## 8.1 General information

### 8.1.1 Purpose of the technology applications

The stepped Inverter Drives 8400 series provides solutions for simple to complex applications – depending on the user's experience and knowledge about the handling of drives and drive tasks.

Some drive tasks are similar in practice so that a few modifications of the corresponding technology applications can provide the required result very quickly.

On the one hand a great scope of standard drive tasks for frequency inverters is covered by the technology applications offered by Lenze, and on the other hand, the user is relieved from time-consuming programming activities.

Other important features of technology applications are:

- Direct implementation of drive tasks without recreating a function block interconnection inside the device
- Operation via keypad and/or operation via convenient operator dialogs in the «Engineer»
- Commissioning via few operating and diagnosing parameters (local keypad operation)
- Achieving a transparency as high as possible via the integrated functionality of the device by representing signal flow diagrams
- Provision of a basic functionality suitable and often sufficient for many applications

### 8.1.2 Application cases for a technology application

You should use a technology application if

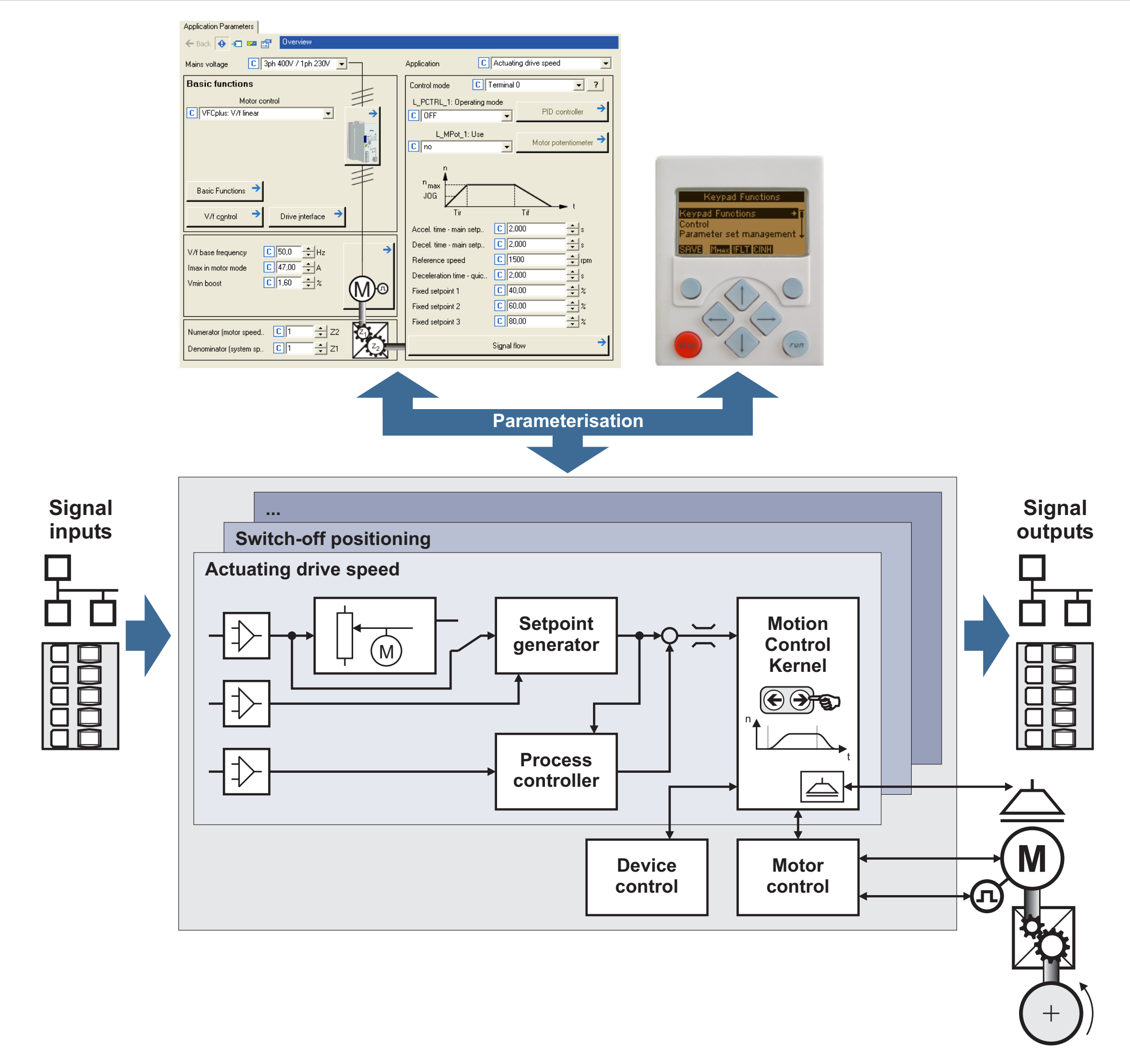
- the task can be solved completely or to a great extent by the basic functionality of the technology application.
- the end customer does not want to create the comprehensive core functions of the corresponding technology on his own.
- the creation time for a project is to be reduced by using the ready-made technology application
- the end customer wants to build upon the know-how of Lenze.

**Tip!**

If the end customer of the machine does not want to use ready-made Lenze functions, it is also possible to implement individual drive solutions with the HighLine version in the form of "free interconnections". In this case, the technology applications integrated in the device can be used as a basis to be adapted to the requirements by changes or extensions by means of the function block editor.

Detailed information on the creation or change of interconnections by means of the function block editor can be found in the main chapter "Working with the FB Editor". (📖 1058)

### 8.1.3 Basic software structure




[8-1] Software structure of the application function

The technology applications integrated in the drive controller provide the main signal flow for the implementation of a general or a special drive task.

Every technology application features an input interface for the connection of the signal sources (e.g. main setpoint) as well as an output interface for the control of output terminals and output ports.

### I/O level & application level

The interconnection of the interfaces is shown in the I/O level of the function block editor according to the control mode set in C00007. In the "deeper" application level, the main signal flow is realised in the form of an interconnection of various function and system blocks.

**Tip!**

Every application block features so-called "free inputs and outputs" which you can use to transfer signals from the I/O level to the application level and vice versa.

- In the Lenze setting, these connectors are hidden in the function block editor.
- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

### Motion Control Kernel

Important basic (drive) functions as well as further basic functionalities are implemented in the firmware of the drive controller in the so-called **Motion Control Kernel** (MCK) which can be accessed by the active technology application via defined internal interfaces. By this means the expensive creation of single function block interconnections is omitted so that the expenditure and the complexity for the realisation of standard functions is minimised.

The **Motion Control Kernel** is integrated in the main setpoint path and, depending on the set operating mode, it creates the required control and setpoint signals for the motor control and the drive interface.

**Tip!**

A detailed description of the basic functions implemented in the **Motion Control Kernel** can be found in the main chapter "Basic drive functions (MCK)". (📖 473)

### Parameter

The setting/parameterisation of internal functions, the selection of setpoints and the display of actual values is executed via parameters. A re-configuration of the TA interfaces is also possible via the corresponding configuration parameters.

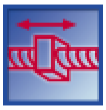
The access to parameters can either be executed with the keypad, the L-force »Engineer« or a master control per field bus communication.

## 8.2 Overview of available technology applications

**Technology application "Actuating drive speed"**

This technology application serves to solve speed-controlled drive tasks, e.g. conveyor drives (interconnected), extruders, test benches, vibrators, travelling drives, presses, machining systems, metering units.

▶ TA "Actuating drive speed" (📖 383)

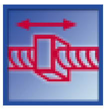


**Technology application "Table positioning"**

This technology application serves to solve position-controlled drive tasks which are normally controlled by a higher-level control via a fieldbus, e.g. transport facilities, rotary tables, storage and retrieval units, feed drives, metering units, hoists.

**Note:** An external sequence control is required for this TA!

▶ TA "Table positioning" (📖 410)



**"Switch-off positioning" technology application**

This technology application is used to solve speed-controlled drive tasks which require a pre-switch off or stopping at certain positions, e.g. roller conveyors and conveying belts. The pre-switch off is implemented by connecting switch-off sensors.

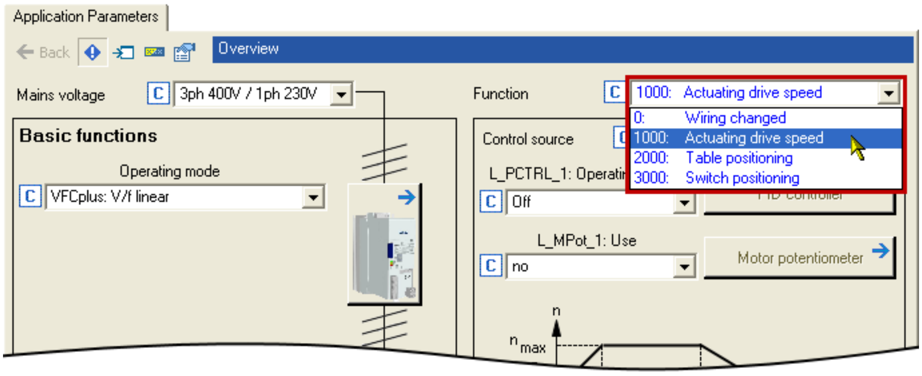
▶ TA "Switch-off positioning" (📖 440)

Further technology applications are in preparation.

## 8.3 Selection of the technology application and the control mode

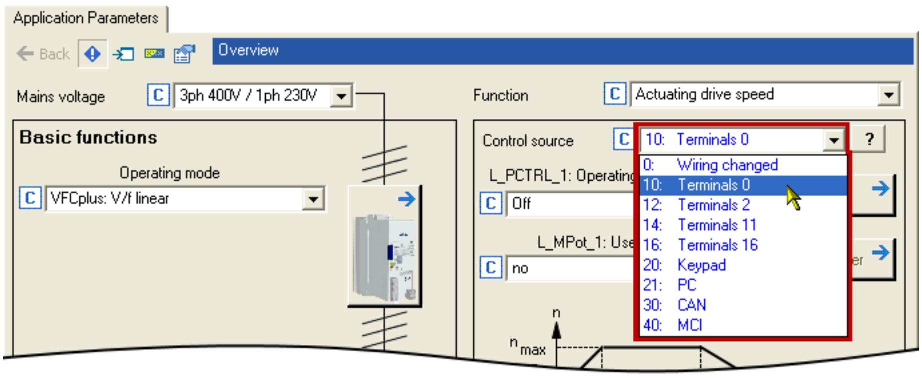
The technology application to be used is selected in C00005.

▶ You can select the technology application in the »Engineer« on the **Application parameter** tab via the **Application** list field:




Different control modes can be selected for every application in C00007. By selecting the control mode you set the way by which the technology application should be controlled, e.g. via terminals or via a fieldbus. The interconnection of the input/output terminals and ports shown in the FB editor in the I/O level changes accordingly.

▶ You can select the control mode in the »Engineer« on the **Application parameter** tab via the **Control mode** list field:




**Tip!**

You can infer the pre-configured assignment of the input/output terminals and ports for each control mode from the description of the corresponding technology application:

**TA "Actuating drive speed":** Terminal assignment of the control modes (🕮 394)

**TA "Table positioning":** Terminal assignment of the control modes (🕮 424)

**TA "Switch-off positioning":** Terminal assignment of the control modes (🕮 451)

Detailed information on the individual configuration of the input/output terminals can be found in the description of the I/O terminals in the subchapter "User-defined terminal assignment". (🕮 370)

## 8.4 TA "Actuating drive speed"

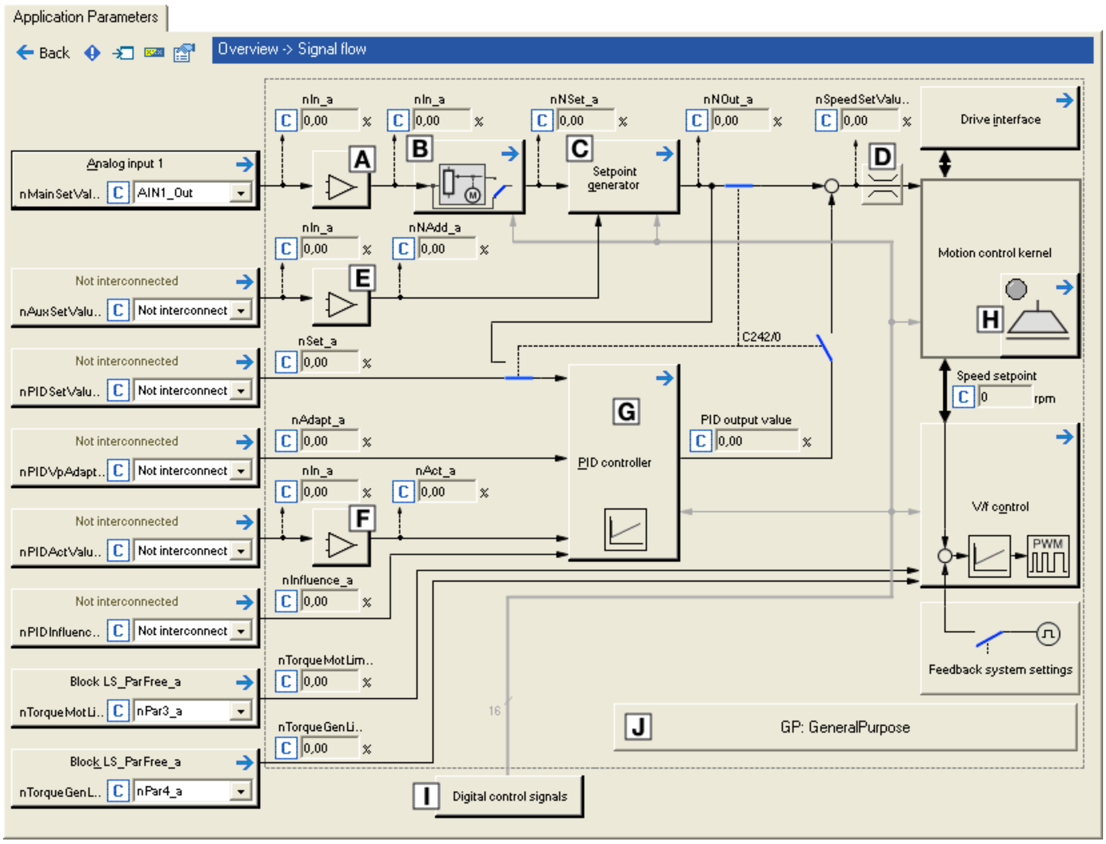
**Features**

- Pre-configured control modes for terminals and bus control (with predefined process data connection to the fieldbus)
- Free configuration of input and output signals
- Offset, gain, and negation of main setpoint, additional setpoint, actual process controller value
- Up to 15 fixed setpoints for speed and ramp time
- Adjustable setpoint ramp times
- Freely selectable, variable ramp shape
- Automatic holding brake control
- Quick stop (QSP) with adjustable ramp time
- Motor potentiometer function
- Process controller
- Load monitoring *(in preparation)*
- Integrated disposable "GeneralPurpose" functions:
  Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop
- Interface to the safety module (optional)
- Integration of encoder feedback

**Related topics:**

- Commissioning of the "Actuating drive speed" technology application (📖 49)

## 8.4.1 Basic signal flow




[8-2] Signal flow of the technology application "Actuating drive speed"

- A Main speed setpoint offset and gain (L_OffsetGainP_1)
- B Motor potentiometer function (L_MPot_1)
- C Setpoint generator (L_NSet_1)
- D Speed setpoint input limitation
- E Additional speed setpoint offset and gain (L_OffsetGainP_2)
- F Actual speed/sensor value offset and gain (L_OffsetGainP_3)
- G Process controller (L_PCTRL_1)
- H Holding brake control
- I Terminal assignment & display of digital control signals
- J Integrated disposable "GeneralPurpose" functions:
  Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop

### Selection of the main speed setpoint

The main speed setpoint is selected in the Lenze setting via the analog input 1.

- Offset and gain of this input signal can be adjusted in C00696 and C00670 for a simple adjustment of a setpoint encoder.
- Scaling: 16384 ≡ 100 % reference speed (C00011)
- The main setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.
- Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.
- For a detailed functional description see FB L_NSet.

### Motor potentiometer function

Alternatively, the main speed setpoint can be generated via a motor potentiometer function.

- In the Lenze setting, the motor potentiometer function is deactivated.
- Activation is possible via C00806 or via the *bMPotEnable* input.
- The behaviour of the motor potentiometer during switch-on of the drive system can be selected in C00805.
- For a detailed functional description see FB L_MPot.

### Optional selection of an additional speed setpoint

You can optionally select an additional speed setpoint (e.g. as a correcting signal).

- The additional speed setpoint can be linked arithmetically with the main speed setpoint behind the ramp function generator.
- You must set the setpoint arithmetic to "1: NOut = NSet + NAdd" in C00190 in order to activate the additional speed setpoint.
- Offset and gain of this input signal can be set in C00697 and C00671 for a simple signal adjustment of a setpoint encoder.
- Scaling: 16384 ≡ 100 % reference speed (C00011)
- The acceleration and deceleration time for the additional speed setpoint can be set in C00220 and C00221.
- For a detailed functional description see FB L_NSet.

**Tip!**

In the case of a grinding machine, the additional speed setpoint can, for instance, be used to control a constant circumferential speed while the grinding disk diameter is reduced.

## 8.4.2 Internal interfaces | application block "LA_NCtrl"

**Note!**

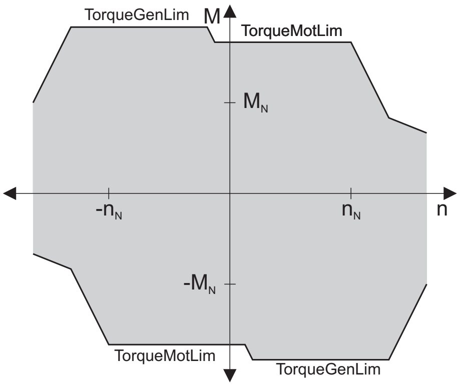
The connectors grayed out in the following table are hidden in the function block editor in the Lenze setting.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

### Inputs

| Identifier<br>Data type | | Information/possible settings |
|---|---|---|
| wCANDriveControl<br>WORD | | Control word via system bus (CAN) to device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. |
| wMCIDriveControl<br>WORD | | Control word via communication module (e.g. PROFIBUS) to device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. |
| wSMControl<br>WORD | | Interface to the optional safety system.<br>• Setting control bit 0 ("SafeStop1") in this control word causes e.g. the automatic deceleration of the drive to standstill within this application (in the **Motion Control Kernel**).<br>• See the "Interface to safety system" subchapter of the chapter on basic drive functions for a detailed description of the control bits. |
| bCInh<br>BOOL | | Enable/Inhibit controller |
| | FALSE | Enable controller: The controller switches to the "OperationEnabled" device status if no other source of a controller inhibit is active.<br>• C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit. |
| | TRUE | Inhibit controller (controller inhibit): The controller switches to the "SwitchedOn" device status. |
| bFailReset<br>BOOL | | Reset error messages<br>In the Lenze setting this input is connected to the digital input controller enable so that a possibly existing error message is reset together with the controller enable (if the cause for the fault is eliminated). |
| | TRUE | The current fault is reset, if the cause for the fault is eliminated.<br>• If the fault still exists, the error status remains unchanged. |
| bSetQuickstop<br>BOOL | | Enable quick stop (QSP)<br>• Also see device command "Activate/Deactivate quick stop". |
| | TRUE | Activate quick stop<br>• Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act} = 0$).<br>• The motor is kept at a standstill during closed-loop operation.<br>• A pulse inhibit (CINH) is set if the auto DCB function has been activated via C00019. |
| | FALSE | Deactivate quick stop<br>• The quick stop is deactivated if no other source for the quick stop is active.<br>• C00159 provides a bit-coded representation of active sources/causes for the quick stop. |

| Identifier | Data type | Information/possible settings | |
|---|---|---|---|
| bSetDCBrake | BOOL | Manual DC-injection braking (DCB)<br>• Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking". | |
| | | **Note!**<br>Holding braking is not possible when this braking mode is used!<br>For controlling a holding brake with low rate of wear, use the basic function "Holding brake control". | |
| | | FALSE | Deactivate DC-injection braking. |
| | | TRUE | Activate DC-injection braking, i.e. the drive is brought to a standstill by means of DC-injection braking.<br>• The braking effect stops when the rotor is at standstill.<br>• After the hold time (C00107) has expired, the controller sets the pulse inhibit (CINH). |
| bRFG_Stop | BOOL | Ramp function generator: Maintain the current value of the main setpoint integrator<br>• The speed, for instance, of a running ramp process is immediately kept constant when *bRFG_Stop* is activated. At the same time, the acceleration/deceleration jumps to the value "0".<br>• For a detailed functional description see the L_NSet FB. | |
| | | TRUE | The current value of the main setpoint integrator is held. |
| bRFG_0 | BOOL | Ramp function generator: Lead the main setpoint integrator to "0" within the current Ti times<br>• For a detailed functional description see the L_NSet FB. | |
| | | TRUE | The current value of the main setpoint integrator is led to "0" within the Ti time set. |
| nVoltageAdd_a | INT | Additive voltage impression<br>• An additional setpoint for the motor voltage can be specified via this process input.<br>• If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time.<br>• If the value is negative, the voltage is reduced.<br>• Scaling: 16384 ≡ 1000 V | |
| | | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! | |
| nBoost_a | INT | Additional setpoint for the motor voltage at speed = 0<br>• The entire voltage-frequency characteristic is provided with an offset.<br>• Scaling: 16384 ≡ 1000 V | |
| | | **Stop!**<br>Values selected too high may cause the motor to heat up due to the resulting current! | |
| nPWMAngleOffset | INT | Additional offset for the electrical angle of rotation<br>• If a torque is connected, e.g. dynamic acceleration processes can be generated.<br>• Scaling: ±32767 ≡ ±180 ° angle of rotation | |

| Identifier / Data type | Information/possible settings | |
|---|---|---|
| nTorqueMotLim_a<br>nTorqueGenLim_a<br>INT | Torque limitation in motor mode and in generator mode<br>• These input signals are directly transferred to the motor control to limit the controller's maximum torque in motor and generator mode.<br>• The drive cannot output a higher torque in motor/generator mode than set here.<br>• The applied values (any polarity) are internally interpreted as absolute values.<br>• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.<br>• If sensorless vector control (SLVC) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| | Torque limits in motor and generator mode:   | |
| bSetSpeedCcw<br>BOOL | Change of direction of rotation<br>• For instance if a motor or gearbox is fixed laterally reversed to a machine part, but the setpoint selection should still be executed for the positive direction of rotation. | |
| | FALSE | Direction of rotation to the right (Cw) |
| | TRUE | Direction of rotation to the left (Ccw) |
| bRLQCw<br>BOOL | Activate clockwise rotation (fail-safe)<br>• For a detailed functional description see FB L_RLQ. | |
| | FALSE | Quick stop |
| | TRUE | Clockwise rotation |
| bRLQCcw<br>BOOL | Activate counter-clockwise rotation (fail-safe)<br>• For a detailed functional description see the L_RLQ FB. | |
| | FALSE | Quick stop |
| | TRUE | Counter-clockwise rotation |
| nMainSetValue_a<br>INT | Main speed setpoint<br>• Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The main setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.<br>• Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.<br>• For a detailed functional description see the L_NSet FB. | |

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nAuxSetValue_a<br>INT | Additional speed setpoint<br>• The additional speed setpoint can be linked arithmetically with the main speed setpoint behind the ramp function generator.<br>• You must set the setpoint arithmetic to "1: NOut = NSet + NAdd" in C00190 in order to activate the additional speed setpoint.<br>• Offset and gain of this input signal can be set in C00697 and C00671 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The acceleration and deceleration time for the additional speed setpoint can be set in C00220 and C00221.<br>• For a detailed functional description see the L_NSet FB. |
| bJogSpeed1<br>bJogSpeed2<br>BOOL<br>bJogSpeed4<br>bJogSpeed8<br>BOOL | Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint<br>• A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via these selection inputs.<br>• The four selection inputs are binary coded, therefore 15 fixed setpoints can be selected.<br>• In the case of binary coded selection "0" (all inputs = FALSE or not assigned), main setpoint *nMainSetValue_a* is active.<br>• The selection of the fixed setpoints is carried out in C00039/1...15 in [%] based on the reference speed (C00011).<br>• For a detailed functional description see the L_NSet FB. |
| bJogRamp1<br>bJogRamp2<br>BOOL<br>bJogRamp4<br>bJogRamp8<br>BOOL | Selection inputs for alternative acceleration/deceleration times for the main setpoint<br>• The four selection inputs are binary coded, therefore 15 alternative acceleration/deceleration times can be selected.<br>• For main setpoint *nMainSetValue_a*, the set acceleration time (C00012) and deceleration time (C00013) are active in the case of the binary coded selection "0" (all inputs = FALSE or not assigned).<br>• Alternative acceleration times are selected in C00101/1...15.<br>• The selection of the alternative deceleration times is carried out in C00103/1...15.<br>• For a detailed functional description see the L_NSet FB. |
| **Motor potentiometer**<br>Alternatively to the input signal *nMainSetValue_a*, the main setpoint can also be generated by a motor potentiometer function.<br>• In the Lenze setting, the motor potentiometer function is deactivated.<br>• Activation is possible via C00806 or via the *bMPotEnable* input.<br>• The behaviour of the motor potentiometer during switch-on of the drive system can be selected in C00805.<br>• For a detailed functional description see the L_MPot FB. | |
| bMPotEnable<br>BOOL | Activating the motor potentiometer function<br>• This input and C00806 are OR'd.<br>TRUE: The motor potentiometer function is active; the speed setpoint can be changed via the *bMPotUp* and *bMPotDown* control inputs. |
| bMPotUp<br>BOOL | Increasing the speed setpoint<br>TRUE: Approach the upper speed limit value set in C00800 with the acceleration time set in C00802. |
| bMPotInAct<br>BOOL | Activating the inactive function<br>TRUE: The speed setpoint behaves according to the inactive function set in C00804.<br>• In the Lenze setting, the speed setpoint is maintained. |
| bMPotDown<br>BOOL | Decreasing the speed setpoint<br>TRUE: Approach the lower speed limit value set In C00801 with the deceleration time set in C00803. |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| **Process controller**<br>• In the Lenze setting, the process controller is deactivated.<br>• The activation is executed by selecting the operating mode in C00242.<br>• For a detailed functional description see FB L_PCTRL. | | |
| bPIDEnableInfluenceRamp<br>BOOL | Activate ramp for influencing factor | |
| | FALSE | Influencing factor of the PID controller is ramped down to "0". |
| | TRUE | Influencing factor of the PID controller is ramped up to the value *nPIDInfluence_a*. |
| bPIDIOff<br>BOOL | Switch off I-component of process controller<br>• In conjunction with the operating mode set in C00242 (Lenze setting: "Off"). | |
| | TRUE | I-component of the process controller is switched off. |
| nPIDVpAdapt_a<br>INT | Adaptation of gain Vp set in C00222 in percent<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Changes can be done online. | |
| nPIDSetValue_a<br>INT | Sensor and process setpoint for operating modes 2, 4 and 5<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| nPIDActValue_a<br>INT | Speed or actual sensor value (actual process value)<br>• Offset and gain for this input signal can be set in C00698 and C00672.<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| nPIDInfluence_a<br>INT | Limitation of the influencing factor in percent<br>• The influence factor of the PID controller can be limited to a certain value (-199.99% ... + 199.99%) via *nPIDInfluence_a*.<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| **MCK basic functions** | | |
| bMBRKRelease<br>BOOL | Holding brake control: Release/apply brake<br>• In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). | |
| | FALSE | Apply brake.<br>• During automatic operation, the internal brake logic controls of the brake. |
| | TRUE | Release brake manually (forced release).<br>• **Note!** The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If the brake control has inhibited the controller, this inhibit is deactivated again.<br>• In semi-automatic operation, the brake is released including feedforward control. |
| bManJogPos<br>bManJogNeg<br>BOOL | reserved (inputs are not interconnected on the application level) | |
| **GP: GeneralPurpose**<br>The following inputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | | |
| nGPAnalogSwitchIn1_a<br>nGPAnalogSwitchIn2_a<br>INT | Analog switch: Input signals<br>• The input signal selected via the selection input *bGPAnalogSwitchSet* is output at output *nGPAnalogSwitchOut_a*. | |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bGPAnalogSwitchSet<br>BOOL | Analog switch: Selection input | |
| | FALSE | *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn1_a* |
| | TRUE | *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn2_a* |
| nGPArithmetikIn1_a<br>nGPArithmetikIn2_a<br>INT | Arithmetic: Input signals<br>• The arithmetic function is selected in C00338.<br>• The result is output at output *nGPArithmetikOut_a*. | |
| nGPMulDivIn_a<br>INT | Multiplication/Division: Input signal<br>• The factor for the multiplication can be set in C00699/1 (numerator) and C00699/2 (denominator).<br>• The result is output at output *nGPMulDivOut_a*. | |
| bGPDigitalDelayIn<br>BOOL | Binary delay element: Input signal<br>• The on-delay can be set in C00720/1.<br>• The off-delay can be set in C00720/2.<br>• The time-delayed input signal is output at output *bGPDigitalDelayOut*. | |
| bGPLogicIn1<br>bGPLogicIn2<br>bGPLogicIn3<br>BOOL | Binary logic: Input signals<br>• The logic operation is selected in C00820.<br>• The result is output at output *bGPLogicOut*. | |
| nGPCompareIn1_a<br>nGPCompareIn2_a<br>INT | Analog comparison: Input signals<br>• The comparison operation is selected in C00680.<br>• Hysteresis and window size can be set in C00680 and C00682.<br>• If the comparison statement is true, the output *bGPCompareOut* will be set to TRUE. | |
| bGPDFlipFlop_InD<br>bGPDFlipFlop_InClk<br>bGPDFlipFlop_InClr<br>BOOL | D-FlipFlop: Input signals<br>• Data, clock and reset input | |
| **Free inputs**<br>The following inputs can freely be interconnected on the application level.<br>The signals can be transferred from the I/O level to the application level via these inputs. | | |
| bFreeIn1 ... bFreeIn8<br>BOOL | Free inputs for digital signals | |
| wFreeIn1 ... wFreeIn4<br>WORD | Free inputs for 16-bit signals | |

## Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| wDriveControlStatus<br>WORD | Status word of the controller (based on DSP-402)<br>• The status word contains information on the currents status of the drive controller.<br>• See the "wDeviceStatusWord status word" subchapter of the chapter on device control for a detailed description of the bit assignment. | |
| wStateDetermFailNoLow<br>WORD | Display of the status determining error (LOW word) | |
| wStateDetermFailNoHigh<br>WORD | Display of the status determining error (HIGH word) | |
| bDriveFail<br>BOOL | TRUE | Drive controller in error status.<br>• "Fault" device status is active. |
| bDriveReady<br>BOOL | TRUE | Controller is ready for operation.<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the controller is still inhibited by the user (controller inhibit). |

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bCInhActive<br>BOOL | TRUE | Controller inhibit is active. |
| bQSPIsActive<br>BOOL | TRUE | Quick stop is active. |
| bSpeedCcw<br>BOOL | Current direction of rotation | |
| | FALSE | Direction of rotation to the right (Cw) |
| | TRUE | Direction of rotation to the left (Ccw) |
| bSpeedActCompare<br>BOOL | Result of the speed comparison | |
| | TRUE | During open-loop operation:<br>Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>Actual speed value < Comparison value (C00024) |
| bOverLoadActive<br>BOOL | In preparation (output is not interconnected on the application level) | |
| bUnderLoadActive<br>BOOL | In preparation (output is not interconnected on the application level) | |
| bImaxActive<br>BOOL | "Current setpoint inside the limitation" status signal | |
| | TRUE | The current setpoint is internally limited (the drive controller operates at the maximum current limit). |
| bSpeedSetReached<br>BOOL | Status signal "setpoint = 0" | |
| | TRUE | Speed setpoint from the ramp function generator = 0 |
| bSpeedActEqSet<br>BOOL | TRUE | Actual speed value = speed setpoint |
| nMotorCurrent_a<br>INT | Current stator current/effective motor current<br>• Scaling: 16384 ≡ 100 % $I_{max\_mot}$ (C00022) | |
| nMotorSpeedSet_a<br>INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorSpeedAct_a<br>INT | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorTorqueAct_a<br>INT | Actual torque<br>• In the "VFC (+encoder)" operating mode of the motor control, this value is determined from the current motor current and corresponds to the actual torque only by approximation.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| nDCVoltage_a<br>INT | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V | |
| nMotorVoltage_a<br>INT | Current motor voltage/inverter output voltage<br>• Scaling: 16384 ≡ 1000 V | |
| **MCK basic functions** | | |
| bBrakeReleaseOut<br>BOOL | Holding brake control: Trigger signal for the holding brake control switching element via a digital output<br>• Use bit 0 in C02582 to activate inverted switching element triggering. | |
| | FALSE | Apply brake. |
| | TRUE | Release brake. |
| bBrakeReleased<br>BOOL | Holding brake control: "Brake released" considering the brake release time<br>• When the holding brake is triggered to close, *bBrakeReleased* is immediately set to FALSE even if the brake closing time has not yet elapsed! | |
| | TRUE | Brake released (when the brake release time has elapsed). |
| **GP: GeneralPurpose**<br>The following outputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | | |

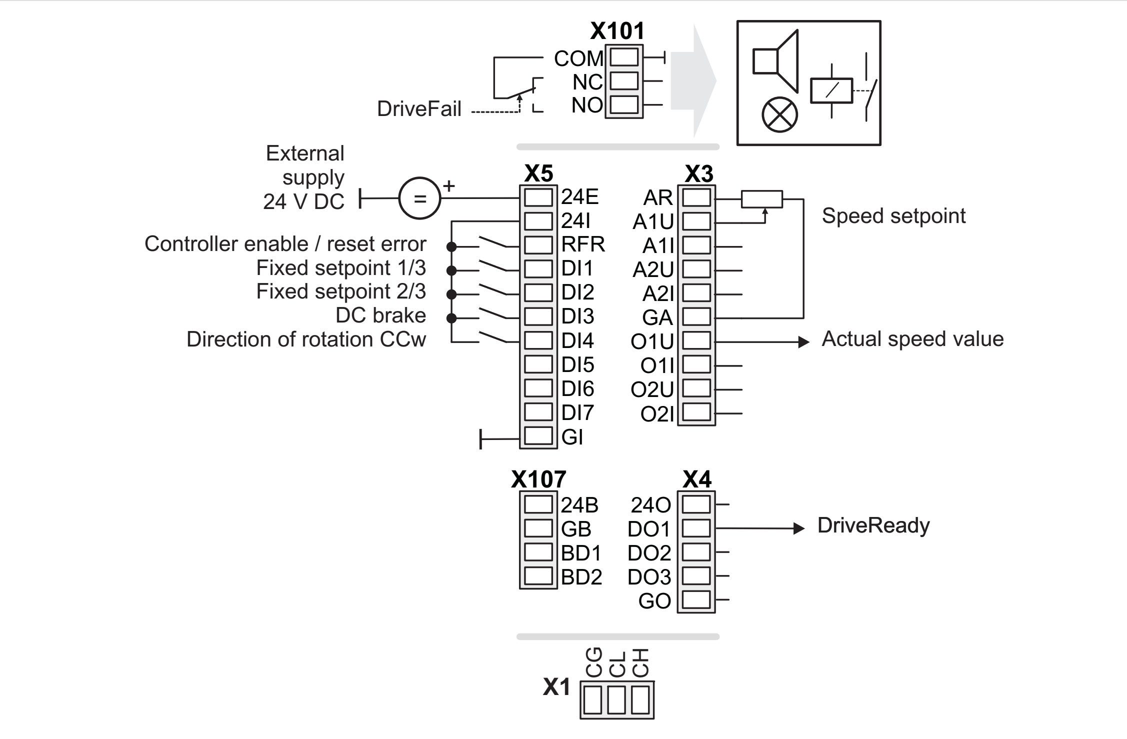
| Identifier<br>Data type | Value/meaning |
|---|---|
| nGPAnalogSwitchInOut_a<br>INT | Analog switch: Output signal |
| nGPArithmetikOut_a<br>INT | Arithmetic: Output signal |
| nGPMulDivOut_a<br>INT | Multiplication/Division: Output signal |
| bGPDigitalDelayOut<br>BOOL | Binary delay element: Output signal |
| bGPLogicOut<br>BOOL | Binary logic: Output signal |
| bGPCompareOut<br>BOOL | Analog comparison: Output signal |
| bGPSignalOut1<br>...<br>bGPSignalOut4<br>BOOL | Binary signal monitor: Output signals<br>• The signal sources to be output are selected in C00411/1...4.<br>• A bit coded inversion of the output signals can be parameterised in C00412. |
| nGPSignalOut1_a<br>...<br>nGPSignalOut4_a<br>BOOL | Analog signal monitor: Output signals<br>• The signal sources to be output are selected in C00410/1...4.<br>• Gain and offset for each output signal can be parameterised in C00413/1...8. |
| bGPDFlipFlop_Out<br>BOOL | D-FlipFlop: Output signal |
| bGPDFlipFlop_NegOut<br>BOOL | D-FlipFlop: Negated output signal |
| **Free outputs**<br>The following outputs can freely be interconnected on the application level.<br>The signals from the application level can be transferred to the I/O level via these outputs. | |
| bFreeOut1 ... bFreeOut8<br>BOOL | Free outputs for digital signals |
| wFreeOut1 ... wFreeOut4<br>WORD | Free outputs for 16-bit signals |

### 8.4.3 Terminal assignment of the control modes

The following comparison provides information about which inputs/outputs of the application block **LA_NCtrl** are interconnected to the digital and analog input/output terminals of the drive controller in the different control modes.

<table>
<tr><th rowspan="2"></th><th colspan="8">Control mode (C00007)</th></tr>
<tr><th>10: Terminals 0</th><th>12: Terminals 2</th><th>14: Terminals 11</th><th>16: Terminal 16</th><th>20: Keypad</th><th>21: PC</th><th>30: CAN</th><th>40: MCI</th></tr>
<tr><td colspan="9">Digital input terminals</td></tr>
<tr><td>X5/RFR</td><td colspan="8">Controller enable / Reset of error message<br>bFailReset</td></tr>
<tr><td>X5/DI1</td><td colspan="2">Fixed setpoint 1/3<br>bJogSpeed1</td><td>Change of direction of rotation<br>bSetSpeedCcw</td><td>Fixed setpoint 1/3<br>bJogSpeed1</td><td>-</td><td>-</td><td colspan="2">Quick stop<br>bSetQuickstop</td></tr>
<tr><td>X5/DI2</td><td colspan="2">Fixed setpoint 2/3<br>bJogSpeed2</td><td>Activate manual DC-injection braking (DCB)<br>bSetDCBrake</td><td>Fixed setpoint 2/3<br>bJogSpeed2</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>X5/DI3</td><td>Activate manual DC-injection braking (DCB)<br>bSetDCBrake</td><td>Quick stop<br>bSetQuickstop</td><td>Motor potentiometer: Increase speed<br>bMPotUp</td><td>CW rotation quick stop<br>bRLQCw</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>X5/DI4</td><td colspan="2">Change of direction of rotation<br>bSetSpeedCcw</td><td>Motor potentiometer: Decrease speed<br>bMPotDown</td><td>CCW rotation quick stop<br>bRLQCcw</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>X5/DI5 ... DI7</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="9">Analog input terminals</td></tr>
<tr><td>X3/A1U, A1I</td><td colspan="4">Main speed setpoint<br>nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011)</td><td>-</td><td>-</td><td colspan="2">Additional speed setpoint<br>nAuxSetValue_a<br>10 V ≡ 100 % reference speed (C00011)</td></tr>
<tr><td>X3/A2U, A2I</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="9">Digital output terminals</td></tr>
<tr><td>X4/DO1</td><td colspan="8">Status "Drive is ready"<br>bDriveReady</td></tr>
<tr><td>X4/DO2 ... DO3</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>X107/BD1, BD2</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>X101/COM, NO</td><td colspan="8">Status "Error is pending"<br>bDriveFail</td></tr>
<tr><td colspan="9">Analog output terminals</td></tr>
<tr><td>X3/O1U</td><td colspan="8">Actual speed value<br>nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011)</td></tr>
<tr><td>X3/O1I</td><td rowspan="3">-</td><td rowspan="3">-</td><td rowspan="3">-</td><td rowspan="3">-</td><td rowspan="3">-</td><td rowspan="3">-</td><td rowspan="3">-</td><td rowspan="3">-</td></tr>
<tr><td>X3/O2U</td></tr>
<tr><td>X3/O2I</td></tr>
</table>

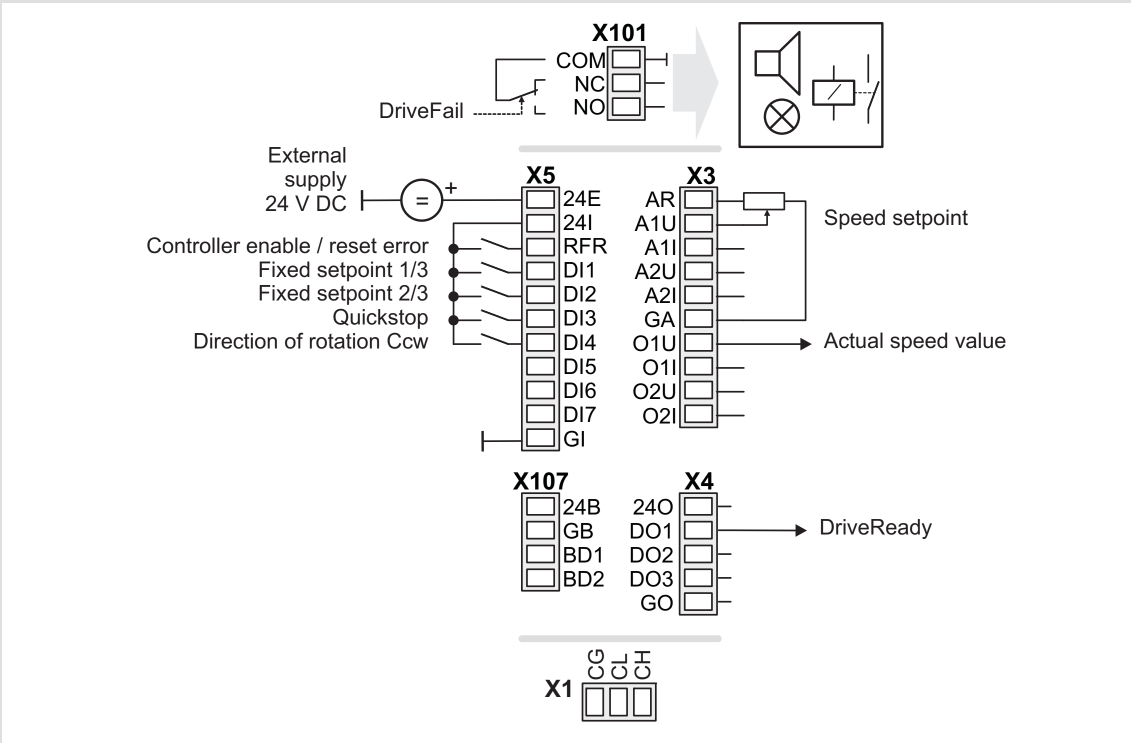
### 8.4.3.1 Terminals 0




| Connection | Assignment |
|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail |
| | |
| **X5/RFR** | LA_NCtrl.bFailReset |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 |
| **X5/DI3** | LA_NCtrl.bSetDCBrake |
| **X5/DI4** | LA_NCtrl.bSetSpeedCcw |
| **X5/DI5** | - |
| **X5/DI6** | - |
| **X5/DI7** | - |
| | |
| **X107/BD1** | - |
| **X107/BD2** | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| **X3/A1U** | LA_NCtrl.nMainSetValue_a * |
| **X3/A1I** | - |
| **X3/A2U** | - |
| **X3/A2I** | - |
| **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X3/O1I** | - |
| **X3/O2U** | - |
| **X3/O2I** | - |
| | * 10 V ≡ 100 % reference speed (C00011) |
| **X4/DO1** | LA_NCtrl.bDriveReady |
| **X4/DO2** | - |
| **X4/DO3** | - |

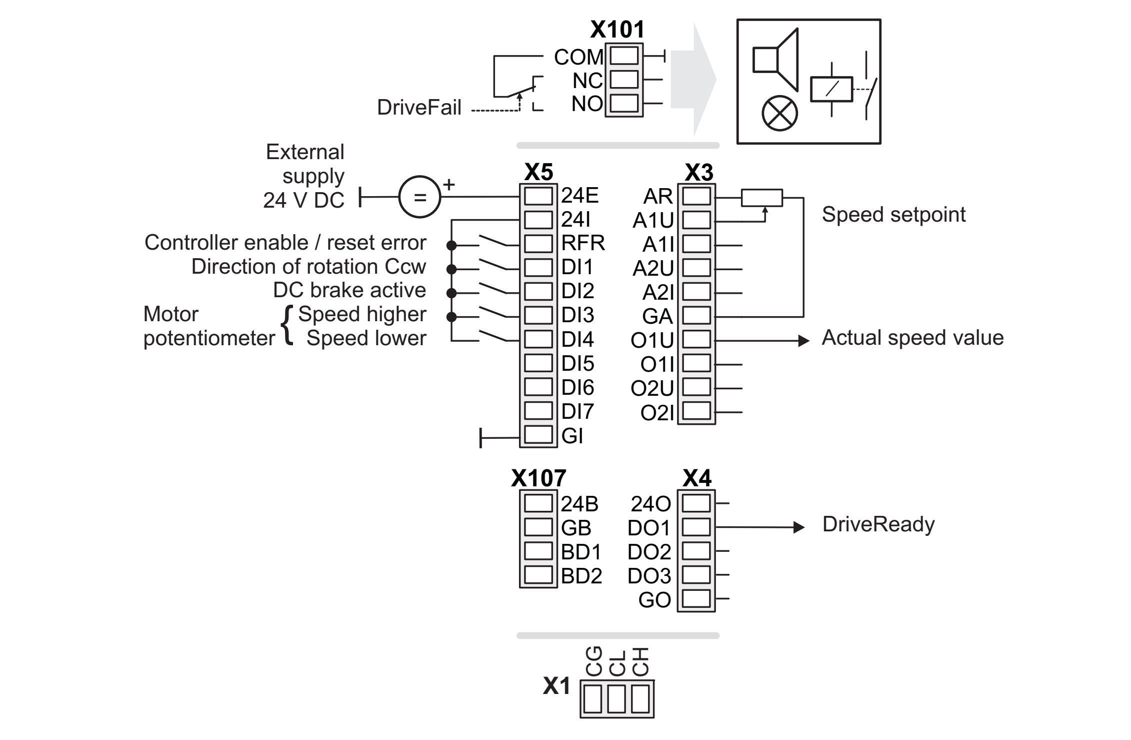
### 8.4.3.2 Terminals 2




| Connection | Assignment |
|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail |
| | |
| **X5/RFR** | LA_NCtrl.bFailReset |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 |
| **X5/DI3** | LA_NCtrl.bSetQuickstop |
| **X5/DI4** | LA_NCtrl.bSetSpeedCcw |
| **X5/DI5** | - |
| **X5/DI6** | - |
| **X5/DI7** | - |
| | |
| **X107/BD1** | - |
| **X107/BD2** | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| **X3/A1U** | LA_NCtrl.nMainSetValue_a * |
| **X3/A1I** | - |
| **X3/A2U** | - |
| **X3/A2I** | - |
| **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X3/O1I** | - |
| **X3/O2U** | - |
| **X3/O2I** | - |
| | * 10 V ≡ 100 % reference speed (C00011) |
| **X4/DO1** | LA_NCtrl.bDriveReady |
| **X4/DO2** | - |
| **X4/DO3** | - |

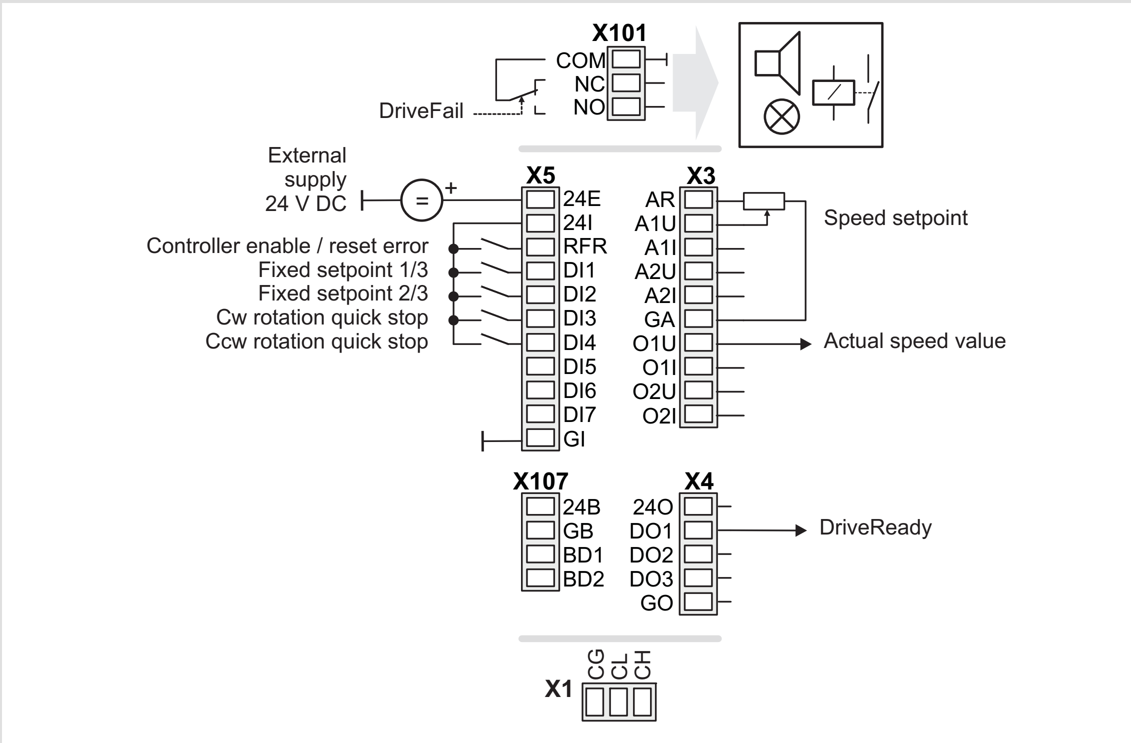
### 8.4.3.3 Terminals 11




| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_NCtrl.bDriveFail |
| | |
| X5/RFR | LA_NCtrl.bFailReset |
| X5/DI1 | LA_NCtrl.bSetSpeedCcw |
| X5/DI2 | LA_NCtrl.bSetDCBrake |
| X5/DI3 | LA_NCtrl.bMPotUp |
| X5/DI4 | LA_NCtrl.bMPotDown |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | - |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_NCtrl.nMainSetValue_a * |
| X3/A1I | - |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_NCtrl.nMotorSpeedAct_a * |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | * 10 V ≡ 100 % reference speed (C00011) |
| X4/DO1 | LA_NCtrl.bDriveReady |
| X4/DO2 | - |
| X4/DO3 | - |

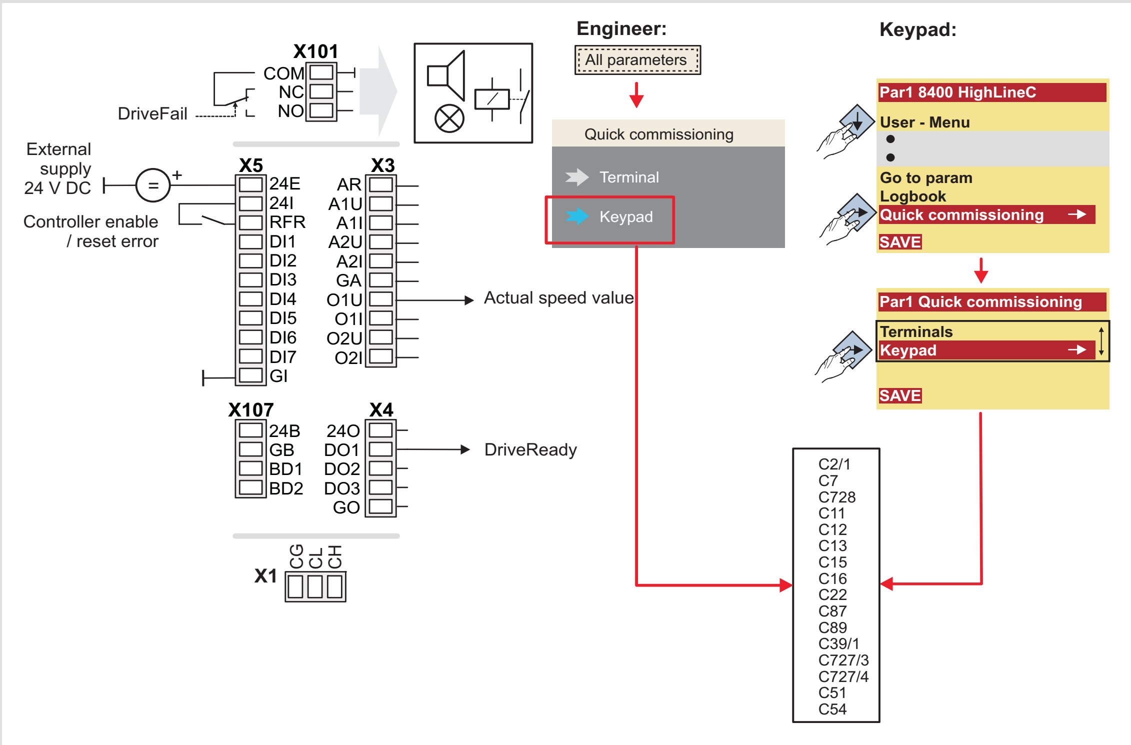
### 8.4.3.4 Terminal 16




| Connection | Assignment |
|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail |
| | |
| **X5/RFR** | LA_NCtrl.bFailReset |
| **X5/DI1** | LA_NCtrl.bJogSpeed1 |
| **X5/DI2** | LA_NCtrl.bJogSpeed2 |
| **X5/DI3** | LA_NCtrl. bRLQCw |
| **X5/DI4** | LA_NCtrl. bRLQCcw |
| **X5/DI5** | - |
| **X5/DI6** | - |
| **X5/DI7** | - |
| | |
| **X107/BD1** | - |
| **X107/BD2** | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| **X3/A1U** | LA_NCtrl.nMainSetValue_a * |
| **X3/A1I** | - |
| **X3/A2U** | - |
| **X3/A2I** | - |
| **X3/O1U** | LA_NCtrl.nMotorSpeedAct_a * |
| **X3/O1I** | - |
| **X3/O2U** | - |
| **X3/O2I** | - |
| | * 10 V ≡ 100 % reference speed (C00011) |
| **X4/DO1** | LA_NCtrl.bDriveReady |
| **X4/DO2** | - |
| **X4/DO3** | - |

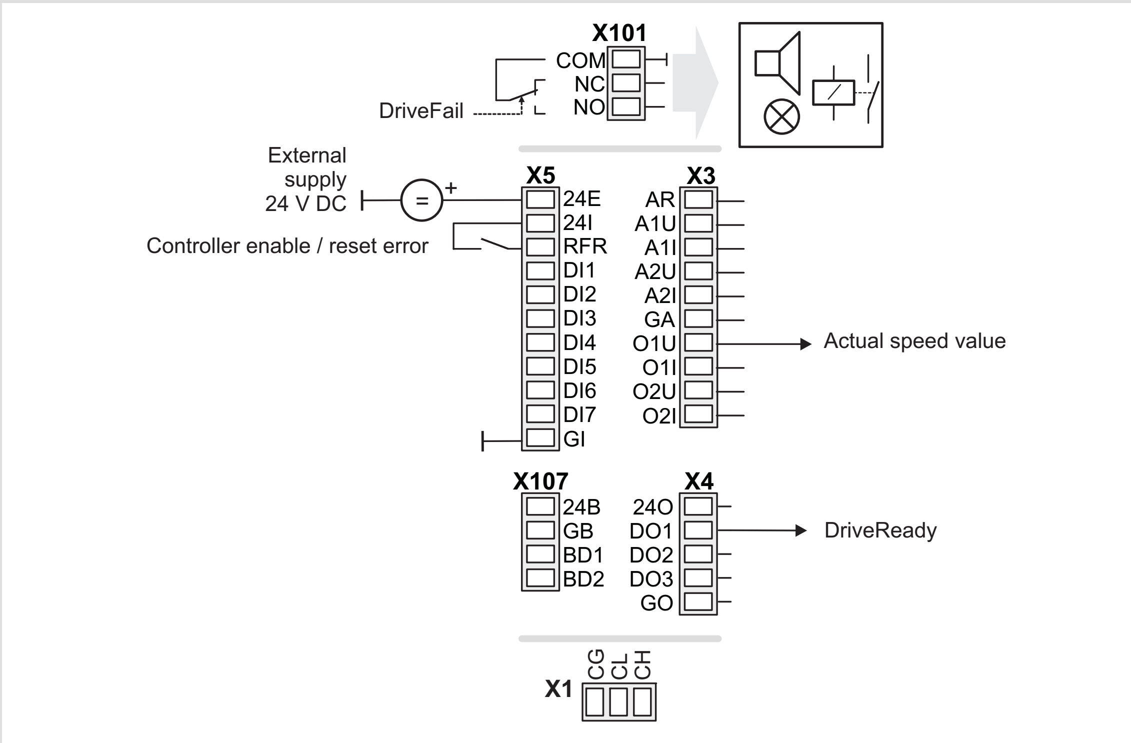
### 8.4.3.5 Keypad




| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_NCtrl.bDriveFail |
| | |
| X5/RFR | LA_NCtrl.bFailReset |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | - |
| X5/DI4 | - |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | - |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_NCtrl.nMainSetValue_a * |
| X3/A1I | - |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_NCtrl.nMotorSpeedAct_a * |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | * 10 V ≡ 100 % reference speed (C00011) |
| X4/DO1 | LA_NCtrl.bDriveReady |
| X4/DO2 | - |
| X4/DO3 | - |

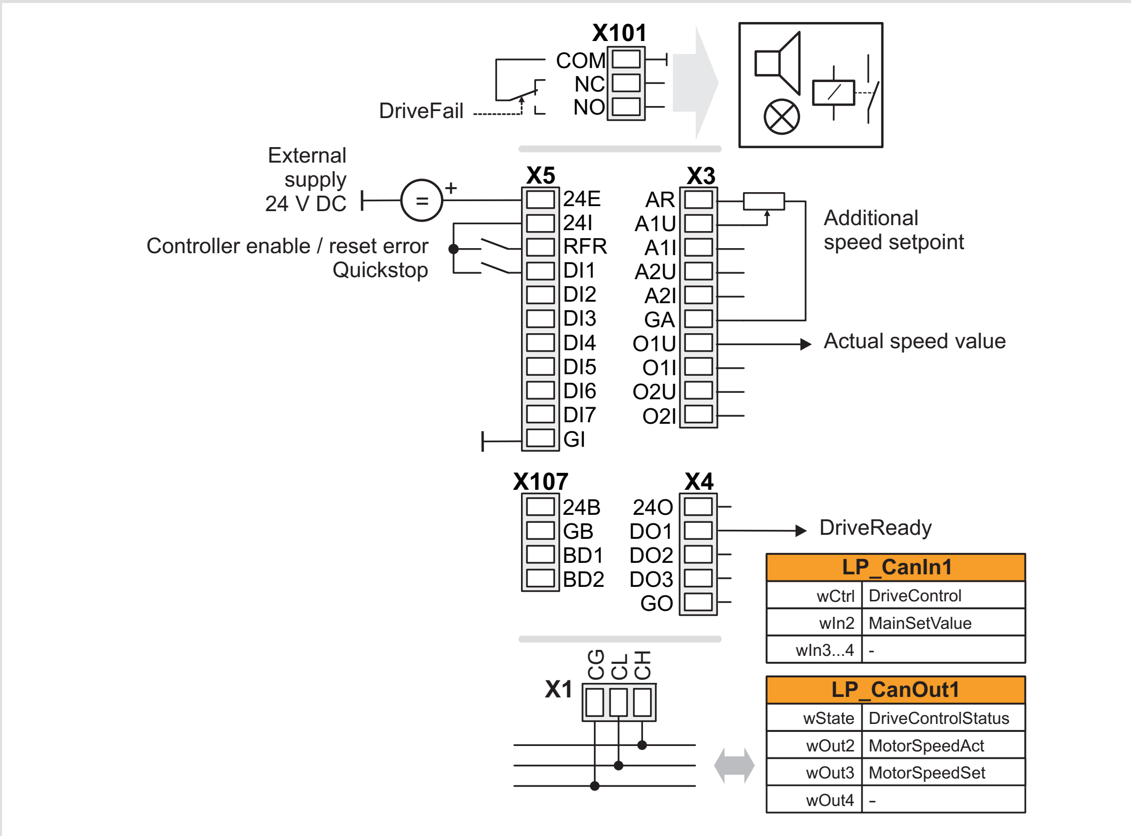
### 8.4.3.6 PC




| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_NCtrl.bDriveFail |
| | |
| X5/RFR | LA_NCtrl.bFailReset |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | - |
| X5/DI4 | - |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | - |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | - |
| X3/A1I | - |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_NCtrl.nMotorSpeedAct_a * |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | * 10 V ≡ 100 % reference speed (C00011) |
| X4/DO1 | LA_NCtrl.bDriveReady |
| X4/DO2 | - |
| X4/DO3 | - |

### 8.4.3.7 CAN




| Connection | Assignment |
|---|---|
| **X101/NC-NO** | LA_NCtrl.bDriveFail |
| | |
| **X5/RFR** | LA_NCtrl.bFailReset |
| **X5/DI1** | LA_NCtrl.bSetQuickStop |
| **X5/DI2** | - |
| **X5/DI3** | - |
| **X5/DI4** | - |
| **X5/DI5** | - |
| **X5/DI6** | - |
| **X5/DI7** | - |
| | |
| **X107/BD1** | - |
| **X107/BD2** | - |
| | |

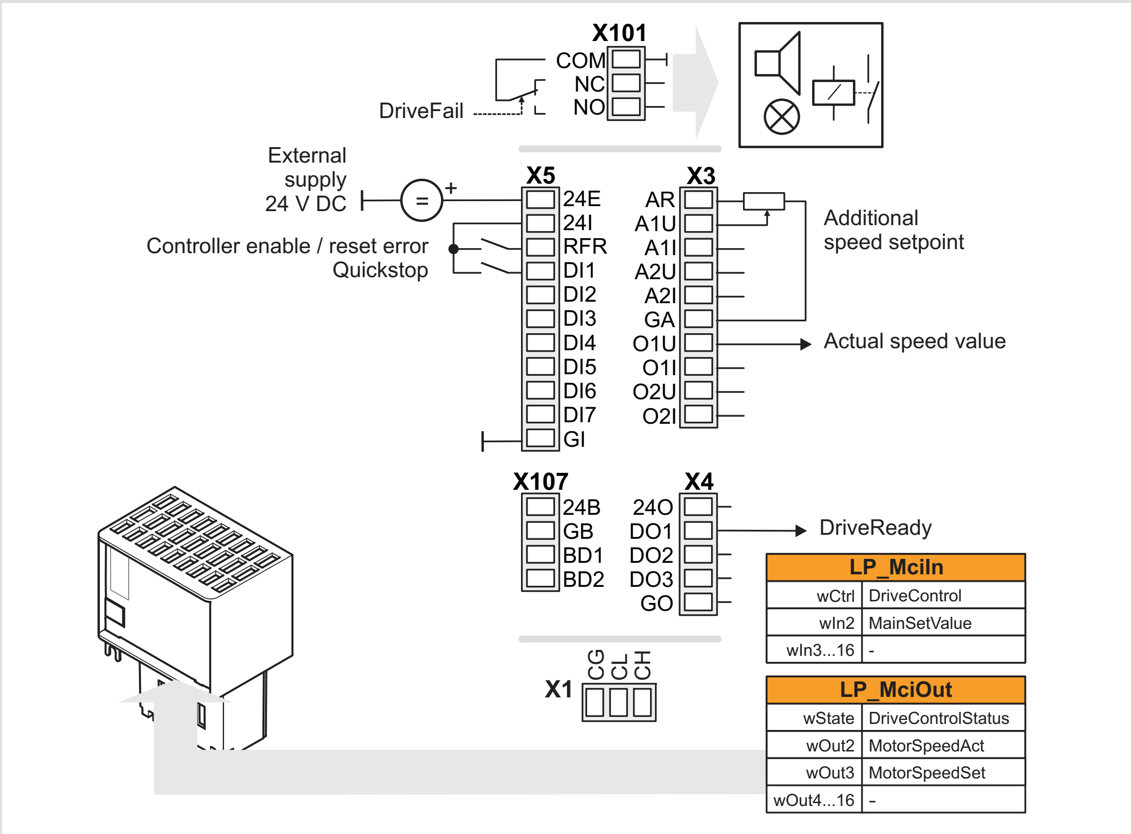
| Connection | Assignment |
|---|---|
| | |
| **X3/A1U** | LA_NCtrl.nAuxSetValue_a * |
| **X3/A1I** | - |
| **X3/A2U** | - |
| **X3/A2I** | - |
| **X3/O1U** | LA_NCtrl.NMotorSpeedAct_a * |
| **X3/O1I** | - |
| **X3/O2U** | - |
| **X3/O2I** | - |
| | * 10 V ≡ 100 % reference speed (C00011) |
| **X4/DO1** | LA_NCtrl.bDriveReady |
| **X4/DO2** | - |
| **X4/DO3** | - |

▶ Process data assignment for fieldbus communication (📖 403)

**Note!**

- You must set the setpoint arithmetic in C00190 to "1: NOut = NSet + NAdd" so that the additional speed setpoint selected via the analog input A1U has an additive effect.
- The "manual jog" function via digital terminals is being prepared!

### 8.4.3.8 MCI




| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_NCtrl.bDriveFail |
| | |
| X5/RFR | LA_NCtrl.bFailReset |
| X5/DI1 | LA_NCtrl.bSetQuickStop |
| X5/DI2 | - |
| X5/DI3 | - |
| X5/DI4 | - |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | - |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_NCtrl.nAuxSetValue_a * |
| X3/A1I | - |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_NCtrl.NMotorSpeedAct_a * |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | * 10 V ≡ 100 % reference speed (C00011) |
| X4/DO1 | LA_NCtrl.bDriveReady |
| X4/DO2 | - |
| X4/DO3 | - |

▸ Process data assignment for fieldbus communication (📖 403)

**Note!**

- You must set the setpoint arithmetic in C00190 to "1: NOut = NSet + NAdd" so that the additional speed setpoint selected via the analog input A1U has an additive effect.
- The "manual jog" function via digital terminals is being prepared!

### 8.4.4 Process data assignment for fieldbus communication

The fieldbus communication is connected (preconfigured) to the previously selected technology application by selecting the corresponding control mode in C00007:

- "30: CAN" for the connection to the system bus (CAN)
- "40: MCI" for the connection to a plugged-on communication module (e.g. PROFIBUS)

The assignment of the process data words does not depend on the applied bus system but exclusively on the application:

| Input words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControl | Control word<br>• See table below for bit assignment. |
| Word 2 | MainSetValue | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 3 | - | Not preconfigured |
| Word 4 | - | Not preconfigured |
| Word 5 ... 16 | - | Not preconfigured<br>• Only available for control mode "40: MCI". |

| Control word | Name | Function |
|---|---|---|
| Bit 0 | SwitchOn | 1 ≡ Change to the "SwitchedOn" device status<br>• This bit must be set in the CAN/MCI control word to ensure that the device changes to the "SwitchedOn" device status after mains connection without the need for a master control specifying this bit via fieldbus.<br>• If control via a bus system is not wanted (e.g. in the case of control via terminals), the *wDriveCtrl* output signal of the LS_ParFix system block can be connected to the control word inputs. |
| Bit 1 | DisableVoltage | 1 ≡ Inhibit inverter control (IMP - pulse inhibit) |
| Bit 2 | SetQuickStop | 1 ≡ Activate quick stop (QSP).<br>▸ Activate/Deactivate quick stop (📖 98) |
| Bit 3 | EnableOperation | 1 ≡ Enable controller (RFR)<br>• If control via terminals is performed, this bit must be set both in the CAN control word and in the MCI control word. Otherwise, the controller is inhibited.<br>▸ Enable/Inhibit controller (📖 97) |
| Bit 4 | ModeSpecific_1 | Reserved (currently not assigned) |
| Bit 5 | ModeSpecific_2 | |
| Bit 6 | ModeSpecific_3 | |
| Bit 7 | ResetFault | 1 ≡ Reset fault (trip reset)<br>• Acknowledge fault message (if the error cause has been eliminated).<br>▸ Reset error (📖 99) |
| Bit 8 | SetHalt | 1 ≡ Activate stop function<br>• Stop drive via stopping ramp (in preparation). |
| Bit 9 | reserved_1 | Reserved (currently not assigned) |
| Bit 10 | reserved_2 | |
| Bit 11 | SetDCBrake | 1 ≡ Activate DC-injection braking<br>▸ Manual DC-injection braking (DCB) (📖 251) |
| Bit 12 | JogSpeed1 | Activation of fixed speed 1 ... 3 |
| Bit 13 | JogSpeed2 | |

| Control word | Name | Function |
|---|---|---|
| Bit 14 | SetFail | 1 ≡ Set error (trip set) |
| Bit 15 | SetSpeedCcw | 0 ≡ Direction of rotation to the right (Cw)<br>1 ≡ Direction of rotation to the left (Ccw) |

| Output words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControlStatus | Status word<br>• See table below for bit assignment. |
| Word 2 | MotorSpeedAct | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 3 | MotorSpeedSet | Resulting overall setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 4 | - | Not preconfigured |
| Word 5 ... 16 | - | Not preconfigured<br>• Only available for control mode "40: MCI". |

| Status word | Name | Status |
|---|---|---|
| Bit 0 | FreeStatusBit0 | Free status bit 0 (not assigned, freely assignable) |
| Bit 1 | PowerDisabled | 1 ≡ Inverter control inhibited (pulse inhibit is active) |
| Bit 2 | FreeStatusBit2 | Free status bit 2 (not assigned, freely assignable) |
| Bit 3 | FreeStatusBit3 | Free status bit 3 (not assigned, freely assignable) |
| Bit 4 | FreeStatusBit4 | Free status bit 4 (not assigned, freely assignable) |
| Bit 5 | FreeStatusBit5 | Free status bit 5 (not assigned, freely assignable) |
| Bit 6 | ActSpeedIsZero | During open-loop operation:<br>1 ≡ Speed setpoint < Comparison value (C00024)<br>During closed-loop operation:<br>1 ≡ Actual speed value < Comparison value (C00024) |
| Bit 7 | ControllerInhibit | 1 ≡ Controller inhibited (controller inhibit is active) |
| Bit 8 | StatusCodeBit0 | Bit coded display of the active device status<br>▸ Device state machine and device statuses (see table [4-1]) |
| Bit 9 | StatusCodeBit1 | |
| Bit 10 | StatusCodeBit2 | |
| Bit 11 | StatusCodeBit3 | |
| Bit 12 | Warning | 1 ≡ A warning is indicated |
| Bit 13 | Trouble | 1 ≡ Controller is in the "Trouble" device status<br>• E.g. if an overvoltage has occurred. |
| Bit 14 | FreeStatusBit14 | Free status bit 14 (not assigned, freely assignable) |
| Bit 15 | FreeStatusBit15 | Free status bit 15 (not assigned, freely assignable) |

## 8.4.5 Setting parameters (short overview)

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00012 | Accel. time - main setpoint | 2.000 | s |
| C00013 | Decel. time - main setpoint | 2.000 | s |
| C00019 | Auto DCB: Threshold | 3 | rpm |
| C00024 | Comparison value N_Act | 0.00 | % |
| C00036 | DCB: Current | 50.00 | % |
| C00039/1 | Fixed setpoint 1 | 40.00 | % |
| C00039/2 | Fixed setpoint 2 | 60.00 | % |
| C00039/3 | Fixed setpoint 3 | 80.00 | % |
| C00039/4...15 | Fixed setpoint 4 ... 15 | 0.00 | % |
| C00101/1...15 | Add. accel. time 1 ... 15 | 0.000 | s |
| C00103/1...15 | Add. decel. time 1 ... 15 | 0.000 | s |
| C00105 | Decel. time - quick stop | 2.000 | s |
| C00106 | Auto DCB: Hold time | 0.500 | s |
| C00107 | DCB: Hold time | 999.000 | s |
| C00134 | Ramp rounding - main setpoint | 0: Off | |
| C00182 | S-ramp time PT1 | 20.00 | s |
| C00190 | Setpoint arithmetic | 0: NOut = NSet | |
| C00220 | Accel. time - add. setpoint | 0.000 | s |
| C00221 | Decel. time - add. setpoint | 0.000 | s |
| C00222 | L_PCTRL_1: Vp | 1.0 | |
| C00223 | L_PCTRL_1: Tn | 400 | ms |
| C00224 | L_PCTRL_1: Kd | 0.0 | |
| C00225 | L_PCTRL_1: MaxLimit | 199.99 | % |
| C00226 | L_PCTRL_1: MinLimit | -199.99 | % |
| C00227 | L_PCTRL_1: Accel. time | 0.010 | s |
| C00228 | L_PCTRL_1: Decel. time | 0.010 | s |
| C00233 | L_PCTRL_1: Root function | 0: Off | |
| C00241 | L_NSet_1: Hyst. NSet reached | 0.50 | % |
| C00242 | L_PCTRL_1: Operating mode | 0: Off | |
| C00243 | L_PCTRL_1: Accel. time influence | 5.000 | s |
| C00244 | L_PCTRL_1: Decel. time influence | 5.000 | s |
| C00632/1 | L_NSet_1: Blocking speed 1 max | 0.00 | % |
| C00632/2 | L_NSet_1: Blocking speed 2 max | 0.00 | % |
| C00632/3 | L_NSet_1: Blocking speed 3 max | 0.00 | % |
| C00633/1 | L_NSet_1: Blocking speed 1 min | 0.00 | % |
| C00633/2 | L_NSet_1: Blocking speed 2 min | 0.00 | % |
| C00633/3 | L_NSet_1: Blocking speed 3 min | 0.00 | % |
| C00635 | L_NSet_1: nMaxLimit | 199.99 | % |
| C00636 | L_NSet_1: nMinLimit | -199.99 | % |
| C00670 | L_OffsetGainP_1: Gain | 1.0000 | |
| C00671 | L_OffsetGainP_2: Gain | 1.0000 | |
| C00672 | L_OffsetGainP_3: Gain | 1.0000 | |

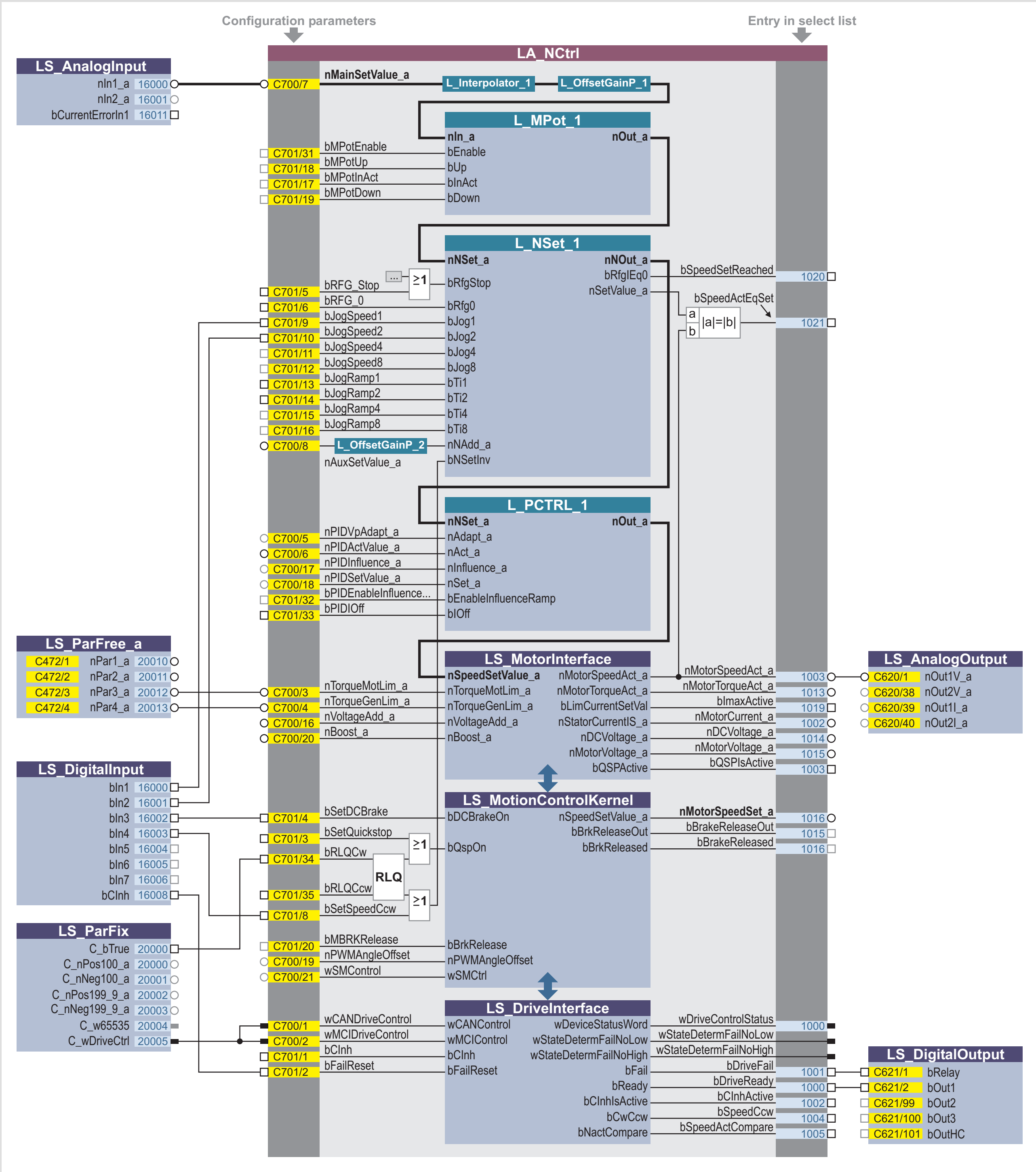
| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00696 | L_OffsetGainP_1: Offset | 0.00 | % |
| C00697 | L_OffsetGainP_2: Offset | 0.00 | % |
| C00698 | L_OffsetGainP_3: Offset | 0.00 | % |
| C00800 | L_MPot_1: Upper limit | 100.00 | % |
| C00801 | L_MPot_1: Lower limit | -100.00 | % |
| C00802 | L_MPot_1: Accel. time | 10.0 | s |
| C00803 | L_MPot_1: Decel. time | 10.0 | s |
| C00804 | L_MPot_1: Inactive fct. | 0: Retain value | |
| C00805 | L_MPot_1: Init fct. | 0: Load last value | |
| C00806 | L_MPot_1: Use | 0: No | |

**Related topics:**

▸ "GeneralPurpose" functions (📖 467)

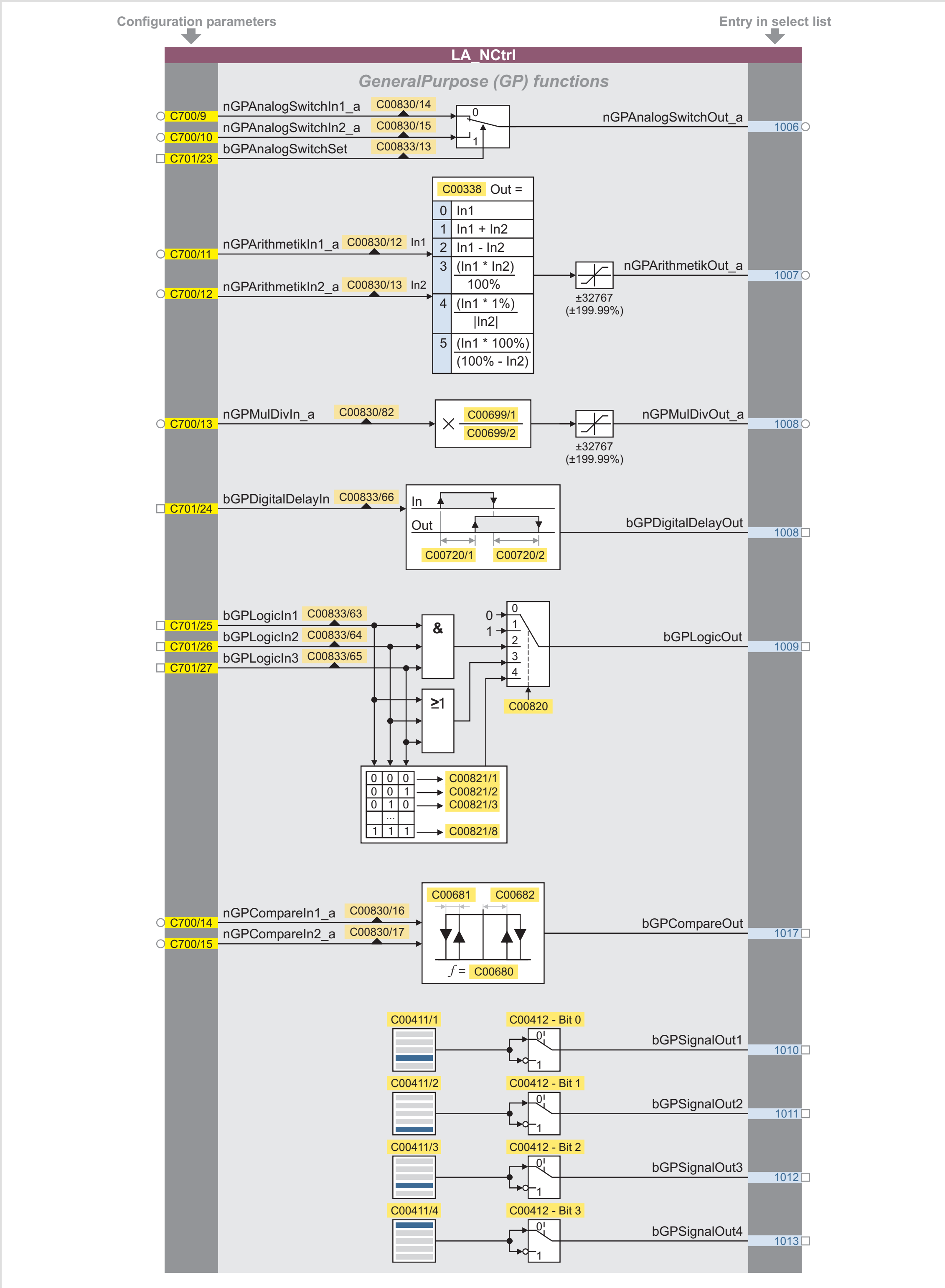
## 8.4.6 Configuration parameters

If required, the subcodes of C00700 and C00701 serve to change the pre-configured assignment of the application inputs:

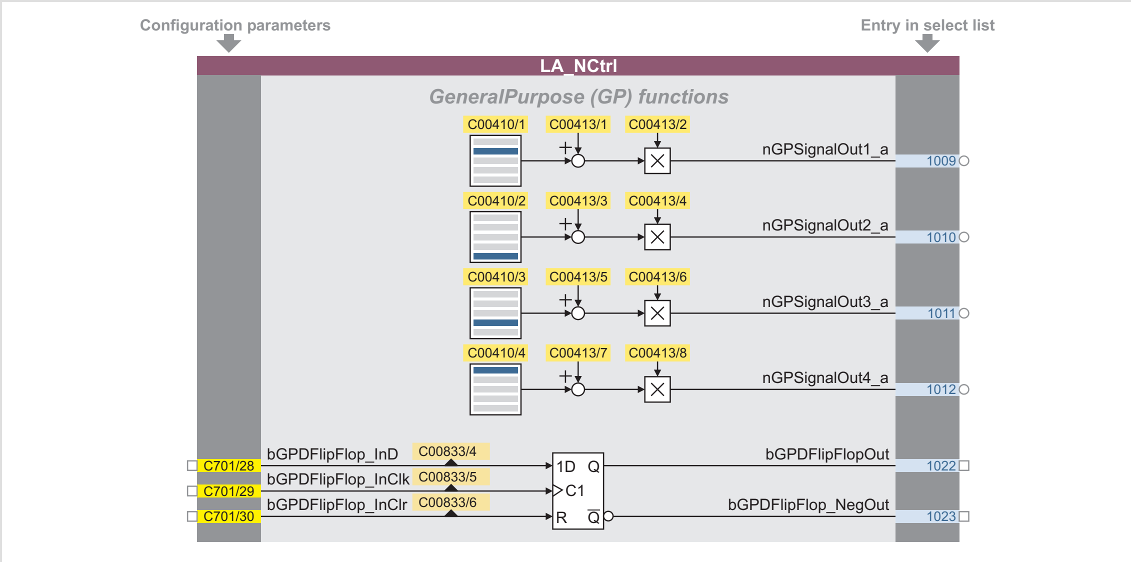



[8-1] Pre-assignment of the "Actuating drive speed" application in the "Terminals 0" control mode

## Configuration parameters for "GeneralPurpose" functions



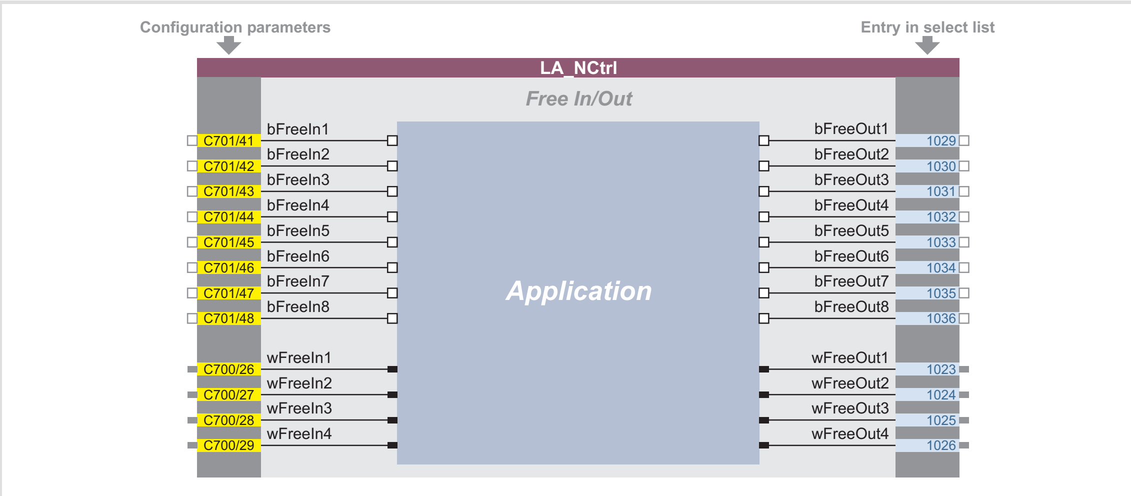

[8-2] "GeneralPurpose" functions




[8-3] "GeneralPurpose" functions (continuation)

## Free inputs and outputs

These inputs can be freely interconnected in the application level. They can be used to transfer signals from the I/O level to the application level and vice versa.




[8-4] Free inputs/outputs

## Related topics:

▶ User-defined terminal assignment (📖 370)

▶ "GeneralPurpose" functions (📖 467)

## 8.5 TA "Table positioning"

Numerous functions for the motion control of a single axis can be executed with the "Table positioning" technology application. For this, the technology application accesses the basic functions implemented in the **Motion Control Kernel** which are described in detail in the main chapter "Basic drive functions (MCK)". (📖 473)

- Manual jog
- Retracting from limit switches
- Reference setting/homing in 18 different modes (*in preparation*)
- Positioning (absolute, relative)
- Continuous travel (speed mode)
- Speed override
- Acceleration override
- Profile sequence block control
- Travel block restart within the profile (online profile start)
- Specification of target position via process signal interface (e.g. via fieldbus)

### Further functions

- Up to 15 travel sets can be set via parameters
- Parameterisation of the profile data in physical units or relative values
- Adjustable torque limitation
- Freely selectable, variable ramp shape
- Motor potentiometer function
- Software limit position monitoring
- Following error monitoring (with static limits)
- Automatic holding brake control
- Quick stop (QSP) with adjustable ramp time
- Enable of individual functions via control word
- Status and diagnostic displays
- Operating mode changeover (manual jog, homing, speed follower, positioning)
- Integrated disposable "GeneralPurpose" functions:
  Analog switch , arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-FlipFlop, counter

**Related topics:**

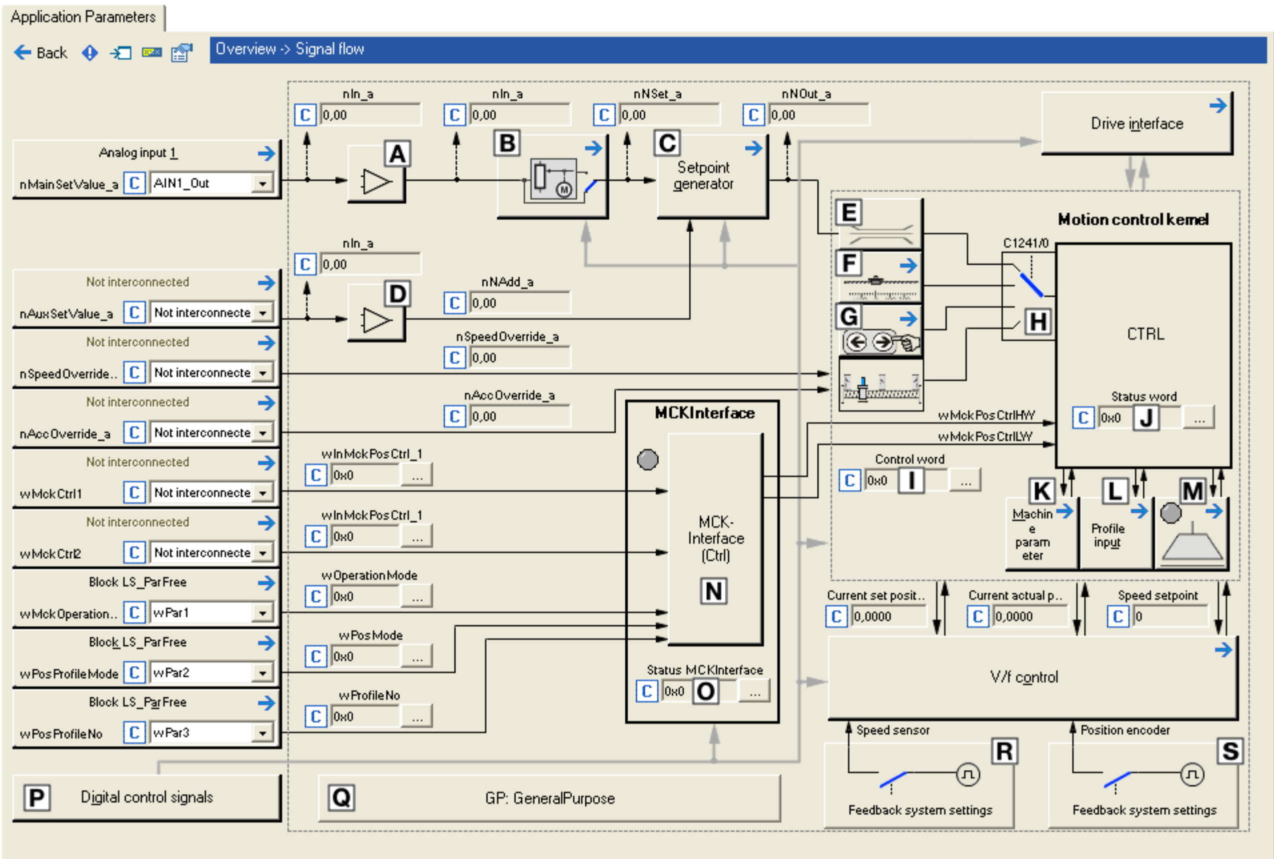
▶ Commissioning of the "Table positioning" technology application (📖 57)


Lenze

### 8.5.1 Basic signal flow

The main element of the technology application is the **Motion Control Kernel**.

- The **Motion Control Kernel** is controlled via a 32-bit double control word. All motion profiles in the different operating modes can be controlled with this interface.
- Upstream to **the Motion Control Kernel** there is a so-called MCKInterface called which provides additional process inputs for the application in order to control the **Motion Control Kernel**.




[8-5] Signal flow of the table positioning

- A Offset and gain (L_OffsetGainP_1)
- B Motor potentiometer function (L_MPot_1)
- C Setpoint generator (L_NSet_1)
- D Offset and gain (L_OffsetGainP_2)
- E Speed setpoint input limitation
- F Basic function "Homing"
- G Basic function "Manual jog"
- H Operating mode changeover (is executed by the MCKInterface)
- I MCK control word
- J MCK status word
- K Selection of the machine parameters
- L Profile entry for the "Positioning" basic function.
- M Holding brake control
- N MCKInterface
- O Status word of the MCKInterface
- P Terminal assignment & display of digital control signals
- Q Integrated disposable "GeneralPurpose" functions: Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop

### 8.5.1.1 Possibilities for the position selection

The following options are available for the selection of different positions:

1. Changeover of profiles
2. Changing the profile position via parameter data (SDO)
3. Changing the profile position via process data (PDO)
   – The profile position is defined in [increments] or in the application unit [units].

| | Changeover of profiles | Changing the profile position via SDO | Changing the profile position via PDO | |
|---|---|---|---|---|
| **Number of positions** | max. 15 | n > 15 | n > 15 | |
| **Position selection** | Parameterisation in the profile | Selection via SDO | Selection via PDO | |
| **Unit** | units | units | Increments | units |
| **Remainder allowance** | in the drive | in the drive | in the PLC | in the drive |

**Note!**

Setting the machine parameters is a basic prerequisite for the operating modes "Homing", "Manual jog" and "Positioning".

The more precisely the machine parameters are set, the better the results of positioning!

▶ Machine parameters (📖 498)

**Possibility 1: Changeover of profiles**

A total of 15 profiles is available in the 8400 TopLine.

- A profile describes a motion task that can be converted into a rotary motion by the **Motion Control Kernel** in the "Positioning" operating mode.
- A detailed explanation of all profile parameters can be found in the subchapter "Profile entry". (📖 545) of the description of the "Positioning" basic drive function.
- The selection of the profile to be executed can either be executed as a data word via the input *wPosProfileNo* or binary coded via the inputs *bPosProfileNo_1* ... *bPosProfileNo_4*.
- The chosen profile is started via the process input *bPosExecute* of the FB L_MckCtrlInterface or the control bit "PosExecute" in the MCK control word.

**Possibility 2: Changing the profile position via parameter data (SDO)**

The parameter data channel of a fieldbus can also be used to change the position in a profile.

- The position of the profiles 1 ... 15 are stored in the subcodes 1 ... 15 of the code C01301.
- More than 15 positions are available via the profile changeover (as described under possibility 1).

**Possibility 3: Changing the profile position via process data (PDO)**

For selecting the position in [increments], the *dnPosProfilePosition_p* input at the application block is available.

**Note!**

For selecting the position in [units], the two inputs *wPosProfileUnitsLW* and *wPosProfileUnitsHW* at the application block are available.

- In the application level, the position in the FB L_MckCtrlInterface is converted from [units] to [increments] again and output to *dnPosSetOut_p*. For this purpose, the respective mode for position calculation must be selected in C01296/1.
- Since the conversion is not executed within one controller cycle, the data at the *dnPosSetOut_p* output can be out-of-date. The *bPosSetDataValid* output is set to TRUE when the conversion is completed and the position in increments is valid. Then, the position can be transmitted to the profile (see the following section).

The acceptance of the position into the currently selected profile is executed with a FALSE-TRUE edge at the *bPosSetProfilePosition* input.

- If bit 2 is set in C01297, the applied setpoint position is accepted into the currently selected profile and the profile is started directly when setting the process input *bPosExecute* to TRUE or setting the control bit "PosExecute" in the MCK control word.
- If bit 3 is set in C01297, the setpoint positions at the MCKInterface are automatically accepted into the profile with the applied profile number if a change of data is detected at the corresponding input for the setpoint position.
- The accepted position is stored in the code C01301/x in the application unit [units]. By reading out the code, you can check if the position was correctly accepted into the profile.

**Note!**

How to store changed profile parameters safely against mains failure in the memory module: Set C00002/11 = "1: On / Start".

## 8.5.2 Internal interfaces | Application block "LA_TabPos"

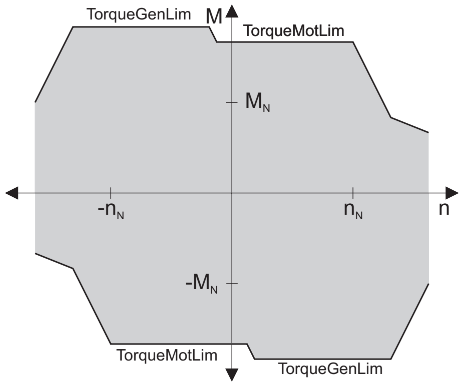
**Note!**

The connectors grayed out in the following table are hidden in the function block editor in the Lenze setting.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

### Inputs

| Identifier | Data type | | Information/possible settings |
|---|---|---|---|
| wCANDriveControl | WORD | | Control word via system bus (CAN) to device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. |
| wMCIDriveControl | WORD | | Control word via communication module (e.g. PROFIBUS) to device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. |
| wSMControl | WORD | | Interface to the optional safety system.<br>• Setting control bit 0 ("SafeStop1") in this control word causes e.g. the automatic deceleration of the drive to standstill within this application (in the **Motion Control Kernel**).<br>• See the "Interface to safety system" subchapter of the chapter on basic drive functions for a detailed description of the control bits. |
| bCInh | BOOL | | Enable/Inhibit controller |
| | | FALSE | Enable controller: The controller switches to the "OperationEnabled" device status if no other source of a controller inhibit is active.<br>• C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit. |
| | | TRUE | Inhibit controller (controller inhibit): The controller switches to the "SwitchedOn" device status. |
| bFailReset | BOOL | | Reset error messages<br>In the Lenze setting this input is connected to the digital input controller enable so that a possibly existing error message is reset together with the controller enable (if the cause for the fault is eliminated). |
| | | TRUE | The current fault is reset, if the cause for the fault is eliminated.<br>• If the fault still exists, the error status remains unchanged. |
| bSetQuickstop | BOOL | | Enable quick stop (QSP)<br>• Also see device command "Activate/Deactivate quick stop". |
| | | TRUE | Activate quick stop<br>• Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0).<br>• The motor is kept at a standstill during closed-loop operation.<br>• A pulse inhibit (CINH) is set if the auto DCB function has been activated via C00019. |
| | | FALSE | Deactivate quick stop<br>• The quick stop is deactivated if no other source for the quick stop is active.<br>• C00159 provides a bit-coded representation of active sources/causes for the quick stop. |

| Identifier (Data type) | Information/possible settings |
|---|---|
| nTorqueMotLim_a<br>nTorqueGenLim_a<br>INT | Torque limitation in motor mode and in generator mode<br>• These input signals are directly transferred to the motor control to limit the controller's maximum torque in motor and generator mode.<br>• The drive cannot output a higher torque in motor/generator mode than set here.<br>• The applied values (any polarity) are internally interpreted as absolute values.<br>• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.<br>• If sensorless vector control (SLVC) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057)<br>**Note:**<br>Setting this input is ineffective in the reference modes 14 and 15 ("Homing to positive stop"). |
| | Torque limits in motor and generator mode:<br> |
| bSetSpeedCcw<br>BOOL | Change of direction of rotation<br>• For instance if a motor or gearbox is fixed laterally reversed to a machine part, but the setpoint selection should still be executed for the positive direction of rotation.<br>FALSE: Direction of rotation to the right (Cw)<br>TRUE: Direction of rotation to the left (Ccw) |
| nMainSetValue_a<br>INT | Main speed setpoint<br>• Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The main setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.<br>• Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.<br>• For a detailed functional description see the L_NSet FB. |
| nAuxSetValue_a<br>INT | Additional speed setpoint<br>• Offset and gain of this input signal can be set in C00697 and C00671 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The additional speed setpoint can be linked arithmetically with the main speed setpoint behind the ramp function generator.<br>• The additional speed setpoint can be shown via ramp times of a second ramp function generator.<br>• For a detailed functional description see the L_NSet FB. |
| bJogSpeed1<br>bJogSpeed2<br>BOOL | Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint<br>• Selection inputs are binary coded. |

| Identifier | Data type | Value | Information/possible settings |
|---|---|---|---|
| bJogSpeed1<br>bJogSpeed2 | BOOL | | Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint<br>• A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via these selection inputs.<br>• The two selection inputs are binary coded, therefore you can select 3 fixed setpoints.<br>• In the case of binary coded selection "0" (all inputs = FALSE or not assigned), main setpoint *nMainSetValue_a* is active.<br>• The selection of the fixed setpoints is carried out in C00039/1...3 in [%] based on the reference speed (C00011).<br>• For a detailed functional description see the L_NSet FB. |
| **Motor potentiometer**<br>Alternatively to the input signal *nMainSetValue_a*, the main setpoint can also be generated by a motor potentiometer function.<br>• In the Lenze setting, the motor potentiometer function is deactivated.<br>• Activation is possible via C00806 or via the *bMPotEnable* input.<br>• The behaviour of the motor potentiometer during switch-on of the drive system can be selected in C00805.<br>• For a detailed functional description see the L_MPot FB. | | | |
| bMPotEnable | BOOL | | Activating the motor potentiometer function<br>• This input and C00806 are OR'd. |
| | | TRUE | The motor potentiometer function is active; the speed setpoint can be changed via the *bMPotUp* and *bMPotDown* control inputs. |
| bMPotUp | BOOL | | Increasing the speed setpoint |
| | | TRUE | Approach the upper speed limit value set in C00800 with the acceleration time set in C00802. |
| bMPotDown | BOOL | | Decreasing the speed setpoint |
| | | TRUE | Approach the lower speed limit value set In C00801 with the deceleration time set in C00803. |
| **MCK basic functions** | | | |
| bMBrakeRelease | BOOL | | Holding brake control: Release/apply brake<br>• In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). |
| | | FALSE | Apply brake.<br>• During automatic operation, the internal brake logic controls of the brake. |
| | | TRUE | Release brake manually (forced release).<br>• **Note!** The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If the brake control has inhibited the controller, this inhibit is deactivated again.<br>• In semi-automatic operation, the brake is released including feedforward control. |
| wMckCtrl1<br>wMckCtrl2 | WORD | | Direct selection of MCK control words<br>• E.g. via a master control which has been connected to the fieldbus, too. For this purpose, the control word inputs can directly be connected to the **LP_MciIn** or **LP_CanIn** fieldbus interface.<br>• See the "MCK control word" chapter for a detailed description of the individual control bits. |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| wMckOperationMode<br>WORD | Selection of the operating mode of the **Motion Control Kernel**<br>• Only bit 0 ... bit 3 of *wMckOperationMode* is evaluated.<br>• If an invalid operating mode is selected, the response set in C00595/11 is activated (Lenze setting: "Warning").<br>• The current operating mode is displayed in C01243. | |
| | 0 | Speed follower |
| | 1 | Homing |
| | 2 | Manual jog |
| | 3 | Positioning |
| | 4 | Stop |
| | 5 | Position follower |
| | 6 ... 15 | Reserved for future extensions |
| bMckOperationMode_1<br>...<br>bMckOperationMode_8<br>BOOLBOOL | Binary-coded selection of the operating mode of the **Motion Control Kernel**<br>• See the "MCK control word" chapter for a detailed description of the individual control bits.<br>• If an invalid operating mode is selected, the response set in C00595/11 is activated (Lenze setting: "Warning").<br>• The current operating mode is displayed in C01243. | |
| bPosCtrlOn<br>BOOL | Position control/Angle control<br>• Pre-configured assignment: TRUE (position control active) | |
| | FALSE | Deactivate position control/angle control |
| | TRUE | Activate position control/angle control |
| nPosCtrlOutLimit_a<br>INT | Limitation of the position controller output<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• Pre-configured assignment: 100 % | |
| nPosCtrlPAdapt_a<br>INT | Adaptation of the position controller gain<br>• Scaling: 16384 ≡ 100 % Vp (C00254)<br>• Pre-configured assignment: 100 % | |
| bLimitSwitchPos<br>bLimitSwitchNeg<br>BOOL | Limit position monitoring: Inputs for positive/negative limit switch | |
| bReleaseLimitSwitch<br>BOOL | Manual jog: Retract operated limit switch | |
| | TRUE | Retract operated limit switch (in opposite direction) |
| bManJogPos<br>bManJogNeg<br>BOOL | Manual jog:<br>*bManJogPos* = TRUE: Manual jog right<br>*bManJogNeg* = TRUE: Manual jog left<br>Both inputs = TRUE: No change compared to previous state<br>Both inputs = FALSE: Stop manual jog | |
| bManJogExecute2ndVel<br>BOOL | Manual jog: Changeover to speed 2 | |
| | FALSE | Speed 1 (C01231/1) active |
| | TRUE | Speed 2 (C01231/2) active |
| bEnableSpeedOverride<br>BOOL | Speed override | |
| | TRUE | Activate speed override |
| nSpeedOverride_a<br>INT | Value for Speed override<br>• Percentage multiplier for the currently active speed.<br>• 16384 ≡ 100 % of the maximum traversing speed (display in C01211/1).<br>• Values > 16384 are ignored.<br>• If the override value is 0 %, the drive is brought to a standstill. | |
| bEnableAccOverride<br>BOOL | Acceleration override | |
| | TRUE | Activate acceleration override |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nAccOverride_a<br>INT | Value for Acceleration override<br>• Percentage multiplier for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised acceleration of the corresponding operating mode.<br>• Values > 16384 are ignored.<br>• If the override value is 0 %, acceleration ceases. | |
| bHomeStartStop<br>BOOL | Homing: Start/stop homing | |
| | TRUE | Start homing |
| | TRUE↘FALSE | Stop homing |
| bHomeSetPosition<br>BOOL | Homing: Set home position | |
| | TRUE | Set home position |
| bHomeResetPosition<br>BOOL | Homing: Reset home position | |
| | TRUE | Reset the "Reference known" status |
| bHomeMark<br>BOOL | Homing: Input for reference switch (pre-switch off mark)<br>• This input responds to the FALSE status (fail-safe) and is to be connected to the corresponding digital input to which the reference switch is connected. | |
| bPosSetProfilePosition<br>BOOL | Position teaching: MCK setpoint position | |
| | TRUE | Teach MCK setpoint position into the selected profile. |
| bPosSetActualPosition<br>BOOL | Position teaching: Current position | |
| | TRUE | Teach current position into the selected profile. |
| bPosExecute<br>BOOL | Positioning: Start travelling | |
| | FALSE↗TRUE | Execute selected profile |
| bPosFinishTarget<br>BOOL | Positioning: Complete cancelled profile | |
| | FALSE↗TRUE | A positioning process previously cancelled, e.g. by *bPosStop* or due to a device error, is resumed by travelling to the original target. |
| bPosDisableFollowProfile<br>BOOL | Positioning: Do not execute sequence profile (switch-off profile linkage) | |
| | TRUE | Evaluation of the sequence profile number parameterised in C01307/1...15 for the selected profile is suppressed. |
| bPosStop<br>BOOL | Positioning: Cancel travelling | |
| | TRUE | Stop positioning<br>From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked). |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| wPosProfileMode<br>WORD | Override of the positioning mode set in the profile data<br>• Via this input, an override of the positioning mode parameterised in C01300/1...15 for the selected profile is possible.<br>• The value set in C01300/1...15 is not overwritten in this case.<br>• Only bit 0 ... bit 3 of *wPosMode* are evaluated.<br>• If *wPosMode* = 0 is selected, the positioning mode set in C01300/1...15 is used. | |
| | 0 | Positioning mode = setting in C01300/1...15 |
| | 1 | Absolute (shortest path) |
| | 2 | Continuous |
| | 3 | Relative |
| | 4 | Absolute (Cw) |
| | 5 | Absolute (Ccw) |
| | 8 | Absolute (shortest path) to TP |
| | 9 | Continuous to TP |
| | 10 | Relative to TP |
| | 11 | Absolute (Cw) on TP |
| | 12 | Absolute (Ccw) on TP |
| | All other possible settings are reserved for future extensions! | |
| wPosProfileNo<br>WORD | Stipulation of the profile to be executed<br>• Optionally as a data word or binary coded. | |
| bPosProfileNo_1<br>...<br>bPosProfileNo_8<br>BOOL | | |
| dnPosProfilePosition_p<br>DINT | Selection of the target position in [increments]<br>• The mode for calculating the position is selected in C01296/1. | |
| wPosProfileUnitsLW<br>wPosProfileUnitsHW<br>WORD | Selection of the target position in [units]<br>• *wPosProfileUnitsLW* = Low word, *wPosProfileUnitsHW* = High word<br>• The mode for calculating the position is selected in C01296/1. | |
| **GP: GeneralPurpose**<br>The following inputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | | |
| nGPAnalogSwitchIn1_a<br>nGPAnalogSwitchIn2_a<br>INT | Analog switch: Input signals<br>• The input signal selected via the selection input *bGPAnalogSwitchSet* is output at output *nGPAnalogSwitchOut_a*. | |
| bGPAnalogSwitchSet<br>BOOL | Analog switch: Selection input | |
| | FALSE | *nGPAnalogSwitchOut_a* = *nGPAnalogSwitchIn1_a* |
| | TRUE | *nGPAnalogSwitchOut_a* = *nGPAnalogSwitchIn2_a* |
| nGPArithmetikIn1_a<br>nGPArithmetikIn2_a<br>INT | Arithmetic: Input signals<br>• The arithmetic function is selected in C00338.<br>• The result is output at output *nGPArithmetikOut_a*. | |
| nGPMulDivIn_a<br>INT | Multiplication/Division: Input signal<br>• The factor for the multiplication can be set in C00699/1 (numerator) and C00699/2 (denominator).<br>• The result is output at output *nGPMulDivOut_a*. | |
| bGPDigitalDelayIn<br>BOOL | Binary delay element: Input signal<br>• The on-delay can be set in C00720/1.<br>• The off-delay can be set in C00720/2.<br>• The time-delayed input signal is output at output *bGPDigitalDelayOut*. | |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bGPLogicIn1<br>bGPLogicIn2<br>bGPLogicIn3<br>BOOL | Binary logic: Input signals<br>• The logic operation is selected in C00820.<br>• The result is output at output *bGPLogicOut*. | |
| nGPCompareIn1_a<br>nGPCompareIn2_a<br>INT | Analog comparison: Input signals<br>• The comparison operation is selected in C00680.<br>• Hysteresis and window size can be set in C00680 and C00682.<br>• If the comparison statement is true, the output *bGPCompareOut* will be set to TRUE. | |
| bGPDFlipFlop_InD<br>bGPDFlipFlop_InClk<br>bGPDFlipFlop_InClr<br>BOOL | D-FlipFlop: Input signals<br>• Data, clock and reset input | |
| bGPCounter1ClkUp<br>BOOL | Numerator: Clock input<br>• With each edge, the module counts up by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. | |
| bGPCounter1ClkDown<br>BOOL | Numerator: Clock input<br>• With each edge, the module counts down by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. | |
| bGPCounter1Load<br>BOOL | Numerator: Loading input<br>• The input has the highest priority. | |
| | TRUE | Accept the starting value *wGPCounter1LdVal*. |
| wGPCounter1LdVal<br>WORD | Numerator: Starting value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. | |
| wGPCounter1CmpVal<br>WORD | Numerator: Comparison value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. | |
| **Free inputs**<br>The following inputs can freely be interconnected on the application level.<br>The signals can be transferred from the I/O level to the application level via these inputs. | | |
| bFreeIn1 ... bFreeIn8<br>BOOL | Free inputs for digital signals | |
| wFreeIn1 ... wFreeIn4<br>WORD | Free inputs for 16-bit signals | |
| dnFreeIn1_p ... dnFreeIn2_p<br>DINT | Free inputs for 32-bit signals | |

## Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| wDriveControlStatus<br>WORD | Status word of the controller (based on DSP-402)<br>• The status word contains information on the currents status of the drive controller.<br>• See the "wDeviceStatusWord status word" subchapter of the chapter on device control for a detailed description of the bit assignment. | |
| wStateDetermFailNoLow<br>WORD | Display of the status determining error (LOW word) | |
| wStateDetermFailNoHigh<br>WORD | Display of the status determining error (HIGH word) | |
| bDriveFail<br>BOOL | TRUE | Drive controller in error status.<br>• "Fault" device status is active. |

| Identifier (Data type) | Value/meaning | |
|---|---|---|
| bDriveReady (BOOL) | TRUE | Controller is ready for operation.<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the controller is still inhibited by the user (controller inhibit). |
| bCInhActive (BOOL) | TRUE | Controller inhibit is active. |
| bQSPIsActive (BOOL) | TRUE | Quick stop is active. |
| bSpeedCcw (BOOL) | Current direction of rotation | |
| | FALSE | Direction of rotation to the right (Cw) |
| | TRUE | Direction of rotation to the left (Ccw) |
| bSpeedActCompare (BOOL) | Result of the speed comparison | |
| | TRUE | During open-loop operation:<br>Speed setpoint < Comparison value (C00024) |
| | | During closed-loop operation:<br>Actual speed value < Comparison value (C00024) |
| bImaxActive (BOOL) | "Current setpoint inside the limitation" status signal | |
| | TRUE | The current setpoint is internally limited (the drive controller operates at the maximum current limit). |
| bSpeedSetReached (BOOL) | Status signal "setpoint = 0" | |
| | TRUE | Speed setpoint from the ramp function generator = 0 |
| nMotorCurrent_a (INT) | Current stator current/effective motor current<br>• Scaling: 16384 ≡ 100 % $I_{max\_mot}$ (C00022) | |
| nMotorSpeedSet_a (INT) | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorSpeedAct_a (INT) | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorTorqueAct_a (INT) | Actual torque<br>• In the "VFC (+encoder)" operating mode of the motor control, this value is determined from the current motor current and corresponds to the actual torque only by approximation.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| nDCVoltage_a (INT) | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V | |
| nMotorVoltage_a (INT) | Current motor voltage/inverter output voltage<br>• Scaling: 16384 ≡ 1000 V | |
| **MCK basic functions** | | |
| bMBrakeReleaseOut (BOOL) | Holding brake control: Trigger signal for the holding brake control switching element via a digital output<br>• Use bit 0 in C02582 to activate inverted switching element triggering. | |
| | FALSE | Apply brake. |
| | TRUE | Release brake. |
| bMBrakeReleased (BOOL) | Holding brake control: "Brake released" considering the brake release time<br>• When the holding brake is triggered to close, *bMBrakeReleased* is immediately set to FALSE even if the brake closing time has not yet elapsed! | |
| | TRUE | Brake released (when the brake release time has elapsed). |
| wMckState1<br>wMckState2 (WORD) | Output of the MCK status words<br>• For a detailed description of each status bit see chapter "MCK status word". | |
| wMckActOperationMode (WORD) | Active setpoint-generating state of the MCK.<br>• Bit B0 ... B3 contain the information of the MCK status word.<br>• Bits B4 ... B15 are fixed at "0". | |

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bHomeDone<br>BOOL | TRUE | Homing has been executed. |
| bHomePosAvailable<br>BOOL | TRUE | Home position is known. |
| bProfileDone<br>BOOL | TRUE | Target position from the profile has been approached. |
| bProfileBusy<br>BOOL | TRUE | Profile positioning is active. |
| bAccelerating<br>BOOL | TRUE | Acceleration phase active. |
| bConstantDuty<br>BOOL | TRUE | Constant phase active. |
| bDecelerating<br>BOOL | TRUE | Braking phase active. |
| bDwellTime<br>BOOL | TRUE | Settling in target position is active |
| bInTarget<br>BOOL | TRUE | Target position (actual value) is in the target window. |
| wActProfileNo<br>WORD | Current profile number<br>• Bit B0 ... B7 contain the information of the MCK status word.<br>• Bits B8 ... B15 are fixed at "0". | |
| wActPosMode<br>WORD | Current positioning mode<br>• Bit B0 ... B3 contain the information of the MCK status word.<br>• Bits B4 ... B15 are fixed at "0". | |
| dnTargetPos_p<br>DINT | Target position in [increments]<br>• 65535 ≡ 1 revolution of the motor shaft | |
| dnSetPos_p<br>DINT | Absolute position setpoint | |
| dnPosAct_p<br>DINT | Current position of the motor shaft in [increments] | |
| dnDeltaPosAct_p<br>DINT | Current following error in [increments]<br>• Following error = Difference between set position and actual position | |
| wPosOutUnitsLW<br>wPosOutUnitsHW<br>WORD | Output of the target position in [units]<br>• *wPosOutUnitsLW* = Low-Word, *wPosOutUnitsHW* = High-Word<br>• The mode for calculating the position is selected in C01296/2. | |
| **GP: GeneralPurpose**<br>The following outputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▸ "GeneralPurpose" functions | | |
| nGPAnalogSwitchInOut_a<br>INT | Analog switch: Output signal | |
| nGPArithmetikOut_a<br>INT | Arithmetic: Output signal | |
| nGPMulDivOut_a<br>INT | Multiplication/Division: Output signal | |
| bGPDigitalDelayOut<br>BOOL | Binary delay element: Output signal | |
| bGPLogicOut<br>BOOL | Binary logic: Output signal | |
| bGPCompareOut<br>BOOL | Analog comparison: Output signal | |
| bGPSignalOut1<br>...<br>bGPSignalOut4<br>BOOL | Binary signal monitor: Output signals<br>• The signal sources to be output are selected in C00411/1...4.<br>• A bit coded inversion of the output signals can be parameterised in C00412. | |

| Identifier / Data type | Value/meaning |
| --- | --- |
| nGPSignalOut1_a<br>...<br>nGPSignalOut4_a<br>BOOL | Analog signal monitor: Output signals<br>• The signal sources to be output are selected in C00410/1...4.<br>• Gain and offset for each output signal can be parameterised in C00413/1...8. |
| bGPDFlipFlop_Out<br>BOOL | D-FlipFlop: Output signal |
| bGPDFlipFlop_NegOut<br>BOOL | D-FlipFlop: Negated output signal |
| bGPCounter1Equal<br>BOOL | Numerator: Status signal "Comparison value reached"<br>TRUE: Comparison value *wGPCounter1CmpVal* reached. |
| wGPCounter1Out<br>WORD | Numerator: Counter content<br>• Internal limitation to ± 32767<br>• The most significant bit determines the sign! |
| **Free outputs**<br>The following outputs can freely be interconnected on the application level.<br>The signals from the application level can be transferred to the I/O level via these outputs. | |
| bFreeOut1 ... bFreeOut8<br>BOOL | Free outputs for digital signals |
| wFreeOut1 ... wFreeOut4<br>WORD | Free outputs for 16-bit signals |
| dnFreeOut1_p<br>dnFreeOut2_p<br>WORD | Free outputs for 32-bit signals |

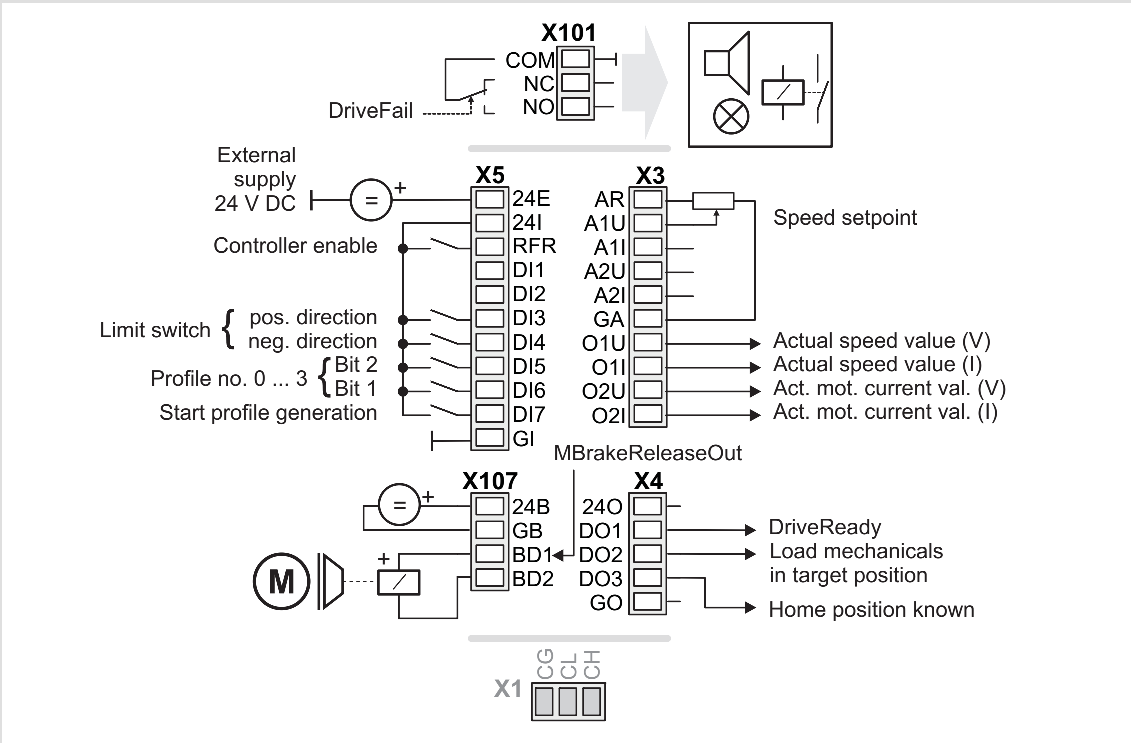
## 8.5.3 Terminal assignment of the control modes

The following comparison provides information about which inputs/outputs of the application block **LA_TabPos** are interconnected to the digital and analog input/output terminals of the drive controller in the different control modes.

<table>
<tr><th rowspan="2"></th><th colspan="8">Control mode (C00007)</th></tr>
<tr><th>10: Terminals 0</th><th>12: Terminals 2</th><th>14: Terminals 11</th><th>16: Terminal 16</th><th>20: Keypad</th><th>21: PC</th><th>30: CAN</th><th>40: MCI</th></tr>
<tr><td colspan="9">Digital input terminals</td></tr>
<tr><td>X5/RFR</td><td colspan="8">Controller enable / Reset of error message<br>bFailReset</td></tr>
<tr><td>X5/DI1</td><td>-</td><td>-</td><td>Positive limit switch<br>bLimitSwitchPos</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>X5/DI2</td><td>-</td><td>-</td><td>Negative limit switch<br>bLimitSwitchNeg</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>X5/DI3</td><td>Positive limit switch<br>bLimitSwitchPos</td><td>-</td><td>-</td><td>Manual jog<br>in positive direction<br>bManJogPos</td><td>-</td><td>-</td><td colspan="2">Positive limit switch<br>bLimitSwitchPos</td></tr>
<tr><td>X5/DI4</td><td>Negative limit switch<br>bLimitSwitchNeg</td><td colspan="2">Positioning profile, array bit (valency 4)<br>bPosProfileNo_4</td><td>Manual jog<br>in negative direction<br>bManJogNeg</td><td>-</td><td>-</td><td colspan="2">Negative limit switch<br>bLimitSwitchNeg</td></tr>
<tr><td>X5/DI5</td><td colspan="4">Positioning profile, array bit (valency 2)<br>bPosProfileNo_2</td><td>-</td><td>-</td><td colspan="2">Home position<br>bHomeMark</td></tr>
<tr><td>X5/DI6</td><td colspan="4">Positioning profile, array bit (valency 1)<br>bPosProfileNo_1</td><td>-</td><td>-</td><td colspan="2">Quick stop<br>bSetQuickstop</td></tr>
<tr><td>X5/DI7</td><td colspan="4">Start of profile generation<br>bPosExecute</td><td>-</td><td>-</td><td>Status word - bit 7<br>LP_CanOut1: bState_B7</td><td>Status word - bit 7<br>LP_MciOut: bState_B7</td></tr>
<tr><td colspan="9">Analog input terminals</td></tr>
<tr><td>X3/A1U<br>X3/A1I</td><td colspan="4">Main speed setpoint<br>nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011)</td><td>-</td><td>-</td><td colspan="2">Main speed setpoint<br>nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011)</td></tr>
<tr><td>X3/A2U<br>X3/A2I</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="9">Digital output terminals</td></tr>
<tr><td>X4/DO1</td><td colspan="8">Status "Drive is ready"<br>bDriveReady</td></tr>
<tr><td>X4/DO2</td><td colspan="8">Status "Target position (actual value) is in the target window"<br>bInTarget</td></tr>
<tr><td>X4/DO3</td><td colspan="6">Status "Home position known"<br>bHomePosAvailable</td><td colspan="2">Status "Maximum current limit"<br>bImaxActive</td></tr>
<tr><td>X107/BD1, BD2</td><td colspan="8">Control of the holding brake<br>bMBrakeReleaseOut</td></tr>
<tr><td>X101/COM, NO</td><td colspan="8">Status "Error is pending"<br>bDriveFail</td></tr>
</table>

| | Control mode (C00007) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10: Terminals 0 | 12: Terminals 2 | 14: Terminals 11 | 16: Terminal 16 | 20: Keypad | 21: PC | 30: CAN | 40: MCI |
| **Analog output terminals** | | | | | | | | |
| **X3/O1U**<br>**X3/O1I** | **Actual speed value**<br>nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) | | | | | | | |
| **X3/O2U**<br>**X3/O2I** | **Current motor current**<br>nMotorCurrent_a<br>10V ≡ 100% of $I_{max\_mot}$ (C00022) | | | | | | | |

### 8.5.3.1 Terminals 0



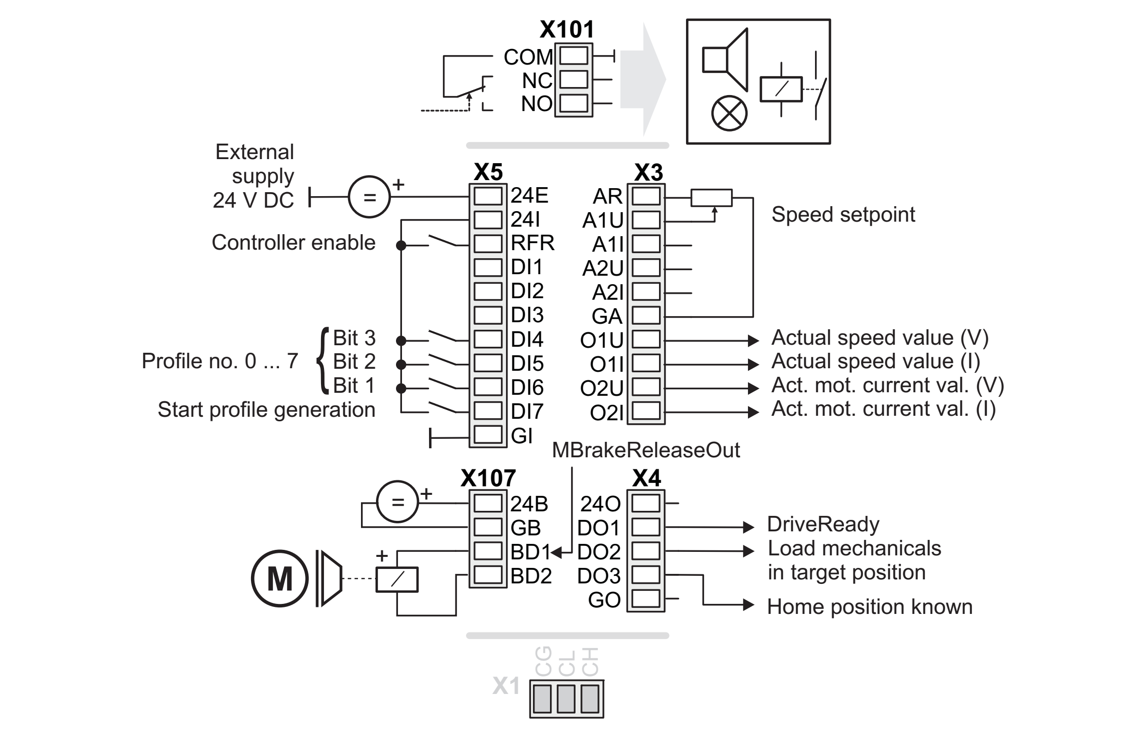

| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_TabPos.bDriveFail |
| | |
| X5/RFR | LA_TabPos.bFailReset |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | LA_TabPos.bLimitSwitchPos |
| X5/DI4 | LA_TabPos.bLimitSwitchNeg |
| X5/DI5 | LA_TabPos.bPosProfileNo_2 |
| X5/DI6 | LA_TabPos.bPosProfileNo_1 |
| X5/DI7 | LA_TabPos.bPosExecute |
| | |
| X107/BD1 | LA_TabPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U<br>X3/A1I | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U<br>X3/O1I | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/O2U<br>X3/O2I | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max\_mot}$ (C00022) |
| | |
| X4/DO1 | LA_TabPos.bDriveReady |
| X4/DO2 | LA_TabPos.bInTarget |
| X4/DO3 | LA_TabPos.bHomePosAvailable |

When the profile is defined, the operating mode in the Lenze setting is changed simultaneously:

| bPosProfileNo_2 (DI5) | bPosProfileNo_1 (DI6) | Selected profile | Activation of operating mode |
|---|---|---|---|
| FALSE | FALSE | 0 | Speed follower |
| FALSE | TRUE | 1 | Homing |
| TRUE | FALSE | 2 | Manual jog |
| TRUE | TRUE | 3 | Positioning |

### 8.5.3.2 Terminals 2



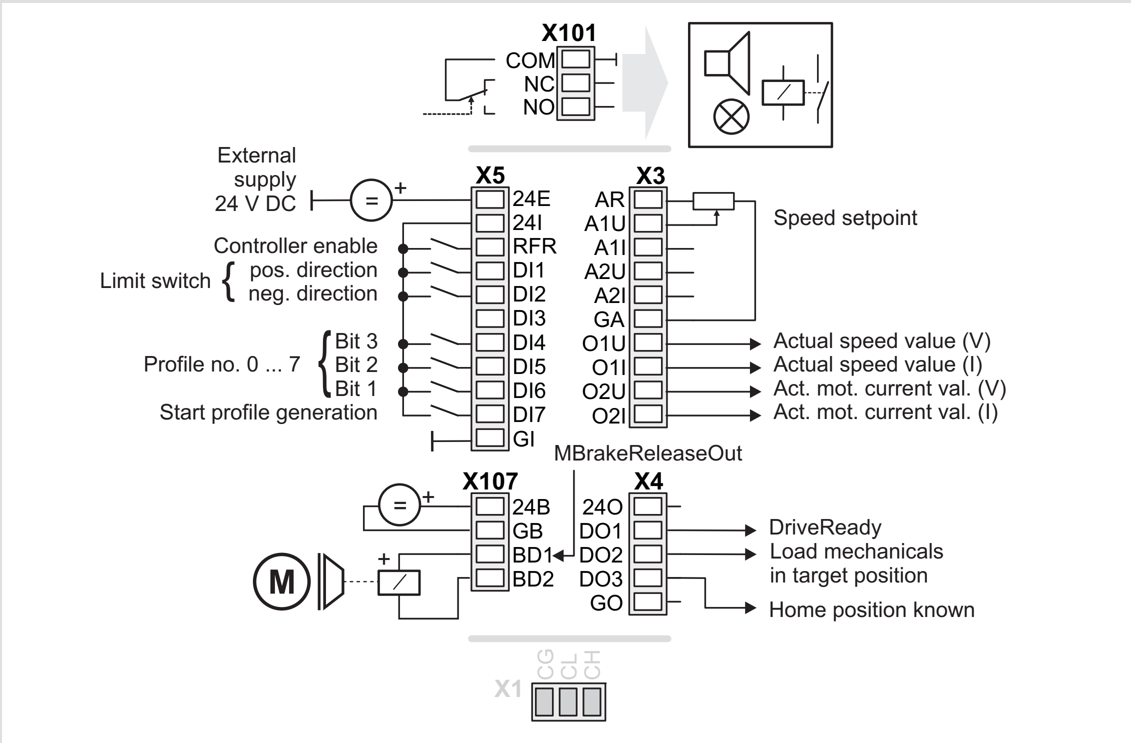

| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_TabPos.bDriveFail |
| | |
| X5/RFR | LA_TabPos.bFailReset |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | - |
| X5/DI4 | LA_TabPos.bPosProfileNo_4 |
| X5/DI5 | LA_TabPos.bPosProfileNo_2 |
| X5/DI6 | LA_TabPos.bPosProfileNo_1 |
| X5/DI7 | LA_TabPos.bPosExecute |
| | |
| X107/BD1 | LA_TabPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/O1I | |
| X3/O2U | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max\_mot}$ (C00022) |
| X3/O2I | |
| | |
| X4/DO1 | LA_TabPos.bDriveReady |
| X4/DO2 | LA_TabPos.bInTarget |
| X4/DO3 | LA_TabPos.bHomePosAvailable |

When the profile is defined, the operating mode in the Lenze setting is changed simultaneously:

| bPosProfileNo_4 **(DI4)** | bPosProfileNo_2 **(DI5)** | bPosProfileNo_1 **(DI6)** | **Selected profile** | **Activation of operating mode** |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | 0 | Speed follower |
| FALSE | FALSE | TRUE | 1 | Homing |
| FALSE | TRUE | FALSE | 2 | Manual jog |
| FALSE | TRUE | TRUE | 3 | Positioning |
| | ... | | ... | |
| TRUE | TRUE | TRUE | 7 | |

### 8.5.3.3 Terminals 11



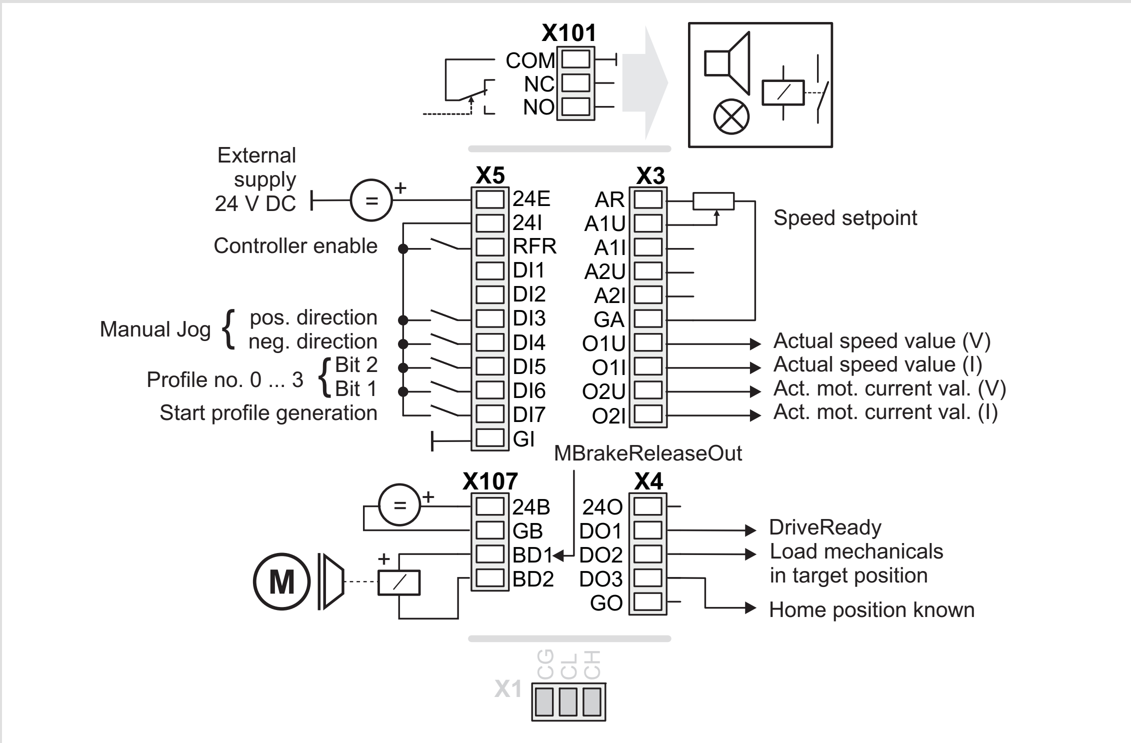

| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_TabPos.bDriveFail |
| | |
| X5/RFR | LA_TabPos.bFailReset |
| X5/DI1 | LA_TabPos.bLimitSwitchPos |
| X5/DI2 | LA_TabPos.bLimitSwitchNeg |
| X5/DI3 | - |
| X5/DI4 | LA_TabPos.bPosProfileNo_4 |
| X5/DI5 | LA_TabPos.bPosProfileNo_2 |
| X5/DI6 | LA_TabPos.bPosProfileNo_1 |
| X5/DI7 | LA_TabPos.bPosExecute |
| | |
| X107/BD1 | LA_TabPos.bMBrakeReleaseOut |
| X107/BD2 | - |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/O1I | |
| X3/O2U | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max\_mot}$ (C00022) |
| X3/O2I | |
| | |
| X4/DO1 | LA_TabPos.bDriveReady |
| X4/DO2 | LA_TabPos.bInTarget |
| X4/DO3 | LA_TabPos.bHomePosAvailable |

When the profile is defined, the operating mode in the Lenze setting is changed simultaneously:

| bPosProfileNo_4 **(DI4)** | bPosProfileNo_2 **(DI5)** | bPosProfileNo_1 **(DI6)** | **Selected profile** | **Activation of operating mode** |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | 0 | Speed follower |
| FALSE | FALSE | TRUE | 1 | Homing |
| FALSE | TRUE | FALSE | 2 | Manual jog |
| FALSE | TRUE | TRUE | 3 | Positioning |
| | ... | | ... | |
| TRUE | TRUE | TRUE | 7 | |

### 8.5.3.4 Terminal 16



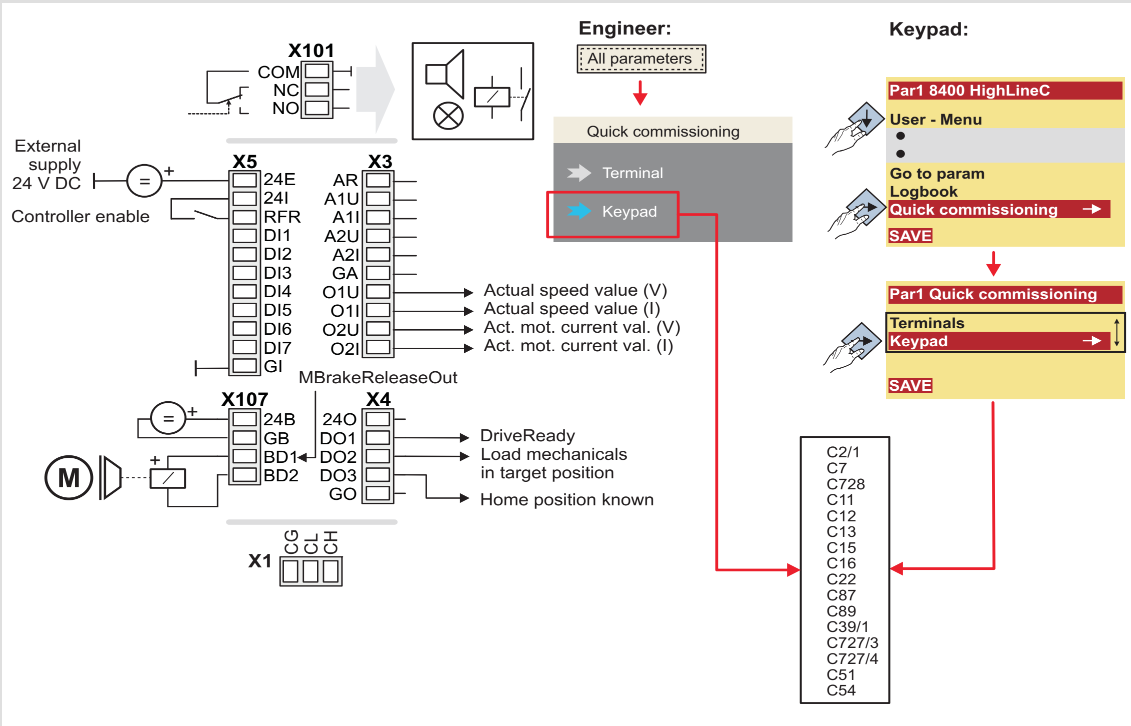

| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_TabPos.bDriveFail |
| | |
| X5/RFR | LA_TabPos.bFailReset |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | LA_TabPos.bManJogPos |
| X5/DI4 | LA_TabPos.bManJogNeg |
| X5/DI5 | LA_TabPos.bPosProfileNo_2 |
| X5/DI6 | LA_TabPos.bPosProfileNo_1 |
| X5/DI7 | LA_TabPos.bPosExecute |
| | |
| X107/BD1 | LA_TabPos.bMBrakeReleaseOut |
| X107/BD2 | - |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/O1I | |
| X3/O2U | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max\_mot}$ (C00022) |
| X3/O2I | |
| | |
| X4/DO1 | LA_TabPos.bDriveReady |
| X4/DO2 | LA_TabPos.bInTarget |
| X4/DO3 | LA_TabPos.bHomePosAvailable |

When the profile is defined, the operating mode in the Lenze setting is changed simultaneously:

| bPosProfileNo_2 (DI5) | bPosProfileNo_1 (DI6) | Selected profile | Activation of operating mode |
|---|---|---|---|
| FALSE | FALSE | 0 | Speed follower |
| FALSE | TRUE | 1 | Homing |
| TRUE | FALSE | 2 | Manual jog |
| TRUE | TRUE | 3 | Positioning |

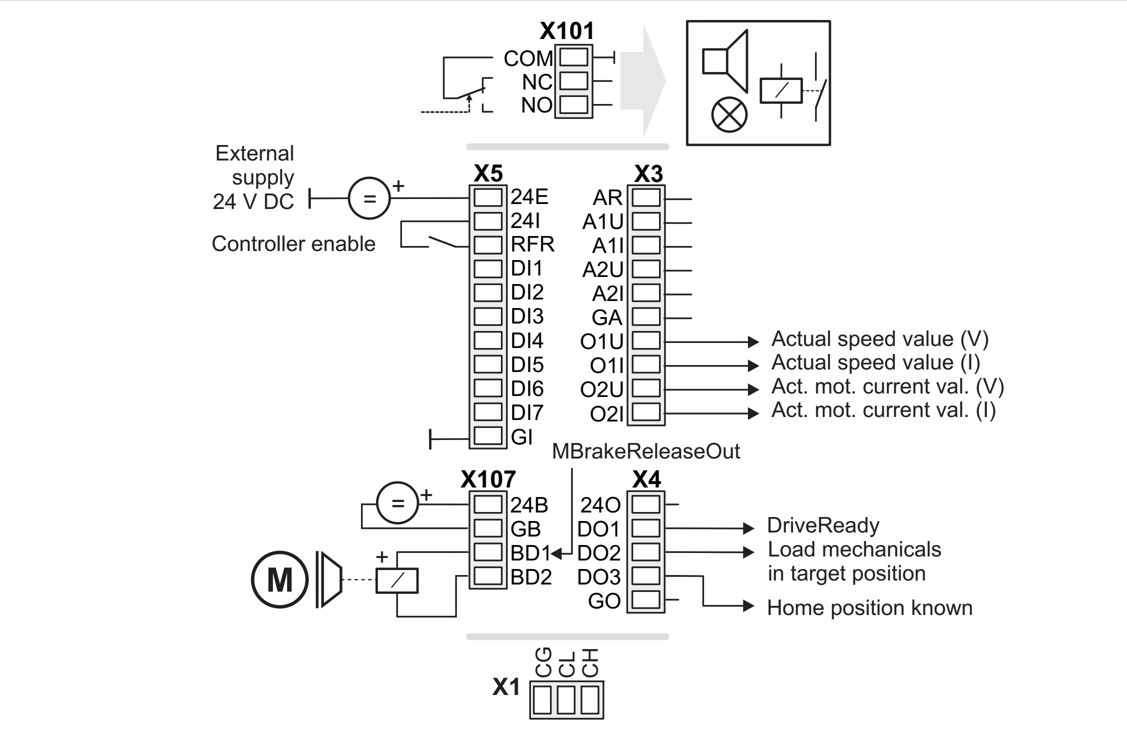
### 8.5.3.5 Keypad




| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_TabPos.bDriveFail |
| | |
| X5/RFR | LA_TabPos.bFailReset |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | - |
| X5/DI4 | - |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_TabPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/O1I | |
| X3/O2U | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max\_mot}$ (C00022) |
| X3/O2I | |
| | |
| X4/DO1 | LA_TabPos.bDriveReady |
| X4/DO2 | LA_TabPos.bInTarget |
| X4/DO3 | LA_TabPos.bHomePosAvailable |

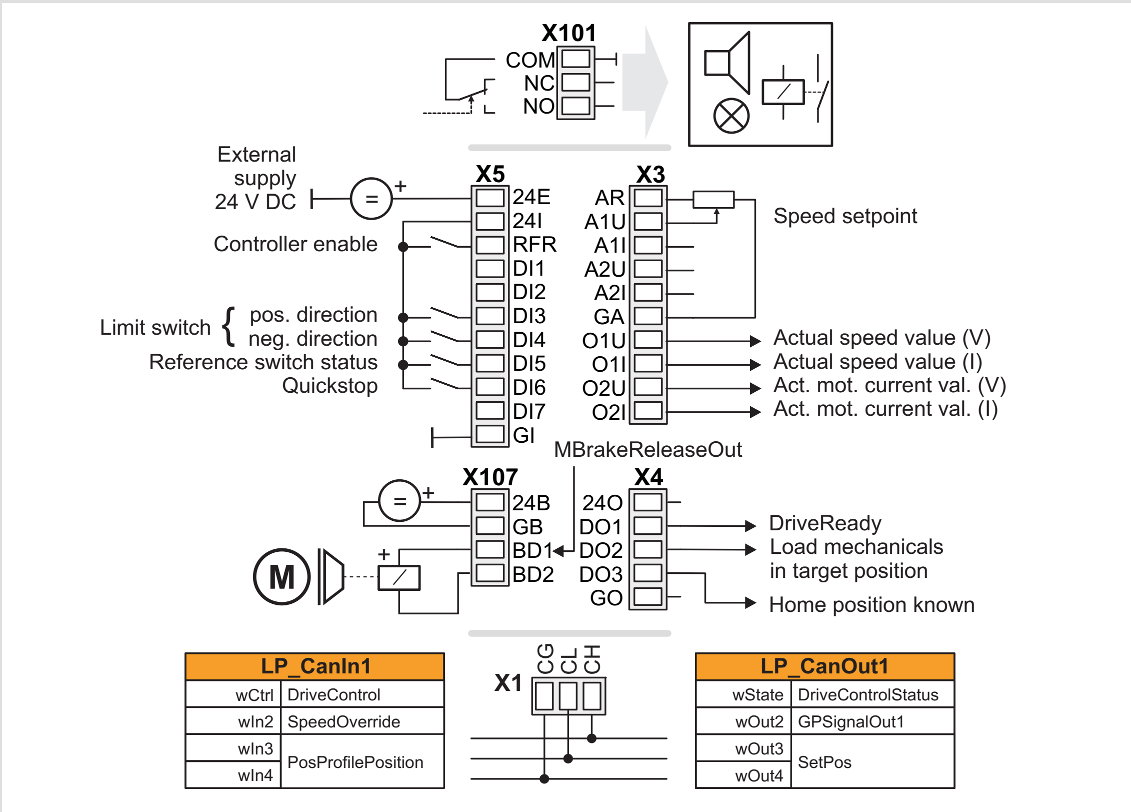
### 8.5.3.6 PC




| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_TabPos.bDriveFail |
| | |
| X5/RFR | LA_TabPos.bFailReset |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | - |
| X5/DI4 | - |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_TabPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | - |
| X3/A1I | - |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/O1I | |
| X3/O2U | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max\_mot}$ (C00022) |
| X3/O2I | |
| | |
| X4/DO1 | LA_TabPos.bDriveReady |
| X4/DO2 | LA_TabPos.bInTarget |
| X4/DO3 | LA_TabPos.bHomePosAvailable |

### 8.5.3.7 CAN

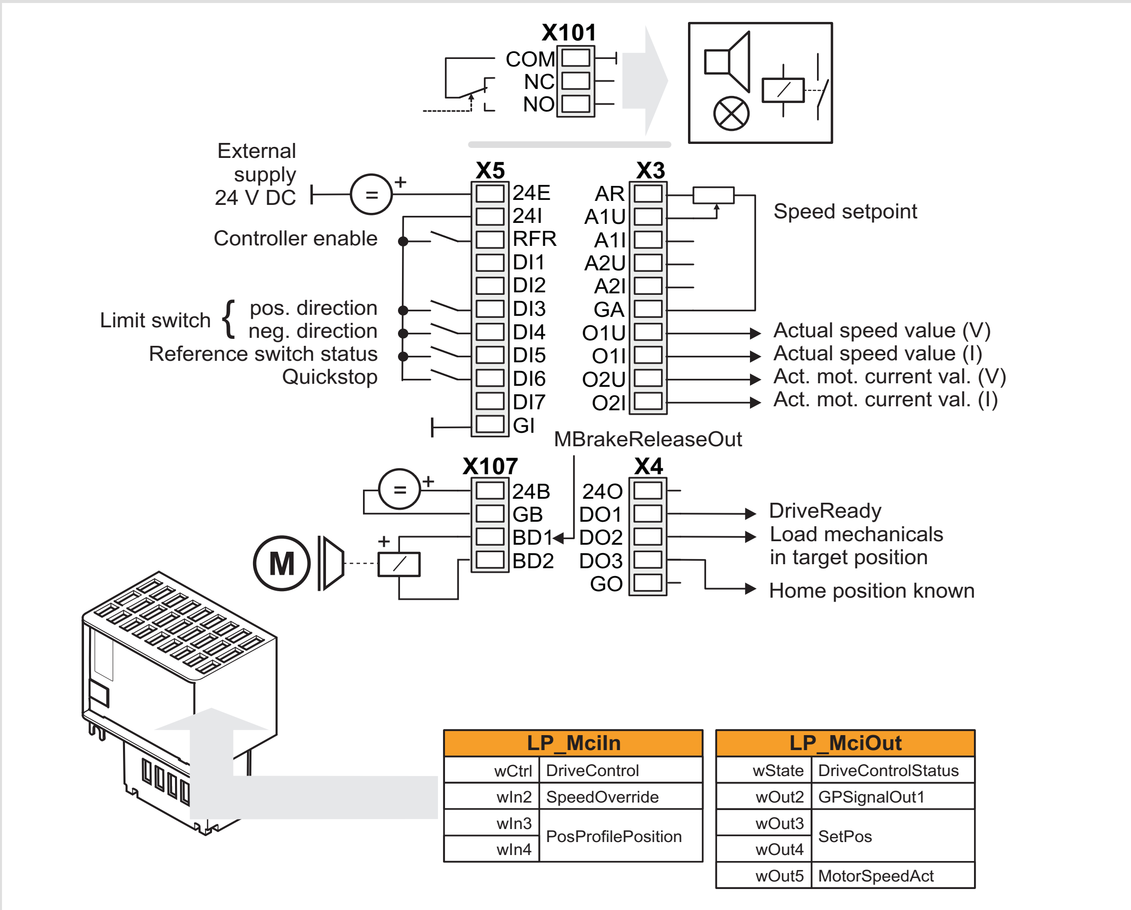



| Connection | Assignment |
|---|---|
| **X101/NC-NO** | LA_TabPos.bDriveFail |
| | |
| **X5/RFR** | LA_TabPos.bFailReset |
| **X5/DI1** | - |
| **X5/DI2** | - |
| **X5/DI3** | LA_TabPos.bLimitSwitchPos |
| **X5/DI4** | LA_TabPos.bLimitSwitchNeg |
| **X5/DI5** | LA_TabPos.bHomeMark |
| **X5/DI6** | LA_TabPos.bSetQuickstop |
| **X5/DI7** | LP_CanOut1: bState_B7 |
| | |
| **X107/BD1** | LA_TabPos.bMBrakeReleaseOut |
| **X107/BD2** | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| **X3/A1U** | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X3/A1I** | |
| **X3/A2U** | - |
| **X3/A2I** | - |
| **X3/O1U** | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| **X3/O1I** | |
| **X3/O2U** | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max\_mot}$ (C00022) |
| **X3/O2I** | |
| | |
| **X4/DO1** | LA_TabPos.bDriveReady |
| **X4/DO2** | LA_TabPos.bInTarget |
| **X4/DO3** | LA_TabPos.bImaxActive |

▸ Process data assignment for fieldbus communication (434)

### 8.5.3.8 MCI




| Connection | Assignment |
|---|---|
| X101/NC-NO | LA_TabPos.bDriveFail |
| | |
| X5/RFR | LA_TabPos.bFailReset |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | LA_TabPos.bLimitSwitchPos |
| X5/DI4 | LA_TabPos.bLimitSwitchNeg |
| X5/DI5 | LA_TabPos.bHomeMark |
| X5/DI6 | LA_TabPos.bSetQuickstop |
| X5/DI7 | LP_MciOut: bState_B7 |
| | |
| X107/BD1 | LA_TabPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_TabPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | LA_TabPos.nMotorSpeedAct_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/O1I | |
| X3/O2U | LA_TabPos.nMotorCurrent_a<br>10V ≡ 100% of $I_{max\_mot}$ (C00022) |
| X3/O2I | |
| | |
| X4/DO1 | LA_TabPos.bDriveReady |
| X4/DO2 | LA_TabPos.bInTarget |
| X4/DO3 | LA_TabPos.bImaxActive |

▶ Process data assignment for fieldbus communication (📖 434)

### 8.5.4 Process data assignment for fieldbus communication

The fieldbus communication is connected (preconfigured) to the previously selected technology application by selecting the corresponding control mode in C00007:

- "30: CAN" for the connection to the system bus (CAN)
- "40: MCI" for the connection to a plugged-on communication module (e.g. PROFIBUS)

The assignment of the process data words does not depend on the applied bus system but exclusively on the application:

| Input words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControl | Control word<br>• See table below for bit assignment. |
| Word 2 | SpeedOverride | Value for speed override<br>• Percentage multiplier for the currently active speed.<br>• 16384 ≡ 100 % of the maximum traversing speed (display in C01211/1).<br>• Values > 16384 are ignored.<br>• If the override value is 0 %, the drive is brought to a standstill.<br>▸ Speed override (📖 560) |
| Word 3 and 4 | PosProfilePosition | Selection of the target position in [increments]<br>• The mode for calculating the position is selected in C01296/1.<br>• 65535 [Increments] ≡ 1 motor revolution<br>• The position resolution, i.e. how many increments form one unit, is displayed in C01205. |
| Word 5 ... 16 | - | Not preconfigured<br>• Only available for control mode "40: MCI". |

| Control word | Name | Function |
|---|---|---|
| Bit 0 | MckOperationMode_1 | Binary-coded selection of the operating mode of the **Motion Control Kernel**<br>• For a detailed description of the individual control bits, see chapter "MCK control word". (📖 481) |
| Bit 1 | MckOperationMode_2 | |
| Bit 2 | MckOperationMode_4 | |
| Bit 3 | CINH | 1 ≡ Inhibit drive controller (controller inhibit): The drive controller changes to the device state "SwitchedOn".<br>▸ Enable/Inhibit controller (📖 97) |
| Bit 4 | PosProfileNo_1 | Selection of the profile number<br>▸ Stipulation of the profile to be executed (📖 552) |
| Bit 5 | PosProfileNo_2 | |
| Bit 6 | PosProfileNo_4 | |
| Bit 7 | PosProfileNo_8 | |
| Bit 8 | PosExecute | 1 ≡ Start travel job<br>▸ Positioning (📖 541) |
| Bit 9 | EnableSpeedOverride | 1 ≡ Activate speed override<br>▸ Speed override (📖 560) |
| Bit 10 | HomeSetPosition | 1 ≡ Set home position<br>▸ Homing (📖 517) |
| Bit 11 | FailReset | 1 ≡ Reset error message<br>▸ Reset error messages (📖 606) |
| Bit 12 | ManJogNeg | Manual jog in positive/negative direction<br>▸ Manual jog (📖 533) |
| Bit 13 | ManJogPos | |
| Bit 14 | - | Free control bit 14 (not assigned, freely assignable) |
| Bit 15 | - | Free control bit 15 (not assigned, freely assignable) |

| Output words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControlStatus | Status word<br>• See table below for bit assignment. |
| Word 2 | GPSignalOut1 | Analog signal monitor: Output signal 1<br>• The selection of the signal source to output is executed in C00410/1.<br>• Gain and offset for the output signal can be parameterised in C00413/1 and C00413/2.<br>• For a detailed functional description see the L_SignalMonitor_a FB. |
| Word 3 and 4 | SetPos | Absolute position setpoint in [increments]<br>• 65535 [Increments] ≡ 1 motor revolution<br>• The position resolution, i.e. how many increments form one unit, is displayed in C01205. |
| Word 5 | MotorSpeedAct | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• Only available for control mode "40: MCI". |
| Word 6 ... 16 | - | Not preconfigured<br>• Only available for control mode "40: MCI". |

| Status word | Name | Status |
|---|---|---|
| Bit 0 | DriveFail | 1 ≡ Drive controller in error status<br>• "Fault" device status is active. |
| Bit 1 | GPSignalOut1 | Binary signal monitor: Output signals 1 & 2<br>• The signal sources to be output are selected in C00411/1...2.<br>• A bit coded inversion of the output signals can be parameterised in C00412.<br>• For a detailed functional description see FB L_SignalMonitor_b. |
| Bit 2 | GPSignalOut2 | |
| Bit 3 | CInhActive | 1 ≡ Controller inhibit is active |
| Bit 4 | DriveReady | 1 ≡ Drive controller is ready for operation<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the controller is still inhibited by the user (controller inhibit). |
| Bit 5 | DigitalInput5 | Signal from the digital input DI5 |
| Bit 6 | DigitalInput6 | Signal from the digital input DI6 |
| Bit 7 | DigitalInput7 | Signal from the digital input DI7 |
| Bit 8 | InTarget | 1 ≡ Target position (actual value) is in the target window |
| Bit 9 | ProfileBusy | 1 ≡ Profile positioning is active |
| Bit 10 | HomePosAvailable | 1 ≡ Home position is known |
| Bit 11 | SpeedActCompare | Result of the speed comparison<br>• In case of the "Open loop" operation:<br>1 ≡ Speed setpoint < comparison value (C00024)<br>• For "Closed loop" operation:<br>1 ≡ actual speed value < comparison value (C00024) |
| Bit 12 | DigitalInput4 | Signal from the digital input DI4 |
| Bit 13 | DigitalInput3 | Signal from the digital input DI3 |
| Bit 14 | QspIsActive | 1 ≡ Quick stop is active |
| Bit 15 | - | Free status bit 15 (not assigned, freely assignable) |

### 8.5.5 Setting parameters (short overview)

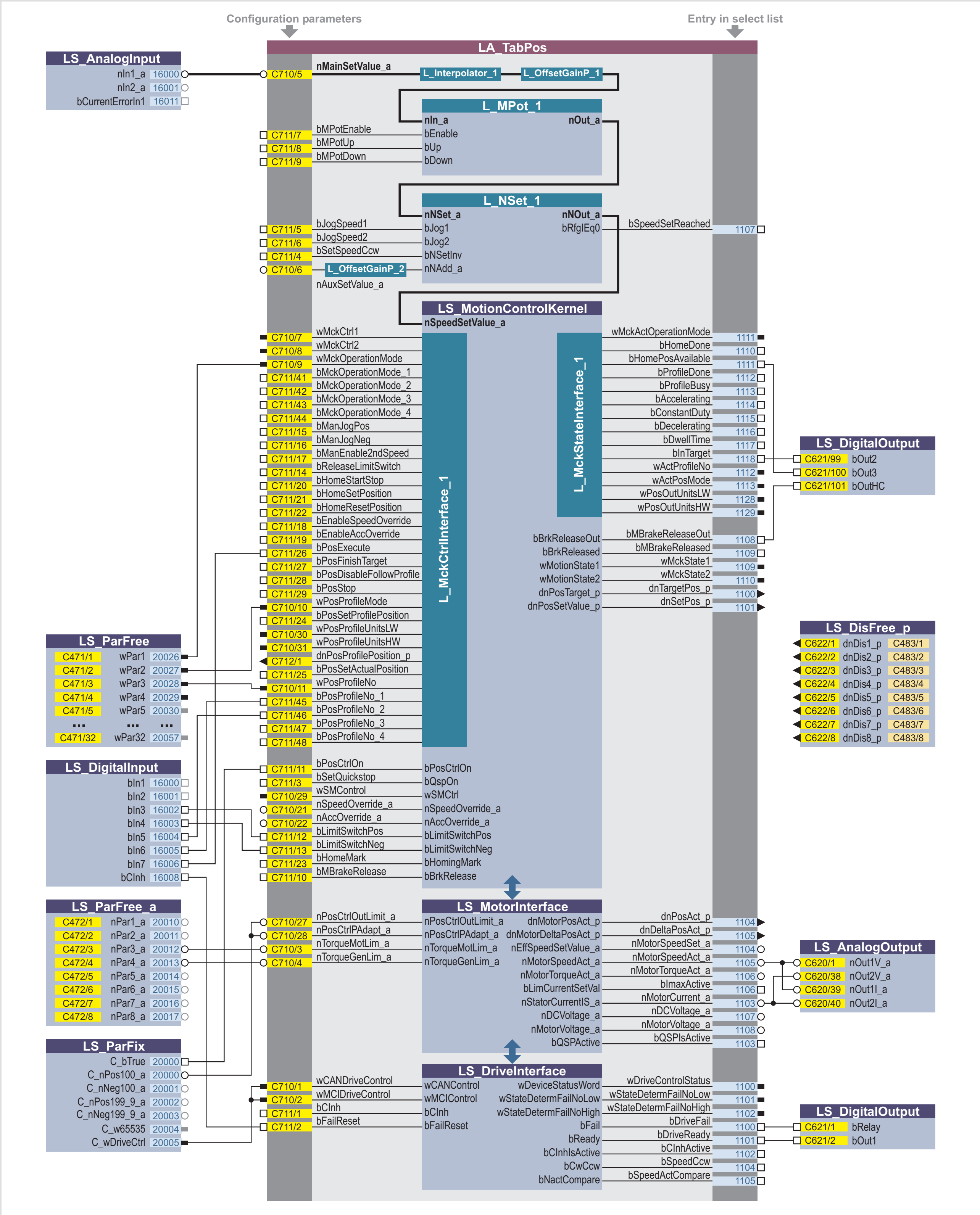
| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00012 | Accel. time - main setpoint | 2.000 | s |
| C00013 | Decel. time - main setpoint | 2.000 | s |
| C00024 | Comparison value N_Act | 0.00 | % |
| C00039/1 | Fixed setpoint 1 | 40.00 | % |
| C00039/2 | Fixed setpoint 2 | 60.00 | % |
| C00182 | S-ramp time PT1 | 20.00 | s |
| C00190 | Setpoint arithmetic | 0: NOut = NSet | |
| C00220 | Accel. time - add. setpoint | 0.000 | s |
| C00221 | Decel. time - add. setpoint | 0.000 | s |
| C00632/1 | L_NSet_1: Blocking speed 1 max | 0.00 | % |
| C00632/2 | L_NSet_1: Blocking speed 2 max | 0.00 | % |
| C00632/3 | L_NSet_1: Blocking speed 3 max | 0.00 | % |
| C00633/1 | L_NSet_1: Blocking speed 1 min | 0.00 | % |
| C00633/2 | L_NSet_1: Blocking speed 2 min | 0.00 | % |
| C00633/3 | L_NSet_1: Blocking speed 3 min | 0.00 | % |
| C00635 | L_NSet_1: nMaxLimit | 199.99 | % |
| C00636 | L_NSet_1: nMinLimit | -199.99 | % |
| C00670 | L_OffsetGainP_1: Gain | 1.0000 | |
| C00671 | L_OffsetGainP_2: Gain | 1.0000 | |
| C00696 | L_OffsetGainP_1: Offset | 0.00 | % |
| C00697 | L_OffsetGainP_2: Offset | 0.00 | % |
| C00800 | L_MPot_1: Upper limit | 100.00 | % |
| C00801 | L_MPot_1: Lower limit | -100.00 | % |
| C00802 | L_MPot_1: Accel. time | 10.0 | s |
| C00803 | L_MPot_1: Decel. time | 10.0 | s |
| C00804 | L_MPot_1: Inactive fct. | 0: Retain value | |
| C00805 | L_MPot_1: Init fct. | 0: Load last value | |
| C00806 | L_MPot_1: Use | 0: No | |
| C01297 | Alternative function | Bit coded | |
| C01298/1 | MCK operating mode at profile no. 0 | 1: Follower | |
| C01298/2 | MCK operating mode at profile no. 1 | 2: Homing | |
| C01298/3 | MCK operating mode at profile no. 2 | 3: ManualJog | |
| C01298/4 | MCK operating mode at profile no. 3...15 | 4: Positioning | |
| C01299 | MCKI: Status MCKInterface | - | |

**Related topics:**

- "GeneralPurpose" functions (📖 467)

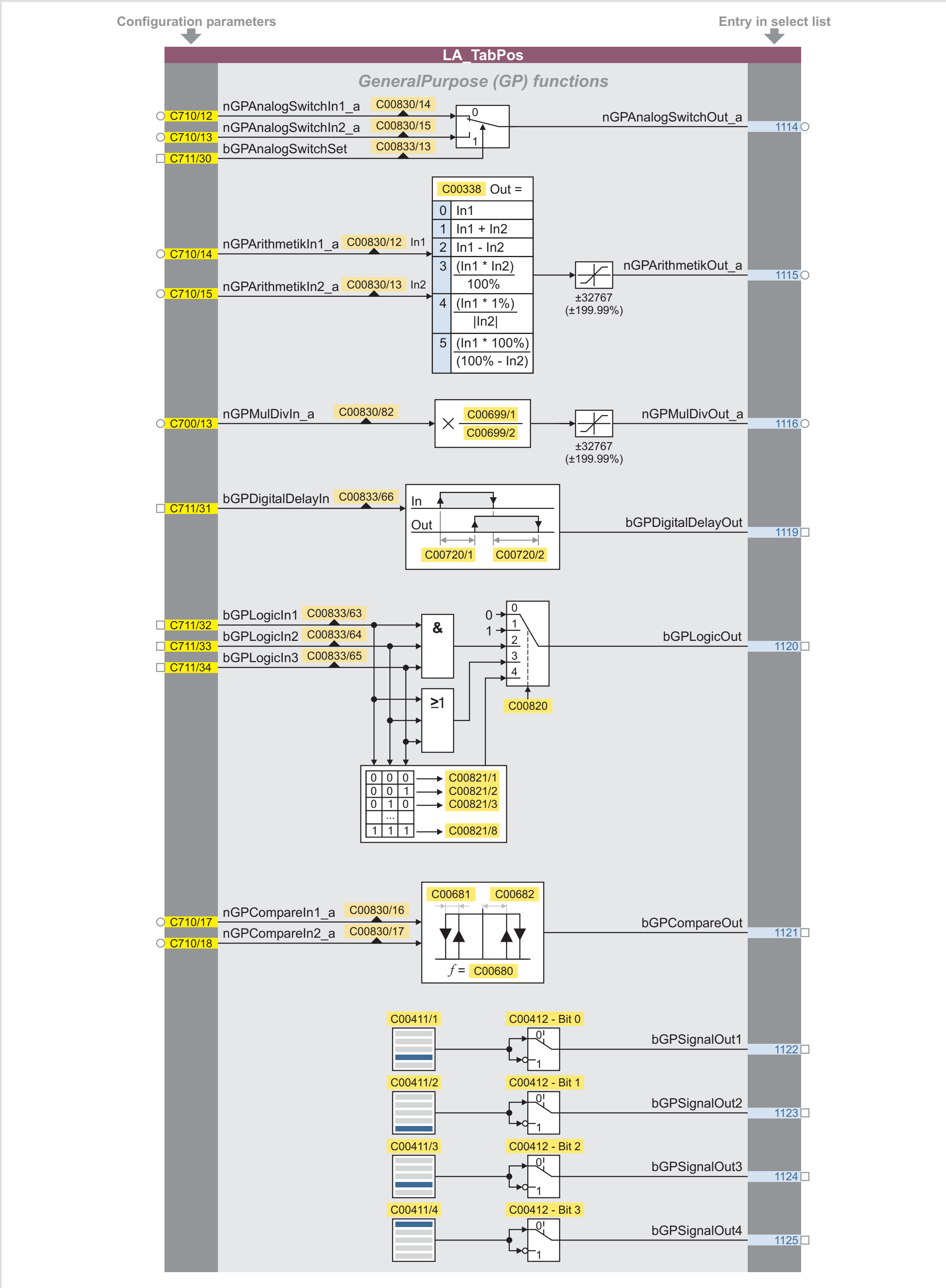
### 8.5.6 Configuration parameters

If required, the subcodes of C00710, C00711 and C00712 serve to change the pre-configured assignment of the application inputs:

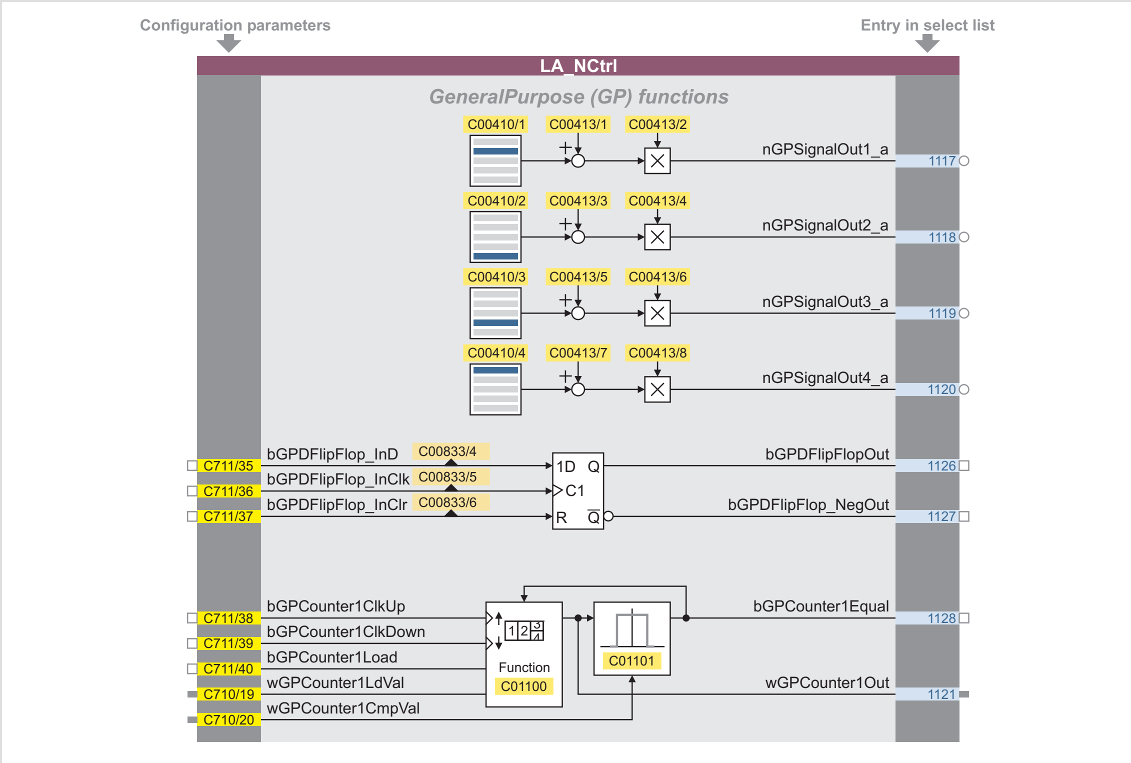


[8-1] Pre-assignment of the "Table positioning" application in the "Terminals 0" control mode

## Configuration parameters for "GeneralPurpose" functions



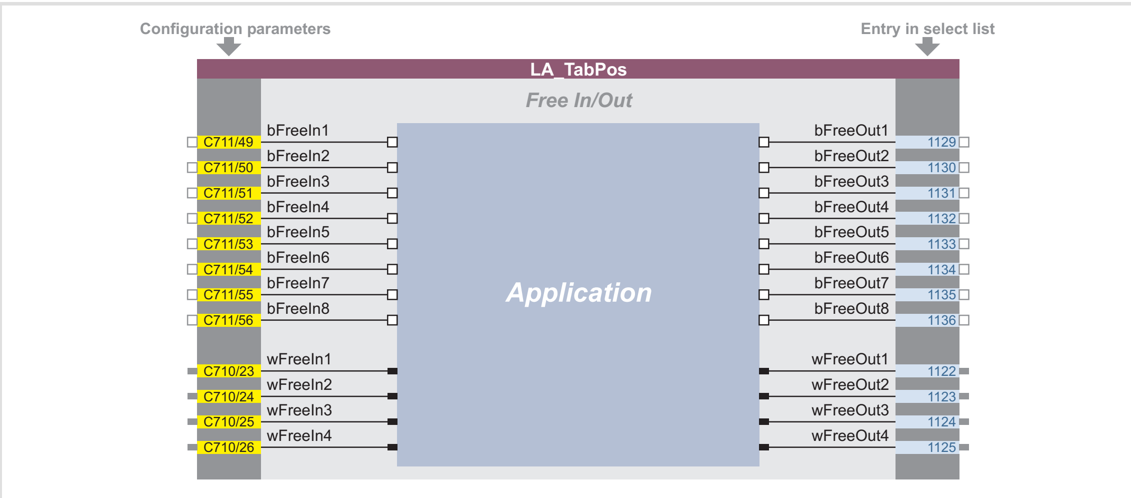

[8-2] "GeneralPurpose" functions




[8-3] "GeneralPurpose" functions (continuation)

## Free inputs and outputs

These inputs can be freely interconnected in the application level. They can be used to transfer signals from the I/O level to the application level and vice versa.




[8-4] Free inputs/outputs

## Related topics:

- User-defined terminal assignment (📖 370)
- "GeneralPurpose" functions (📖 467)

## 8.6 TA "Switch-off positioning"

The basic principle of this technology application is to travel to a switch-off sensor (e.g. a limit switch) in a speed-controlled manner and to stop as close as possible at this position. Unlike other positioning controls, the switch-off positioning neither has a position feedback nor calculates the path in advance. Thus, the accuracy that can be achieved depends on various factors such as the speed at which the switch-off sensor is advanced.

In addition, a pre-switch off can be implemented which requires a sufficient number of unassigned digital inputs on the controller which can be used to connect other sensors for the additional stop positions. These sensors effect a reduction in speed before the last switch-off sensor is reached.

### Features

- Pre-configured control modes for terminals and bus control (with predefined process data connection to the fieldbus)
- Free configuration of input and output signals
- Offset, gain, and negation of main setpoint & additional setpoint
- Up to 15 fixed setpoints for speed and ramp time
- Adjustable setpoint ramp times
- Freely selectable, variable ramp shape
- Automatic holding brake control
- Quick stop (QSP) with adjustable ramp time
- Integrated disposable "GeneralPurpose" functions:
  Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop
- Interface to the safety module (optional)
- Integration of encoder feedback
- Switch-off sensor management for the implementation of a pre-switch off

### Decision criteria

| Criteria | Switch-off positioning with constant load | Switch-off positioning with variable load |
|---|---|---|
| **Operating mode** | V/f characteristic without speed sensor.<br>Alternatively for large breakaway torques: Use of a sensorless vector control (only applicable for horizontal movements). | |
| **Limit switch evaluation** | One limit switch is required per direction of movement. When the limit switch is reached, the drive is brought to a standstill leaded by the deceleration ramp or the QSP ramp. | Per direction of movement, one limit switch and one initiator is required for fast/slow changeover. When this initiator is reached, the speed of the drive is reduced to a creeping speed (selected jog value). When the limit switch is reached, the drive is brought to a standstill leaded by the deceleration ramp or the QSP ramp. |

| Criteria | Switch-off positioning with constant load | Switch-off positioning with variable load |
| --- | --- | --- |
| **Positioning accuracy at the motor shaft**<br>The positioning accuracy of the load depends e.g. on the selected mechanics by clearance and friction and must be determined individually. | The ideal case is 5-10° at the motor shaft.<br>Consider the influence of the motor temperature.<br>In the case of a constant load, you can assume a good repeat accuracy during positioning. In the case of variable loads, you must take significant deviations into account. | 5-10° at the motor shaft. As the positioning is executed in a creeping speed, a good repeat accuracy is reached even for variable loads. |
| **Speed setting range** | 1 : 50, based on 50Hz and $M_n$ | 1 : 50, based on 50Hz and $M_n$ |
| **Typical applications** | Switch-off positioning with constant load, e.g. travelling drive, roll-up door. | Switch-off positioning with variable load, e.g. travelling drive, conveying belt, hoists approaching a stop position |

### System limits and exclusion criteria

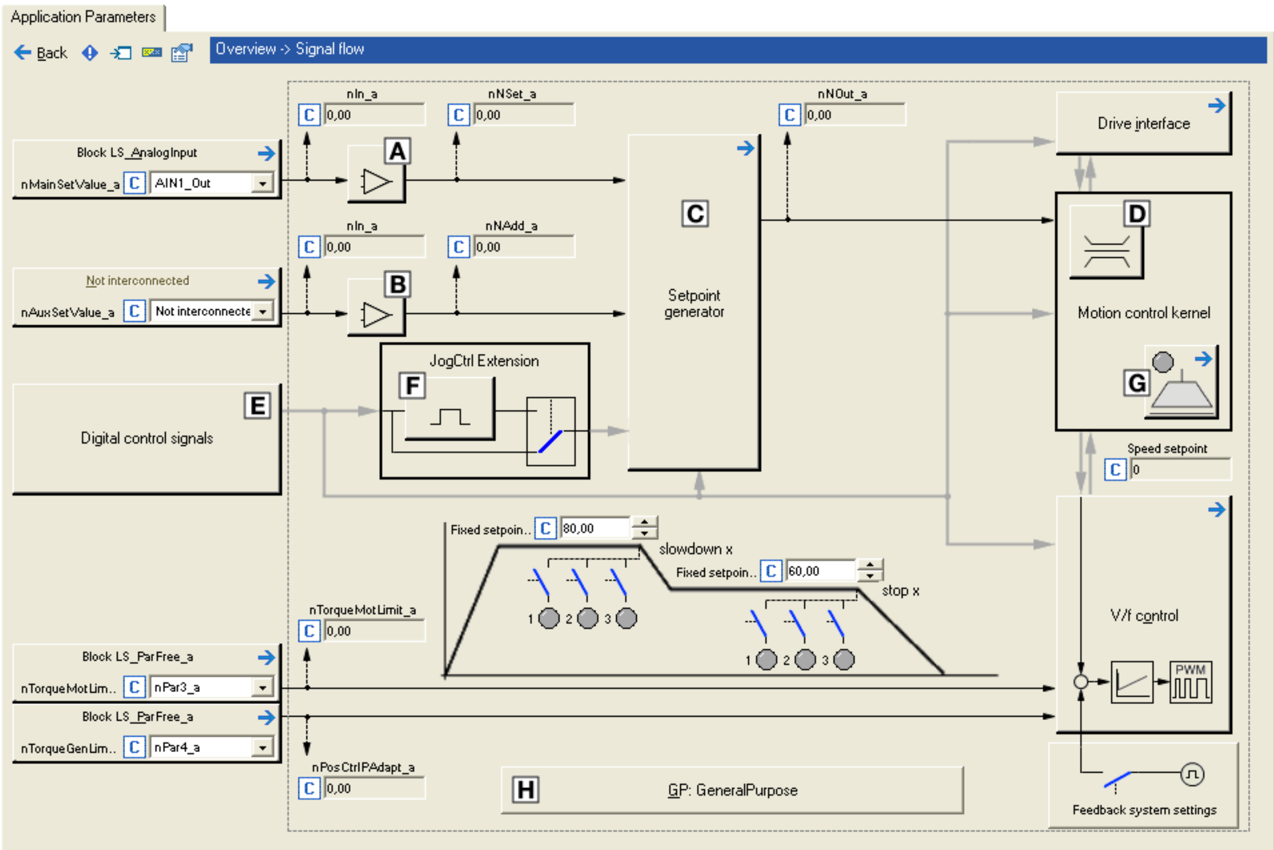
They result from the non-compliance with the decision criteria.

- Compared to systems with speed feedback, the positioning and repeat accuracy is reduced.
- Due to the mechanical hardware limit switches, this concept is only applicable for systems with only a few fixed positions. Changing the target position during the operation or the teaching is not possible.
- If necessary, additional functions like manual jog or homing must be realised externally, e.g. via a control.
- As the 8400 TopLine controller does not meet safety-related functions except STO (Safe Torque Off), you must observe that all safety-related aspects are realised by the plant instructor.
- Especially in the case of an outdoor use or in wet areas, you must consider the corresponding discharge currents when operated with a fault current circuit breaker.
- A table positioning or sequential positioning control is required for highly dynamic applications and jerk-free traversing profiles which is available with the "HighLine" device version.

**Related topics:**

Commissioning of the "Switch-off positioning" technology application (📖 72)

## 8.6.1 Basic signal flow




[8-5] Signal flow of the switch-off positioning

- A Main speed setpoint offset and gain (L_OffsetGainP_1)
- B Additional speed setpoint offset and gain (L_OffsetGainP_2)
- C Setpoint generator (L_NSet_1)
- D Speed setpoint input limitation
- E Terminal assignment & display of digital control signals
- F Selection of edge/level for tripping the ramp down and stop functions (L_JogCtrlExtension_1)
- G Holding brake control
- H Integrated disposable "GeneralPurpose" functions:
  Analog switch, arithmetic, multiplication/division, binary delay element, binary logic, analog comparison, D-flipflop

### 8.6.2 Internal interfaces | application block "LA_SwitchPos"

**Note!**

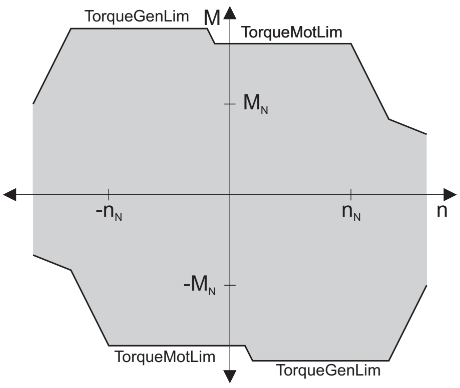
The connectors grayed out in the following table are hidden in the function block editor in the Lenze setting.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| wCANDriveControl<br>WORD | Control word via system bus (CAN) to device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. | |
| wMCIDriveControl<br>WORD | Control word via communication module (e.g. PROFIBUS) to device control<br>• See the "wCANControl/wMCIControl control words" subchapter of the chapter on device control for a detailed description of the individual control bits. | |
| wSMControl<br>WORD | Interface to the optional safety system.<br>• Setting control bit 0 ("SafeStop1") in this control word causes e.g. the automatic deceleration of the drive to standstill within this application (in the **Motion Control Kernel**).<br>• See the "Interface to safety system" subchapter of the chapter on basic drive functions for a detailed description of the control bits. | |
| bCInh<br>BOOL | Enable/Inhibit controller | |
| | FALSE | Enable controller: The controller switches to the "OperationEnabled" device status if no other source of a controller inhibit is active.<br>• C00158 provides a bit coded representation of all active sources/triggers of a controller inhibit. |
| | TRUE | Inhibit controller (controller inhibit): The controller switches to the "SwitchedOn" device status. |
| bFailReset<br>BOOL | Reset error messages<br>In the Lenze setting this input is connected to the digital input controller enable so that a possibly existing error message is reset together with the controller enable (if the cause for the fault is eliminated). | |
| | TRUE | The current fault is reset, if the cause for the fault is eliminated.<br>• If the fault still exists, the error status remains unchanged. |
| bSetQuickstop<br>BOOL | Enable quick stop (QSP)<br>• Also see device command "Activate/Deactivate quick stop". | |
| | TRUE | Activate quick stop<br>• Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0).<br>• The motor is kept at a standstill during closed-loop operation.<br>• A pulse inhibit (CINH) is set if the auto DCB function has been activated via C00019. |
| | FALSE | Deactivate quick stop<br>• The quick stop is deactivated if no other source for the quick stop is active.<br>• C00159 provides a bit-coded representation of active sources/causes for the quick stop. |

<table>
<tr><th>Identifier<br>Data type</th><th colspan="2">Information/possible settings</th></tr>
<tr><td rowspan="4">bSetDCBrake<br>BOOL</td><td colspan="2">Manual DC-injection braking (DCB)<br>• Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking".</td></tr>
<tr><td colspan="2">Note!<br>Holding braking is not possible when this braking mode is used!<br>For controlling a holding brake with low rate of wear, use the basic function "Holding brake control".</td></tr>
<tr><td>FALSE</td><td>Deactivate DC-injection braking.</td></tr>
<tr><td>TRUE</td><td>Activate DC-injection braking, i.e. the drive is brought to a standstill by means of DC-injection braking.<br>• The braking effect stops when the rotor is at standstill.<br>• After the hold time (C00107) has expired, the controller sets the pulse inhibit (CINH).</td></tr>
<tr><td rowspan="2">bRFG_Stop<br>BOOL</td><td colspan="2">Ramp function generator: Maintain the current value of the main setpoint integrator<br>• The speed, for instance, of a running ramp process is immediately kept constant when bRFG_Stop is activated. At the same time, the acceleration/deceleration jumps to the value "0".<br>• For a detailed functional description see the L_NSet FB.</td></tr>
<tr><td>TRUE</td><td>The current value of the main setpoint integrator is held.</td></tr>
<tr><td rowspan="2">nVoltageAdd_a<br>INT</td><td colspan="2">Additive voltage impression<br>• An additional setpoint for the motor voltage can be specified via this process input.<br>• If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time.<br>• If the value is negative, the voltage is reduced.<br>• Scaling: 16384 ≡ 1000 V</td></tr>
<tr><td colspan="2">Stop!<br>Values selected too high may cause the motor to heat up due to the resulting current!</td></tr>
<tr><td rowspan="2">nBoost_a<br>INT</td><td colspan="2">Additional setpoint for the motor voltage at speed = 0<br>• The entire voltage-frequency characteristic is provided with an offset.<br>• Scaling: 16384 ≡ 1000 V</td></tr>
<tr><td colspan="2">Stop!<br>Values selected too high may cause the motor to heat up due to the resulting current!</td></tr>
<tr><td>nPWMAngleOffset<br>INT</td><td colspan="2">Additional offset for the electrical angle of rotation<br>• If a torque is connected, e.g. dynamic acceleration processes can be generated.<br>• Scaling: ±32767 ≡ ±180 ° angle of rotation</td></tr>
</table>

| Identifier | Data type | Information/possible settings |
|---|---|---|
| nTorqueMotLim_a<br>nTorqueGenLim_a | INT | Torque limitation in motor mode and in generator mode<br>• These input signals are directly transferred to the motor control to limit the controller's maximum torque in motor and generator mode.<br>• The drive cannot output a higher torque in motor/generator mode than set here.<br>• The applied values (any polarity) are internally interpreted as absolute values.<br>• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called $I_{max}$ controller.<br>• If sensorless vector control (SLVC) or servo control (SC) is selected, limitation has a direct effect on the torque-producing current component.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) |
| | | Torque limits in motor and generator mode:<br><br> |
| bSetSpeedCcw | BOOL | Change of direction of rotation<br>• For instance if a motor or gearbox is fixed laterally reversed to a machine part, but the setpoint selection should still be executed for the positive direction of rotation. |
| | | FALSE: Direction of rotation to the right (Cw) |
| | | TRUE: Direction of rotation to the left (Ccw) |
| bRLQCw | BOOL | Activate clockwise rotation (fail-safe)<br>• For a detailed functional description see the L_RLQ FB. |
| | | FALSE: Quick stop |
| | | TRUE: Clockwise rotation |
| bRLQCcw | BOOL | Activate counter-clockwise rotation (fail-safe)<br>• For a detailed functional description see the L_RLQ FB. |
| | | FALSE: Quick stop |
| | | TRUE: Counter-clockwise rotation |
| nMainSetValue_a | INT | Main speed setpoint<br>• Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The main setpoint is transformed to a speed setpoint in the setpoint encoder via a ramp function generator with linear or S-shaped ramps.<br>• Upstream to the ramp function generator, a blocking speed masking function and a setpoint MinMax limitation are effective.<br>• For a detailed functional description see the L_NSet FB. |
| nAuxSetValue_a | INT | Additional speed setpoint<br>• Offset and gain of this input signal can be set in C00697 and C00671 for a simple signal adjustment of a setpoint encoder.<br>• Scaling: 16384 ≡ 100 % reference speed (C00011)<br>• The additional speed setpoint can be linked arithmetically with the main speed setpoint behind the ramp function generator.<br>• The additional speed setpoint can be shown via ramp times of a second ramp function generator.<br>• For a detailed functional description see the L_NSet FB. |

| Identifier<br>Data type | Information/possible settings |
|---|---|
| **Switch-off positioning** | |
| bJogCtrlInputSel1<br>bJogCtrlInputSel2<br>BOOL | Selection inputs for a binary coded selection of the switch-off position 1 ... 3<br>• Activation of the signal pairs *bJogCtrlSlowDown1/bJogCtrlStop1*, *bJogCtrlSlowDown2/bJogCtrlStop2* or *bJogCtrlSlowDown3/bJogCtrlStop3* according to the Truth table for activating the pre-switch off. |
| bJogCtrlRfgIn<br>BOOL | Ramping down of the setpoint generator in the downstream L_NSet FB according to the Truth table for activating the pre-switch off |
| bJogCtrlJog1<br>bJogCtrlJog2<br>BOOL | Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint<br>• If the pre-switch off is inactive (*bJogCtrlInputSel1* and *bJogCtrlInputSel2* are both set to FALSE), the two control signals are passed through 1:1 to the downstream FB L_NSet.<br>• To achieve the desired behaviour (starting at high speed, pre-switch off at low speed), both inputs must be set to TRUE.<br>• Fixed setpoint 2 must be less than fixed setpoint 3!<br>Otherwise, the drive will start at a low speed and accelerate after the pre-switch off.<br>• If in additional to the inputs *bJogCtrlJog1* and *bJogCtrlJog2* the selection inputs *bJogSpeed4* and *bJogSpeed8* are assigned, different fixed setpoints can result from this and the drive may travel with different speeds than selected via *bJogCtrlJog1* and *bJogCtrlJog2*. |
| bJogCtrlSlowDown1<br>bJogCtrlSlowDown2<br>bJogCtrlSlowDown3<br>BOOL | Activation of fixed setpoint 2 in the downstream L_NSet FB<br>• These inputs do only have a function if they were previously activated via *bJogCtrlInputSel1* and *bJogCtrlInputSel2* (see Truth table for activating the pre-switch off). |
| bJogCtrlStop1<br>bJogCtrlStop2<br>bJogCtrlStop3<br>BOOL | Ramping down of the ramp function generator in the downstream L_NSet FB<br>• These inputs do only have a function if they were previously activated via *bJogCtrlInputSel1* and *bJogCtrlInputSel2* (see Truth table for activating the pre-switch off). |
| bJogSpeed4<br>bJogSpeed8<br>BOOL | Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint<br>• A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via these selection inputs.<br>• The selection inputs are binary coded.<br>• For a detailed functional description see the L_NSet FB. |
| bJogRamp1 ... bJogRamp8<br>BOOL | Selection inputs for alternative acceleration/deceleration times for the main setpoint<br>• The four selection inputs are binary coded, therefore 15 alternative acceleration/deceleration times can be selected.<br>• For main setpoint *nMainSetValue_a*, the set acceleration time (C00012) and deceleration time (C00013) are active in the case of the binary coded selection "0" (all inputs = FALSE or not assigned).<br>• Alternative acceleration times are selected in C00101/1...15.<br>• The selection of the alternative deceleration times is carried out in C00103/1...15.<br>• For a detailed functional description see the L_NSet FB. |
| **MCK basic functions** | |
| bMBRKRelease<br>BOOL | Holding brake control: Release/apply brake<br>• In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off").<br>**FALSE**: Apply brake.<br>• During automatic operation, the internal brake logic controls of the brake.<br>**TRUE**: Release brake manually (forced release).<br>• **Note!**<br>The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If the brake control has inhibited the controller, this inhibit is deactivated again.<br>• In semi-automatic operation, the brake is released including feedforward control. |

| Identifier<br>Data type | Information/possible settings |
|---|---|
| **GP: GeneralPurpose**<br>The following inputs are interconnected with logic/arithmetic functions on application level for free usage.<br>"GeneralPurpose" functions | |
| nGPAnalogSwitchIn1_a<br>nGPAnalogSwitchIn2_a<br>INT | Analog switch: Input signals<br>• The input signal selected via the selection input *bGPAnalogSwitchSet* is output at output *nGPAnalogSwitchOut_a*. |
| bGPAnalogSwitchSet<br>BOOL | Analog switch: Selection input<br>FALSE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn1_a*<br>TRUE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn2_a* |
| nGPArithmetikIn1_a<br>nGPArithmetikIn2_a<br>INT | Arithmetic: Input signals<br>• The arithmetic function is selected in C00338.<br>• The result is output at output *nGPArithmetikOut_a*. |
| nGPMulDivIn_a<br>INT | Multiplication/Division: Input signal<br>• The factor for the multiplication can be set in C00699/1 (numerator) and C00699/2 (denominator).<br>• The result is output at output *nGPMulDivOut_a*. |
| bGPDigitalDelayIn<br>BOOL | Binary delay element: Input signal<br>• The on-delay can be set in C00720/1.<br>• The off-delay can be set in C00720/2.<br>• The time-delayed input signal is output at output *bGPDigitalDelayOut*. |
| bGPLogicIn1<br>bGPLogicIn2<br>bGPLogicIn3<br>BOOL | Binary logic: Input signals<br>• The logic operation is selected in C00820.<br>• The result is output at output *bGPLogicOut*. |
| nGPCompareIn1_a<br>nGPCompareIn2_a<br>INT | Analog comparison: Input signals<br>• The comparison operation is selected in C00680.<br>• Hysteresis and window size can be set in C00680 and C00682.<br>• If the comparison statement is true, the output *bGPCompareOut* will be set to TRUE. |
| bGPDFlipFlop_InD<br>bGPDFlipFlop_InClk<br>bGPDFlipFlop_InClr<br>BOOL | D-FlipFlop: Input signals<br>• Data, clock and reset input |
| **Free inputs**<br>The following inputs can freely be interconnected on the application level.<br>The signals can be transferred from the I/O level to the application level via these inputs. | |
| bFreeIn1 ... bFreeIn8<br>BOOL | Free inputs for digital signals |
| wFreeIn1 ... wFreeIn4<br>WORD | Free inputs for 16-bit signals |
| dnFreeIn1_p ... dnFreeIn2_p<br>DINT | Free inputs for 32-bit signals |

## Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wDriveControlStatus<br>WORD | Status word of the controller (based on DSP-402)<br>• The status word contains information on the currents status of the drive controller.<br>• See the "wDeviceStatusWord status word" subchapter of the chapter on device control for a detailed description of the bit assignment. |
| wStateDetermFailNoLow<br>WORD | Display of the status determining error (LOW word) |

| Identifier | Data type | Value/meaning | |
|---|---|---|---|
| wStateDetermFailNoHigh | WORD | Display of the status determining error (HIGH word) | |
| bDriveFail | BOOL | TRUE | Drive controller in error status.<br>• "Fault" device status is active. |
| bWarningActive | BOOL | TRUE | A monitoring in the drive controller, for which the error response "Warning" or "WarningLocked" has been parameterised, responded. |
| bSafeTorqueOff | BOOL | TRUE | Safe torque off.<br>• "SafeTorqueOff" device status is active. |
| bDriveReady | BOOL | TRUE | Controller is ready for operation.<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the controller is still inhibited by the user (controller inhibit). |
| bCInhActive | BOOL | TRUE | Controller inhibit is active. |
| bImpIsActive | BOOL | TRUE | Pulse inhibit is active. |
| bQSPIsActive | BOOL | TRUE | Quick stop is active. |
| bSpeedCcw | BOOL | Current direction of rotation | |
| | | FALSE | Direction of rotation to the right (Cw) |
| | | TRUE | Direction of rotation to the left (Ccw) |
| bSpeedActCompare | BOOL | Result of the speed comparison | |
| | | TRUE | During open-loop operation:<br>Speed setpoint < Comparison value (C00024) |
| | | | During closed-loop operation:<br>Actual speed value < Comparison value (C00024) |
| bImaxActive | BOOL | "Current setpoint inside the limitation" status signal | |
| | | TRUE | The current setpoint is internally limited (the drive controller operates at the maximum current limit). |
| bSpeedSetReached | BOOL | Status signal "setpoint = 0" | |
| | | TRUE | Speed setpoint from the ramp function generator = 0 |
| bSpeedActEqSet | BOOL | TRUE | Actual speed value = speed setpoint |
| nMotorCurrent_a | INT | Current stator current/effective motor current<br>• Scaling: 16384 ≡ 100 % $I_{max\_mot}$ (C00022) | |
| nMotorSpeedSet_a | INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorSpeedAct_a | INT | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nMotorTorqueAct_a | INT | Actual torque<br>• In the "VFC (+encoder)" operating mode of the motor control, this value is determined from the current motor current and corresponds to the actual torque only by approximation.<br>• Scaling: 16384 ≡ 100 % $M_{max}$ (C00057) | |
| nDCVoltage_a | INT | Actual DC-bus voltage<br>• Scaling: 16384 ≡ 1000 V | |
| nMotorVoltage_a | INT | Current motor voltage/inverter output voltage<br>• Scaling: 16384 ≡ 1000 V | |

| Identifier (Data type) | Value/meaning | |
|---|---|---|
| **MCK basic functions** | | |
| bBrakeReleaseOut<br>BOOL | Holding brake control: Trigger signal for the holding brake control switching element via a digital output<br>• Use bit 0 in C02582 to activate inverted switching element triggering. | |
| | FALSE | Apply brake. |
| | TRUE | Release brake. |
| bBrakeReleased<br>BOOL | Holding brake control: "Brake released" considering the brake release time<br>• When the holding brake is triggered to close, *bBrakeReleased* is immediately set to FALSE even if the brake closing time has not yet elapsed! | |
| | TRUE | Brake released (when the brake release time has elapsed). |
| **GP: GeneralPurpose**<br>The following outputs are interconnected with logic/arithmetic functions on application level for free usage.<br>▶ "GeneralPurpose" functions | | |
| nGPAnalogSwitchInOut_a<br>INT | Analog switch: Output signal | |
| nGPArithmetikOut_a<br>INT | Arithmetic: Output signal | |
| nGPMulDivOut_a<br>INT | Multiplication/Division: Output signal | |
| bGPDigitalDelayOut<br>BOOL | Binary delay element: Output signal | |
| bGPLogicOut<br>BOOL | Binary logic: Output signal | |
| bGPCompareOut<br>BOOL | Analog comparison: Output signal | |
| bGPSignalOut1<br>...<br>bGPSignalOut4<br>BOOL | Binary signal monitor: Output signals<br>• The signal sources to be output are selected in C00411/1...4.<br>• A bit coded inversion of the output signals can be parameterised in C00412. | |
| nGPSignalOut1_a<br>...<br>nGPSignalOut4_a<br>BOOL | Analog signal monitor: Output signals<br>• The signal sources to be output are selected in C00410/1...4.<br>• Gain and offset for each output signal can be parameterised in C00413/1...8. | |
| bGPDFlipFlop_Out<br>BOOL | D-FlipFlop: Output signal | |
| bGPDFlipFlop_NegOut<br>BOOL | D-FlipFlop: Negated output signal | |
| **Free outputs**<br>The following outputs can freely be interconnected on the application level.<br>The signals from the application level can be transferred to the I/O level via these outputs. | | |
| bFreeOut1 ... bFreeOut8<br>BOOL | Free outputs for digital signals | |
| wFreeOut1 ... wFreeOut4<br>WORD | Free outputs for 16-bit signals | |
| dnFreeOut1_p<br>dnFreeOut2_p<br>WORD | Free outputs for 32-bit signals | |

### 8.6.2.1 Truth table for activating the pre-switch off

| Input | | Function | Response in the setpoint generator (FB L_NSet) |
|---|---|---|---|
| **bJogCtrl InputSel1** | **bJogCtrl InputSel2** | | |
| FALSE | FALSE | Pre-switch off inactive | No response<br>• The input signal *bJogCtrlRfgIn* is output directly at output *bRfgOut*.<br>• The input signals *bJogCtrlJog1* and *bJogCtrlJog2* are passed through 1:1 to the downstream FB L_NSet for the selection of fixed setpoints. |
| TRUE | FALSE | The *bJogCtrlSlowDown1* and *bJogCtrlStop1* inputs are evaluated. | **Pre-switch off can be activated**<br>• If the slowdown function is activated via the selected *bJogCtrlSlowDown* input, fixed setpoint 2 is activated in the setpoint generator.<br>• If the stop function is activated via the selected *bJogCtrlStop* input, the setpoint generator is deactivated. |
| FALSE | TRUE | The *bJogCtrlSlowDown2* and *bJogCtrlStop2* inputs are evaluated. | |
| TRUE | TRUE | The inputs *bSJogCtrllowDown3* and *bJogCtrlStop3* are evaluated. | |

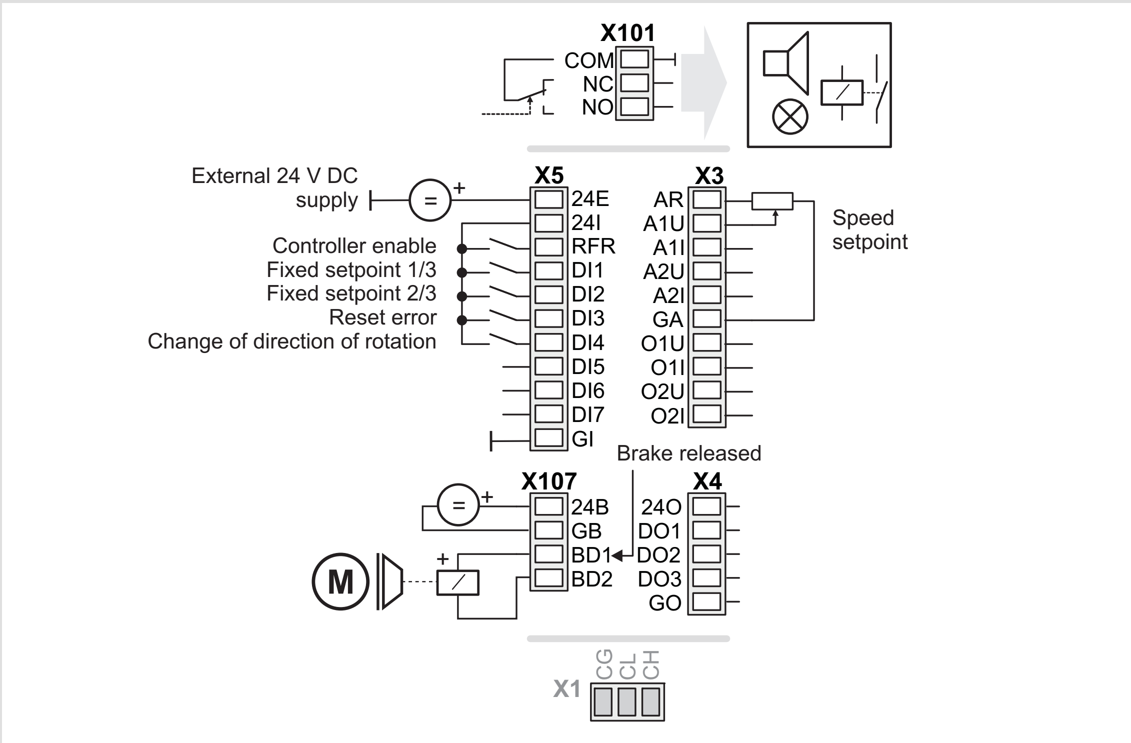
[8-1] Truth table for activating the pre-switch off

### 8.6.3 Terminal assignment of the control modes

The following comparison provides information about which inputs/outputs of the application block **LA_SwitchPos** are interconnected to the digital and analog input/output terminals of the drive controller in the different control modes.

| | **Control mode (C00007)** | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 10: Terminals 0 | 12: Terminals 2 | 14: Terminals 11 | 16: Terminal 16 | 20: Keypad | 21: PC | 30: CAN | 40: MCI |
| **Digital input terminals** | | | | | | | | |
| **X5/RFR** | **Controller enable** | **Controller enable / Reset of error message**<br>bFailReset | | | | | | |
| **X5/DI1** | **Fixed setpoint 1/3**<br>bJogCtrlJog1 | Stop function 1<br>bJogCtrlStop1 | | **Fixed setpoint 1/3**<br>bJogCtrlJog1 | - | - | **Stop function 1**<br>bJogCtrlStop1 | |
| **X5/DI2** | **Fixed setpoint 2/3**<br>bJogCtrlJog2 | Stop function 2<br>bJogCtrlStop2 | **Selection: Pre-switch off 1**<br>bJogCtrlSlowDown 1 | **Fixed setpoint 2/3**<br>bJogCtrlJog2 | - | - | **Selection: Pre-switch off 1**<br>bJogCtrlSlowDown 1 | |
| **X5/DI3** | **Reset error messages**<br>bFailReset | **CW rotation quick stop**<br>bRLQCw<br>Selection: Switch-off position 1<br>bJogCtrlInputSel1 | | **CW rotation quick stop**<br>bRLQCw | - | - | **Stop function 2**<br>bJogCtrlStop2 | |
| **X5/DI4** | **Change of direction of rotation**<br>bSetSpeedCcw | **CCW rotation quick stop**<br>bRLQCcw<br>Selection: Switch-off position 2<br>bJogCtrlInputSel2 | | **CCW rotation quick stop**<br>bRLQCcw | - | - | **Selection: Pre-switch off 2**<br>bJogCtrlSlowDown 2 | |
| **X5/DI5** | - | - | Stop function 2<br>bJogCtrlStop2 | - | - | - | **Stop function 3**<br>bJogCtrlStop3 | |
| **X5/DI6** | - | - | **Selection: Pre-switch off 2**<br>bJogCtrlSlowDown 2 | - | - | - | **Selection: Pre-switch off 3**<br>bJogCtrlSlowDown 3 | |
| **X5/DI7** | - | - | - | - | - | - | - | - |
| **Analog input terminals** | | | | | | | | |
| **X3/A1U, A1I** | **Main speed setpoint**<br>nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) | | | | - | - | **Additional speed setpoint**<br>nAuxSetValue_a<br>10 V ≡ 100 % reference speed (C00011) | |
| **X3/A2U, A2I** | - | - | - | - | - | - | - | - |
| **Digital output terminals** | | | | | | | | |
| **X4/DO1 ... DO3** | - | - | - | - | - | - | - | - |
| **X107/BD1, BD2** | **Control of the holding brake**<br>bMBrakeReleaseOut | | | | | | | |
| **X101/COM, NO** | - | - | - | - | - | - | - | - |
| **Analog output terminals** | | | | | | | | |
| **X3/O1U, O1I** | - | - | - | - | - | - | - | - |
| **X3/O2U, O2I** | | | | | | | | |

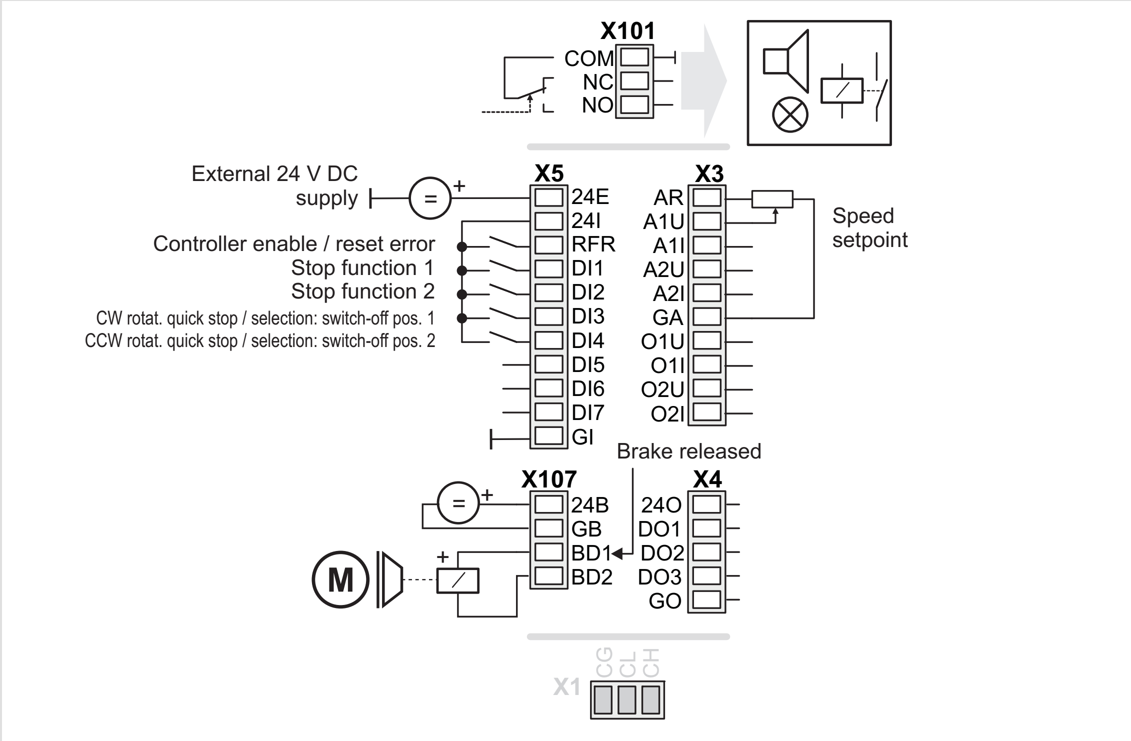
### 8.6.3.1 Terminals 0




| Connection | Assignment |
|---|---|
| X101/NC-NO | - |
| | |
| X5/RFR | - |
| X5/DI1 | LA_SwitchPos.bJogCtrlJog1 |
| X5/DI2 | LA_SwitchPos.bJogCtrlJog2 |
| X5/DI3 | LA_SwitchPos.bFailReset |
| X5/DI4 | LA_SwitchPos.bSetSpeedCcw |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_SwitchPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | - |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | |
| X4/DO1 | - |
| X4/DO2 | - |
| X4/DO3 | - |

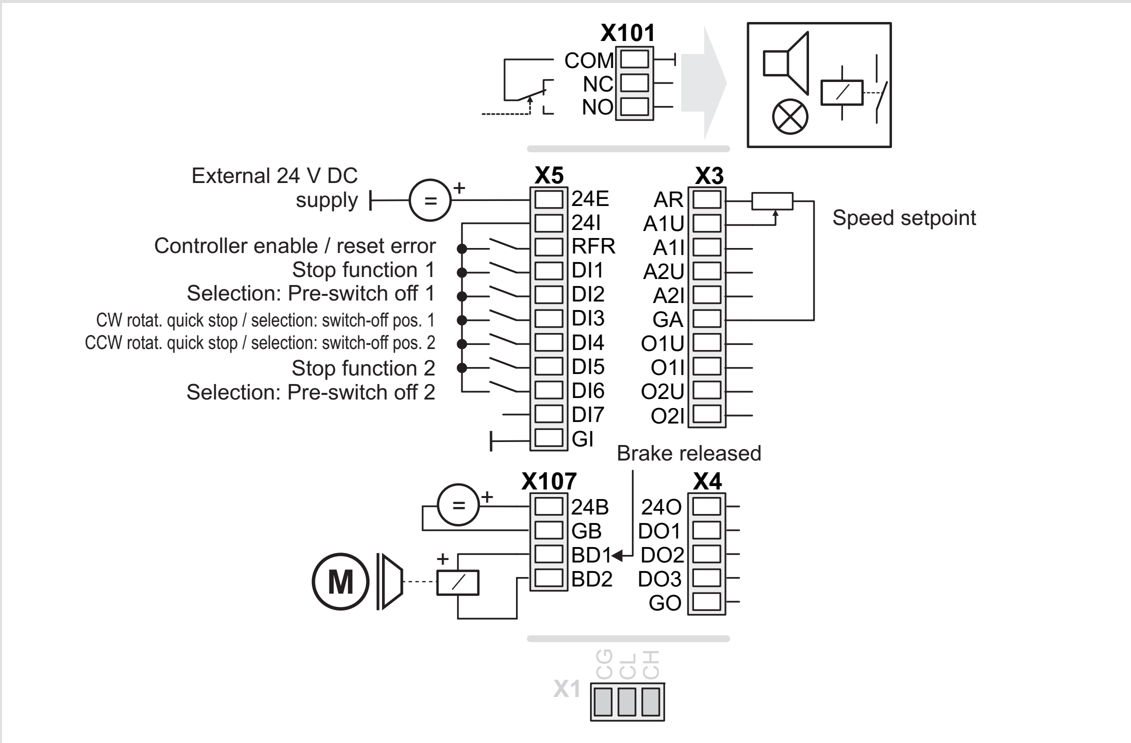
### 8.6.3.2 Terminals 2




| Connection | Assignment |
|---|---|
| X101/NC-NO | - |
| | |
| X5/RFR | LA_SwitchPos: bFailReset |
| X5/DI1 | LA_SwitchPos: bJogCtrlStop1 |
| X5/DI2 | LA_SwitchPos: bJogCtrlStop2 |
| X5/DI3 | LA_SwitchPos: bRLQCw |
| | LA_SwitchPos: bJogCtrlInputSel1 |
| X5/DI4 | LA_SwitchPos: bRLQCcw |
| | LA_SwitchPos: bJogCtrlInputSel2 |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut |
| X107/BD2 | - |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_SwitchPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | - |
| X3/O1I | - |
| X3/O2U | - |
| | |
| X4/DO1 | - |
| X4/DO2 | - |
| X4/DO3 | - |
| | |

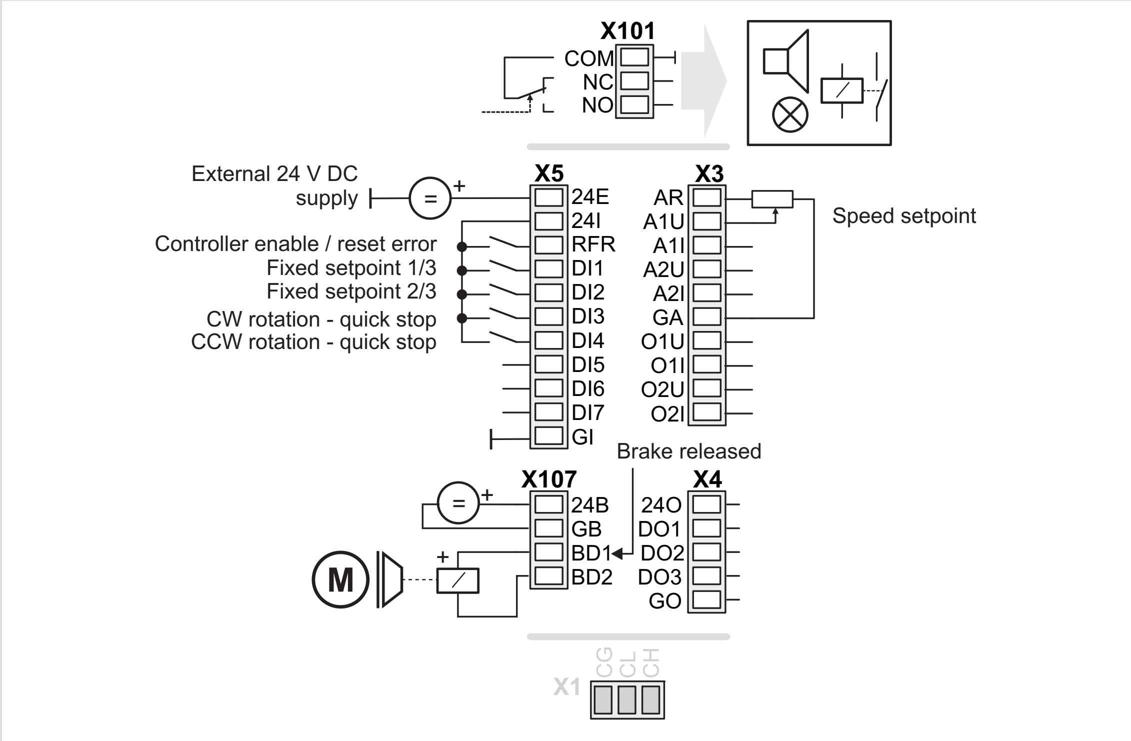
## 8.6.3.3 Terminals 11




| Connection | Assignment |
|---|---|
| X101/NC-NO | - |
| | |
| X5/RFR | LA_SwitchPos: bFailReset |
| X5/DI1 | LA_SwitchPos: bJogCtrlStop1 |
| X5/DI2 | LA_SwitchPos: bJogCtrlSlowDown1 |
| X5/DI3 | LA_SwitchPos: bRLQCw |
| | LA_SwitchPos: bJogCtrlInputSel1 |
| X5/DI4 | LA_SwitchPos: bRLQCcw |
| | LA_SwitchPos: bJogCtrlInputSel2 |
| X5/DI5 | LA_SwitchPos: bJogCtrlStop2 |
| X5/DI6 | LA_SwitchPos: bJogCtrlSlowDown2 |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut |
| X107/BD2 | - |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_SwitchPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | - |
| X3/O1I | - |
| X3/O2U | - |
| | |
| X4/DO1 | - |
| X4/DO2 | - |
| X4/DO3 | - |
| | |

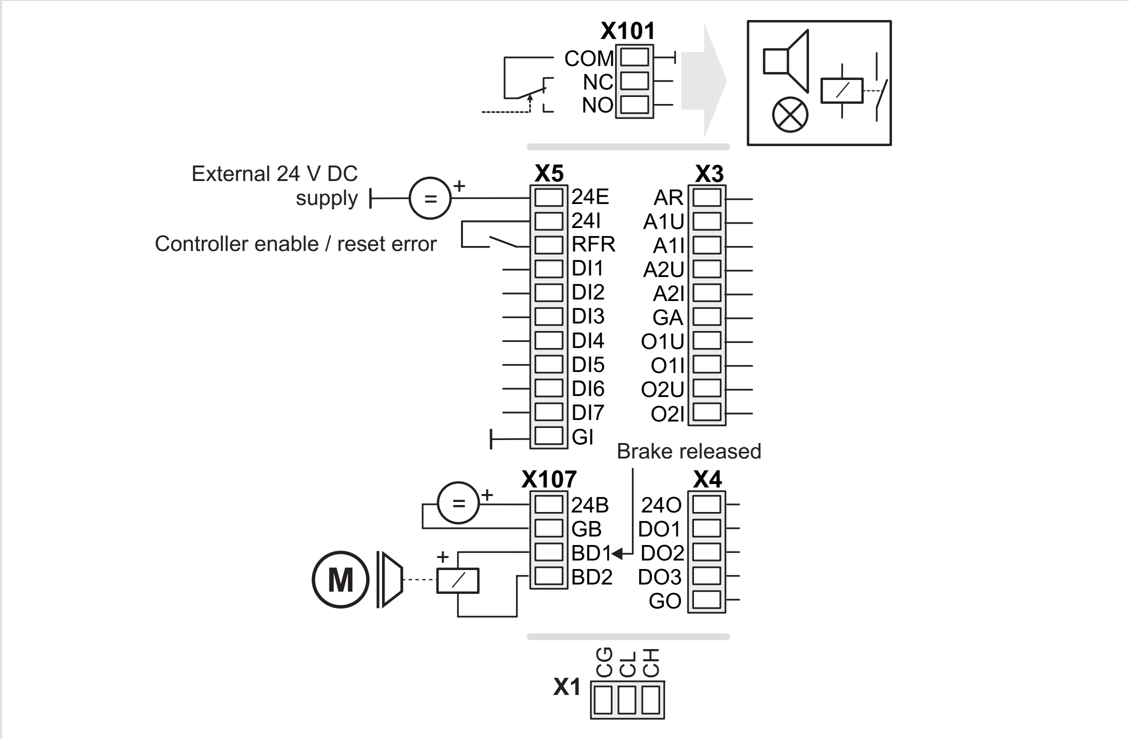
### 8.6.3.4 Terminal 16




| Connection | Assignment |
|---|---|
| X101/NC-NO | - |
| | |
| X5/RFR | LA_SwitchPos: bFailReset |
| X5/DI1 | LA_SwitchPos: bJogCtrlJog1 |
| X5/DI2 | LA_SwitchPos: bJogCtrlJog2 |
| X5/DI3 | LA_SwitchPos: bRLQCw |
| X5/DI4 | LA_SwitchPos: bRLQCcw |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_SwitchPos.nMainSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | - |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | |
| X4/DO1 | - |
| X4/DO2 | - |
| X4/DO3 | - |

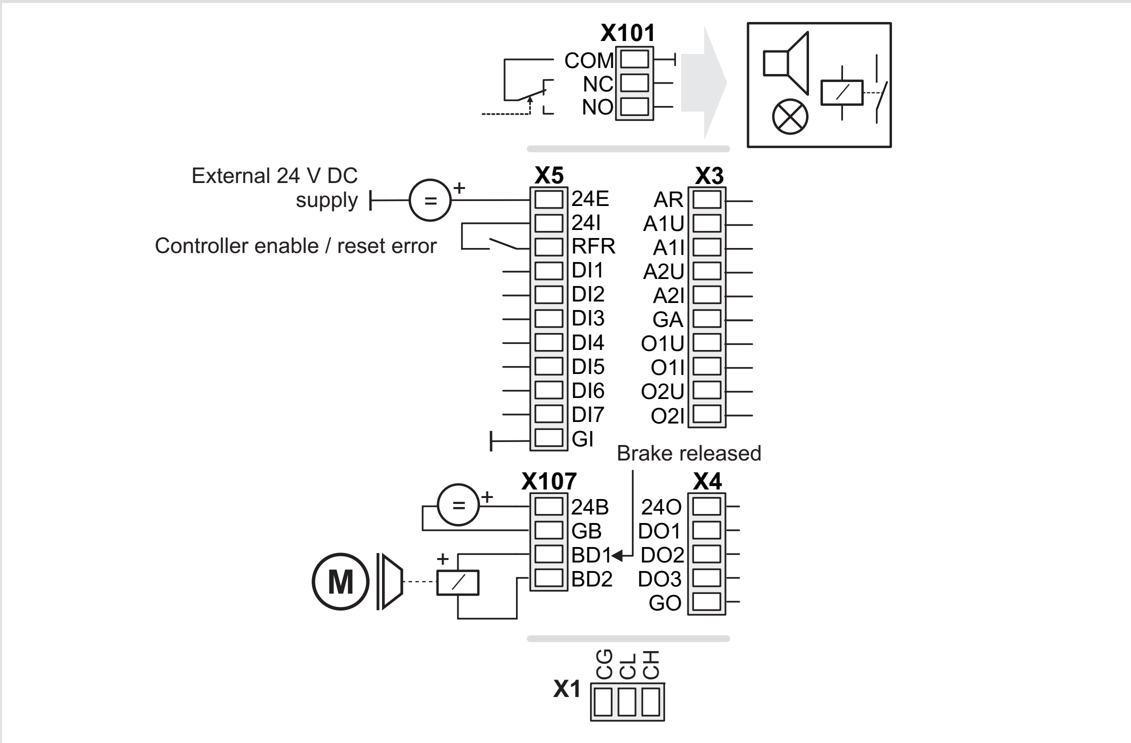
### 8.6.3.5 Keypad




| Connection | Assignment |
|---|---|
| X101/NC-NO | - |
| | |
| X5/RFR | - |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | - |
| X5/DI4 | - |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | - |
| X3/A1I | - |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | - |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | |
| X4/DO1 | - |
| X4/DO2 | - |
| X4/DO3 | - |

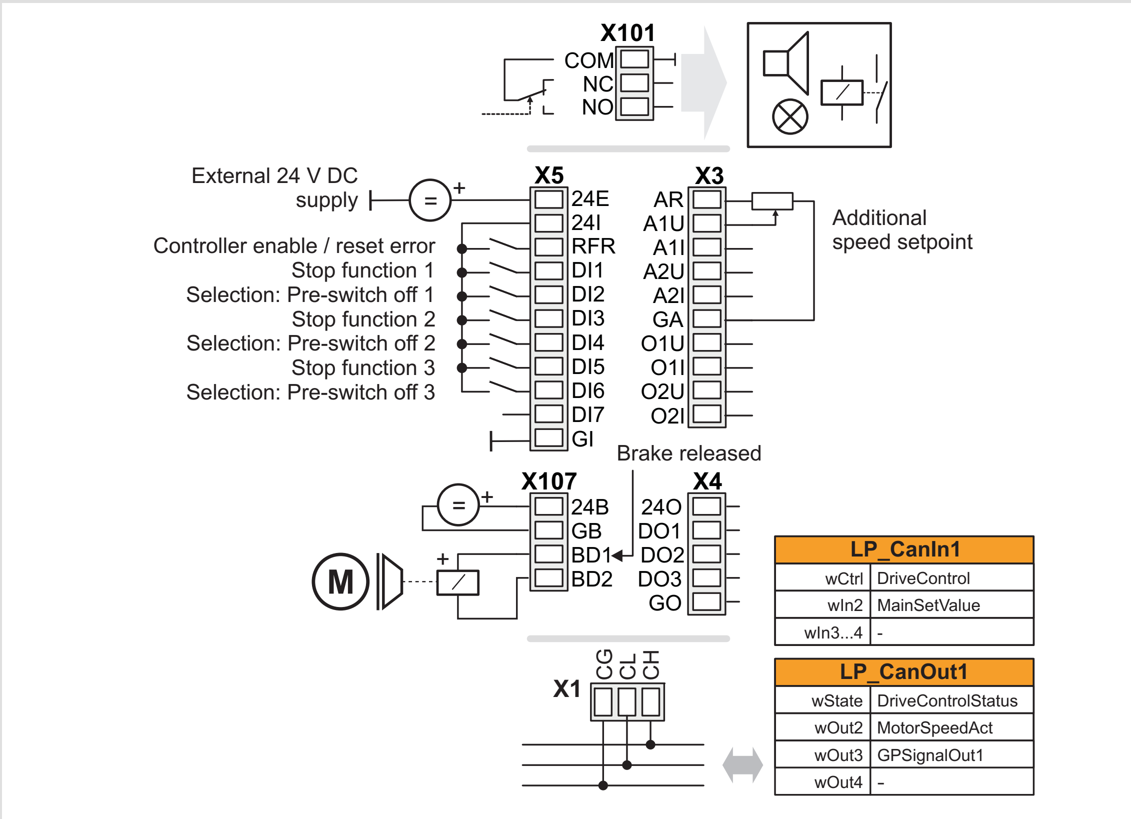
### 8.6.3.6 PC




| Connection | Assignment |
|---|---|
| X101/NC-NO | - |
| | |
| X5/RFR | - |
| X5/DI1 | - |
| X5/DI2 | - |
| X5/DI3 | - |
| X5/DI4 | - |
| X5/DI5 | - |
| X5/DI6 | - |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | - |
| X3/A1I | - |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | - |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | |
| X4/DO1 | - |
| X4/DO2 | - |
| X4/DO3 | - |

### 8.6.3.7 CAN




| Connection | Assignment |
|---|---|
| X101/NC-NO | - |
| | |
| X5/RFR | LA_SwitchPos: bFailReset |
| X5/DI1 | LA_SwitchPos: bJogCtrlStop1 |
| X5/DI2 | LA_SwitchPos: bJogCtrlSlowDown1 |
| X5/DI3 | LA_SwitchPos: bJogCtrlStop2 |
| X5/DI4 | LA_SwitchPos: bJogCtrlSlowDown2 |
| X5/DI5 | LA_SwitchPos: bJogCtrlStop3 |
| X5/DI6 | LA_SwitchPos: bJogCtrlSlowDown3 |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

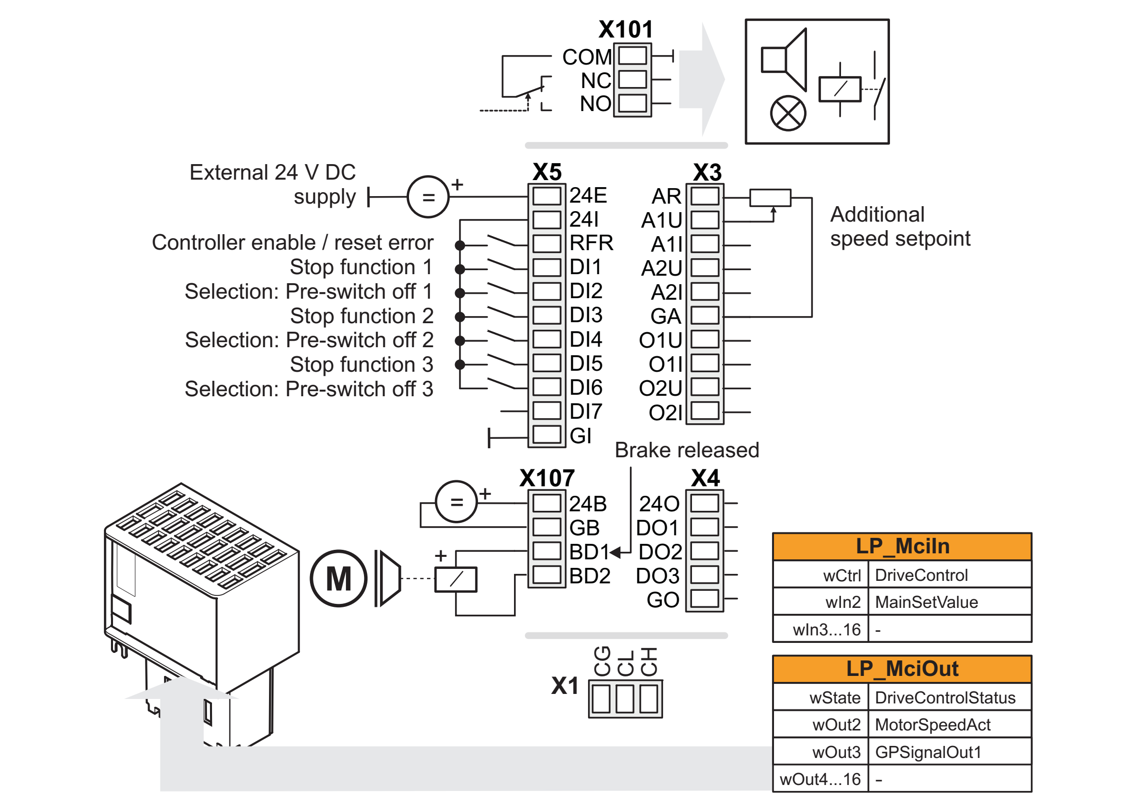
| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_SwitchPos.nAuxSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | - |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | |
| X4/DO1 | - |
| X4/DO2 | - |
| X4/DO3 | - |

▶ Process data assignment for fieldbus communication (📖 460)

**Note!**

You must set the setpoint arithmetic in C00190 to "1: NOut = NSet + NAdd" so that the additional speed setpoint selected via the analog input A1U has an additive effect.

## 8.6.3.8 MCI




| Connection | Assignment |
|---|---|
| X101/NC-NO | - |
| | |
| X5/RFR | LA_SwitchPos: bFailReset |
| X5/DI1 | LA_SwitchPos: bJogCtrlStop1 |
| X5/DI2 | LA_SwitchPos: bJogCtrlSlowDown1 |
| X5/DI3 | LA_SwitchPos: bJogCtrlStop2 |
| X5/DI4 | LA_SwitchPos: bJogCtrlSlowDown2 |
| X5/DI5 | LA_SwitchPos: bJogCtrlStop3 |
| X5/DI6 | LA_SwitchPos: bJogCtrlSlowDown3 |
| X5/DI7 | - |
| | |
| X107/BD1 | LA_SwitchPos.bMBrakeReleaseOut |
| X107/BD2 | - |
| | |

| Connection | Assignment |
|---|---|
| | |
| X3/A1U | LA_SwitchPos.nAuxSetValue_a<br>10 V ≡ 100 % reference speed (C00011) |
| X3/A1I | |
| X3/A2U | - |
| X3/A2I | - |
| X3/O1U | - |
| X3/O1I | - |
| X3/O2U | - |
| X3/O2I | - |
| | |
| X4/DO1 | - |
| X4/DO2 | - |
| X4/DO3 | - |

▶ Process data assignment for fieldbus communication (📖 460)

**Note!**

You must set the setpoint arithmetic in C00190 to "1: NOut = NSet + NAdd" so that the additional speed setpoint selected via the analog input A1U has an additive effect.

### 8.6.4 Process data assignment for fieldbus communication

The fieldbus communication is connected (preconfigured) to the previously selected technology application by selecting the corresponding control mode in C00007:

- "30: CAN" for the connection to the system bus (CAN)
- "40: MCI" for the connection to a plugged-on communication module (e.g. PROFIBUS)

The assignment of the process data words does not depend on the applied bus system but exclusively on the application:

| Input words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControl | Control word<br>• See table below for bit assignment. |
| Word 2 | MainSetValue | Speed setpoint<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 3 | - | Not preconfigured |
| Word 4 | - | Not preconfigured |
| Word 5 ... 16 | - | Not preconfigured<br>• Only available for control mode "40: MCI". |

| Control word | Name | Function |
|---|---|---|
| Bit 0 | SwitchOn | 1 ≡ Change to the "SwitchedOn" device status<br>• This bit must be set in the CAN/MCI control word to ensure that the device changes to the "SwitchedOn" device status after mains connection without the need for a master control specifying this bit via fieldbus.<br>• If control via a bus system is not wanted (e.g. in the case of control via terminals), the *wDriveCtrl* output signal of the LS_ParFix system block can be connected to the control word inputs. |
| Bit 1 | DisableVoltage | 1 ≡ Inhibit inverter control (IMP - pulse inhibit) |
| Bit 2 | SetQuickStop | 1 ≡ Activate quick stop (QSP).<br>▸ Activate/Deactivate quick stop (📖 98) |
| Bit 3 | EnableOperation | 1 ≡ Enable controller (RFR)<br>• If control via terminals is performed, this bit must be set both in the CAN control word and in the MCI control word. Otherwise, the controller is inhibited.<br>▸ Enable/Inhibit controller (📖 97) |
| Bit 4 | ModeSpecific_1 | Reserved (currently not assigned) |
| Bit 5 | JogCtrlInputSel1 | Binary coded selection of the switch-off position 1 ... 3<br>• Activation of the signal pairs *bJogCtrlSlowDown1/bJogCtrlStop1*, *bJogCtrlSlowDown2/bJogCtrlStop2* or *bJogCtrlSlowDown3/bJogCtrlStop3* according to the Truth table for activating the pre-switch off. |
| Bit 6 | JogCtrlInputSel2 | |
| Bit 7 | ResetFault | 1 ≡ Reset fault (trip reset)<br>• Acknowledge fault message (if the error cause has been eliminated).<br>▸ Reset error (📖 99) |
| Bit 8 | bJogCtrlRfgIn | Ramping down of the setpoint generator in the downstream L_NSet FB according to the Truth table for activating the pre-switch off |
| Bit 9 | reserved_1 | Reserved (currently not assigned) |
| Bit 10 | reserved_2 | |

| Control word | Name | Function |
|---|---|---|
| Bit 11 | MBrkRelease | Holding brake control:<br>0 ≡ Apply brake<br>1 ≡ Release brake<br>• In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). |
| Bit 12 | JogCtrlJog1 | Binary coded selection of the fixed setpoints (JOG setpoints) |
| Bit 13 | JogCtrlJog2 | |
| Bit 14 | SetFail | 1 ≡ Set error (trip set) |
| Bit 15 | SetSpeedCcw | 0 ≡ Direction of rotation to the right (Cw)<br>1 ≡ Direction of rotation to the left (Ccw) |

| Output words | Name | Assignment |
|---|---|---|
| Word 1 | DriveControlStatus | Status word<br>• See table below for bit assignment. |
| Word 2 | MotorSpeedAct | Actual speed value<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) |
| Word 3 | GPSignalOut1 | Analog signal monitor: Output signal 1<br>• The selection of the signal source to output is executed in C00410/1.<br>• Gain and offset for the output signal can be parameterised in C00413/1 and C00413/2.<br>• For a detailed functional description see the L_SignalMonitor_a FB. |
| Word 4 | - | Not preconfigured |
| Word 5 ... 16 | - | Not preconfigured<br>• Only available for control mode "40: MCI". |

| Status word | Name | Status |
|---|---|---|
| Bit 0 | DriveFail | 1 ≡ Drive controller in error status<br>• "Fault" device status is active. |
| Bit 1 | PowerDisabled | 1 ≡ Inverter control inhibited (pulse inhibit is active) |
| Bit 2 | DriveReady | 1 ≡ Drive controller is ready for operation<br>• "SwitchedOn" device status is active.<br>• The drive is in this device status if the DC bus voltage is applied and the controller is still inhibited by the user (controller inhibit). |
| Bit 3 | SpeedCcw | 0 ≡ Direction of rotation to the right (Cw)<br>1 ≡ Direction of rotation to the left (Ccw) |
| Bit 4 | QspIsActive | 1 ≡ Quick stop is active |
| Bit 5 | BrakeReleased | 1 ≡ Brake released (after the brake opening time has elapsed) |
| Bit 6 | ActSpeedIsZero | During open-loop operation:<br>1 ≡ Speed setpoint < Comparison value (C00024)<br>During closed-loop operation:<br>1 ≡ Actual speed value < Comparison value (C00024) |
| Bit 7 | ControllerInhibit | 1 ≡ Controller inhibited (controller inhibit is active) |
| Bit 8 | StatusCodeBit0 | Bit coded display of the active device status<br>▸ Device state machine and device statuses (see table [4-1]) |
| Bit 9 | StatusCodeBit1 | |
| Bit 10 | StatusCodeBit2 | |
| Bit 11 | StatusCodeBit3 | |
| Bit 12 | Warning | 1 ≡ A warning is indicated |

| Status word | Name | Status |
|---|---|---|
| Bit 13 | Trouble | 1 ≡ Controller is in the "Trouble" device status<br>• E.g. if an overvoltage has occurred. |
| Bit 14 | JogCtrlInputSel1 | Binary coded selection of the switch-off position 1 ... 3<br>• Bits 5 and 6 of the control word. |
| Bit 15 | JogCtrlInputSel2 | |

## 8.6.5 Setting parameters (short overview)

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00011 | Appl.: Reference speed | 1500 | rpm |
| C00012 | Accel. time - main setpoint | 2.000 | s |
| C00013 | Decel. time - main setpoint | 2.000 | s |
| C00105 | Decel. time - quick stop | 2.000 | s |
| C00039/1 | Fixed setpoint 1 | 40.00 | % |
| C00039/2 | Fixed setpoint 2 | 60.00 | % |
| C00039/3 | Fixed setpoint 3 | 80.00 | % |
| C00039/4...15 | Fixed setpoint 4 ... 15 | 0.00 | % |
| C00101/1...15 | Add. accel. time 1 ... 15 | 0.000 | s |
| C00103/1...15 | Add. decel. time 1 ... 15 | 0.000 | s |
| C00105 | Decel. time - quick stop | 2.000 | s |
| C00106 | Auto DCB: Hold time | 0.500 | s |
| C00107 | DCB: Hold time | 999.000 | s |
| C00134 | Ramp rounding - main setpoint | 0: Off | |
| C00182 | S-ramp time PT1 | 20.00 | s |
| C00190 | Setpoint arithmetic | 0: NOut = NSet | |
| C00220 | Accel. time - add. setpoint | 0.000 | s |
| C00221 | Decel. time - add. setpoint | 0.000 | s |
| C00241 | L_NSet_1: Hyst. NSet reached | 0.50 | % |
| C00488/1 | InputSens.SlowDown1 | 0: Level | |
| C00488/2 | InputSens.Stop1 | 0: Level | |
| C00488/3 | InputSens.SlowDown2 | 0: Level | |
| C00488/4 | InputSens.Stop2 | 0: Level | |
| C00488/5 | InputSens.SlowDown3 | 0: Level | |
| C00488/6 | InputSens.Stop3 | 0: Level | |
| C00632/1 | L_NSet_1: Blocking speed 1 max | 0.00 | % |
| C00632/2 | L_NSet_1: Blocking speed 2 max | 0.00 | % |
| C00632/3 | L_NSet_1: Blocking speed 3 max | 0.00 | % |
| C00633/1 | L_NSet_1: Blocking speed 1 min | 0.00 | % |
| C00633/2 | L_NSet_1: Blocking speed 2 min | 0.00 | % |
| C00633/3 | L_NSet_1: Blocking speed 3 min | 0.00 | % |
| C00635 | L_NSet_1: nMaxLimit | 199.99 | % |
| C00636 | L_NSet_1: nMinLimit | -199.99 | % |
| C00670 | L_OffsetGainP_1: Gain | 1.0000 | |
| C00671 | L_OffsetGainP_2: Gain | 1.0000 | |

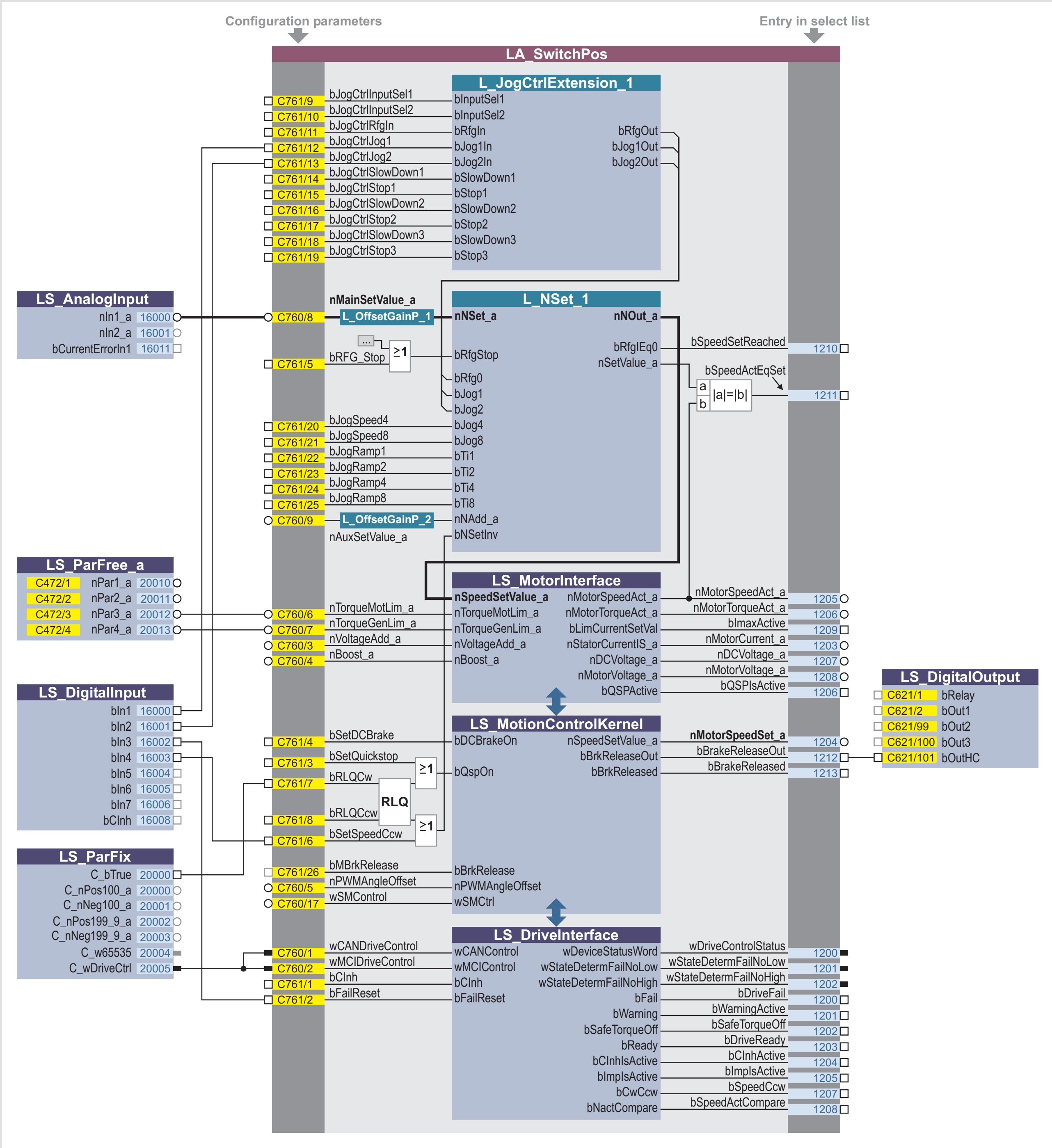
| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00672 | L_OffsetGainP_3: Gain | 1.0000 | |
| C00696 | L_OffsetGainP_1: Offset | 0.00 | % |
| C00697 | L_OffsetGainP_2: Offset | 0.00 | % |
| C00698 | L_OffsetGainP_3: Offset | 0.00 | % |
| C00800 | L_MPot_1: Upper limit | 100.00 | % |
| C00801 | L_MPot_1: Lower limit | -100.00 | % |
| C00802 | L_MPot_1: Accel. time | 10.0 | s |
| C00803 | L_MPot_1: Decel. time | 10.0 | s |
| C00804 | L_MPot_1: Inactive fct. | 0: Retain value | |
| C00805 | L_MPot_1: Init fct. | 0: Load last value | |
| C00806 | L_MPot_1: Use | 0: No | |
| C02610/2 | MCK: Ramp time synchr. setpoint | 2.000 | s |
| C02611/1 | MCK: Pos. max. speed | 199.99 | % |
| C02611/2 | MCK: Pos. min. speed | 0.00 | % |
| C02611/3 | MCK: Neg. min. speed | 0.00 | % |
| C02611/4 | MCK: Neg. max. speed | 199.99 | % |

**Related topics:**

- "GeneralPurpose" functions (🕮 467)

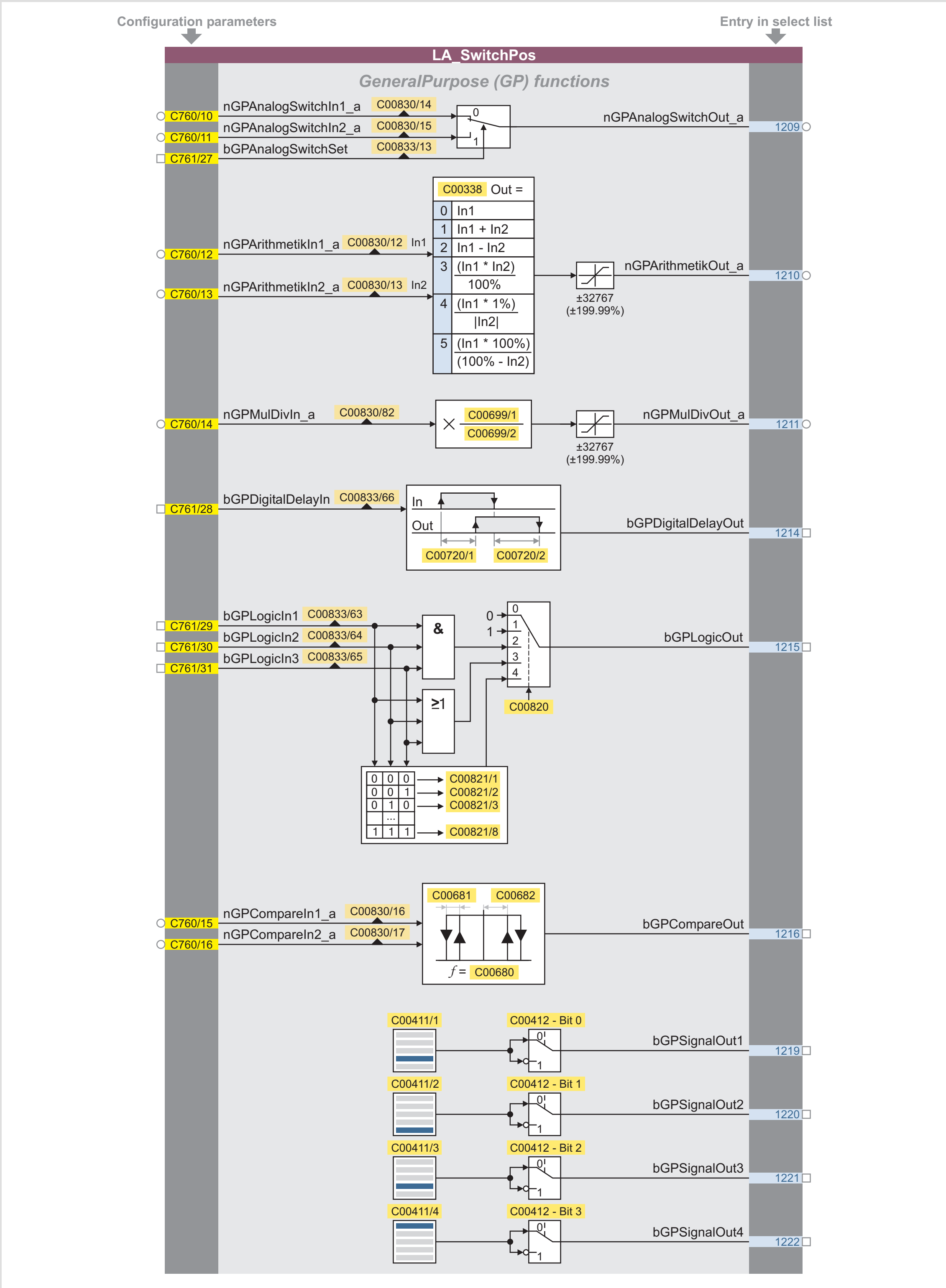
## 8.6.6 Configuration parameters

If required, the subcodes of C00760 and C00761 serve to change the pre-configured assignment of the application inputs:

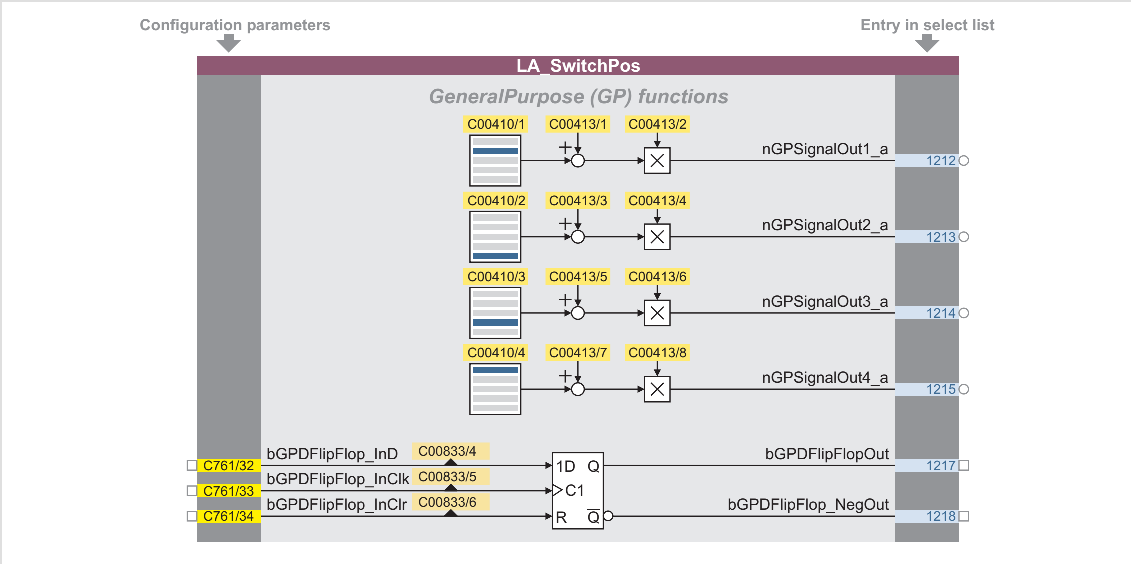



[8-1] Pre-assignment of the "Switch-off positioning" application in the "Terminals 0" control mode

### Configuration parameters for "GeneralPurpose" functions



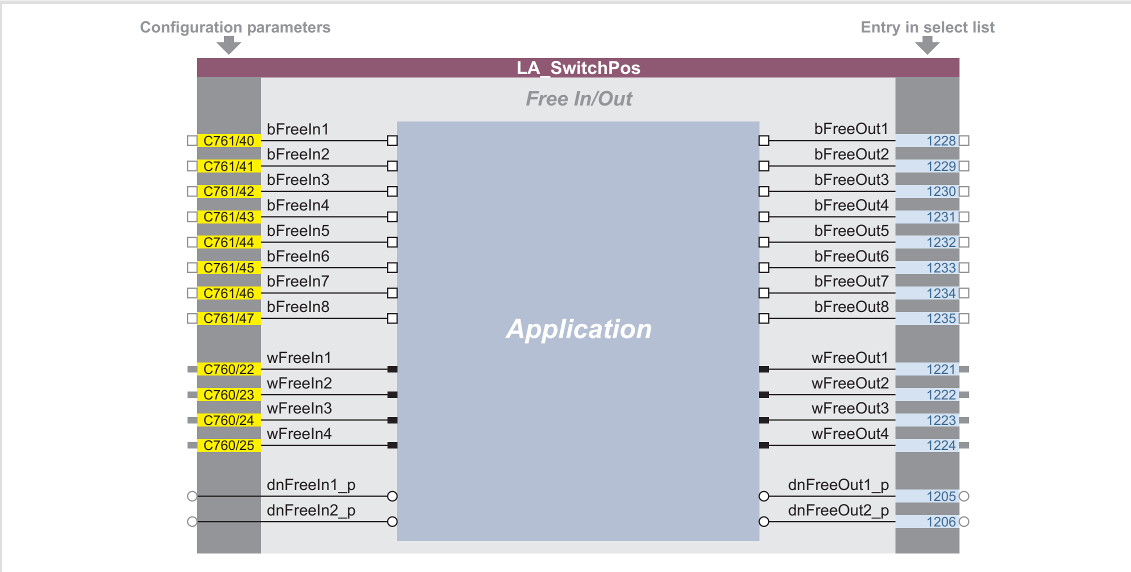

[8-2] "GeneralPurpose" functions




[8-3] "GeneralPurpose" functions (continuation)

## Free inputs and outputs

These inputs can be freely interconnected in the application level. They can be used to transfer signals from the I/O level to the application level and vice versa.




[8-4] Free inputs/outputs

### Related topics:

- User-defined terminal assignment (📖 370)
- "GeneralPurpose" functions (📖 467)

## 8.7 "GeneralPurpose" functions

Each technology application provides different free logic and arithmetic functions, so-called "GeneralPurpose" functions.

For the interconnection of these functions, the application block features inputs and outputs on the I/O level, which are linked to the logic/arithmetic function.

**Note!**

In the Lenze setting, the connectors for the "GeneralPurpose" functions are hidden in the function block editor.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

**Tip!**

The inputs of the "GeneralPurpose" functions can also be linked to other output signals via the configuration parameters of the technology application.

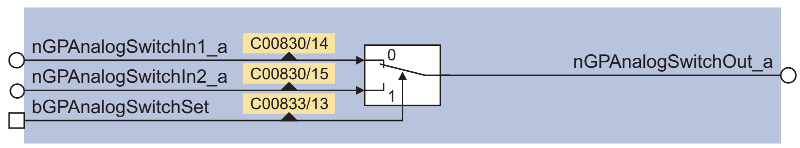
On the other hand, the outputs of the "GeneralPurpose" functions can be selected in the configuration parameters of other inputs.

**Related topics:**

- User-defined terminal assignment (📖 370)
- TA "Actuating drive speed": Configuration parameters (📖 407)
- TA "Table positioning": Configuration parameters (📖 436)
- TA "Switch-off positioning": Configuration parameters (📖 464)

### 8.7.1 Analog switch

This function switches between two analog input signals. The switch-over is controlled by a boolean input signal.

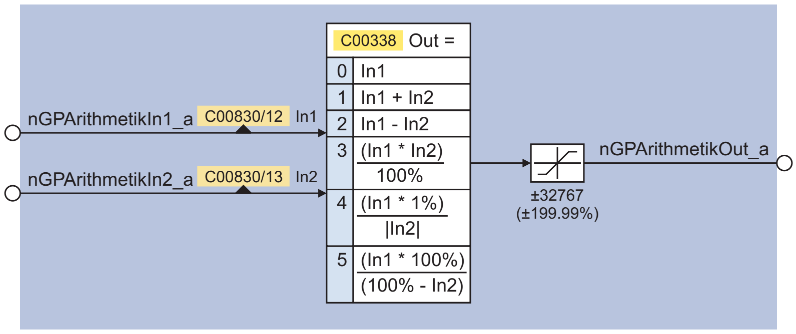



[8-5] GeneralPurpose function "Analog switch"

▶ For a detailed functional description see FB L_AnalogSwitch.

## 8.7.2 Arithmetic

This function links two analog signals arithmetically. The arithmetic function can be parameterised.



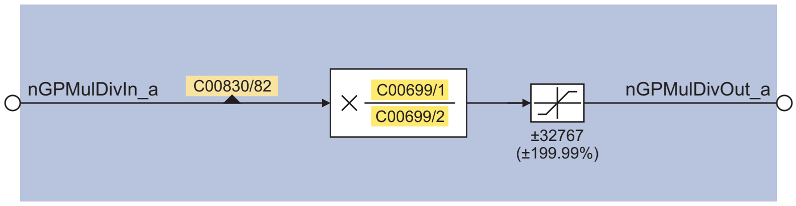

[8-6] GeneralPurpose function "Arithmetic"

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00338 | L_Arithmetik_1: Function | 0: nOut_a = nIn1_a | |

▶ For a detailed functional description see the L_Arithmetik FB.

## 8.7.3 Multiplication/Division

This function multiplies an analog input signal with a parameterisable factor. The factor must be selected in the form of a quotient (numerator and denominator).



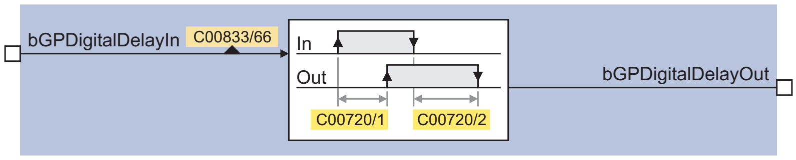

[8-7] GeneralPurpose function "Multiplication/division"

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00699/1 | L_MulDiv_1: Numerator | 0 | |
| C00699/2 | L_MulDiv_1: Denominator | 10000 | |

▶ For a detailed functional description see FB L_MulDiv.

### 8.7.4 Binary delay element

This function timely delays binary signals. On-delay and off-delay can be parameterised separately.



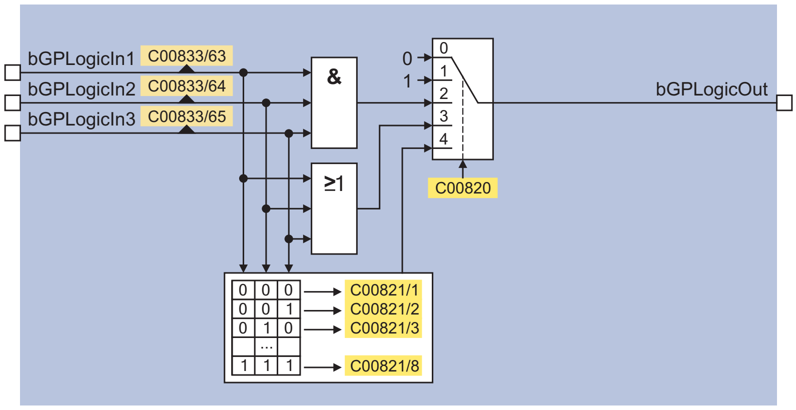

[8-8] GeneralPurpose function "Binary delay element"

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00720/1 | L_DigitalDelay_1: On delay | 0.000 | s |
| C00720/2 | L_DigitalDelay_1: Off delay | 0.000 | s |

▶ For a detailed functional description see FB L_DigitalDelay.

### 8.7.5 Binary logic

This function provides a binary output signal which is formed by a logic operation of the input signals. Alternatively, you can also select a fixed binary value which is independent of the input signals.



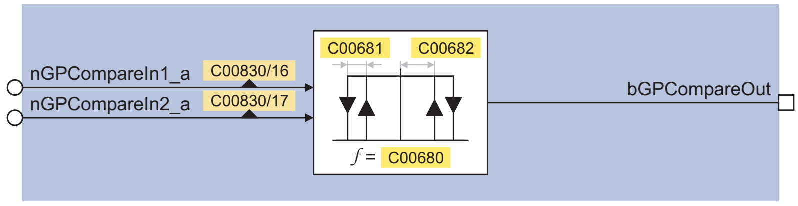

[8-9] GeneralPurpose function "Binary logic"

| Parameter | Info | Lenze setting |
|---|---|---|
| | | Value Unit |
| C00820 | L_DigitalLogic_1: Function | 0: bOut = 0 |
| C00821/1 | bIn1=0/bIn2=0/bIn3=0 | 0: FALSE |
| C00821/... | ... | ... |
| C00821/8 | bIn1=1/bIn2=1/bIn3=1 | 0: FALSE |

▶ For a detailed functional description see FB L_DigitalLogic.

## 8.7.6 Analog comparison

This function compares two analog signals and can be used e.g. to realise a trigger. The comparison operation, hysteresis and window size can be parameterised.



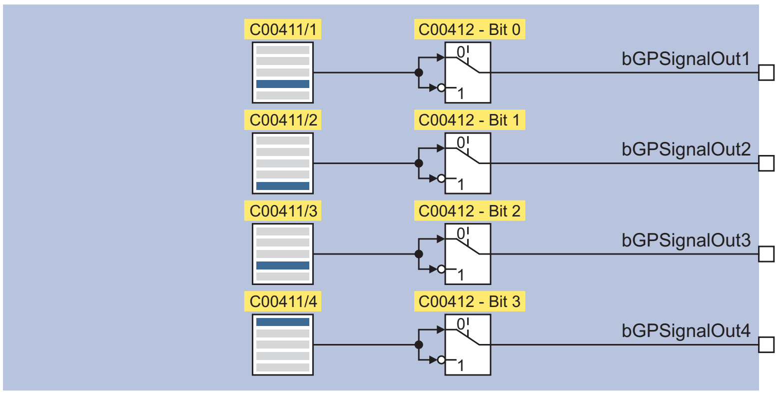

[8-10] GeneralPurpose function "Analog comparison"

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C00680 | L_Compare_1: Fct. | 6: \|In1\| < \|In2\| | |
| C00681 | L_Compare_1: Hysteresis | 0.50 | % |
| C00682 | L_Compare_1: Window | 2.00 | % |

▶ For a detailed functional description see FB L_Compare.

## 8.7.7 Binary signal monitor

This function serves to output four binary signals selected from a list of all binary output signals available in the drive controller. You can set an inversion of the output signals.



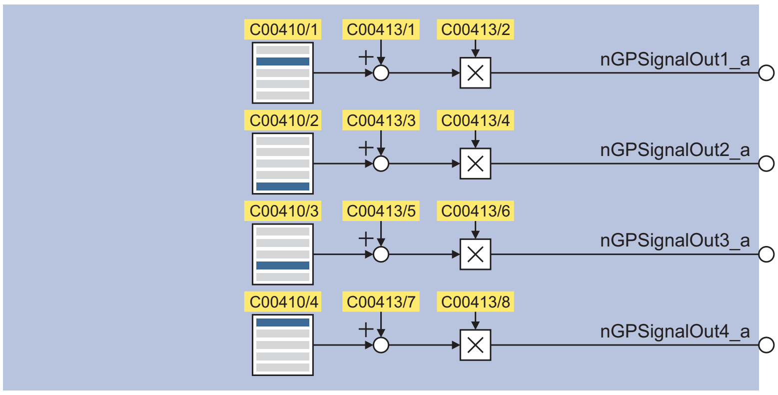

[8-11] GeneralPurpose function "Binary signal monitor"

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C00411/1...4 | L_SignalMonitor_b: Signal 1 ... 4 | 0: Not connected | |
| C00412 | L_SignalMonitor_b: Inversion | Bit coded | |

▶ For a detailed functional description see FB L_SignalMonitor_b.

## 8.7.8 Analog signal monitor

This function serves to output four analog signals selected from a list of all analog output signals available in the drive controller. Offset and gain of the source signals can be adjusted.



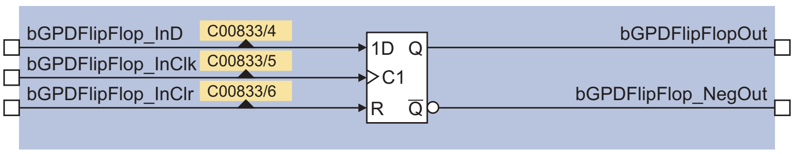

[8-12] GeneralPurpose function "Analog signal monitor"

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00410/1...4 | L_SignalMonitor_a: Signal 1 ... 4 | 0: Not connected | |
| C00413/1 | L_SignalMonitor_a: Signal 1 offset | 0.00 | % |
| C00413/2 | L_SignalMonitor_a: Signal 1 gain | 100.00 | % |
| C00413/3 | L_SignalMonitor_a: Signal 2 offset | 0.00 | % |
| C00413/4 | L_SignalMonitor_a: Signal 2 gain | 100.00 | % |
| C00413/5 | L_SignalMonitor_a: Signal 3 offset | 0.00 | % |
| C00413/6 | L_SignalMonitor_a: Signal 3 gain | 100.00 | % |
| C00413/7 | L_SignalMonitor_a: Signal 4 offset | 0.00 | % |
| C00413/8 | L_SignalMonitor_a: Signal 4 gain | 100.00 | % |

▶ For a detailed functional description see the L_SignalMonitor_a FB.

## 8.7.9 D-FlipFlop

This function saves the logic status of the data input (1D) in case of an active clock edge at the clock input (C1) and puts out its value in sequence at the output Q. If there is no active clock edge, the input value is not accepted.

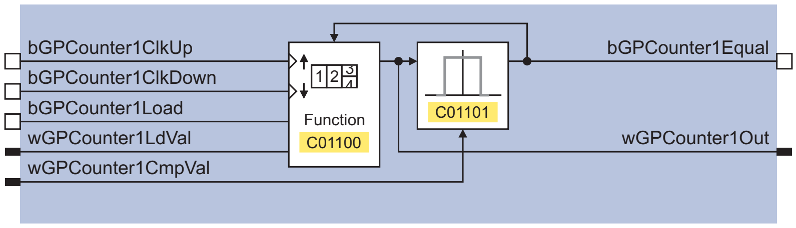



[8-13] GeneralPurpose function "D-FlipFlop" (clock-edge controlled)

▶ For a detailed functional description see FB L_DFlipFlop.

### 8.7.10 Numerator

This function is a digital upcounter and downcounter with a comparison operation.




[8-14] GeneralPurpose function "Counter"

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C01100/1 | L_Counter_1: Function | 0: Normal counting | |
| C01101/1 | L_Counter_1: Comparison | 0: Greater than or equal to | |

- Only available with TA "Table positioning".
- For a detailed functional description see FB L_Counter.

# 9 Basic drive functions (MCK)

In this chapter, the standard and basic drive functions integrated in the **Motion Control Kernel** (MCK) of the 8400 TopLine are described to which the active technology application can gain access via defined internal interfaces. As a result, the time-consuming creation of individual FB interconnections is avoided and the amount of work and complexity involved in the implementation of standard functions is minimised.

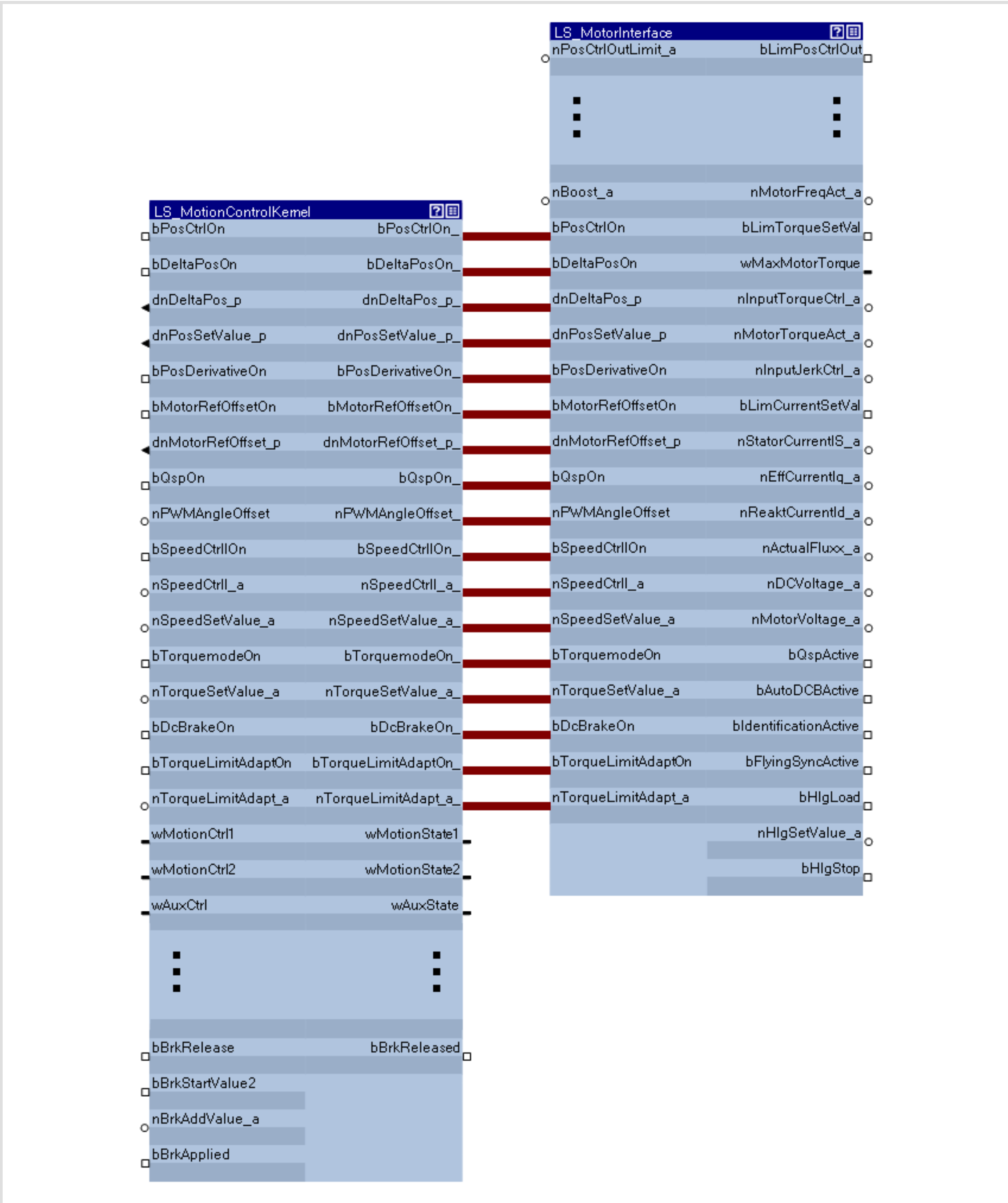
In the **Motion Control Kernel**, for example, an automatic holding brake control function is integrated which controls the holding brake in relation to the speed setpoint and various other internal control signals. Due to integrated automatic brake operation, the user is relieved of the task of managing these control signals.

Other standard functions integrated in the **Motion Control Kernel** in the case of the 8400 TopLine controller are, for example:

- Manual jog, e.g. for manual setting-up operation
- Homing for positioning functions of the drive
- Position profile generator for positioning functions of the drive

## 9.1 Basic signal flow

The **Motion Control Kernel** is connected between setpoint generator (e.g. ramp generator, PID process controller, etc.) and the motor control function in the case of the available technology applications. For problem-free interaction of the **Motion Control Kernel** and motor control function, the two associated system blocks LS_MotionControlKernel and LS_MotorInterface have interfaces with corresponding inputs/outputs. These are visible in the FB Editor for monitoring purposes and must be connected to each other:




[9-1] Interconnection of Motion Control Kernel and motor control function

In the interconnection previously shown, the **Motion Control Kernel** monitors every interface. Some of the signals such as a quick stop request or a DC-injection braking request are directly passed through to the motor control. However, other signals are passed through or modified depending on the operating mode (e.g. synchronising a setpoint selection via ramp function).

## 9.2 Internal interfaces | System block "LS_MotionControlKernel"

In the Function Block editor, the system block **LS_MotionControlKernel** provides the interfaces to the **Motion Control Kernel**.

### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| **Control and setpoint signals for motor control**<br>The purpose of the following inputs is to transfer control signals and setpoints to the internal motor control function (LS_MotorInterface). | | |
| bPosCtrlOn<br>BOOL | Activation of position/angle control | |
| | FALSE | Position/angle control deactivated. |
| | TRUE | Position/angle control activated. |
| bDeltaPosOn<br>BOOL | Activate position difference as setpoint selection<br>• In order to position the motor shaft, the position control function can work within the motor control function with the absolute position setpoint *dnPosSetValue_p* or alternatively with the speed setpoint *nSpeedSetValue_a* and the position difference *dnDeltaPos_p*. | |
| | FALSE | Positioning with position setpoint *dnPosSetValue_p*. |
| | TRUE | Positioning with speed setpoint *nSpeedSetValue_a* and position difference *dnDeltaPos_p*. |
| dnDeltaPos_p<br>DINT | Position difference (following error input)<br>• Difference between setpoint position and actual position in [increments]<br>• Is used for position control if *bDeltaPosOn* = TRUE.<br>• Scaling: a revolution is displayed with 65536 increments or steps. | |
| dnPosSetValue_p<br>DINT | Absolute position setpoint in [increments]<br>• Is used for position control if *bDeltaPosOn* = FALSE.<br>• Scaling: a revolution is displayed with 65536 increments or steps. | |
| bPosDerivativeOn<br>BOOL | Create a setpoint for the speed controller from the position setpoint<br>• For highly dynamic control systems, the setpoint for the speed controller can be created from the absolute position setpoint *dnPosSetValue_p* instead of the speed setpoint *nSpeedSetValue_a*. ▸ Position control/additive speed specification | |
| | TRUE | Create a speed setpoint from the position setpoint.<br>• The absolute position setpoint *dnPosSetValue_p* is differentiated and a speed value is created which is the setpoint for the speed controller.<br>• Internal limitation of 65536 increments/ms. |
| bMotorRefOffsetOn<br>BOOL | In preparation - Input has no function at present time!<br>Adopt home position ("Homing on the fly") | |
| | FALSE↗TRUE | Adopt home position *dnMotorRefOffset_p*. |

| Identifier | Data type | | Information/possible settings |
|---|---|---|---|
| dnMotorRefOffset_p | DINT | | Home position in [increments]<br>• Value is adopted when control bit 9 ("HomeSetPos") is set from "0" to "1" in the MCK control word and is sent to the motor control function via the *dnMotorRefOffset_p_* output.<br>• Transfer is possible in any operating mode ("Homing on the fly").<br>• For this function, the two outputs *bMotorRefOffsetOn_* and *dnMotorRefOffset_p_* must be connected to the inputs of the same name of the LS_MotorInterface system block. |
| bQspOn | BOOL | | Trigger quick stop (QSP) via the MCK<br>• Also see device command "Activate/Deactivate quick stop". |
| | | TRUE | Activate quick stop<br>• Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act}$ = 0).<br>• A pulse inhibit (CINH) is set if the auto-DCB function has been activated via C00019.<br>• The motor is kept at a standstill during closed-loop operation (function in preparation). |
| | | FALSE | Deactivate quick stop<br>• The quick stop is deactivated if no other source for the quick stop is active.<br>• C00159 displays a bit code of active sources/causes for the quick stop. |
| nPWMAngleOffset | INT | | Angular offset input<br>• Scaling: 16384 ≡ 100 %<br>• Setting range: 0 ... 199.99 % |
| bSpeedCtrlIOn | BOOL | | Directly set the I-component of speed controller<br>• In order to statically specify a minimum torque, e.g. when a load is being lifted. |
| | | TRUE | Se the I-component of the speed controller to the value *nSpeedCtrlI_a*. |
| nSpeedCtrlI_a | INT | | I-component of the speed controller<br>• Value is adopted in the case of a FALSE-TRUE edge at the input *bSpeedCtrlIOn*. |
| nSpeedSetValue_a | INT | | Rotation speed/velocity setpoint |
| bTorquemodeOn | BOOL | TRUE | Switch on torque-controlled operation |
| nTorqueSetValue_a | INT | | Torque setpoint |
| bDcBrakeOn | BOOL | | Manual DC-injection braking (DCB)<br>• For this function, the *bDcBrakeOn_* output signal must be connected to the *bDcBrakeOn* input of the same name of the LS_MotorInterface system block.<br>• Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking". |
| | | | **Note!**<br>Holding braking is not possible when this braking mode is used!<br>Use the basic "Holding brake control" function for controlling the holding brake with a low rate of wear. |
| | | FALSE | Deactivate DC-injection braking. |
| | | TRUE | Activate DC-injection braking, i.e. the drive is brought to a standstill by means of DC-injection braking.<br>• The braking effect stops when the rotor is at standstill.<br>• After the hold time (C00107) has expired, the controller sets the pulse inhibit (CINH). |

| **Identifier**<br>Data type | **Information/possible settings** |
|---|---|
| bTorqueLimitAdaptOn<br>BOOL | Adaptation of torque limitation<br>TRUE: Activate adaptation of torque limitation. |
| nTorqueLimitAdapt_a<br>INT | Value for adaptation of torque limitation<br>• Scaling: 16384 ≡ 100 % of the torque limit specified at the LS_MotorInterface system block via the *nTorqueMotLimit_a* and *nTorqueGenLimit_a* inputs. |
| **Control words** | |
| MCK: wMotionCtrl1<br>wMotionCtrl2<br>WORD | MCK control word 1 & 2<br>• The two control words together form a 32-bit double control word with which the entire Motion Control Kernel is controlled.<br>• All motion profiles in the different operating modes can be operated via this interface.<br>• See the "MCK control word" subchapter for a detailed description of the individual control bits.<br>• Display parameter: C01240 |
| wAuxCtrl<br>WORD | Fur future extensions - Input has not function at present time!<br>Additional control word |
| wSMCtrl<br>WORD | Interface to the optional safety system.<br>• Setting control bit 0 ("SafeStop1") in this control word causes e.g. the automatic deceleration of the drive to standstill within this application (in the **Motion Control Kernel**).<br>• See the "Interface to safety system" subchapter for a detailed description of the individual control bits. |
| **Control and setpoint signals for Motion Control Kernel function** | |
| dnProfilePosition_p<br>DINT | Profile position in [increments]<br>• Position which is to be entered into a profile data set selected via the MCK control word.<br>• In the "absolute" mode, this position is a target position. However, in the "relative" mode it is a relative traverse path. |
| nSpeedAddValue_v<br>INT | Additive speed setpoint in [inc/ms] |
| nSpeedOverride_a<br>INT | Value for Speed override<br>• Percentage multiplier (0 ... 199.99 %) for the currently active speed.<br>• 16384 ≡ 100 % of the maximum traversing speed (display in C01211/1).<br>• If the override value is 0 %, the drive is brought to a standstill. |
| nAccOverride_a<br>INT | Value for Acceleration override<br>• Percentage multiplier (0 ... 199.99 %) for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised acceleration of the corresponding operating mode.<br>• If the override value is 0 %, acceleration ceases. |
| nSRampOverride_a<br>INT | Value for S-ramp smoothing override<br>• Percentage multiplier (0 ... 100 %) for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised S-ramp time (C01306/1...15).<br>• Values > 16384 are ignored.<br>**Note!**<br>If the *nSRampOverride_a* input remains unconnected or if an override value of "0 %" is selected, activation of the S-ramp override causes deactivation of the S-ramp time.<br>• Deactivation of the S-ramp time before the start of a profile with S-ramp time causes linear ramp generation.<br>• Deactivation of the S-ramp time during a traversing process, however, is not accepted immediately in the profile generator, but the profile generator checks automatically when an online change of the ramp form can be carried out and then initiates it automatically. |
| bLimitSwitchPos<br>BOOL | Input for Hardware limit switch (positive) |
| bLimitSwitchNeg<br>BOOL | Input for Hardware limit switch (negative) |

| Identifier / Data type | | Information/possible settings | |
|---|---|---|---|
| bHomingMark | BOOL | Input for pre-stop mark/pre-stop signal for homing | |
| bBrkRelease | BOOL | Holding brake control:<br>Releasing/applying the brake in connection with the selected operating mode | |
| | | FALSE | Apply brake.<br>• During automatic operation, the internal brake logic controls of the brake. |
| | | TRUE | Release brake manually (forced release).<br>• **Note!**<br>The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If the brake control has inhibited the controller, this inhibit is deactivated again.<br>• In semi-automatic operation, the brake is released including feedforward control. |
| bBrkStartValue2 | BOOL | Holding brake control:<br>Selection of the torque feedforward control value for manual specification of the feedforward control value<br>• Only effective if bit 4 in C02582 is set to "1".<br>▸ Feedforward control of the motor before release | |
| | | FALSE | Starting value 1 (C02581/4) active. |
| | | TRUE | Starting value 2 (C02581/5) active. |
| nBrkAddValue_a | INT | Holding brake control:<br>Additive feedforward control value (speed or torque) in [%] for torque feedforward control when the respective control mode is started<br>• For speed control: 100 % ≡ reference speed (C00011)<br>• For torque control: 100 % ≡ maximum torque (C00057)<br>▸ Feedforward control of the motor before release | |
| bBrkApplied | BOOL | Holding brake control:<br>Input for status detection via switching contacts at the brake<br>• Only effective if bit 5 in C02582 is set to "1". | |
| | | FALSE | Brake is released. |
| | | TRUE | Brake is applied. |

## Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| **Control and setpoint signals for motor control**<br>The following outputs are used to transfer control signals and setpoints to the internal motor control function (LS_MotorInterface). | | |
| bPosCtrlOn_<br>BOOL | TRUE | Activate position control. |
| bDeltaPosOn_<br>BOOL | TRUE | Control to adjust for following errors. |
| dnDeltaPos_p_<br>DINT | Following error input | |
| dnPosSetValue_p_<br>DINT | Absolute position setpoint | |
| bPosDerivativeOn_<br>BOOL | TRUE | Activate precontrol function of speed controller. |
| bMotorRefOffsetOn_<br>BOOL | FALSE↗TRUE | Accept home position. |
| dnMotorRefOffset_p_<br>DINT | Home position | |
| bQspOn_<br>BOOL | TRUE | Activate quick stop |
| nPWMAngleOffset_a_<br>INT | Angular offset input | |
| bSpeedCtrlIOn_<br>BOOL | TRUE | Set I-component of speed controller. |
| nSpeedCtrlI_a_<br>INT | I-component of the speed controller | |
| nSpeedSetValue_a_<br>INT | Main setpoint of speed | |
| bTorqueModeOn_<br>BOOL | TRUE | Switch on torque-guided operation. |
| nTorqueSetValue_a_<br>INT | Torque setpoint | |
| bDcBrakeOn_<br>BOOL | TRUE | Activate DC-injection braking. |
| bTorqueLimitAdaptOn_<br>BOOL | TRUE | Activate adaptation of torque limitation. |
| nTorqueLimitAdapt_a_<br>INT | Value for adaptation of torque limitation | |
| **Status words** | | |
| wMotionState1<br>wMotionState2<br>WORD | MCK status word 1 & 2<br>• For a detailed description of the individual status bits, see subchapter entitled "MCK status word.<br>• Display parameter: C01241 | |
| wAuxState<br>WORD | For future extensions - Output has no function at present time!<br>Additional status word | |

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| **Status signal and actual-value signals from Motion Control Kernel functions** | | |
| nSpeedSet_v<br>INT | Speed setpoint selection in [increments/ms]<br>• 16384 ≡ 15000 rpm | |
| dnPosTarget_p<br>DINT | Target position in [increments]<br>• 65535 ≡ 1 revolution of the motor shaft | |
| dnPosSet_p<br>DINT | Module position in [increments]<br>• 65535 ≡ 1 revolution of the motor shaft<br>▶ Activation of the modulo measuring system | |
| dnPosSetRelative_p<br>DINT<br>(from version 02.00.00) | Relative feed in positioning processes in [increments]<br>• 65535 ≡ 1 revolution of the motor shaft | |
| wActProfileNo<br>WORD | Number of current profile | |
| wFollowProfileNo<br>WORD | Number of sequence profile | |
| bPosBusy<br>BOOL | TRUE | Setpoint profile generation is active. |
| bPosDone<br>BOOL | TRUE | Setpoint profile generation is completed<br>(set position = target position). |
| bHomingDone<br>BOOL | TRUE | Homing has been carried out.<br>• The *bHomingDone* output, in contract to the *bHomePosAvailable* output, remains set to TRUE even if a travel command that resulted in a reset of the home position has been carried out. |
| bHomePosAvailable<br>BOOL | TRUE | Home position is known. |
| bBrkReleaseOut<br>BOOL | Trigger signal for switching element holding brake control via a digital output<br>• Use bit 0 under C02582 to activate inverted switching element triggering.<br>▶ Holding brake control | |
| | FALSE | Apply brake. |
| | TRUE | Release brake. |
| bBrkReleased<br>BOOL | "Brake released" status signal considering the brake release time<br>• If the holding brake is triggered to be applied, *bBrkReleased* is immediately reset to FALSE even if the brake application time has not elapsed yet!<br>▶ Holding brake control | |
| | TRUE | Brake released (when the brake release time has elapsed). |
| wGearNum<br>wGearDenom<br>WORD<br>(from version 02.00.00) | Output of the gearbox factor set in C01202/1 and C01202/2<br>• These outputs can be connected to the *GearNum-/GearDenom* inputs of the FBs which process a gearbox factor (FB L_PhiIntegrator_1, FB L_DFSET_1, FB L_CalcDiameter_1).<br>• C01067/1...3 can be used to inform these FBs that the gearbox direction of rotation is inverted is mostly done by selecting "motor mounting position inverted"). | |

### Related topics:

▶ MCKInterface

▶ Control inputs | "L_MckCtrlInterface" function block

▶ Status outputs | FB "L_MckStateInterface"

### 9.2.1 MCK control word

The motion control function implemented in the LS_MotionControlKernel system block can be controlled via the *wMotionCtrl1* and *wMotionCtrl2* control words. Together, both control words form a 32-bit control double word which serves to control the entire MotionControlKernel. All motion profiles in the various operating modes can be operated via this interface.

For direct control via a fieldbus system, the two control words can be triggered by the field bus via a port block. In addition to a few other signals (e.g. limit switch, speed override) that are connected to the digital inputs of the drive controller, all control signals can therefore be activated/deactivated via the fieldbus used (CAN, PROFIBUS, etc.).

As an alternative to this, changes to the individual control bits or bit fields (e.g. for specifying the profile number) can also be carried out via separate process inputs provided by the L_MckCtrlInterface function block. The control words *wOutMckPosCtrl_1* and *wOutMckPosCtrl_2* output by this function block constitute the input information for the LS_MotionControlKernel system block in this case.

#### MCK control word 1 (wMotionCtrl1)

| Bit | Designation | Description | | | | |
|---|---|---|---|---|---|---|
| 0<br>...<br>3 | OpMode_Bit0<br>...<br>OpMode_Bit3 | **Operating mode** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Speed follower | 0 | 0 | 0 | 0 |
| | | Homing | 0 | 0 | 0 | 1 |
| | | Manual jog | 0 | 0 | 1 | 0 |
| | | Positioning | 0 | 0 | 1 | 1 |
| | | Stop | 0 | 1 | 0 | 0 |
| | | Position follower | 0 | 1 | 0 | 1 |
| | | All other possible settings are reserved for future extensions! | | | | |
| 4<br>5 | ManJogPos<br>ManJogNeg | **Manual jog** | | | **Bit 5** | **Bit 4** |
| | | Stop manual jogging | | | 0 | 0 |
| | | Manual jog, right | | | 0 | 1 |
| | | Manual jog, left | | | 1 | 0 |
| | | No change from previous status | | | 1 | 1 |
| 6 | ManExecute 2ndSpeed | "1" ≡ Change over to manual speed 2 | | | | |
| 7 | ReleaseLimitSwitch | "1" ≡ Retract operated hardware limit switch | | | | |
| 8 | HomStartStop | "1" ≡ Start/stop homing process | | | | |
| 9 | HomSetPos | "1" ≡ Set homing position | | | | |
| 10 | HomResetPos | "1" ≡ Reset the "Reference known" status<br>• *bHomePosDone* and *bHomePosAvailable* are reset to FALSE.<br>• The positions remain unaffected. | | | | |
| 11 | EnableSpeedOverride | "1" ≡ Activate speed override | | | | |
| 12 | EnableAccOverride | "1" ≡ Activate acceleration override | | | | |
| 13 | Enable SRampOverride | "1" ≡ Activate S-ramp override | | | | |
| 14 | PosTeachSetPos | "1" ≡ Teach MCK set position into the selected profile | | | | |
| 15 | PosTeachActPos | "1" ≡ Teach current position into the selected profile | | | | |

### MCK control word 2 (wMotionCtrl2)

<table>
<tr><th>Bit</th><th>Designation</th><th colspan="9">Description</th></tr>
<tr><td>16</td><td>PosExecute</td><td colspan="9">"0↗1" ≡ Start travelling</td></tr>
<tr><td>17</td><td>PosFinishTarget</td><td colspan="9">"0↗1" ≡ Complete cancelled profile</td></tr>
<tr><td>18</td><td>PosDisable<br>FollowProfile</td><td colspan="9">"1" ≡ Do not travel sequence profile</td></tr>
<tr><td>19</td><td>PosStop</td><td colspan="9">"1" ≡ Cancel travelling<br>From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked).</td></tr>
<tr><td rowspan="13">20<br>...<br>23</td><td rowspan="13">PosModeBit0<br>...<br>PosModeBit3</td><td colspan="5">Positioning mode</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td colspan="5">Positioning mode = setting in C01300/1...15</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="5">Absolute (shortest path)</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td colspan="5">Continuous</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">Relative</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td colspan="5">absolute (Cw)</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td colspan="5">absolute (Ccw)</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td colspan="5">Absolute (shortest path) to TP</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td colspan="5">Continuous to TP</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td colspan="5">Relative to TP</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td colspan="5">Absolute (Cw) on TP</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td colspan="5">Absolute (Ccw) on TP</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td colspan="9">All other possible settings are reserved for future extensions!</td></tr>
<tr><td rowspan="7">24<br>...<br>31</td><td rowspan="7">ProfileNo_Bit0<br>...<br>ProfileNo_Bit7</td><td>Profile</td><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>Profile 0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Profile 1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>Profile 2</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>...</td><td colspan="8">...</td></tr>
<tr><td>Profile 15</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td colspan="9">All other possible settings are reserved for future extensions!</td></tr>
</table>

**Note!**

The profile 0 is no valid profile for the "Positioning" operating mode.

If a driving request with an invalid profile number is started, the response set in C00595/12 occurs (Lenze setting: "WarningLocked").

**Tip!**

Travel requests/profiles can also be started while the drive is running. The drive does not need to be at standstill.

## 9.2.2 MCK status word

### MCK status word 1 (wMotionState1)

| Bit | Designation | Description | | | | |
|---|---|---|---|---|---|---|
| 0<br>...<br>3 | ActOpModeBit00<br>...<br>ActOpModeBit03 | **Active operating mode** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Speed follower | 0 | 0 | 0 | 0 |
| | | Homing | 0 | 0 | 0 | 1 |
| | | Manual jog | 0 | 0 | 1 | 0 |
| | | Positioning | 0 | 0 | 1 | 1 |
| | | Stop or Safe stop 1 (SS1) | 0 | 1 | 0 | 0 |
| | | Position follower | 0 | 1 | 0 | 1 |
| | | StandBy (internal operating mode in the event of quick stop, pulse inhibit, and DC-injection braking) | 1 | 1 | 1 | 1 |
| 4 | Busy | "1" ≡ Internal profile generation is active. A speed profile is being generated. | | | | |
| 5 | Done | "1" ≡ Generation of a speed profile with the selected position has been completed. | | | | |
| 6 | AcceleratingActive | "1" ≡ Profile generation phase is in the acceleration process. | | | | |
| 7 | ConstSpeedDuty | "1" ≡ Profile generation phase at constant speed active. | | | | |
| 8 | DeceleratingActive | "1" ≡ Profile generation phase is in the deceleration process. | | | | |
| 9 | S_ShapingActive | In preparation - status bit without function at the moment!<br>"1" ≡ Rounding during acceleration/deceleration active. | | | | |
| 10 | Pos. HW-Limit Detected | "1" ≡ Positive limit switch has triggered.<br>• Reset only possible via"Manual jog" mode! | | | | |
| 11 | Neg. HW-Limit Detected | "1" ≡ Negative limit switch has triggered.<br>• Reset only possible via"Manual jog" mode! | | | | |
| 12 | HomPosDone | "1" ≡ Homing has been completed. | | | | |
| 13 | HomPosAvailable | "1" ≡ The home position has been detected and is known in the drive. | | | | |
| 14 | Reserved | - | | | | |
| 15 | Reserved | - | | | | |

**Note!**

The internal "StandBy" operating mode is assumed if pulse inhibit, quick stop and/or DC-injection braking are activated.

- No setpoint generation via the **Motion Control Kernel** in this operating mode.
- If the holding brake control sets a controller inhibit when the holding brake is closed, the internal "StandBy" operating mode is not assumed.

### MCK status word 2 (wMotionState2)

| Bit | Designation | Description | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 16 | DwellTime | "1" ≡ Dwell time after reaching the setpoint position is active.<br>▸ Target position monitoring (status "drive in target") | | | | | | | | |
| 17 | InTarget | "1" ≡ Dwell time has expired and current actual position is in the set target window.<br>▸ Target position monitoring (status "drive in target") | | | | | | | | |
| 18 | PosDone | "1" ≡ Positioning profile has been completed in the "Positioning" or "Homing" mode. Setpoint position of a profile data set is in target. | | | | | | | | |
| 19 | Reserved | - | | | | | | | | |
| 20<br>...<br>23 | ActPosMode_Bit00<br>...<br>ActPosMode_Bit03 | **Active positioning mode** | | | | | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Absolute (shortest path) | | | | | 0 | 0 | 0 | 1 |
| | | Continuous | | | | | 0 | 0 | 1 | 0 |
| | | Relative | | | | | 0 | 0 | 1 | 1 |
| | | absolute (Cw) | | | | | 0 | 1 | 0 | 0 |
| | | absolute (Ccw) | | | | | 0 | 1 | 0 | 1 |
| | | Absolute (shortest path) to TP | | | | | 1 | 0 | 0 | 0 |
| | | Continuous to TP | | | | | 1 | 0 | 0 | 1 |
| | | Relative to TP | | | | | 1 | 0 | 1 | 0 |
| | | Absolute (Cw) on TP | | | | | 1 | 0 | 1 | 1 |
| | | Absolute (Ccw) on TP | | | | | 1 | 1 | 0 | 0 |
| | | **Note:** The display of the active positioning mode depends on further factors. See subchapter "Override of the parameterised positioning mode". | | | | | | | | |
| 24<br>...<br>31 | ActProfileNo_Bit00<br>...<br>ActProfileNo_Bit07 | **Active profile** | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Profile 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Profile 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | Profile 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | | ... | ... | | | | | | | |
| | | Profile 15 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

## Note!

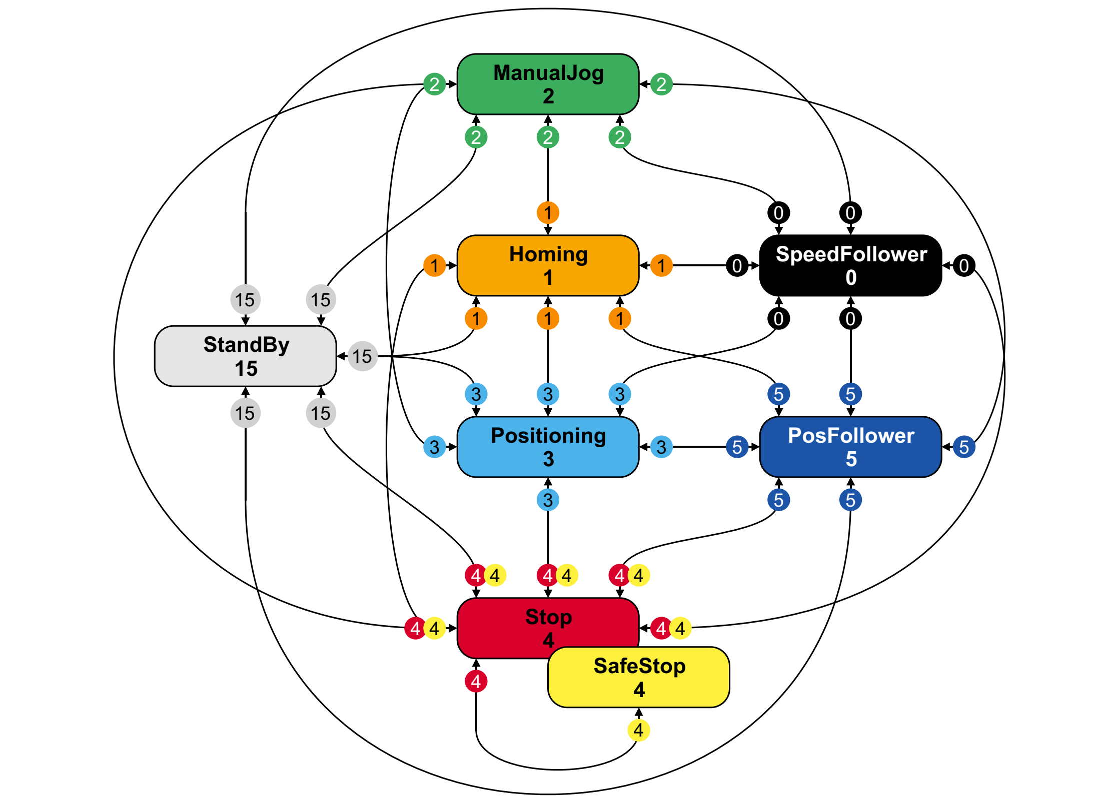
The effective positioning mode is generated via the MCK control word (bit 20 ... 23) and/or the parameters C01300/1...15 for defining the positioning mode in the profile data. Here, the setting in the MCK control word superimposes the mode selection via the corresponding parameter. This means that the mode selection and the touch probe enable are possible via process data.

For generating the effective positioning mode, the following applies:

- Valid PosMode in the MCK control word:
  → Use positioning mode of the MCK control word
- PosMode in the MCK control word = 0:
  → Use positioning mode set in C01300/x
- Invalid PosMode in the MCK control word:
  → Error message "Ck09: Positioning mode invalid"

## 9.2.3 MCK state machine

| Prio | | Condition | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|
| **Global conditions:** | | | | | | |
| 1 | 15 | Setpoint generation through Motor control (MCTRL):<br>• DCB = DC-injection braking<br>• QSP = quick stop<br>• CINH = controller inhibit | DCB, QSP, CINH → ≥1 → 15 | | | |
| 2 | 4 | "Safe stop 1" (SS1) requested<br>Interface to safety system | DCB (inverted), QSP (inverted), CINH (inverted), SafeStop1 → & → 4 | | | |
| **Conditions requested by MCK control word:** | | | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| 3 | 2 | Manual jog requested | 0 | 0 | 1 | 0 |
| 4 | 1 | Homing requested | 0 | 0 | 0 | 1 |
| 5 | 3 | Positioning requested | 0 | 0 | 1 | 1 |
| 6 | 0 | Speed follower requested | 0 | 0 | 0 | 0 |
| 7 | 5 | Position follower requested | 0 | 1 | 0 | 1 |
| 8 | 4 | Stop requested | 0 | 1 | 0 | 0 |




[9-2] MCK state machine

### 9.2.4 Interface to safety system

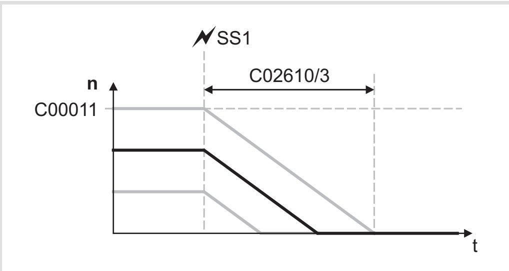
For operation with optional safety system, the LS_MotionControlKernel system block has the *wSMCtrl* input, This interface is used to transfer a control word by means of which the **Motion Control Kernel** can be supplied with information on requested or active safety functions. The **Motion Control Kernel** then initiates the necessary motion sequence (e.g. braking).

At the moment, only bit 0 in the *wSMCtrl* control word has a function. Additional functions are in preparation:

| Bit | Designation | Description |
|---|---|---|
| 0 | SafeStop1 | "1" ≡ Request for "Safe Stop 1" (SS1). |
| 1<br>...<br>15 | Reserved | In preparation - Still without function! |

#### Behaviour in case of request for "Safe Stop 1" (SS1)

The drive is brought to a standstill with the stopping ramp set in C02610/3.




[9-3] Ramping down to standstill

- The time set in C02610/3 refers to the down-ramping of the reference speed set in C00011.
- If the current speed is lower, the time to standstill is accordingly lower as well.

If the request is reset during the down-ramping process (bit 0 = "1↘0"), the behaviour depends on the active operating mode:

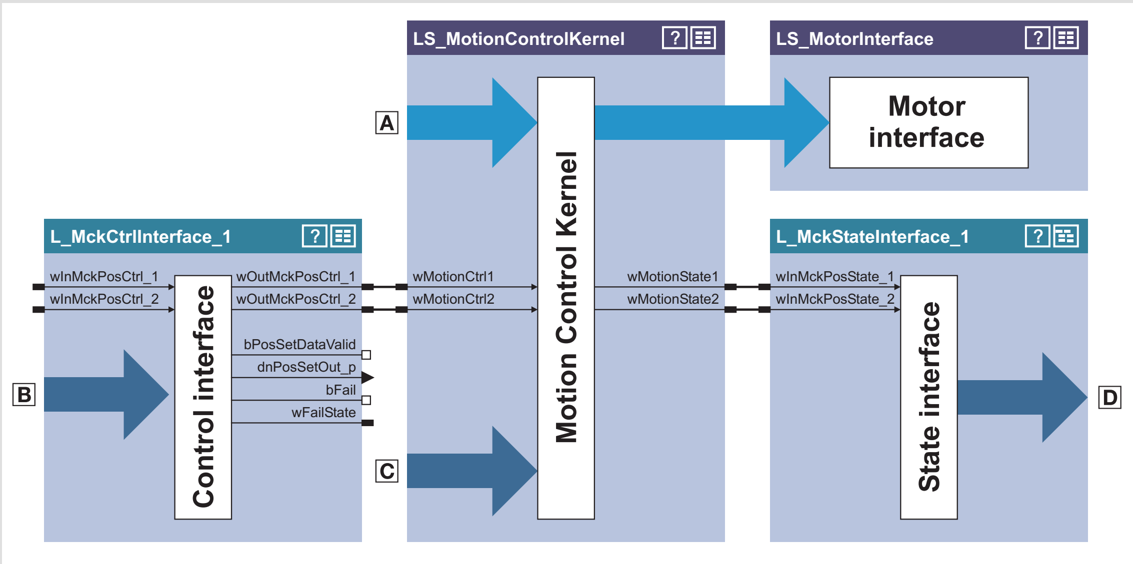
- In the "Speed follower" operating mode, direct synchronisation with the target speed takes place with the ramp time set in C02610/2.
- In the "Homing" operating mode, the ramp-down is continued with the deceleration for stop set in C01251 if the bit 8 ("HomStartStop") in the MCK control word = "0". If bit 8 is set, the homing process starts immediately in the selected homing mode.
- In the "Manual jog" operating mode, the behaviour depends on whether the manual jog initiator (*bManJogPos* or *bManJogNeg*) is still set:

| Bit 0 (SafeStop1) | bManJogPos bManJogNeg | Behaviour |
|---|---|---|
| 1 | TRUE | Ramping down to standstill |
| 0 | TRUE | Accelerating to manual speed |
| 0 | FALSE | Ramping down to standstill |

- In the "Positioning" operating mode, the behaviour depends on the setting in C01216.
- In the "Position follower" operating mode, (forward) positioning to the signalled absolute position always takes place if this position differs from the internal position.

## 9.3 MCKInterface

The so-called "MCK interface" described in this chapter consists of the two function blocks L_MckCtrlInterface and L_MckStateInterface, which are connected upstream respectively downstream of the LS_MotionControlKernel system block:




[9-4] Detail of the interconnection architecture for the "table positioning" technology application

A Control and setpoint signals for motor control

B Additional process inputs for controlling the **Motion Control Kernel**, e.g.:
- Selection of the operating mode
- Selection of the profile number
- Override of the positioning mode
- Control inputs for manual jogging, homing, positioning
- Control inputs for speed/acceleration/S-ramp override

C Control and setpoint signals for the **Motion Control Kernel** such as
- Selection of speed setpoint for speed follower
- Selection of positioning setpoint for position follower
- Selection of override values
- Connection for limit switch & pre-stop mark for homing
- Holding brake control

D Output of status signals of the **Motion Control Kernel**

### Control of the Motion Control Kernel

Control of the basic drive functions implemented in the Motion Control Kernel is carried out by means of

- direct specification of the control words, for example via a master control unit also connected to the fieldbus.
  - For this purpose, the control word inputs can be directly connected to the field bus interface **LP_MciIn** respectively **LP_CanIn**.
  - See the "MCK control word" chapter for a detailed description of the individual control bits.
- the specification of individual process signals at the L_MckCtrlInterface FB which are then ORed with the control word.

### Plausibility check

The process signals applied to the L_MckCtrlInterface FB are logically linked by means of an OR logic operation to the specified control word via the two control word inputs *wInMckPosCtrl_1* and *wInMckPosCtrl_2* and, after a plausibility check, are output via the two control word outputs *wOutMckPosCtrl_1* and *wOutMckPosCtrl_2* .

- The profile number, the operating mode, and the positioning mode are checked for plausibility.
- If implausibility is detected, only control bit information is output and the *bFail* output is set to TRUE.
- The result of the plausibility check is provided as the *wFailState* output word and displayed in C01299.

The control words that are output, namely *wOutMckPosCtrl_1* and *wOutMckPosCtrl_2*, constitute the input information for the LS_MotionControlKernel system block.

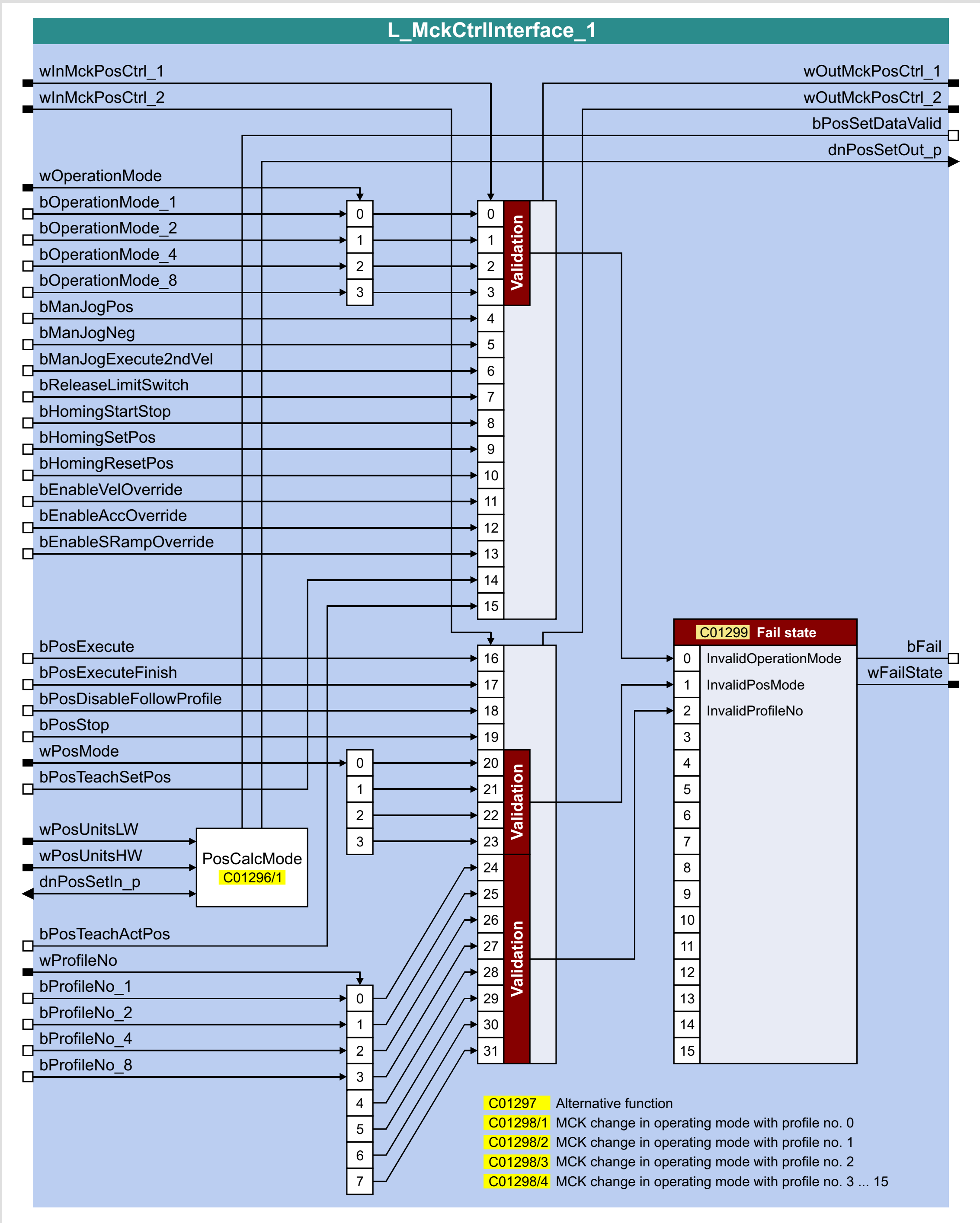
### Status information of the Motion Control Kernel

Output by the LS_MotionControlKernel, the status words *wMotionState1* and *wMotionState2* also constitute input information for the downstream L_MckStateInterface FB, which provides this information to the application in the form of process signals.

### 9.3.1 Control inputs | "L_MckCtrlInterface" function block

This FB provides process inputs for controlling different basic functions of the **Motion Control Kernel**.

In addition to the ORing of discrete input signals with the control word inputs, the FB has further functions that are described in the following subchapters.



## Inputs

| Identifier / Data type | Information/possible settings | |
|---|---|---|
| wInMckPosCtrl_1<br>wInMckPosCtrl_2<br>WORD | Direct specification of the MCK control word 1 & 2<br>• For example, via a master control unit that is also connected to the fieldbus. For this purpose, the control word inputs can be directly connected to the **LP_MciIn** respectively **LP_CanIn1** fieldbus interface.<br>• The two control words together form a 32-bit double control word with which the entire Motion Control Kernel is controlled.<br>• All motion profiles in the different operating modes can be operated via this interface.<br>• See the "MCK control word" chapter for a detailed description of the individual control bits. | |
| wOperationMode<br>WORD | Selection of the operating mode of the **Motion Control Kernel**<br>• Only bit 0 ... bit 3 of *wOperationMode* are evaluated.<br>• If an invalid operating mode is selected, the response set in C00595/11 occurs (Lenze setting: "Warning").<br>• The current operating mode is displayed in C01243. | |
| | 0 | Speed follower |
| | 1 | Homing |
| | 2 | Manual jog |
| | 3 | Positioning |
| | 4 | Stop |
| | 5 | Position follower |
| | 6 ... 15 | Reserved for future extensions |
| bOperationMode_1<br>...<br>bOperationMode_8<br>BOOL | Binary-coded selection of the operating mode of the **Motion Control Kernel**<br>• See the "MCK control word" chapter for a detailed description of the individual control bits.<br>• If an invalid operating mode is selected, the response set in C00595/11 is activated (Lenze setting: "Warning").<br>• The current operating mode is displayed in C01243. | |
| bManJogPos<br>bManJogNeg<br>BOOL | Manual jog:<br>*bManJogPos* = TRUE: Manual jog right<br>*bManJogNeg* = TRUE: Manual jog left<br>Both inputs = TRUE: No change compared to previous state<br>Both inputs = FALSE: Stop manual jog | |
| bManJogExecute2ndVel<br>BOOL | Manual jog: Changeover to speed 2 | |
| | FALSE | Speed 1 (C01231/1) active |
| | TRUE | Speed 2 (C01231/2) active |
| bReleaseLimitSwitch<br>BOOL | Manual jog: Retract operated limit switch | |
| | TRUE | Retract operated limit switch (in opposite direction) |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bHomingStartStop<br>BOOL | Homing: Start/stop homing<br>• Only possible in the "referencing" operating mode. | |
| | TRUE | If one of referencing modes "0" ... "15" in C01221 is selected:<br>**Start reference search**<br>• The current status of the reference search is indicated via the status outputs *bHomingDone* and *bHomePosAvailable*.<br>• If the 100_ Set_Ref_directly" referencing mode has been selected in C01221, the home position can be set manually via the *bHomeStartStop* input with the drive at a standstill. The current actual position is set as the home position (C01227/2) |
| | | If referencing mode "100: SetRef" is selected:<br>**Setting the home position manually**<br>• The home position is set manually with the drive at a standstill. The current actual position now corresponds to the reference position set in C01227/2 in the machine measuring system. |
| | TRUE↘FALSE | Stop homing.<br>• If the *bHomingStartStop* input is reset to FALSE during active referencing, homing is cancelled and the drive is brought to a standstill. |
| bHomingSetPos<br>BOOL | Homing: Set home position ("on the fly" homing)<br>• With referencing "on the fly", the home position of a machine can be set during ongoing movement. Jerking and compensating movements do not occur. | |
| | FALSE↗TRUE | The position at the *dnMotorRefOffset_p* input of the LS_MotionControlKernel SB at the instant of activation is the set home position. |
| bHomingResetPos<br>BOOL | Homing: Delete home position<br>**Note:** With this function, positions are not deleted but only the status signals *bHomePosAvailable* and *bHomePosDone* are reset. Setpoints and actual positions remain untouched until a renewed reference setting or homing. | |
| | FALSE↗TRUE | The internal status "reference known" is reset.<br>• The controller is no longer referenced.<br>• The process outputs *bHomePosAvailable* and *bHomingDone* are reset to FALSE. |
| bEnableVelOverride<br>BOOL | Speed override | |
| | TRUE | Activate speed override |
| bEnableAccOverride<br>BOOL | Acceleration override | |
| | TRUE | Activate acceleration override |
| bEnableSRampOverride<br>BOOL | S-ramp smoothing override<br>**Note!**<br>If the *nSRampOverride_a* input on the LS MotionControlKernel remains unconnected or if "0 %" is specified as the override value, activation of the S-ramp override results in deactivation of the S-ramp time.<br>• Deactivation of the S-ramp time before the start of a profile with S-ramp time causes linear ramp generation.<br>• Deactivation of the S-ramp time during a traversing process, however, is not accepted immediately in the profile generator, but the profile generator checks automatically when an online change of the ramp form can be carried out and then initiates it automatically. | |
| | TRUE | Activate S-ramp smoothing override |
| bPosExecute<br>BOOL | Positioning: Start travelling | |
| | FALSE↗TRUE | Execute selected profile |
| bPosExecuteFinish<br>BOOL | Positioning: Complete cancelled profile | |
| | FALSE↗TRUE | A positioning process previously cancelled, e.g. by *bPosStop* or due to a device error, is resumed by travelling to the original target. |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bPosDisableFollowProfile<br>BOOL | Positioning: Do not execute sequence profile (switch-off profile linkage) | |
| | TRUE | Evaluation of the sequence profile number parameterised in C01307/1...15 for the selected profile is suppressed. |
| bPosStop<br>BOOL | Positioning: Cancel travelling | |
| | TRUE | Stop positioning<br>From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked). |
| wPosMode<br>WORD | Override of the positioning mode set in the profile data<br>• Via this input, an override of the positioning mode parameterised in C01300/1...15 for the selected profile is possible.<br>• The value set in C01300/1...15 is not overwritten in this case.<br>• Only bit 0 ... bit 3 of *wPosMode* are evaluated.<br>• If *wPosMode* = 0 is selected, the positioning mode set in C01300/1...15 is used. | |
| | 0 | Positioning mode = setting in C01300/1...15 |
| | 1 | Absolute (shortest path) |
| | 2 | Continuous |
| | 3 | Relative |
| | 4 | absolute (Cw) |
| | 5 | absolute (Ccw) |
| | 8 | Absolute (shortest path) to TP |
| | 9 | Continuous to TP |
| | 10 | Relative to TP |
| | 11 | Absolute (Cw) on TP |
| | 12 | Absolute (Ccw) on TP |
| | All other possible settings are reserved for future extensions! | |
| bPosTeachSetPos<br>BOOL | Position teaching: MCK setpoint position | |
| | FALSE↗TRUE | Teach MCK setpoint position into the selected profile. |
| wPosUnitsLW<br>wPosUnitsHW<br>WORD | Selection of the target position in [units]<br>• *wPosUnitsLW* = LOW word, *wPosUnitsHW* = HIGH word<br>• The mode for calculating the position is selected in C01296/1. | |
| dnPosSetIn_p<br>DINT | Selection of the target position in [increments]<br>• The mode for calculating the position is selected in C01296/1. | |
| bPosTeachActPos<br>BOOL | Position teaching: Current position | |
| | FALSE↗TRUE | Teach current position into the selected profile. |
| wProfileNo<br>WORD<br><br>bProfileNo_1<br>...<br>bProfileNo_8<br>BOOL | Stipulation of the profile to be executed<br>• Optionally as a data word or binary coded.<br>• When the profile is stipulated, this FB carries out a mode change in the Lenze setting at the same time:<br>–If profile 0 is stipulated: Activation of "Speed follower" operating mode<br>–If profile 1 is stipulated: Activation of "Homing" operating mode<br>–If profile 2 is stipulated: Activation of "Manual jog" operating mode<br>–If profile 3 ... 15 is stipulated: Activation of "Positioning" operating mode | |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| wOutMckPosCtrl_1<br>wOutMckPosCtrl_2<br>WORD | Output of the MCK control word 1 & 2<br>• For transfer to the LS_MotionControlKernel system block.<br>• For a detailed description of the individual control bits, see chapter "MCK control word". (📖 481) | |
| bPosSetDataValid<br>BOOL | Status signal "Position conversion completed, data consistent"<br>• This output is permanently TRUE if the "0: dnPosOut_p=dnPosIn_p" mode has been set for calculating the position in C01296/1 since this is just a copy process and not a complex conversion. | |
| | TRUE | Conversion of the target position from [units] into [increments] has been completed.<br>• The travel profile data are valid and the profile is ready to start. |
| dnPosSetOut_p<br>DINT | Output of the target position in [increments]<br>• Observe the *bPosSetDataValid* status output! | |
| wFailState<br>WORD | Result of the plausibility check<br>• Display parameter: C01299<br>• Result is bit coded: | |
| | Bit 0 | Invalid operating mode selection<br>• "1" ≡ The selected operating mode is not defined/invalid. |
| | Bit 1 | Invalid positioning mode selection<br>• "1" ≡ The selected positioning mode is not defined/invalid. |
| | Bit 2 | Invalid profile number selection<br>• "1" ≡ The selected profile number refers to a profile data set that does not exist. |
| | Bit 3 | Reserved |
| | ... | |
| | Bit 15 | |
| bFail<br>BOOL | FALSE | Okay, no error |
| | TRUE | • Plausibility check error or<br>• control information error (in this case after ORing the individual signals with the control words) |

### 9.3.1.1 Alternative functions for control bit "PosExecute"

In C01297, alternative functions for bit 16 (PosExecute) in the MCK control word can be selected with bit-coding.

#### PosStop with PosExecute = FALSE

If bit 0 has been set in C01297, positioning can only be started/cancelled with the "PosExecute" control bit.

- If the "Positioning" operating mode has been set on the MCK, the "Pos-Execute" control bit has the following effects when being activated:

| Signals at the input | Signals in the control word to the MCK |
|---|---|
| bPosExecute = TRUE | bPosExecute = TRUE<br>bPosStop = FALSE<br>bHomingStartStop remains unchanged |
| bPosExecute = FALSE | bPosExecute = FALSE<br>bPosStop = TRUE<br>bHomingStartStop remains unchanged |

#### HomingStartStop with PosExecute

If bit 1 has been set in C01297, homing can only be started/stopped with the "PosExecute" control bit.

- If the "Homing" operating mode has been set on the MCK, the "PosExecute" control bit has the following effect when activated:

| Signals at the input | Signals in the control word to the MCK |
|---|---|
| bPosExecute = TRUE | bPosExecute = TRUE<br>bPosStop = FALSE<br>bHomingStartStop = TRUE |
| bPosExecute = FALSE | bPosExecute = FALSE<br>bPosStop remains unchanged<br>bHomingStartStop = FALSE |

#### SetProfilPosition with PosExecute

If bit 2 has been set in C01297 and control bit ("PosExecute") is being set, the setpoint position being applied is incorporated in the currently selected profile and then the profile is started immediately.

- If the "Positioning" operating mode has been set on the MCK, the "PosExecute" control bit has the following effect when activated:

| Signals at the input | Signals in the control word to the MCK |
|---|---|
| bPosExecute = TRUE | bPosExecute = TRUE<br>bPosStop = FALSE<br>bHomingStartStop remains unchanged<br>bPosTeachSetPos = TRUE (edge) |
| bPosExecute = FALSE | bPosExecute = FALSE<br>bPosStop remains unchanged<br>bHomingStartStop remains unchanged<br>bPosTeachSetPos = FALSE |

#### SetProfilPosition at position change

If bit 3 is set in C01297, the setpoint positions at the MCKInterface are automatically accepted into the profile with the applied profile number if a change of data is detected at the corresponding input for the setpoint position.

- If the "0: dnPosOut_p=dnPosIn_p" mode has been set in C01296/1 for converting the position, automatic acceptance is executed if the incremental position selection at the *dnPosIn_p* input was changed.
- If another mode (>0) has been set in C01296/1, automatic acceptance is executed if the incremental position selection at the *wPosUnitsLW* and *wPosUnitsHW* inputs was changed.
- From version 02.00.00 a hysteresis can be set in C01245/3 for the position change.

#### PosExecute at position change

(from version 02.00.00)

If bit 4 is set in C01297, an automatic "PosExecute" takes place if the incremental position selection changes and this change is higher than set in the hysteresis for position change (C01245/3).

If the selection is made in units, so that in a first step it has to be converted into increments, and if this automatic function is then activated, the "PosExecute" will only be created automatically when the internal conversion is completed (*bPosSetDataValid*).

**Tip!**

This option serves to start travel requests very easily by simply defining the new target position.

### 9.3.1.2 Operating mode change with profile number

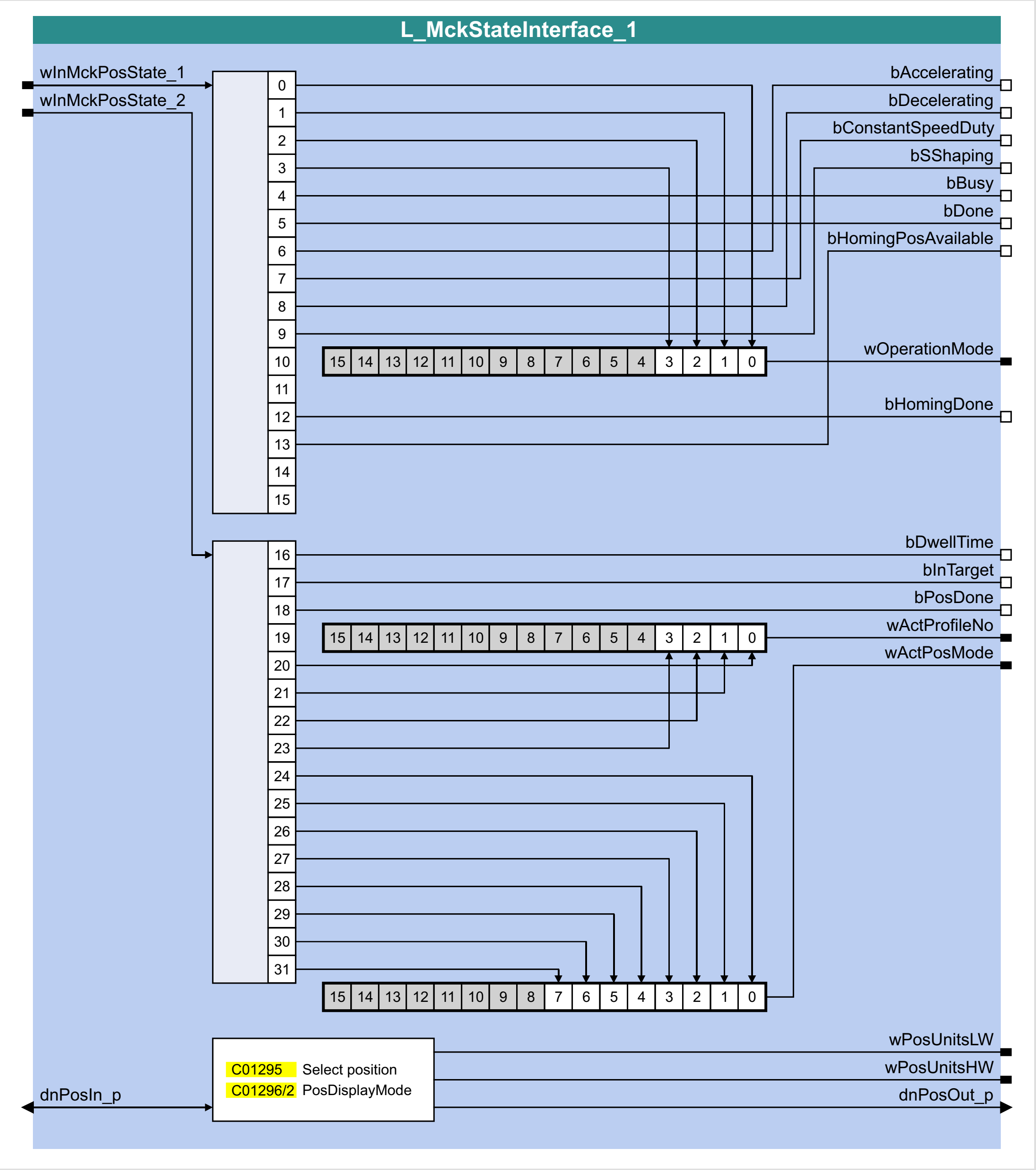
One of the operating modes of the **LS_MotionControlKernel** system block can be assigned to the positioning profiles via the four subcodes of C01298:

| Parameter | Info | Lenze setting |
|---|---|---|
| C01298/1 | Operating mode if profile 0 is selected | Speed follower |
| C01298/2 | Operating mode if profile 1 is selected | Homing |
| C01298/3 | Operating mode if profile 2 is selected | Manual jog |
| C01298/4 | Operating mode if profile 3 ... 15 is selected | Positioning |

- When a changeover to the corresponding profile is carried out, the set operating module is requested at the same time.
- In the case of the setting "0: No change", the operating mode is not changed when the profile changeover occurs. Instead, the operating mode requested "from outside" via the MCK control word is applicable.

### 9.3.2 Status outputs | FB "L_MckStateInterface"

This FB provides the application with different status information of the **Motion Control Kernel** via process outputs.



### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| wInMckPosState_1<br>wInMckPosState_2<br>WORD | Inputs for accepting the status words from the LS_MotionControlKernel system block. |
| dnPosIn_p<br>DINT | Position in [increments]<br>• Is converted into [units] if C01295 = "0: dnPosIn_p" and output at the *wPosUnitsLW* and *wPosUnitsHW* outputs.<br>• The mode for calculating the position is selected in C01296/2. |

### Outputs

| Identifier / Data type | Value/meaning | |
|---|---|---|
| bAccelerating<br>BOOL | TRUE | Acceleration phase active. |
| bDecelerating<br>BOOL | TRUE | Braking phase active. |
| bConstantSpeedDuty<br>BOOL | TRUE | Constant phase active. |
| bSShaping<br>BOOL | In preparation - output without function at the moment! | |
| | TRUE | Profile rounding active. |
| bBusy<br>BOOL | TRUE | Setpoint generation active. |
| bDone<br>BOOL | TRUE | Target position (setpoint) has been approached. |
| bHomingPosAvailable<br>BOOL | TRUE | Home position is known. |
| wOperationMode<br>WORD | Active setpoint-generating state of the **Motion Control Kernel**.<br>• Bit B0 ... B3 contain the information of the MCK status word.<br>• Bits B4 ... B15 are fixed at "0". | |
| bHomingDone<br>BOOL | TRUE | Homing has been executed. |
| bDwellTime<br>BOOL | TRUE | Settling in target position is active |
| bInTarget<br>BOOL | TRUE | Target position (actual value) is in the target window. |
| bPosDone<br>BOOL | TRUE | Target position from the profile has been approached. |
| wActProfileNo<br>WORD | Current traversing profile number<br>• Bit B0 ... B7 contain the information of the MCK status word.<br>• Bits B8 ... B15 are fixed at "0". | |
| wActPosMode<br>WORD | Current positioning mode<br>• Bit B0 ... B3 contain the information of the MCK status word.<br>• Bits B4 ... B15 are fixed at "0". | |
| wPosUnitsLW<br>wPosUnitsHW<br>WORD | Output of the position selected in C01295 in [units]<br>• *wPosUnitsLW* = LOW word, *wPosUnitsHW* = HIGH word<br>• The mode for calculating the position is selected in C01296/2. | |
| dnPosOut_p<br>DINT | Output of the position selected in C01295 in [increments]<br>• The mode for calculating the position is selected in C01296/2. | |

## 9.4 Basic settings

### 9.4.1 Machine parameters

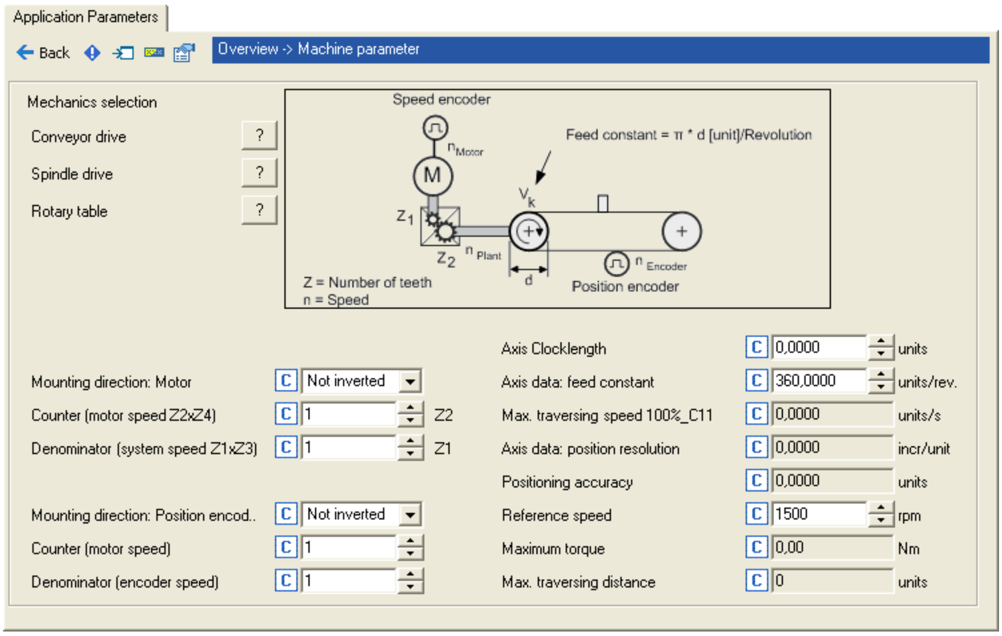
The motor end, among other things, is described by the machine parameters indicated below in respect of the mechanics used.

**Note!**

Setting the machine parameters is a basic prerequisite for the operating modes "Homing", "Manual jog" and "Positioning".

The more precisely the machine parameters are set, the better the results of positioning!

For TA "Table positioning", you can set the machine parameters in the »Engineer» on the tab headed **Application Parameters** on the dialog level *Overview* ➞ *Machine parameters*:




**Tip!**

You are provided with more detailed information on the machine parameters "Gearbox ratio" and "Feed constant" in the following subchapters.

**Short overview of machine parameters:**

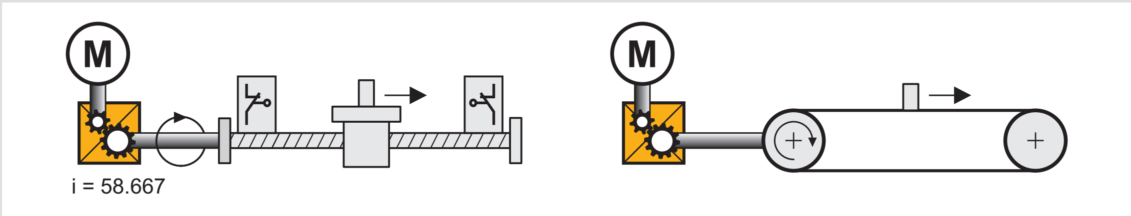
| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C01206/1 | Mounting direction: Motor | 0: Not inverted | |
| C01202/1 | iM: Nominator of gearbox factor Z2 | 1 | |
| C01202/2 | iM: Denominator of gearbox factor Z1 | 1 | |
| C01206/2 | Mounting direction: Position encoder | 0: Not inverted | |
| C01203/1 | iG: Nominator (motor speed) | 1 | |
| C01203/2 | iG: Denominator (encoder speed) | 1 | |
| C01201/1 | Axis data: Axis cycle<br>▸ Activation of the modulo measuring system | 0.0000 | units |
| C01204 | Axis data: Feed constant | 360.0000 | units/rev. |
| C00011 | Appl.: Reference speed | 1500 | rpm |

After the machine parameters have been entered, the drive controller sends back application-relevant data via the following display parameters:

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C01211/1 | Max. traversing speed 100%_C11<br>• Is used for orientation when the velocity is being set in the profile sets. | - | units/s |
| C01205 | Axis data: Position resolution<br>• Is needed for the incremental specification of positions. | - | incr/unit |
| C01210/5 | MCK: Positioning accuracy<br>• Theoretical accuracy during positioning with account being taken of the machine data and encoder data. | - | units |
| C00057 | Maximum torque | - | Nm |
| C01213/1 | MCK: Max. traversing distance | - | units |
| Highlighted in grey = display parameter | | | |

### 9.4.1.1 Gearbox ratio

The gearbox ratio indicates the number of rotations of the motor axis it takes for exactly one rotation of the load axis (e.g. spindle or drive roll) to take place.

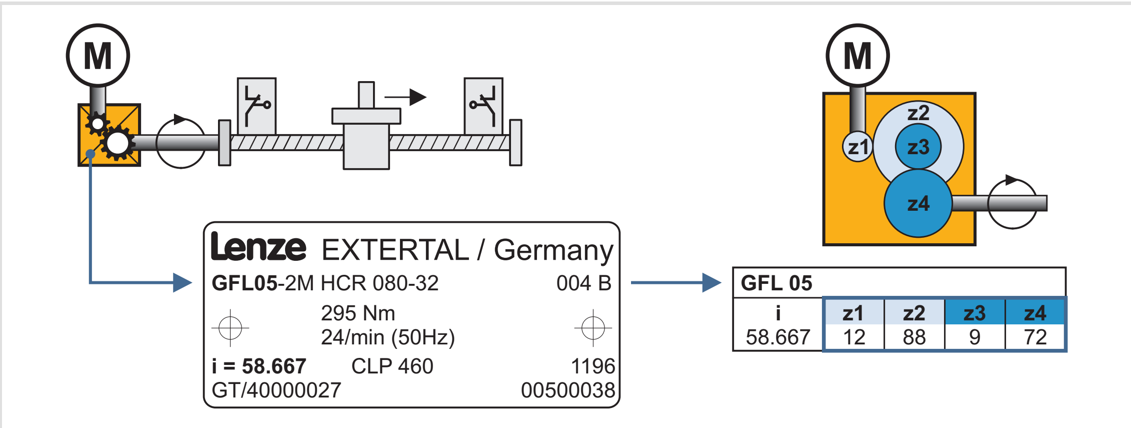



[9-5] Schematic diagram of gearbox ratio

- In the example shown in the illustration[9-5], the spindle rotates exactly once for every 58,667 rotations of the motor axis.

#### Specification of the gearbox ratio

- The gearbox ratio is to be specified in the form of a quotient (numerator/denominator). You can find the necessary data in the technical data for the gearbox:




[9-6] Example: Technical data relating to the gearbox (from gearbox catalogue)

**Tip!**

In order to specify the gearbox ratio exactly, use the number of teeth indicated on the data sheet or in the catalogue, if possible, instead of the information on the nameplate (see following calculation).

#### Calculation as an example based on the technical data relating to the gearbox:

Gearbox factor numerator (C01202/1) $= z2 \times z4 = 88 \times 72 = 6336$

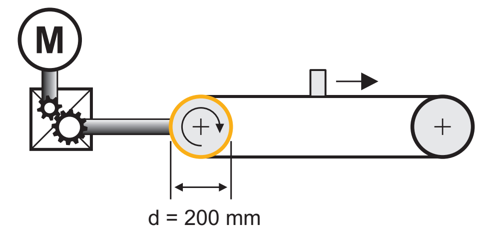
Gearbox factor denominator (C01202/2) $= z1 \times z3 = 12 \times 9 = 108$

[9-7] Calculation example (for 2-stage gearbox)

### 9.4.1.2 Feed constant

The feed constant corresponds to the machine movement for one revolution of the gearbox output shaft.

- The entry in the **Feed constant** (C01204) field is made in the user unit [units] in respect of the revolution.
- In the case of a conveyor drive, the feed constant is obtained from the drive roll's circumference, which, in the following example, is calculated on the basis of the indicated diameter:

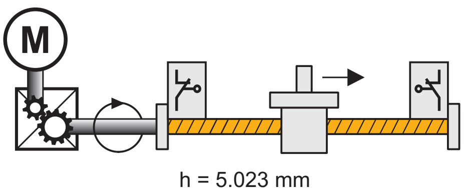



d = diameter

$$\text{Feed constant} = \pi \cdot d \frac{[\text{unit}]}{\text{Revolution}} = \pi \cdot 200 \frac{\text{mm}}{\text{Revolution}} = 628.3185 \frac{\text{mm}}{\text{Revolution}}$$

[9-8] Schematic diagram: Feed constant for a conveyor driver

- In the case of a spindle drive (linear axis), the feed constant is derived from the leadscrew pitch. The feed constant indicates the distance the slide travels during one revolution of the spindle (in the following example: 5.023 mm).



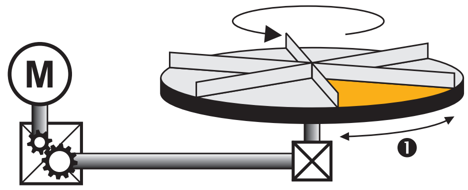

h = leadscrew pitch (to be obtained from the technical data of the linear axis)

[9-9] Schematic diagram: Feed constant for a spindle drive

- In the case of a rotary table and its specification as an angle, the feed constant is = 360°/revolution.

### 9.4.1.3 Activation of the modulo measuring system

The Modulo measuring system is also called "rotary table application".




❶ C01201/1: Cycle

[9-10] Example: Rotary table application

- The measuring system is repeated.
- If the set cycle is exceeded, a defined overflow occurs.
  - In a rotative system, the cycle usually corresponds to one revolution or tool distance.
- The home position must be known for positioning.
  - Exception: Positioning mode relative (TP) and continuous (TP)
- Software limit positions are not effective.
- Absolute targets can be approached by exceeding the measuring system limit, e.g. from 10° via 0° to 350°.

#### Activating the Modulo measuring system

The Modulo system is activated by setting a cycle (C01201/1) > 0 units.

- The cycle can be set if the controller is enabled.
- When the cycle (C01201/1) is set to 0 units (Lenze setting), the traversing range is unlimited (classical measuring system).

#### Generation of the Modulo measuring system

When the Modulo measuring system is active, it is displayed internally via an integrator. The Modulo position is provided at the *dnPosSet_p* process output of the SB LS_MotionControlKernel and displayed in C01210/7. When the Modulo measuring system is not active, the continuous (*dnPosSetValue_p*) setpoint position is output instead.

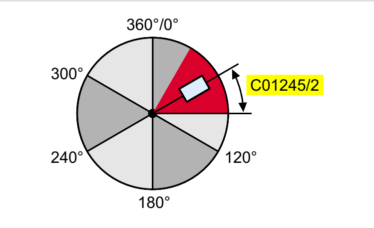
#### Blocking zone for "absolute (Cw)" and "absolute (Ccw)"positioning modes

Set a blocking zone in C01245/2 for the "absolute (Cw)" and "absolute (Ccw)" positioning modes in which no target positions are approached. This blocking zone around the current setpoint position serves to consider drift motions of the shaft in case of controller inhibit/enable cycles in order to prevent e.g. an unwanted motion by one cycle.

**Stop!**

Target positions in the set blocking zone are always approached with the shortest possible distance in the "absolute (Cw)" and "absolute (Ccw)" positioning modes!

If the blocking zone is too large, the drive may travel to the opposite direction!




[9-11] Definition of the blocking zone

- Target positions outside the blocking zone are approached with the selected positioning mode.
- The blocking zone is limited internally to half the cycle.

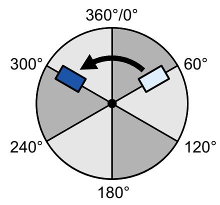
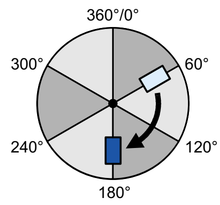
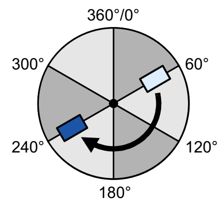
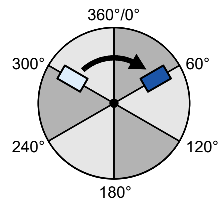
#### Display of the target position in the "Positioning" mode

| Positioning mode | Target position displayed in C01210/6 in [units] |
|---|---|
| Absolute (shortest path) | Defined position |
| absolute (Cw) | Defined position |
| absolute (Ccw) | Defined position |
| Continuous | 214748.3647 |
| Relative | Modulo position + defined position |

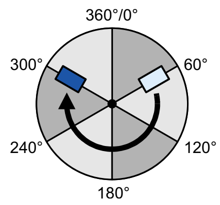
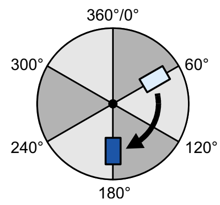
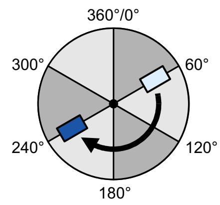
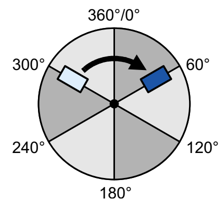
### Case studies for Modulo positioning

In the following, some case studies are shown in the various positioning modes. The target positions are defined by the user.

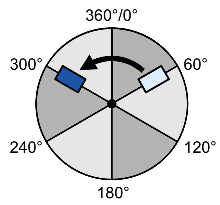
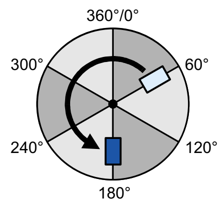
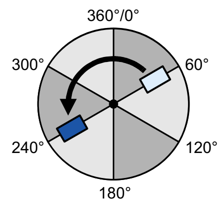
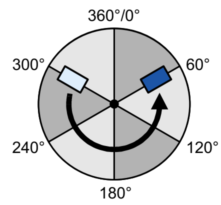
**"Absolute" positioning mode (shortest path)**

| Case 1 | Case 2 | Case 3 | Case 4 |
|---|---|---|---|
|  |  |  |  |

**"Absolute (Cw)" positioning mode**

| Case 1 | Case 2 | Case 3 | Case 4 |
|---|---|---|---|
|  |  |  |  |

**"Absolute (Ccw)" positioning mode**

| Case 1 | Case 2 | Case 3 | Case 4 |
|---|---|---|---|
|  |  |  |  |

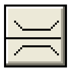
## 9.4.2 Min/Max speed

For the "Speed follower" operating mode, yo can initially limit the speed setpoint by means of the following parameters:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02610/2 | MCK: Ramp time synchr. setpoint | 2.000 | s |
| C02611/1 | MCK: Pos. max. speed | 199.99 | % |
| C02611/2 | MCK: Pos. min. speed | 0.00 | % |
| C02611/3 | MCK: Neg. min. speed | 0.00 | % |
| C02611/4 | MCK: Neg. max. speed | 199.99 | % |

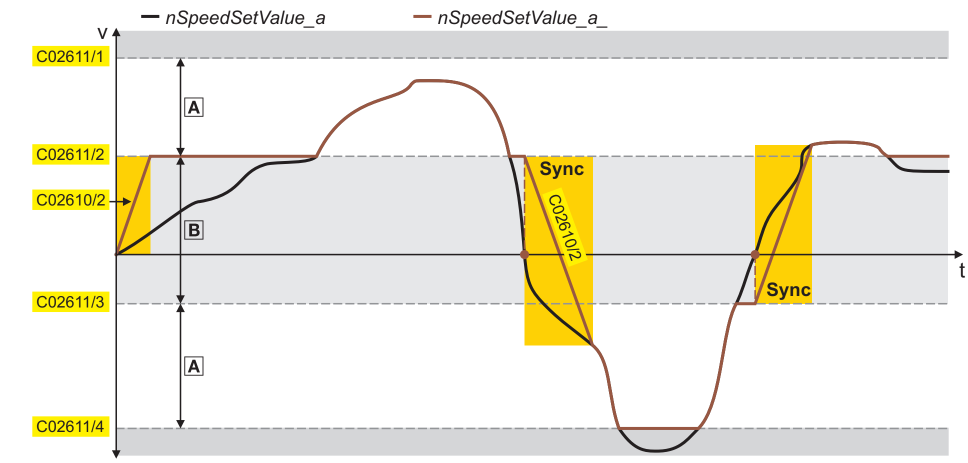
In the »Engineer», you can set the initial limit by means of the dialog box *Min/Max speed*.

▶ Open the *Min/max speed* dialog box by opening the **Application Parameters** tab and clicking on the following button on the dialog level *Overview* → *Signal flow*:



### Speed setpoint generation

When the speed limit values are set, the **Motion Control Kernel** influences the setpoint generation with a synchronisation mode The synchronisation mode serves to travel the synchronisation range dynamically with the synchronisation ramp set in C02610/2. Synchronisation always starts in the zero crossing of the defined speed.




A Setpoint range
B Synchronisation range

[9-12] Example: Speed setpoint generation in the "Speed follower" operating mode (with nSpeedAddValue_v = 0)

### 9.4.3 Limit position monitoring

#### 9.4.3.1 Software limit positions

The parameterisable limit positions are used by the software to limit the traversing range.

▶ The positive software limit position is set in C01229/1 and the negative software limit position is set in C01229/2.

**Stop!**

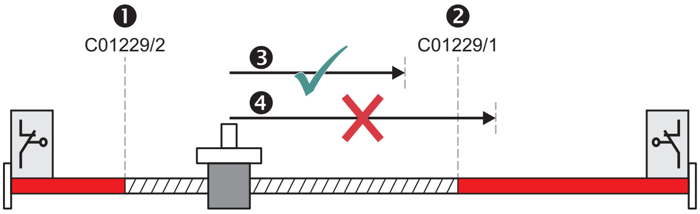
The software limit positions are only evaluated and monitoring if the drive knows the home position and the software limit positions for the respective operating mode have been activated.

- For the "manual jogging" operating mode, travel range monitoring is switched off by means of parameterisable software limit positions in the Lenze setting of C01230.
- In the "Homing" and "Positioning" operating modes, the travel range monitoring is generally switched-on.

| Parameter | Info | Lenze setting |
|---|---|---|
| C01218 - Bit 3 | SW limit positions for position follower on/off | On |
| C01219 - bit 3 | SW limit positions for speed follower on/off | On |
| C01230 - bit 3 | SW limit position for manual jogging on/off | Off |

### Behaviour in the case of active software limit positions

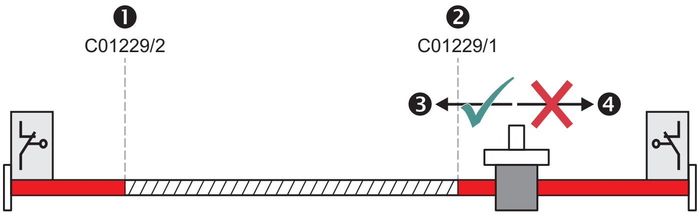
- If the software limit positions are active, travelling commands that would result in exiting from the permissible travel range can no longer be executed:




❶ C01229/2: Negative software limit position
❷ C01229/1: Positive software limit position
❸ Travel command possible
❹ Travel command not possible because target position is outside the permissible traversing range.

[9-13] Example: Traversing range limitation by means of software limit positions

- If the drive is already outside the permissible travel range and the software limit positions have been activated, only travel commands that result in the drive moving back into the permissible travel range can be executed:




❶ C01229/2: Negative software limit position
❷ C01229/1: Positive software limit position
❸ Traversing in "retracting direction" possible
❹ Traversing opposite to "retracting direction" not possible

[9-14] Example: Permissible traversing direction if software limit positions active

- If the software limit positions are active and a software limit position is passed ("overtravel"):
  - The error response "TroubleQuickStop" takes place in the Lenze setting, i.e. the drive is brought to a standstill in the deceleration time set for the quick stop function and does so irrespective of the setpoint selection. The error response can be parameterised in C00595/3 and C00595/4.
  - The error message "Ck03: Pos. SW limit position" or "Ck04: Neg. SW limit position" is entered in the logbook of the drive controller.
  - Depending on the parameterised error response, the drive cannot traverse until the error has been acknowledged.

### 9.4.3.2 Hardware limit switch

The travel range limits are monitored by means of limit switches via the inputs *bLimitSwitchPos* and *bLimitSwitchNeg* of the LS_MotionControlKernel system block.

- The two inputs react to the TRUE state.
- In TA "Table positioning", the two inputs are connected to the digital inputs provided for connection of the limit switches.

**Stop!**

The limit switches are only evaluated if the limit switches for the respective operating mode have been activated (see the following table)!

| Operating mode | Hardware limit switch effective |
|---|---|
| Speed follower | Yes (adjustable in C01219 - bit 2) |
| Homing | Depending on the selected homing mode (see description of the homing modes) |
| Manual jog | No (adjustable in C01230 - bit 2) |
| Positioning | Yes |
| Stop | Yes |
| Position follower | Yes (adjustable in C01218 - bit 2) |

**Note!**

If the digital inputs used for connecting the limit switches are fail-safe inputs (tripping at LOW level), you simply change the terminal polarity of the corresponding digital inputs in C00114.

#### Behaviour when hardware limit switches are active

- If one of the two monitoring inputs is set to TRUE:
  - The error response "TroubleQuickStop" takes place in the Lenze setting, i.e. the drive is brought to a standstill in the deceleration time set for the quick stop function and does so irrespective of the setpoint selection. The error response can be parameterised in C00595/1 and C00595/2.
  - The error message "Ck01: Pos. HW limit switch" or. "Ck02: Neg. HW limit switch" is entered in the logbook.
  - Bit 10 ("Pos. HW-Limit Detected") or bit 11 ("Neg. HW-Limit Detected") is set in the MCK status word.
  - Depending on the parameterised error response, the drive cannot traverse until the error has been acknowledged.

**Note!**

An activated limit switch can be retracted again by manual jog in the opposed direction or with the "Retract limit switch" function. ▶ Retracting of an operated limit switch (📖 540)

Only in the "Manual jog" operating mode, retracting of the limit switch resets bit 10 ("Pos. HW-Limit Detected") or bit 11 ("Neg. HW-Limit Detected") in the MCK status word.

### Re-activation after acknowledging the error

When the error has been acknowledged and the limit switch is still active, the following action is required for a renewed activation of the monitoring depending on the operating mode:

| Operating mode | Action for (re) activation |
|---|---|
| Speed follower | Limit switch is activated and setpoint in direction of the activated limit switch is pending |
| Homing | Limit switch is activated and *bHomStartStop* = TRUE |
| Manual jog | Limit switch is activated and manual jog in direction of the activated limit switch |
| Positioning | Limit switch is activated and setpoint command has been transmitted. |
| Stop | Setting of the operating mode |
| Position follower | Limit switch is activated and setpoint in direction of the activated limit switch is pending |

### 9.4.4 Target position monitoring (status "drive in target")

The target position monitoring detects whether the drive has reached the target.

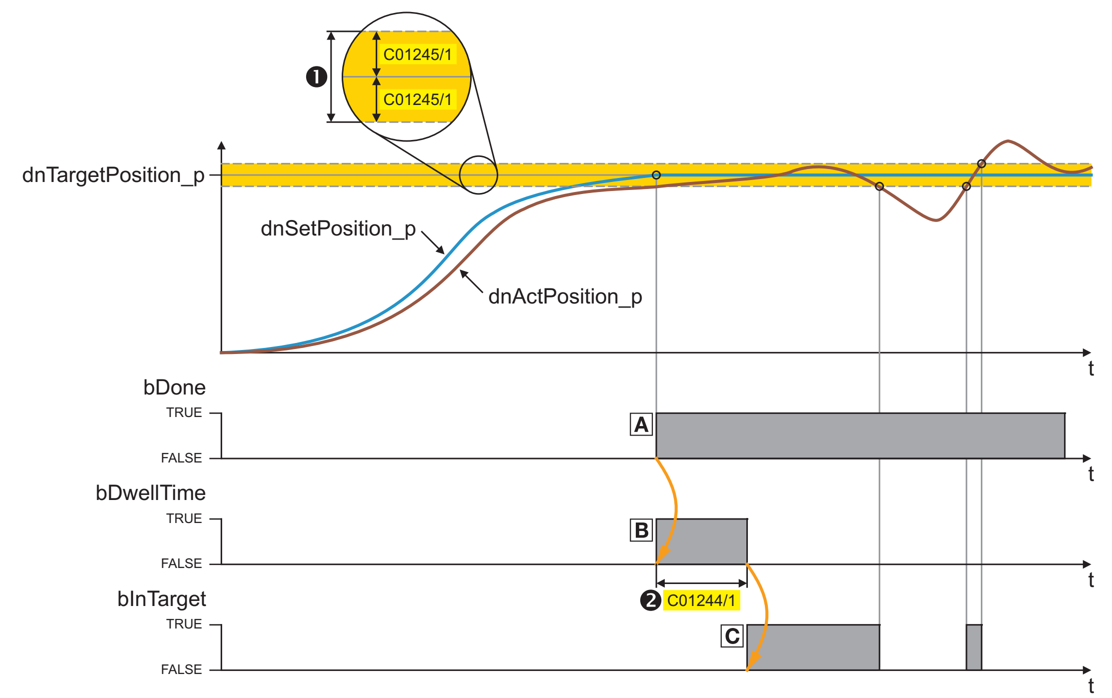
**Note!**

- The target position detection is active with the following operating modes/functions:
  – Operating mode "Positioning": Absolute and relative positioning mode
  – Operating mode "Homing": After approaching the starting position via start profile (if start profile has been selected)
  – Operating mode "Manual jog": When the breakpoints and the software limit positions are approached
- The target position monitoring is active until the operating mode changes or a new travel command influences the setpoint.

**Short overview of the parameters for target position monitoring:**

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01244/1 | MCK: Dwell time - target position<br>• For considering mechanical transient phenomenons of the tool when the target position is reached. | 100 | ms |
| C01245/1 | MCK: Window target position<br>• Symmetrical window around the target position. | 1.0000 | units |

### Principal procedure of the position detection




❶ target position window (symmetrically around the target position; 2 x C01245/1)
❷ dwell time

[9-15] Signal characteristics

### Status "Done" (A)

- The position setpoint has reached the target position.

### Status "DwellTime" (B)

- Bit 16 in the MCK status word or the *bDwellTime* output of the L_MckStateInterface FB is set when the setpoint position has been reached after the dwell time set in C01244/1 has expired.
- The status detects the time after the *bDone* signal when settling to target position is executed and actual position detection is not active yet.

### Status "InTarget" (C)

- If the actual position is within the symmetrical target position window after the dwell time has expired, bit 17 is set to "1" in the MCK status word or the *bInTarget* output of the L_MckStateInterface FB is set to TRUE.

**Note!**

- The target position including the target position window has to be within the maximum traversing range.
- Please observe the following if you use *bInTarget* as stepping condition in a step sequence:
The *bInTarget* signal needs the set position to be located in the target. A profile restart immediately deletes the setpoint position in the first cycle and thus also resets *bInTarget*.

▶ The "InTarget" status is reset if:
  – The motor shaft leaves the target position window,
  – a new profile or a new motion process is started,
  – *bInTarget* has been set via manual jog (intermediate stops) and the "Manual jog" mode is left,
  – the target position window has been left in the "Speed follower" mode and a speed setpoint not equal to "0" is injected into the process,
  – a new homing process is started,
  – the reference is set,
  – the device is switched off/on.

### 9.4.5 Monitoring of the maximum travel distance

Continuous travel requests in the "Positioning" operating mode (or relative positioning with feed in the same direction) cause an overflow of the position integrators when the max. display area of the position is reached and the reference is set. The same behaviour takes place in the "Speed follower" and "Manual jog" operating modes. For this reason, a monitoring mode of the maximum travel distance is implemented in the **Motion Control Kernel**.

▶ If the maximum travel distance (C01213/1) is exceeded:
  – The error response set in C00595/7 will be carried out (Lenze setting: "TroubleQuickStop").
  – The "Ck07: Travel range limit exceeded" error message is entered into the logbook.
  – The internal status "reference known" is reset (the controller is no longer referenced).

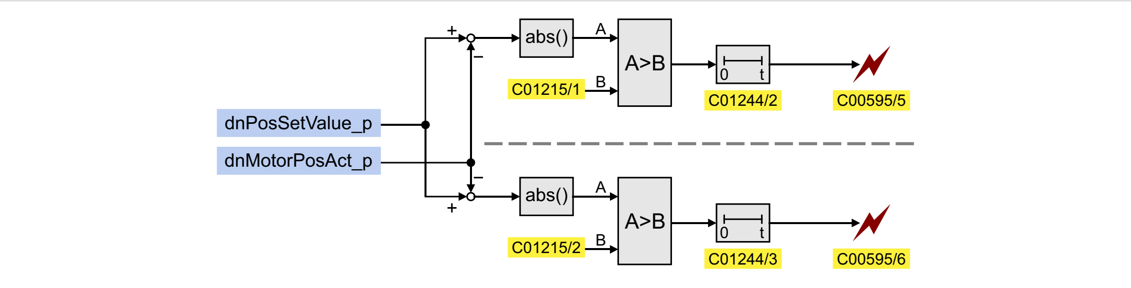
▶ A setting of C00595/7 = "0: No Reaction" deactivates the monitoring.

### 9.4.6 Following error monitoring system

The difference between set position and actual position is called the following error. Ideally, the following error should be "0". The set position is created by the internal definition of the traversing profiles of the **Motion Control Kernel**. The actual position is created by the integration of the speed supplied by the position encoder. If the position control is adjusted optimally, only a mininum following error arises which is always compensated dynamically and not increases continuously.

Certain processes, however, require that a defined limit as a difference between set position and actual position is not exceeded. If it is exceeded, it may have been caused by a mechanical blocking in the machine and the system part is not situated at the position defined at that time. In such a case, it makes sense to activate the "Fault" error response to make the motor torqueless.

In the 8400 TopLine controller, two independent following error monitoring systems can be parameterised:




[9-16] Two-channel following error monitoring system

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01215/1 | MCK: Following error limit 1 | 0.0000 | units |
| C01215/2 | MCK: Following error limit 2 | 0.0000 | units |
| C01244/2 | MCK: Following error deceleration 1 | 0.000 | s |
| C01244/3 | MCK: Following error deceleration 2 | 0.000 | s |
| C00595/5 | MCK: Resp. to following error 1 | Warning | |
| C00595/6 | MCK: Resp. to following error 2 | Warning | |

**Note!**

If the limit for the following error is set to "0.0000 units" (Lenze setting, the following error monitoring system is not active.

**Tip!**

In certain situations (e.g. dynamic acceleration of the load), higher system-dependent following errors occur than while approaching the target position.

In order that no error is triggered during acceleration and a close tolerance limit can be monitored all the same at standstill in the target, the addressing of the following error monitoring system can be decelerated. Thus, dynamic processes or torque impulses occurring for short periods can be "masked out".

### Operating mode

If the limit for the following error in C01215/x is set higher than "0.0000 units" and if the current following error exceeds this limit over the time set in C01244/x:

- The error response set in C00595/5 or C00595/6 will be carried out (Lenze setting: "Warning").
- The error message "Ck05: Error, following error 1" or "Ck06: Error, following error 2" will be entered into the logbook.

## 9.5 Speed follower

In the "speed follower" operating mode, the drive follows a speed setpoint.

### 9.5.1 Parameter setting

**Short overview of parameters for the "speed follower" operating mode:**

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01219 | MCK: Speed follower setting | Bit coded | |
| C02610/2 | MCK: Ramp time synchr. setpoint | 2.000 | s |
| C02611/1 | MCK: Pos. max. speed | 199.99 | % |
| C02611/2 | MCK: Pos. min. speed | 0.00 | % |
| C02611/3 | MCK: Neg. min. speed | 0.00 | % |
| C02611/4 | MCK: Neg. max. speed | 199.99 | % |

#### 9.5.1.1 Functional settings

In C01219, various functional settings for the speed follower can be made in bit-coded form.

| Function | | Lenze setting |
|---|---|---|
| Bit 0 | Reserved | Off |
| Bit 1 | Reserved | Off |
| Bit 2 | **HW limit switch on**<br>In the "speed follower" operating mode, a travel range monitoring mode via hardware limit switch is active.<br>▸ Limit position monitoring (📖 506) | On |
| Bit 3 | **SW limit switch on**<br>In the "speed follower" operating mode, a travel range monitoring mode via parameterised software limit positions.<br>▸ Limit position monitoring (📖 506) | On |
| Bit 4 | Reserved | Off |
| Bit 5 | Reserved | Off |
| Bit 6 | Reserved | Off |
| Bit 7 | **Position controller off**<br>In the "Speed follower" operating mode, the position controller is deactivated. Thus, the compensation of the following error is switched off. | Off |

### 9.5.2 Requesting the operating mode

Request for "speed follower" operating mode by means of the MCK control word:

MCK control word

| Bit 31 | ... | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 0 | 0 | 0 |

X = Status not significant

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.5.3 Setpoint selection

The speed setpoint is selected via the *nSpeedSetValue_a* process input and additively via the *nSpeedAddValue_v* process input.

- Usually, the ramp generator L_NSet and, optionally, the process controller L_PCTRL are upstream of the *nSpeedSetValue_a* process input.
- The speed setpoint is limited internally to the speed limits set in C02611/1...4.

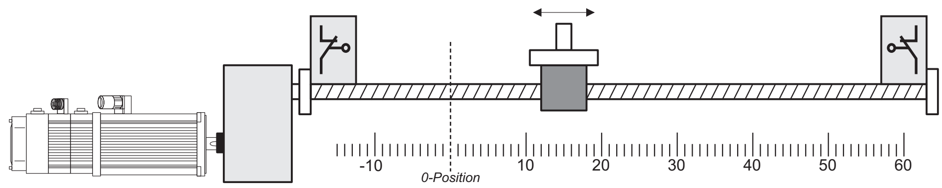
**Note!**

When the speed limit values are set, the **Motion Control Kernel** influences the setpoint generation with a synchronisation mode The synchronisation mode serves to travel the synchronisation range dynamically with the synchronisation ramp set in C02610/2.

Min/Max speed (📖 505)

## 9.6 Homing

The measuring system in the machine is selected by means of homing and the 0 position is set within the possible physical travel range:




[9-17] Homing (selection of 0 position)

The zero position (home) can be defined by a reference run or by setting a home position:

- If a reference run is carried out, the drive follows a previously selected path to find the home position.
- When the reference is set, it is selected manually when the drive stands still.

**Danger!**

During homing, specially assigned profile parameters are active. If they have not been set correctly, the drive can engage in uncontrolled movement!

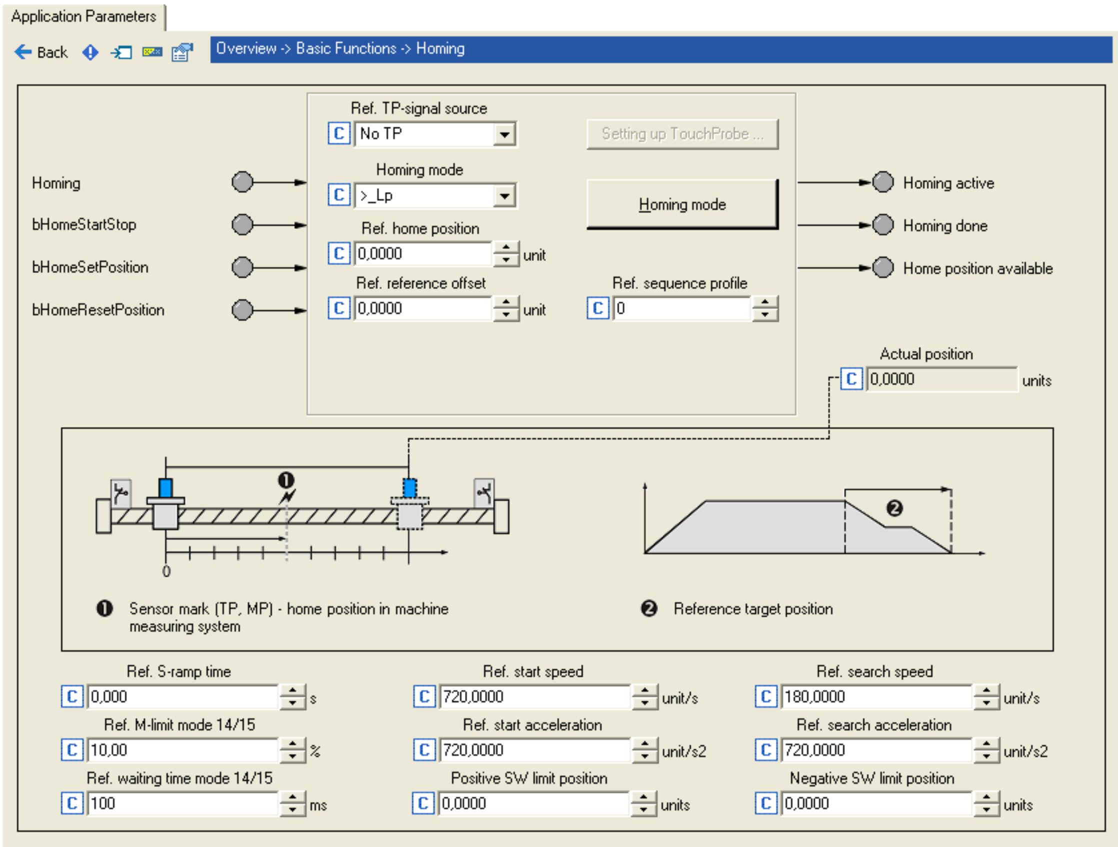
**Tip!**

A reference run is mainly used for systems with feedback via encoders, resolvers, or single-turn absolute value encoders, as in the case of these systems the home position is lost when the supply voltage is switched off.

Usually the reference is set only once during commissioning or in the event of service (e. g. if drive components are replaced), and it is mostly used for systems with feedback via absolute value encoders.

## 9.6.1 Parameter setting

### Parameterisation dialog in the »Engineer«




### Short overview of parameters for "Referencing" operating mode:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01221 | MCK: Ref. mode | 12: >_Lp | |
| C01224/1 | MCK: Ref. initial speed | 720.0000 | unit/s |
| C01225/1 | MCK: Ref. initial acceleration | 720.0000 | unit/s2 |
| C01224/2 | MCK: Ref. search speed | 180.0000 | unit/s |
| C01225/2 | MCK: Ref. search acceleration | 720.0000 | unit/s2 |
| C01226/1 | MCK: Ref. S-ramp time | 0.000 | s |
| C01222 | MCK: Ref. M limit mode 14/15 | 10.00 | % |
| C01223 | MCK: Ref. waiting time mode 14/15 | 100 | ms |
| C01227/1 | MCK: Ref. offset reference degree | 0.0000 | unit |
| C01227/2 | MCK: Ref. home position | 0.0000 | unit |
| C01228 | MCK: Ref. sequence profile | 0 | |
| C01229/1 | MCK: Positive SW limit position | 0.0000 | units |
| C01229/2 | MCK: Negative SW limit position | 0.0000 | units |
| C01246/1 | MCK: Ref. TP signal source | 0: No TP | |
| C01246/2 | MCK: Set.Ref. signal source | 0: No TP | |

### 9.6.1.1 Referencing mode

Specify the referencing mode in C01221 , i.e. the way in which referencing is to take place.

- For reference setting, the referencing mode "100" is to be selected in C01221.
- For a reference search, C01221 contains referencing modes "6"..."15" which can be selected from.

| Referencing mode C01221 | Evaluated signals/sensors | | | |
|---|---|---|---|---|
| | Sensor reference signal | Travel range limit switch | | Pre-stop mark at *bHomingMark* |
| | | Negative limit switch | Positive limit switch | |
| 6 | ☑ | | | ☑ |
| 7 | ☑ | | | ☑ |
| 8 | ☑ | | | |
| 9 | ☑ | | | |
| 10 | ☑ | | ☑ | |
| 11 | ☑ | ☑ | | |
| 12 | | | ☑ | |
| 13 | | ☑ | | |
| 14 | Positive direction of rotation to torque limit. | | | |
| 15 | Negative direction of rotation to torque limit. | | | |
| 100 | Set reference directly. | | | |

#### Internal interfaces

The switches/sensors are evaluated via the following internal interfaces:

| Switch/sensor | Internal interface for digital input signal |
|---|---|
| Touch probe sensor<br>(Sensor reference signal) | • In the Lenze setting, a signal pending at the digital input DI3 is interpreted level-sensitive as TP sensor signal.<br>–In this case, the search speed must be selected in order that the signal at DI3 will be detected.<br>• Moreover, the high-precision homing with touch probe is supported.<br>–The touch probe signal source can be selected in C01246/1.<br>–If the reference signal is to follow a real touch probe, configure the touch probe signal accordingly. ▶ Touch probe detection (📖 362) |
| Positive travel range limit switch | *bLimitSwitchPos* |
| Negative travel range limit switch | *bLimitSwitchNeg* |
| Pre-stop mark/pre-stop signal | *bHomingMark*<br>• This input responds to the FALSE status and is to be connected to the corresponding digital input to which the reference switch is connected.<br>• The response to the pre-switch off signal depends on the referencing mode selected. |

**Note!**

For a reference search with touch probe detection:

- The touch probe signal source can be selected in C01246/1.
- Go to C02810/x and select the edge, the digital input used for the connection of the touch-probe sensor is to respond. In the Lenze setting of C02810/x, no touch probe is detected!
- Moreover, no "acceptance window" must be set in C2813/x and C02814/x for accepting the touch probe signal in order that a touch probe will always be detected independent of the position.

▶ Touch probe detection (📖 362)

### Start and search profile data set

Certain referencing modes use two different profile data sets for homing in order to shorten the homing time and, at the same time, increase accuracy.

- ▶ At first, a quick approach of the limit switch/pre-stop mark (depending on the selected mode) is carried out using the start profile data set.
- ▶ After reversing at the limit switch/pre-stop mark, the search profile data set results in slower – but more accurate – approaching of the touch probe sensor.

| Start profile data set | | Search profile data set | |
|---|---|---|---|
| C01224/1 | Start speed | C01224/2 | Search speed |
| C01225/1 | Start acceleration<br>(deceleration as well) | C01225/2 | Search acceleration<br>(deceleration as well) |
| C01226/1 | S-ramp time<br>(identical in the two profile data sets) | C01226/1 | S-ramp time<br>(identical in the two profile data sets) |

**Note!**

A changeover to the search profile data set is only carried out of the search speed has been set to (C01224/2) > "0"!

The exact time of the changeover to search profile data set occurs in the respective referencing mode is indicated in the process descriptions of the referencing modes.

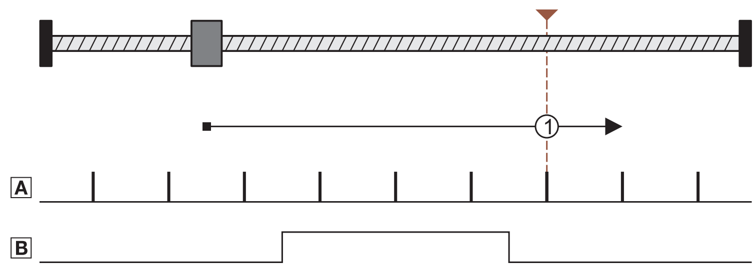
**Tip!**

The result of setting a lower search speed in C01224/2 and a high search acceleration in C01225/2 is that deceleration to search speed takes place quickly and position detection is exact (at slower search speed).

### Abbreviations used for the referencing modes:

| Abbreviation | Meaning |
|---|---|
| > | Movement in positive direction |
| < | Movement in negative direction |
| Ln | Negative travel range limit switch |
| Lp | Positive travel range limit switch |
| Rn | Negative edge - pre-stop mark/pre-stop signal |
| TP | Touch probe or sensor reference signal/reference switch detected |
| Mlim | Torque limit value reached |

### Mode 6: >_Rn_>_TP



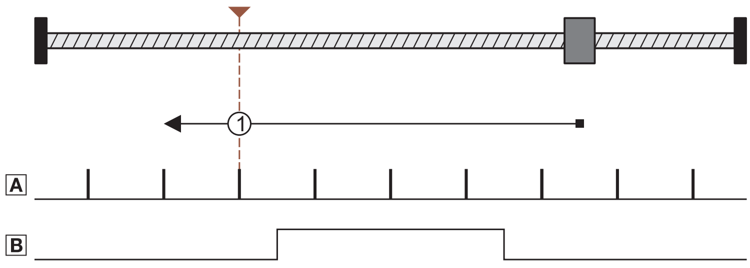

A Touch probe
B Pre-stop mark

| Mode 6 | > | Rn | > | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| **Lp** | Active | Active | Active | Active | Active | Active |
| **Ln** | Inactive | Inactive | Inactive | Inactive | Active | Active |

Functional sequence:

1. Movement in positive direction with start profile data set.
2. Positive edge at *bHomingMark* activates search profile data set for further reference search.
3. Negative edge at *bHomingMark* enables home position detection.
4. Following edge of the touch probe sensor sets home position.

### Mode 7: <_Rn_<_TP



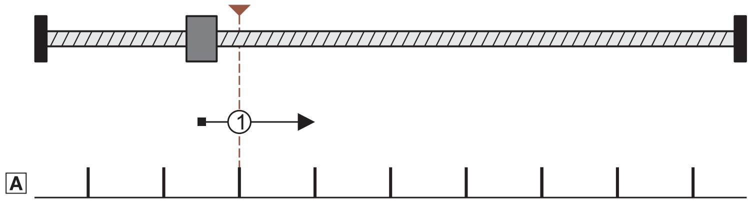

A Touch probe
B Pre-stop mark

| Mode 7 | < | Rn | < | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| Lp | Inactive | Inactive | Inactive | Inactive | Active | Active |
| Ln | Active | Active | Active | Active | Active | Active |

Functional sequence:

1. Movement in negative direction with start profile data set.
2. Positive edge at *bHomingMark* activates search profile data set for further reference search.
3. Negative edge at *bHomingMark* enables home position detection.
4. Following edge of the touch probe sensor sets home position.

### Mode 8: >_TP



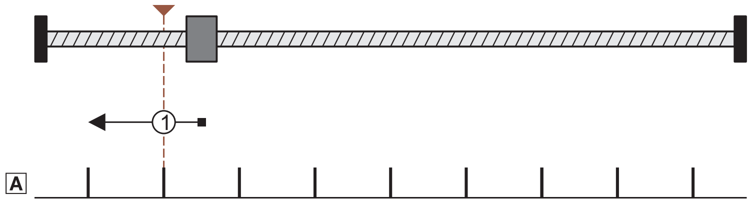

A Touch probe

| Mode 8 | > | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| Lp | Active | Active | Active | Active |
| Ln | Inactive | Inactive | Active | Active |

Functional sequence:

1. Movement in positive direction with start profile data set.
2. Following edge of the touch probe sensor sets home position.

### Mode 9: <_TP



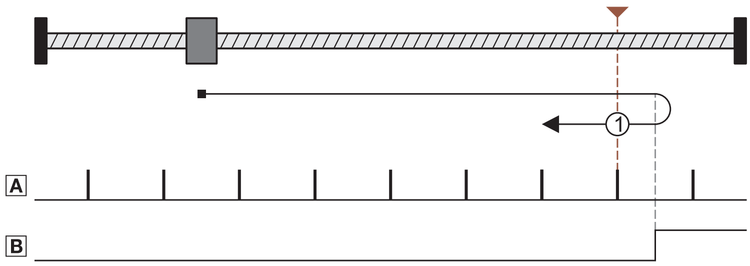

A Touch probe

| Mode 9 | < | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| Lp | Inactive | Inactive | Active | Active |
| Ln | Active | Active | Active | Active |

Functional sequence:

1. Movement in negative direction with start profile data set.
2. Following edge of the touch probe sensor sets home position.

### Mode 10: >_Lp_<_TP



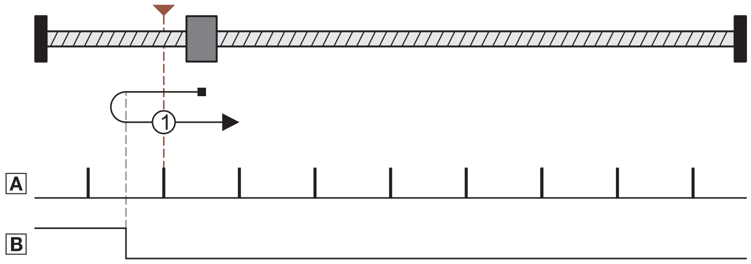

A Touch probe
B Positive travel range limit switch

| Mode 10 | > | Lp | < | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| **Lp** | Inactive | Inactive | Inactive | Inactive | Inactive | Active |
| **Ln** | Inactive | Inactive | Active | Active | Active | Active |

Functional sequence:

1. Movement in positive direction with start profile data set.
2. Reversing when the edge of the positive travel range limit switch is positive and, at the same time, activation of search profile data set for continued reference searching.
3. Negative edge of the travel range limit switch enables home position detection.
4. Following edge of the touch probe sensor sets home position.

## Mode 11: <_Ln_>_TP



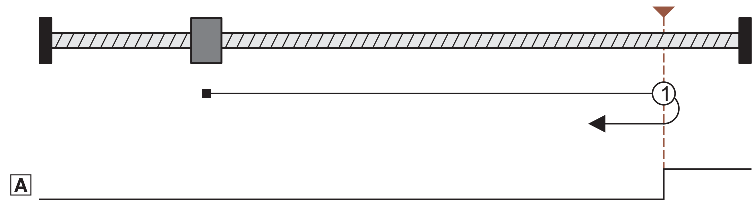

A Touch probe
B Negative travel range limit switch

| Mode 11 | < | Ln | > | TP | Offset path | Sequence profile (optional) |
|---|---|---|---|---|---|---|
| | Start speed | | Search speed | | | Profile speed |
| **Lp** | Inactive | Inactive | Active | Active | Active | Active |
| **Ln** | Inactive | Inactive | Inactive | Inactive | Inactive | Active |

Functional sequence:

1. Movement in negative direction with start profile data set.
2. Reversing when the edge of the negative travel range limit switch is positive and, at the same time, activation of search profile data set for continued reference searching.
3. Negative edge of the travel range limit switch enables home position detection.
4. Following edge of the touch probe sensor sets home position.

### Mode 12: >_Lp




A Positive travel range limit switch

| Mode 12 | > | Lp | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| Lp | Inactive | Inactive | Inactive | Active |
| Ln | Inactive | Inactive | Active | Active |

Functional sequence:

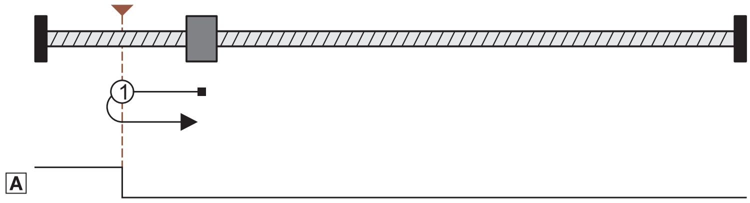
1. Movement in positive direction with start profile data set.
2. Positive edge of the travel range limit switch sets reference.

**Note!**

The load machine can also leave the travel range limit switch. There follows a return to the home position that was set with the positive edge of the travel range limit switch.

- It is possible that, as a result, the machine will remain on an operated limit switch.
- It is therefore recommended that a home value offset be set in C01227/1 in order to release the operated limit switch.

### Mode 13: <_Ln




A Negative travel range limit switch

| Mode 13 | < | Ln | Offset path | Sequence profile (optional) |
| --- | --- | --- | --- | --- |
| | Start speed | | | Profile speed |
| **Lp** | Inactive | Inactive | Active | Active |
| **Ln** | Inactive | Inactive | Inactive | Active |

Functional sequence:

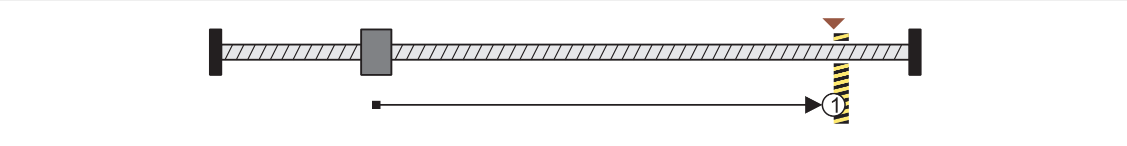
1. Movement in negative direction with start profile data set.
2. Positive edge of the travel range limit switch sets reference.

**Note!**

The load machine can also leave the travel range limit switch. There follows a return to the home position that was set with the positive edge of the travel range limit switch.

- It is possible that, as a result, the machine will remain on an operated limit switch.
- It is therefore recommended that a home value offset be set in C01227/1 in order to release the operated limit switch.

### Mode 14: >_Mlim



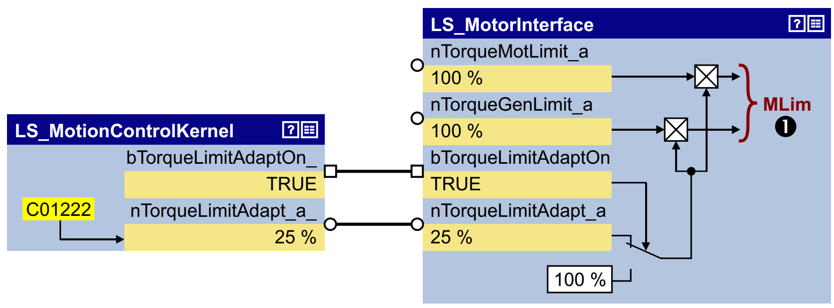
| Mode 14 | > | Mlim | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| **Lp** | Active | Active | Active | Active |
| **Ln** | Inactive | Inactive | Active | Active |

<u>Functional sequence:</u>

1. Movement in a positive direction with reduced torque and start profile data set.
2. The reference is set if it is detected that the torque limit set in C01222 is exceeded beyond the time set in C01223 ("Homing to positive stop").
   – If a home value offset has not been set, this position is the home position.
   – If a home value offset has been set, correctly signed traversing by this offset takes place and the home position is set at the end of this travel distance.

**Tip!**

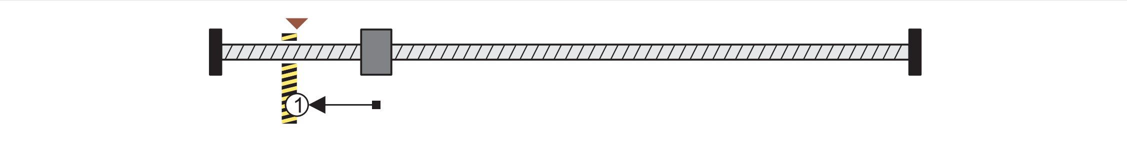
Set a home value offset in C01227/1 to avoid stopping at the positive stop.




❶ Effective torque limit

[9-18] Wiring for torque limit when homing to "Positive stop" (homing modes 14/15)

### Mode 15: <_Mlim




| Mode 15 | < | Mlim | Offset path | Sequence profile (optional) |
|---|---|---|---|---|
| | Start speed | | | Profile speed |
| Lp | Inactive | Inactive | Active | Active |
| Ln | Active | Active | Active | Active |

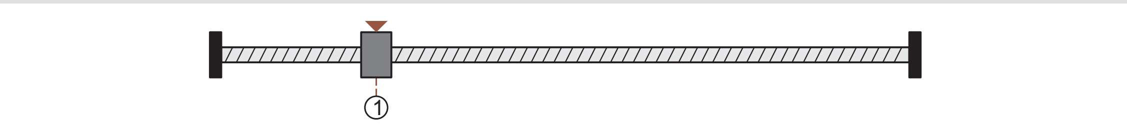
Functional sequence:

1. Movement in a negative direction with reduced torque and start profile data set.
2. The reference is set if it is detected that the torque limit set in C01222 is exceeded beyond the time set in C01223 ("Homing to positive stop").
   – If a home value offset has not been set, this position is the home position.
   – If a home value offset has been set, correctly signed traversing by this offset takes place and the home position is set at the end of this travel distance.

**Tip!**

Set a home value offset in C01227/1 to avoid stopping at the positive stop.

### Mode 100: SetRef




| Mode 15 | SetRef | Offset path | Sequence profile (optional) |
|---|---|---|---|
| | - | Start speed | Profile speed |
| Lp | Active | Active | Active |
| Ln | Active | Active | Active |

When the drive is at standstill, the measuring system is set via the bit 9 ("HomSetPos") in MCK control word. The current actual position now corresponds to the home position set in C01227/2 in the machine measuring system.

### 9.6.1.2 Home position & home value offset

If the home position is set in the course of the reference search, this determined position in the machine measuring system now corresponds to the home position set in C01227/2.

If a home value offset has been set in C01227/1, the drive continues correctly signed traversing by this offset at search speed after the home position has been found. The home position is not set until after this movement.

### 9.6.1.3 Traversing a sequence profile after completion of homing

If necessary, a sequence (following or subsequent) profile can be entered in C01228 in order to arrange for a positioning movement to be carried out immediately after completion of homing.

**Note!**

The positioning movement for the sequence profile set in C01228 is also carried out in the "referencing" mode. If a second sequence (following or subsequent) profile has been defined in the sequence profile, it is not automatically started!

## 9.6.2 Requesting the operating mode

Request for the "referencing" operating mode by means of the MCK control word:

| MCK control word | | | | | | |
|---|---|---|---|---|---|---|
| **Bit 31** | **...** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 0 | 0 | 1 |
| X = Status not significant | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.6.3 Carrying out homing

Control is exercised by means of bits 8 ... 10 in the MCK control word:

| Homing | MCK control word | | |
|---|---|---|---|
| | Bit 10 | Bit 9 | Bit 8 |
| | HomResetPos | HomSetPos | HomStartStop |
| Stop homing | 0 | 0 | 0 |
| Start homing | 0 | 0 | 1 |
| Set home position | 0 | 1 | X |
| Delete home position | 1 | 0 | X |
| X = Status not significant | | | |

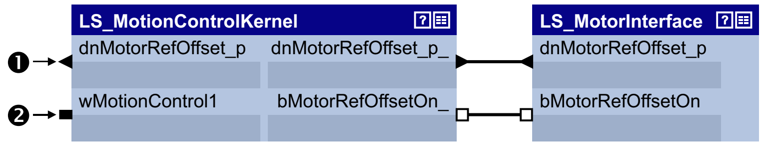
If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process inputs for controlling the operating mode:

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bHomingStartStop<br>BOOL | Start/stop homing<br>• Only possible in the "referencing" operating mode. | |
| | TRUE | If one of referencing modes "0" ... "15" in C01221 is selected:<br>**Start reference search**<br>• The current status of the reference search is indicated via the status outputs *bHomingDone* and *bHomePosAvailable*.<br>• If the 100_ Set_Ref_directly" referencing mode has been selected in C01221, the home position can be set manually via the *bHomeStartStop* input with the drive at a standstill. The current actual position is set as the home position (C01227/2) |
| | | If referencing mode "100: SetRef" is selected:<br>**Setting the home position manually**<br>• The home position is set manually with the drive at a standstill. The current actual position now corresponds to the reference position set in C01227/2 in the machine measuring system. |
| | TRUE↘FALSE | Stop homing.<br>• If the *bHomingStartStop* input is reset to FALSE during active referencing, homing is cancelled and the drive is brought to a standstill. |
| bHomingSetPos<br>BOOL | Set home position (Homing on the fly)<br>• With referencing "on the fly", the home position of a machine can be set during ongoing movement. Jerking and compensating movements do not occur. | |
| | FALSE↗TRUE | The position applied at the *dnMotorRefOffset_p* input of the SB LS_MotionControlKernel at the time of activation is set as home position if C01246/2 = "0: No TP" is set. |
| bHomingResetPos<br>BOOL | Delete home position<br>**Note:** With this function, positions are not deleted but only the status signals *bHomePosAvailable* and *bHomePosDone* are reset. Setpoints and actual positions remain untouched until a renewed reference setting or homing. | |
| | FALSE↗TRUE | The internal status "reference known" is reset.<br>• The controller is no longer referenced.<br>• The process outputs *bHomePosAvailable* and *bHomingDone* are reset to FALSE. |

### 9.6.3.1 Homing on the fly

With referencing "on the fly", the home position of a machine can be set during ongoing movement. Jerking and compensating movements do not occur.

The following illustration shows the relevant interfaces for selecting the home position at the SB LS_MotionControlKernel and SB LS_MotorInterface:



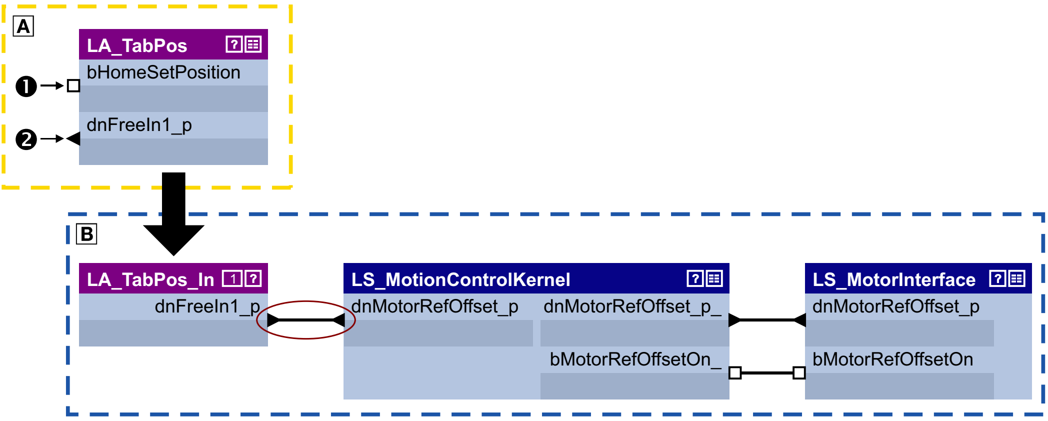

❶ Selection of the home position
❷ Setting of the home position with bit 9 ("HomeSetPos") = 1

[9-19] Interface for selecting the home position

In order that position selections can be accepted while homing on the fly, an additional connection is required in the application level.

- The following illustration shows the required modification using the example of the "table positioning" technology application.
- For transmitting the home position from the I/O level to the application level, the "free" input *dnFreeIn1_p* is used here:

A I/O level
B Application level




❶ Setting of the home position
❷ Selection of the home position

[9-20] Modified interconnection for selecting the home position

## 9.7 Manual jog

In this operating mode, the drive can be traversed manually in a clockwise or anticlockwise direction ("jogging mode").

- As an option, it is possible to change over to a second speed during traversing.
- "Retraction" of operated (travel range) limit switches is also supported. Only traversing in the corresponding retracting direction is then possible.

**Danger!**

During manual jogging, specially assigned profile parameters are active. If they have not been set correctly, the drive can engage in uncontrolled movement!

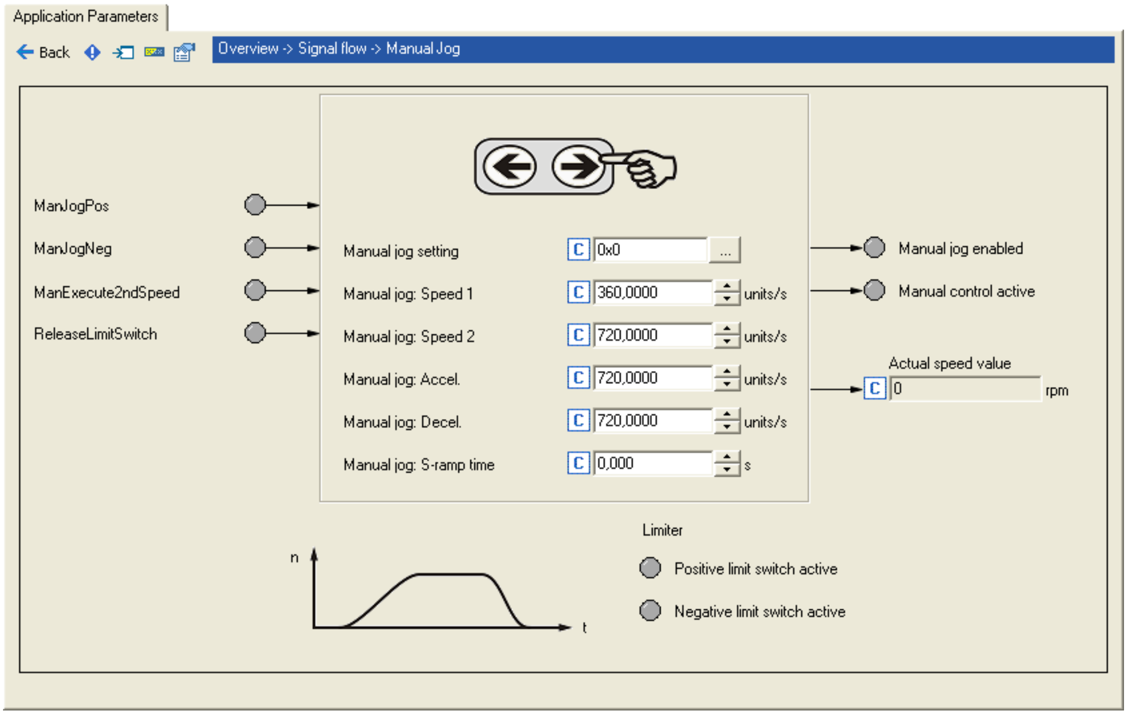
**Stop!**

In the Lenze setting, travel range monitoring is switched-off in C01230 by means of hardware limit switches and software limit positions for the "Manual jog" operating mode!

If travel range monitoring has been deactivated, the drive can travel into a mechanical limit during manual jogging and machine parts can be destroyed or damaged!

## 9.7.1 Parameter setting

### Parameterisation dialog in the »Engineer«




### Short overview of parameters for "manual jogging" mode":

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C01230 | MCK: Manual jog setting | Bit coded | |
| C01231/1 | Manual jog: speed 1 | 360.0000 | units/s |
| C01231/2 | Manual jog: Speed 2 | 720.0000 | units/s |
| C01232/1 | Manual jog: Acceleration | 720.0000 | units/s2 |
| C01232/2 | Manual jog: Deceleration | 720.0000 | units/s2 |
| C01233/1 | Manual jog: S-ramp time | 0.000 | s |
| C01235/1 | Waiting time 2nd speed | 5.000 | s |
| C01234/1 | Manual jog: Breakpoint 1 | 0.0000 | unit |
| C01234/2 | Manual jog: Breakpoint 2 | 0.0000 | unit |
| C01234/3 | Manual jog: Breakpoint 3 | 0.0000 | unit |
| C01234/4 | Manual jog: Breakpoint 4 | 0.0000 | unit |

**Note!**

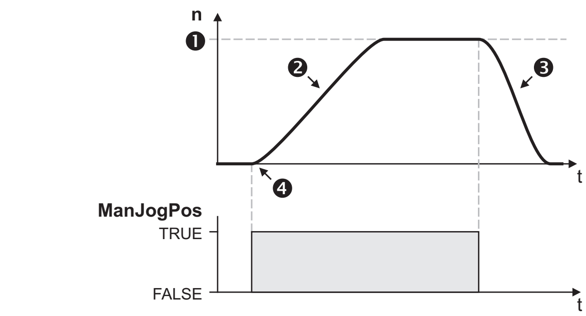
For trouble-free operation, the Machine parameters (at least gearbox factor and feed constant) must also be set correctly!

### 9.7.1.1 Functional settings

In C01230, various functional settings for manual jogging can be carried out in bit-coded form.

| Function | | Lenze setting |
|---|---|---|
| Bit 0 | **Manual jog with breakpoint**<br>During manual jog, a stop is automatically made at the breakpoint positions set in C01234/1...4.<br>• The home position must be known for this function.<br>• Travelling is continued when the pushbutton is "released" and pressed again. This button has been assigned the *ManJogPos* and/or *ManJogNeg* control function. | Off |
| Bit 1 | **Time-based start of second speed**<br>In manual jog mode, an automatic changeover to manual speed 2 takes place after a parameterisable waiting time.<br>▶ Time-based start of second speed (📖 537) | Off |
| Bit 2 | **HW limit switch on**<br>In the "manual jog" operating mode, a travel range monitoring mode via hardware limit switch is active.<br>▶ Limit position monitoring (📖 506) | Off |
| Bit 3 | **SW limit switch on**<br>In the "manual jog" operating mode, a travel range monitoring mode via parameterised software limit positions.<br>▶ Limit position monitoring (📖 506) | Off |
| Bit 4 | Reserved | Off |
| Bit 5 | Reserved | Off |
| Bit 6 | Reserved | Off |
| Bit 7 | **Position controller off**<br>In the "manual jog" operating mode, the position controller is deactivated. Thus, the compensation of the following error is switched off. | Off |

### 9.7.1.2 Smooth start and quick stop of the drive




❶ C01231/1: Manual jog: Speed 1
❷ C01232/1: Manual jog: acceleration
❸ C01232/2: Manual jog: deceleration
❹ C01233/1: Manual jog: S-ramp time

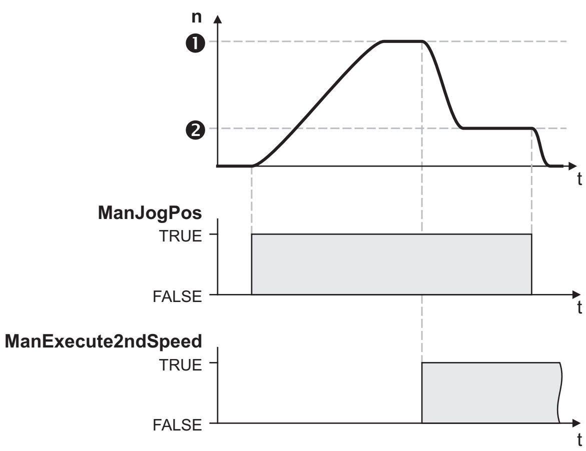
[9-21] Example: Smooth start and quick stop

- For accelerating and decelerating, different values can be set in C01232/1...2 so that smooth starting and quick stopping of the drive can be implemented.
- In order to reduce jerking, the two ramps can be set in such a way that they are s-shaped. This is done by entering a relative S-ramp time in C01233/1.

**Tip!**

Rapid deceleration (C01232/2) reduces the time from releasing of the "jog button" to actual stopping of the drive, with the result that it is easier to position the drive "by sight" and the desired stop position is not passed.

### 9.7.1.3 Second speed




❶ C01231/1: Manual jog: Speed 1
❷ C01231/2: Manual jog: Speed 2

[9-22] Example: Changing over to second speed

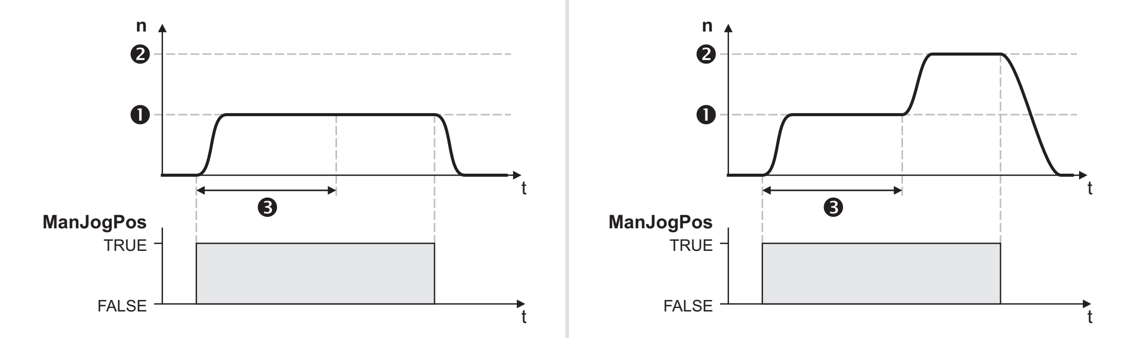
- By setting control bit 6 (*ManExecute2ndSpeed*) in theMCK control word, you can changeover to a second speed (C01231/2) during traversing.

### 9.7.1.4 Time-based start of second speed

If this function has been activated by means of bit 1 in C01230 and a waiting time > "0 s" has been set in C01235/1, an automatic change-over to manual jog speed 2 takes place when the manual jog function is activated and after the waiting time has expired.

- When the waiting time is set = 0 s, the automatic change-over is deactivated.

| **Time-based start deactivated** (C01230 - bit 1 = 0; Lenze setting) | **Time-based start activated** (C01230 - bit 1 = 1) |
| --- | --- |




❶ C01231/1: Manual jog: Speed 1
❷ C01231/2: Manual jog: Speed 2
❸ C01235/1: Waiting time 2nd speed (Lenze setting: 5 s)

[9-23] Time-based start of second speed

**Tip!**

By leaving the corresponding button for manual jog pressed down for longer than the waiting time and parameterising manual jog speed 2 so that it is higher than manual jog speed 1, you can enable longer distances to be travelled by means of manual jogging with this function.

### 9.7.2 Requesting the operating mode

Request for "manual jog" mode via the MCK control word:

| MCK control word | | | | | | |
|---|---|---|---|---|---|---|
| **Bit 31** | **...** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 0 | 1 | 0 |

X = Status not significant

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.7.3 Executing manual jogging

Control is exercised by means of bits 4 ... 7 in the MCK control word:

| Manual jog | MCK control word | | | |
|---|---|---|---|---|
| | **Bit 7** | **Bit 6** | **Bit 5** | **Bit 4** |
| | Release LimitSwitch | ManExecute 2ndSpeed | ManJogNeg | ManJogPos |
| Stop manual jogging | 0 | X | 0 | 0 |
| Manual jog, right<br>• With speed 1 (C01231/1) | | 0 | 0 | 1 |
| Manual jog, right<br>• With speed 2 (C01231/2) | | 1 | | |
| Manual jog, left<br>• With speed 1 (C01231/1) | | 0 | 1 | 0 |
| Manual jog, left<br>• With speed 2 (C01231/2) | | 1 | | |
| No change in relation to previous state | | X | 1 | 1 |
| Retract operated limit switch | 1 | 0 | 0 | 0 |

X = Status not significant

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process inputs for controlling the operating mode:

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bMotJogPos | | |
| bManJogPos<br>bManJogNeg<br>BOOL | bManJogPos = TRUE: Manual jog right<br>bManJogNeg = TRUE: Manual jog left<br>Both inputs = TRUE: No change compared to previous state<br>Both inputs = FALSE: Stop manual jog | |
| bManJogExecute2ndVel<br>BOOL | Change over to speed 2 for manual jog | |
| | FALSE | Speed 1 (C01231/1) active. |
| | TRUE | Speed 2 (C01231/2) active. |
| bReleaseLimitSwitch<br>BOOL | Retract operated limit switch | |
| | TRUE | Retract operated limit switch (in opposite direction) |

**Case 1: Reference known**

If the reference is known and the software limit positions have been set, i.e. at least one software limit position is > 0, manual jogging is carried out until a position at the corresponding software limit position is reached unless manual jogging is aborted first. Overtravelling the set software limit positions is not possible.

**Case 2: Reference not known**

If the reference is not known, the travel range limits are only monitored via the limit switches (if connected).

If you end manual jogging manually by resetting *bManJogNeg* or *bManJogPos*, the drive is brought to a standstill at the rate of deceleration set for manual jogging.

### 9.7.3.1 Manual jog to limit position

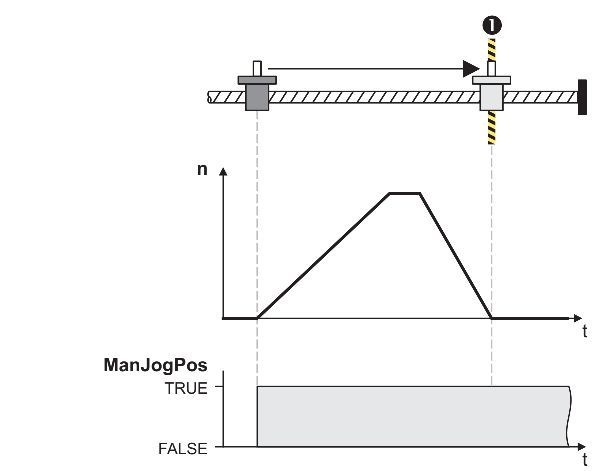
**Stop!**

In the Lenze setting, travel range monitoring is switched-off in C01230 by means of hardware limit switches and software limit positions for the "Manual jog" operating mode!

If travel range monitoring has been deactivated, the drive can travel into a mechanical limit during manual jogging and machine parts can be destroyed or damaged!

▸ Limit position monitoring (506)

**Manual jog to software limit position**

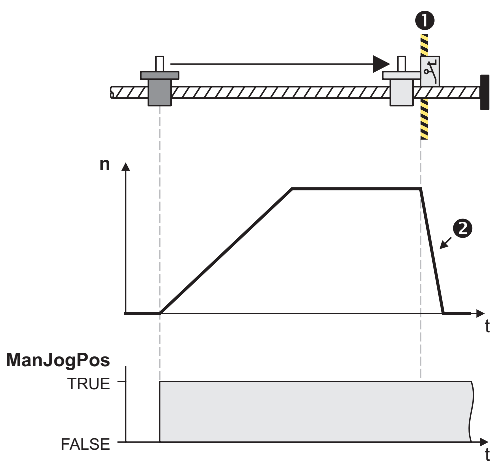


❶ Positive software limit position

[9-24] Example: Manual jogging to positive software limit position

- f the reference is known and the software limit positions have been set and are active, positioning to the corresponding software limit position is carried out unless you manually stop manual jogging first by resetting control bit 4/5 (*ManJogPos*/*ManJogNeg*) in theMCK control word.
- The drive decelerates at the set rate of deceleration (C01232/2) to the position of the corresponding software limit position.

#### Manual jogging to hardware limit position (limit switch)




❶ Positive limit switch
❷ Deceleration ramp of the quick stop function

- When monitoring of the limit switches is switched on and a limit switch is approached during manual jog, the drive is braked within the deceleration time set for the quick stop function if the "TroubleQuickStop" error response is set in C00595/1 or C00595/2.

[9-25] Example: Manual jogging to positive limit switch

### 9.7.3.2 Retracting of an operated limit switch

If control bit 7 (*ReleaseLimitSwitch*) in the MCK control word is set, retracting from an operated limit switch is possible. Traversing is carried out in the corresponding retracting direction until the limit switch is not longer operated.

- If a direction preselection is made for retracting by means of control bit 4/5 (*ManJogPos*/*ManJogNeg*) in the retracting direction, travelling is continued, even after the limit switch has been left, until the corresponding control bit is reset.
- If, instead, a direction preselection is made against the retracting direction, the drive remains at a standstill.

**Note!**

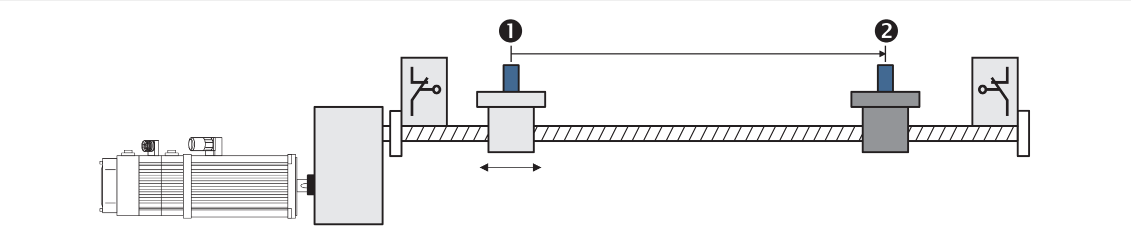
Retracting from a limit switch is only possible if this switch is still in the operated state, i.e. the corresponding limit switch input is still active. You must therefore make sure that, when travelling to a limit switch, its tripping mechanism is not "driven over" due to e.g. an excessively high mass or too much momentum so that the limit switch is no longer in the operated state as a result.

**Tip!**

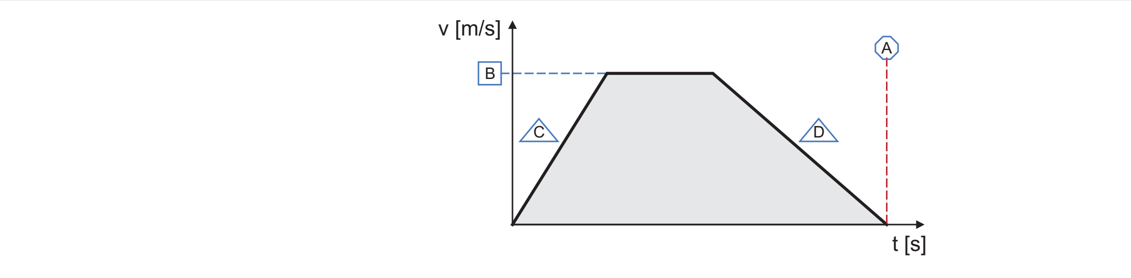
A limit switch that is in the operated state can also be moved away from again through manual jogging in the retracting direction by means of control bits 4/5 (*ManJogPos*/*ManJogNeg*).

## 9.8 Positioning

Positioning means that a workpiece/tool or material is moved from a starting position ❶ to a defined destination ❷:



To carry out positioning, a travel profile has to be stored in the drive controller for at least the following profile parameter:

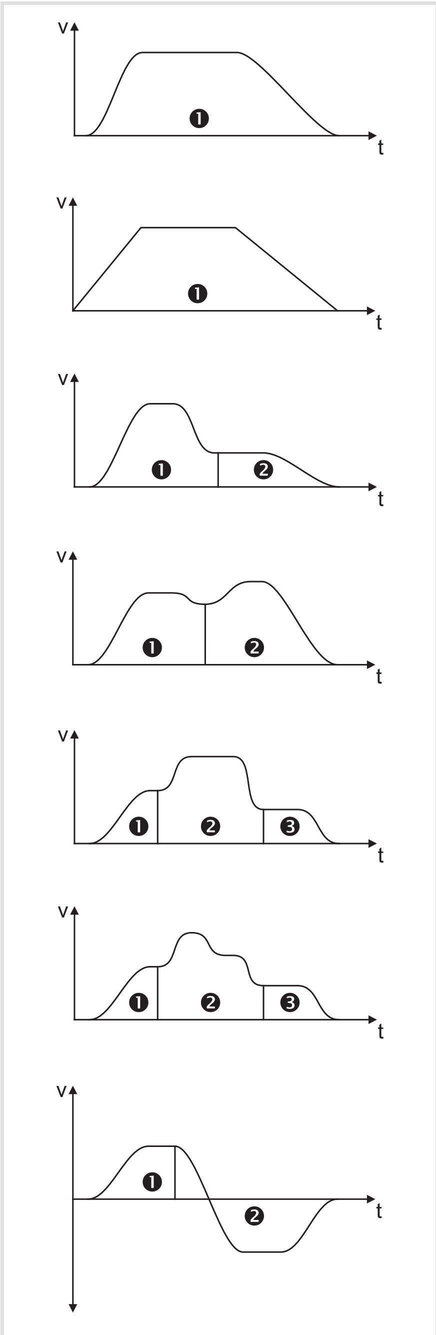



| Symbol | Profile parameter |
| --- | --- |
| A | **Position**<br>Target position or distance to be traversed. |
| B | **Speed**<br>Maximum speed at which the target is to be approached. |
| C | **Acceleration**<br>Stipulation of the change in speed at which maximum acceleration is to take place. |
| D | **Deceleration**<br>Stipulation of the change in speed at which maximum deceleration to a standstill is to take place. |

- A profile describes a motion task that can be converted into a rotary motion by the **Motion Control Kernel** in the "Positioning" operating mode.
- A positioning process can be composed of a large number of profiles that are executed in a fixed manner.
- You can find a detailed explanation of all profile parameters in the subchapter entitled "Profile entry". (📖 545)

### 9.8.1 Possible motion profiles

The following illustration shows different movement profiles that can be travelled in the "positioning" mode:



Asymmetrical trapezoidal profile
- with S-shaped ramps

Asymmetrical trapezoidal profile
- with linear ramps

Velocity changeover for profile with S-shaped ramps
- here: $v_{Profile\ 1} > v_{Profile\ 2}$ or $v_{End1} = v_{Profile\ 2}$

Velocity changeover for profile with S-shaped ramps
- here: $v_{Profile\ 1} < v_{Profile\ 2}$

Profile linkage

Profile linkage
- with speed override in profile 2

Profile linkage
- here: forwards/backwards profile

[9-26] Examples of possible motion profiles

## 9.8.2 Parameter setting

### Short overview of parameters for "positioning" mode:

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C01216 | MCK: Positioning setting | Bit coded | |
| C01300/1...15 | Profile 1 ... 15: Mode | 1: absolute (beeline) | |
| C01301/1...15 | Profiles 1 ... 15: Position | 360.0000 | unit |
| C01302/1...15 | Profile 1 ... 15: Speed | 360.0000 | unit/s |
| C01303/1...15 | Profile 1 ... 15: Accel. | 720.0000 | unit/s2 |
| C01304/1...15 | Profile 1 ... 15: Decel. | 720.0000 | unit/s2 |
| C01305/1...15 | Profile 1 ... 15: Final speed | 0.0000 | unit/s |
| C01306/1...15 | Profiles 1 ... 15: S-ramp time | 0.000 | s |
| C01307/1...15 | Profiles 1 ... 15: Sequence profile | 0 | |
| C01308/1...15 | Profile 1 ... 15: TP profile | 0 | |
| C01309/1...15 | Profile 1 ... 15: TP source | 3: TP-DigIn3 | |
| C00595/9 | MCK: Resp. to invalid pos.mode | 4: WarningLocked | |
| C00595/10 | MCK: Resp. to invalid profile data | 4: WarningLocked | |
| C00595/12 | MCK: Resp. to invalid profile no. | 4: WarningLocked | |
| C00595/14 | MCK: React target out of travel range | 4: WarningLocked | |
| C01210/1 | MCK: Curr. feed | - | units |
| C01210/2 | MCK: Akt. Set position | - | units |
| C01210/3 | MCK: Akt. Actual position | - | units |
| C01210/4 | MCK: Curr. following error | - | units |
| C01210/5 | MCK: Positioning accuracy | - | units |
| C01210/6 | MCK: Target position | - | units |
| C01211/1 | Max. traversing speed 100%_C11 | - | units/s |
| C01213/1 | MCK: Max. traversing distance | - | units |
| C01242 | MCK: Current pos profile number | - | |

Highlighted in grey = display parameter

**Note!**

For trouble-free operation, the Machine parameters (at least gearbox factor and feed constant) must also be set correctly!

### 9.8.2.1 Functional settings

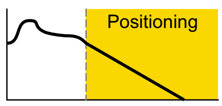
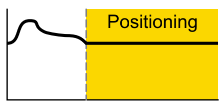
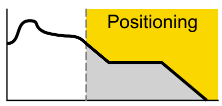
In C01216 , various functional settings regarding behaviour in the case of a changeover to the "positioning" mode can be made in bit-coded form.

- When this is being done, any mode change must be taken into account.
- Positioning with controller inhibit/enable is also possible if the corresponding setting is made.

| Function | |
|---|---|
| Bit 0 | **PosAbort at PosInit**<br>When a changeover to "positioning" mode is made, ramping down at the rate of deceleration set in C01251 for normal stopping is carried out. |
| Bit 1 | **PosExecute active at PosInit**<br>When a changeover is made to the "positioning" mode, the specified profile is immediately executed if the "MCK PosExecute" control bit has also been set. If the MCK "PosExecute" control bit has not been set, the setpoint is continued. |
| Bit 2 | Reserved |
| Bit 3 | Reserved |
| Bit 4 | **ProfilStart at PosInit**<br>When it is changed to the "Positioning" operating mode, the defined profile is executed immediately. |
| Bit 5 | Reserved |
| Bit 6 | Reserved |
| Bit 7 | Reserved |

**Note!**

In the case of multiple selection, the "PosAbort at PosInit" function, which can be activated by means of bit 0, has priority over the other functions (see the following truth table).

| **Bit 4**<br>ProfilStart | **Bit 1**<br>PosExecute active | **Bit 0**<br>PosAbort | **MCK control bit "PosExecute"** | **Behaviour when it is changed to "Positioning" operating mode** | |
|---|---|---|---|---|---|
| X | X | 1 | X |   | Ramping down the setpoint |
| **0** | **0** | **0** | X |   | Continuing the setpoint |
| 0 | 1 | 0 | 0 | | |
| 0 | 1 | 0 | 1 |   | Positioning from the setpoint |
| 1 | X | 0 | X | | |

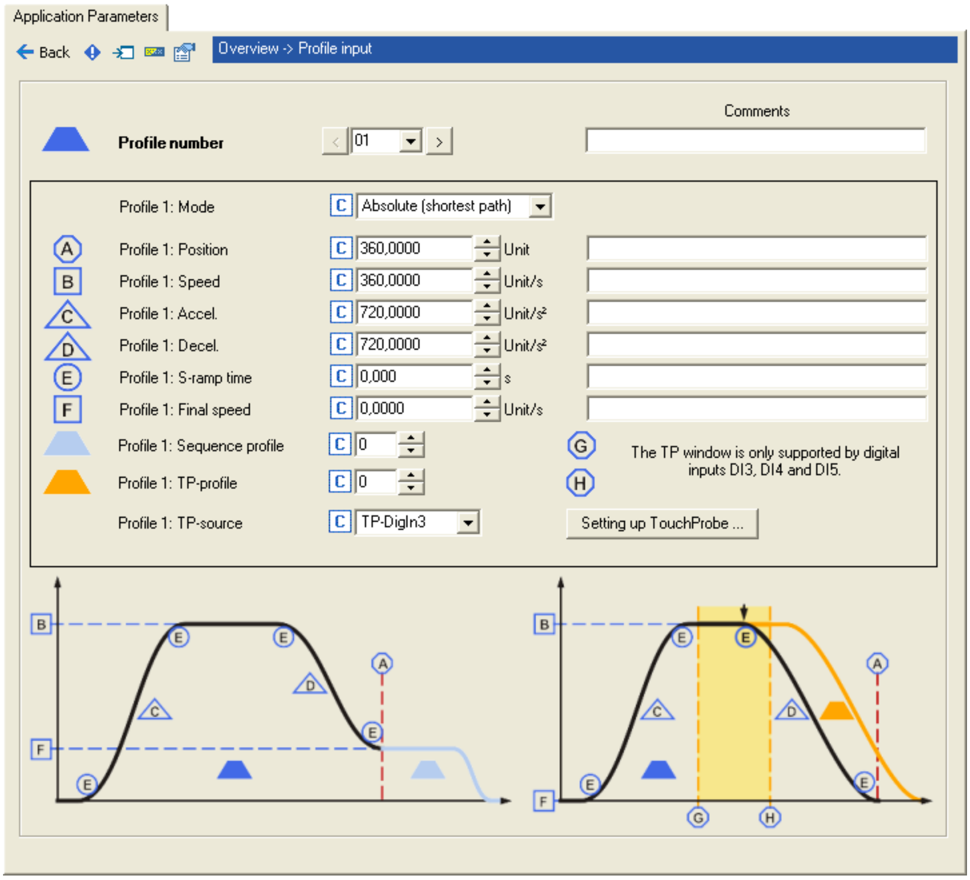
printed in bold = Lenze setting; x = any state

**Related topics:**

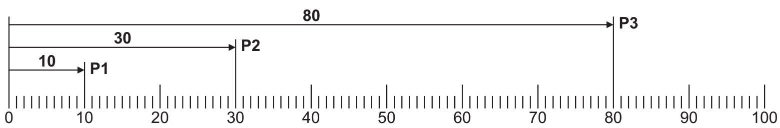
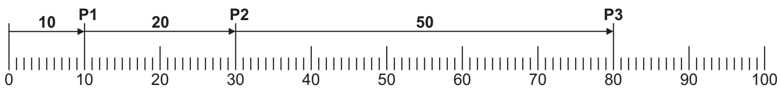

- Stipulation of the profile to be executed (552)
- Starting/cancelling a traversing task (553)

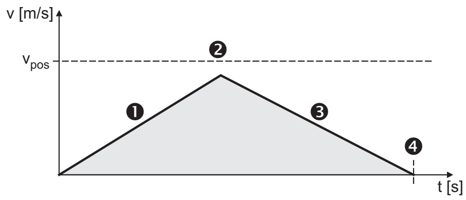
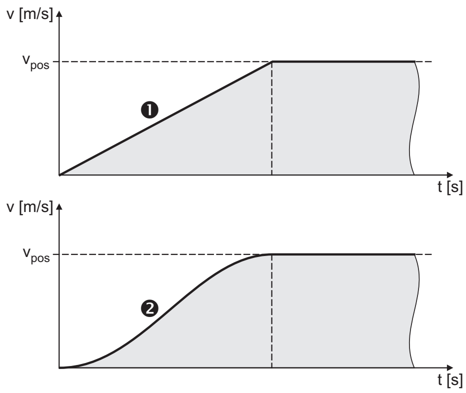
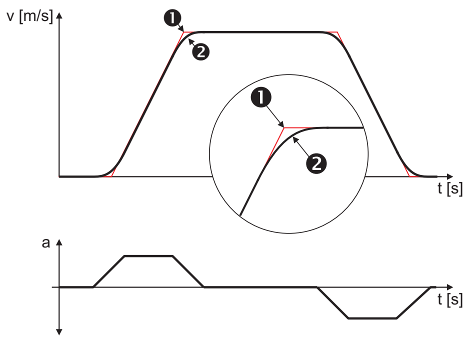
### 9.8.2.2 Profile entry

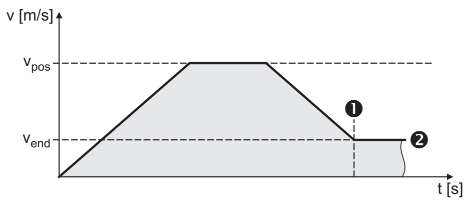
In the »Engineer«, the following parameterisation dialog is available for setting the profile parameters:




A profile is described by the following profile parameters:

| Symbol | Profile parameter |
|---|---|
|  | **(Standard) profile**<br>Profile data set (profile numbers 1 ... 15), in which the profile data are stored. |
| | **Mode** (C01300/1...15)<br>Selection of the way in which positioning is to be carried out. ▸ Positioning modes (📖 548) |
|  | **Position** (C01301/1...15)<br>Target position or distance to be traversed.<br>When the position is indicated, a distinction is made between absolute position and relative position:<br>• An absolute position always indicates the distance to the defined zero position:<br>Absolute position = Target position<br><br>• A relative position indicates the distance to the starting position (current position):<br>Relative position = Target position - Starting position<br> |

<table>
<tr><th>Symbol</th><th colspan="2">Profile parameter</th></tr>
<tr><td rowspan="2">B</td><td colspan="2">Speed (C01302/1...15)<br>Maximum speed at which the target is to be approached.<br>• Depending on the profile parameters of position, acceleration and deceleration , it is possible that the drive will not even reach the maximum speed. In this case, the graphic representation will be a trapezium instead of a triangle:</td></tr>
<tr><td>
</td><td>❶ Acceleration<br>❷ Travelling speed (is not reached in this case)<br>❸ Deceleration<br>❹ Target position (or traversing distance)</td></tr>
<tr><td rowspan="2">C</td><td colspan="2">Acceleration (C01303/1...15)<br>Stipulation of the change in speed at which maximum acceleration is to take place.<br>• Two types of acceleration are distinguished:<br>–Constant acceleration: the speed increases linearly.<br>–Linearly increasing acceleration: Speed increases in S-shape.<br>A linearly increasing acceleration (S-profile) results from the setting of an S-ramp time (see more below).</td></tr>
<tr><td>
</td><td>❶ Constant acceleration (L-profile)<br>❷ Linearly increasing acceleration (S-profile)</td></tr>
<tr><td>D</td><td colspan="2">Deceleration (C01304/1...15)<br>Stipulation of the change in speed at which maximum deceleration to a standstill is to take place.</td></tr>
<tr><td rowspan="2">E</td><td colspan="2">S-ramp time (C01305/1...15)<br>Due to stipulation of an S-ramp time for a profile, the profile is executed with S-shaped ramps, i.e. acceleration and braking processes are initiated smoothly in order to reduce jerk and thus the stress on the drive components.<br>• The acceleration/deceleration stipulated in the profile is not achieved until after the specified S-ramp time.<br>• This kind of acceleration/deceleration is needed for sensitive machine parts with a certain amount of play.<br>• The unavoidable consequence of the slower increase in acceleration in the case of the S profile is that the positioning time is longer compared to the L profile, which is more efficient in terms of time.<br>▶ S-ramp time for jerk limitation (📖 550)</td></tr>
<tr><td>
</td><td>❶ Without jerk limitation (L profile)<br>❷ With jerk limitation (S profile)</td></tr>
</table>

| Symbol | Profile parameter |
|---|---|
| F | **Final speed** (C01305/1...15)<br>This specifies the speed at which the drive is to start the next profile after reaching the target position. With a final speed not equal to "0", "velocity changeover" or "overchange" is possible, i.e. when the target position is reached, a second positioning process is started immediately without the drive coming to a standstill at the first target position. |
| |  <br>❶ Target position<br>❷ Final speed (in this case, not equal to "0") |
| | **Sequence profile** (C01307/1...15)<br>**for profile linkage/following block control**<br>A special feature is automatic advancing to subsequent profiles with and without velocity changeover. For this purpose, the profile number of the desired subsequent profile (1 to 15) is simply set in the parameter "Sequence profile" (C01307/1...15) of a profile.<br>After execution of the profile (target position reached), the set following (subsequent) profile is started automatically. In this way, profile chains can be stipulated without additional control processes.<br>• If the profile parameter "Final speed" (C01305/1...15) is set to <> "0", there is a velocity changeover leading into the following (subsequent) profile at the set final speed.<br>• If "0" is set for the following (i.e. subsequent) profile, profile linkage does not take place.<br>• This function can be performed in all positioning modes. |
| | **TP profile** (C01308/1...15)<br>Profile number of the profile (1 ... 15) that is to be executed after a touch probe has been detected.<br>• If "0" is set, there will be no profile stepping through touch probe.<br>• Only relevant for positioning modes with touch-probe.<br>▶ Touch probe positioning (📖 549) |
| | **TP source** (C01309/1...15)<br>Selection of the signal source for touch probe detection.<br>• Only relevant for positioning modes with touch-probe.<br>▶ Touch probe positioning (📖 549) |

### 9.8.2.3 Positioning modes

For positioning, it is possible to select from different positioning modes in relation to the type of axis/application. These modes are described in the following table.

**Stop!**

In the positioning modes "continuous" and "relative", the "TroubleQuickStop" error response occurs in the Lenze setting when the maximum travel distance is exceeded (C01213/1). ▶ Monitoring of the maximum travel distance (📖 512)

**Note!**

For absolute positioning, the home (reference) position must be known!

- If absolute positioning (positioning modes 1, 4, 5, 8, 11, 12) is started although the home position is not known:
  - The error message set in C00595/8 is displayed (Lenze setting: "WarningLocked").
  - The error message "Ck08: Home position unknown" is entered into the logbook.

| Positioning mode | | Info |
|---|---|---|
| 1 | Absolute (shortest path) | Travelling along the axis takes place until an absolute position is reached via the shortest path.<br>• Reference for the absolute position is the zero position. |
| 8 | Absolute (shortest path) to TP | |
| 2 | Continuous | In these two modes, no particular position is approached but travelling takes place at the traversing speed stipulated by means of the profile.<br>• The values of the profile are used for accelerating and decelerating.<br>• The direction of travel is determined by the sign of the traversing speed. |
| 9 | Continuous to TP | |
| 3 | Relative | The axis is traversed by a distance.<br>• Reference for the distance is the target position of the previously executed profile. |
| 10 | Relative to TP | |
| **Positioning modes for Modulo measuring system (rotary table application)**<br>▶ Activation of the modulo measuring system (📖 502) | | |
| 4 | absolute (Cw) | Clockwise travelling along the axis takes place until an absolute position is reached.<br>• Reference for the absolute position is the zero position.<br>• The zero position of the axis can be exceeded in this direction. |
| 11 | Absolute (Cw) on TP | |
| 5 | absolute (Ccw) | Counter-clockwise travelling along the axis takes place until an absolute position is reached.<br>• Reference for the absolute position is the zero position.<br>• The zero position of the axis can be exceeded in this direction. |
| 12 | Absolute (Ccw) on TP | |

### 9.8.2.4 Touch probe positioning

In case of a touch probe positioning, first the profile is executed according to the set profile parameters. If a touch probe is detected during the process, it is automatically changed to the profile defined in the "TP profile" profile parameter. This profile mode is also defined via the mode selection like in the following profile start processes.

If no valid TP profile is set, the travel request is aborted ("PosStop").

**Preconditions for touch probe positioning**

- The mode of the current profile data set contains the setting "to TP".
- For absolute positioning, the home (reference) position must be known.
- In the corresponding profile data set, the following profile parameters must be set in addition:
  - TP profile (C1308/x)
  - TP source (C1309/x)
- The touch probe interface is configured such that the selected TP source is enabled.
  - See chapter entitled "Touch probe detection". (📖 362)

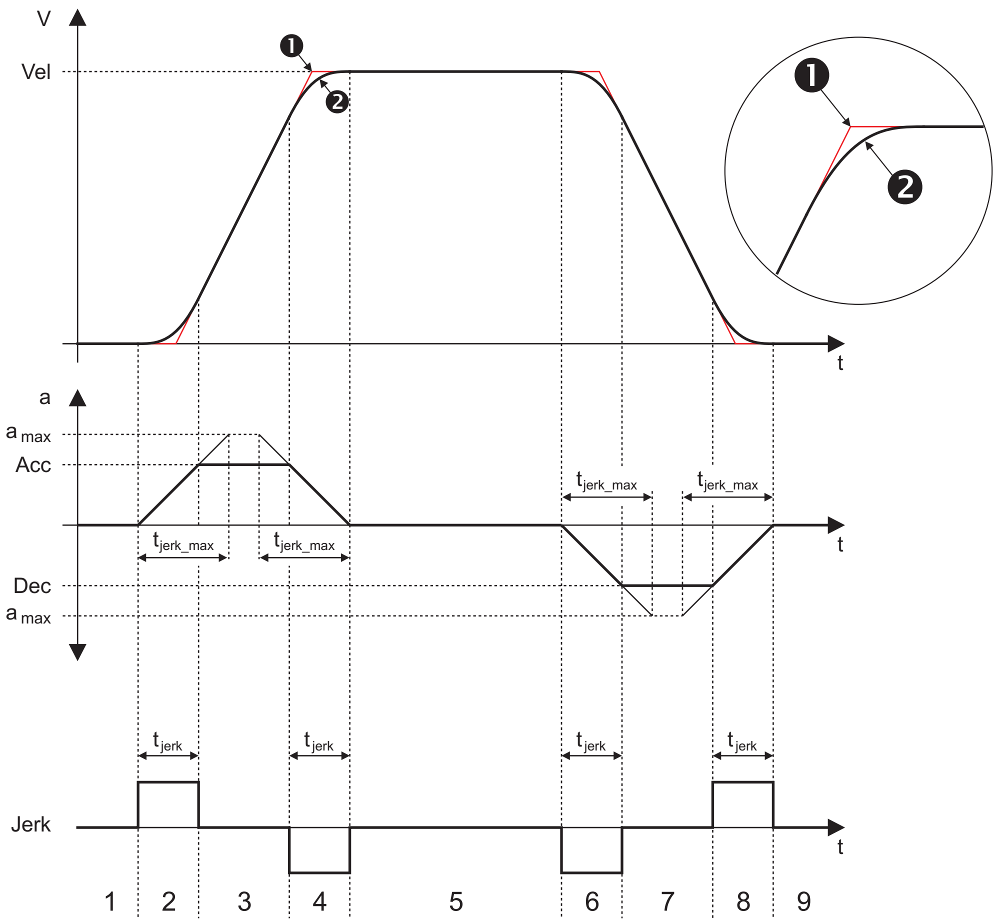
**Note!**

Go to C02810/x and select the edge, the digital input used for the connection of the touch-probe sensor is to respond. In the Lenze setting of C02810/x, no touch probe is detected!

### 9.8.2.5 S-ramp time for jerk limitation

The maximum jerk is defined by the selection of an S-ramp time $t_{jerk\_max}$, after which the max. acceleration ($a_{max}$) only is reached.

- The actual jerk time $t_{jerk}$ is reduced according to the actual acceleration Acc:




❶ Without jerk limitation
❷ With jerk limitation

1. Standstill
2. Acceleration with set jerk limitation
3. Acceleration according to acceleration profile (Acc)
4. Reduce acceleration (jerk limitation)
5. Traversing with Vel according to speed profile
6. Deceleration with set jerk limitation
7. Deceleration according to deceleration profile (Dec)
8. Reduce deceleration (jerk limitation)
9. Standstill (position target reached)

**Note!**

If disproportionately high S-ramp times are set for low acceleration times, this may result in an incorrect profile generation.

Example: v = 100 mm/s, a = 1000 mm/$s^2$

→ $t_{acc}$ = 0.1 s
→ $t_{jerk\_max}$ = 1 s

For this reason, you must only set plausible S-ramp times (this should not exceed half the value of $t_{acc}$).

### 9.8.3 Requesting the operating mode

Request for "positioning" mode by means of the MCK control word:

| MCK control word | | | | | | | |
|---|---|---|---|---|---|---|---|
| Bit 31 | ... | | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | X | 0 | 0 | 1 | 1 |
| X = Status not significant | | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.8.4 Carrying out positioning

#### 9.8.4.1 Stipulation of the profile to be executed

The profile to be executed is stipulated by means of the MCK control word in bit-coded form:

| Bit | Designation | Description | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 24<br>...<br>31 | ProfileNo_Bit0<br>...<br>ProfileNo_Bit7 | **Profile** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | Profile 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | Profile 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | Profile 2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | | ... | ... | | | | | | | |
| | | Profile 15 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| | | All other possible settings are reserved for future extensions! | | | | | | | | |

If the **Motion Control Kernel** is downstream from the **MCKInterface**, the L_MckCtrlInterface FB provides the following processing inputs for stipulating the profile to be executed:

| Identifier<br>Data type | Information/possible settings |
|---|---|
| wProfileNo<br>WORD | Selection of the profile number<br>• Optionally as a data word or binary coded.<br>• In the Lenze setting, an operating mode change is carried out by the L_MckCtrlInterface FB at the same time as the profile is stipulated:<br>–If profile 0 is stipulated: Activation of "Speed follower" operating mode<br>–If profile 1 is stipulated: Activation of "Homing" operating mode<br>–If profile 2 is stipulated: Activation of "Manual jog" operating mode<br>–If profile 3 ... 15 is stipulated: Activation of "Positioning" operating mode<br>• The operating mode change with profile number can be set in C01298/1...4. |
| bProfileNo_1<br>...<br>bProfileNo_8<br>BOOL | |

**Note!**

The profile 0 is no valid profile for the "positioning" operating mode.

If a driving request with an invalid profile number is started, the response set in C00595/12 occurs (Lenze setting: "WarningLocked").

### 9.8.4.2 Starting/cancelling a traversing task

Control is exercised by means of bits 16 ... 19 in the MCK control word:

| **Homing** | **MCK control word** | | | |
|---|---|---|---|---|
| | **Bit 19** | **Bit 18** | **Bit 17** | **Bit 16** |
| | PosStop | Pos DisableFollowProfile | PosFinishTarget | PosExecute |
| Start travelling | 0 | 0 | 0 | 0↗1 |
| Complete interrupted profile | 0 | 0 | 0↗1 | 0 |
| Do not travel sequence profile | 0 | 1 | X | X |
| Cancel travelling* | 1 | X | X | X |

*From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked).

X = Status not significant

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process inputs for controlling the operating mode:

| **Identifier** / Data type | **Information/possible settings** | |
|---|---|---|
| bPosExecute<br>BOOL | Start travelling | |
| | FALSE↗TRUE | Execute selected profile |
| bPosExecuteFinish<br>BOOL | Complete interrupted profile | |
| | FALSE↗TRUE | A positioning process previously cancelled, e.g. by *bPosStop* or due to a device error, is resumed by travelling to the original target. |
| bPosDisableFollowProfile<br>BOOL | Do not execute following profile (switch off profile linkage) | |
| | TRUE | Evaluation of the sequence profile number parameterised in C01307/1...15 for the selected profile is suppressed. |
| bPosStop<br>BOOL | Cancel travelling | |
| | TRUE | Stop positioning<br>From version 02.00.00, more travel requests will be inhibited ("PosExecute" will be blocked). |

**Tip!**

Travel requests/profiles can also be started while the drive is running. The drive does not need to be at standstill.

**Related topics:**


- Target position monitoring (status "drive in target") (📖 510)
- Monitoring of the maximum travel distance (📖 512)
- Following error monitoring system (📖 513)

### 9.8.4.3 Override of the parameterised positioning mode

The setting of the positioning mode in MCK control word superimposes the mode selection via the "Mode" profile parameter (C1300/1...15). This means that for selected profiles the mode selection and the request of a touch probe positioning via process data are possible. The value set in C01300/1...15 is not overwritten. The following applies to the override:

A. Valid positioning mode in the MCK control word:
The positioning mode set in the MCK control word is used.

B. Positioning mode in the MCK control word = 0:
The positioning mode set in "mode" profile parameter (C1300/1...15) is used.

C. Invalid positioning mode in the MCK control word:
Ck09Error message "Ck09: Invalid positioning mode". The response set in C00595/9 occurs (Lenze setting: "WarningLocked").

| Bit | Designation | Description | | | | |
|---|---|---|---|---|---|---|
| 20<br>...<br>23 | PosModeBit0<br>...<br>PosModeBit3 | **Positioning mode** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | Positioning mode = setting in C01300/1...15 | 0 | 0 | 0 | 0 |
| | | Absolute (shortest path) | 0 | 0 | 0 | 1 |
| | | Continuous | 0 | 0 | 1 | 0 |
| | | Relative | 0 | 0 | 1 | 1 |
| | | absolute (Cw) | 0 | 1 | 0 | 0 |
| | | absolute (Ccw) | 0 | 1 | 0 | 1 |
| | | Absolute (shortest path) to TP | 1 | 0 | 0 | 0 |
| | | Continuous to TP | 1 | 0 | 0 | 1 |
| | | Relative to TP | 1 | 0 | 1 | 0 |
| | | Absolute (Cw) on TP | 1 | 0 | 1 | 1 |
| | | Absolute (Ccw) on TP | 1 | 1 | 0 | 0 |
| | | All other possible settings are reserved for future extensions! | | | | |

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process input for stipulating an override of the positioning mode:

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| wPosMode<br>WORD | Override of the positioning mode set in the profile data<br>• Only bit 0 ... bit 3 of *wPosMode* are evaluated. | |
| | 0 | Positioning mode = setting in C01300/1...15 |
| | 1 | Absolute (shortest path) |
| | 2 | Continuous |
| | 3 | Relative |
| | 4 | absolute (Cw) |
| | 5 | absolute (Ccw) |
| | 8 | Absolute (shortest path) to TP |
| | 9 | Continuous to TP |
| | 10 | Relative to TP |
| | 11 | Absolute (Cw) on TP |
| | 12 | Absolute (Ccw) on TP |
| | All other possible settings are reserved for future extensions! | |

### 9.8.4.4 Position teaching

The MCK control word can be used to initiate "teaching" or "latching" of the MCK setpoint position or the current position into the currently selected profile.

| Bit | Designation | Description |
|---|---|---|
| 14 | PosTeachSetPos | "1" ≡ Teach MCK set position into the selected profile |
| 15 | PosTeachActPos | "1" ≡ Teach current position into the selected profile |

If the **Motion Control Kernel** is downstream from the **MCKInterface**, the L_MckCtrlInterface FB provides the following process inputs for teaching:

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bPosTeachSetPos<br>BOOL | Teach MCK setpoint position | |
| | TRUE | Teach MCK setpoint position into the selected profile. |
| bPosTeachActPos<br>BOOL | Teach current position | |
| | TRUE | Teach current position into the selected profile. |

**Note!**

If request is made at the same time, the current position is taught.

## 9.9 Stop

If the "normal stop" operating mode is active, the drive is brought to a standstill with a parameterisable deceleration ramp.

### 9.9.1 Parameter setting

**Short overview of parameters for "Normal stop" operating mode:**

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01251/1 | Normal stop: Deceleration | 720.0000 | units/s2 |
| C01252/1 | Stop: S-ramp time | 0.000 | s |

### 9.9.2 Requesting the operating mode

Request for "Normal stop" operating mode by means of the MCK control word:

| MCK control word | | | | | | |
|---|---|---|---|---|---|---|
| Bit 31 | ... | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 1 | 0 | 0 |
| X = Status not significant | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

## 9.10 Position follower

In the "position follower" operating mode, the drive follows a position setpoint.

### 9.10.1 Parameter setting

**Short overview of parameters for "position follower" operating mode:**

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C01218 | MCK: Pos follower setting | Bit coded | |
| C01236/1 | PosFollower: Sync. speed | 360.0000 | units/s |
| C01237/1 | Pos follower: Sync. accel. | 720.0000 | units/s2 |
| C01237/2 | Pos follower: Sync. decel. | 720.0000 | units/s2 |
| C01238/1 | Pos follower: Sync. S-ramp time | 0.000 | s |
| C01239 | MCK: Interpolator cycle pos. follower | 0.000 | ms |

#### 9.10.1.1 Functional settings

In C01218, various functional settings for the position follower can be made in bit-coded form.

| Function | | Lenze setting |
|---|---|---|
| Bit 0 | Speed FF control.: nSpeedSetValue_a | Off |
| Bit 1 | Speed FF control: nSpeedAddValue_v | Off |
| Bit 2 | HW limit switch on | On |
| Bit 3 | SW limit switch on | On |
| Bit 4 | Reserved | Off |
| Bit 5 | Reserved | Off |
| Bit 6 | Reserved | Off |
| Bit 7 | Position controller off | Off |

**Speed feedforward control by means of nSpeedSetValue_a**

If this function has been activated by means of bit 0 in C01218, the main setpoint *nSpeedSetValue_a* is used as the speed feedforward control value.

**Speed feedforward control by means of nSpeedAddValue_v**

If this function has been activated by means of bit 1 in C01218, the additive speed value *nSpeedAddValue_v* is used as the speed feedforward control value.

**HW limit switch on**

If this function has been activated by means of bit 2 in C01218, travel range monitoring by means of hardware limit switches is active in this mode.

▶ Limit position monitoring (📖 506)

#### SW limit switch on

If this function has been activated by means of bit 3 in C01218, travel range monitoring by means of parameterised software limit positions is active in this mode.

▶ Limit position monitoring (📖 506)

#### Position controller off

If this function is activated via bit 7 in C01218, the position controller is deactivated in this operating mode. Hence, following error control is switched-off.

### 9.10.2 Requesting the operating mode

Requesting the "position follower" operating mode by means of the MCK control word:

| MCK control word | | | | | | |
|---|---|---|---|---|---|---|
| **Bit 31** | **...** | **Bit 4** | **Bit 3** | **Bit 2** | **Bit 1** | **Bit 0** |
| | | | OpMode_Bit3 | OpMode_Bit2 | OpMode_Bit1 | OpMode_Bit0 |
| X | ... | X | 0 | 1 | 0 | 1 |
| X = Status not significant | | | | | | |

If the **MCKInterface** is connected upstream to the **Motion Control Kernel** and if the operating mode is requested at the L_MckCtrlInterface FB, the *wOperationMode* and *bOperationMode_1...8* process inputs are available.

### 9.10.3 Setpoint selection

The absolute position setpoint is stipulated via the *dnPosSetValue_p* process input.

**Note!**

The *bPosCtrlOn* process input must be set to TRUE so that position/angle control is active within motor control.

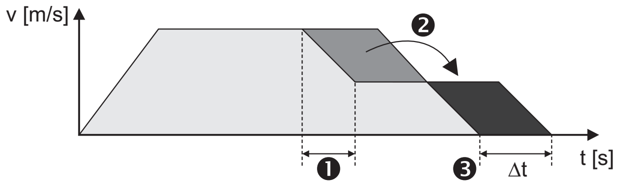
#### Compensating following errors

If the *bDeltaPosOn* process input is set to TRUE, a position difference (following error) is compensated for which is not internally generated from setpoint/actual positions but must be defined via the *dnDeltaPos_p* process input instead.

## 9.11 Override

An "Override" is the change of profile parameters and their acceptance during the positioning process.

- In this case the traversing profile has to be adapted accordingly, so that positioning is carried out exactly to the target position defined, even if for example a change in speed occurs during the positioning process ("Speed override"):




❶ Speed is reduced during the positioning process.
❷ In order to reach the position defined, the missing area has to be "attached" to the profile.
❸ Due to the speed reduction the positioning process takes longer (Δt).

[9-27] Override (here: speed override)

- The override for speed, acceleration, and S-ramp time affects all motion profiles that are controlled by the internal profile generator:
  - Manual jog
  - Retracting from limit switches
  - Homing
  - Point-to-point positioning

**Note!**

The online change of speed and acceleration is in effect from the start of the profile until the deceleration phase begins. Changing the deceleration phase by means of an override is therefore not possible!

- In the case of an override value of 0 % for the speed, the drive is brought to a standstill.
- In the case of an override value of 0 % for the acceleration, acceleration does not take place any longer.

The acceleration override also has the same effect on the deceleration ramp to the same extent specified, but only until the deceleration phase is initiated.

The override has no impact on:

- The "speed follower" operating mode
- Synchronising processes
- Setpoint controls via external setpoints
- Abort via input *bPosStop*
- Speed setpoint selection in case of error (e. g. "Fail-QSP")

### 9.11.1 Speed override

#### Activating override

If control bit 11 (*EnableSpeedOverride*) has been set to "1" in the MCK control word, a speed override is carried out in accordance with the stipulated override value.

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface provides the following process input for activating the speed override:

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bEnableVelOverride<br>BOOL | Speed override | |
| | TRUE | Activate speed override |

#### Stipulate override value

The override value is selected via the *nSpeedOverride_a* input at the SB LS_MotionControlKernel. The override value is a percentage of the speed of the current profile.

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nSpeedOverride_a<br>INT | Value for speed override<br>• Percentage multiplier (0 ... 199.99 %) for the currently active speed.<br>• 16384 ≡ 100 % of the maximum traversing speed (display in C01211/1).<br>• If the override value is 0 %, the drive is brought to a standstill. |

#### Deactivate override

If control bit 11 (*EnableSpeedOverride*) is reset to "0", travelling takes place at the speeds that have been specified by means of the profile parameters. Acceleration/deceleration from the speed defined with override to the speed set in the profile takes places immediately in this case.

Deactivating the override within the braking/deceleration ramp does not have any effect.

## 9.11.2 Acceleration override

### Activating override

If control bit 12 (*EnableAccOverride*) in the MCK control word has been set to "1", an acceleration override takes place in accordance with the stipulated override value.

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process input for activating the acceleration override:

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bEnableAccOverride<br>BOOL | Acceleration override | |
| | TRUE | Activate acceleration override |

### Stipulate override value

The override value is stipulated via the *nAccOverride_a* input at the LS_MotionControlKernel SB. The override value is a percentage of the maximum acceleration that has been set for the respective operating mode (referencing, manual jog, positioning etc.) in the corresponding profile parameter.

The acceleration override results in a deceleration override. Both ramps are evaluated by multiplication by the acceleration override until the deceleration process sets in.

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nAccOverride_a<br>INT | Value for acceleration override<br>• Percentage multiplier (0 ... 199.99 %) for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised acceleration of the corresponding operating mode.<br>• If the override value is 0 %, acceleration ceases. |

### Deactivate override

If control bit 12 (*EnableAccOverride*) is reset to "0", travelling takes place at the acceleration rates specified by means of the profile parameters. "Ramping up" from the acceleration defined with override to the acceleration set in the profile takes place immediately in this case.

### 9.11.3 S-ramp smoothing override

#### Activating override

If control bit 13 (*EnableSRampOverride*) in the MCK control word is set to "1", S-ramp smoothing override is executed according to the selected override value.

If the **Motion Control Kernel** is downstream from the **MCKInterface** in the circuit, the L_MckCtrlInterface FB provides the following process input for activating the S-ramp smoothing override:

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bEnableSRampOverride<br>BOOL | S-ramp smoothing override | |
| | TRUE | Activate S-ramp smoothing override |

#### Stipulate override value

Optional stipulation of the override value is carried out via the *nSRampOverride_a* input at the LS_MotionControlKernel SB. The override value is a percentage of the S-ramp time set in the profile data.

**Note!**

If the *nSRampOverride_a* input remains unconnected or if an override value of "0 %" is selected, activation of the S-ramp override causes deactivation of the S-ramp time.

- Deactivation of the S-ramp time before the start of a profile with S-ramp time causes linear ramp generation.
- Deactivation of the S-ramp time during a traversing process, however, is not accepted immediately in the profile generator, but the profile generator checks automatically when an online change of the ramp form can be carried out and then initiates it automatically.

**Tip!**

Thus it is possible to start a travel profile with S-ramp time and then deactivate the S-ramp time, e.g. to traverse with a linear characteristic after reaching the profile speed.

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nSRampOverride_a<br>INT | Value for S-ramp smoothing override<br>• Percentage multiplier (0 ... 100 %) for the currently active acceleration.<br>• 16384 ≡ 100 % of the parameterised S-ramp time (C01306/1...15).<br>• Values > 16384 are ignored. |

#### Deactivate override

If control bit 13 (*EnableSRampOverride*) is reset to "0", the S-ramp time specified by means of the profile parameters is used for profile generation.

## 9.12 Holding brake control

This basic function is used for low-wear control of a holding brake.

**Danger!**

Please note that the holding brake is an important element of the safety concept of the entire machine.

Thus, proceed very carefully when commissioning this system part!

**Stop!**

Holding brakes on Lenze motors are not intended for braking during operation. The increased wear caused by braking during operation can destroy the motor holding brake!

**Note!**

- **Deactivate automatic DC-injection braking when a holding brake is used!**
  - For this purpose, go to C00019 and set the Auto DCB threshold to "0".
  - Background: Controller inhibit is already activated by the holding brake control.
- If an electrically holding (self-releasing) brake is to be controlled instead of an electrically released (self-holding) brake, the trigger signal must be inverted!
  ▸ Functional settings (📖 569)
- Detailed information on mounting and electrical installation of the motor holding brake can be found in the documentation on the motor holding brake.

### Intended use

Motor holding brakes are used to lock axes if the controller is inhibited or in case of "mains off" system status. This is not only important for vertical axes but also for e.g. horizontal axes which may cause various problems if the motion is not controlled.

Examples:

- Loss of the reference information after mains OFF and further spinning of the drive.
- Collision with other moving machine parts.

### 9.12.1 Internal interfaces

In the function block editor, the LS_MotionControlKernel system block provides the following internal interfaces for the basic function "holding brake control":

#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bBrkRelease<br>BOOL | Releasing/applying the brake in connection with the selected operating mode | |
| | FALSE | Apply brake.<br>• During automatic operation, the internal brake logic controls of the brake. |
| | TRUE | Release brake manually (forced release).<br>• **Note!**<br>The brake can also be released when the controller is inhibited!<br>• During automatic operation, the internal brake logic is deactivated and the brake is released (supervisor operation). If the brake control has inhibited the controller, this inhibit is deactivated again.<br>• In semi-automatic operation, the brake is released including feedforward control. |
| bBrkStartValue2<br>BOOL | Selection of the torque feedforward control value for manual specification of the feedforward control value<br>• Only effective if bit 4 in C02582 is set to "1".<br>▶ Feedforward control of the motor before release | |
| | FALSE | Starting value 1 (C02581/4) active. |
| | TRUE | Starting value 2 (C02581/5) active. |
| nBrkAddValue_a<br>INT | Additive feedforward control value (speed or torque) in [%] for torque feedforward control when the respective control mode is started<br>• For speed control: 100 % ≡ reference speed (C00011)<br>• For torque control: 100 % ≡ maximum torque (C00057)<br>▶ Feedforward control of the motor before release | |
| bBrkApplied<br>BOOL | Input for status detection via switching contacts at the brake<br>• Only effective if bit 5 in C02582 is set to "1". | |
| | FALSE | Brake is released. |
| | TRUE | Brake is applied. |

#### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bBrkReleaseOut<br>BOOL | Trigger signal for switching element holding brake control via a digital output<br>• Use bit 0 under C02582 to activate inverted switching element triggering.<br>▶ Functional settings | |
| | FALSE | Apply brake. |
| | TRUE | Release brake. |
| bBrkReleased<br>BOOL | "Brake released" status signal considering the brake release time<br>• If the holding brake is triggered to be applied, *bBrkReleased* is immediately reset to FALSE even if the brake application time has not elapsed yet! | |
| | TRUE | Brake released (when the brake release time has elapsed). |

**Stop!**

The digital outputs are not suitable for the "direct" control of a holding brake!

- Connect the digital output connected to the trigger signal *bBrkReleaseOut* with a relay or power contactor which switches the brake supply.
- When a power contactor is used, the response and release time of the earth contact is added to the response and release time of the brake. Both times must also be considered for parameterising the closing and opening time of the holding brake

**Tip!**

For direct triggering of a holding brake, the high current output of the 8400 TopLine controller can be used that can switch max. 2.5 A. For this purpose, connect the *bBrkReleaseOut* trigger signal to the *bOutHC* input of the SB LS_DigitalOutput.

### 9.12.2 Parameter setting

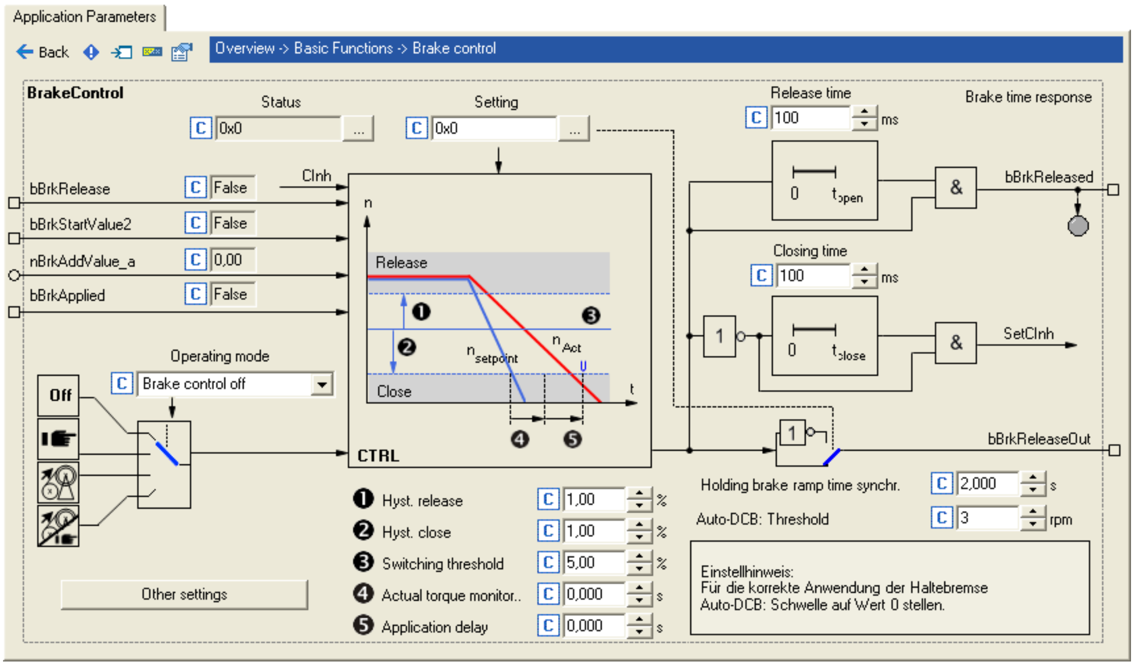
**Danger!**

A faultless holding brake control function requires a correct setting of the different deceleration times in the following parameters!

A wrong setting of the delay times can cause a faulty control of the brake!

**How to go to the parameterisation dialog of the holding brake control:**

1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Go to the *Overview* dialog level and click the "**Basic functions**" button.
4. Go to the *Overview* → *Basic functions* dialog box and click the **Holding brake control** button.




## Short overview of parameters for the holding brake control:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C02580 | Holding brake: Operating mode | 0: Brake control off | |
| C02581/1 | Holding brake: Switching threshold | 5.00 | % |
| C02581/2 | Holding brake: Hysteresis for releasing | 1.00 | % |
| C02581/3 | Holding brake: Hysteresis for applying | 1.00 | % |
| C02581/4 | Holding brake: FF control starting value 1 | 0 | % |
| C02581/5 | Holding brake: FF control starting value 2 | 0 | % |
| C02582 | Holding brake: Setting | 0 | |
| C02589/1 | Holding brake: Application time | 100 | ms |
| C02589/2 | Holding brake: Release time | 100 | ms |
| C02589/3 | Holding brake: Waiting time status | 100 | ms |
| C02589/4 | Holding brake: Ramp FF control | 0 | ms |
| C02593/1 | Holding brake: Actual value monitoring | 0.000 | ms |
| C02593/2 | Holding brake: Application delay | 0.000 | ms |
| C02610/1 | MCK: Holding brake ramp time synchr. | 2.000 | s |
| C02607 | Holding brake: Status | - | |
| C00830/68 | MCK: nBrkTorqueAdd_a | - | % |
| C00833/80 | MCK: bBrkRelease | - | |
| C00833/81 | MCK: bBrkStartTorqueDir | - | |
| C00833/82 | MCK: bBrkApplied | - | |
| Highlighted in grey = display parameter | | | |

### 9.12.2.1 Operating mode

For different applications and tasks, different operating modes are available in C02580. The selected operating mode determines whether the holding brake control is used and how the holding brake will be switched.

**Mode 0: Brake control off**

In this mode, brake control is switched off (not active).

- The trigger signal *bBrkReleaseOut* for the holding brake control switching element is set to FALSE.
- The status signal *bBrkReleased* is set to FALSE.

**Note!**

In the Lenze setting, the mode 0 is preset to get into a safe state after the mains is switched on.

**Mode 11: Manual control**

In this mode, brake release and brake application can be directly controlled via the input *bBrkReleaseBrake* without special logic or automatic.

- Setting pulse inhibit or controller inhibit has no influence on the trigger signal *bBrkReleaseOut* for the holding brake control switching element.
- After the brake has been activated and the brake application time has expired, the controller is inhibited automatically by the basic "Holding brake control" function.

**Tip!**

You can use mode 11 to easily check if the brake switches correctly.

**Mode 12: Automatic control**

In this mode, the brake is controlled automatically.

**Danger!**

In this mode, the input *bBrkReleaseBrake* should be permanently set to FALSE unless manual release (supervisor operation) is required.

If the *bBrkReleaseBrake* input is set to TRUE, the brake is released immediately, even if the controller is inhibited!

- If the requested speed setpoint reaches a parameterisable upper speed threshold that allows traversing of the drive, the brake will be released and operation enabled.
- On the other hand, if speed setpoint and actual speed fall below a parameterisable lower speed threshold, the brake will be applied under consideration of different time parameters.

- For operating modes with setpoint request via control signal (e.g. "PosExecute" in the Positioning" operating mode), the speed thresholds do not apply. Here the control logic opens and closes the holding brake through internal commands in the **Motion Control Kernel**.
- The brake will also be activated automatically if quick stop is activated in the drive, e.g. by a device command or as response to an error, and in the event of controller inhibit or pulse inhibit.
- After automatic brake activation and elapse of the brake application time, the controller is inhibited automatically by the basic "Holding brake control" function.

**Tip!**

The 2/12 mode is the common mode to control the brake.

#### Modus 13: Semi-automatic control

In this mode, brake release and brake application can be directly controlled via the input *bBrkReleaseBrake* without special logic or automatic.

In contrast to the manual operation (mode 11)

- the feedforward control is active in this mode, preventing a sagging e.g. in case of a hoist.
- the brake in this mode also closes when the controller is inhibited in order to prevent the axis in a hoist from falling.

**Related topics:**

- TroubleQSP (109)
- Behaviour in case of pulse inhibit (578)

### 9.12.2.2 Functional settings

The following bit coded functional settings for the holding brake control can be made in C02582:

| Bit | Option | Info |
|---|---|---|
| Bit 0 | bBrkReleaseOut invert. | Activation of inverted control<br>• "1" ≡ Inverted logic of trigger signal for holding brake control switching element |
| Bit 1 | Horizontal brake protection | Brake response in case of pulse inhibit<br>• "1" ≡ In the case of a pulse inhibit, the actual speed value is monitored which must reach the "Close" threshold value to cause the holding brake to be applied.<br>**Note:**<br>• This function is only active if bit 3 (horizontal/winding technology) is set as well. The function is used in order that, when the controller is inhibited, the holding brake of a drive with horizontal traverse path does not wear out during rotation.<br>• With vertical motion (bit 3 = 0), this function is not active. Especially with hoists and activated pulse inhibit of the controller, an immediate application of the brake is essential for safety-related reasons! |
| Bit 2 | with hoist inv. feedfwd. control | Direction of feedforward control with vertical/hoist technology:<br>• "0" ≡ Positive direction<br>• "1" ≡ Negative direction<br>**Note:**<br>Reversal (Ccw) is then considered. |
| Bit 3 | Horizontal application | Direction of movement of the axis<br>• "0" ≡ The axis performs vertical movements. Gravitational acceleration causes movements.<br>• "1" ≡ The direction of the axis is horizontal or rotary. The gravitational acceleration does not cause any movement. |
| Bit 4 | Feedforward control C2581 | Selection of the feedforward control value<br>• "0" ≡ Automatic selection.<br>– The torque saved at the last stop is used.<br>• "1" ≡ Manual selection.<br>– *bBrkStartValue* = FALSE: The feedforward control value 1 set in C02581/4 is used.<br>– *bBrkStartValue* = TRUE: The feedforward control value 2 set in C02581/5 is used. |
| Bit 5 | Feedback monitoring | Activation of status monitoring<br>• "1" ≡ The *bBrkApplied* input for status detection of the brake (via a switching contact at the brake) is monitored after the waiting time set in C02589/3 has expired. |
| Bit 6 | Sync ramp L_NSet_1<br>(from version 02.00.00) | Selection of the ramp time for the synchronisation process to setpoint speed after the brake opening time has elapsed<br>**Revised behaviour from version 02.00.00:**<br>• "1" ≡ The ramp time of the effective acceleration of the ramp function generator (L_NSet_1) is used (Lenze setting).<br>• "0" ≡ As before, the ramp time set in C02610/1 is used.<br>**Note:**<br>The changeover can be dynamically both via the ramp parameter and via bit 6. |
| Bit 7 | Reserved | |

**Related topics:**

- Behaviour in case of pulse inhibit (578)
- Feedforward control of the motor before release (579)

### 9.12.2.3 Switching thresholds

**Stop!**

Do not set the lower speed threshold for brake application too high to prevent an excessive wear of the brake!

**Note!**

For the speed comparison, only the absolute motor speed value is considered, the direction of rotation is not taken into account.

**Upper speed threshold for brake release:**

Switching threshold (C02581/1) + hysteresis for release (C02581/2)

**Lower speed threshold for brake application:**

Switching threshold (C02581/1) - hysteresis for application (C02581/3)

**Tip!**

The lower speed threshold for brake application should be set to approximately 5 ... 20 % of the maximum speed to minimise the wear of the brake and provide for an optimum brake reaction by a low grinding of the brake.

**Related topics:**

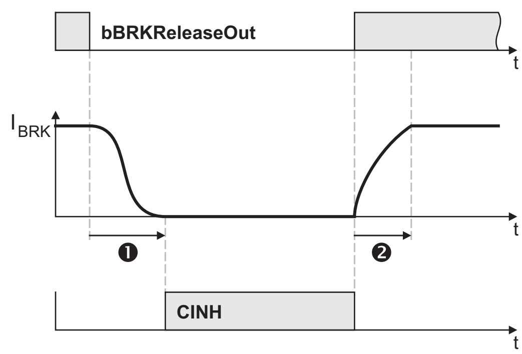
- Process when brake is released (575)
- Process when brake is closed (576)

### 9.12.2.4 Application and release time

**Danger!**

A wrong setting of the application and release time can cause a faulty control of the brake!

- If the application time is set too low, the controller is inhibited and the drive becomes torqueless before the brake is applied completely.




❶ C02589/1: Application time
❷ C02589/2: Release time
CINH = controller inhibit

- Every mechanical holding brake comes with a construction-conditioned application and release time which must be considered by the holding brake control and is set in C02589.
- The application and release time of the Lenze holding brake is indicated in the supplied operating instructions in the "Technical data" chapter.
- If the application and release times are too long, this is uncritical in respect of safety but leads to unnecessarily long delays during cyclical braking processes.

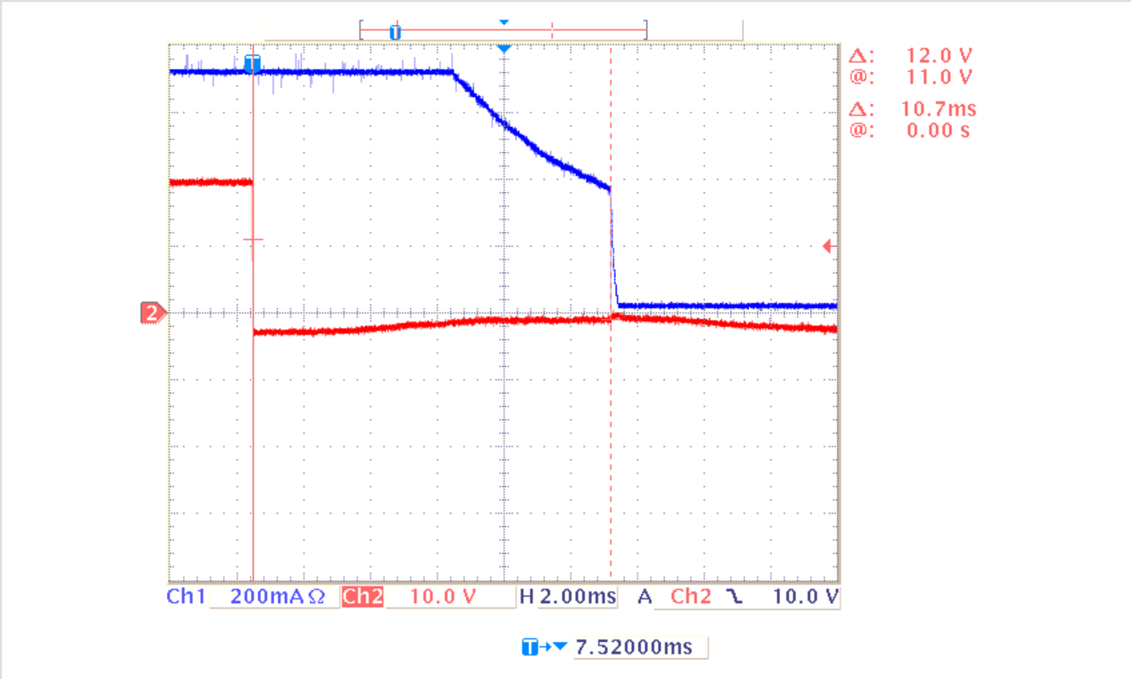
[9-28] Definition of the application and release time with the example of the PM brake

**Tip!**

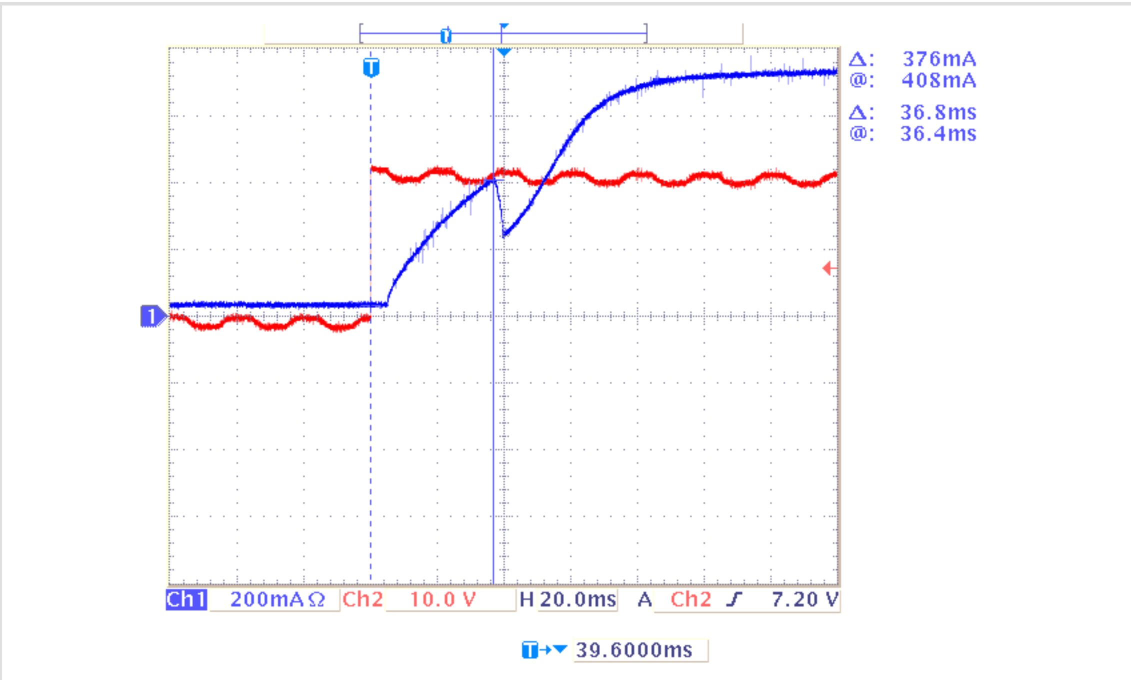
The application and release times do not only vary between the brake types but also depend on the basic conditions in the plant:

- Parameters of the hardware (cable length, temperature, level of supply voltage etc.)
- Contact elements used (brake module or contactor at the digital output)
- Type of overvoltage limitation/suppressor circuit

For optimisation purposes, detect in individual cases the response times by measurement.




[9-29] Oscillogram 1: Current characteristic when a mechanical holding brake is closed (application time: 10.7 ms)




[9-30] Oscillogram 2: Current characteristic when a mechanical holding brake is released (release time: 36.8 ms)

**Related topics:**


- Process when brake is released (575)
- Process when brake is closed (576)

### 9.12.2.5 Ramp time for approaching the setpoint speed

For the "Speed follower" operating mode, a ramp time can be set in C02610/1 if the setpoint is already out of reach while the holding brake is initiating the feedforward control process.

**Note!**

The "Homing", "Manual jog", and "Positioning" operating modes are based on a different control/release process of the holding brake. In these operating modes, the C02610/1 setting parameter does not have any effect!

**Example:**

A setpoint of 90 % is selected via the ramp function generator while the brake is applied (controller is inhibited).

1. At the set ramp (in most cases C00012), the ramp function generator ramps up to 90 %.
2. The brake identifies the setpoint selection of 5 % (release switching threshold). The feedforward control of the brake provides 3 % of the setpoint and will not report the release of the brake after approx. 1 s has expired.

Conclusion: 90 % of the selected setpoint is already ramped up while the brake is only providing 3 % of the setpoint via the feedforward control.

Since at this point a step change from 3 % to 90 % may cause mechanical jerks, the setpoint is ramped up from 3 % to 90 %, using the ramp time set in C02610/1 (Lenze setting: 2 s).

Our example is based on the V/f characteristic control (VFCplus) operating mode since the servo control (SC) operating mode does not use the switching threshold for the application of the holding brake for speed feedforward control. However, ramping up to a setpoint that is out of reach is carried out for all motor control operating modes because there is always a mechanical/electric delay when controlling a holding brake.

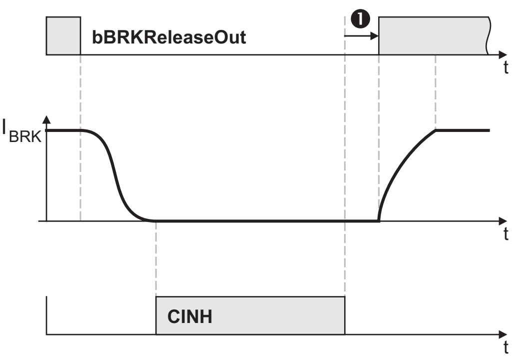
This delay is due to:

- Motor magnetisation (in the case of servo control only)
- Mechanical delay of all switching elements connected upstream to the holding brake
- Mechanical delay of the holding brake itself
- Generation of the holding torque by the motor

**Related topics:**

- Process when brake is released (575)

### 9.12.2.6 Motor magnetising time (only with asynchronous motor)




❶ Motor magnetising time
CINH = controller inhibit

[9-31] Considering the motor magnetising time taking the PM brake as an example

- When an asynchronous motor is used, first the magnetic field required for the holding torque is created (which is already available when a synchronous motor is used) after the controller inhibit is deactivated.
- The motor is internally magnetised through internal feedforward control of the lower speed threshold. The release time set in C02589/2 is considered here.

**Related topics:**

▸ Process when brake is released (📖 575)

### 9.12.2.7 Actual value monitoring

If an actual value monitoring time > 0 s is selected in C02593/1, the actual speed time monitoring is active.

- The monitoring time starts when the speed setpoint has reached the lower switching threshold and the actual speed is still above this threshold. (see illustration [9-34] in chapter "Process when brake is closed".)
- If the actual speed is still above the threshold when the monitoring time has expired, the brake will be automatically applied in the automatic brake control mode (mode 12).

**Note!**

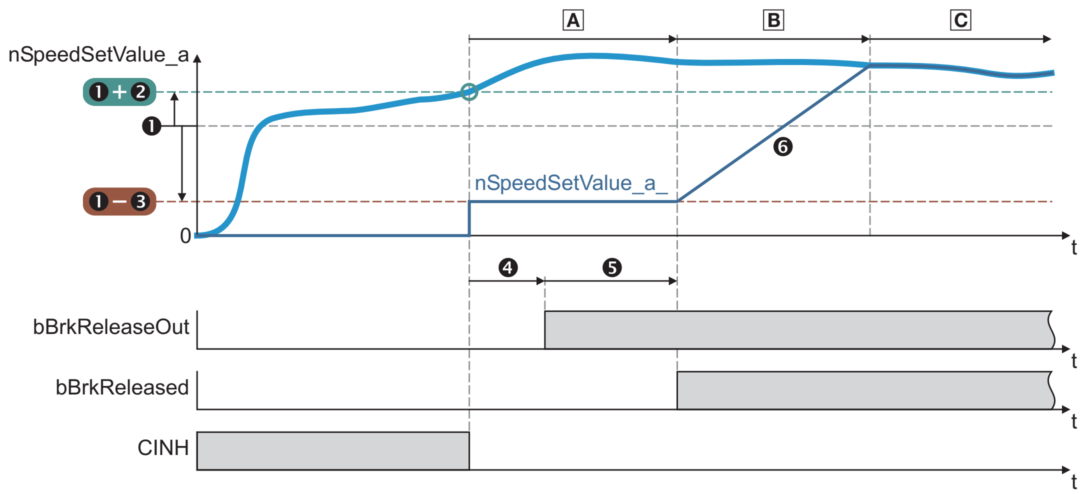
In the Lenze setting, the actual speed time monitoring is deactivated (C02593/1 = "0 s"), i.e. the brake will only be applied when the actual speed has reached the lower switching threshold.

### 9.12.3 Process when brake is released

1. The controller inhibit is deactivated.
2. The magnetic field required for the holding torque is created in the motor (is already available when a synchronous machine is used).
3. The *bBrkReleaseOut* trigger signal for holding brake switching element is set to TRUE for releasing the brake.
4. After the brake opening time has elapsed:
   - The *bBrkReleased* status signal ("brake released") is set to TRUE.
   - In the "Speed follower" operating mode, the drive synchronises to the already accelerated speed setpoint.
   - In the operating modes with setpoint request via control signal (operating modes "Homing", "Manual jog" and "Positioning"), the ramping process starts after the brake release at 0.
5. After the waiting time set in C02589/3 has additionally expired, the status monitoring starts again (if activated via bit 5 in C02582).

#### Time diagram

A Feedforward control
B Setpoint synchronisation
C Follow setpoint




❶ C02581/1: Switching threshold
❷ C02581/2: Hysteresis for release
❸ C02581/3: Hysteresis for application
❹ Time slot for feedforward control/magnetising
❺ C02589/2: Release time
❻ C02610/1: Ramp time for approaching the setpoint speed

[9-32] Release holding brake in automatic mode via speed threshold

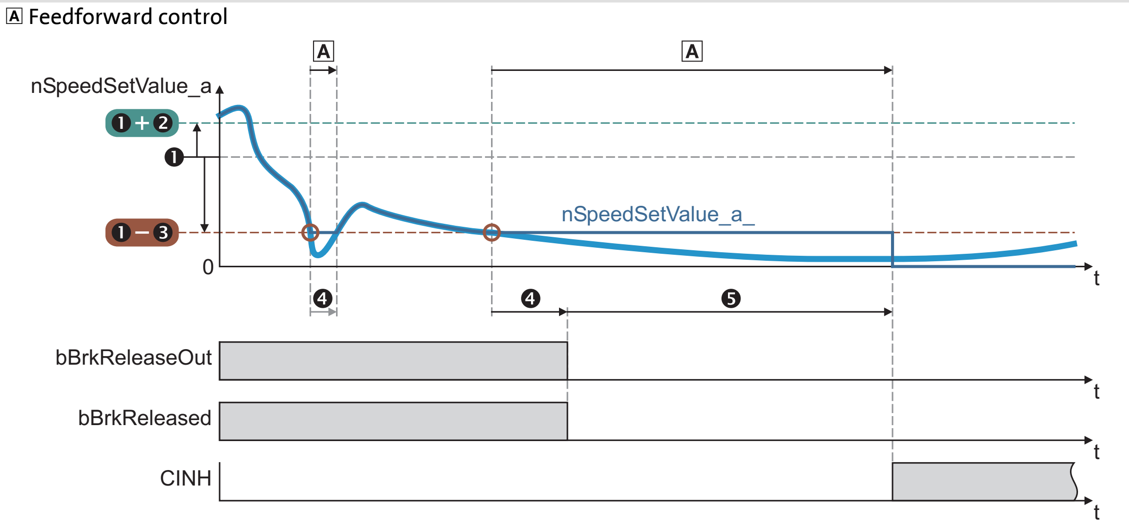
**Related topics:**

- Feedforward control of the motor before release (📖 579)

### 9.12.4 Process when brake is closed

1. The motor is decelerated when the setpoint is reduced by the user (e.g. turn down the potentiometer, setpoint selection via CAN).
   – The motor can also be decelerated by the "Quick stop" function or by "DC-injection braking", either directly requested by the user or as response to an error.
2. If the speed setpoint and the actual speed have fallen below the lower speed threshold or only the speed setpoint has fallen below the lower speed threshold and the actual value monitoring time has expired:
   – The *bBrkReleaseOut* trigger signal for the holding brake switching element is set to FALSE for closing the brake.
   – The *bBrkReleased* status signal is reset to FALSE.
   – In the operating modes with setpoint request via control signal (operating modes "Homing", "Manual jog" and "Positioning"), the brake closes depending on the internal state *bBusy* (setpoint generation through active profile generator) of the **Motion Control Kernel**.
   – The elapse of the brake application time starts.
3. After the brake application time has expired, the controller is inhibited.
4. After the waiting time set in C02589/3 has additionally expired, the status monitoring starts again (if activated via bit 5 in C02582).
5. In order to prevent the drive from further rotating/accelerating in the event of an error of the feedback contact, controller inhibit is cancelled again and the drive is held at standstill in a speed-controlled manner.

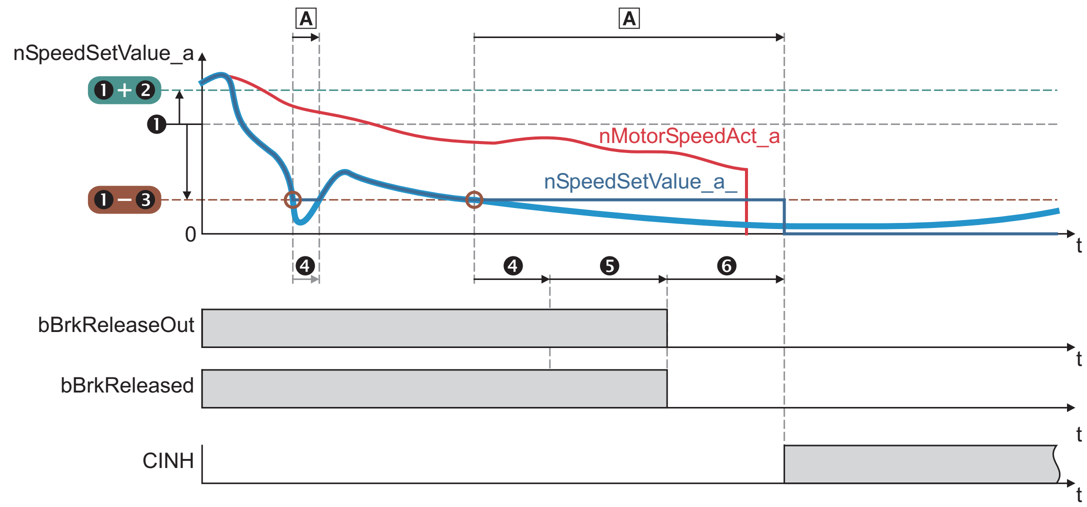
#### Time diagrams




❶ C02581/1: Switching threshold
❷ C02581/2: Hysteresis for release
❸ C02581/3: Hysteresis for application
❹ C02593/2: Application delay
❺ C02589/1: Application time

[9-33] Close holding brake in automatic mode via speed threshold (actual value = setpoint)

A Feedforward control




❶ C02581/1: Switching threshold
❷ C02581/2: Hysteresis for release
❸ C02581/3: Hysteresis for application
❹ C02593/2: Application delay
❺ C02593/1: Actual value monitoring time
❻ C02589/1: Application time

[9-34] Close holding brake in automatic mode with actual value monitoring time (C02593/1 > 0 s)

### 9.12.5 Behaviour in case of pulse inhibit

Setting the pulse inhibit causes a load-controlled coasting of the motor until the pulse is enabled again. In the enabled controller, the pulse can be inhibited e.g. due to a DC overvoltage, DC undervoltage or the "Safe torque off" request.

The brake response to pulse inhibit can be parameterised under C02582.

**Stop!**

For parameterising the response to pulse inhibit in C02582, the energy conditions of the machine should be evaluated first.

The energy stored in the machine can be considerably higher than the permissible switching energy and thus lead to the destruction of the brake if applied directly!

#### Activate brake immediately when pulse is inhibited

If bit 1 is set to "0" in C02582 (Lenze setting), the brake will be immediately applied when the pulse is inhibited to avoid damage to the mechanical components.

Especially in the case of hoist drives, immediate engagement of the brake is absolutely necessary for safety reasons if the pulse inhibit function of the drive controller has been activated!

**Danger!**

This behaviour is valid in (semi) automatic operation when the *bBrkRelease* input is set to FALSE.

When the *bBrkRelease* input is set to TRUE (supervisor operation) in automatic mode, the brake is not applied at pulse inhibit!

#### Only activate brake below threshold for brake activation

If bit 1 and bit 3 are set to "1" in C02582, the brake remains released until the lower speed threshold is reached to avoid an excessive wear of the brake.

- The braking action only takes places due to the friction in the load mechanics.
- Only when the motor speed has reached the threshold for brake activation, the brake will be closed. Thus, the function depends on the signal of the speed encoder.

During uncritical operation (horizontal loading condition), delayed brake application may be required to protect the brake in case of high centrifugal masses.

In case of vertical motion (Bit 3 = 0), this function is not active due to safety-related reasons.

**Related topics:**

- Functional settings (569)
- Switching thresholds (570)

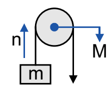
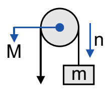
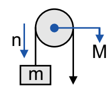
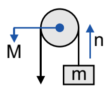
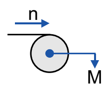
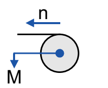
## 9.12.6 Feedforward control of the motor before release

The motor is precontrolled by selecting the lower speed threshold for applying the brake (control modes without feedback). When the upper speed threshold for brake release is reached, the motor is precontrolled with the for the release time set in C02589 with the lower threshold value before the brake switches to the release mode.

The direction of the feedforward control depends on two conditions:

1. On the settings selected under C02582:
   - Bit 2 = feedforward control inverted (for vertical drives/hoists)
   - Bit 3 = Direction of movement of the axis
2. On the sign of the setpoint.

### Truth table for the direction of the feedforward control

| Setpoint | Direction | Feedforward control | Scheme | Direction | |
|---|---|---|---|---|---|
| | | | | Feedforward control value | Start value |
| n ≥ 0 | Vertical/hoist (C02582: Bit 3 = 0) | not inverted (C02582: Bit 2 = 0) |  | + | + |
| | | inverted (C02582: Bit 2 = 1) |  | − | + |
| n < 0 | | not inverted (C02582: Bit 2 = 0) |  | + | − |
| | | inverted (C02582: Bit 2 = 1) |  | − | − |
| n ≥ 0 | Horizontal/winding drive (C02582: Bit 3 = 1) | Inversion via bit 2 with horizontal direction not effective |  | + | + |
| n < 0 | | |  | − | − |

### Selection of the feedforward control value

The feedforward control value can be selected via bit 4 in C02582:

- Bit 4 = 0: Automatic selection
  - The torque saved at the last stop is used.
- Bit 4 = 1: Manual selection
  - *bBrkStartValue* = FALSE: The feedforward control value 1 set in C02581/4 is used.
  - *bBrkStartValue* = TRUE: The feedforward control value 2 set in C02581/5 is used.

### Additive torque

If the servo control (SC) operating mode has been selected, an additive torque value in [%] can be selected via the *nBrkTorqueAdd_a* input.

**Related topics:**

- Functional settings (📖 569)
- Switching thresholds (📖 570)

# 10 Diagnostics & error management

This chapter provides information on error handling, drive diagnostics, and fault analysis.

## 10.1 Basics on error handling in the controller

Many of the functions integrated into the controller can

- detect errors and thus protect the device from damage or overload, e.g. short-circuit detection, Ixt overload detection, overtemperature detection, etc.
- detect an operating error by the user, e.g. a missing memory module, a required or missing communication module, etc.
- output a warning signal if desired, e.g. if the speed is too high or too low, etc.

Depending on the importance, the error detection in the device responds very fast (e.g. short-circuit detection < 1 ms) or in a slower cycle (e.g. temperature monitoring approx. 100 ms).

All functions provided with an error detection (e.g. the motor control) supply information to a so-called error handler. The error handler is processed every 1 ms and evaluates all information.

In this evaluation, the so-called status determining error (display in C00168) and the current error (display in C00170) are generated, and the controller is caused to take the respective error status (e.g. TroubleQSP).

These two types of error information serve to diagnose errors systematically and contain the following information:

1. The error type (e.g. "Warning")
2. The error subject area (e.g. "CAN generally integrated")
3. The error ID within the error subject area

Together all types of information form the real error number which is unique in the whole device system. ▶ Structure of the 32-bit error number (bit coding) (📖 602)

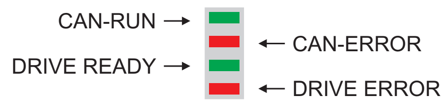
In addition to the control of the device status by the error handler, a logbook function records the errors and their histories. ▶ Logbook (📖 590)

**Tip!**

For many device errors, the error type and hence the response of the controller to the error can be parameterised. ▶ Setting the error response (📖 597)

## 10.2 LED status displays

Information on some operating statuses can quickly be obtained via LED displays:




[10-1] LED display on the controller front panel

| Labelling | Colour | Description | |
|---|---|---|---|
| CAN-RUN | green | CAN bus ok | LED status displays for the system bus (644) |
| CAN-ERR | red | CAN bus error | |
| DRIVE READY | green | Standard device ready for operation | LED status displays of the device status (583) |
| DRIVE ERROR | red | Warning/fault/error | |

### Note!

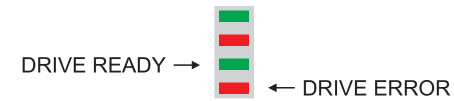
The four other LEDs have no function at present!

**Related topics:**

- Device control (DCTRL) (89)
- Device state machine and device statuses (101)
- System bus "CAN on board" (638)

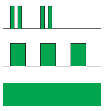
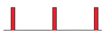
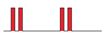
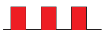
### 10.2.1 LED status displays of the device status

The control of the two "DRIVE READY" and "DRIVE ERROR" LED in the lower part on the front of the controller depends on the device status. ▶ Device state machine and device statuses (📖 101)

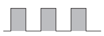



[10-2] DRIVE READY and DRIVE ERROR LED status displays

The meaning can be seen from the table below:

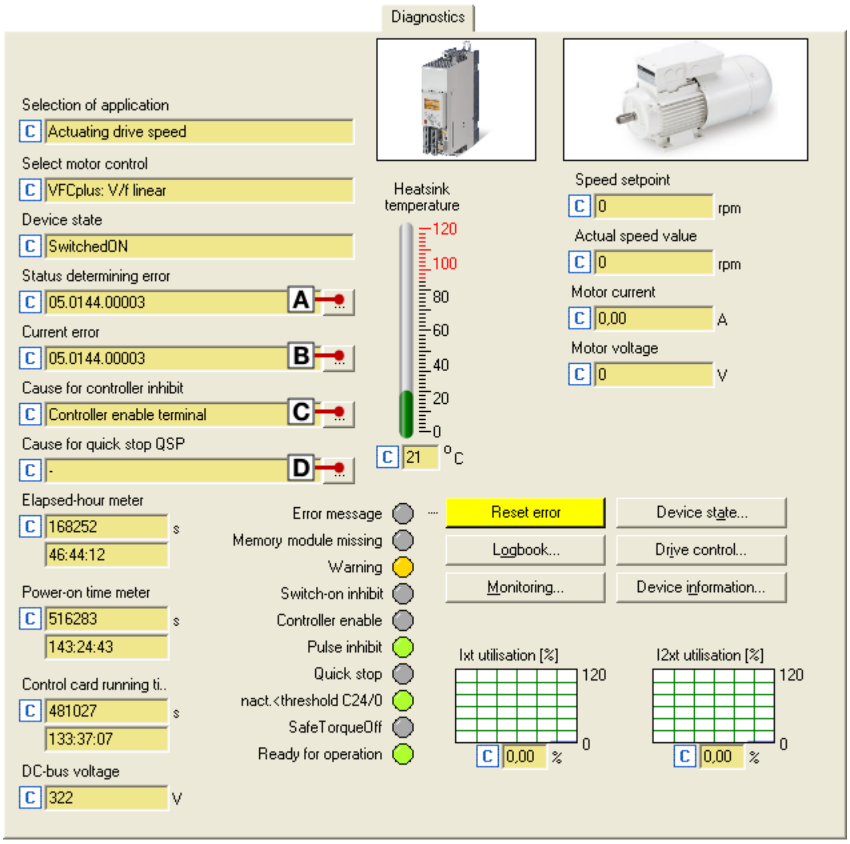
| DRIVE READY | DRIVE ERROR | Description | Device status (Display in C00137) |
|---|---|---|---|
| Off | Off | OFF or initialisation active | Init |
|  | Off | Safe torque off is active | SafeTorqueOff |
|  | Off | Device is ready to start | ReadyToSwitchOn |
|  | Off | Device is switched on | SwitchedOn |
|  | Off | Motor data identification/operation | OperationEnabled |
|  |  | The controller is ready to switch on, switched on or the operation is enabled and a warning is indicated. | |
|  |  | Fault active, quick stop | TroubleQSP |
| Off |  | Trouble is active | Trouble |
| Off |  | Fault is active | Fault |
| Off |  | System fault is active | SystemFault |

**Legend**
The symbols used for indicating the LED states have the following meaning:

| | |
|---|---|
|  | LED is flashing once approx. every 3 seconds (*slow flash*) |
|  | LED is flashing once approx. every 1.25 seconds (*flash*) |
|  | LED is flashing twice approx. every 1.25 seconds (*double flash*) |
|  | LED is blinking every second |
|  | LED is permanently on |

## 10.3 Drive diagnostics with the »Engineer«

When an online connection to the controller has been established, the connected controller can be diagnosed and relevant actual controller states can be displayed in a clearly arranged visualisation using the »Engineer«:




| Button | | Function |
|---|---|---|
| … | A | Display details of the status determining error. |
| | B | Display details of the current error. |
| | C | Display all active sources of a controller inhibit. |
| | D | Display all active sources of a quick stop. |
| **Resetting an error** | | Acknowledge fault message (if the error cause has been eliminated). |
| **Logbook...** | | Display the Logbook of the controller. (📖 590) |
| **Monitoring...** | | Configure the Monitoring. (📖 595) |
| **Device status...** | | Display the internal state machine including the current device status. |
| **Drive control...** | | Display the bit assignment of the following control-related words:<br>• MCI control word (C00136/1)<br>• CAN control word (C00136/2)<br>• Cause of controller inhibit (C00158)<br>• Cause of quick stop (C00159)<br>• Status word (C00150)<br>• Status word 2 (C00155) |
| **Device information...** | | Display identification data, e.g. firmware information or serial number of individual controller components. |

**How to diagnose a drive with the »Engineer«:**

1. Select the 8400 TopLine controller in the *project view*.
2. Click the icon or execute the **Online→Go online** command to establish an online connection to the controller.
3. Select the **Diagnostics** tab.
   - With an online connection, the **Diagnostics** tab displays current status information about the controller.

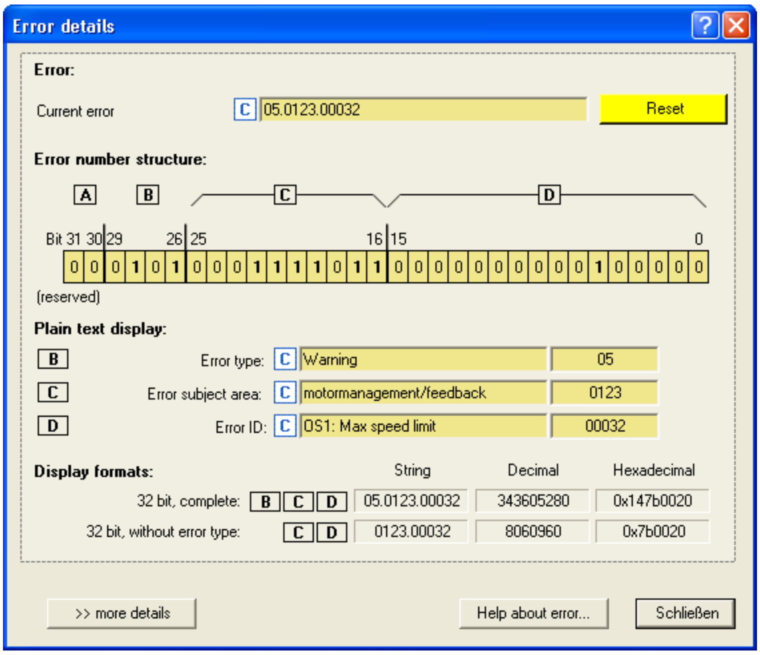
**Tip!**

The online connection to the controller can be established via the following device interfaces:

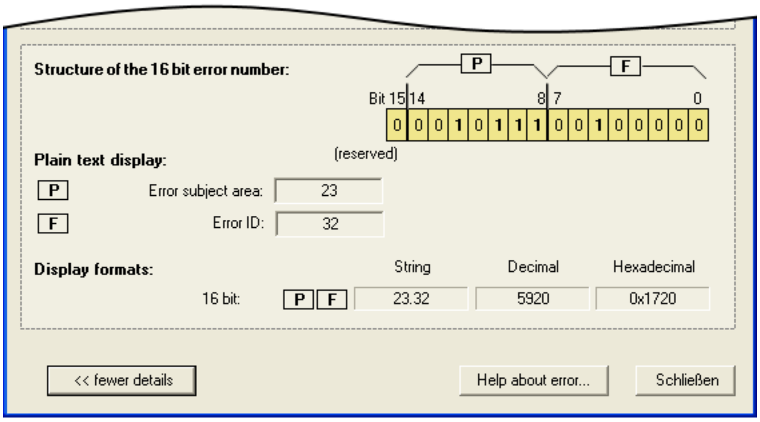
- CAN interface X1
  Diagnostics via the System bus "CAN on board" (638)
- X6 diagnostic interface
  We recommend this diagnostic connection when the X1 CAN interface is used for process communication.

## 10.3.1 Display details of the status determining error.

If you go to the **Diagnostics** tab and click the ... button for the status determining or current error, the *Error details* dialog box displays further information on the error:




- Click the **Help about error...** button to open the online help with information on the error cause and possible remedies.
- The **>> more details** button serves to provide more information about the structure of the 16-bit error number:

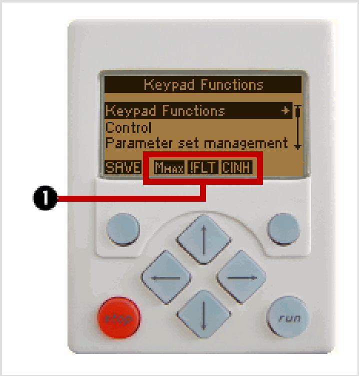



**Related topics:**

- Structure of the 32-bit error number (bit coding) (📖 602)
- Structure of the 16-bit error number (bit coding) (📖 605)

## 10.4 Drive diagnostics via keypad/bus system

### Keypad display of the controller status




▶ When the keypad is connected to the diagnostic interface X6 at the front of the controller, the status of the controller is displayed via different symbols on the LCD display in the area ❶.

| Symbol | Meaning | Note |
|---|---|---|
| RDY | Controller is switched on. | ▸ SwitchedOn (📖 107) |
| RUN | Controller is enabled. | |
| QSP | Quick stop is active | |
| CINH | Controller is inhibited. | The power outputs are inhibited. |
| Mmax | Speed controller 1 in the limitation. | The drive is torque-controlled. |
| Imax | Set current limit has been exceeded in motor or generator mode. | |
| IMP | Pulse inhibit is active | The power outputs are inhibited. |
| !SFLT | System fault is active | |
| !FLT | Fault | ▸ Fault (📖 111) |
| !TRB | Trouble | ▸ Trouble (📖 110) |
| !TQSP | TroubleQSP | ▸ TroubleQSP (📖 109) |
| WRN | Warning is active | |

### Display parameters

The parameters listed in the following tables serve to query current states and actual values of the controller for diagnostic purposes, e.g. by using the keypad, a bus system or the »Engineer« (with an online connection to the controller).

- These parameters are listed in the »Engineer« parameter list and the keypad in the **Diagnostics** category.
- A detailed description of these parameters can be found in the chapter "Parameter reference" (📖 742).

| Parameter | Display |
|---|---|
| C00051 | MCTRL: Actual speed value |
| C00052 | Motor voltage |
| C00053 | DC-bus voltage |
| C00054 | Motor current |
| C00056/1 | Torque setpoint |
| C00056/2 | Actual torque value |
| C00058 | Output frequency |
| C00061 | Heatsink temperature |
| C00064/1 | Device utilisation (Ixt) |
| C00064/2 | Device utilisation (Ixt) 15s |
| C00064/3 | Device utilisation (Ixt) 3 min |
| C00133 | Brake resistor utilisation |
| C00136/1 | MCI control word |
| C00136/2 | CAN control word |
| C00137 | Device status |
| C00138/1 | SYS control signals |
| C00138/2 | MCK control signals |
| C00138/3 | FWM control signals |
| C00150 | Status word |
| C00158 | Cause of controller inhibit |
| C00159 | Cause of quick stop QSP |
| C00165/1 | Status determining error (displayed as a numeric text) |
| C00165/2 | Current error (displayed as a numeric text) |
| C00168 | Status determining error (display of 32-bit number) |
| C00170 | Current error |
| C00166/1 | Error type, status determining |
| C00166/2 | Error subject area, status determining |
| C00166/3 | Error ID, status determining |
| C00166/4 | Error type, current |
| C00166/5 | Error subject area, current |
| C00166/6 | Error ID, current |

| Parameter | Display |
|---|---|
| C00177/1 | Switching cycles mains switching |
| C00177/2 | Switching cycles output relay |
| C00177/3 | Stress counter - short circuit |
| C00177/4 | Stress counter - earth fault |
| C00177/5 | Stress meter clamp |
| C00178 | Time the controller was enabled (elapsed-hour meter) |
| C00179 | Power-up time (power-on time meter) |
| C00180/1 | Running time - control card |
| C00180/2 | Running time - heatsink fan |
| C00180/3 | Running time - internal fan |

### Identification data

The parameters listed in the following table belong to the **Identification →Controller** category of the »Engineer« parameter list and the keypad and serve to display the identification data of the controller:

| Parameter | Display |
|---|---|
| C00099 | Firmware version (as a string) |
| C00100 | Firmware version (divided into subitems) |
| C00200 | Firmware product type |
| C00201/1...6 | Firmware of the control card and the power section |
| C00203/1...9 | Product type code of the individual device components |
| C00204/1...9 | Serial numbers of the individual device components |

## 10.5 Logbook

The integrated logbook function of the controller chronologically logs important events within the system and plays an important role for troubleshooting and controller diagnostics.

### Events that can be logged

The following events can be logged in the logbook:

- Error messages of the operating system (📖 602)
- Error messages generated by the application (via LS_SetError)
- Loading/saving of parameter sets, loading of the Lenze setting *(in preparation)*
- Transmitting the firmware to the controller *(in preparation)*
- Switching the controller on/off

### Information saved

For each event, the following information is saved in the logbook:

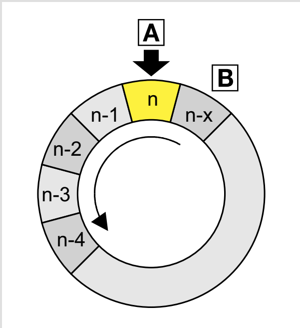
- Type of response to the event (e.g. fault, warning or information)
- Subject area that activated the event (e.g. CAN or USER).
- Event
- Value of power-on time meter
- Selected process values (analog % signals, binary signals)

### Memory depth

Maximum number of logbook entries: 14 *(extension in preparation)*

### 10.5.1 Functional description

The structure of the logbook corresponds to a ring buffer:




- As long as free logbook memory locations are available, the entries will be saved to the next free memory location (A).
- If all memory locations are occupied, the oldest entry (B) will be deleted in favour of a new entry.
- The newest entries will always remain available.

**Note!**

In the event of a supply voltage failure, the logbook is saved and reloaded automatically when the controller is switched on. This ensures that the error history of the device does not get lost. For this reason it is very important to act with caution when deleting the logbook entries.

### 10.5.2 Filtering logbook entries

The logbook adds new entries to the ring buffer after they have been passed through a parameterisable filter. This filter helps you to exclude certain events from being entered into the logbook which would trigger a certain error response (fault, trouble, warning, information, etc.).

C00169 (bit 1 ... bit 6) includes a bit coded specification of the events which are to be entered into the logbook. In the Lenze setting, all events are entered into the logbook.

**Note!**

Events with the "No response" setting are not entered into the logbook.

#### Counter for multiple entries

In order to prevent the ring buffer from overflowing with identical errors with frequent occurrence e.g. during commissioning, identical errors will not lead to new line entries in the configuration of the logbook in the Lenze setting. Instead, one counter will be counted up for this error.

- The time of the error is always the time of its first occurrence. Hence, a new logbook line will only be generated if a new error occurs.
- The error counter can be deactivated by resetting bit 9 in C00169.

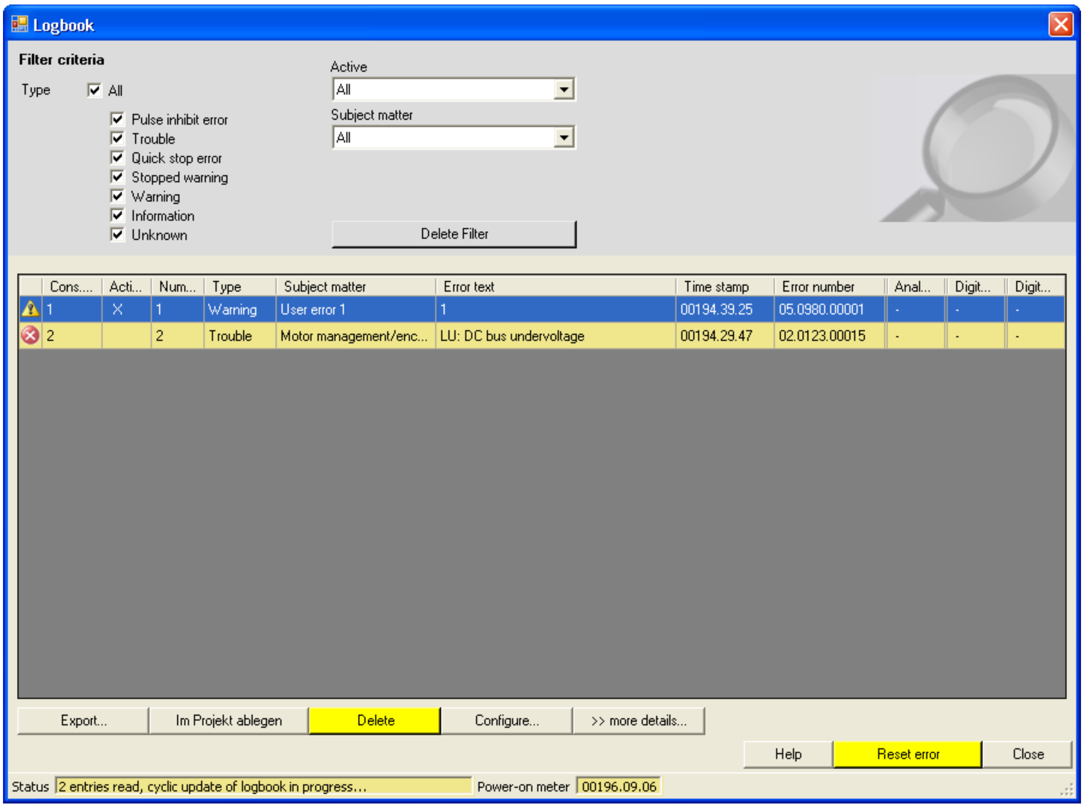
### 10.5.3 Reading out logbook entries

With an online connection, the existing logbook entries can easily be displayed in the »Engineer«. Alternatively, the logbook entries can also be read out via the corresponding parameters (e.g. using the keypad).

**How to display logbook entries in the »Engineer«:**

1. Select the 8400 TopLine controller in the *project view*.
2. Click the icon or execute the **Online→Go online** command to establish an online connection to the controller.
3. Select the **Diagnostics** tab from the *Workspace*.
4. Click **Logbook**.

**Example: Representation of the logbook in the »Engineer« V2.13**




| Button | Function |
|---|---|
| **Reset filters** | Reset set filter criteria to display all available logbook entries. |
| **Export...** | Export the entries available in the logbook into a *.log file.<br>▸ Exporting logbook entries to a file (📖 593) |
| **File in project** | File the current logbook in the Engineer project to be able to access it offline, too.<br>▸ File logbook in project (📖 594) |
| **Delete** | Delete all entries available in the logbook. |
| **Configure...** | Open parameterisation dialog for configuring the logbook. |

| Button | Function |
|---|---|
| **>> more details** | Show more details:<br>• Analog value 1, digital values 1 & 2<br>• More output types of the error numbers (32-bit, internal 32-bit and internal 16-bit).<br>• Instead of the **>> more details** button, the **<< less details** is now displayed via which the details can be hidden again. |
| **Help** | Open online help for the logbook. |
| **Resetting an error** | Acknowledge existing error message if the error cause has been eliminated and thus the error is not pending anymore.<br>• After the reset (acknowledgement) of the current error, further errors may be pending which must also be reset. |
| **Close** | Close the *Logbook* dialog box again. |

### 10.5.4 Exporting logbook entries to a file

**How to export the logbook entries to a file:**

1. Go to the *Logbook* dialog box and click the **Export...** button.
   - The *Export logbook* dialog box is displayed.
2. Specify the folder, file name, and file type for the file.
3. Click the **Save** button to export the logbook entries to the specified file.
   - Hidden logbook entries are not exported, i.e. the filter criteria specified are accounted for during the export.
   - The logbook entries are written to the file in the form of a semicolon separated list.

#### Structure of the semicolon separated list

The list includes the following information:

1. Cons. no.
2. Active
3. Numerator
4. Type
5. Subject area
6. Error text
7. Time stamp
8. Relative time
9. Error number
10. Source - analog value 1
11. Analog value 1
12. Source - digital value 1
13. Digital value 1
14. Source - digital value 2
15. Digital value 2

### 10.5.5 File logbook in project

If you want to display the currently available logbook entries at a later date in offline mode, i.e. without a connection to the controller, you can file the current logbook in the project.

**How to file the logbook in the project:**

Go to the *Logbook* dialog box and click the **File in project** button.

- The logbook with all the entries uploaded up to now is filed in the Engineer project independent of the set filter criteria.
- A logbook of the same device already filed before will be overwritten without querying the user.
- The filter settings are not filed in the project.
- When a logbook is filed in the project, the logbook can also be opened in offline mode via the **Logbook** button on the **Diagnostics** tab.

**Note!**

Filing the logbook changes the project.

- When the project is closed, you are asked to save the changed project.
- Only if the changed project is saved, the new logbook entries filed in the project remain stored.

## 10.6 Monitoring

The controller is provided with various monitoring functions which protect the drive against impermissible operating conditions.

- If a monitoring function responds,
  - an entry will be made into the Logbook of the controller,
  - the response (TroubleQSP, Warning, Fault, etc.) set for this monitoring function will be triggered,
  - the status of the internal device control changes according to the selected response, controller inhibit is set, and the "DRIVE ERROR" LED on the front of the controller goes on:

| Response | Entry in the logbook | Display in C00168 | Pulse inhibit | Controller inhibit | Acknowledgement required | LED "DRIVE ERROR" |
|---|---|---|---|---|---|---|
| None | | | | | | Off |
| Fault | ☑ | ☑ | ☑ | ☑ | ☑ | |
| Trouble | ☑ | ☑ | ☑ | ☑ (after 0.5 s) | | |
| TroubleQSP | ☑ | ☑ | | | ☑ | |
| WarningLocked | ☑ | ☑ | | | ☑ | |
| Warning | ☑ | ☑ | | | | |
| Information | ☑ | | | | | Off |
| System error | ☑ | ☑ | ☑ | ☑ | Mains switching is required! | |

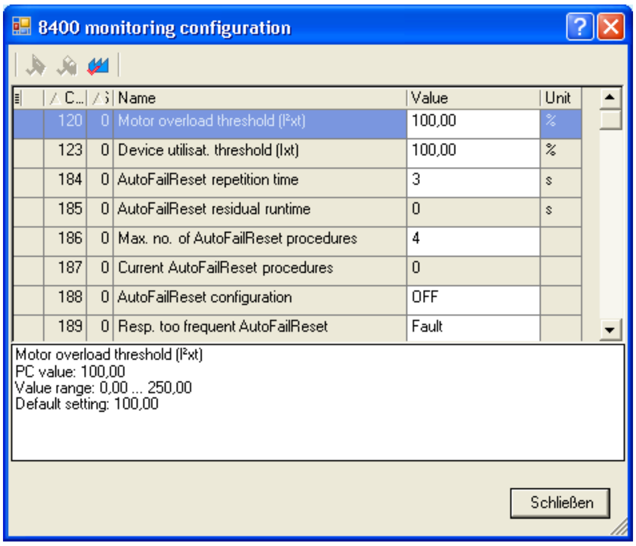
**Related topics:**

- LED status displays of the device status (📖 583)
- Device state machine and device statuses (📖 101)
- Device overload monitoring (Ixt) (📖 275)
- Motor load monitoring (I2xt) (📖 276)
- Motor temperature monitoring (PTC) (📖 278)
- Brake resistor monitoring (I2xt) (📖 279)
- Motor phase failure monitoring (📖 281)
- Mains phase failure monitoring (📖 283)
- Maximum current monitoring (📖 283)
- Maximum torque monitoring (📖 284)
- Resolver/multi-encoder open-circuit monitoring (📖 320)
- Open-circuit monitoring HTL encoder (📖 322)
- Motor temperature monitoring (KTY) (📖 323)

### 10.6.1 Monitoring configuration

**How to configure the monitoring functions using the »Engineer«:**

1. Select the 8400 TopLine controller in the *project view*.
2. Select the **Diagnostics** tab from the *Workspace*.
3. Click the **Monitoring...** button.
   - The *8400 monitoring configuration* dialog box is displayed via which the desired settings can be made:



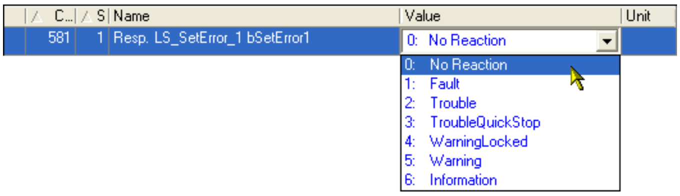

**Related topics:**


▸ Setting the error response (597)

## 10.6.2 Setting the error response

When a monitoring function responds, the response set for this monitoring function (TroubleQSP, Warning, Fault, etc.) will be triggered.

- For many monitoring functions the response can be individually parameterised via parameters.




**Tip!**

The table in chapter "Short overview (A-Z)" contains the error messages for which the response can be set . (📖 608)

### Warning thresholds

Some of the monitoring functions are activated if a defined warning threshold (e.g. temperature) has been exceeded.

- The corresponding preset threshold values can be changed via the following parameters:

| Parameter | Info |
|---|---|
| C00120 | Setting of motor overload (I²xt) |
| C00123 | Device utilisat. threshold (Ixt) |
| C00572 | Brake resistor overload threshold |
| C00599 | Motor phase failure threshold |

### 10.6.3 AutoFailReset function

The AutoFailReset function serves to automatically reset the latching "Fault" and "TroubleQSP" errors as well as the latching "WarningLocked" warning.

The "latching" term means that the effect on the controller remains active even after the error cause has been removed.

To reset latching errors and warnings, the following options are available:

- Manual reset
  - with device command C00002/19 (activated by LOW/HIGH edge)
  - by a LOW/HIGH edge at the *bResetFail* input of the LS_DriveInterface system block (the "FailReset" control bit in the control word must be parameterised with a value of "1").
- Automatic reset
  - using the AutoFailReset function.

#### Overview of the relevant parameters

| Parameter | Info |
|---|---|
| C00184 | Repetition time of the error reset processes |
| C00185 | Time left until the next error reset process |
| C00186 | Max. number of error reset processes |
| C00187 | Current number of error reset processes carried out ineffectively |
| C00188 | Configuration of the AutoFailReset function<br>• 0: Off<br>• 1: Fault + TroubleQSP<br>• 2: WarningLocked<br>• 3: All locking |
| C00189 | Response after max. number of error reset processes has been reached |
| Highlighted in grey = display parameter | |

## 10.7 Maloperation of the drive

| Maloperation | Cause | Remedy |
|---|---|---|
| Motor does not rotate | DC-bus voltage is too low<br>• Red LED is blinking every 1 s<br>• Display in the keypad: *LU* | Check mains voltage |
| | Controller is inhibited<br>• Green LED is blinking<br>• Display in the keypad: CINH | Deactivate controller inhibit<br>• Note: Controller inhibit can be set via several sources !<br>• C00158 displays all active sources for controller inhibit. |
| | Automatic start is inhibited<br>(Bit 0 in C00142 = 1) | LOW/HIGH edge at RFR<br>If required, correct starting condition with C00142 |
| | DC-injection braking (DCB) is active | Deactivate DC injection brake |
| | Mechanical motor brake is not released | Release mechanical motor brake manually or electrically |
| | Quick stop (QSP) is active<br>• Display in the keypad: IMP | Deactivate quick stop<br>• Note: Quick stop can be set via several sources!<br>• C00159 displays all active sources of quick stop. |
| | Setpoint = 0 | Select setpoint |
| | JOG frequency = 0 at activated JOG setpoint | Set JOG setpoint in C00039/1...15 |
| | Trouble is active | Clear fault |
| | With C00006 = 4 "SLVC: Vector control" has been set, but no motor parameter identification has been carried out. | Execute automatic motor parameter identification with the C00002/23 device command |
| | Assignment of several mutually exclusive functions with a signal source in C00701 | Correct configuration in C00701 |
| Motor rotates irregularly | Motor cable is defective | Check motor cable |
| | Maximum motor current in motor or generator mode is set too low | Adjust settings to the application: C00022: Imax in motor mode<br>C00023: Imax in generator mode |
| | Motor is underexcited or overexcited | Check parameterisation:<br>C00006: Motor control<br>C00015: VFC: V/f base frequency<br>C00016: VFC: Vmin boost |
| | Rated motor data (stator resistance, speed, current, frequency, voltage) and cos φ and/or magnetising inductance is not adapted to the motor data | Execute automatic motor parameter identification with the C00002/23 device command<br>- or -<br>Adjust motor parameters manually:<br>C00084: Motor stator resistance<br>C00087: Rated motor speed<br>C00088: Rated motor current<br>C00089: Rated motor frequency<br>C00090: Rated motor voltage<br>C00091: Motor cosine phi<br>C00092: Motor magnetising inductance |
| | Motor windings are wired incorrectly | Reverse from star connection to delta connection |

| Maloperation | Cause | Remedy |
|---|---|---|
| Motor consumes too much current | $V_{min}$ boost has been selected too high | Correct setting with C00016 |
| | V/f base frequency has been selected too low | Correct setting with C00015 |
| | Rated motor data (stator resistance, speed, current, frequency, voltage) and cos φ and/or magnetising inductance is not adapted to the motor data | Execute automatic motor parameter identification with the C00002/23 device command<br>- or -<br>Adjust motor parameters manually:<br>C00084: Motor stator resistance<br>C00087: Rated motor speed<br>C00088: Rated motor current<br>C00089: Rated motor frequency<br>C00090: Rated motor voltage<br>C00091: Motor cosine phi<br>C00092: Motor magnetising inductance |
| Motor parameter identification is aborted with error LP1 | Motor is too small compared to the rated device power (>1 : 3) | Use device with lower rated power |
| | DC injection brake (DCB) is active via terminal | Deactivate DC injection brake |
| Drive behaviour with vector control is not satisfactory | different | Optimise or manually adapt vector control |
| | | Execute automatic motor parameter identification with the C00002/23 device command |
| Torque dip in field weakening range<br>or<br>motor stalling when being operated in the field weakening range | Motor is overloaded | Check motor load |
| | Motor windings are wired incorrectly | Reverse from star connection to delta connection |
| | V/f reference point is set too high | Correct setting with C00015 |
| | Override point of field weakening is set too low | Correct setting with C00080 |
| | Mains voltage is too low | Increase mains voltage |
| An asynchronous motor with feedback rotates without control and with too low speed | Motor phases have been interchanged<br>• Thus the rotating field of the motor is not identical anymore with the rotating field of the feedback system.<br>• Therefore, the drive shows the following behaviour if V/f characteristic control (C00006 = 7) is performed:<br>– The motor rotates faster than the speed setpoint by the value set in C00074.<br>– After the controller has been enabled, the controller will not stop if the speed setpoint = 0 or a quick stop (QSP) occurs.<br>– Among other things, the final motor current depends on the value set for the $V_{min}$ boost and may rise up to $I_{max}$ which can trigger the "OC5: Ixt overload" fault message. | Check the phase position of the motor cable<br>If possible: Operate the motor with deactivated feedback (C00006 = 6) and check direction of rotation of the motor |
| Motor phase (LP1) monitoring does not respond if the motor phases are interrupted | Monitoring is not active (C00597 = 0) | Activate monitoring (C00597 = 1) |

## 10.8 Operation without mains supply

The following display parameters have a value of "0" if the mains supply is switched off and the external 24 V supply of the controller is switched on:

| Parameter | Info |
|---|---|
| C00050 | MCTRL: Speed setpoint |
| C00051 | MCTRL: Actual speed value |
| C00052 | Motor voltage |
| C00053 | DC-bus voltage |
| C00054 | Motor current |
| C00058 | Output frequency |
| C00061 | Heatsink temperature |
| C00064/1...3 | Device utilisation (Ixt) |
| C00066 | Thermal motor load ($I^2xt$) |
| C00177 | Switching cycles |
| C00725 | Current switching frequency |

## 10.9 Error messages of the operating system

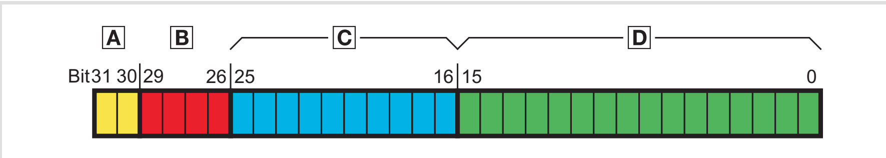
This chapter describes all error messages of the controller operating system and possible causes & remedies.

**Tip!**

Each error message is also saved to the logbook in chronological order. ▶ Logbook (📖 590)

### 10.9.1 Structure of the 32-bit error number (bit coding)

If an error occurs in the controller, the internal fault memory saves a 32-bit value which contains the following information:

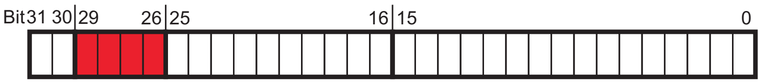



A Reserved
B Error type
C Error subject area
D Error ID

[10-3] Structure of the error number

- Display parameter: C00168
- For the sake of legibility, the error number in the logbook and in C00165 is displayed with the following syntax:
  [**Error type**].[**Error subject area no.**].[**Error ID**]

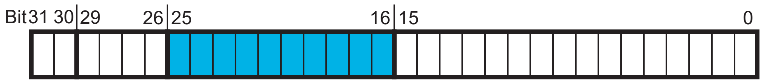
#### Error type




The error type gives information about the behaviour/response of the controller to the error. The error type for some device errors can also be parameterised.

| Bit 29 | Bit 28 | Bit 27 | Bit 26 | Meaning |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0: No response |
| 0 | 0 | 0 | 1 | 1: Fault |
| 0 | 0 | 1 | 0 | 2: Trouble |
| 0 | 0 | 1 | 1 | 3: TroubleQSP |
| 0 | 1 | 0 | 0 | 4: WarningLocked |
| 0 | 1 | 0 | 1 | 5: Warning |
| 0 | 1 | 1 | 0 | 6: Information |

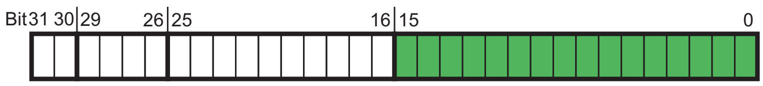
### Error subject area




The error subject area indicates the internal "function unit" of the controller in which the error has occurred:

| Error subject area | | Assigned errors | Remedy possible by user? |
|---|---|---|---|
| **No.** | **Name** | | |
| 111 | Supply voltage | Errors that occur in connection with the supply voltage of the device. | Yes |
| 119 | Temperature | Errors that occur for temperature reasons. | Yes |
| 123 | Motor management / encoder | Errors that occur within the motor control or encoder evaluation. | Yes |
| 125 | Analog I/O integrated | Errors that occur in connection with the analog inputs and outputs. | Yes |
| 127 | Extension module slot 1 | Errors that are reported by the extension module, and communication errors to the plugged-in extension module. | Yes if it is a fieldbus error. |
| 131 | CAN integrated (general) | Errors related to general CAN functions. | Yes |
| 135 | CAN process data object (PDO) | Errors that are explicitly only related to the CAN-PDO (process data objects). | Yes |
| 140 | Device configuration | Errors that occur due to incompatibilities of the plugged-in individual components (fieldbus module, safety module, et al.). | Yes if the error relates to a module plugged-in by the user. |
| 144 | Parameter set | Errors that occur in connection with the parameter set or the parameter set memory (memory module). | Yes if the error relates to a missing or incompatible memory module. |
| 145 | Device firmware (internal error) | Internal error of the device firmware. | No |
| 184 | MotionControlKernel | Errors that occur within the MotionControl basic functions (e.g. profile generation, brake control, positioning). | Yes |
| 400 | Defective device hardware | Errors that occur due to defective device hardware. | No |
| 444 | Fieldbus | Errors that occur in connection with fieldbus communication. | Yes |
| 980<br>...<br>983 | User error 1<br>...<br>User error 4 | Errors generated by the user (by the application) via the LS_SetError_1 system block. | Yes |
| 984<br>...<br>987 | User error 5<br>...<br>User error 8 | Errors generated by the user (by the application) via the LS_SetError_2 system block. | Yes |

### Error ID

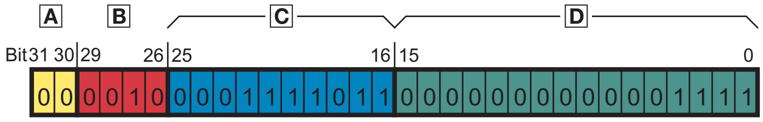



16-bit value (0 ... 65535) for error identification within the error subject area.

### Example for bit coding of the error number

C00168 displays an internal error number: "142278671".

▶ This decimal value corresponds to the following bit sequence:

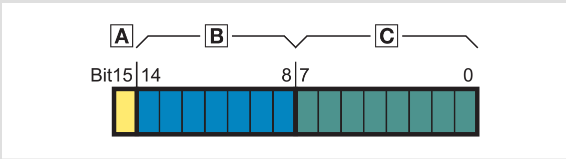



| Assignment | Information | Meaning in the example |
|---|---|---|
| 0 0 | Reserved | - |
| 0 0 1 0 | Error type | 2: Trouble |
| 0 0 0 1 1 1 1 0 1 1 | Error subject area | 123: Motor management / encoder |
| 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 | Error ID | 15: "LU: DC bus undervoltage" |

▶ Thus, error number "142278671" means:
A DC bus undervoltage has been detected in the "Motor management / encoder" subject area. The error response is a "Fault" which must be unlocked separately after the error has been eliminated.

## 10.9.2 Structure of the 16-bit error number (bit coding)

In addition to the 32-bit error number, a 16-bit error number is generated if an error occurs. It consists of the following information:

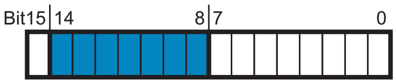



A Reserved
B Error subject area
C Error ID

[10-4] Structure of the error number

- Display parameter: C00160
- For the sake of legibility, the 16-bit error number in the logbook is displayed with the following syntax::
  [**Error subject area no.**].[**Error ID**]

### Error subject area



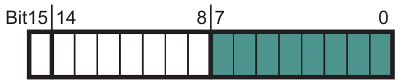

The error subject area indicates the internal "function unit" of the controller in which the error has occurred.

> **Note!**
>
> Due to the smaller value range (0 ...127), the number assignment to the error subject area differs from the 32-bit error number.

| Error subject area | | Assigned errors | Remedy possible by user? |
|---|---|---|---|
| **No.** | **Name** | | |
| 11 | Supply voltage | Errors that occur in connection with the supply voltage of the device. | Yes |
| 19 | Temperature | Errors that occur for temperature reasons. | Yes |
| 23 | Motor management / encoder | Errors that occur within the motor control or encoder evaluation. | Yes |
| 25 | Analog I/O integrated | Errors that occur in connection with the analog inputs and outputs. | Yes |
| 26 | Defective device hardware | Errors that occur due to defective device hardware. | No |
| 27 | Extension module slot 1 | Errors that are reported by the extension module, and communication errors to the plugged-in extension module. | Yes if it is a fieldbus error. |
| 31 | CAN integrated (general) | Errors related to general CAN functions. | Yes |
| 35 | CAN process data object (PDO) | Errors that are explicitly only related to the CAN-PDO (process data objects). | Yes |
| 40 | Device configuration | Errors that occur due to incompatibilities of the plugged-in individual components (fieldbus module, safety module, et al.). | Yes if the error relates to a module plugged-in by the user. |
| 44 | Parameter set | Errors that occur in connection with the parameter set or the parameter set memory (memory module). | Yes if the error relates to a missing or incompatible memory module. |
| 45 | Device firmware (internal error) | Internal error of the device firmware. | No |
| 54 | Fieldbus | Errors that occur in connection with fieldbus communication. | Yes |
| 84 | MotionControlKernel | Errors that occur within the MotionControl basic functions (e.g. profile generation, brake control, positioning). | Yes |

| Error subject area | | Assigned errors | Remedy possible by user? |
|---|---|---|---|
| No. | Name | | |
| 100<br>...<br>103 | User error 1<br>...<br>User error 4 | Errors generated by the user (by the application) via the LS_SetError_1 system block. | Yes |
| 104<br>...<br>107 | User error 5<br>...<br>User error 8 | Errors generated by the user (by the application) via the LS_SetError_2 system block. | Yes |

### Error ID




8-bit value (0 ... 255) for error identification within the error subject area.

**Tip!**

All possible 16-bit error numbers are listed in the table "Short overview (A-Z)" in the second column. (📖 608)

## 10.9.3 Reset error messages

An error message with the response "Fault", "Trouble", "TroubleQSP" or "Warning locked" must be explicitly reset (acknowledged) after the cause of the error has been eliminated.

To reset (acknowledge) a pending error message, execute the device command C00002/19 = "1".

**Tip!**

With an online connection to the controller, use the **Diagnostics** tab of the »Engineer« and click **Error reset** to reset a pending error message.

### 10.9.4 Export error texts

All error texts of the controller can be exported into a text file (*.txt) for further processing.

- The error text is preceded with the corresponding 32-bit and 16-bit error number (decimal).
- If there is no corresponding 16-bit error number for a 32-bit error number, the field remains empty.

**Example**

Output of the German and English error texts:

```
32-BitError   16-BitError   DE-de                        EN-en
0             0             Kein Fehler                  No error
111           11            Versorgungsspannung          Supply voltage
119 / 19
12323 : Motor management / encoder
125           25            E/A integriert               I/O integrated
...
26214416      6672          dH10: Lüfterausfall          dH10: Fan failure
26214505      6761          dH69: Abgleichdatenfehler    dH69: Adjustment fault
```

**How to export the error texts into a text file:**

1. Go to the *Project view* in the *context menu* of the 8400 TopLine controller and execute the **Export error texts...** command.
2. Define the following options in the *Export error texts* dialog box:
   - Output file and memory location
   - Languages to be exported (German/English/French)
   - Device/module to be exported
   - Separator (tabulator or semicolon)
   - Font (UTF8, standard font or ASCII)
3. Click **OK** to start the export.
   - After the export, a message appears indicating whether the export was successful.

## 10.9.5 Short overview (A-Z)

The table below contains all error messages of the controller operating system in alphabetical order.

**Note!**

For the sake of legibility, the Logbook and C00165 display the 32-bit error number with the following syntax:
[**Error type**].[**Error subject area no.**].[**Error ID**]

In this documentation, "xx", a wildcard, stands for the error type since it is configurable for many error messages.

**Tip!**

If you click the cross-reference in the first column, "Error number", you will reach the detailed description of the respective error message in the following chapter, "Cause & possible remedies". (611)

| Error number 32 bits | 16 bits$_{hex}$ | 16 bits$_{dec}$ | Error message | Response (Lenze setting) | can be set in | CAN Emergency Error Code |
|---|---|---|---|---|---|---|
| xx.0125.00001 | 0x1901 | 6401 | An01: AIN1_I < 4 mA | TroubleQuickStop | C00598/1 | 0xF000 |
| xx.0125.00002 | 0x1902 | 6402 | An02: AIN2_I < 4 mA | TroubleQuickStop | C00598/2 | 0xF000 |
| xx.0131.00006 | 0x1f06 | 7942 | CA06: CAN CRC error | No Reaction | C00592/1 | 0x8000 |
| xx.0131.00007 | 0x1f07 | 7943 | CA07: CAN Bus Warn | No Reaction | C00592/3 | 0x8000 |
| xx.0131.00008 | 0x1f08 | 7944 | CA08: CAN Bus Stopped | No Reaction | C00592/4 | 0x8000 |
| xx.0131.00011 | 0x1f0b | 7947 | CA0b: CAN HeartBeatEvent | No Reaction | C00592/5 | 0x8130 |
| xx.0131.00015 | 0x1f0f | 7951 | CA0F: CAN control word | Fault | C00594/1 | 0xF000 |
| xx.0127.00002 | 0x1b02 | 6914 | CE04: MCI communication error | No Reaction | C01501/1 | 0x7000 |
| xx.0127.00015 | 0x1b0f | 6927 | CE0F: MCI control word | Fault | C00594/2 | 0xF000 |
| xx.0135.00001 | 0x2301 | 8961 | CE1: CAN RPDO1 | No Reaction | C00593/1 | 0x8100 |
| xx.0135.00002 | 0x2302 | 8962 | CE2: CAN RPDO2 | No Reaction | C00593/2 | 0x8100 |
| xx.0135.00003 | 0x2303 | 8963 | CE3: CAN RPDO3 | No Reaction | C00593/3 | 0x8100 |
| xx.0131.00000 | 0x1f00 | 7936 | CE4: CAN Bus Off | No Reaction | C00592/2 | 0x8000 |
| xx.0184.00001 | 0x5401 | 21505 | Ck01: Pos. HW limit switch | TroubleQuickStop | C00595/1 | 0x8600 |
| xx.0184.00002 | 0x5402 | 21506 | Ck02: Neg. HW limit switch | TroubleQuickStop | C00595/2 | 0x8600 |
| xx.0184.00007 | 0x5407 | 21511 | Ck03: Pos. SW limit position | TroubleQuickStop | C00595/3 | 0x8600 |
| xx.0184.00008 | 0x5408 | 21512 | Ck04: Neg. SW limit position | TroubleQuickStop | C00595/4 | 0x8600 |
| xx.0184.00153 | 0x5499 | 21657 | Ck05: Error following error 1 | Warning | C00595/5 | 0x8611 |
| xx.0184.00154 | 0x549a | 21658 | Ck06: Following Error 2 | Warning | C00595/6 | 0x8611 |
| xx.0184.00155 | 0x549b | 21659 | Ck07: Travel range limit exceeded | TroubleQuickStop | C00595/7 | 0x8612 |
| xx.0184.00156 | 0x549c | 21660 | Ck08: Home position unknown | WarningLocked | C00595/8 | 0x8612 |
| xx.0184.08005 | 0x54cd | 21709 | Ck09: Positioning mode invalid | WarningLocked | C00595/9 | 0x8600 |
| xx.0184.08007 | 0x54cf | 21711 | Ck10: Profile data implausible | WarningLocked | C00595/10 | 0x8600 |
| xx.0184.08009 | 0x54d1 | 21713 | Ck11: Operating mode invalid | Warning | C00595/11 | 0x8600 |
| xx.0184.08014 | 0x54d6 | 21718 | Ck12: Profile no. invalid | WarningLocked | C00595/12 | 0x8600 |
| xx.0184.08015 | 0x54d7 | 21719 | Ck13: Error FB MCKCtrlInterface | Warning | C00595/13 | 0x8600 |
| xx.0184.00015 | 0x540f | 21519 | Ck14: Target position beyond SW limit position | WarningLocked | C00595/14 | 0x8600 |
| xx.0184.00005 | 0x5405 | 21509 | Ck15: Error status sign. brake | TroubleQuickStop | - | 0x8600 |
| xx.0184.00064 | 0x5440 | 21568 | Ck16: Time overflow manual control | Fault | C00595/15 | 0x8600 |

| Error number 32 bits | 16 bits$_{hex}$ | 16 bits$_{dec}$ | Error message | Response (Lenze setting) | can be set in | CAN Emergency Error Code |
|---|---|---|---|---|---|---|
| xx.0140.00013 | 0x280d | 10253 | CI01: Module missing/incompatible | No Reaction | C01501/2 | 0x7000 |
| xx.0135.00004 | 0x2304 | 8964 | CP04: CAN RPDO4 | No Reaction | C00593/4 | 0x8100 |
| xx.0145.00035 | 0x2d23 | 11555 | dF10: AutoTrip Reset | Fault | C00189 | 0xF000 |
| xx.0145.00014 | 0x2d0e | 11534 | dF14: SW/HW invalid | Fault | - | |
| xx.0145.00015 | 0x2d0f | 11535 | dF15: DCCOM CU2 error | Fault | - | |
| xx.0145.00024 | 0x2d18 | 11544 | dF18: BU RCOM error | Fault | - | 0x6100 |
| xx.0145.00033 | 0x2d21 | 11553 | dF21: BU Watchdog | Fault | - | 0x6100 |
| xx.0145.00034 | 0x2d22 | 11554 | dF22: CU watchdog | Fault | - | 0x6100 |
| xx.0145.00025 | 0x2d19 | 11545 | dF25: CU RCOM error | Fault | - | 0x6100 |
| xx.0145.00026 | 0x2d1a | 11546 | dF26: Appl. watchdog | No Reaction | C00580/1 | 0x6200 |
| xx.0145.00050 | 0x2d32 | 11570 | dF50: Retain error | Fault | - | 0x6100 |
| xx.0145.00051 | 0x2d33 | 11571 | dF51: CuCcr error | Fault | - | 0x6100 |
| xx.0145.00052 | 0x2d34 | 11572 | dF52: BuCcr error | Fault | - | 0x6100 |
| xx.0400.00009 | 0x1a09 | 6665 | dH09: EEPROM power section | Fault | - | 0x5530 |
| xx.0400.00016 | 0x1a10 | 6672 | dH10: Fan failure | Warning | C00566 | 0x5000 |
| xx.0400.00104 | 0x1a68 | 6760 | dH68: Adjustment data error CU | Fault | - | 0x5530 |
| xx.0400.00105 | 0x1a69 | 6761 | dH69: Adjustment data error BU | Fault | - | 0x5530 |
| xx.0123.00094 | 0x175e | 5982 | FC01: Switching frequency reduction | No Reaction | C00590 | 0x2000 |
| xx.0123.00095 | 0x175f | 5983 | FC02: Maximum speed for Fchop | No Reaction | C00588 | 0xF000 |
| xx.0123.00099 | 0x1763 | 5987 | FC03: Field controller limitation | No Reaction | C00570/4 | 0xF000 |
| xx.0123.00057 | 0x1739 | 5945 | Id1: Motor data identification error | WarningLocked | - | 0xF000 |
| xx.0123.00056 | 0x1738 | 5944 | Id2: Motor data identification error | Fault | - | 0xF000 |
| xx.0123.00058 | 0x173a | 5946 | Id3: CINH motor data identification | WarningLocked | - | 0xF000 |
| xx.0123.00059 | 0x173b | 5947 | Id4: Resistance identification error | Warning | - | 0xF000 |
| xx.0123.00074 | 0x174a | 5962 | Id5: Pole position identification error | Fault | C00643/1 | |
| xx.0123.00075 | 0x174b | 5963 | Id6: Resolver ident error | Fault | - | |
| xx.0123.00145 | 0x1791 | 6033 | LP1: Motor phase failure | No Reaction | C00597 | 0x3000 |
| xx.0123.00015 | 0x170f | 5903 | LU: DC bus undervoltage | Trouble | C00600/1 | 0x3100 |
| xx.0123.00016 | 0x1710 | 5904 | oC1: Power section - short circuit | Fault | - | 0x2000 |
| xx.0123.00017 | 0x1711 | 5905 | oC2: Power section - earth fault | Fault | - | 0x2000 |
| xx.0119.00050 | 0x1332 | 4914 | oC5: Ixt overload | Warning | C00604 | 0x2000 |
| xx.0123.00105 | 0x1769 | 5993 | oC6: I2xt motor overload | Warning | C00606 | 0x2000 |
| xx.0123.00007 | 0x1707 | 5895 | oC7: Motor overcurrent | Fault | - | 0x2000 |
| xx.0123.00030 | 0x171e | 5918 | oC10: Maximum current reached | No Reaction | C00609 | 0x2000 |
| xx.0123.00071 | 0x1747 | 5959 | oC11: Clamp operation active | Fault | - | 0xF000 |
| xx.0123.00065 | 0x1741 | 5953 | oC12: I2xt brake resistor overload | No Reaction | C00574 | 0xF000 |
| xx.0123.00090 | 0x175a | 5978 | oC13: Maximum current for Fch exceeded | Fault | - | 0xF000 |
| xx.0123.00096 | 0x1760 | 5984 | oC14: Direct-axis current controller limitation | No Reaction | C00570/1 | 0xF000 |
| xx.0123.00097 | 0x1761 | 5985 | oC15: Cross current controller limitation | No Reaction | C00570/2 | 0xF000 |
| xx.0123.00098 | 0x1762 | 5986 | oC16: Torque controller limitation | No Reaction | C00570/3 | 0xF000 |
| xx.0123.00031 | 0x171f | 5919 | oC17: Clamp sets pulse inhibit | No Reaction | C00569/1 | 0xF000 |
| xx.0119.00001 | 0x1301 | 4865 | oH1: Heatsink overtemperature | Fault | - | 0x4000 |
| xx.0119.00015 | 0x130f | 4879 | oH3: Motor temperature (X106) triggered | Fault | C00585 | 0x4000 |
| xx.0119.00000 | 0x1300 | 4864 | oH4: Heatsink temp. > shutdown temp. -5°C | No Reaction | C00582 | 0x4000 |
| xx.0119.00020 | 0x1314 | 4884 | oH6: Motor temperature MultiEncoder > C121 | Warning | C00583/4 | |
| xx.0119.00002 | 0x1302 | 4866 | oH7: Motor temperature resolver > C121 | Warning | C00583/3 | |
| xx.0119.00003 | 0x1303 | 4867 | oH9: Motor overtemperature resolver | Fault | C00583/1 | |
| xx.0119.00021 | 0x1315 | 4885 | oH12: Motor overtemperature MultiEncoder | Fault | C00583/2 | |
| xx.0123.00032 | 0x1720 | 5920 | oS1: Maximum speed limit reached | No Reaction | C00579 | 0x8400 |
| xx.0123.00033 | 0x1721 | 5921 | oS2: Max. motor speed | Fault | - | 0x8400 |
| xx.0123.00001 | 0x1701 | 5889 | ot1: Max. torque reached | No Reaction | C00608 | 0x8300 |

| Error number 32 bits | 16 bits$_{hex}$ | 16 bits$_{dec}$ | Error message | Response (Lenze setting) | can be set in | CAN Emergency Error Code |
|---|---|---|---|---|---|---|
| xx.0123.00093 | 0x175d | 5981 | ot2: Speed controller output limited | No Reaction | C00567 | 0xF000 |
| xx.0123.00014 | 0x170e | 5902 | OU: DC bus overvoltage | Trouble | - | 0x3100 |
| xx.0144.00001 | 0x2c01 | 11265 | PS01: No memory module | Warning | - | 0x6300 |
| xx.0144.00002 | 0x2c02 | 11266 | PS02: Invalid par. set | Fault | - | 0x6300 |
| xx.0144.00003 | 0x2c03 | 11267 | PS03: Invalid device par. set | Fault | - | 0x6300 |
| xx.0144.00004 | 0x2c04 | 11268 | PS04: Invalid MCI par. set | Fault | - | 0x6300 |
| xx.0144.00007 | 0x2c07 | 11271 | PS07: Par. mem. module invalid | Fault | - | 0x6300 |
| xx.0144.00008 | 0x2c08 | 11272 | PS08: Par. device invalid | Fault | - | 0x6300 |
| xx.0144.00009 | 0x2c09 | 11273 | PS09: Par. format invalid | Fault | - | 0x6300 |
| xx.0144.00010 | 0x2c0a | 11274 | PS10: Memory module binding invalid | Fault | - | |
| xx.0123.00024 | 0x1718 | 5912 | Sd2: Resolver open circuit | Fault | C00603/2 | |
| xx.0123.00205 | 0x17cd | 6093 | Sd3: Feedback system open circuit | Fault | C00586 | 0x7300 |
| xx.0123.00027 | 0x171b | 5915 | Sd4: MultiEncoder open circuit | Fault | C00603/1 | |
| xx.0119.00012 | 0x130c | 4876 | Sd6: Error thermal detector resolver | Fault | C00583/5 | |
| xx.0123.00026 | 0x171a | 5914 | Sd7: Error encoder communication | Fault | C00603/4 | |
| xx.0123.00062 | 0x173e | 5950 | Sd8: Encoder angular drift monit. | Fault | C00603/3 | |
| xx.0123.00200 | 0x17c8 | 6088 | Sd10: Speed limit for feedback system 12 | Fault | C00607 | 0x7300 |
| xx.0123.00201 | 0x17c9 | 6089 | Sd11: Speed limit for feedback system 67 | Fault | C00607 | 0x7300 |
| xx.0119.00022 | 0x1316 | 4886 | Sd12: Error thermal detector MultiEncoder | Fault | C00583/6 | |
| xx.0111.00002 | 0x0b02 | 2818 | Su02: One mains phase is missing | Warning | C00565 | 0x3000 |
| xx.0111.00003 | 0x0b03 | 2819 | Su03: Too frequent mains switching | Fault | - | 0x3000 |
| xx.0111.00004 | 0x0b04 | 2820 | Su04: CU Insufficiently Supplied | Fault | - | 0x3000 |
| xx.0111.00006 | 0x0b06 | 2822 | Su06: Mains input overload | Fault | - | 0x3000 |
| xx.0980.00001 | - | - | US01: User error 1 | No Reaction | C00581/1 | 0x6200 |
| xx.0981.00002 | - | - | US02: User error 2 | No Reaction | C00581/2 | 0x6200 |
| xx.0982.00003 | - | - | US03: User error 3 | No Reaction | C00581/3 | 0x6200 |
| xx.0983.00004 | - | - | US04: User error 4 | No Reaction | C00581/4 | 0x6200 |
| xx.0984.00001 | - | - | US05: User error 5 | No Reaction | C00581/5 | 0x6200 |
| xx.0985.00002 | - | - | US06: User error 6 | No Reaction | C00581/6 | 0x6200 |
| xx.0986.00003 | - | - | US07: User error 7 | No Reaction | C00581/7 | 0x6200 |
| xx.0987.00004 | - | - | US08: User error 8 | No Reaction | C00581/8 | 0x6200 |

### 10.9.6 Cause & possible remedies

This chapter contains all error messages of the controller operating system in numerical order of the error numbers. The list provides detailed information on the response to the error message as well as information on the cause & possible remedies.

**Note!**

For the sake of legibility, the Logbookand C00165 display the error number with the following syntax:
[**Error type**].[**Error subject area no.**].[**Error ID**]

In this documentation, "xx", a wildcard, stands for the error type since it is configurable for many error messages.

**Tip!**

A list of all error messages of the controller operating system in alphabetical order can be found in the previous chapter "Short overview (A-Z)" (📖 608).

**Su02: One mains phase is missing [xx.0111.00002]**

| **Response** (Lenze setting printed in bold) | **Setting:** C00565 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information | |
| **Cause** | **Remedy** |
| One mains phase of a three-phase supply has failed. | Check mains connection (terminal X100). |

**Su03: Too frequent mains switching [xx.0111.00003]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Too frequent mains switching of the power section.<br>• The device recognises if the power section is switched on and off too frequently.<br>• To protect internal charging connections from destruction, the device reports this error and prevents the controller inhibit. All other functions are active. | The error must be acknowledged by mains switching.<br>The charging circuit can only cool down when the mains is switched off.<br>• After switching the mains 3 times in one minute, there must be a switching pause of 9 minutes.<br>• Cyclic mains switching every 3 minutes is permissible. |

**Su04: CU insufficiently supplied [xx.0111.00004]**

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| After switching on the device, the 24V supply voltage for the control electronics is too low (100ms after switch-on U is < 19V).<br>• The current supply voltage is displayed in C00065. | The error must be acknowledged.<br>With internal supply voltage via the power electronics, the controller must be replaced.<br>With external supply voltage, check the correct connection and/or the stability of the supply voltage. |

### Su06: Mains input overload [xx.0111.00006]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Overtemperature in case of devices from type 6 in the input rectifier or the mains choke. | • Check whether all mains phases are connected (a 2-phase supply may be existent).<br>• Provide for sufficient cooling of the device. |

### oH4: Heatsink temp. > shutdown temp. -5°C [xx.0119.00000]

| **Response** (Lenze setting printed in bold) | **Setting:** C00582 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The heatsink temperature now only differs by 5 °C from the shutdown temperature of the motor. | Prevent further heating, i.e. reduce motor load or set controller inhibit so that the heatsink can cool down again. |

### oH1: Overtemperature heatsink [xx.0119.00001]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The heatsink temperature is higher than the fixed limit temperature (90 ° C).<br>Maybe the ambient temperature of the controller is too high or the fan or its ventilation slots are dirty. | • Check control cabinet temperature.<br>• Clean filter.<br>• Clean controller.<br>• If required, clean or replace the fan.<br>• Provide for sufficient cooling of the device. |

### oH7: Motor temperature resolver > C121 [xx.0119.00002]

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/3 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature detected via the resolver interface has exceeded the warning threshold set in C00121/1. | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

### oH9: Motor overtemperature resolver [xx.0119.00003]

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature detected via the resolver interface is higher than the fixed limit temperature (90 °C). | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

### Sd6: Error thermal detector resolver [xx.0119.00012]

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/5 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The signals of the encoder connected to the resolver interface are outside the defined operating range of the motor temperature detection. | • Check the contacting of the encoder cable at the motor and the controller.<br>• Check the selected motor thermal sensor in C01190/1 whether it complies with the assembly in the motor.<br>• If required, switch off thermal sensor monitoring (C00583/5="0"). |

### oH3: Motor temperature (X106) triggered [xx.0119.00015]

| **Response** (Lenze setting printed in bold) | **Setting:** C00585 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature monitoring function at the plug connector X106, terminal T1 /T2, has tripped.<br>Possible causes:<br>• The motor is overheated so that the thermal contact integrated into the motor has been switched.<br>• An open circuit or a loose contact at the connections mentioned above has occurred. | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

### oH6: Motor temperature MultiEncoder > C121 [xx.0119.00020]

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/4 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature detected via the encoder interface has exceeded the warning threshold set in C00121/2. | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

### oH12: Motor overtemperature MultiEncoder [xx.0119.00021]

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The motor temperature detected via the encoder interface is higher than the fixed limit temperature (90 °C). | • Check motor temperature monitoring.<br>• Provide for sufficient cooling of the motor.<br>• Check terminals for open circuit or loose contact. |

### Sd12: Error thermal detector MultiEncoder [xx.0119.00022]

| **Response** (Lenze setting printed in bold) | **Setting:** C00583/6 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| The signals of the encoder connected to the encoder interface are outside the defined operating range of the motor temperature detection. | • Check the contacting of the encoder cable at the motor and the controller.<br>• Check the selected motor thermal sensor in C01190/2 whether it complies with the assembly in the motor.<br>• If required, switch off thermal sensor monitoring (C00583/6="0"). |

### oC5: Ixt overload [xx.0119.00050]

| **Response** (Lenze setting printed in bold) | **Setting:** C00604 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| The Ixt overload check has tripped.<br>• Operating threshold = 100 % Ixt (adjustable in C00123)<br>Possible causes:<br>• Wrong dimensioning of the device with regard to its motor load.<br>• Load cycles are not complied with. | • Check and, if required, correct dimensioning of the device and the motor load with regard to technical data.<br>• Reduce motor load cycles (observe load cycles according to documentation). |

### ot1: Maximum torque reached [xx.0123.00001]

| **Response** (Lenze setting printed in bold) | **Setting:** C00608 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| The device indicates that the maximally possible torque at the motor shaft has been reached.<br>• C00057 displays the current torque. | Reduce motor load. |

### oC7: Motor overcurrent [xx.0123.00007]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| **Cause** | **Remedy** |
|---|---|
| The maximum current monitoring has tripped.<br>• The instantaneous value of the motor current has exceeded the limit value set in C00939. | Check and, if required, correct dimensioning of the load with regard to the installed device power. |

### oU: DC bus overvoltage [xx.0123.00014]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☑ 1: Fault ☒ **2: Trouble** ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| **Cause** | **Remedy** |
|---|---|
| The device has detected an overvoltage in the DC bus. To protect the device hardware, the inverter control is switched off.<br>• Depending on the configuration of the auto-start lock function, C00142 serves to set that, if this error has been tripped, the controller only starts after the controller inhibit is switched.<br>• If this error message remains active longer than the time set in C00601, a "Fault" is tripped. | • Reduce load in generator mode.<br>• Use a brake resistor.<br>• Use a regenerative power supply unit.<br>• Establish a DC-bus connection. |

### LU: DC bus undervoltage [xx.0123.00015]

| Response (Lenze setting printed in bold) | Setting: C00600/1 (☑ Adjustable response) |
|---|---|
| ☐ 0: No Reaction ☑ 1: Fault ☒ **2: Trouble** ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The device has detected a DC bus undervoltage. The inverter control is switched off because the drive properties of the motor control cannot be provided anymore due to the DC bus undervoltage.<br>• Depending on the configuration of the auto-start lock function, C00142 serves to set that, if this error has been tripped, the controller only starts after the controller inhibit is switched. | • Switch on mains supply or ensure sufficient supply via DC bus.<br>• Adjust setting in C00142 if required. |

### oC1: Power section - short circuit [xx.0123.00016]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has recognised a short circuit of the motor phases. To protect the device electronics, the inverter control is switched off.<br>• Mostly, incorrectly executed motor connections are the cause.<br>• If the device is inappropriately dimensioned with regard to the motor load and the current limitation in the controller (Imax controller) is set incorrectly, this error message may also occur.<br>▶ Motor control: Defining current limits | • Check motor connections and the corresponding plug connector on the device.<br>• Only use permissible combinations of device power and motor power.<br>• Do not set the dynamics of the current limitation controller too high. |

### oC2: Power section - earth fault [xx.0123.00017]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has recognised an earth fault at one of the motor phases. To protect the device electronics, the inverter control is switched off.<br>• Mostly, incorrectly executed motor connections are the cause.<br>• If motor filter, motor cable length, and cable type (shielding capacity) are dimensioned incorrectly, this error message may occur due to leakage currents to PE.<br>• If motor filters with additional terminals for +UG and –UG and devices greater or equal 3 kW are used, the earth fault detection may be triggered due to leakage currents to +UG and –UG.<br>• A cause can also be the use of shielded motor cables longer than 50 m. | • Check motor connections and the corresponding plug connector on the device.<br>• Use motor filters, cable lengths, and cable types recommended by Lenze.<br>• If motor filters with additional terminals for +UG and –UG and devices greater or equal 3 kW are used:<br>– Deactivate earth fault detection during operation by setting the filter time (C01770) to 250 ms.<br>• If motor cables longer than 50 m are used:<br>– Increase filter time for earth fault detection during operation (C01770). |

### Sd2: Open circuit resolver [xx.0123.00024]

| Response (Lenze setting printed in bold) | Setting: C00603/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| • Encoder signal interferences (EMC).<br>• Resolver cable interrupted.<br>• Resolver defective. | • Check resolver cable.<br>• Check resolver.<br>• Switch off monitoring (C00603/2 = "0: No reaction") if the resolver is not used. |

### Sd7: Encoder communication error [xx.0123.00026]

| Response (Lenze setting printed in bold) | Setting: C00603/4 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| • Encoder cable interrupted.<br>• Encoder defective. | • Check encoder cable<br>• Check encoder.<br>• Check parameter setting. |

### Sd4: Open circuit MultiEncoder [xx.0123.00027]

| Response (Lenze setting printed in bold) | Setting: C00603/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| • Encoder cable interrupted.<br>• Encoder defective. | • Check encoder cable.<br>• Check encoder.<br>• Switch off monitoring (C00603/1 = "0: No reaction") is the encoder is not used. |

### oC10: Maximum current reached [xx.0123.00030]

| Response (Lenze setting printed in bold) | Setting: C00609 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The device displays that the maximum current has been reached. | • Check and, if required, correct dimensioning of the load with regard to the installed device power.<br>• Check the maximum current settings in C00022 (Imax in motor mode) and C00023 (Imax in generator mode). |

### oC17: Clamp sets pulse inhibit [xx.0123.00031]

| Response (Lenze setting printed in bold) | Setting: C00569/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Due to a short overcurrent, the inverter was switched off for a short time (clamp disconnection). | • Check and, if required, correct dimensioning of the load with regard to the installed device power.<br>• Reduce the dynamics of the setpoint change or speed control. |

### oS1: Maximum speed limit reached [xx.0123.00032]

| Response (Lenze setting printed in bold) | Setting: C00579 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| The device has recognised that the maximum speed has been reached. | • Limit setpoint selection to maximum values.<br>• Adjust set speed limitation (C00909) and frequency limitation (C00910) if necessary. |

### oS2: Max. motor speed [xx.0123.00033]

| Response (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| The device has recognised that the maximally permissible motor speed has been reached. | • Limit setpoint selection to the maximally permissible motor speed.<br>• If required, adapt set maximum motor speed (C00965). |

### Id2: Motor data identification error [xx.0123.00056]

| Response (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| During the identification of the motor parameters, an error has occurred that caused an abortion of the identification process.<br>Possible causes:<br>• Interrupted motor cable.<br>• Switched-off power section during the identification.<br>• Implausible start parameter settings. | • Check the motor connections and the corresponding plug connector on the device and, if necessary, the motor terminal box.<br>• Correct start parameters for the motor parameter identification (motor nameplate data).<br>• Stable power supply of the device. |

### Id1: Motor data identification error [xx.0123.00057]

| Response (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☒ **4: WarningLocked** ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| During the identification of the motor parameters, an error has occurred.<br>Possible causes:<br>• Interrupted motor cable.<br>• Switched-off power section during the identification.<br>• Implausible start parameter settings. | • Check the motor connections and the corresponding plug connector on the device and, if necessary, the motor terminal box.<br>• Correct start parameters for the motor parameter identification (motor nameplate data).<br>• Stable power supply of the device. |

### Id3: CINH motor data identification [xx.0123.00058]

| Response (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☒ **4: WarningLocked** ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| The device has detected controller inhibit during the motor data identification.<br>• This cancels the identification process. The Lenze setting of the motor data is used. | • Do not set controller inhibit during the motor data identification.<br>• Do not execute any device function which may activate controller inhibit. |

### Id4: Resistor identification error [xx.0123.00059]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information | |
| **Cause** | **Remedy** |
| The device has recognised that an error has occurred in the calculation of the motor cable resistance.<br>• The parameters for cable cross-section and cable length are implausible. | Enter sensible values for cable cross-section and motor cable length. |

### Sd8: Encoder angular drift monit. [xx.0123.00062]

| **Response** (Lenze setting printed in bold) | **Setting:** C00603/3 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| • Encoder signal interferences (EMC).<br>• Encoder cable interrupted.<br>• Encoder is defective.<br>• Faulty parameter setting of the encoder. | • Check encoder cable, use shorter encoder cable if required.<br>• Check encoder.<br>• Check parameter setting.<br>• If required, check monitoring (C00603/3). |

### oC12: I2xt overload - brake resistor [xx.0123.00065]

| **Response** (Lenze setting printed in bold) | **Setting:** C00574 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Too frequent and too long braking processes. | Check drive dimensioning. |

### oC11: Clamp operation active [xx.0123.00071]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| The device indicates that the "CLAMP" overcurrent limitation has been activated.<br>• A permanent clamp operation causes an overload disconnection. | Reduce setpoint generation dynamics or motor load. |

### Id5: Pole position identification error [xx.0123.00074]

| **Response** (Lenze setting printed in bold) | **Setting:** C00643/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| With 360° pole position identification:<br>• The rotor position detected via the encoder system does not comply with the controlled output position (plausibility check).<br>With pole position identification with minimum motion:<br>• The encoder system detected a motion higher than the permitted one set in C00645/1. | With 360° pole position identification:<br>• Check parameter setting of the pole position identification.<br>• If required, adapt error tolerance for plausibility check in C00645/2.<br>With pole position identification with minimum motion:<br>• If required, adjust permitted motion in C00645/1. |

### Id6: Resolver ident. error [xx.0123.00075]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| • A controller inhibit was set during resolver error identification.<br>• A time-out occurred while the algorithm was being processed. This error may occur if more than 60 seconds pass by between the setting of the "Resolver error identification" device command and the enable of the controller.<br>• The setpoint speed was too small (\|nset\| < 500 rpm).<br>• The setpoint speed was not traversed for at least 1 second. | • Reduce the gain of the speed controller.<br>• Check the shielding of the motor and encoder.<br>• Check encoder/encoder connection.<br>• Controller is enabled after the "Resolver error identification" device command has been set.<br>• Ensure that the speed profile is traversed for at least 1 second at constant setpoint speed (\|nset\| > 500 rpm). |

### oC13: Maximum current for Fch exceeded [xx.0123.00090]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The device has detected a motor current which exceeds the maximum current limit at permanent switching frequency of the inverter.<br>• If a permanent switching frequency inverter is set, a certain limit arises for the maximum current, depending on the setting. If this current limit is exceeded due to a load impulse or overload, an error message is displayed. | • Observe the maximum current setting depending on the set switching frequency of the inverter.<br>• Reduce the required load or setting of the dynamic switching frequency if necessary. |

### ot2: Speed controller output limited [xx.0123.00093]

| Response (Lenze setting printed in bold) | Setting: C00567 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| The output of the speed controller has reached the internal limit value. In this state, the speed controller is not able anymore to correct the system deviation.<br>• Only with "Closed loop" operation or vector control (SLVC). | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>▶ Motor control |

### FC01: Switching frequency reduction [xx.0123.00094]

| Response (Lenze setting printed in bold) | Setting: C00590 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Load-dependent switching frequency reduction | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>▶ Motor control |

#### FC02: Maximum speed for Fchop [xx.0123.00095]

| **Response** (Lenze setting printed in bold) | **Setting:** C00588 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| Maximum speed for chopper frequency has been reached.<br>• The maximum speed has been exceeded depending on the switching frequency. | Select the correct maximum speed as a function of the switching frequency.<br>▸ Motor control: Defining speed limits |

#### oC14: Direct-axis current controller limitation [xx.0123.00096]

| **Response** (Lenze setting printed in bold) | **Setting:** C00570/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| Direct-axis current controller limitation is active. | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>▸ Motor control |

#### oC15: Cross current controller limitation [xx.0123.00097]

| **Response** (Lenze setting printed in bold) | **Setting:** C00570/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| Cross current controller limitation is active. | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>• Check parameter setting of the current controller with regard to the motor controllers (e.g. reduce Vp).<br>▸ Motor control |

#### oC16: Torque controller limitation [xx.0123.00098]

| **Response** (Lenze setting printed in bold) | **Setting:** C00570/3 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| Actuator limitation according to speed controller. | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint generation dynamics if necessary.<br>▸ Motor control |

#### FC03: Field controller limitation [xx.0123.00099]

| **Response** (Lenze setting printed in bold) | **Setting:** C00570/4 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| The output of the field controller has reached its maximum limit value. The drive is at the torque limit in the field weakening range. | • Observe load requirements.<br>• Correct dimensioning or reduce setpoint from the field weakening range if necessary.<br>▸ Motor control |

### oC6: I2xt overload - motor [xx.0123.00105]

| Response (Lenze setting printed in bold) | Setting: C00606 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Thermal overload of the motor. | • Observe load requirements.<br>• Correct dimensioning if necessary.<br>• In case of VFCplus operation: Check Vmin boost (C00016).<br>Set ▶ Vmin boost |

### LP1: Motor phase failure [xx.0123.00145]

| Response (Lenze setting printed in bold) | Setting: C00597 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Motor phase failure - power section<br>• This error message is displayed if a motor phase carries less current of one half-wave than set in C00599. | • Check the motor connections and the corresponding plug connector on the device and, if necessary, the motor terminal box.<br>• Check the trigger threshold C00599. |

### Sd10: Speed limit - feedback system 12 [xx.0123.00200]

| Response (Lenze setting printed in bold) | Setting: C00607 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Maximally permissible speed of the feedback system connected to DI1/DI2 reached. | Reduce speed of the rotation shaft/feedback system.<br>$n_{encoder}$ <= ($f_{max}$ x 60) / encoder increment<br>(for $f_{max}$ = 10 kHz) |

### Sd11: Speed limit for feedback system 67 [xx.0123.00201]

| Response (Lenze setting printed in bold) | Setting: C00607 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Maximally permissible speed of the feedback system connected to DI6/DI7 reached. | Reduce speed of the rotation shaft/feedback system.<br>$n_{encoder}$ <= ($f_{max}$ x 60) / encoder increment<br>(for $f_{max}$ = 10 kHz) |

### Sd3: Open circuit - feedback system [xx.0123.00205]

| Response (Lenze setting printed in bold) | Setting: C00586 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| • HTL encoder cable interrupted.<br>• HTL encoder is defective.<br>Note: May also be caused by a very dynamic acceleration or starting up against a blocked motor shaft (e.g. with a closed holding brake). | • Check HTL encoder cable.<br>• Check HTL encoder.<br>• Check related terminals.<br>• Switch off monitoring (C00603/ = "0: No reaction") is the HTL encoder is not used. |

#### An01: AIN1_I < 4 mA [xx.0125.00001]

| **Response** (Lenze setting printed in bold) | **Setting:** C00598/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☑ 2: Trouble ☒ **3: TroubleQuickStop** ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Open-circuit monitoring for analog input 1 has tripped.<br>• Only if the analog input has been configured as a current loop of 4 ... 20 (C00034/1 = 2). | • Check wiring of the analog X3/A1I input terminal for open circuit.<br>• Check minimum current values of the signal sources. |

#### An02: AIN2_I < 4 mA [xx.0125.00002]

| **Response** (Lenze setting printed in bold) | **Setting:** C00598/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☑ 2: Trouble ☒ **3: TroubleQuickStop** ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Open-circuit monitoring for analog input 2 has tripped.<br>• Only if the analog input has been configured as a current loop of 4 ... 20 (C00034/2 = 2). | • Check wiring of the analog X3/A2I input terminal for open circuit.<br>• Check minimum current values of the signal sources. |

#### CE04: MCI communication error [xx.0127.00002]

| **Response** (Lenze setting printed in bold) | **Setting:** C01501/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| Communication error with extension module in slot 1. | • Eliminate EMC interference.<br>• Switch off controller, correctly plug in the module, switch on the controller again.<br>• Mains switching or restart of the controller, respectively.<br>• Replace module/controller.<br>• If the problem occurs again, you needs to consult Lenze. |

#### CE0F: MCI control word [xx.0127.00015]

| **Response** (Lenze setting printed in bold) | **Setting:** C00594/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Bit 14 ("SetFail") of the wMciCtrl control word of the LS_DriveInterface system block has been set. | Trace back signal source on the bus (e.g. PROFIBUS) that sets bit 14 ("SetFail"). |

#### CE4: CAN bus off [xx.0131.00000]

| **Response** (Lenze setting printed in bold) | **Setting:** C00592/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: "Bus off" status<br>• Received too many faulty telegrams.<br>• Damaged cable (e.g. loose contact).<br>• Two nodes have the same ID. | • Check wiring and bus terminating resistor.<br>• Set identical baud rate for each bus node.<br>• Assign different IDs to nodes.<br>• Eliminate electrical interference (e.g. EMC). |

#### CA06: CAN CRC error [xx.0131.00006]

| **Response** (Lenze setting printed in bold) | **Setting:** C00592/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: A faulty CAN telegram has been detected. | • Check wiring and bus terminating resistor.<br>• Eliminate electrical interference (e.g. EMC). |

#### CA07: CAN bus warning [xx.0131.00007]

| **Response** (Lenze setting printed in bold) | **Setting:** C00592/3 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: Incorrect transmission or reception of more than 96 CAN telegrams.<br>• The current number of incorrectly transmitted CAN telegrams is displayed in C00372/1.<br>• The current number of incorrectly received CAN telegrams is displayed in C00372/2.<br>• The current CAN error status is displayed in C00345. | • Check wiring and bus terminating resistor.<br>• Set identical baud rate for each bus node.<br>• Assign different IDs to nodes.<br>• Eliminate electrical interference (e.g. EMC). |

#### CA08: CAN bus stopped [xx.0131.00008]

| **Response** (Lenze setting printed in bold) | **Setting:** C00592/4 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: The device has received the "Stop Remote Node" NMT telegram. | Check CAN master (NMT master). |

#### CA0b: CAN HeartBeatEvent [xx.0131.00011]

| **Response** (Lenze setting printed in bold) | **Setting:** C00592/5 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| CAN on board: Cyclic node monitoring<br>• Being a Heartbeat consumer, the device has not received a Heartbeat telegram from Heartbeat producer 1 ... 15 within the defined time.<br>• The current statuses of the Heartbeat producers are displayed in C00347/1...15. | • Reactivate Heartbeat producers by mains switching, restarting the controller, or a CAN Reset Node.<br>• Reparameterise CAN Heartbeat producer time or switch off consumer monitoring and reset error status if latched.<br>▸ Heartbeat protocol |

#### CA0F: CAN control word [xx.0131.00015]

| **Response** (Lenze setting printed in bold) | **Setting:** C00594/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |

| Cause | Remedy |
|---|---|
| Bit 14 ("SetFail") in the wCANControl control word of the LS_DriveInterface system block has been set. | Trace back signal source on the CAN bus that sets bit 14 ("SetFail"). |

### CE1: CAN RPDO1 [xx.0135.00001]

| **Response** (Lenze setting printed in bold) | **Setting:** C00593/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| CAN on board: Time monitoring for RPDO1 has tripped.<br>• RPDO1 has not been received within the monitoring time set in C00357/1 or was faulty. | • Set the correct telegram length at the CAN master (transmitter).<br>• Eliminate electrical interference (e.g. EMC).<br>• Adjust monitoring time in C00357/1 or switch off time monitoring. |

### CE2: CAN RPDO2 [xx.0135.00002]

| **Response** (Lenze setting printed in bold) | **Setting:** C00593/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| CAN on board: Time monitoring for RPDO2 has tripped.<br>• RPDO2 has not been received within the monitoring time set in C00357/2 or was faulty. | • Set the correct telegram length at the CAN master (transmitter).<br>• Eliminate electrical interference (e.g. EMC).<br>• Adjust monitoring time in C00357/2 or switch off time monitoring. |

### CE3: CAN RPDO3 [xx.0135.00003]

| **Response** (Lenze setting printed in bold) | **Setting:** C00593/3 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| CAN on board: Time monitoring for RPDO3 has tripped.<br>• RPDO3 has not been received within the monitoring time set in C00357/3 or was faulty. | • Set the correct telegram length at the CAN master (transmitter).<br>• Eliminate electrical interference (e.g. EMC).<br>• Adjust monitoring time in C00357/3 or switch off time monitoring. |

### CP04: CAN RPDO4 [xx.0135.00004]

| **Response** (Lenze setting printed in bold) | **Setting:** C00593/4 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| CAN on board: Time monitoring for RPDO4 has tripped.<br>• RPDO4 has not been received within the monitoring time set in C00357/4 or was faulty. | • Set the correct telegram length at the CAN master (transmitter).<br>• Eliminate electrical interference (e.g. EMC).<br>• Adjust monitoring time in C00357/4 or switch off time monitoring. |

### CI01: Module missing/incompatible [xx.0140.00013]

| **Response** (Lenze setting printed in bold) | **Setting:** C01501/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| The optional communication module has been removed or there is a connection problem or incompatibility with the standard device. | • Check connection between the communication module and standard device.<br>• Check if the module is plugged in correctly.<br>• In case of an incompatibility, either the module or the software of the standard device is out of date. In this case, please contact Lenze. |

#### PS01: No memory module [xx.0144.00001]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Memory module is either not available or not snapped into place correctly. | • If a memory module has been provided: Plug the memory module into the slot of the standard device intended for this purpose.<br>• If a memory module has been provided: Check if the memory module has been plugged-in correctly. |

#### PS02: Par. set invalid [xx.0144.00002]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set saved to the memory module is invalid because it has not been saved completely.<br>• This can be due to voltage failure or caused by removing the memory module while saving the parameter set. | Ensure voltage supply during the storage process and that the module remains plugged into the slot. |

#### PS03: Par. set device invalid [xx.0144.00003]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set saved to the memory module is incompatible to the standard device.<br>• An incompatibility of the parameter set is caused e.g. when the memory module of an 8400 HighLine is plugged into an 8400 StateLine or the parameter set in the memory module has a higher version than expected by the standard device. | When the memory modules are exchanged, observe the downward compatibility:<br>• OK: StateLine V2.0 to StateLine V3.0<br>• OK: StateLine V2.0 to HighLine V2.0<br>• Not OK: HighLine Vx.x to StateLine Vx.x<br>• Not OK: StateLine V3.0 to StateLine < V3.0 |

#### PS04: Par. set Mci invalid [xx.0144.00004]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set saved to the communication module is incompatible to the standard device.<br>• An incompatibility of the parameter set is caused e.g. when the MCI module parameters in the memory module do not match the plugged communication module. | When the memory modules are exchanged, observe the downward compatibility:<br>• OK: StateLine V2.0 to StateLine V3.0<br>• OK: StateLine V2.0 to HighLine V2.0<br>• Not OK: HighLine Vx.x to StateLine Vx.x<br>• Not OK: StateLine V3.0 to StateLine < V3.0 |

**PS07: Par. memory module invalid [xx.0144.00007]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set saved to the memory module is invalid.<br>• The error occurs while loading the parameter set.<br>• The memory module plugged in the device lacks a code or a code is incorrect. | Consultation with Lenze required. |

**PS08: Par. device invalid [xx.0144.00008]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The parameter set in the device is invalid.<br>• The error occurs while loading the parameter set.<br>• One code in the device is incorrect. | Consultation with Lenze required. |

**PS09: Par. format invalid [xx.0144.00009]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| The code format is invalid.<br>• The error occurs while loading the parameter set. | Consultation with Lenze required. |

**PS10: Memory module binding invalid [xx.0144.00010]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Active device personalisation: The memory module and the controller do not have identical binding IDs. | • Use memory modules/controllers with matching binding IDs.<br>• Consult the machine manufacturer.<br>**Note:**<br>Lenze cannot modify e.g. a replacement device via special access to make it work with a personalised memory module. |

**dF14: SW-HW invalid [xx.0145.00014]**

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Device error | Consultation with Lenze required. |

### dF15: DCCOM CU2 error [xx.0145.00015]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Consultation with Lenze required. |

### dF18: BU RCOM error [xx.0145.00024]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Consultation with Lenze required. |

### dF25: CU RCOM error [xx.0145.00025]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Mains switching too frequent.<br>• Cyclic mains switching every 3 minutes is permissible. | • After switching the mains 3 times in one minute, there must be a switching pause of 9 minutes.<br>• If the problem occurs again, you needs to consult Lenze. |

### dF26: Appl. watchdog [xx.0145.00026]

| **Response** (Lenze setting printed in bold) | **Setting:** C00580/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☑ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Time-out of the application. The required computing time of the application exceeds the available computing time. | Reduction of the function block interconnection or the complexity of the application. |

### dF21: BU watchdog [xx.0145.00033]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Consultation with Lenze required. |

### dF22: CU watchdog [xx.0145.00034]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Consultation with Lenze required. |

### dF10: AutoTrip reset [xx.0145.00035]

| Response (Lenze setting printed in bold) | Setting: C00189 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| Too frequent auto-trip reset. | • Check the error cause that activates the auto-trip reset.<br>• Eliminate error cause and reset (acknowledge) error manually afterwards. |

### dF50: Retain error [xx.0145.00050]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| An error has occurred when accessing retain data.<br>• Either caused by an internal hardware error or by lack of mains switching after a firmware download. | Mains switching<br>• If the problem occurs again, you needs to consult Lenze. |

### dF51: CuCcr error [xx.0145.00051]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Device error | Mains switching<br>• If the problem occurs again, you needs to consult Lenze. |

### dF52: BuCcr error [xx.0145.00052]

| Response (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| Cause | Remedy |
|---|---|
| Device error | Mains switching<br>• If the problem occurs again, you needs to consult Lenze. |

### Ck01: Pos. HW limit switch [xx.0184.00001]

| Response (Lenze setting printed in bold) | Setting: C00595/1 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |

| Cause | Remedy |
|---|---|
| MCK: The hardware limit switch in positive traversing direction has tripped.<br>• The *bLimitSwitchPos* input for travel range monitoring via positive hardware limit switch has been set to FALSE (fail-safe). | Reset error message and retract limit switch. |

### Ck02: Neg. HW limit switch [xx.0184.00002]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/2 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| MCK: The hardware limit switch in negative traversing direction has tripped.<br>• The *bLimitSwitchNeg* input for travel range monitoring via negative hardware limit switch has been set to FALSE (fail-safe). | Reset error message and retract limit switch. |

### Ck15: Error status sign. brake [xx.0184.00005]

| **Response** (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☐ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| MCK: The status monitoring of the holding brake control has tripped. | • Check configuration of the *bBrkApplied* input for status detection of the brake (via a switching contact at the brake).<br>• Check wiring/function of the switching contact.<br>• Adapt waiting time (C02589/3).<br>• Deactivate status monitoring (via bit 5 in C02582). |

### Ck03: Pos. SW limit position [xx.0184.00007]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/3 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| MCK: The device has detected that the position is beyond the positive software limit position (C01229/1). | • Enlarge permissible traversing range (change settings of the software limit positions).<br>• Deactivate software limit position monitoring. |

### Ck04: Neg. SW limit position [xx.0184.00008]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/4 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| MCK: The device has detected that the position is beyond the negative software limit position (C01229/2). | • Enlarge permissible traversing range (change settings of the software limit positions).<br>• Deactivate software limit position monitoring. |

### Ck14: Target position outside SW limit position [xx.0184.00015]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/14 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| MCK: A target beyond the software limit positions (C01229/1 and C01229/2) has been attempted to position to. | • Select a target within the software limit positions.<br>• Enlarge permissible traversing range (change settings of the software limit positions).<br>• Deactivate software limit position monitoring. |

### Ck16: Time overflow manual operation [xx.0184.00064]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/15 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| Manual PC control: The connection monitoring has tripped.<br>• The online connection between the PC and the controller has been interrupted for a longer period of time than the timeout set in C00464/1. | • Check the communication link between the PC and the controller.<br>• Check the voltage supply/function of the controller.<br>• Adjust the timeout (C00464/1). |

### Ck05: Following error 1 [xx.0184.00153]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/5 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |
| **Cause** | **Remedy** |
| MCK: The following error limit 1 (C01215/1) has been exceeded. | • Optimise control mode.<br>• Increase following error limit.<br>• Deactivate following error monitoring. |

### Ck06: Following error 2 [xx.0184.00154]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/6 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |
| **Cause** | **Remedy** |
| MCK: The following error limit 2 (C01215/2) has been exceeded. | • Optimise control mode.<br>• Increase following error limit.<br>• Deactivate following error monitoring. |

### Ck07: Travel range limit exceeded [xx.0184.00155]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/7 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☒ **3: TroubleQuickStop** ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| MCK: The maximum travel distance has been exceeded.<br>• The maximum travel distance is displayed in C01213/1. | • Check profile parameters.<br>• Deactivate travel range limit monitoring. |

### Ck08: Home position unknown [xx.0184.00156]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/8 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| MCK: Home position is unknown. | Perform homing. |

### Ck09: Positioning mode invalid [xx.0184.08005]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/9 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| MCK: The positioning mode defined via the *wPosProfileMode* input is not supported. | Define a valid positioning mode. |

### Ck10: Implausible profile data [xx.0184.08007]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/10 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| MCK: Implausible profile data are at hand. | Check profile data set. |

### Ck11: Invalid operating mode [xx.0184.08009]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/11 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| MCK: The operating mode defined via the *wMckOperationMode* input is not supported. | Define a valid operating mode. |

### Ck12: Invalid profile number [xx.0184.08014]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/12 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☒ **4: WarningLocked** ☑ 5: Warning ☑ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| MCK: The positioning profile number in the positioning operating mode specified via the *wPosProfileNo* input is invalid. | Define a valid profile number. |

### Ck13: Error - MCKCtrlInterface function block [xx.0184.08015]

| **Response** (Lenze setting printed in bold) | **Setting:** C00595/13 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☒ **5: Warning** ☑ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| MCK: An error has occurred in the L_MckCtrlInterface_1 function block. | Check wiring and parameterisation of the L_MckCtrlInterface_1 function block. |

### dH09: EEPROM power section [xx.0400.00009]

| **Response** (Lenze setting printed in bold) |
|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information |

| **Cause** | **Remedy** |
|---|---|
| Device error | Consultation with Lenze required. |

### dH10: Fan failure [xx.0400.00016]

| **Response** (Lenze setting printed in bold) | **Setting:** C00566 (☑ Adjustable response) |
|---|---|
| ☑ 0: No Reaction ☑ 1: Fault ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information | |

| **Cause** | **Remedy** |
|---|---|
| The device fan has failed.<br>Possible causes:<br>• The short-circuit check of the fan connection has tripped.<br>• The speed monitoring of the fan has tripped. | • Check the fan for short-circuit.<br>• Clean the fan. |

### dH68: Adjustment data error CU [xx.0400.00104]

| Response (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Consultation with Lenze required. |

### dH69: Adjustment data error BU [xx.0400.00105]

| Response (Lenze setting printed in bold) | |
|---|---|
| ☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information | |
| **Cause** | **Remedy** |
| Device error | Consultation with Lenze required. |

### US01: User error 1 [xx.0980.00001]

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/1 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| User error 1 has been tripped via the *bSetError1* input of the LS_SetError_1 system block. | User-defined. |

### US02: User error 2 [xx.0981.00002]

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/2 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| User error 2 has been tripped via the *bSetError2* input of the LS_SetError_1 system block. | User-defined. |

### US03: User error 3 [xx.0982.00003]

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/3 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| User error 3 has been tripped via the *bSetError3* input of the LS_SetError_1 system block. | User-defined. |

### US04: User error 4 [xx.0983.00004]

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/4 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| User error 4 has been tripped via the *bSetError4* input of the LS_SetError_1 system block. | User-defined. |

### US05: User error 5 [xx.0984.00001]

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/5 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| User error 5 has been tripped via the *bSetError1* input of the LS_SetError_2 system block. | User-defined. |

### US06: User error 6 [xx.0985.00002]

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/6 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| User error 6 has been tripped via the *bSetError2* input of the LS_SetError_2 system block. | User-defined. |

### US07: User error 7 [xx.0986.00003]

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/7 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| User error 7 has been tripped via the *bSetError3* input of the LS_SetError_2 system block. | User-defined. |

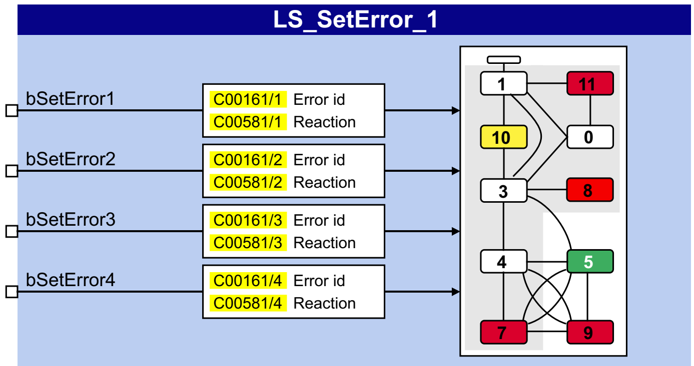
### US08: User error 8 [xx.0987.00004]

| **Response** (Lenze setting printed in bold) | **Setting:** C00581/8 (☑ Adjustable response) |
|---|---|
| ☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information | |
| **Cause** | **Remedy** |
| User error 8 has been tripped via the *bSetError4* input of the LS_SetError_2 system block. | User-defined. |

## 10.10 System block "LS_SetError_1"

This system block is used for error handling within the application.

- The application can trip up to four different user error messages with parameterisable error IDs and error responses via the four boolean inputs.
- If several inputs are set to TRUE at the same time, the input with the lowest number will trip the error message.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bSetError1<br>BOOL | Input for tripping "US01: User error 1"<br>• Error subject number: 980<br>• Error number: (C00581/1 x 0x0400000) + (980 x 0x10000) + (C00161/1) |
| bSetError2<br>BOOL | Input for tripping "US02: User error 2"<br>• Error subject number: 981<br>• Error number: (C00581/2 x 0x0400000) + (981 x 0x10000) + (C00161/2) |
| bSetError3<br>BOOL | Input for tripping "US03: User error 3"<br>• Error subject number: 982<br>• Error number: (C00581/3 x 0x0400000) + (982 x 0x10000) + (C00161/3) |
| bSetError4<br>BOOL | Input for tripping "US04: User error 4"<br>• Error subject number: 983<br>• Error number: (C00581/4 x 0x0400000) + (983 x 0x10000) + (C00161/4) |

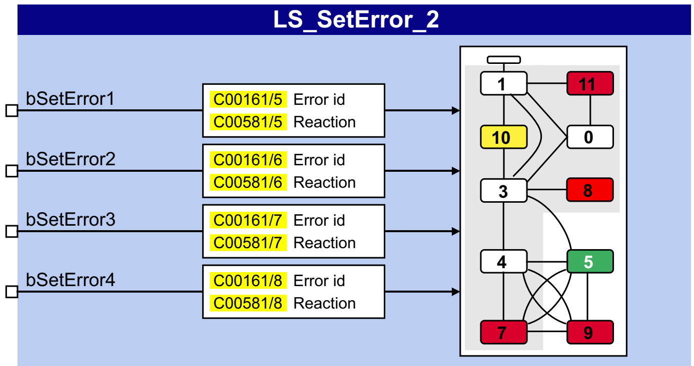
## Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00161/1...4 | 0 | ... | 65535 | Error ID for user errors 1 ... 4 |
| C00581/1...4 | | | | Response to user errors 1 ... 4 |
| | 0 | No Reaction | | |
| | 1 | Fault (pulse inhibit) | | |
| | 2 | Trouble | | |
| | 3 | TroubleQuickStop | | |
| | 4 | WarningLocked | | |
| | 5 | Warning | | |
| | 6 | Information | | |

## 10.11 System block "LS_SetError_2"

This system block is used for error handling within the application.

- The application can trip up to four different user error messages with parameterisable error IDs and error responses via the four boolean inputs.
- If several inputs are set to TRUE at the same time, the input with the lowest number will trip the error message.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bSetError1<br>BOOL | Input for tripping "US05: User error 5"<br>• Error subject number: 984<br>• Error number: (C00581/5 x 0x0400000) + (984 x 0x10000) + (C00161/5) |
| bSetError2<br>BOOL | Input for tripping "US06: User error 6"<br>• Error subject number: 985<br>• Error number: (C00581/6 x 0x0400000) + (985 x 0x10000) + (C00161/6) |
| bSetError3<br>BOOL | Input for tripping "US07: User error 7"<br>• Error subject number: 986<br>• Error number: (C00581/7 x 0x0400000) + (986 x 0x10000) + (C00161/7) |
| bSetError4<br>BOOL | Input for tripping "US08: User error 8"<br>• Error subject number: 987<br>• Error number: (C00581/8 x 0x0400000) + (987 x 0x10000) + (C00161/8) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00161/5...8 | 0 | ... | 65535 | Error ID for user errors 5 ... 8 |
| C00581/5...8 | | | | Response to user errors 5 ... 8 |
| | 0 | No Reaction | | |
| | 1 | Fault (pulse inhibit) | | |
| | 2 | Trouble | | |
| | 3 | TroubleQuickStop | | |
| | 4 | WarningLocked | | |
| | 5 | Warning | | |
| | 6 | Information | | |

# 11 System bus "CAN on board"

The controller has an integrated CANopen system bus interface ("CAN on board") which is used to exchange i.a. process data and parameter values between the nodes. Furthermore, other modules can be connected via this interface such as decentralised terminals, operator and input devices (HMIs), as well as external controls and host systems.

The interface transfers CAN objects following the CANopen communication profile (CiA DS301, version 4.02) developed by the umbrella organisation of CiA (CAN in Automation) in conformity with the CAL (CAN Application Layer).

**Tip!**

- In the »Engineer« parameter list and in the keypad, category **CAN**, you can find the parameters relevant for the CANopen system bus interface classified in different subcategories.
- Information on CAN communication modules and CANopen system bus interfaces of other Lenze devices is provided in the "CAN" communication manual in the Lenze library.

## 11.1 General information

For many years, the system bus (CAN) based on the CANopen communication profile has been integrated in Lenze controllers. Due to the lower number of data objects available, the functionality and compatibility of the previous system bus are lower as compared to CANopen. For parameter setting, two parameter data channels are always available to the user while CANopen provides only one active parameter channel.

The system bus (CANopen) of the Inverter Drives 8400 is a further development of the system bus (CAN) including the following properties:

- Full compatibility according to CANopen DS301, V4.02.
- Support of the "Heartbeat" NMT slave function (DS301, V4.02).
- Number of parameterisable server SDO channels:
  - Max. 2 channels with 1 ... 8 bytes
  - Due to the 2 server SDO channels, an address range of 1 ... 63 is provided.
- Number of parameterisable PDO channels:
  - For device version "BaseLine C":
    max. 2 transmit PDOs (TPDOs) with 1 ... 8 bytes (adjustable)
    max. 2 receive PDOs (RPDOs) with 1 ... 8 bytes (adjustable)
  - From device version "StateLine":
    max. 3 transmit PDOs (TPDOs) with 1 ... 8 bytes (adjustable)
    max. 3 receive PDOs (RPDOs) with 1 ... 8 bytes (adjustable)
- All PDO channels are functionally equivalent.
- Monitoring of the RPDOs for data reception
- Adjustable error response to ...
  - physical CAN errors (frame, bit, ACK error)
  - bus-stop, bus working
  - absent PDOs
- Telegram counters for SDOs and PDOs
- Bus status diagnostics
- Boot-up telegram generation
- Emergency telegram generation
- Reset node telegram generation (in case of master configuration)
- Sync telegram generation and response to sync telegrams:
  - Data transmission/reception
  - Device-internal time base synchronisation
- Abort codes
- All CAN on board functions can be parameterised via codes
- Object directory (all mandatory functions, optional functions, indexes)

### 11.1.1 General data and application conditions

| Range | Values |
|---|---|
| Communication profile | CANopen, DS301 V4.02 |
| Communication medium | DIN ISO 11898 |
| Network topology | Line terminated at both ends |
| Adjustable node addresses<br>(max. number of nodes) | Depending on the number of SDO channels set in C00366:<br>• 1 SDO: Node address 1 ... 127 (max. 127 nodes)<br>• 2 SDO: Node address 1 ... 63 (max. 63 nodes)<br>• adjustable via DIP switches or via code C00350. |
| Adjustable baud rates | 20, 50, 125, 250, 500, 1000 kbps<br>• adjustable via DIP switches or via code C00351. |
| Process data | • Max. 3 transmit PDOs (TPDOs) with 1 ... 8 bytes (adjustable)<br>• Max. 3 receive PDOs (RPDOs) with 1 ... 8 bytes (adjustable) |
| Parameter data | Max. 2 server SDO channels with 1 ... 8 bytes |
| Transfer mode for TPDOs | • in case of data change (including adjustable blocking time)<br>• Time-controlled, 1 to x ms<br>• After the reception of 1 to 240 sync telegrams |

### 11.1.2 Supported protocols

| Protocols | |
|---|---|
| Standard PDO protocols | PDO write<br>PDO read |
| SDO protocols | SDO download<br>SDO download initiate<br>SDO download segment |
| | SDO upload<br>SDO upload initiate<br>SDO upload segment |
| | SDO abort transfer |
| | SDO block download<br>SDO block download initiate<br>SDO block download end |
| | SDO block upload<br>SDO block upload initiate<br>SDO block upload end |
| NMT protocols | Start remote node (master and slave) |
| | Stop remote node (slave) |
| | Enter pre-operational (slave) |
| | Reset node (slave and local device) |
| | Reset communication protocol (slave) |
| Monitoring protocols | Heartbeat (heartbeat producer and heartbeat consumer)<br>• Up to 15 Heartbeat Producers can be monitored. |
| | Emergency telegram (to master) |
| More protocols | Transmitting and receiving a sync telegram<br>• Synchronisation of the internal time base to the reception of the CAN sync telegram is possible. ▶ Synchronisation of the internal time base |

### 11.1.3 Communication time

The communication time is the time between the start of a request and the arrival of the corresponding response.

**Tip!**

The communication times in the CAN network depend on:

- the processing time in the device
- the telegram runtime (baud rate/telegram length)
- the bus load (especially if the bus is charged with PDOs and SDOs at a low baud rate)

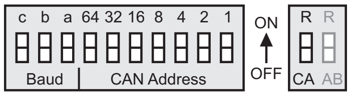
**Processing time in the 8400 controller**

No dependencies exist between parameter data and process data.

- ▶ Parameter data: approx. 5 ms (typical value)
  - – For parameters concerning the motor control (e.g. C00011), the processing time may be longer (up to 30 ms).
- ▶ Process data: 1 ms

## 11.2 Possible settings via DIP switch

The following settings for the "CAN on board" system bus can be made via the front panel DIP switches:

| DIP switch | | Possible settings/detailed information |
|---|---|---|
|   | a ... c | ▸ Setting the baud rate |
| | 1 ... 64 | ▸ Setting the node address |
| | CA | ▸ Activating the bus terminating resistor |

Lenze setting: All DIP switches are in the "OFF" position

**Note!**

- The DIP switch settings are accepted if a node address is unequal zero when the device or the 24-V supply is switched on by the DIP address.
- If all DIP switches are OFF when the device or the 24 V supply is switched on, the setting of the baud rate and node address are read out of the parameter set/parameter.

**Tip!**

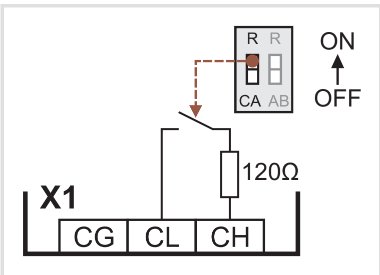
The current DIP switch settings are displayed in code C00349.

Bit 15 indicates that the setting of the DIP switches has been accepted when the device or the 24V supply has been switched on.

### 11.2.1 Activating the bus terminating resistor

The CAN bus must be terminated between CAN low and CAN high at the first and last physical node each by a resistor (120 Ω).

The 8400 controller is provided with an integrated bus terminating resistor, which can be activated via the DIP switch labelled with "CA":




- ▶ OFF = bus terminating resistor is inactive
- ▶ ON = bus terminating resistor is active

[11-1] Activation of the integrated bus terminating resistor

### 11.2.2 Setting the baud rate

The baud rate can be set via code C00351 or with the DIP switches a to c:

**Note!**

- All DIP switches (a ... c, 1 ... 64) = OFF (Lenze setting):
  - At switching on, the settings under code C00350 (node address) and C00351 (baud rate) will become active.
- Preset baud rate: 500 kbps

| DIP switch position | | | Baud rate |
|---|---|---|---|
| c | b | a | |
| **ON** | OFF | **ON** | 20 kbps |
| OFF | **ON** | **ON** | 50 kbps |
| OFF | **ON** | OFF | 125 kbps |
| OFF | OFF | **ON** | 250 kbps |
| OFF | OFF | OFF | 500 kbps |
| **ON** | OFF | OFF | 1000 kbps |

### 11.2.3 Setting the node address

The node address can be set via code C00350 or with the DIP switches 1 to 64.

- The labelling on the housing corresponds to the values of the individual DIP switches for determining the node address.
- The valid address range depends on the number of SDO channels set in C00366:
  - 1 SDO (Lenze setting): 1 ... 127
  - 2 SDO: 1 ... 63

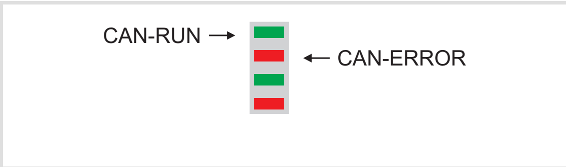
**Note!**

- The addresses of the nodes must differ from each other.
- All DIP switches (a ... c, 1 ... 64) = OFF (Lenze setting):
  - At switching on, the settings under code C00350 (node address) and C00351 (baud rate) will become active.

### Example: Setting of the node address 23

| DIP switch | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| Switch position | OFF | OFF | **ON** | OFF | **ON** | **ON** | **ON** |
| Value | 0 | 0 | 16 | 0 | 4 | 2 | 1 |
| Node address | **= Sum of the values = 16 + 4 + 2 + 1 = 23** | | | | | | |

## 11.3 LED status displays for the system bus




- CAN-RUN:
  Signals the CANopen state
- CAN-ERROR:
  Signals a CANopen error

| Frequency of the display | CAN signalling and meaning |
|---|---|
| Permanently red | CAN-Run: -, CAN-Error: **Bus Off**  |
| Flashes | Automatic detection of baud rate is active  |
| Green is blinking every 0.2 s | CAN-Run: **Pre-Operational**, CAN-Error: -  |
| Green is blinking every 0.2 s<br>Red is blinking once, 1 s off | CAN-Run: **Pre-Operational**, CAN-Error: **Warning Limit reached**  |
| Green is blinking every 0.2 s<br>Red is blinking twice, 1 s off | CAN-Run: **Pre-Operational**, CAN-Error: **Node Guard Event**  |
| Permanently green | CAN-Run: **Operational**, CAN-Error: -  |
| Permanently green<br>Red is blinking once, 1 s off | CAN-Run: **Operational**, Fault: **Warning Limit reached**  |
| Permanently green<br>Red is blinking twice, 1 s off | CAN-Run: **Operational**, CAN-Error: **Node Guard Event**  |
| Permanently green<br>Red is blinking 3 times, 1 s off | CAN-Run: **Operational**, CAN-Error: **Sync Message Error**  |
| Green is blinking every second | CAN-Run: **Stopped**, CAN-Error: -  |
| Green is blinking every second<br>Red is blinking once, 1 s off | CAN-Run: **Stopped**, CAN-Error: **Warning Limit reached**  |
| Green is blinking every second<br>Red is blinking twice, 1 s off | CAN-Run: **Stopped**, CAN-Error: **Node Guard Event**  |

## 11.4 Going online via system bus (CAN on board)

The integrated system bus interface (CAN on board, X1 terminal) can also be used for the communication between the »Engineer« and the controller, alternatively to the USB diagnostic adapter.

▶ Lenze offers the following communication accessories for connection to the PC:

| Communication accessories | PC interface |
|---|---|
| **PC system bus adapter 2173**<br>incl. connection cable and voltage supply adapter<br>• for DIN keyboard connection (EMF2173IB)<br>• for PS/2 keyboard connection (EMF2173IBV002)<br>• for PS/2 keyboard connection with electrical isolation (EMF2173IBV003) | Parallel interface<br>(LPT port) |
| **PC system bus adapter 2177**<br>incl. connection cable (EMF2177IB) | USB<br>(Universal Serial Bus) |

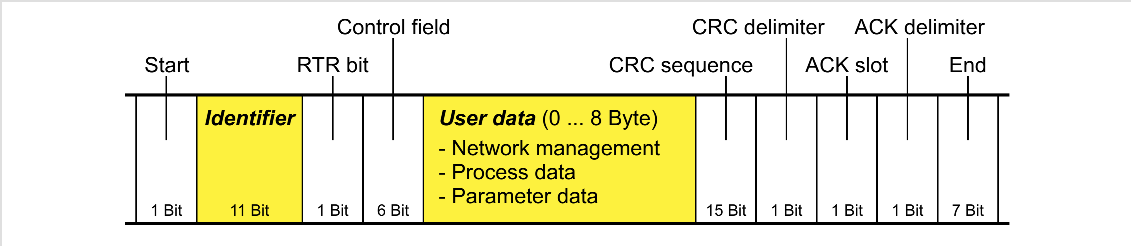
**Note!**

- For detailed information about the PC system bus adapter, please see the "CAN Communication Manual".
- Please observe the documentation for the PC system bus adapter!
- In the »Engineer«, go to the *Device assignment offline devices* dialog box and select the "System bus CAN" entry from the **Bus connection** list field to establish an online connection.

## 11.5 Reinitialising the CANopen system bus interface

The C00002/26 = "1: On / start" device command reinitialises the CANopen system bus interface of the controller ("Reset node"), which is required after e.g. changing the data transfer rate, the node address or the identifiers, respectively.

## 11.6 Structure of the CAN data telegram




[11-2] Basic structure of the CAN telegram

The following subchapters provide a detailed description of the identifier and the user data. The other signals refer to the transfer characteristics of the CAN telegram whose description is not included in the scope of this documentation.

**Tip!**

Please visit the homepage of the CAN user organisation CiA (CAN in automation) for further information:
http://www.can-cia.org

### 11.6.1 Identifier

The principle of the CAN communication is based on a message-oriented data exchange between a transmitter and many receivers. All nodes can transmit and receive quasi-simultaneously.

The identifier, also called COB-ID (abbr. for communication object identifier), is used to control which node is to receive a transmitted message. In addition to the addressing, the identifier contains information on the priority of the message and the type of user data.

The identifier consists of a basic identifier and the node address of the node to be addressed:

**Identifier (COB-ID) = basic identifier + node address (node ID)**

Exception: The identifier for process data/heartbeat/emergency objects as well as network management and sync telegrams is freely assigned by the user (either manually or automatically by the network configurator), or is permanently assigned.

### Node address (node ID)

Every node of the system bus network must be assigned to a node address (also called node ID) within the valid address range (1 ... 127) for unambiguous identification.

- Assigning a node address more than once within a network is impermissible.
- The own node address can be configured via the DIP switches or via code C00350.
  ▸ Setting the node address (📖 643)

### Identifier assignment

The system bus is message-oriented instead of node-oriented. Every message has an unambiguous identification, the identifier. For CANopen, node-oriented transfer is achieved by the fact that every message has only one transmitter.

- The basic identifiers for network management (NMT) and sync as well as the basic SDO channel (SDO1) are defined in the CANopen protocol and cannot be changed.
- In the Lenze setting, the basic identifiers of the PDOs are preset according to the "Predefined connection set" of DS301, V4.02 and can be changed via parameters/indexes if required. ▸ Identifiers of the process data objects (📖 661)

| Object | | Direction | | Lenze-Base-ID | | CANopen-Base-ID | |
|---|---|---|---|---|---|---|---|
| | | from device | to device | dec | hex | dec | hex |
| Network management (NMT) | | | | 0 | 0 | 0 | 0 |
| Sync 1) | | | | 128 | 80 | 128 | 80 |
| Emergency 1) | | ● | | 128 | 80 | 128 | 80 |
| PDO1 (Process data channel 1) | TPDO1 | ● | | 384 | 180 | 384 | 180 |
| | RPDO1 | | ● | 512 | 200 | 512 | 200 |
| PDO2 (Process data channel 2) | TPDO2 | ● | | 640 | 280 | 640 | 280 |
| | RPDO2 | | ● | 641 | 281 | 768 | 300 |
| PDO3 (Process data channel 3) | TPDO3 | ● | | 768 | 300 | 896 | 380 |
| | RPDO3 | | ● | 769 | 301 | 1024 | 400 |
| SDO1 (Parameter data channel 1) | TSDO1 | ● | | 1408 | 580 | 1408 | 580 |
| | RSDO1 | | ● | 1536 | 600 | 1536 | 600 |
| SDO2 (Parameter data channel 2) | TSDO2 | ● | | 1472 | 5C0 | 1472 | 5C0 |
| | RSDO2 | | ● | 1600 | 640 | 1600 | 640 |
| Heartbeat | | ● | | 1792 | 700 | 1792 | 700 |
| Boot-up | | ● | | 1792 | 700 | 1792 | 700 |

1) If you set the sync transmit/receive identifier manually, observe the use of the emergency telegram, since it has the same COB-ID.

### 11.6.2 User data

All nodes communicate by exchanging data telegrams via the system bus. The user data area of the CAN telegram either contains network management data or parameter data or process data:

**Network management data**

(NMT data)

- Control information on start, stop, reset, etc. of communication to specific nodes or to all nodes of the CAN network.

**Process data**

(PDOs – process data objects)

- Process data are transferred via the process data channel.
- Process data can be used to control the controller.
- Process data are not saved to the controller.
- Process data are transmitted between host system and nodes to ensure continuous exchange of current input and output data.
- Process data usually are unscaled/scalable raw data.
- Process data are, for instance, setpoints and actual values.
- The exact meaning of the PDO file contents is determined via the function block editor (FB Editor) in the I/O level or via the PDO mapping.

**Parameter data**

(SDOs – service data objects)

- Parameter data are the CANopen indexes or, in case of Lenze devices, the codes.
- Parameters are, for instance, used for one-off plant setting during commissioning or when the material is changed on a production machine.
- Parameter data are transmitted as SDOs via the parameter data channel. They are acknowledged by the receiver, i.e. the transmitter gets a feedback about the transmission being successful or not.
- The parameter data channel enables access to all Lenze codes and CANopen indexes.
- Parameter changes are automatically saved to the controller until mains switching.
- In general, the parameter transfer is not time-critical.
- Parameter data are, for instance, operating parameters, diagnostic information and motor data as well as control information on the interconnection of function blocks in the I/O level of the FB Editor.

## 11.7 Communication phases/network management

Regarding communication via the system bus, the controller distinguishes between the following statuses:

| Status | Explanation |
| --- | --- |
| **"Initialisation"**<br>(Initialisation) | After switch-on, an initialisation run is carried out.<br>• During this phase, the controller is not involved in the data exchange via the bus.<br>• The standard values are re-written to all CAN-relevant parameters.<br>• After initialisation is completed, the controller is automatically set to the "Pre-Operational" status. |
| **"Pre-Operational"**<br>(before being ready for operation) | Parameter data can be received, process data are ignored. |
| **"Operational"**<br>(ready for operation) | Parameter data and process data can be received! |
| **"Stopped"**<br>(stopped) | Only network management telegrams can be received. |

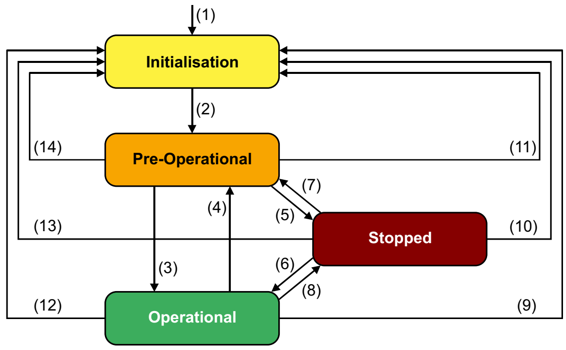
| Communication object | Initialisation | Pre-Operational | Operational | Stopped |
| --- | --- | --- | --- | --- |
| PDO | | | ● | |
| SDO | | ● | ● | |
| Sync | | ● | ● | |
| Emergency | | ● | ● | |
| Boot-up | ● | | | |
| Network management (NMT) | | ● | ● | ● |

**Tip!**

Part of the initialisation or the entire initialisation can be carried out anew in every status by transferring the corresponding network management telegrams.

The current CAN status is displayed in C00359 for diagnostic purposes.

### 11.7.1 Status transitions

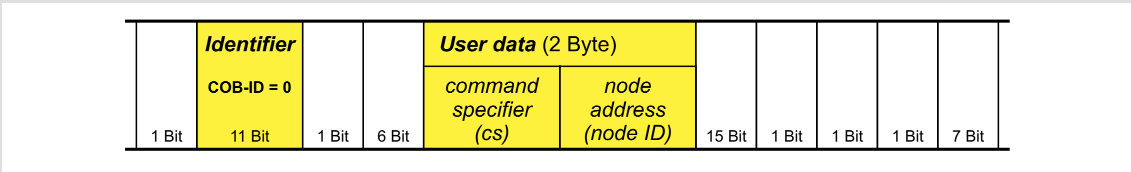



[11-3] NMT status transitions in the CAN network

| Transition | NMT command | Status after change | Effects on process/parameter data after status change |
|---|---|---|---|
| (1) | - | Initialisation | Initialisation starts automatically when the mains is switched on.<br>• During initialisation, the controller is not involved in the data exchange.<br>• After the initialisation is completed, the node sends a boot-up message with an individual identifier and automatically changes to the "pre-operational" status. |
| (2) | - | Pre-Operational | In this phase, the master determines the way in which the node(s) takes/take part in communication. |
| | From here, the master changes the statuses for the entire network.<br>• A target address included in the NMT command defines the receiver(s).<br>• If the 8400 controller is configured as CAN master, the status is automatically changed to "Operational" after a waiting time has expired (C00356/1), and the 0x0100 ("Start remote node") NMT command is transmitted to all nodes.<br>• Data can only be exchanged via process data objects if the status is "Operational"! | | |
| (3), (6) | 0x01 xx<br>Start remote node | Operational | Network management/sync/emergency telegrams as well as process data (PDO) and parameter data (SDO) are active.<br>Optional: When the status is changed, event and time-controlled process data (PDOs) are transmitted once. |
| (4), (7) | 0x80 xx<br>Enter Pre-Operational | Pre-Operational | Network management/sync/emergency telegrams and parameter data (SDO) are active. |
| (5), (8) | 0x02 xx<br>Stop remote node | Stopped | Only network management telegrams can be received. |
| (9), (10), (11) | 0x81 xx<br>Reset node | Initialisation | All CAN-relevant parameters (CiA DS 301) are initialised with the saved values. |
| (12), (13), (14) | 0x82 xx<br>Reset communication | | All CAN-relevant parameters (CiA DS 301) are initialised with the saved values. |
| | Meaning of the node address in the NMT command:<br>• xx = 0x00: If this assignment is selected, the telegram addresses all nodes (broadcast telegram). The status of all nodes can be changed at the same time.<br>• xx = Node ID: If a node address is specified, only the status of the node with the corresponding address changes. | | |

## 11.7.2 Network management telegram (NMT)

The telegram for the network management contains identifier "0" and the command included in the user data which consists of the command byte and the node address:




[11-4] Network management telegram for changing over the communication phases

| Command specifier (cs) | | NMT command |
|---|---|---|
| dec | hex | |
| 1 | 0x01 | Start remote node |
| 2 | 0x02 | Stop remote node |
| 128 | 0x80 | Enter Pre-Operational |
| 129 | 0x81 | Reset node |
| 130 | 0x82 | Reset communication |

The change-over of the communication phases for the entire network is carried out by one node, the CAN master. The function of the CAN master can also be carried out by the controller. ▸ Parameterising the controller as CAN master (📖 652)

Meaning of the node address in the user data:

- node ID = "0": The telegram addresses all nodes (broadcast telegram). The status of all nodes can be changed at the same time.
- node ID = "1" ... "127": If a node address is specified, only the status of the node with the corresponding address changes.

**Example:**

Data can only be exchanged via process data objects if the status is "Operational".If the CAN master is supposed to switch all nodes connected to the bus from the "Pre-Operational" communication status to the "Operational" communication status, the identifier and user data in the transmission telegram must be set as follows:

- Identifier: 0x00 (network management)
- User data: 0x0100 ("Start remote node" NMT command to all nodes)

### 11.7.3 Parameterising the controller as CAN master

If the initialisation of the system bus and the associated status change from "Pre-Operational" to "Operational" is not effected by a superimposed host system, the controller can instead be defined to be a "quasi" master to execute this task.

The controller is configured as CAN master in C00352.

- Being the CAN master, the controller sets all nodes connected to the bus (broadcast telegram) to the "Operational" communication status with the "Start remote node" NMT telegram. Only in this communication status, data can be exchanged via process data objects.
- A delay time can be set in C00356/1 which must expire after mains switching before the controller transmits the "Start remote node" NMT telegram.

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00352 | CAN Slave/Master | Slave | |
| C00356/1 | CAN delay boot-up - Operational | 3000 | ms |

**Note!**

The changes of the master/slave operation in C00352 will not be activated until

- another mains switching of the controller

or

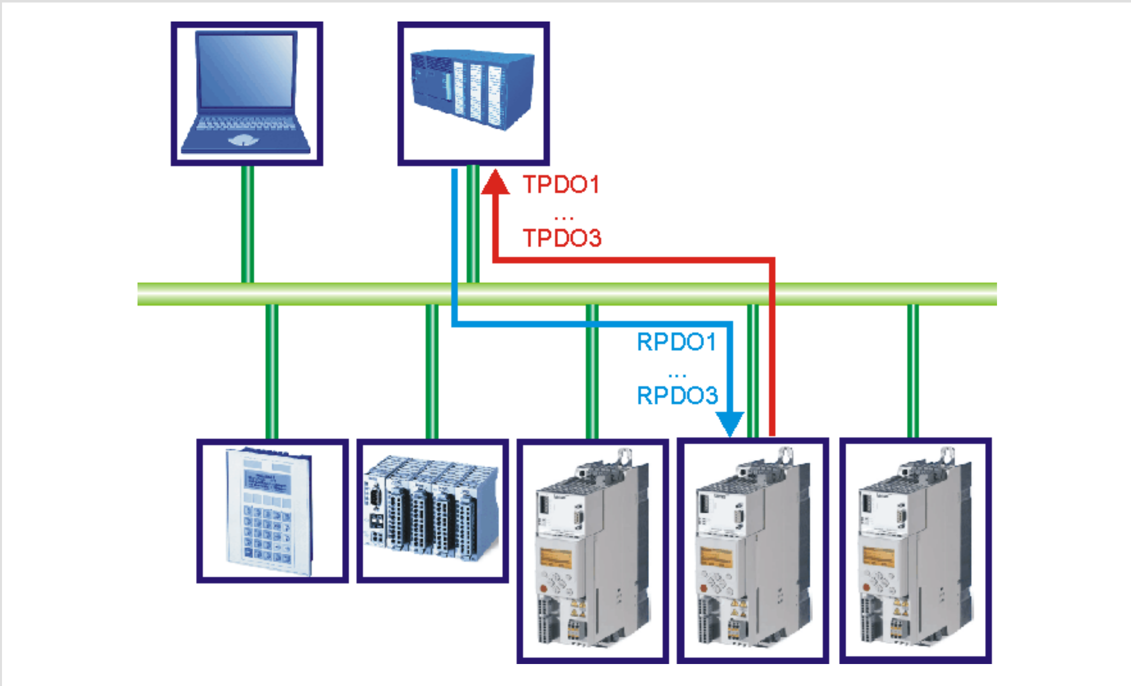
- the "Reset node" or "Reset communication" NMT telegram has been transmitted to the controller.

The "CAN reset node" device command (C00002/26) is provided as an alternative to the "Reset node" NMT telegram for the reinitialisation of the CAN-specific device parameters.

**Tip!**

Master functionality is only required during the initialisation phase of the drive system.

## 11.8 Process data transfer




[11-5] PDO data transfer from / to the higher-level host system

"BaseLine C" versions have two separate process channels (PDO1 and PDO2) and from the "StateLine" version three separate process data channels (PDO1 ... PDO3) for process data transfer.

### Definitions

- Process data telegrams between the host system and the devices are distinguished in terms of direction as follows:
  - Process data telegrams to the device (RPDO)
  - Process data telegrams from the device (TPDO)
- The CANopen process data objects are designated as seen from the node's view:
  - Receive PDOs (RPDOx): Process data object received by a node
  - Transmit PDOs (TPDOx): Process data object sent by a node

**Note!**

Data can only be exchanged via process data objects if the status is "Operational"!

▶ Communication phases/network management (📖 649)

### 11.8.1 Available process data objects

Controllers of the 8400 series have a maximum number of 3 receive PDOs (RPDOs) and 3 transmit PDOs (TPDOs).

| Process data object | Version "BaseLine C" | from version "StateLine" |
|---|---|---|
| RPDO1 \| Port block "LP_CanIn1" | ● | ● |
| RPDO2 \| Port block "LP_CanIn2" | ● | ● |
| RPDO3 \| Port block "LP_CanIn3" | | ● |
| TPDO1 \| Port block "LP_CanOut1" | ● | ● |
| TPDO2 \| Port block "LP_CanOut2" | ● | ● |
| TPDO3 \| Port block "LP_CanOut3" | | ● |

#### Receive PDOs (RPDOs)

The process data objects transmitted from the CAN bus to the drive are processed via the LP_CanIn1 ... LP_CanIn3 port blocks.

- Every port block provides 4 words (2 bytes/word). The data of every first word are provided in a bit decoded manner (bit 0 ... 15).
- The first word of the LP_CanIn1 port block is defined as control word *wCtrl*. The *wCtrl* control word does not have a permanent connection to the device control and can be used as required. The predefined assignment of the *wCtrl* control word in the C00007 = "30: CAN" control mode depends on the technology application selected in C00005:
  - **TA "Actuating drive speed":** Process data assignment for fieldbus communication (📖 403)
  - **TA "Table positioning":** Process data assignment for fieldbus communication (📖 434)
  - **TA "Switch-off positioning":** Process data assignment for fieldbus communication (📖 460)

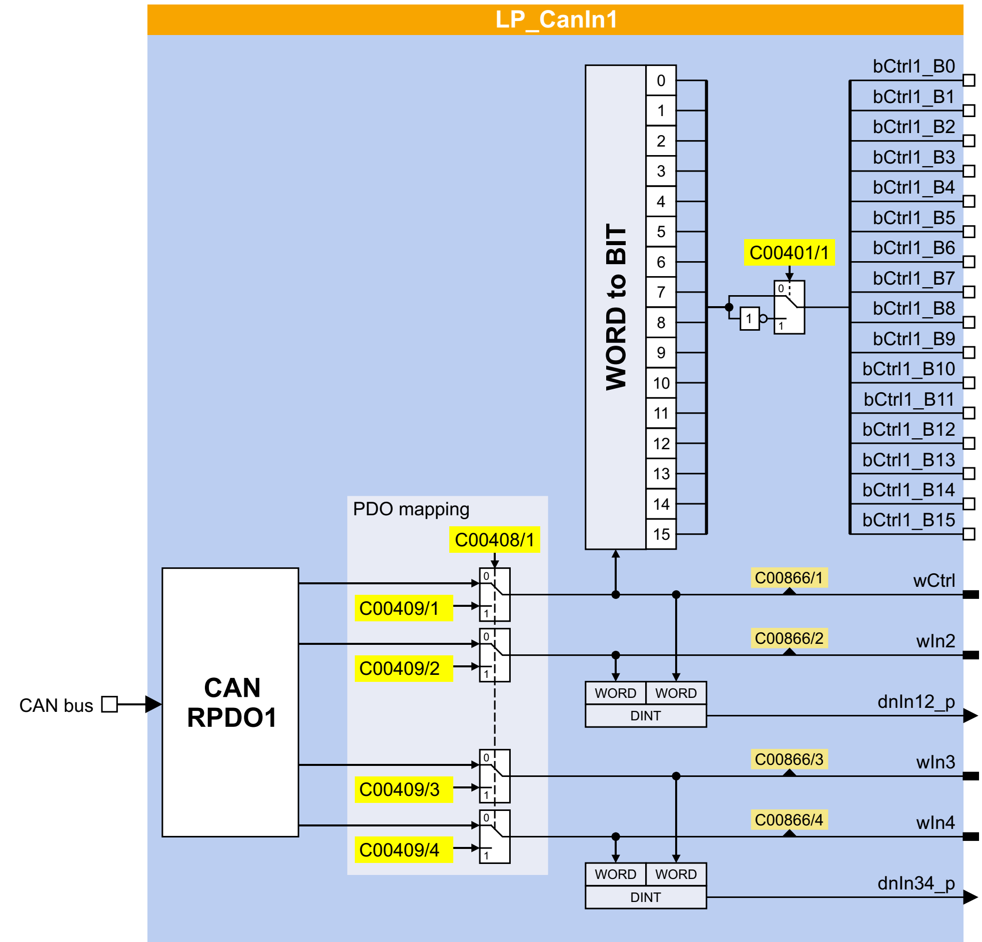
#### Transmit PDOs (TPDOs)

The process data transmitted from the drive to the CAN bust are processed via the LP_CanOut1 ... LP_CanOut3 port blocks.

- Every port block receives 4 words (2 bytes/word). The data of every first word are transmitted bit by bit (bit 0 ... 15).
- The first word of the LP_CanOut1 port block is defined as the *wState* status word. The *wState* status word does not have a permanent connection to the device control and can be used as required.
  - For the predefined assignment, see the wDeviceStatusWord status word of the drive interface.

### 11.8.1.1 RPDO1 | Port block "LP_CanIn1"

The **LP_CanIn1** port block maps process data object RPDO1 in the FB Editor.



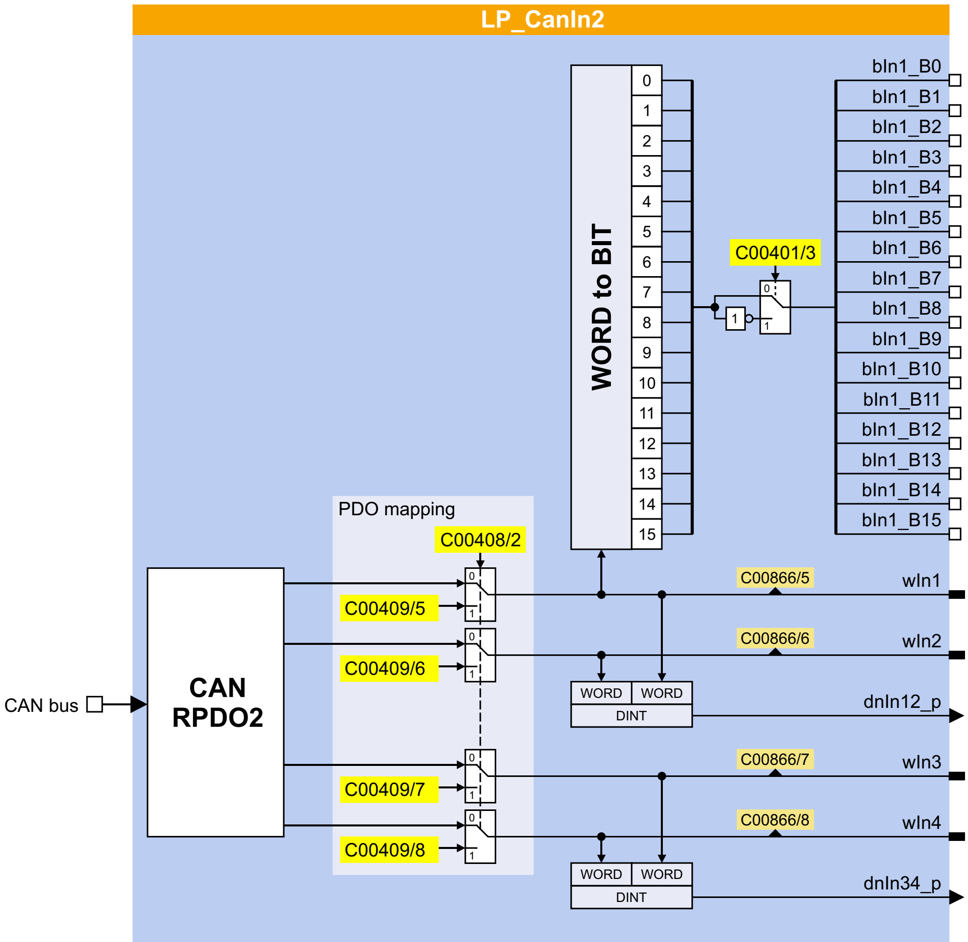

*dnIn12_p* output is available from version 02.00.00!

#### Short overview of the parameters for LP_CanIn1:

| Parameter | Info | Lenze setting |
|---|---|---|
| C00401/1 | LP_CanIn1: Inversion bCtrl1_B0..15 | 0x0000 |
| C00866/1 | LP_CanIn1: wCtrl | - |
| C00866/2 | LP_CanIn1: wIn2 | - |
| C00866/3 | LP_CanIn1: wIn3 | - |
| C00866/4 | LP_CanIn1: wIn4 | - |
| **PDO mapping** | | |
| C00408/1 | LP_CanIn1: Mapping selection | CanIn |
| C00409/1 | LP_CanIn1: wCtrl MapVal | 0 |
| C00409/2 | LP_CanIn1: wIn2 MapVal | 0 |
| C00409/3 | LP_CanIn1: wIn3 MapVal | 0 |
| C00409/4 | LP_CanIn1: wIn4 MapVal | 0 |
| Highlighted in grey = display parameter | | |

### 11.8.1.2 RPDO2 | Port block "LP_CanIn2"

The **LP_CanIn2** port block maps process data object RPDO2 in the FB Editor.



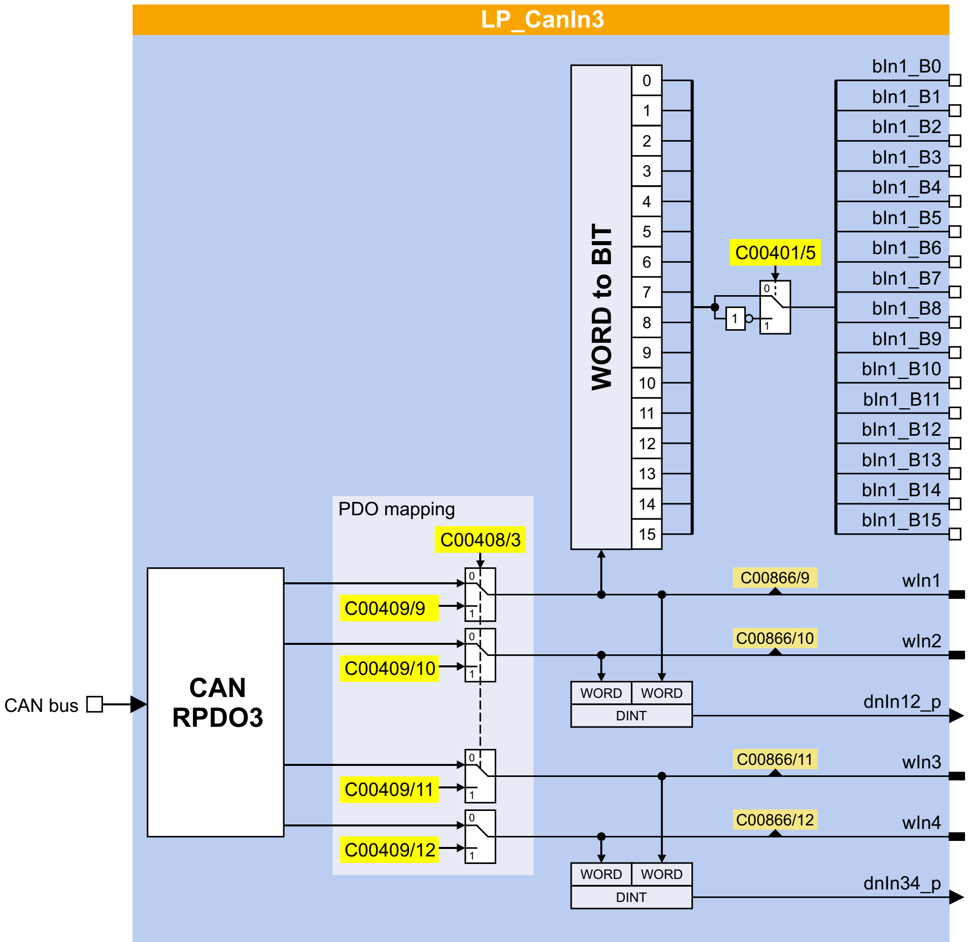

*dnIn12_p* output is available from version 02.00.00!

### Short overview of the parameters for LP_CanIn2:

| Parameter | Info | Lenze setting |
|---|---|---|
| C00401/3 | LP_CanIn2: Inversion bIn1_B0..15 | 0x0000 |
| C00866/5 | LP_CanIn2: wIn1 | - |
| C00866/6 | LP_CanIn2: wIn2 | - |
| C00866/7 | LP_CanIn2: wIn3 | - |
| C00866/8 | LP_CanIn2: wIn4 | - |
| **PDO mapping** | | |
| C00408/2 | LP_CanIn2: Mapping selection | CanIn |
| C00409/5 | LP_CanIn2: wIn1 MapVal | 0 |
| C00409/6 | LP_CanIn2: wIn2 MapVal | 0 |
| C00409/7 | LP_CanIn2: wIn3 MapVal | 0 |
| C00409/8 | LP_CanIn2: wIn4 MapVal | 0 |
| Highlighted in grey = display parameter | | |

### 11.8.1.3 RPDO3 | Port block "LP_CanIn3"

The **LP_CanIn3** port block maps process data object RPDO3 in the FB Editor.



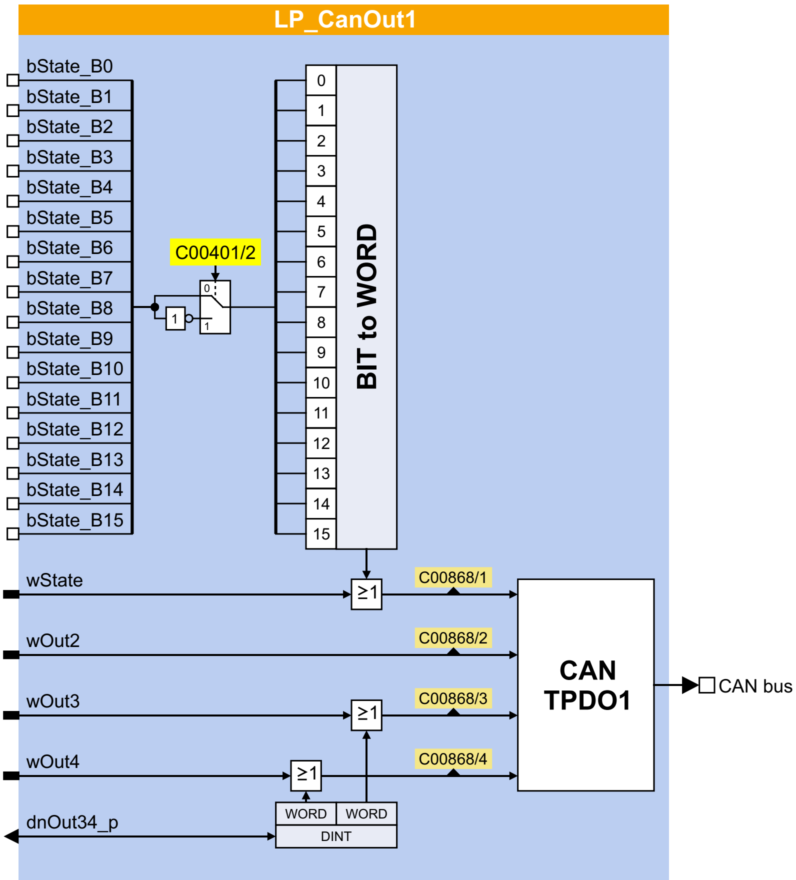

*dnIn12_p* output is available from version 02.00.00!

#### Short overview of the parameters for LP_CanIn3:

| Parameter | Info | Lenze setting |
|---|---|---|
| C00401/5 | LP_CanIn3: Inversion bIn1_B0..15 | 0x0000 |
| C00866/9 | LP_CanIn3: wIn1 | - |
| C00866/10 | LP_CanIn3: wIn2 | - |
| C00866/11 | LP_CanIn3: wIn3 | - |
| C00866/12 | LP_CanIn3: wIn4 | - |
| **PDO mapping** | | |
| C00408/3 | LP_CanIn3: Mapping selection | CanIn |
| C00409/9 | LP_CanIn3: wIn1 MapVal | 0 |
| C00409/10 | LP_CanIn3: wIn2 MapVal | 0 |
| C00409/11 | LP_CanIn3: wIn3 MapVal | 0 |
| C00409/12 | LP_CanIn3: wIn4 MapVal | 0 |
| Highlighted in grey = display parameter | | |

### 11.8.1.4 TPDO1 | Port block "LP_CanOut1"

The **LP_CanOut1** port block maps process data object TPDO1 in the FB Editor.

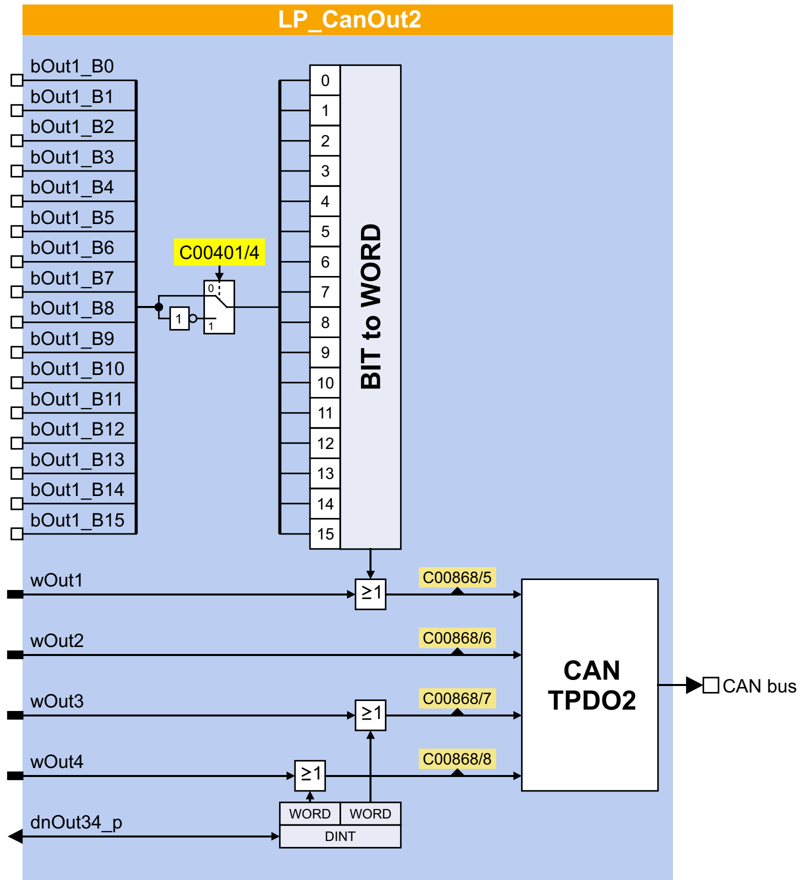



#### Short overview of the parameters for LP_CanOut1:

| Parameter | Info | Lenze setting |
|---|---|---|
| C00401/2 | LP_CanOut1: Inversion bState_B0..15 | 0x0000 |
| C00868/1 | LP_CanOut1:wState | - |
| C00868/2 | LP_CanOut1:wOut2 | - |
| C00868/3 | LP_CanOut1:wOut3 | - |
| C00868/4 | LP_CanOut1: wOut4 | - |
| Highlighted in grey = display parameter | | |

### 11.8.1.5 TPDO2 | Port block "LP_CanOut2"

The **LP_CanOut2** port block maps process data object TPDO2 in the FB Editor.

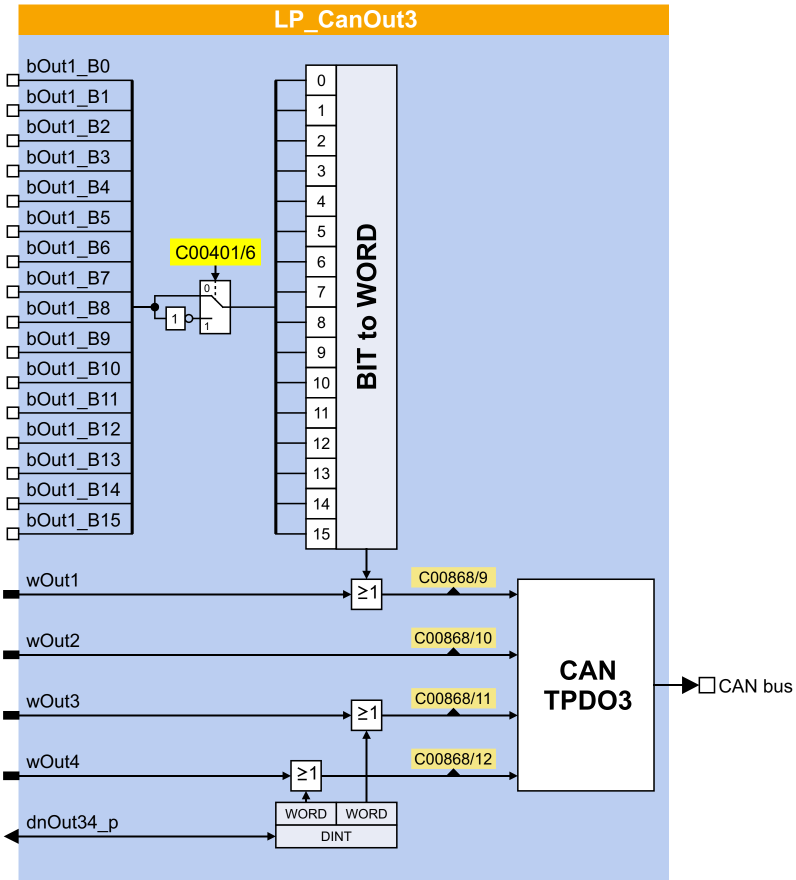



**Short overview of the parameters for LP_CanOut2:**

| Parameter | Info | Lenze setting |
|---|---|---|
| C00401/4 | LP_CanOut2: Inversion bOut1_B0..15 | 0x0000 |
| C00868/5 | LP_CanOut2: wOut1 | - |
| C00868/6 | LP_CanOut2: wOut2 | - |
| C00868/7 | LP_CanOut2: wOut3 | - |
| C00868/8 | LP_CanOut2: wOut4 | - |
| Highlighted in grey = display parameter | | |

### 11.8.1.6 TPDO3 | Port block "LP_CanOut3"

The **LP_CanOut3** port block maps process data object TPDO3 in the FB Editor.




#### Short overview of the parameters for LP_CanOut3:

| Parameter | Info | Lenze setting |
|---|---|---|
| C00401/6 | LP_CanOut3: Inversion bOut1_B0..15 | 0x0000 |
| C00868/9 | LP_CanOut3: wOut1 | - |
| C00868/10 | LP_CanOut3: wOut2 | - |
| C00868/11 | LP_CanOut3: wOut3 | - |
| C00868/12 | LP_CanOut3: wOut4 | - |
| Highlighted in grey = display parameter | | |

### 11.8.2 Identifiers of the process data objects

In the Lenze setting, the identifier for process data objects PDO1 ... PDO3 consists of a so-called basic identifier (CANBaseID) and the node address set in C00350:

**Identifier (COB-ID) = basic identifier + node address (node ID)**

- The basic identifiers of the PDOs comply with the "Predefined connection set" of DS301, V4.02.
- Alternatively, define via code C00353 that the identifiers of the PDOs are to be assigned according to Lenze definition or that individual settings are to be made.
  - If C00353 = "2: COBID = C0354/x", the identifiers of the PDOs can be individually set via the Lenze codes and CANopen indexes listed in the table below. That way, identifiers independent of the node address can be set for specific PDOs.
  - If identifiers are assigned individually, all PDOs must have basic identifier values in the range of 385 ... 1407.

| Process data object | Basic identifier | | Individual setting | |
|---|---|---|---|---|
| | dec | hex | Lenze code | CANopen index |
| PDO1 | | | | |
| RPDO1 | 512 | 0x200 | C00354/1 | I-1400/1 |
| TPDO1 | 384 | 0x180 | C00354/2 | I-1800/1 |
| PDO2 | | | | |
| RPDO2 | 768 | 0x300 | C00354/3 | I-1401/1 |
| TPDO2 | 640 | 0x280 | C00354/4 | I-1801/1 |
| PDO3 | | | | |
| RPDO3 | 1024 | 0x400 | C00354/5 | I-1402/1 |
| TPDO3 | 896 | 0x380 | C00354/6 | I-1802/1 |

**Note!**

After a node address change (C00350) and a CAN reset node afterwards, the subcodes of C00354 automatically resume the values which result from the respective basic identifier and the set node address.

#### Short overview: Parameters for setting the identifiers

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00353/1 | COBID source CAN1_IN/OUT | 0: COBID = C0350 + CANBaseID | |
| C00353/2 | COBID source CAN2_IN/OUT | 0: COBID = C0350 + CANBaseID | |
| C00353/3 | COBID source CAN3_IN/OUT | 0: COBID = C0350 + CANBaseID | |
| C00354/1 | COBID CAN1_IN | 0x00000201 | |
| C00354/2 | COBID CAN1_OUT | 0x00000181 | |
| C00354/3 | COBID CAN2_IN | 0x00000301 | |
| C00354/4 | COBID CAN2_OUT | 0x00000281 | |
| C00354/5 | COBID CAN3_IN | 0x00000401 | |
| C00354/6 | COBID CAN3_OUT | 0x00000381 | |

### 11.8.3 Transmission type

Process data objects can be transmitted in an event-controlled or time-controlled manner. The below table shows that it is possible to combine the different methods by means of logic operations (AND, OR):

- Event-controlled
  The PDO is sent when a special device-internal event has occurred, e.g. when the data contents of the TPDO have changed or when a transmission cycle time has elapsed
- Synchronous transmission
  A TPDO (or RPDO) is transmitted (or received) after the device has received a sync telegram (COB-ID 0x80).
- Cyclic transmission
  The cyclic transmission of PDOs takes place when the transmission cycle time has elapsed.
- Polled via RTR
  A TPDO is transmitted when another device requests it by means of a data request telegram (RTR remote transmit request). For this purpose, the data requester (e.g. the master) sends the data request telegram with the COB-ID of the TPDO requested to be sent. The receiver recognises the RTR and transmits the corresponding PDO.

| Transmission type | PDO transmission | | | Logic combination of different transmission types |
|---|---|---|---|---|
| | cyclic | synchronous | event-controlled | |
| 0 | | ● | ● | AND |
| 1 ... 240 | | ● | | - |
| 254, 255 | ● | | ● | OR |

| Transmission type | Description |
|---|---|
| 0 | Synchronous and acyclic:<br>The PDO is transmitted on an event-controlled basis with every sync (e.g. when a bit change occurs in the PDO). |
| 1 ... 240 | Synchronous and cyclic (sync-controlled with response):<br>• Selection n = 1: The PDO is transmitted with every sync.<br>• Selection 1 < n ≤ 240: The PDO is transmitted with every n-th sync. |
| 241 ... 251 | Reserved |
| 252 | Synchronous - RTR only |
| 253 | Asynchronous - RTR only |
| 254, 255 | Asynchronous - manufacturer-specific / device profile-specific:<br>If this value is entered, the PDO transmission is event-controlled or cyclic.<br>(Note: The values "254" and "255" have the same meaning).<br>For a cyclic transmission, a cycle time must be entered for the respective PDO. In this case, cyclic transmission takes place in addition to event-controlled transmission. |

The communication parameters such as the transmission mode and cycle time can be set freely for every PDO and independently of the settings of other PDOs:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| **CAN1_OUT** | | | |
| C00322/1 | Transmission mode | 254 | |
| C00324/2 | Blocking time | 0 | ms |
| C00356/5 | Cycle time | 0 | ms |
| C00358/1 | Data length | 8 | Byte |
| **CAN2_OUT** | | | |
| C00322/2 | Transmission mode | 254 | |
| C00324/3 | Blocking time | 0 | ms |
| C00356/2 | Cycle time | 0 | ms |
| C00358/2 | Data length | 8 | Byte |
| **CAN3_OUT** | | | |
| C00322/3 | Transmission mode | 254 | |
| C00324/4 | Blocking time | 0 | ms |
| C00356/3 | Cycle time | 0 | ms |
| C00358/3 | Data length | 8 | Byte |
| **CAN1_IN ... CAN3_IN** | | | |
| C00323/1...3 | Transmission mode CAN1_IN ... CAN3_IN<br>• In the case of the RPDO serves as monitoring setting in the case of sync-controlled PDOs. | 254 | |

### Blocking time

In C00324/x a "blocking time" can be set which defines the shortest transmission cycle with the transmission type"asynchronous - manufacturer-specific/device profile-specific".

Example: Cycle time = 500 ms, blocking time = 100 ms, data change sporadically:

- In the case of a sporadic data change < 500 ms, due to the blocking time set, a transmission takes place every 100 ms as quickly as possible (event-controlled transmission). The transmission cycle timer is reset to 0 if "event-controlled transmission" has been activated.
- In the case of a sporadic data change > 500 ms, due to the cycle time set, transmission takes place every 500 ms (cyclic transmission).

**Tip!**

The communication parameters can also be set via the following CANopen objects:

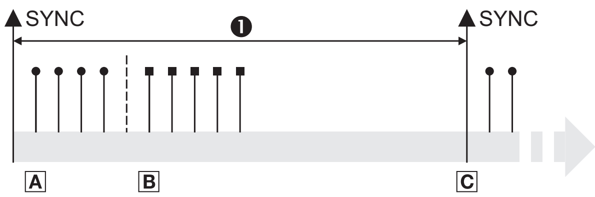
- I-1400 ... I-1402: Communication parameters for RPDO1 ... RPDO3
- I-1800 ... I-1802: Communication parameters for TPDO1 ... TPDO3

### 11.8.4 PDO synchronisation via sync telegram

During cyclic transmission, one or more PDOs are transmitted/received in fixed time intervals. An additional specific telegram, the so-called sync telegram, is used for synchronising cyclic process data.

- The sync telegram is the trigger point for the transmission of process data from the slaves to the master and for the acceptance of process data from the master in the slaves.
- For sync-controlled process data processing, the sync telegram must be generated accordingly.
- The response to a sync telegram is determined by the selected transmission type. Transmission type (662)

#### Basic workflow




❶ Sync cycle time (C01121)

[11-6] Sync telegram

A. After the sync telegram has been received, the slaves transmit the synchronous process data to the master (TPDOs). The master reads them as process input data.
B. When the transmission process is completed, the slaves receive (RPDOs) the process output data (of the master).
   - All other telegrams (e.g. parameters or event-controlled process data) are accepted acyclically by the slaves after the transmission is completed.
   - Illustration [11-6] does not include acyclic data. However, they need to be considered when dimensioning the cycle time.
C. The data are accepted in the slave with the next sync telegram if the Rx mode is set to 1 ... 240. If the Rx mode is 254 or 255, the data are accepted in the next device cycle, irrespective of the sync telegram.

#### Short overview: Parameters for the synchronisation via sync telegram

| Parameter | Info | Lenze setting | | Assignment | |
|---|---|---|---|---|---|
| | | Value | Unit | Sync master | Sync slave |
| C00367 | CAN Sync-Rx-Identifier | 128 | | | ● |
| C00368 | CAN Sync-Tx-Identifier | 128 | | ● | |
| C00369 | CAN Sync transmission cycle time | 0 | ms | ● | |

#### Related topics:

Synchronisation of the internal time base (731)

### 11.8.5 Monitoring of the RPDOs for data reception

Every RPDO1 ... RPDO3 has a parameterisable monitoring time in which the RPDO must arrive. If the RPDO is not received within the monitoring time or with the configured sync, the response parameterised for the respective RPDO is activated.

#### Short overview: Parameters for RPDO monitoring

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00357/1...3 | CAN1...3_IN monitoring time | 3000 | ms |
| C00593/1...3 | Resp. to CAN1...3_IN monitoring | No response | |

### 11.8.6 Configuring exception handling of the CAN PDOs

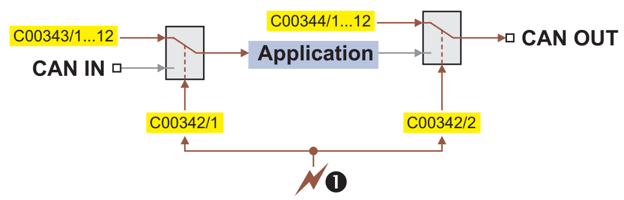
Exception handling for the CAN PDOs in the event of an error can be set via decoupling configuration and decoupling values.

- Bit coded selection is carried out in C00342/1 for the process data words read by the bus, defining the events that will trigger decoupling.
- Bit coded selection is carried out in C00342/2 for the process data words output by the application, defining the events that will trigger decoupling.

| Bit | Event |
|---|---|
| Bit 0 ☐ | BusOff_MsgErr |
| Bit 1 ☐ | Warning |
| Bit 2 ☐ | NodeStopped |
| Bit 3 ☐ | HeartBeatEvent |
| Bit 4 ☐ | CAN1_In_Überw. |
| Bit 5 ☐ | CAN2_In_Überw. |
| Bit 6 ☐ | CAN3_In_Überw. |
| Bit 7 ☐ | Reserved |
| Bit 8 ☐ | Reserved |
| Bit 9 ☐ | Reserved |
| Bit 10 ☐ | Reserved |
| Bit 11 ☐ | Reserved |
| Bit 12 ☐ | Reserved |
| Bit 13 ☐ | Reserved |
| Bit 14 ☐ | Trouble |
| Bit 15 ☐ | Fault |

Finally, the following parameters define the value that the process data words are to have when they are decoupled:

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00343/1 | LP_CanIn1:wCtrl DiscVal | 0 | |
| C00343/2...4 | LP_CanIn1:wIn2...wIn4 DiscVal | 0 | |
| C00343/5...8 | LP_CanIn2:wIn1...wIn4 DiscVal | 0 | |
| C00343/9...12 | LP_CanIn3:wIn1...wIn4 DiscVal | 0 | |
| C00344/1 | LP_CanOut1:wState DiscVal | 0 | |
| C00344/2...4 | LP_CanOut1:wOut2...wOut4 DiscVal | 0 | |
| C00344/5...8 | LP_CanOut2:wOut1...wOut4 DiscVal | 0 | |
| C00344/9...12 | LP_CanOut3:wOut1...wOut4 DiscVal | 0 | |



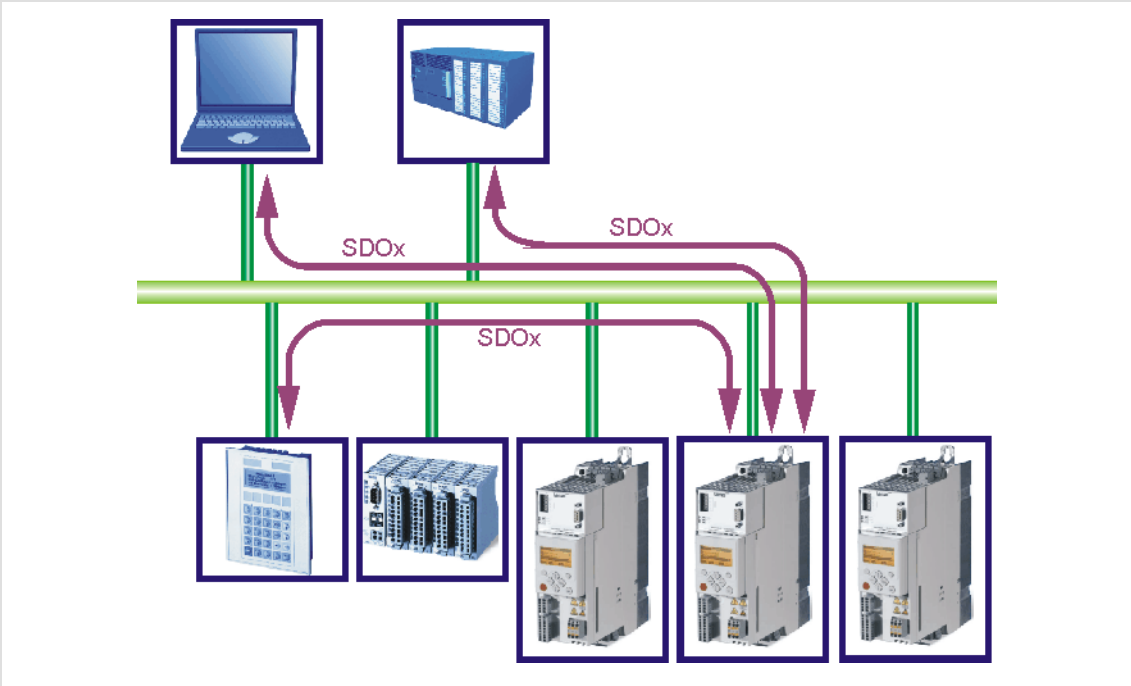

❶ Exception event (an error has occurred)

[11-7] General signal flow in the event of a configured exception

**Related topics:**

- Configuring exception handling of the output terminals (📖 369)

## 11.9 Parameter data transfer




[11-8] Parameter data transfer via the available parameter data channels

Parameters are values stored in codes on Lenze controllers.

Two parameter data channels are available for parameter setting, enabling the simultaneous connection of different devices for configuration purposes.

Parameter data are transmitted via the system bus as SDOs (*Service Data Objects*) and acknowledged by the receiver. The SDO enables read and write access to all device parameters and to the CANopen object directory integrated in the device. Indices (e.g. 0x1000) ensure access to device parameters and functions included in the object directory. To transfer SDOs, the information contained in the user data must comply with the CAN SDO protocol.

**Note!**

In the Lenze setting, only the parameter data channel 1 is activated according to CANopen.

- In C00366, set "2 SDO Lenze" to activate both parameter data channels.

### 11.9.1 Identifiers of the parameter data objects

In the Lenze setting, the basic identifiers of the SDOs are preset according to the "Predefined Connection Set".

The identifiers of the parameter data objects SDO1 and SDO2 result from the basic identifier and the node address set under code C00350:

**Identifier = basic identifier + node address**

| Object | | Direction | | Lenze-Base-ID | | CANopen-Base-ID | |
|---|---|---|---|---|---|---|---|
| | | from device | to device | dec | hex | dec | hex |
| SDO1 (Parameter data channel 1) | TSDO1 | ● | | 1408 | 580 | 1408 | 580 |
| | RSDO1 | | ● | 1536 | 600 | 1536 | 600 |
| SDO2 (Parameter data channel 2) | TSDO2 | ● | | 1472 | 5C0 | 1472 | 5C0 |
| | RSDO2 | | ● | 1600 | 640 | 1600 | 640 |
| Heartbeat | | ● | | 1792 | 700 | 1792 | 700 |
| Boot-up | | ● | | 1792 | 700 | 1792 | 700 |

### 11.9.2 User data

**Structure of the user data of the parameter data telegram**

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| **Command** | **Index** | | **Subindex** | **Data 1** | **Data 2** | **Data 3** | **Data 4** |
| | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

**Note!**

For the user data, the Motorola format is used.

▶ Parameter data telegram examples (📖 675)

The following subchapters provide detailed information on user data.

### 11.9.2.1 Command

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

The following commands can be transmitted or received for writing and reading the parameters:

| Command | 1st byte | | Data length | Info |
|---|---|---|---|---|
| | hex | dec | | |
| Write request | 0x23 | 35 | 4 bytes | Writing of a parameter to the controller. |
| | 0x2B | 43 | 2 bytes | |
| | 0x2F | 47 | 1 byte | |
| | 0x21 | 33 | Block | |
| Write response | 0x60 | 96 | 4 bytes | Controller acknowledges a write request. |
| Read request | 0x40 | 64 | 4 bytes | Reading of a parameter from the controller. |
| Read response | 0x43 | 67 | 4 bytes | Controller's response to a read request with the current parameter value. |
| | 0x4B | 75 | 2 bytes | |
| | 0x4F | 79 | 1 byte | |
| | 0x41 | 65 | Block | |
| Error response | 0x80 | 128 | 4 bytes | Controller's response if the write/read request could not be executed correctly. Error messages (673) |

More precisely, the command byte comprises the following information:

| Command | 1st byte | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Command specifier (cs) | | | Toggle (t) | Length* | | e | s |
| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Write request | 0 | 0 | 1 | 0 | 0/1 | 0/1 | 1 | 1 |
| Write response | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| Read request | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| Read response | 0 | 1 | 0 | 0 | 0/1 | 0/1 | 1 | 1 |
| Error response | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

*Bit coding of the length: 00 = 4 bytes, 01 = 3 bytes, 10 = 2 bytes, 11 = 1 byte
e: expedited (shortened block service)
s: segmented (normal block service)

**Tip!**

More commands are defined in CANopen specification DS301, V4.02 (e.g. segmented transfer).

### 11.9.2.2 Addressing by means of index and subindex

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| **Command** | **Index** | | **Subindex** | **Data 1** | **Data 2** | **Data 3** | **Data 4** |
| | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

A parameter (a Lenze code) is addressed as per the following formula:
**Index = 24575 - (Lenze code number)**

#### Example

The C00011 parameter (motor reference speed) is to be addressed.

Calculation:

- Index:
  - Decimal: 24575 - 11 = 24564
  - Hexadecimal: 0x5FFF - 0xB = 0x5FF4
- Subindex: 0x00 (subindex 0 since the parameter does not have any subcodes)

Entries:

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| **Command** | **Index** | | **Subindex** | **Data 1** | **Data 2** | **Data 3** | **Data 4** |
| | **0xF4** | **0x5F** | **0x00** | | | | |

### 11.9.2.3 Data 1 ... Data 4

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

Maximally 4 bytes are available for parameter value entries. Depending on the data format, they are assigned as follows:

| 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|
| Parameter value (1 byte) | 0x00 | 0x00 | 0x00 |
| Parameter value (2 bytes) | | 0x00 | 0x00 |
| LOW byte | HIGH byte | | |
| Parameter value (4 bytes) | | | |
| LOW word | | HIGH word | |
| LOW byte | HIGH byte | LOW byte | HIGH byte |

**Note!**

The "Factor" column of the Table of attributes contains a so-called scaling factor for all Lenze parameters. The scaling factor is relevant to the transfer of parameter values which have one or more decimal positions in the parameter list.

If the scaling factor is > 1, the value must be multiplied by the indicated scaling factor prior to transmission to be able to transfer the value as an integer. At the SDO client end, the integer must be divided by the scaling factor to obtain the original value including decimal positions again.

#### Example

A value of "123.45" is to be transmitted for a code, unit: "%" (e.g. C00039/1: "Fixed setpoint-JOG1").

In controllers of the 8400 series, parameters with the "%" unit have two decimal positions and hence a scaling factor of "100".

Calculation:

- Value to be transmitted = scaling factor x value
- Data $_{(1 \ldots 4)}$ = 100 x 123.45 = 12345 (0x00 00 30 39)

Entries:

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| | | | | 0x39 | 0x30 | 0x00 | 0x00 |

### 11.9.2.4 Error messages

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| **Command** | **Index** | | **Subindex** | **Error code** | | | |
| 0x80 (128) | LOW byte | HIGH byte | | LOW word | | HIGH word | |
| | | | | LOW byte | HIGH byte | LOW byte | HIGH byte |

In the event of an error, the addressed node generates a telegram with the "Error response" (0x80) command.

- The telegram includes the index and subindex of the code where the error occurred.
- The error code is entered in bytes 5 ... 8.
  - The error codes are standardised according to DS301, V4.02.
  - The representation of the error codes is provided in reverse read direction (see example below).

#### Example

Representation of error code "0x06 04 00 41" in bytes 5 ... 8:

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| **Command** | **Index** | | **Subindex** | **Error code** | | | |
| | | | | **0x41** | **0x00** | **0x04** | **0x06** |

### Meaning of the error codes

The error codes are standardised acc. to DS301, V4.02.

| Error code | Explanation |
|---|---|
| 0x0503 0000 | Toggle bit not changed |
| 0x0504 0000 | SDO protocol expired |
| 0x0504 0001 | Invalid or unknown client/server command specifier |
| 0x0504 0002 | Invalid block size (only block mode) |
| 0x0504 0003 | Invalid sequence number (only block mode) |
| 0x0504 0004 | CRC error (only block mode) |
| 0x0504 0005 | Not sufficient memory |
| 0x0601 0000 | Object access not supported |
| 0x0601 0001 | Attempt to read a write-only object |
| 0x0601 0002 | Attempt to write to a read-only object |
| 0x0602 0000 | Object not listed in object directory |
| 0x0604 0041 | Object not mapped to PDO |
| 0x0604 0042 | Number and length of the objects to be transferred exceed the PDP length. |
| 0x0604 0043 | General parameter incompatibility |
| 0x0604 0047 | General internal device incompatibility |
| 0x0606 0000 | Access denied because of hardware error |
| 0x0607 0010 | Unsuitable data type, unsuitable service parameter length |
| 0x0607 0012 | Unsuitable data type, service parameter length exceeded |
| 0x0607 0013 | Unsuitable data type, service parameter length not long enough |
| 0x0609 0011 | Subindex does not exist |
| 0x0609 0030 | Parameter value range exceeded |
| 0x0609 0031 | Parameter values too high |
| 0x0609 0032 | Parameter values too low |
| 0x0609 0036 | Maximum value falls below minimum value |
| 0x0800 0000 | General error |
| 0x0800 0020 | Data cannot be transferred/saved for application. |
| 0x0800 0021 | Data cannot be transferred/saved for application due to local control. |
| 0x0800 0022 | Data cannot be transferred/saved for application due to current device status. |
| 0x0800 0023 | Dynamic generation of object directory failed or no object directory available (e.g. object directory generated from file, generation not possible because of a file error). |

## 11.9.3 Parameter data telegram examples

### 11.9.3.1 Read parameters

Task: The heatsink temperature of 43 °C (code C00061, data format INTEGER16, scaling factor 1) of the controller with node address "5" is to be read.

#### Telegram to drive

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x0605 | 0x40 | 0xC2 | 0x5F | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram to the drive | |
|---|---|
| Identifier | = 1536 + node address = 1536 + 5 = 1541 = 0x0605<br>(1536 = SDO1 basic identifier to the controller) |
| Command | = 0x40 = "Read request" (request to read a parameter from the controller) |
| Index | = 24575 - code number = 24575 - 61 = 24514 = 0x5FC2 |
| Subindex | = 0 (code C00061 does not have any subcodes) |

#### Response telegram from drive (if data have been correctly transmitted)

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | - | - |
| 0x0585 | 0x4B | 0xC2 | 0x5F | 0x00 | 0x2B | 0x00 | - | - |

| Explanations on the telegram from the drive | |
|---|---|
| Identifier | = 1408 + node address = 1408 + 5 = 1413 = 0x0585<br>(1408 = SDO1 basic identifier from the controller) |
| Command | = 0x4B = "Read Response" (response to the read request with current value) |
| Index | as in telegram to the drive |
| Subindex | |
| Data 1 ... 2 | = 0x002B = 43 [°C] |

### 11.9.3.2 Write parameters

Task: The rated current of the connected motor is to be entered with $I_N$ = 10.20 A (code C00088) into the controller with node address "2".

| Data 1 ... 4 | Calculation |
|---|---|
| Value for motor current, (data type U16; display factor 1/100) | 10.20 x 100 = 1020 (0x03 FC) |

#### Telegram to drive

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x0602 | 0x23 | 0xA7 | 0x5F | 0x00 | 0xFC | 0x03 | 0x00 | 0x00 |

| Explanations on the telegram to the drive | |
|---|---|
| Identifier | = 1536 + node address = 1536 + 2 = 1538 = 0x0602<br>(1536 = SDO1 basic identifier to the controller) |
| Command | = 0x23 = "Write request" (request to write a parameter to the controller) |
| Index | = 24575 - code number = 24575 - 88 = 24487 = 0x5FA7 |
| Subindex | = 0 (code C00088 does not have any subcodes) |
| Data 1 ... 4 | = 10.20 x 100 = 1020 = 0x000003FC<br>(motor current value; data type U32; display factor 1/100) |

#### Response telegram from drive (if data have been correctly transmitted)

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x0582 | 0x60 | 0xA7 | 0x5F | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram from the drive | |
|---|---|
| Identifier | = 1408 + node address = 1408 + 2 = 1410 = 0x0582<br>(1408 = SDO1 basic identifier from the controller) |
| Command | = 0x60 = "Write response" (acknowledgement of the write access from the controller) |
| Index | as in telegram to the drive |
| Subindex | |

### 11.9.3.3 Read block parameters

Task: The firmware version (code C00099) is to be read from the parameter set of the controller with node address "12". The firmware version has a length of 11 ASCII characters which are transmitted as a block parameter. Depending on the block, the data width from the 2nd to 8th byte is assigned within the user data.

**Telegram 1 to the drive: Read request**

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x060C | 0x40 | 0x9C | 0x5F | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram to the drive | |
|---|---|
| Identifier | = 1536 + node address = 1536 + 12 = 1548 = 0x060C<br>(1536 = SDO1 basic identifier to the controller) |
| Command | = 0x40 = "Read request" (request to read a parameter from the controller) |
| Index | = 24575 - code number = 24575 - 99 = 24476 = 0x5F9C |
| Subindex | = 0 (code C00099 does not have any subcodes) |

**Response telegram 1 from the drive: Indication of the block length (11 characters)**

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x058C | 0x41 | 0x9C | 0x5F | 0x00 | 0x0B | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram from the drive | |
|---|---|
| Identifier | = 1408 + node address = 1408 + 12 = 1420 = 0x058C<br>(1408 = SDO1 basic identifier from the controller) |
| Command | = 0x41 = "Read response" (response is block telegram) |
| Index | as in telegram to the drive |
| Subindex | |
| Data 1 ... 4 | = 0x0000000B = data length of 11 characters in the ASCII format |

### Telegram 2 to the drive: Request of the 1st data block

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |
| 0x060C | 0x60 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on the telegram to the drive | |
|---|---|
| Command | = 0x60 = "Read segment request" (request: read data block)<br>• Bit 4 = 0 (toggle bit) |
| | **Influence of the toggle bit on the request command**<br>The blocks are toggled one after another, i.e. the request ist made with the "0x60" (= 0110*0000$_{bin}$) command, then with the "0x70" (= 0111*0000$_{bin}$) command, and then again with the "0x60" command, etc.<br>* Toggle bit |

### Response telegram 2 from the drive: Transmission of the 1st data block

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |
| 0x058C | 0x00 | 0x30 | 0x31 | 0x2E | 0x30 | 0x30 | 0x2E | 0x30 |
| | | $0_{asc}$ | $1_{asc}$ | $._{asc}$ | $0_{asc}$ | $0_{asc}$ | $._{asc}$ | $0_{asc}$ |

| Explanations on the telegram to the drive | |
|---|---|
| Command | = 0x00 = 00000000$_{bin}$<br>• Bit 4 = 0 (toggle bit) |
| | **Influence of the toggle bit on the transmission command**<br>• The 1st response of the controller in the command byte is "0x0000*0000$_{bin}$" if bytes 2 ... 8 are completely filled with data and other telegrams are following.<br>• The 2nd response of the controller in the command byte is "0x0001*0000$_{bin}$" if bytes 2 ... 8 are completely filled with data and other telegrams are following, etc.<br>* Toggle bit |
| Data 1 ... 7 | = "01.00.0" (ASCII representation) |

### Telegram 3 to the drive: Request of the 2nd data block

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |
| 0x060C | 0x70 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 | 0x00 |

| Explanations on telegram 3 to the drive | |
|---|---|
| Command | = 0x70 = "Read segment request" (request: read data block)<br>• Bit 4 = 1 (toggle bit) |

### Response telegram 3 from the drive: Transmission of the 2nd data block including end identifier

| Identifier | User data | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
| | Command | Data 1 | Data 2 | Data 3 | Data 4 | Data 5 | Data 6 | Data 7 |
| 0x058C | 0x17 | 0x30 | 0x2E | 0x30 | 0x30 | 0x00 | 0x00 | 0x00 |
| | | $0_{asc}$ | $._{asc}$ | $0_{asc}$ | $0_{asc}$ | - | - | - |

| Explanations on telegram 3 from the drive | |
|---|---|
| Command | = 0x17 = $00010111_{bin}$:<br>• Bit 0 = 1 (end of transmission)<br>• Bit 1 ... bit 3 = $011_{bin}$ (3 bytes do not contain any data)<br>• Bit 4 = 1 (toggle bit) |
| | **Influence of the final bit and the residual data length on the transmission command**<br>• The end of transmission is signalled via the set final bit 0.<br>• Bits 1 ... 3 reveal the number of bytes that do not contain any data anymore.<br>* Toggle bit |
| Data 1 ... 7 | = "0.00" (ASCII representation)<br>The result of the data block transmission is: "01.00.00.00" |

## 11.10 Monitoring

### 11.10.1 Integrated error detection

If a node detects an error, it rejects the CAN telegram bits received so far and transmits an error flag. The error flag consists of 6 consecutive bits with the same logic value.

The following errors are detected:

#### Bit error

The sending node follows the transmission on the bus and interrupts the transmission if it receives a different logic value than the value transmitted. With the next bit, the sending node starts the transmission of an error flag.

In the arbitration phase, the transmitter only detects a bit error if a dominantly sent bit is received as recessive bit. In the ACK slot as well, the dominant overwriting of a recessive bit is not indicated as a bit error.

#### Stuff-bit error

If more than 5 consecutive bits have the same logic value before the ACK delimiter in the CAN telegram, the previously transmitted telegram will be rejected and an error flag will be sent with the next bit.

#### CRC error

If the received CRC checksum does not correspond to the checksum calculated in the CAN chip, the CAN controller will send an error flag after the ACK delimiter and the previously transmitted telegram will be annulled.

#### Acknowledgement error

If the sent ACK slot recessively sent by the transmitting node is not dominantly overwritten by a receiver, the transmitting node will cancel the transmission. The transmitting node will annul the transmitted telegram and will send an error flags with the next bit.

#### Format error

If a dominant bit is detected in the CRC delimiter, in the ACK delimiter or in the first 6 bits of the EOF field, the received telegram will be rejected and an error flag will be sent with the next bit.

**Tip!**

The errors mentioned before indicate that a physical error has occurred in the bus system.

Possible causes are:

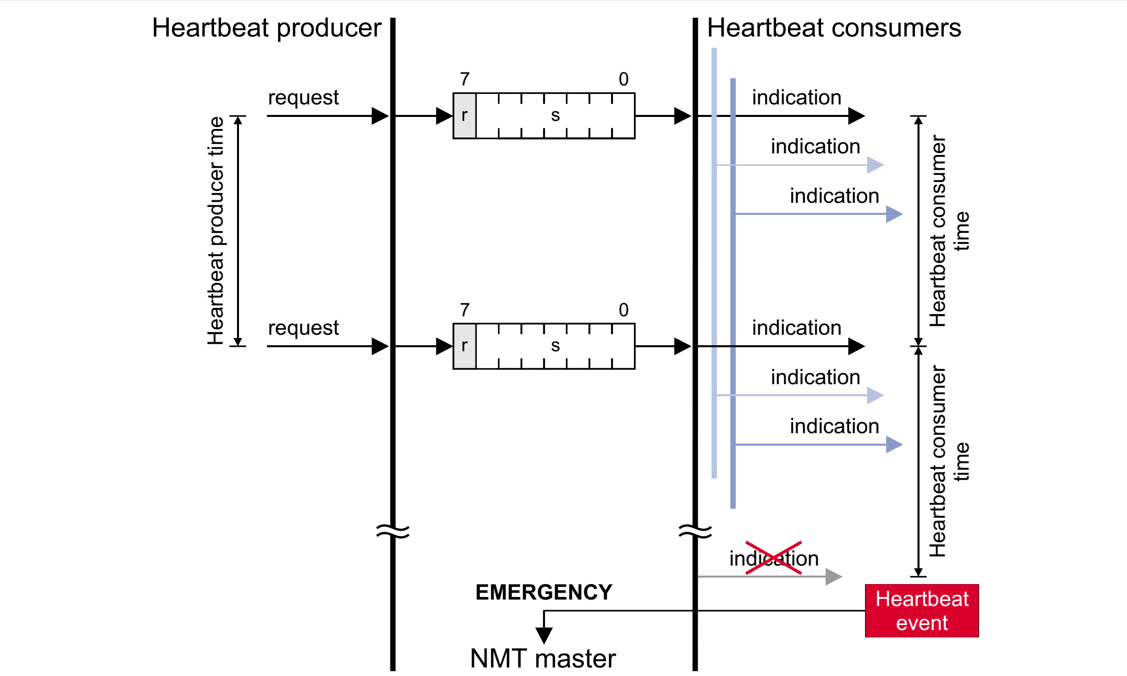
- Several nodes with identical node address
- Wrong baud rate of one or several nodes
- Too high cable length
- Too many or no terminating resistors
- Too high bus load/too many data telegrams
  (e.g. since a node permanently transmits event-controlled due to data changes of an analog signal/actual value.)
- EMC interferences on the CAN bus
  (e.g. since the CAN bus cable next to the motor cable is unshielded.)

C00364 displays whether such an error is active.

### 11.10.2 Heartbeat protocol

The heartbeat protocol can be used for node monitoring purposes within a CAN network.

#### Basic workflow




r: Reserved (always 0)
s: Status of the producer (0: Boot-up, 4: Stopped, 5: Operational, 127: Pre-Operational)

[11-9] Heartbeat protocol

1. A heartbeat producer cyclically transmits a so-called heartbeat telegram to one or more consumers.
2. The consumer(s) monitor the heartbeat telegram for arrival on a regular basis.

### 11.10.2.1 Telegram structure

- The heartbeat telegram of the producer has the following identifier:
  Identifier (COB-ID) = 1792 + producer's node address
- The user data (1 byte) contain the status (s) of the producer:

| Heartbeat producer status | | Data | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Communication status | Decimal value (s) | (r) | Producer status (s) | | | | | | |
| | | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| Boot-up | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Stopped | 4 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| Operational | 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Pre-Operational | 127 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

### 11.10.2.2 Parameter setting

**Short overview of the parameters for the "Heartbeat" monitoring function:**

| Parameter | Info | Lenze setting | | Assignment | |
|---|---|---|---|---|---|
| | | Value | Unit | Consumer | Producer |
| C00347/1...n | CAN status<br>of the heartbeat producer 1 ... n | - | | ● | |
| C00381 | Heartbeat producer time | 0 | ms | | ● |
| C00385/1...n | CAN node address<br>of the heartbeat producer 1 ... n | 0 | | ● | |
| C00386/1...n | Heartbeat consumer time<br>for the heartbeat producer 1 ... n | 0 | ms | ● | |
| C00592/5 | Resp. to heartbeat event | No response | | ● | |
| Highlighted in grey = display parameter | | | | | |

#### Heartbeat producer time

Time interval for the transmission of the heartbeat telegram to the consumer(s).

- Parameterisable in C00381 or via object I-1017. The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is sent automatically as soon as a time > 0 ms is set.

#### Heartbeat consumer time

Monitoring time for the nodes (producers) to be monitored.

- Parameterisable in C00386/1...n or via object I-1016.
- The parameterised time is rounded down to an integer multiple of 5 ms and must have a greater value than the heartbeat producer time of the node to be monitored.

- The maximum number of the nodes to be monitored depends on the device version:
  - "BaseLine C": 1 Heartbeat Producer can be monitored.
  - "StateLine": Up to 7 Heartbeat Producers can be monitored.
  - "HighLine": Up to 15 Heartbeat Producers can be monitored.
- The node address(es) of the nodes to be monitored is/are set in C00385/1...n or via object I-1016, too.

### Heartbeat event

The "Heartbeat event" is activated in the consumer if it does not receive any heartbeat telegram from the producer within the heartbeat consumer time:

- The consumer changes from the "Operational" communication status to the "Pre-Operational" communication status.
- The NMT master receives an emergency telegram containing emergency error code 0x8130.
- The response parameterised in C00592/5 is activated (Lenze setting: "No response").

**Note!**

The heartbeat monitoring will not start until the first heartbeat telegram of a monitored producer has been received successfully and the "Pre-Operational" NMT status has been assumed.

The boot-up telegram counts as the first heartbeat telegram.

### 11.10.2.3 Commissioning example

**Task**

An 8400 controller (node 2) which is configured as heartbeat consumer is to monitor another 8400 controller (heartbeat producer, node 1).

- The heartbeat producer is to transmit a heartbeat telegram to the heartbeat consumer every 10 ms.
- The heartbeat consumer monitors the heartbeat telegram for arrival. A response is to be activated in the event of an error.

**Parameterising the heartbeat producer (node 1)**

1. Set the heartbeat producer time (C00381) to 10 ms.

**Parameterising the heartbeat consumer (node 2)**

1. Set the CAN node address of the producer in C00385/1.
2. Set the heartbeat consumer time in C00386/1.
   - Note: The heartbeat consumer time must be greater than the heartbeat producer time of the node to be monitored set in C00381.
3. Set the desired response in C00592/5 which is to be activated should a heartbeat event in the consumer occur.

**Tip!**

C00347/1...n displays the heartbeat status of the monitored nodes.

**Heartbeat telegram**

- The heartbeat telegram of the producer has the following identifier:
  Identifier (COB-ID) = 1792 + producer's node address = 1792 + 1 = 1793 = 0x701

### 11.10.3 Emergency telegram

If the error status changes because an internal device error occurs or has been eliminated, the NMT master receives an emergency telegram once with the following structure:

<table>
<tr><th>1st byte</th><th>2nd byte</th><th>3rd byte</th><th>4th byte</th><th>5th byte</th><th>6th byte</th><th>7th byte</th><th>8th byte</th></tr>
<tr><td colspan="2">Emergency error codes</td><td>Error register</td><td colspan="5">Manufacturer-specific error message</td></tr>
<tr><td rowspan="2">LOW byte</td><td rowspan="2">HIGH byte</td><td rowspan="2">I-1001</td><td rowspan="2">0x00<br>(Reserved)</td><td colspan="2">LOW word</td><td colspan="2">HIGH word</td></tr>
<tr><td>LOW byte</td><td>HIGH byte</td><td>LOW byte</td><td>HIGH byte</td></tr>
<tr><td colspan="3">See table below</td><td colspan="5">• For emergency error code 0xF000: Lenze error number (value displayed in C00168)<br>• All other emergency error codes have a value of "0" here.</td></tr>
</table>

<table>
<tr><th>Emergency error codes</th><th>Error register</th><th>Cause</th></tr>
<tr><td rowspan="2">0x0000</td><td>0xXX</td><td>One of several errors eliminated</td></tr>
<tr><td>0x00</td><td>One error has been eliminated (error-free status afterwards)</td></tr>
<tr><td>0x3100</td><td>0x01</td><td>Supply voltage of standard device faulty or failed</td></tr>
<tr><td>0x8100</td><td>0x11</td><td>Communication error (warning)</td></tr>
<tr><td>0x8130</td><td>0x11</td><td>Life guarding error or heartbeat error</td></tr>
<tr><td>0x8150</td><td>0x11</td><td>Collision of identifiers (COB-IDs): An identifier parameterised for reception is also used for transmission.</td></tr>
<tr><td>0x8210</td><td>0x11</td><td>PDO length shorter than expected</td></tr>
<tr><td>0x8220</td><td>0x11</td><td>PDO length greater than expected</td></tr>
<tr><td>0x8700</td><td>0x11</td><td>Monitoring of the sync telegram</td></tr>
<tr><td>0xF000</td><td>0x01</td><td>Generic error<br>• An error with a "Fault", "Trouble", "TroubleQSP", "Warning", or "SystemFault" error response occurred in the standard device.<br>• Error message is the Lenze error number (C00168).</td></tr>
</table>

The Short overview (A-Z) of error messages of the operating system includes a list of more emergency error codes. (🕮 608)

#### Example

<table>
<tr><th>1st byte</th><th>2nd byte</th><th>3rd byte</th><th>4th byte</th><th>5th byte</th><th>6th byte</th><th>7th byte</th><th>8th byte</th></tr>
<tr><td colspan="2">Emergency error codes</td><td>Error register</td><td colspan="5">Manufacturer-specific error message</td></tr>
<tr><td>0x00</td><td>0xF0</td><td rowspan="2">0x01</td><td rowspan="2">0x00<br>(Reserved)</td><td colspan="4">Lenze error number</td></tr>
<tr><td colspan="2">Generic error</td><td colspan="4">▸ Error messages of the operating system<br>Corresponding error-free message: Value "0x00000000"</td></tr>
</table>

**Tip!**

A detalied description can be found in CAN specification DS301, V4.02.

## 11.11 Implemented CANopen objects

Lenze devices can both be parameterised with Lenze codes and manufacturer-independent "CANopen objects". A completely CANopen-compliant communication can only be achieved by using CANopen objects for parameter setting. The CANopen objects described in this chapter are defined in the CAN specification DS301 V4.02.

Many CANopen objects can be mapped on Lenze codes. In the following table, the corresponding Lenze codes are listed in the column "Relationship to Lenze codes".

**Note!**

Some of the terms used here derive from the CANopen protocol.

### Overview of CANopen indices and their relationship to Lenze codes

| CANopen object | | | Relationship to Lenze code |
|---|---|---|---|
| **Index** | **Subindex** | **Name** | |
| I-1000 | 0 | Device type | - |
| I-1001 | 0 | Error register | - |
| I-1003 | Predefined error field | | |
| | 0 | Number of errors | - |
| | 1 ... 10 | Standard error field | - |
| I-1005 | 0 | COB-ID SYNC message | C00367 |
| | | | C00368 |
| I-1006 | 0 | Communication cycle period | C00369 |
| I-1014 | 0 | COB-ID EMCY | - |
| I-1016 | Consumer heartbeat time | | |
| | 0 | Highest subindex supported | - |
| | 1 ... n | Consumer heartbeat time<br>• "BaseLine C" version: n = 1<br>• "StateLine" version: n = 7<br>• "HighLine" version: n = 15 | C00385/1...n<br>C00386/1...n |
| I-1017 | 0 | Producer heartbeat time | C00381 |
| I-1018 | Identity object | | |
| | 0 | Highest subindex supported | - |
| | 1 | Vendor ID | - |
| | 2 | Product code | - |
| | 3 | Revision number | - |
| | 4 | Serial number | - |
| I-1200 | SDO1 server parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID client –> server (rx) | - |
| | 2 | COB-ID server –> client (tx) | - |
| I-1201 | SDO2 server parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID client –> server (rx) | - |
| | 2 | COB-ID server –> client (tx) | - |

| CANopen object | | | Relationship to Lenze code |
|---|---|---|---|
| Index | Subindex | Name | |
| I-1400 | RPDO1 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by RPDO | C00355/1 |
| | 2 | Transmission type | C00323/1 |
| I-1401 | RPDO2 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by RPDO | C00355/3 |
| | 2 | Transmission type | C00323/2 |
| I-1402 | RPDO3 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by RPDO | C00355/5 |
| | 2 | Transmission type | C00323/3 |
| I-1600 | RPDO1 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00409/1...4<br>C00866/1...4 |
| I-1601 | RPDO2 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00409/5...8<br>C00866/5...8 |
| I-1602 | RPDO3 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00409/9...12<br>C00866/9...12 |
| I-1800 | TPDO1 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by TPDO | C00355/2 |
| | 2 | Transmission type | C00322/1 |
| | 3 | Inhibit time | C00324/2 |
| | 5 | Event timer | C00356/5<br>C00369 |
| I-1801 | TPDO2 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by TPDO | C00355/4 |
| | 2 | Transmission type | C00322/2 |
| | 3 | Inhibit time | C00324/3 |
| | 5 | Event timer | C00356/2<br>C00369 |
| I-1802 | TPDO3 communication parameter | | |
| | 0 | Highest subindex supported | - |
| | 1 | COB-ID used by TPDO | C00355/6 |
| | 2 | Transmission type | C00322/3 |
| | 3 | Inhibit time | C00324/4 |
| | 5 | Event timer | C00356/3<br>C00369 |

| CANopen object | | | Relationship to Lenze code |
|---|---|---|---|
| Index | Subindex | Name | |
| I-1A00 | TPDO1 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00868/1...4 |
| I-1A01 | TPDO2 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00868/5...8 |
| I-1A02 | TPDO3 mapping parameter | | |
| | 0 | Number of mapped application objects in PDO | - |
| | 1 ... 4 | Application object 1 ... 4 | C00868/9...12 |

**I-1000**

| Index<br>**I-1000** | Name:<br>**Device type** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Display range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Device type | 0 | 0 | | 4294967295 | ro | U32 |

The CANopen index I-1000 specifies the profile for this device. Furthermore, additional information defined in the device profile itself can be stored here.

| 8th byte | 7th byte | 6th byte | 5th byte |
|---|---|---|---|
| **Data 4** | **Data 3** | **Data 2** | **Data 1** |
| HIGH word | | LOW word | |
| HIGH byte | LOW byte | HIGH byte | LOW byte |
| **Additional information** | | **Device profile number** | |

[11-1] Data telegram assignment

In case of 8400 series controllers, the four bytes contain the following values:

- 5th and 6th byte: The data content is 0x0000, i.e. no profile definition.
- 7th byte: The data content specifies the device type: Here the value is 0x00 for controllers.
- 8th byte: The data content is 0x00.

The data content for the 8400 controller thus is: 00 00 00 00

**I-1001**

| Index:<br>**I-1001** | Name:<br>**Error register** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Display range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Error register | - | 0 | | 255 | ro | U8 |

Error register

The error status in the data byte (U8) is bit coded. The following error states are coded in the data byte (U8):

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Error status |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | No error |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Device error message |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Communication error |

**I-1003**

| Index: **I-1003** | Name: **Predefined error field** | | | | |
|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of errors | 0 | 0 | | 255 | rw | U8 |
| 1 ... 10: Standard error field | - | 0 | | 4294967295 | ro | U32 |

Error history

This object indicates that an error has occurred in the module and in the standard device.

| **Subindex** | **Meaning** |
|---|---|
| 0 | Number of saved error messages |
| 1 ... 10 | Display of the error list<br>The error messages (U32) consist of a 16-bit error code and a manufacturer-specific information field comprising 16 bits. |

**Note!**

The values in the "standard error field" under subindex 1 ... 10 will be deleted if the subindex "number of recorded errors" is overwritten with the value "0".

| **Emergency error codes** | **Cause** | **Entry in the error register (I-1001)** |
|---|---|---|
| 0x0000 | One of several errors eliminated | 0xXX |
| | Elimination of one single error<br>(afterwards no more errors) | 0x00 |
| 0x1000 | Standard device is in error status (error response "fault", "message", "warning", "error", "quick stop by trouble", or "system error") | 0x01 |
| 0x3100 | Supply voltage of standard device faulty or failed | 0x01 |
| 0x8100 | Communication error (warning) | 0x11 |
| 0x8130 | Life guard error or heartbeat error | 0x11 |
| 0x8150 | Collision of COB-IDs: An ID parameterised for reception is also used for transmission. | 0x11 |
| 0x8210 | PDO length shorter than expected | 0x11 |
| 0x8220 | PDO length greater than expected | 0x11 |
| 0x8700 | Monitoring of the sync telegram | 0x11 |

**I-1005**

| Index: **I-1005** | Name: **COB-ID SYNC message** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: COB-ID SYNC message | 0x0000 0080<br>or<br>0x8000 0080 | 0 | | 4294967295 | rw | U32 |

This object can be used to activate the generation of sync telegrams and to write the identifier value.

- This object relates to codes C00367 and C00368.

### Creating sync telegrams

Sync telegrams are created by setting bit 30 (see below) to "1". The time between the sync telegrams can be set using the object I-1006.

### Writing identifiers

To receive PDOs, the value 0x80 must be entered in the 11-bit identifier in the Lenze setting (and according to CANopen specification) . This means that all modules are by default set to the same sync telegram.

- If sync telegrams are only to be received by certain communication modules, their identifiers can be entered with values up to and including 0x07FF.
- The identifier can only be changed if the communication module does not send any sync telegrams (bit 30 = "0").
- How to change the identifier:
  - Deactivate identifier (set bit 30 to "0").
  - Change identifier.
  - Activate identifier (set bit 30 to "1").

| 8th byte | | 7th byte | 6th byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **X** | **0/1** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[11-2] Data telegram assignment

**I-1006**

| Index: **I-1006** | Name: **Communication cycle period** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Communication cycle period | 0 µs | 0 | µs | 65535000 | rw | U32 |

Setting the sync telegram cycle time.

- The cycle time can be selected as "1000" or as an integer multiple of it.
- If "0 µs" is set (Lenze setting), no sync telegrams are created.
- This object relates to code C00369.

**I-1014**

| Index: **I-1014** | Name: **COB-ID EMCY** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: COB-ID EMCY | 0x80 + node ID | 0 | | 4294967295 | rw | U32 |

When communication errors occur and are acknowledged or when internal errors occur in the communication module or controller (e.g. "fault"), the system bus sends an error message. The telegram is sent once for every error. This function can be activated or deactivated with bit 31.

| 8th byte | | 7th byte | 6th byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **0/1** | **0** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[11-3] Data telegram assignment

| Bit | Setting | |
|---|---|---|
| Bit 31 | 0 | Emergency object is valid. |
| | 1 | Emergency object is invalid. |

**Note!**

The identifier can only be changed in the "emergency object invalid" status (bit 31 = 1).

### I-1016

| Index: **I-1016** | Name: **Consumer heartbeat time** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 1 (for BaseLine)<br>7 (for StateLine)<br>15 (for HighLine) | - (read access only) | | | ro | U16 |
| 1 … n: Consumer heartbeat time | 0 | 0 | | 65535 | rw | U16 |

Monitoring time for the nodes to be monitored via heartbeat. Heartbeat protocol (682)

- The parameterised time is rounded down to an integer multiple of 5 ms and must have a greater value than the heartbeat producer time of the node to be monitored.

| Subindex | Meaning | Lenze code |
|---|---|---|
| 0 | Number of nodes to be monitored | |
| 1 … n | Node ID and heartbeat time of the node to be monitored | Node ID:<br>C00385/x<br>Heartbeat time:<br>C00386/x |

| 8th byte | 7th byte | 6th byte | 5th byte |
|---|---|---|---|
| Data 4 | Data 3 | Data 2 | Data 1 |
| Bit 31 ... bit 24 | Bit 23 ... Bit 16 | Bit 15 ... Bit 0 | |
| **0**<br>(Reserved) | **Node ID** | **Heartbeat time**<br>in [ms] | |

[11-4] Data telegram assignment

**I-1017**

| Index:<br>**I-1017** | Name:<br>**Producer heartbeat time** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Producer heartbeat time | 0 | 0 | ms | 65535 | rw | U16 |

Time interval for sending the heartbeat telegram to the consumer(s). ▶ Heartbeat protocol (📖 682)

- The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is automatically sent as soon as a time > 0 ms is entered. In this case, the "node guarding" monitoring function is deactivated.
- This object relates to code C00381.

**I-1018**

| Index:<br>**I-1018** | Name:<br>**Identity object** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Display range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | see below | 0 | | 4294967295 | ro | U32 |
| 1: Vendor ID | | | | | | |
| 2: Product code | | | | | | |
| 3: Revision number | | | | | | |
| 4: Serial number | | | | | | |

| Subindex | Meaning | |
|---|---|---|
| 1 | Manufacturer's identification number<br>• The identification number allocated to Lenze by the organisation "CAN in Automation e. V." is "0x0000003B". | |
| 2 | Product code | |
| | 0x84001 | 8400 BaseLine C |
| | 0x84002 | 8400 StateLine C |
| | 0x84003 | 8400 HighLine C |
| 3 | Main and subversion of firmware | |
| 4 | Serial number | |

**I-1200**

| Index: **I-1200** | Name: **SDO1 server parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Display range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 2 | 2 | | 2 | ro | U8 |
| 1: COB-ID client -> server (rx) | node ID + 0x600 | 0 | | 4294967295 | ro | U32 |
| 2: COB-ID server -> client (tx) | node ID + 0x580 | 0 | | 4294967295 | ro | U32 |

Identifiers for SDO server channel 1 (basic SDO channel).

▶ According to DS301 V4.02, the basic SDO channel can neither be changed nor deactivated.

| Subindex | Meaning |
|---|---|
| 1 | Specification of receive identifier<br>• For SDO server channel 1: node address (C00350) + 0x600 |
| 2 | Specification of send identifier<br>• For SDO server channel 1: node address (C00350) + 0x580 |

| 8th byte | | 7th byte | 6th byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **0** | **0** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[11-5] Data telegram assignment

**I-1201**

| Index:<br>**I-1201** | Name:<br>**SDO2 server parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 3 | - (read access only) | | | ro | U8 |
| 1: COB-ID client -> server (rx) | 0x80000000 | 0 | | 4294967295 | rw | U32 |
| 2: COB-ID server -> client (tx) | 0x80000000 | 0 | | 4294967295 | rw | U32 |

Identifiers for SDO server channel 2.

- The SDO server parameter is only valid, if bit 31 is set to "0" for both transmission directions (subindex 1 and 2).
- In the Lenze setting, the SDO server channels 2 are deactivated (bit 31 = "1").
- The identifier can only be changed if the SDO is invalid (bit 31 = "1").

| Subindex | Meaning |
|---|---|
| 1 | Specification of receive identifier |
| 2 | Specification of send identifier |

<table>
<tr><th colspan="3">8th byte</th><th>7th byte</th><th>6th byte</th><th>5th byte</th></tr>
<tr><th colspan="3">Data 4</th><th>Data 3</th><th>Data 2</th><th>Data 1</th></tr>
<tr><td>Bit 31</td><td>Bit 30</td><td colspan="3">Bit 29 ... bit 11</td><td>Bit 10 ... bit 0</td></tr>
<tr><td>0/1</td><td>0</td><td colspan="3">Extended identifier*</td><td>11-bit identifier</td></tr>
<tr><td colspan="6">* The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0".</td></tr>
</table>

[11-6] Data telegram assignment

| Bit | Setting | |
|---|---|---|
| Bit 31 | 0 | SDO is valid. |
| | 1 | SDO is invalid. |

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**Example**

Parameter data channel 2 of the controller with node address 4 shall be activated.

- For this, bit 31 must be set to "0" (≡ "SDO is valid") in subindices 1 and 2 of the object I-1201.
- The master must send the two "write request" commands to the nodes via the basic SDO channel.

### Identifier calculation

- Identifier (COB-ID) = basic identifier + node address (node ID)
- Basic identifier SDO2 from master to drive: 1600 (0x640)
  ➞ Identifier = 0x640 + 0x4 = 0x644
- Basic identifier SDO2 from drive to master: 1472 (0x5C0)
  ➞ Identifier = 0x5C0 + 0x4 = 0x5C4

### Resulting data (data 1 ... data 4)

| 8th byte | | 7th byte | 6th byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| 0 | 0 | Extended identifier = 0 | | 11-bit identifier = 0x644 |
| **0x00** | | **0x00** | **0x06** | **0x44** |

[11-7] Data telegram assignment for subindex 1

| 8th byte | | 7th byte | 6th byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| 0 | 0 | Extended identifier = 0 | | 11-bit identifier = 0x5C4 |
| **0x00** | | **0x00** | **0x05** | **0xC4** |

[11-8] Data telegram assignment for subindex 2

### User data assignment

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x23 | 0x01 | 0x12 | 0x01 | 0x44 | 0x06 | 0x00 | 0x00 |

[11-9] User data assignment for writing to subindex 1

| 1st byte | 2nd byte | 3rd byte | 4th byte | 5th byte | 6th byte | 7th byte | 8th byte |
|---|---|---|---|---|---|---|---|
| Command | Index | | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 0x23 | 0x01 | 0x12 | 0x02 | 0xC4 | 0x05 | 0x00 | 0x00 |

[11-10] User data assignment for writing to subindex 2

### I-1400

| Index: **I-1400** | Name: **RPDO1 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by RPDO | 0x200 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | - | - (not used for RPDOs) | | | rw | U16 |
| 4: Compatibility entry | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | - | - (not used for RPDOs) | | | rw | U16 |

Communication parameter for receiving process data via RPDO1

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | RPDO1 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x200 + node ID | C00354/1 |
| 2 | RPDO Transmission type according to DS301 V4.02<br>▸ Transmission type (📖 662) | C00323/1 |

| 8th byte | | 7th byte | 6th byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **0/1** | **0/1** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[11-11] Data telegram assignment

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

Description of subindex 1

| Bit no. | Value | Explanation |
|---|---|---|
| 0 ... 10 | 0/1 | 11-bit identifier |
| (11 ... 28)* | 0 | *) The extended identifier (29 bits) is not supported. Any of these bits must be "0". |
| 29* | 0 | |
| 30 | 0 | RTR to this PDO possible (cannot be set) |
| | 1 | RTR to this PDO not possible (Lenze) |
| 31 | 0 | PDO active |
| | 1 | PDO not active |

[11-12] I-1400 ... I-1402, subindex 1

Description of subindex 2

| PDO transmission | | | Transmission type | Explanation |
|---|---|---|---|---|
| cyclic | synchronous | event-controlled | | |
| X | X | | n = 1 ... 240 | When a value n is entered, this PDO will be accepted with every nth SYNC. |
| | | X | n = 254 | PDO will be accepted immediately. |

[11-13] I-1400 ... I-1402, subindex 2

**I-1401**

| Index: **I-1401** | Name: **RPDO2 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by RPDO | 0x300 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | - | - (not used for RPDOs) | | | rw | U16 |
| 4: Compatibility entry | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | - | - (not used for RPDOs) | | | rw | U16 |

Communication parameter for receiving process data via RPDO2

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | RPDO2 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x300 + node ID | C00354/3 |
| 2 | RPDO Transmission type according to DS301 V4.02<br>▸ Transmission type (📖 662) | C00323/2 |

▶ For data telegram assignment and description of subindices 1 and 2, see object I-1400.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1402**

| Index: **I-1402** | Name: **RPDO3 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by RPDO | 0x400 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | - | - (not used for RPDOs) | | | rw | U16 |
| 4: Compatibility entry | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | - | - (not used for RPDOs) | | | rw | U16 |

Communication parameter for receiving process data via RPDO3

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | RPDO3 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x400 + node ID | C00354/5 |
| 2 | RPDO Transmission type according to DS301 V4.02<br>▸ Transmission type (662) | C00323/3 |

▶ For data telegram assignment and description of subindices 1 and 2, see object I-1400.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1600**

| Index:<br>**I-1600** | Name:<br>**RPDO1 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1600 serves to receive parameter data as RPDO1.

▶ This object relates to codes C00409/1...4 and C00866/1...4.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for RPDO1<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

| 8th byte | 7th byte | 6th byte | 5th byte |
|---|---|---|---|
| **Data 4** | **Data 3** | **Data 2** | **Data 1** |
| Bit 31 ... bit 16 | | Bit 15 ... bit 8 | Bit 7 ... bit 0 |
| **Index** | | **Subindex** | **Length** |

[11-14] Data telegram assignment

IEC 61131 process data words are mapped. Only whole bytes can be mapped (1-byte/mapping entry).

**Related topics:**

▶ RPDO1 | Port block "LP_CanIn1" (📖 655)

**I-1601**

| Index:<br>**I-1601** | Name:<br>**RPDO2 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1601 serves to receive parameter data as RPDO2.

▶ This object relates to codes C00409/5...8 and C00866/5...8.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for RPDO2<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

▶ For data telegram assignment, see object I-1600.

**Related topics:**

▶ RPDO2 | Port block "LP_CanIn2" (📖 656)

**I-1602**

| Index: **I-1602** | Name: **RPDO3 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1602 serves to receive parameter data as RPDO3.

▶ This object relates to codes C00409/9...12 and C00866/9...12.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for RPDO3<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

▶ For data telegram assignment, see object I-1600.

**Related topics:**

▸ RPDO3 | Port block "LP_CanIn3" (📖 657)

**I-1800**

| Index: **I-1800** | Name: **TPDO1 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by TPDO | 0x180 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | 0 ms | 0 | 0.1 ms | 65535 | rw | U16 |
| 4: Reserved | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | 0 ms | 0 | ms | 65535 | rw | U16 |

Communication parameter for sending process data via TPDO1

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | TPDO1 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x180 + node ID | C00354/2 |
| 2 | TPDO transmission type according to DS301 V4.02<br>▸ Transmission type (📖 662) | C00322/1 |
| 3 | Minimum time between sending two identical TPDOs (see DS301 V4.02). | C00324/2 |
| 5 | Cycle time for PDO transmission with transmission type "254". | C00356/5<br>C00369 |

| 8th byte | | 7th byte | 6th byte | 5th byte |
|---|---|---|---|---|
| Data 4 | | Data 3 | Data 2 | Data 1 |
| Bit 31 | Bit 30 | Bit 29 ... bit 11 | | Bit 10 ... bit 0 |
| **0/1** | **0/1** | **Extended identifier*** | | **11-bit identifier** |
| * The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0". | | | | |

[11-15] Data telegram assignment

| Bit | Setting | |
|---|---|---|
| Bit 30 | 0 | RTR to this PDO possible (Lenze). |
| | 1 | RTR to this PDO not possible (not adjustable) |
| Bit 31 | 0 | PDO active |
| | 1 | PDO inactive |

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

### Subindex 2 - transmission type

| PDO transmission | | | Transmission type | Explanation |
|---|---|---|---|---|
| cyclic | synchronous | event-controlled | | |
| ● | ● | | n = 1 ... 240 | When a value n is entered, this PDO will be accepted with every nth SYNC. |
| | ● | | n = 252 | On sync, the PDO is filled with new data, but only sent on RTR. |
| | | ● | n = 254, 255 | Event-controlled or cyclic |

### Subindex 3 - inhibit time

**Note!**

The delay time can only be changed when the PDO is not active (see subindex 1, bit 31 = 1).

The entered value multiplied by 0.1 gives the delay time in [ms]. Only integers will be considered, i.e. fractional numbers will be **rounded down** to integers.

Example:

- Entered value: 26
- Calculated time = 26 x 0.1 [ms] = 2.6 [ms] → delay time = 2 [ms]

### Subindex 5 - event timer

For cyclic operation (transmission type 254), the cycle time for sending the process data object on the CAN bus can be set under subindex 5:

The entered value corresponds to the time in [ms].

**I-1801**

| Index: **I-1801** | Name: **TPDO2 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by TPDO | 0x280 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | 0 ms | 0 | 0.1 ms | 65535 | rw | U16 |
| 4: Reserved | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | 0 ms | 0 | ms | 65535 | rw | U16 |

Communication parameter for sending process data via TPDO2

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | TPDO2 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x280 + node ID | C00354/4 |
| 2 | TPDO transmission type according to DS301 V4.02<br>▸ Transmission type (📖 662) | C00322/2 |
| 3 | Minimum time between sending two identical TPDOs (see DS301 V4.02). | C00324/3 |
| 5 | Cycle time for PDO transmission with transmission type "254". | C00356/2<br>C00369 |

▶ For data telegram assignment and description of subindices, see object I-1800.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1802**

| Index: **I-1802** | Name: **TPDO3 communication parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Highest subindex supported | 5 | - (read access only) | | | ro | U8 |
| 1: COB-ID used by TPDO | 0x380 + node ID | 0 | | 4294967295 | rw | U32 |
| 2: Transmission type | 254 | 0 | | 255 | rw | U8 |
| 3: Inhibit time | 0 ms | 0 | 0.1 ms | 65535 | rw | U16 |
| 4: Reserved | - | - (reserved, read or write access leads to error message 0x06090011) | | | rw | U8 |
| 5: Event timer | 0 ms | 0 | ms | 65535 | rw | U16 |

Communication parameter for sending process data via TPDO3

| Subindex | Meaning | Code |
|---|---|---|
| 0 | "5" is permanently set.<br>• Max. 5 subindices are supported. | - |
| 1 | TPDO3 identifier<br>• According to the "Predefined Connection Set", the basic setting is: Identifier = 0x380 + node ID | C00354/6 |
| 2 | TPDO transmission type according to DS301 V4.02<br>▸ Transmission type (📖 662) | C00322/3 |
| 3 | Minimum time between sending two identical TPDOs (see DS301 V4.02). | C00324/4 |
| 5 | Cycle time for PDO transmission with transmission type "254". | C00356/3<br>C00369 |

▶ For data telegram assignment and description of subindices, see object I-1800.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

**I-1A00**

| Index: **I-1A00** | Name: **TPDO1 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1A00 serves to send parameter data as TPDO1.

▶ This object relates to code C00868/1...4.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for TPDO1<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

| 8th byte | 7th byte | 6th byte | 5th byte |
|---|---|---|---|
| **Data 4** | **Data 3** | **Data 2** | **Data 1** |
| Bit 31 ... bit 16 | | Bit 15 ... bit 8 | Bit 7 ... bit 0 |
| **Index** | | **Subindex** | **Length** |

[11-16] Data telegram assignment

IEC 61131 process data words are mapped. Only whole bytes can be mapped (1-byte/mapping entry).

**Related topics:**

▸ TPDO1 | Port block "LP_CanOut1" (658)

**I-1A01**

| Index: **I-1A01** | Name: **TPDO2 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1A01 serves to send parameter data as TPDO2.

▶ This object relates to code C00868/5...8.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for TPDO2<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

▶ For data telegram assignment, see object I-1A00.

**Related topics:**

▸ TPDO2 | Port block "LP_CanOut2" (659)

**I-1A02**

| Index: **I-1A02** | Name: **TPDO3 mapping parameter** | | | | | |
|---|---|---|---|---|---|---|
| Subindex | Default setting | Setting range (min. value \| unit \| max. value) | | | Access | Data type |
| 0: Number of mapped application objects in PDO | 0 | 0 | | 8 | rw | U8 |
| 1 ... 4: Application object 1 ... 4 | 0 | 0 | | 4294967295 | rw | U32 |

The object I-1A02 serves to send parameter data as TPDO3.

▶ This object relates to code C00868/9...12.

| Subindex | Meaning |
|---|---|
| 0 | Number of mapped objects |
| 1 ... 4 | Mapping entries 1 ... 4 for TPDO3<br>• The 4th mapping entry is used for the statistic mapping. For this, there is no value available. |

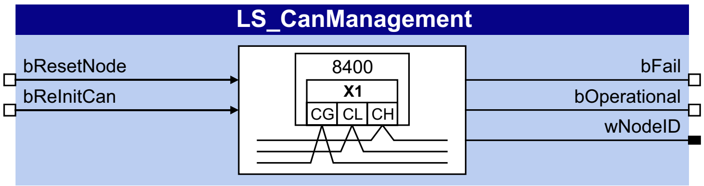
▶ For data telegram assignment, see object I-1A00.

**Related topics:**

▸ TPDO3 | Port block "LP_CanOut3" (📖 660)

## 11.12 Internal interfaces | System block "LS_CANManagement"

The **LS_CANManagement** system block serves to control internal functions of the CAN driver (reset node and re-initialisation) and to display the "Operational" status as well as the node address (analogous to the 9300 ServoPLC and ECS devices).




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bResetNode<br>BOOL | Reset node | |
| | TRUE | Carry out reset node<br>• If the controller is configured as CAN master in C00352, the NMT command "Start Remote Node" is sent to all nodes at the bus (broadcast telegram). ▸ Network management telegram (NMT) |
| bReInitCAN<br>BOOL | Reinitialisation | |
| | TRUE | Reinitialise "CAN on board" interface. |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bFail<br>BOOL | Error | |
| | TRUE | An event according to the error configuration in C00341 has occurred |
| bOperational<br>BOOL | "Operational" status signal | |
| | TRUE | The system bus is in the "Operational" status |
| wNodeID<br>WORD | Output of the node address | |

**Note!**

If a "Bus off" error is detected, the "CAN on board" interface will automatically be reinitialised after 1 second.

Hence, 1 second after the "Bus off" has occurred, the controller will automatically be active again on the CAN bus ("Auto bus off recovery").

# 12 Fieldbus interface

The drive controllers of the 8400 series can accommodate plug-in communication modules and can therefore take part in the data transfer of an existing fieldbus system.

When using a communication module, the major advantage for the user is the possibility of parameterising, controlling, and diagnosing the drive system via the available fieldbus.

The following fieldbuses are supported by the 8400 TopLine controller:

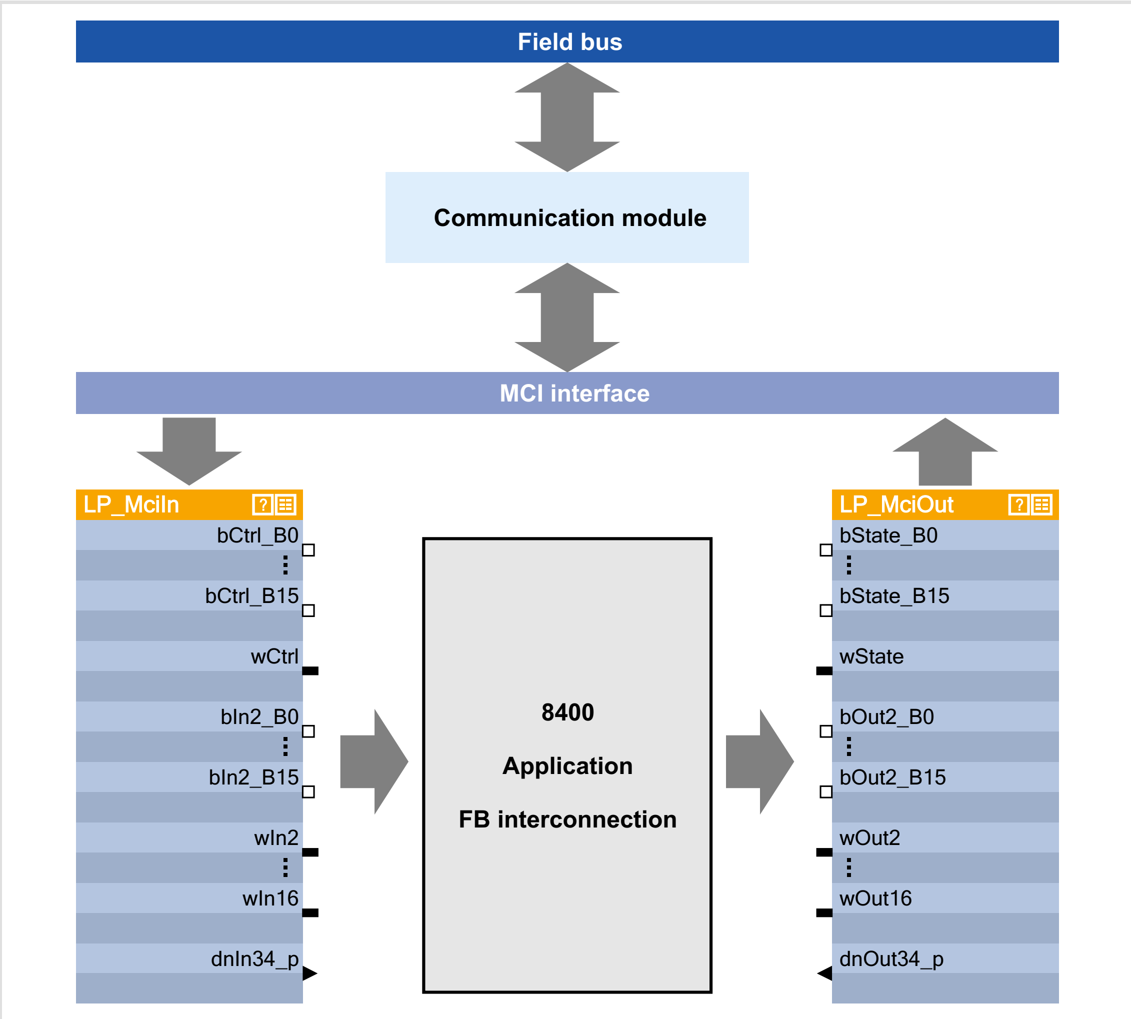
- EtherCAT
- Ethernet POWERLINK
- INTERBUS
- PROFIBUS
- PROFINET

Detailed information is provided in the communication manual (KHB) for the respective fieldbus and in the »Engineer« online help.

## 12.1 Process data transfer

The process data serve to control the controller. Thus the transfer of the process data is time-critical.

- The process cycle is 1 ms, irrespective of the respectively plugged-in bus system and the type of drive controller.
- Process data transfer takes place cyclically between the master system and the drive controllers.
  - This concerns the continuous exchange of current input and output data.
  - In the case of the 8400 drive controller, 16 words per direction are exchanged.
- The master computer can directly access the process data.
  Access to the process data takes place via the port blocks **LP_MciIn** and **LP_MciOut** (see FB interconnection of the »Engineer«).
  These port blocks are also called process data channels.
- The process data are not saved in the controller.




[12-1] External and internal data transfer between bus system, drive controller and function block interconnection

### Voltage supply

Depending on the complexity and functional range of the fieldbus, the communication modules are supplied by the standard device or an external 24 supply at the module.

The external 24 V voltage supply of the communication module is required if the supply of the standard device fails but the communication via the bus is to continue.

### Parameter setting of the communication modules

All codes which must be parameterised for establishing the fieldbus communication are saved in the memory module of the controller.

The archived data can be addressed by all bus systems supported by the controller.

### Hotplug

The communication module (MCI module) can be plugged in/out while the controller is switched on. When the module is plugged in, it is automatically detected and checked for plausibility regarding the function and version.

### Fieldbus-specific device profiles and PDO mapping

When specific bus systems are used, the controller is to behave according to a defined, manufacturer-spanning standard. The following definitions have been made for this:

- Definitions of the device state machine (e.g. DSP402, DriveCOM, ProfiDrive etc.)
- Definition of the bit assignment of control and status words
- Definition of signal scaling (on a limited scale)
- Definition of parameter scaling (on a limited scale)
- Definition of the process data mapping

These device profiles are not mapped in the communication module since some definitions have a strong effect on the device-internal behaviour and the device profiles are not uniform regarding this matter.

- The task of the communication modules is
  - to address parameters (SDOs),
  - to transfer PDOs and
  - the signal mapping of the PDOs.
- The process data objects (e.g. the meaning of the control word bits or the speed setpoint stipulated) are interpreted in the drive controller.

## 12.2 Control mode "MCI"

"40: MCI" can be selected as a control mode in C00007 in order to quickly and easily set-up drive controller control by means of MCI-PDOs via the fieldbus interface.

Given that the technology applications are fundamentally different and have different requirements regarding the signals sent to them, predefined assignment of the MCI-PDOs depends on the technology application selected in C00005:

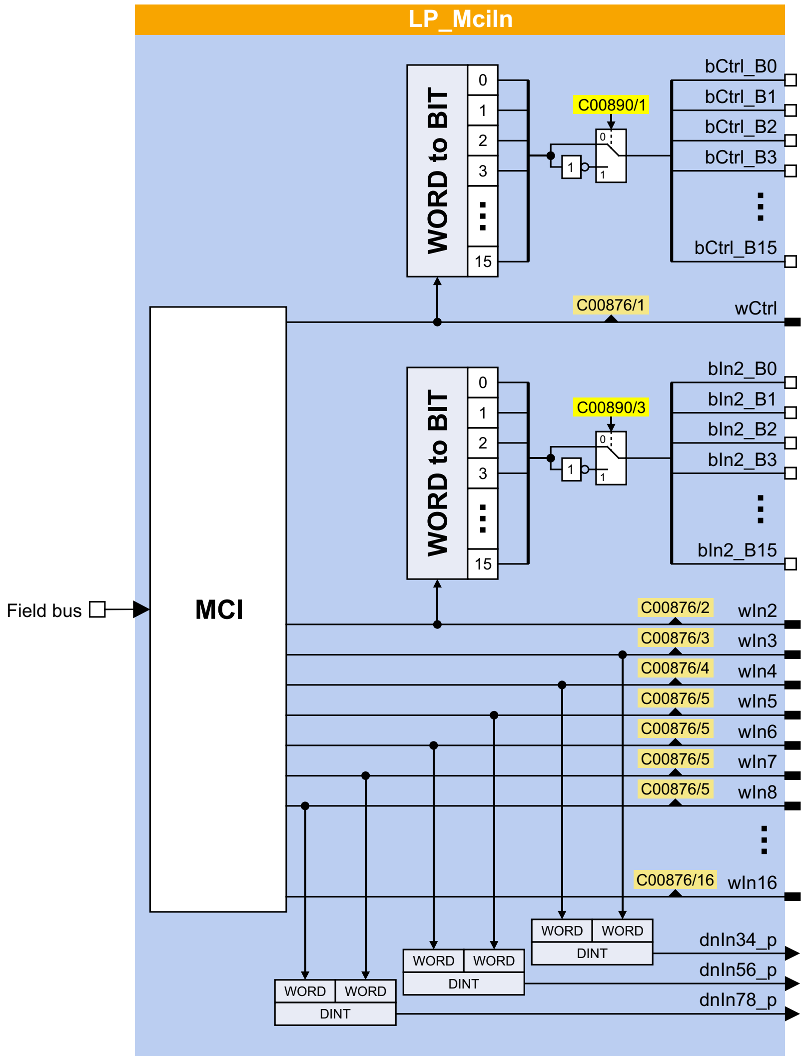
- **TA "Actuating drive speed":**
  Process data assignment for fieldbus communication (📖 403)
- **TA "Table positioning":**
  Process data assignment for fieldbus communication (📖 434)
- **TA "Switch-off positioning":**
  Process data assignment for fieldbus communication (📖 460)

**Tip!**

The predefined assignment of the MCI-PDOs can be parameterised by means of PDO mapping and can be freely configured on the I/O level in the function block editor (FB editor).

## 12.2.1 Port block "LP_MciIn"

The **LP_MciIn** port block maps the received MCI-PDOs in the FB Editor.



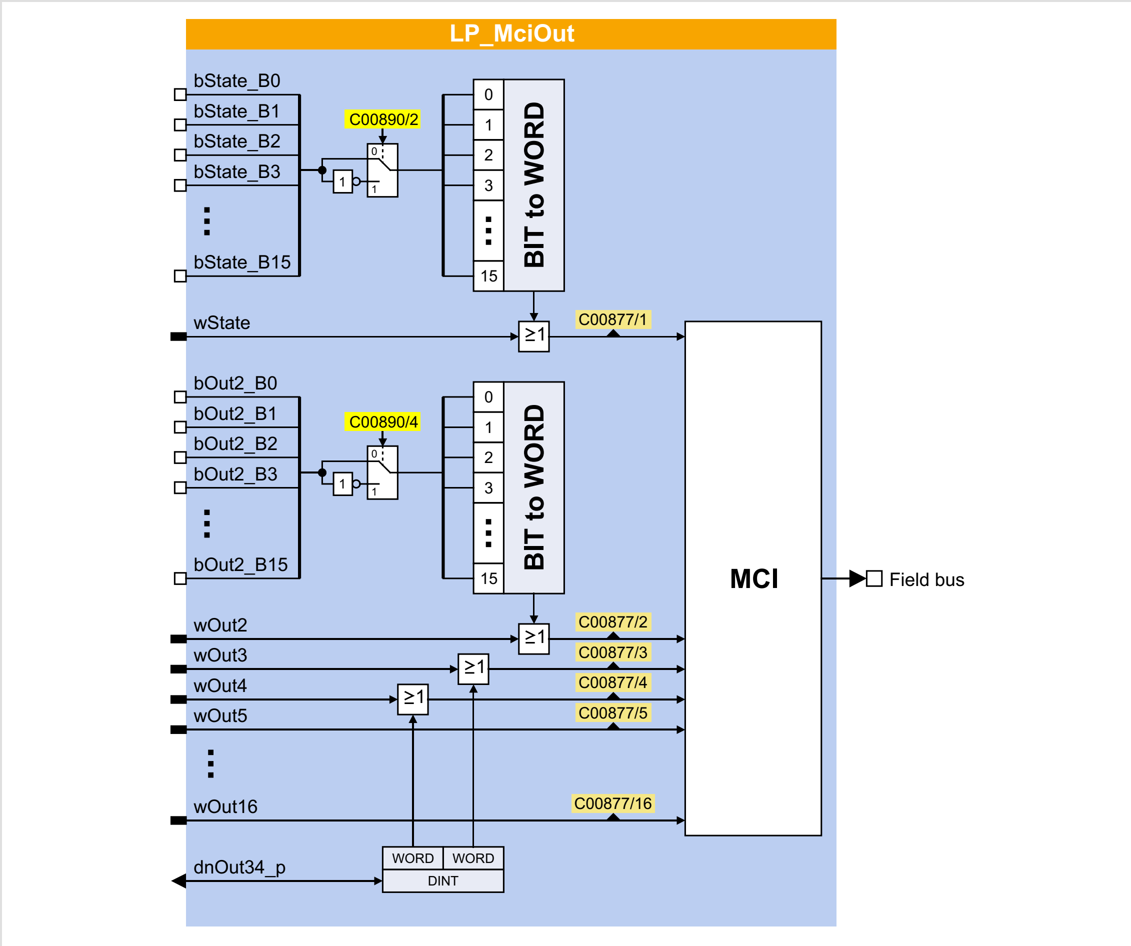

Outputs *dnIn56_p* and *dnIn78_p* are available from version 02.00.00!

### Short overview of the parameters for LP_MciIn:

| Parameter | Info | Lenze setting |
|---|---|---|
| C00876/1 | LP_MciIn:wCtrl | - |
| C00876/2...16 | LP_MciIn: wIn2 ... wIn16 | - |
| C00890/1 | LP_MciIn: Inversion bCtrl_B0..15 | 0x0000 |
| C00890/3 | LP_MciIn: Inversion bIn2_B0..15 | 0x0000 |
| Highlighted in grey = display parameter | | |

### 12.2.2 Port block "LP_MciOut"

The **LP_MciOut** port block maps the MCI-PDOs to be transmitted in the FB Editor.




#### Short overview of the parameters for LP_MciOut:

| Parameter | Info | Lenze setting |
|---|---|---|
| C00877/1 | LP_MciOut:wState | - |
| C00877/2...16 | LP_MciOut: wOut2 ... wOut16 | - |
| C00890/2 | LP_MciOut: Inversion bState_B0..15 | 0x0000 |
| C00890/4 | LP_MciOut: Inversion bOut2_B0..15 | 0x0000 |
| Highlighted in grey = display parameter | | |

# 13 Axis bus

This function extension is available from version 02.00.00!

The 8400 TopLine controller has an integrated axis bus which is used to couple several 8400 TopLine controllers in an axis interconnection. The main task of the axis bus is a simple cross-data exchange from axis to axis.

Axis bus operation uses two transmission media, "data transfer axis bus" and "IO axis bus" that can be used independently of each other:

- The data transfer axis bus is based on CAN physics and enables a high-performance data transfer of master values and control signals to other 8400 TopLine controllers.
- The IO axis bus is a 1-wire bus with open-collector circuitry (5 V isolated). The IO axis bus can either be used to transfer controller errors in the interconnection ("release cord" principle) or as a pure open-collector IO function. In the first case, the internal time base of the controllers can be synchronised via the IO axis bus.

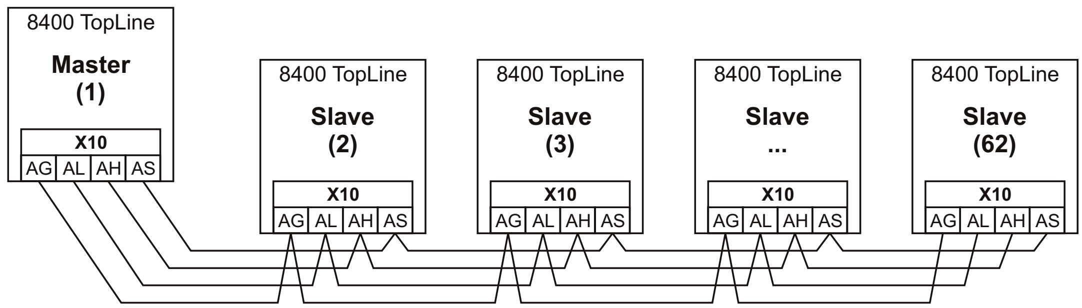
Max. 62 controllers can be connected to the axis bus.

STOP

**Stop!**

The data transfer axis bus of the 8400 TopLine controller is designed especially for performance and simplicity. HMIs and other peripherals as well as the L-force »Engineer« are not supported at the axis bus.

The IO axis bus of the 8400 TopLine controller is not compatible to the state bus of the 9300/9400 device series due to different voltage levels!




| | |
|---|---|
| X10/AL, AH | Data transfer axis bus |
| X10/AS | IO axis bus |
| X10/AG | GND, reference potential of axis bus |

[13-1] Axis bus topology

## 13.1 Data transfer axis bus

**Stop!**

The data transfer axis bus of the 8400 TopLine controller is designed especially for performance and simplicity. HMIs and other peripherals as well as the L-force »Engineer« are not supported at the axis bus.

The data transfer axis bus is based on isolated CAN physics.

- Max. 62 nodes are supported at the bus (1 master and max. 61 slaves).
- The baud rate is permanently set to 500 kbits.
- Transfer cycle = 1 ms (at baud rate = 500 kbits)

### 13.1.1 Topologies

There are generally two different topologies for an interconnected operation:

A. All nodes of the network are connected to the higher-level master control via MCI module and the respective fieldbus.
   - The axes can exchange data via the axis bus.
   - Moreover, the CAN bus is available.

B. Only the axis bus master of the network is connected to the higher-level master control via MCI module and the respective fieldbus.
   - For cost reasons, this topology only makes sense if only few information has to be transferred from the master control to the network.

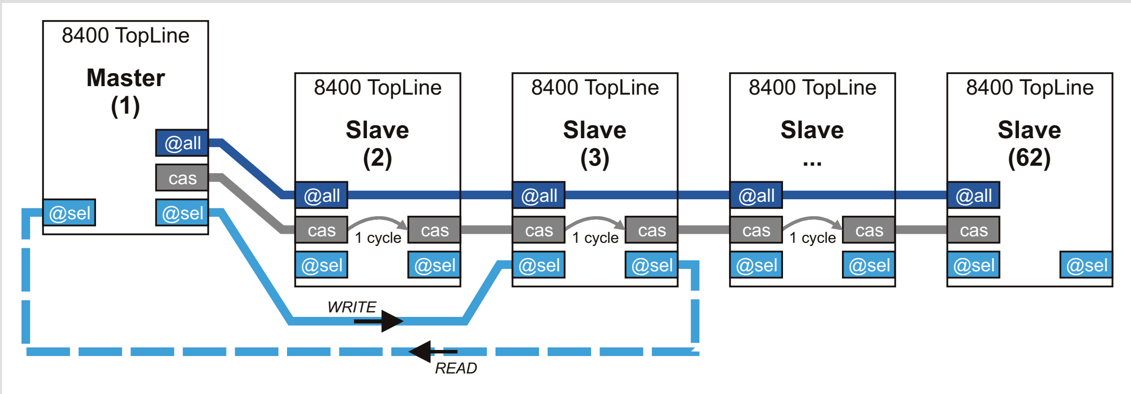
In both cases, the axes can exchange data via the axis bus and the CAN bus is available in addition.

### 13.1.2 Transfer mechanisms

The axis bus supports three different transfer mechanisms simultaneously according to the following methodology:

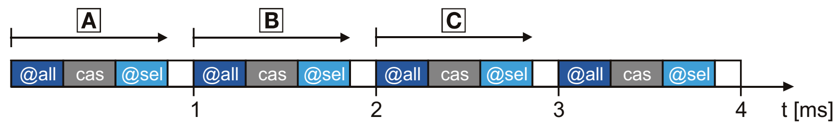
- "@all": Data transfer from master to all slaves
  - A data packet is sent from the master to all slaves in the transmission cycle.
  - Application: Data distribution according to line topology
  - Cycle time = transmission cycle = 1 ms (at baud rate = 500 kbits)
- "cas": Data transfer from node to node (cascade)
  - In the first transmission cycle, a data packet is sent from master to slave 1, in the next transmission cycle from slave 1 to slave 2, etc.
  - Application: Data distribution according to cascade topology
  - Cycle time = transmission cycle * no. of slaves+1

- "@sel": Data transfer from master to a slave
  - A data packet is sent from the master to one selected slave in the transmission cycle.
  - More options: Send to no slave, send to all slaves, and read one selected slave.
  - Application: Data distribution according to line topology for controlling one single node.
  - Cycle time = transmission cycle




[13-2] Transfer mechanisms

The axis bus data is transferred in 1-ms transmission cycle:




**Case 1:**
A Master sends setpoints
B Slave works with new data

**Case 2:**
A Master requests slave to send
B Slave sends as requested
C Data is processed by master

[13-3] Data transfer

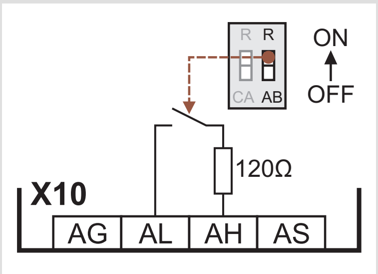
The axis bus data transfer is connected to the master/slave application via the **LS_AxisBus** systems blocks:

- For a data transfer from the master to all slaves (@all) and/or a cascaded data transfer (cas), the systems blocks LS_AxisBusIn and LS_AxisBusOut are available in the FB Editor.
- For a data transfer from the master to a certain slave (@sel), the LS_AxisBusAux system block is available.

### 13.1.3 Activating the bus terminating resistor

The axis bus must be terminated between axis bus low (AL) and axis bus high (AH) at the first and last physical node each by a resistor (120 Ω).

The 8400 controller is provided with an integrated bus terminating resistor, which can be activated via the DIP switch labelled with "AB":




- OFF = bus terminating resistor is inactive
- ON = bus terminating resistor is active

[13-4] Activation of the integrated bus terminating resistor

### 13.1.4 Parameter setting

#### Short overview of the relevant parameters:

| Parameter | Info | Lenze setting Value | Unit |
|---|---|---|---|
| C01120 | Sync signal source | 0: Off | |
| C02430/1 | Axis bus address<br>• 1 ≡ Master<br>• 2 ... 62 ≡ slave 1 ... slave 61 | 1 | |
| C02430/2 | Axis bus no. of nodes<br>• Number of slaves connected to the axis bus.<br>• Setting only required for the master. | 2 | |
| C02431/1 | Axis bus decel. boot-up operat.<br>• Delay during status change from "Boot-up" to "Operational". | 3000 | ms |
| C02431/2 | Axis bus decel. operat.- 1.transmit<br>• "Operational" time until "first transmission". | 1000 | ms |
| C02431/3 | Axis bus decel. BusOffRecovery | 1000 | ms |
| C02435 | Axis bus status | - | |
| C02436 | Axis bus error status | - | |
| C02437 | Axis bus MessageError | - | |
| C02440/1 | Axis bus IO function | 0: Off | |

Highlighted in grey = display parameter

### Defining the master

For a "safe" data transfer, one controller must be the master in the network. The master controls and synchronises the network.

Make the following settings for the controller which is to be the master in the network:

- Axis bus address (C02430/1) = "1"
  - With this setting, the controller automatically takes over the control in the network.
  - Only one master is allowed in the network.
- Axis bus IO function (C02440/1) = "1: Master"
  - With this setting, the controller outputs a synchronisation cycle to the I/O axis bus to which the slaves can orient themselves.

### Required settings for the slaves

Make the following settings for the other controllers in the network:

- Axis bus address (C02430/1) = "2" ... "62"
  - Make sure that all controllers connected to the axis bus have different axis bus addresses.
- Sync signal source (C01120) = "2: AxisBusIO"
  - With this setting, the synchronisation cycle output by the master is used as synchronisation source.
  - Basically, only one source is allowed to synchronise the internal time base.
- Optional: Axis bus IO function (C02440/1) = "2: Slave"
  - Only required if the IO axis bus is to be used for transfer of controller errors in the network ("release cord" principle).

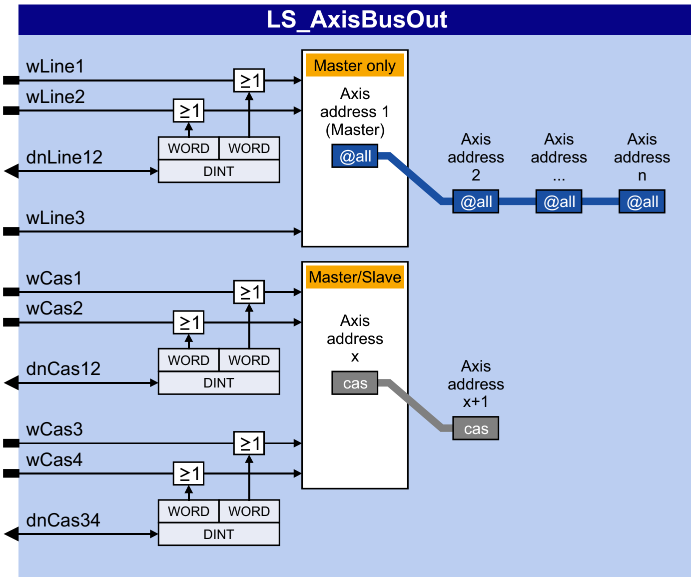
### Related topics:

- "Master/slave" function (📖 727)

## 13.1.5 Internal interfaces | System block "LS_AxisBusOut"

This system block is used to send line data from the master to all slaves (@all) and/or send the cascaded data (cas).

- The line data is sent as process data object (PDO) per transmission cycle to all slaves and forwarded to the application. Moreover, this PDO serves to transmit an internal control word (thus, only three words are available for data transfer).
- The cascaded data is also sent as PDO per transmission cycle from one slave to the next. The control which slave has to send when and at what time a slave in the chain has to accept the data, is executed by the internal control word in the PDO of the line data.

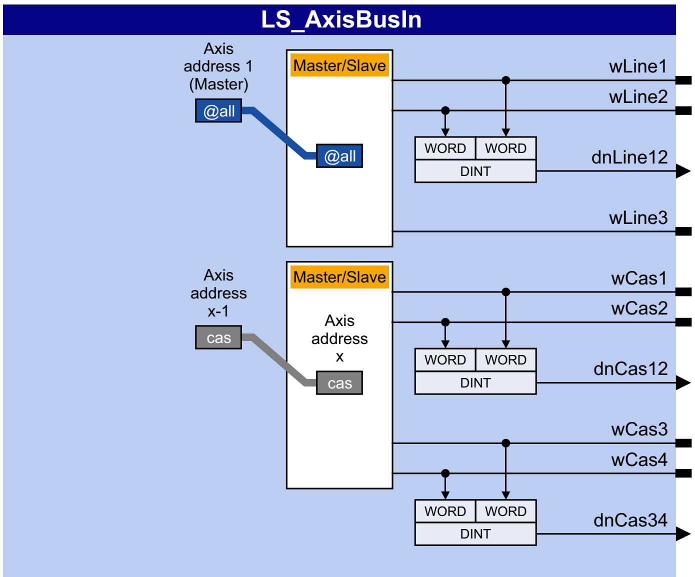



### Inputs

| Input<br>Data type | Information/possible settings |
|---|---|
| wLine1/wLine2<br>WORD | Line data - word 1 and word 2<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 1 and word 2. |
| dnLine12<br>DINT | |
| wLine3<br>WORD | Line data - word 3 |
| wCas1/wCas2<br>WORD | Cascaded data - word 1 and word 2<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 1 and word 2. |
| dnCas12<br>DINT | |
| wCas3/wCas4<br>WORD | Cascaded data - word 3 and word 4<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 3 and word 4. |
| dnCas34<br>DINT | |

### 13.1.6 Internal interfaces | System block "LS_AxisBusIn"

This system block is used to receive the (line) data from the master (@all) and/or receive the cascaded data (cas).

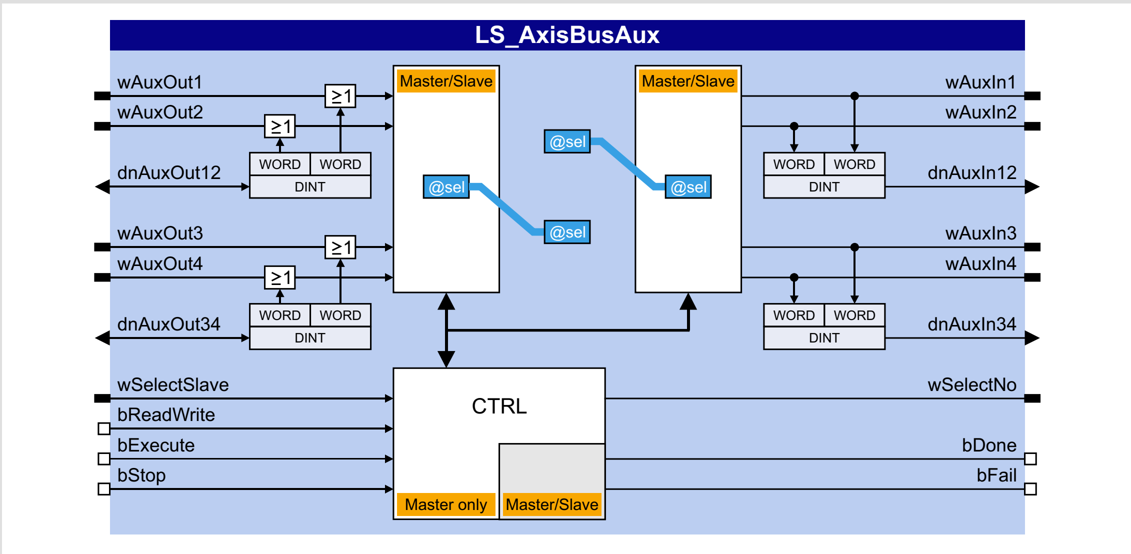



#### Outputs

| Output<br>Data type | Value/meaning |
|---|---|
| wLine1/wLine2<br>WORD | Line data - word 1 and word 2 |
| dnLine12<br>DINT | Line data - word 1 and word 2 as double word |
| wLine3<br>WORD | Line data - word 3 |
| wCas1/wCas2<br>WORD | Cascaded data - word 1 and word 2 |
| dnCas12<br>DINT | Cascaded data - word 1 and word 2 as double word |
| wCas3/wCas4<br>WORD | Cascaded data - word 3 and word 4 |
| dnCas34<br>DINT | Cascaded data - word 3 and word 4 as double word |

### 13.1.7 Internal interfaces | System block "LS_AxisBusAux"

This system block is used for data transfer between the master and one selected slave (@sel). Data can either be received from the slave ("read") or sent to the slave ("write"). By selecting the axis bus address "63", data can also be sent to all slaves at the axis bus.




#### Inputs

| Input / Data type | Information/possible settings | |
|---|---|---|
| wAuxOut1/wAuxOut2 (WORD) | Data to be sent - word 1 and word 2<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 1 and word 2. | |
| dnAuxOut12 (DINT) | | |
| wAuxOut3/wAuxOut4 (WORD) | Data to be sent - word 3 and word 4<br>• Selection as separate words or as double word.<br>• The double word is OR'd with word 3 and word 4. | |
| dnAuxOut34 (DINT) | | |
| The following inputs are only relevant for the master.<br>Via these inputs, the master controls the data acceptance of the "@sel" data in the slave. | | |
| wSelectSlave (WORD) | Axis bus address of the slave data is to be received of or data is to be sent to.<br>• "0" ≡ data is sent to no slave.<br>• "63" ≡ data is sent to all slaves. | |
| bReadWrite (BOOL) | Read/write access | |
| | FALSE | Data is received by selected slave. |
| | TRUE | Data is sent to selected slave. |
| bExecute (BOOL) | Data acceptance | |
| | TRUE | The pending data is accepted and the read/write access is executed. |
| bStop (BOOL) | Stop data transfer | |
| | TRUE | The axis bus changes to the "Stopped" status. In this status, the entire data transfer is switched off. |

## Outputs

| Output<br>Data type | Value/meaning | |
|---|---|---|
| wAuxIn1/wAuxIn2<br>WORD | Received data - word 1 and word 2 | |
| dnAuxIn12<br>DINT | Received data - word 1 and word 2 as double word | |
| wAuxIn3/wAuxIn4<br>WORD | Received data - word 3 and word 4 | |
| dnAuxIn34<br>DINT | Received data - word 3 and word 4 as double word | |
| wSlaveNo<br>WORD | Display which slave has sent its data.<br>• Output only relevant for master. | |
| bDone<br>BOOL | Status signal "Data transfer completed" | |
| | TRUE | Data has been received correctly. |
| bFail<br>BOOL | Status signal "Error" | |
| | TRUE | Data has not been received correctly (e.g. bus error). |

## 13.2 IO axis bus

**Stop!**

The IO axis bus of the 8400 TopLine controller is not compatible to the state bus of the 9300/9400 device series due to different voltage levels!

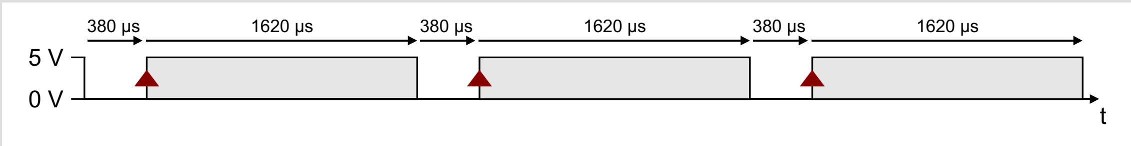
In order to use the IO axis bus, first decide whether the IO axis bus is to be used for transferring controller errors in the network ("release cord" principle) or as a pure open-collector IO function. The required function of the IO axis bus must then be set for all nodes in C02440/1.

- "Master/slave" function ("release cord" principle)
  - In this setting, the IO axis bus only knows the two statuses "OK" and "Error".
  - Each node connected to the IO axis bus can set the IO axis bus into the "Error" status.
  - In the "Error" status, all nodes start their adjustable response, e.g. a synchronised braking of the drive system or the master only brakes the network to standstill.
  - The "Error" status can only be reset by the node defined as "master".
  - The internal time base of the controllers can be synchronised via the IO axis bus.
- "IO" function
  - With this setting, the IO axis bus is used as a pure IO transmission medium.
  - Each node connected to the IO axis bus can transmit a TRUE signal to all nodes.

### 13.2.1 "Master/slave" function

With this setting, the IO axis bus can be used as "release cord", i.e. each node can set all other nodes into an error status.

One IO axis bus node takes over the master role. The master's task is to output a synchronisation cycle to the IO axis bus to which the slaves can orient themselves:

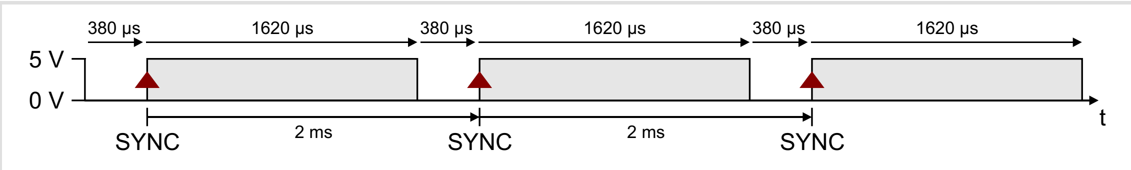



[13-5] IO axis bus: Synchronisation cycle from the master (error-free status)

The controller is configured as master or slave in C02440/1 or by selecting "1: Master" or "2: Slave".

#### Synchronisation of the internal time base

The internal time base of the slaves can be synchronised via the IO axis bus. The slaves orient themselves to the synchronisation cycle output by the master:




[13-6] Synchronisation cycle

- In order that the IO axis bus is used as synchronisation source, all slaves must be set to "AxisBusIO" in C01120.
  - The internal time base is synchronised based on the HIGH edge in a 2-ms cycle (see illustration above).
  - Internal timing and trimming of the phase position are set automatically.
- Basically, only one source is allowed to synchronise the internal time base. If "AxisBusIO" is selected, this is the IO axis bus master.

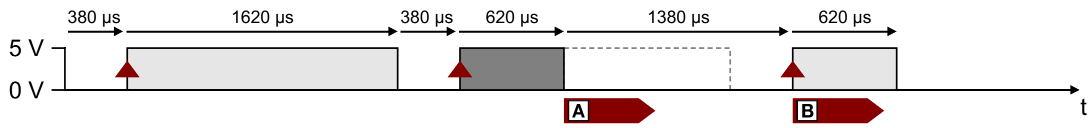
More information on this subject can be found in the following main chapter "Synchronisation of the internal time base". (📖 731)

#### Error-free status

In the error-free status, all nodes are error-active, i.e. each node can set the IO axis bus into the "Error" status.

### Error status

By setting the LS_AxisBusIO.*bSetFail_DigOut* input to TRUE, each node can set the IO axis bus into the "Error" status. The node signalises this by reducing the HIGH pulse length to 620 µs:




A A node signalises "Error" by a reduced HIGH pulse length
B "Error" status is active for all nodes

[13-7] Synchonisation cycle (error status)

All other nodes detect the "error status" due to the changed pulse length which has the following effect:

- The LS_AxisBusIO.*bFail_DigIn* output is set to TRUE. This digital signal can be used for any application within the function block interconnection.
- The nodes are now error-passive, i.e. they cannot signalise any further errors in order that the HIGH edge for synchronisation can be detected after 2 ms.

**Tip!**

In order to trigger a parameterisable error response, you can connect the *bFail_DigIn* output signal of the SB LS_AxisBusIO to a *bSetError* input of the SB LS_SetError_1 or SB LS_SetError_2.

### Reset "error" status

**Note!**

The "error" status can only be reset by the master!

A FALSE/TRUE edge at the LS_AxisBusIO.*bResetFail* input (for the master) serves to reset the "error" status again. The master signalises this by providing a LOW pulse for 3 ms.

All other nodes detect the error reset due to this signal, which has the following effect:

- The LS_AxisBusIO.*bFail_DigIn* output is reset to FALSE.
- After a delay time of 2 ms, the LS_AxisBusIO.*bResetFailIn* output is set for 3 ms to TRUE. This digital signal can be used within the function block interconnection to e.g. reset errors.

### 13.2.2 "IO" function

With this setting, the IO axis bus is used as a pure IO transmission medium, just like it the case for the 9300 and 9400 device series.

The "IO" function is activated in C02440/1 by selecting "3: IO".

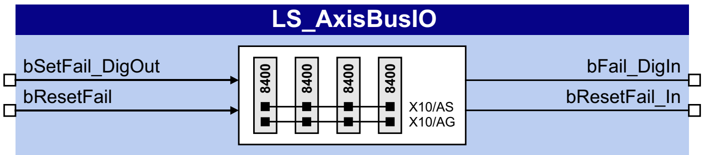
**Note!**

- "IO" function and Data transfer axis bus are mutually exclusive since with activated "IO" function, a synchronisation of the internal time base is not possible (which is required for a "safe" data transfer).
- The "IO" function must be set for all IO axis bus nodes.
- Triggering an error message with this function is only possible by function block interconnection.
- Resetting an error message is not possible with this function.

- ▶ By setting the LS_AxisBusIO.*bSetFail_DigOut* input to TRUE, the IO axis bus is set to the dominant level and for all nodes, the LS_AxisBusIO.*bFail_DigIn* output is set to TRUE.
- ▶ The LS_AxisBusIO.*bResetFail* input has no function.
- ▶ The LS_AxisBusIO.*bResetFail_In* output is always FALSE.

### 13.2.3 Internal interfaces | System block "LS_AxisBusIO"

The **LS_AxisBusIO** system block maps the IO axis bus in the FB Editor:




#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bSetFail_DigOut<br>BOOL | Set IO axis bus into the "error" status / IO data exchange | |
| | TRUE | For all IO axis bus nodes, the *bFail_DigIn* output is set to TRUE. |
| bResetFail<br>BOOL | Reset "error" status<br>• Only possible with "Master/slave" function via the IO axis bus master. | |
| | FALSE↗TRUE | For all IO axis bus nodes, the *bFail_DigIn* output is reset to FALSE.<br>After a delay time of 2 ms, the *bResetFail_In* output is set to TRUE for 3 ms for error reset. |

#### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bFail_DigIn<br>BOOL | With "master/slave" function: "Error" status<br>With "IO" function: Data exchange | |
| | TRUE | An IO axis bus nodes has set the *bSetFail_Digout* input to TRUE. |
| bResetFail_In<br>BOOL | "Error" status has been reset<br>• With "IO" function, this output is always FALSE. | |
| | TRUE | The IO axis bus master has reset the error message.<br>This status is only pending for 3 ms. |

# 14 Synchronisation of the internal time base

In a drive system, synchronising the internal time bases of all controllers involved makes sense because cyclic process data should be processed synchronously in all drives.

- One of the following signal sources can be used for automatic synchronisation of the internal time base of the controller:
  - CAN bus ("CAN on board") ➜ sync telegram
  - Axis bus ("AxisBusIO") ➜ master/slave function
  - MCI ➜ sync signal of a plugged-in communication module (EtherCAT, PROFINET or Powerlink)

**Short overview of the parameters for the synchronisation of the internal time base:**

| Parameter | Info | Lenze setting | |
|---|---|---|---|
| | | Value | Unit |
| C00370/1 | CAN Sync instant of transmission | - | µs |
| C00370/2 | Sync instant of reception | - | µs |
| C01120 | Sync signal source | Off | |
| C01121 | Sync cycle time setpoint | 1000 | µs |
| C01122 | Sync phase position | 0 | µs |
| C01123 | Sync window | 100 | µs |
| C01124 | Sync correction width | 320 | ns |
| Highlighted in grey = display parameter | | | |

### Sync signal source

The synchronisation signal source can be selected in C01120. As a general rule, only one source can be used to synchronise the internal time base.

### Sync cycle time setpoint

Time after which the internal phase-locking loop (PLL) anticipates the synchronisation signals. The time must be set in C01121 according to the cycle of the synchronisation source selected in C01120.

**Note!**

- Only integer multiples of 1000 µs can be set in C01121.
- Intelligent communication modules usually define the cycle time setpoint derived from the bus cycle. In this case, a manual change is not possible.
- Even in case of a synchronisation via axis bus, a setting of the cycle time and the phase position is not required/possible and is executed automatically.

Example: For the CAN bus, 2 ms has been selected as interval between two synchronisation signals. If the CAN bus is to be used as synchronisation source, a cycle time setpoint of 2000 µs must be selected in C01121.

### Sync phase position

The phase position determines the zero-time of the internal system cycle with regard to the synchronisation signal (bus cycle). Since PDO processing is an inherent part of the system part of the application, the instant of acceptance of the PDOs is postponed as well by a changed phase position.

- If "0" is set, the internal system cycle starts at the same time as the synchronisation signal.
- If a value > 0 is set, the internal system cycle starts by the set time earlier (the phase position has a negative effect) than the synchronisation signal.
- Intelligent communication modules define the optimal time with activated synchronisation by themselves. In this case, a manual change is not possible.
- For determining C01122, the point in time where all bus nodes have valid PDOs is decisive.

Example: If the phase position is set to 550 μs, the system part of the application starts 550 μs before the arrival of the synchronisation signal.

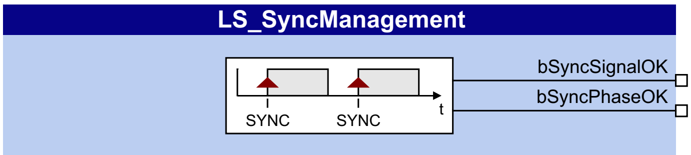
### Sync correction width

If the cycle times of the synchronisation signal and the phase-locking loop (PLL) are different, the setting in C01124 defines the correction increments for the phase-locking loop.

- The recommended reset time for the CAN bus as synchronisation source in case of occurring deviations is 320 ns (Lenze setting).
- If synchronisation is not reached, select a higher correction width.
- The optimum setting depends on quartz precision and must be determined empirically if required.

## 14.1 Internal interfaces | System block "LS_SyncManagement"

The SB **LS_SyncManagement** provides status information for synchronising the internal time base:




### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bSyncSignalOK<br>BOOL | TRUE | Sync signal OK |
| bSyncPhaseOK<br>BOOL | TRUE | Sync phase position OK |

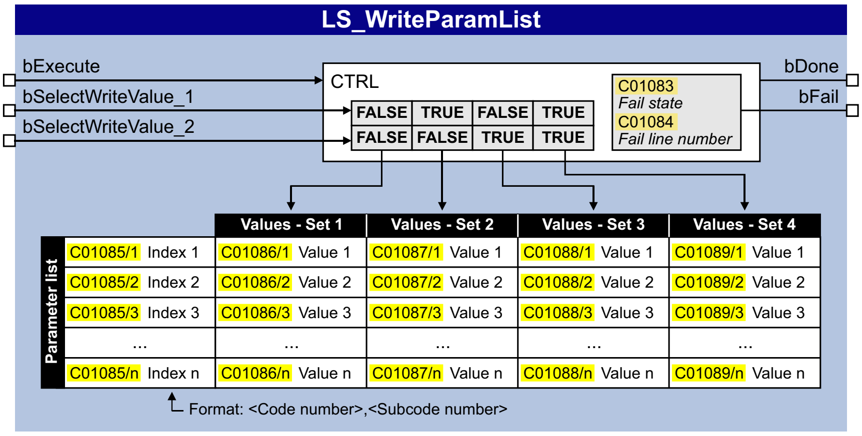
# 15 Parameter change-over

For up to 32 freely selectable parameters, this basic function provides a change-over between four sets with different parameter values.

The parameter list is created in the same way as the user menu is composed, namely by means of parameterisation. In the »Engineer«, a user-friendly parameterisation dialog with import and export functions is available for this purpose.

## 15.1 Internal interfaces | System block "LS_WriteParamList"

The **LS_WriteParamList** system block provides the internal interfaces for the basic "Parameter change-over" function:




### Inputs

| **Identifier** (Data type) | **Information/possible settings** | | |
|---|---|---|---|
| bExecute (BOOL) | FALSE⇗TRUE | If Execute Mode (C01082) = "0: by Execute":<br>Activate writing of the parameter list | |
| bSelectWriteValue_1<br>bSelectWriteValue_2 (BOOL) | Binary coded selection of the value set 1 ... 4 to be used. | | |
| | *bSelectWrite Value_1* | *bSelectWrite Value_2* | |
| | FALSE | FALSE | Value set 1 (C01086/1 ... n) |
| | TRUE | FALSE | Value set 2 (C01087/1 ... n) |
| | FALSE | TRUE | Value set 3 (C01088/1 ... n) |
| | TRUE | TRUE | Value set 4 (C01089/1 ... n) |

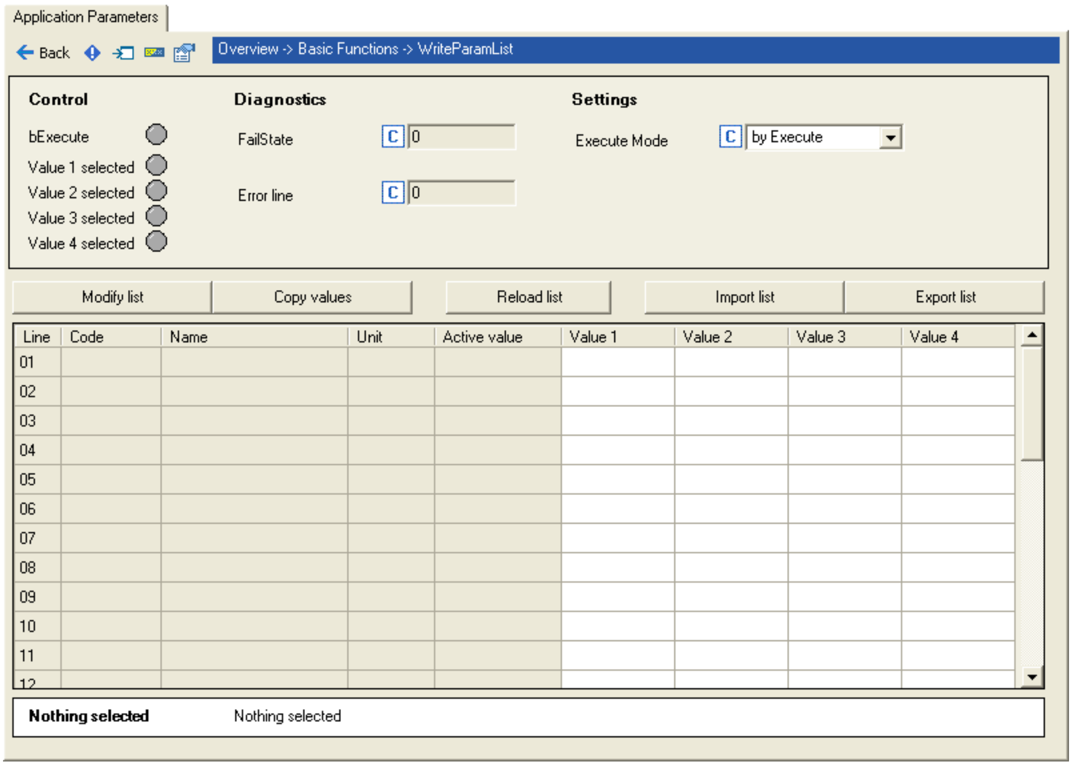
### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bDone<br>BOOL | "Writing of the parameter list completed" status signal<br>• The output is automatically reset to FALSE if writing via *bExecute* is activated again. | |
| | TRUE | Writing of the parameter list successfully completed. |
| | FALSE | The FALSE status can have the following meanings:<br>1. There is no active writing of the parameter list.<br>2. Writing of the parameter list has not been completed yet.<br>3. An error has occurred (if *bFail* = TRUE). |
| bFail<br>BOOL | "Error" status | |
| | TRUE | An error has occurred (group signal).<br>• For details see display parameter C01083. |

## 15.2 Configuring the list using the »Engineer« parameterisation dialog

**Proceed as follows to open the dialog for parameterising the parameter change-over:**

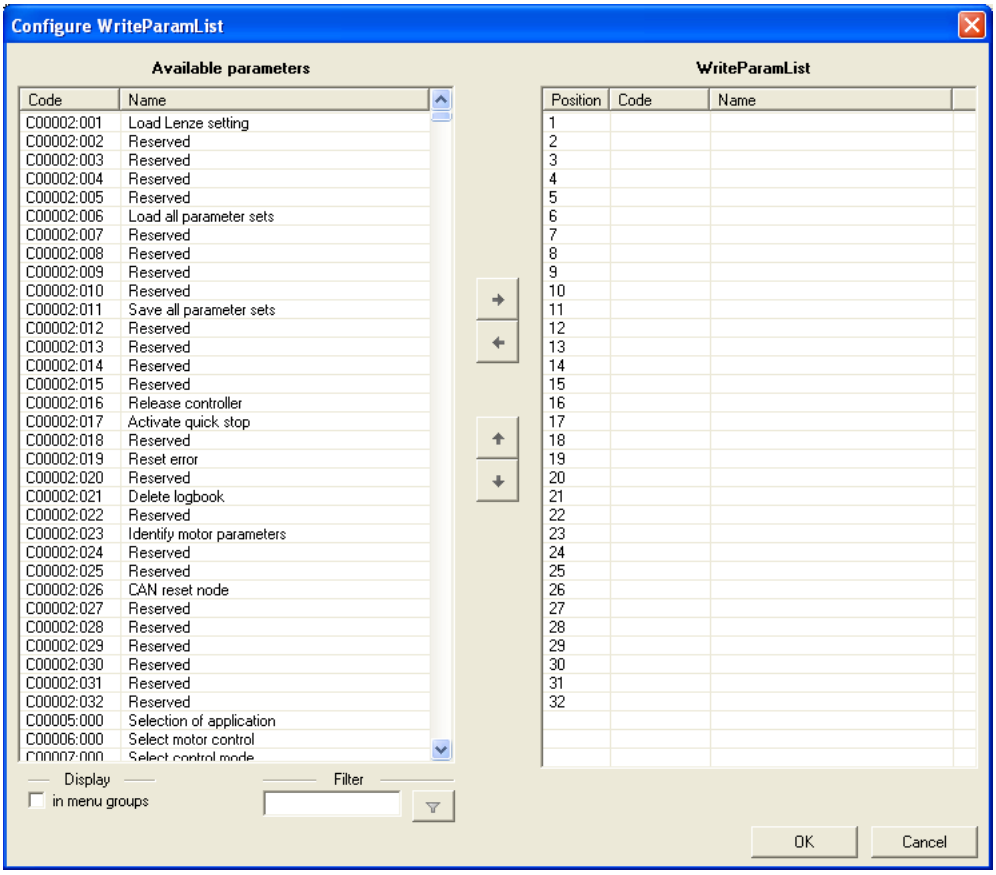
1. »Engineer« Go to the *Project view* and select the 8400 TopLine controller.
2. Go to *Workspace* and change to the **Application parameters** tab.
3. Go to the *Overview* dialog level and click the "basic functions" button.
4. Go to the *Overview* → *Basic functions* dialog box and click the **Parameter change-over** button.


Application Parameters
Back
Overview -> Basic Functions -> WriteParamList
Control
bExecute
Value 1 selected
Value 2 selected
Value 3 selected
Value 4 selected
Diagnostics
FailState
0
Error line
0
Settings
Execute Mode
by Execute
Modify list
Copy values
Reload list
Import list
Export list
Line
Code
Name
Unit
Active value
Value 1
Value 2
Value 3
Value 4
01
02
03
04
05
06
07
08
09
10
11
Nothing selected
Nothing selected

### Creating/changing the list

**To create or change the list, proceed as follows:**

1. Click on **Change list** button.
   - The dialog box entitled *Configure WriteParamList* is shown:




   - On the left-hand side, all the parameters of the drive controller with write and read access are shown in the list entitled **Available parameters**.
   - If the option **In menu groups** is activated, all parameters are shown assigned to their functions.
   - By clicking on the button in the **Filter** area, you can shorten the list of available parameters. If, for example, you enter the text "ain1" and then click on the button, only those parameters whose designation contains this text are shown for selection.
2. Highlight the parameter/parameters in the **Available parameters** list that is/are to be added to the *WriteParamList*.
   - Here, you can use the <**Ctrl**> key and the<**Shift**> key for multiple selection, as in the case of general Windows functions.

3. Click on the button in order to add the highlighted parameters to the *WriteParamList* on the right-hand side.
    - With the and buttons, you can alter the sequence of parameters in the *WriteParamList*.

    To remove parameters from the *WriteParamList*, proceed as follows:
    - Highlight the parameter/parameters in the **WriteParamList** that is/are to be removed from the *WriteParamList*.
    - Click on the button to remove the highlighted parameters from the *WriteParamList*.
4. Click on the **OK** button to accept the configuration and close the dialog box.
    - You can call the configuration dialog again at any time in order to change or expand the *WriteParamList* retrospectively.

### Entering values

After composing the list, you can directly enter the desired parameter values into the input fields (columns **1st value** ... **4th value**).

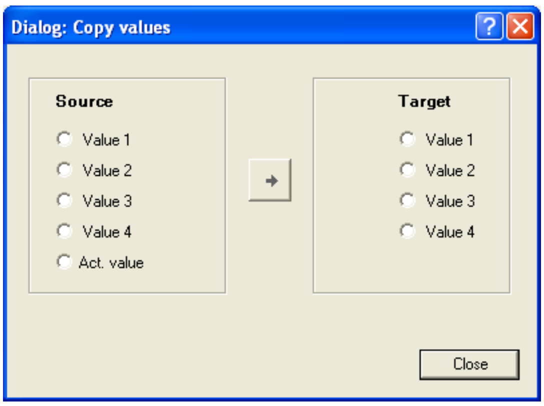
If you place the cursor in an input field, the permitted value range for the corresponding parameter is shown under the table.

### Copying values

All the settings of a value set can be copied to another value set.

**To copy values, proceed as follows:**

1. Click on the **Copy values** button.
    - The *Copy values* dialog box is displayed:




2. Select **Source** and **Target**.
3. Click on button in order to copy the values from **Source** to **target**.

### Importing/exporting the list

For cross-device reuse of the configured *WriteParamList*, you can click on the **Export list** and **Import list** buttons to save the parameter selection as an *.epc file and then to re-import the saved *.epc file into another drive controller 8400.

## 15.3 Configuring the list by means of parameterisation

The following application example shows the necessary procedure for configuring the list without using the »Engineer« parameterisation dialog.

Task:

Using the **LS_WriteParamList** SB, the C00012, C00026/1, C00027/1, and C00222 to C00224 parameters are to be written.

### Compiling the parameter list

In C01085/1 ... n, specify the above-named parameters in the <Code>,<Subcode> format:

- C01085/1 = 12.000
- C01085/2 = 26.001
- C01085/3 = 27.001
- C01085/4 = 222.000
- C01085/5 = 223.000
- C01085/6 = 224.000
- C01085/7 ... n = 0.000 (no parameter)

**Note!**

Gaps in the parameter list (setting = 0.000) are permissible and are skipped in the process.

Invalid parameter entries are not accepted at the input.

### Entering values for the parameters (value set 1)

In C01086/1 ... n, specify the values to be used to describe the selected parameters. The values are entered according to the scaling format/scaling factor of the respective parameter.

- C01086/1 = <value> for list entry 1 (in our example: for parameter C00012)
- C01086/2 = <value> for list entry 2 (in our example: for parameter C00026/1)
- C01086/3 = <value> for list entry 3 (in our example: for parameter C00027/1)
- etc.

These values are used in the writing process if the two *bSelectWriteValue_1* and *bSelectWriteValue_2* inputs are not assigned or both set to FALSE.

### Entering other values for the parameters (value sets 2 ... 4)

If required, up to three other sets can be set in the same way in C01087/1 ... n to C01089/1 ... n which can optionally be written to the parameters. The decision as to which value set is finally used is dependent upon the assignment of the two bSelectWriteValue_1 and bSelectWriteValue_2 inputs:

## 15.4 Selecting a value set

The value set to be used is selected via the selection inputs *bSelectWriteValue_1* and *bSelectWriteValue_2* of the SB LS_WriteParamList:

| bSelectWriteValue_1 | bSelectWriteValue_2 | Value set used |
|---|---|---|
| FALSE | FALSE | Value set 1 (C01086/1 ... n) |
| TRUE | FALSE | Value set 2 (C01087/1 ... n) |
| FALSE | TRUE | Value set 3 (C01088/1 ... n) |
| TRUE | TRUE | Value set 4 (C01089/1 ... n) |

## 15.5 Activating the writing of the parameters

For writing the parameter list, two modes are available in C01082:

- 0: by Execute (Lenze setting)
  The writing of the parameter list is activated by a FALSE/TRUE edge at the *bExecute* control input.
- 1: by Input Select
  The parameter list is written when the selection inputs *bSelectWriteValue_1* and *bSelectWriteValue_2* are changed and once during the initialisation of the controller.

The parameters are written one at a time every time the main program is executed until the entire parameter list is processed. In the event of an error, respective error messages are output.

**Note!**

The "parameter change-over" basic function is always processed, even if the **LS_WriteParamList** system block has been removed from the interconnection by the FB Editor.

If you do not require this basic function anymore, delete the composed parameter list in order that no unwanted parameter write operations take place.

**After successful completion**

... the *bDone* output is set to TRUE.

- The *bDone* output is automatically reset to FALSE if writing via *bExecute* is activated again.

**In the event of an error**

... the *bDone* output remains set to FALSE and the *bFail* output is set to TRUE.

- C01083 displays an error status and C01084 displays the number of the list entry at which the error occurred (in connection with the selected value set).
- If several errors occur at the same time, only the first incorrect list entry will be displayed. Hence, after elimination of the displayed error and another activation, more errors may be displayed.
- The parameter list will always be processed from beginning to end, even if errors occur in the meantime.

# 16 Parameter reference

This chapter describes all parameters which can be used for parameterising and monitoring the controller.

Parameters which are only available in the controller from a certain software version onwards are marked with a corresponding note in the parameter description ("from version xx.xx.xx").

The parameter descriptions are based on software version V02.00.00

**Tip!**

For quick reference of a parameter with a certain name, simply use the **index** of the online documentation. The index always contains the corresponding code in parentheses after the name.

General information on parameter setting can be found in the chapter "Introduction: Parameterising the controller". (📖 30)

For general information on how to read and change parameters, please see the online documentation for the »Engineer«.

## 16.1 Structure of the parameter descriptions

Each parameter is described in the Parameter list in the form of a table which consists of the following three areas:

### Table header

The table header contains the following general information:

- Parameter number (Cxxxxx)
- Parameter name (display text in the »Engineer» and keypad)
- Data type
- Parameter index in decimal and hexadecimal notation for access via a fieldbus (e.g. CAN system bus).

**Tip!**

The parameter index is calculated as follows:

- Index [dec] = 24575 - code
- Index [hex] = 0x5FFF - code

Example for code C00005:

- Index [dec] = 24575 - 5 = 24570
- Index [hex] = 0x5FFF - 0x{5} = 0x5FFA

### Table contents

The table contains further general explanations & notes on the parameter and the possible settings, which are represented in different ways depending on the parameter type:

- Parameters with read-only access
- Parameters with write access

### Table footer

The table footer contains the Parameter attributes.

### 16.1.1 Data type

The parameters can be of the following data types:

| Data type | Meaning |
|---|---|
| INTEGER_16 | 16-bit value with sign |
| INTEGER_32 | 32-bit value with sign |
| UNSIGNED_8 | 8-bit value without sign |
| UNSIGNED_16 | 16-bit value without sign |
| UNSIGNED_32 | 32-bit value without sign |
| VISIBLE_STRING | String of characters from printable characters |

### 16.1.2 Parameters with read-only access

Parameters for which the "write access" attribute has not been set, can only be read. They cannot be changed by the user.

#### Description structure

<table>
<tr><td colspan="4">Parameter | Name:<br>**Cxxxxx | ______________**</td><td>Data type: _______<br>Index: _______</td></tr>
<tr><td colspan="5">Description</td></tr>
<tr><td colspan="3">**Display range** **(min. value | unit | max. value)**</td><td colspan="2"></td></tr>
<tr><td></td><td></td><td></td><td colspan="2"></td></tr>
<tr><td colspan="5">☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1</td></tr>
</table>

#### Representation in the »Engineer«

The »Engineer« displays these parameters with a grey background or, with an online connection, with a pale-yellow background:

| | C... | S | Name | Value | Unit |
|---|---|---|---|---|---|
| | 3 | 0 | Status of last device command | Successful | |

## 16.1.3 Parameters with write access

Only parameters with a check mark (☑) in front of the "write access" attribute can be changed by the user. The Lenze setting for these parameters is **printed in bold**.

- The settings can either be selected from a selection list or the values can be entered directly.
- Values outside the valid setting range are represented in red in the »Engineer«.

### 16.1.3.1 Parameters with setting range

#### Description structure

| Parameter \| Name:<br>**Cxxxxx \| ______________** | | | | Data type: _______<br>Index: _______ |
|---|---|---|---|---|
| Description | | | | |
| **Setting range** **(min. value \| unit \| max. value)** | | | | **Lenze setting** |
| | | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | | |

#### Parameter setting in the »Engineer«

In the »Engineer«, parameters are set by entering the desired value into the input field:

| | C... | S | Name | Value | Unit |
|---|---|---|---|---|---|
| | 11 | 0 | Appl.: Reference speed | 1500 | rpm |

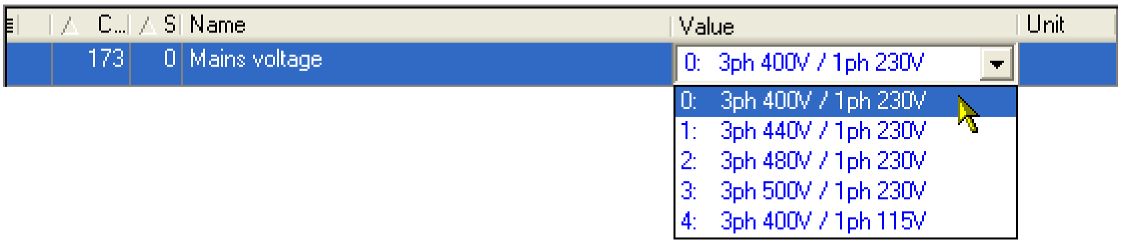
### 16.1.3.2 Parameters with selection list

#### Description structure

| Parameter \| Name:<br>**Cxxxxx \| ______________** | | Data type: _______<br>Index: _______ |
|---|---|---|
| Description | | |
| **Selection list** **(Lenze setting printed in bold)** | | |
| **1** | | |
| 2 | | |
| 3 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### Parameter setting in the »Engineer«

In the »Engineer«, a list field is used for parameter setting:



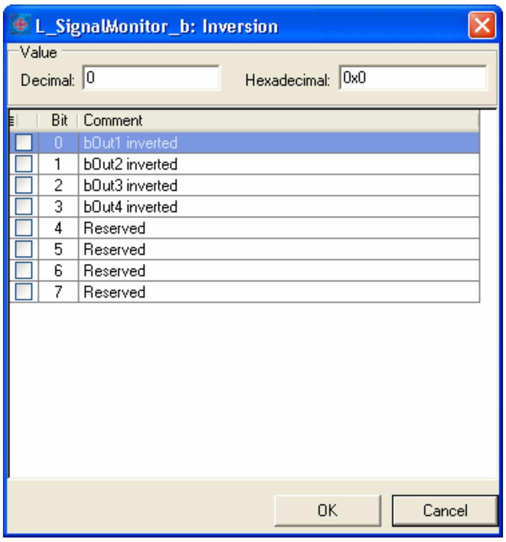

## 16.1.3.3 Parameters with bit-coded setting

### Description structure

| Parameter \| Name:<br>**Cxxxxx \| ______________** | | Data type: _______<br>Index: _______ |
|---|---|---|
| Description | | |
| **Value is bit-coded:** | | |
| Bit 0 | | |
| ... | | |
| Bit 31 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### Parameter setting in the »Engineer«

The »Engineer« uses a dialog box for parameter setting in which the individual bits can be set or reset. Alternatively, the value can be entered as a decimal or hexadecimal value:



### 16.1.3.4 Parameters with subcodes

#### Description structure

| Parameter \| Name:<br>**Cxxxxx \| ______________** | | | Data type: _______<br>Index: _______ |
|---|---|---|---|
| Description | | | |
| **Setting range** **(min. value \| unit \| max. value)** | | | |
| | | | |
| **Subcodes** | **Lenze setting** | | |
| Cxxxxx/1 | | | |
| Cxxxxx/2 | | | |
| Cxxxxx/3 | | | |
| Cxxxxx/4 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

#### Parameter setting in the »Engineer«

The »Engineer« parameter list displays each subcode individually. The parameters are set as described in the previous chapters.

| | C... | S | Name | Value | Unit |
|---|---|---|---|---|---|
| | 39 | 1 | Fixed setpoint 1 | 40.00 | % |
| | 39 | 2 | Fixed setpoint 2 | 60.00 | % |
| | 39 | 3 | Fixed setpoint 3 | 80.00 | % |
| | 39 | 4 | Fixed setpoint 4 | 0.00 | % |

### 16.1.4 Parameter attributes

The table footers contain the parameter attributes:

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

| **Attribute** | **Meaning** | |
|---|---|---|
| ☑ Read access | Read access to parameter possible. | |
| ☑ Write access | Write access to parameter possible.<br>• Please also observe the following attributes: | |
| | ☑ CINH | Parameter value can only be changed when the controller is inhibited. |
| | ☑ PLC STOP | Parameter value can only be changed when the application is stopped. |
| ☑ No transfer | Parameter is **not** transferred to controller when the command Download parameter set is executed. | |
| ☑ COM | Communication-relevant parameter<br>• This parameter is relevant for parameter data transfer via the (CAN) system bus. | |
| ☑ MOT | Motor control parameters | |

### Scaling factor

The "scaling factor" is important for parameter access via a bus system.

| Signal type | Scaling factor | Resolution | Value range |
|---|---|---|---|
| Analog (scaled) | 100 | 16 bits signed | ± 199.99 % |
| Angular velocity | 1 | 16 bits signed | ± 32767 incr./ms |
| Position in [units] | 10000 | 32 bits signed | ± 214748.3647 [units] |
| Digital (BOOL) | 1 | 8 bits unsigned | 0 ≡ FALSE; 1 ≡ TRUE |
| Time | 1000 | 16 bits unsigned | 0 ... 999.000 s |
| Selection value | 1 | 16 bits unsigned | 0 ... 65535 |

Example 1: The value "654" of the parameter C00028/1 (AIN1: input voltage) read via a bus system must be divided by the corresponding scaling factor "100" to obtain the actual display value "6.54 V".

$$\frac{\text{Read value (via bus system)}}{\text{Scaling factor}} = \text{Indicated value (Engineer)}$$

[16-1] Conversion formula for read access via bus system

Example 2: In order to set the parameter C00012 (acceleration time main setpoint) to the value "123.45 %" via a bus system, the integer value "12345" must be transferred, i.e. the value to be set must be multiplied by the corresponding scaling factor "100".

$$\text{Value to be written (via bus system)} = \text{Value to be set} \cdot \text{Scaling factor}$$

[16-2] Conversion formula for write access via bus system

### Character length

In case of parameters of "VISIBLE_STRING" data type, the character length is given in addition. This is also important for the parameter access via a bus system.

## 16.2 Parameter list

This chapter lists all parameters of the operating system in numerically ascending order.

**Note!**

The parameter descriptions are based on the software version V02.00.00.

**C00002**

| Parameter \| Name: **C00002 \| Device command** | | Data type: UNSIGNED_8 Index: $24573_d$ = $5FFD_h$ |
|---|---|---|

**Note:**
- Before switching off the supply voltage after carrying out a device command, check whether the device command has been carried out successfully via the status display under C00003!
- Before activating device commands by a master control, wait for the "Ready" signal of the controller.
- The device will reject a write process to C00002/x if the value is >1 and issue an error message.

▸ Drive control (DCTRL): Device commands

| Selection list | |
|---|---|
| 0 | Off / ready |
| 1 | On / start |
| 4 | Action cancelled |
| 5 | No access |
| 6 | No access controller inhibit |
| 20 | 20% working |
| 40 | 40% working |
| 60 | 60% working |
| 80 | 80% working |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00002/1 | 0: Off / ready | Load Lenze setting<br>• All parameters are reset to the Lenze setting.<br>• Only possible when the controller is inhibited. |
| C00002/2 | 0: Off / ready | Reserved |
| C00002/3 | 0: Off / ready | Reserved |
| C00002/4 | 0: Off / ready | Reserved |
| C00002/5 | 0: Off / ready | Reserved |
| C00002/6 | 0: Off / ready | Load all parameter sets<br>• All parameter sets are loaded by the memory module.<br>• Only possible when the controller is inhibited. |
| C00002/7 | 0: Off / ready | Reserved |
| C00002/8 | 0: Off / ready | Reserved |
| C00002/9 | 0: Off / ready | Reserved |
| C00002/10 | 0: Off / ready | Reserved |
| C00002/11 | 0: Off / ready | Save all parameter sets<br>• All parameter sets are saved to the memory module safe against mains failure. |
| C00002/12 | 0: Off / ready | Reserved |
| C00002/13 | 0: Off / ready | Reserved |

| Parameter \| Name: C00002 \| Device command | | Data type: UNSIGNED_8 Index: $24573_d$ = $5FFD_h$ |
|---|---|---|
| C00002/14 | 0: Off / ready | Reserved |
| C00002/15 | 0: Off / ready | Reserved |
| C00002/16 | 1: On / start | Enable controller<br>"1" ≡ Enable controller<br>"0" ≡ Inhibit controller |
| C00002/17 | 0: Off / ready | Activate quick stop<br>"1" ≡ Activate quick stop<br>"0" ≡ Deactivate quick stop |
| C00002/18 | 0: Off / ready | Reserved |
| C00002/19 | 0: Off / ready | Reset error<br>• After the reset (acknowledgement) of the current error, further errors may be pending which must also be reset.<br>• The status determining error is displayed in C00168.<br>• The current error is displayed in C00170. |
| C00002/20 | 0: Off / ready | Reserved |
| C00002/21 | 0: Off / ready | Delete logbook<br>• All entries in the logbook of the controller are deleted.<br>• In the logbook, information on the error history is saved. |
| C00002/22 | 0: Off / ready | Reserved |
| C00002/23 | 0: Off / ready | Motor parameter identification<br>• This device command serves to carry out automatic motor parameter identification.<br>• The identification method can be selected in C02867.<br>• The device command is only executed when the drive controller is in the "SwitchedOn" status.<br>• In order to identify the motor parameters, the controller must be enabled after this device command.<br>▶ Automatic motor parameter identification |
| C00002/24 | 0: Off / ready | Reserved |
| C00002/25 | 0: Off / ready | Identify resolver error<br>e ▶ Optimise resolver behaviour |
| C00002/26 | 0: Off / ready | CAN reset node<br>• Reinitialise "CAN on board" interface.<br>• Required when changing the baud rate, node address, or identifiers.<br>▶ System bus "CAN on board" |
| C00002/27 | 0: Off / ready | Device search function<br>• This device command serves to optically locate a controller connected online (e.g. for maintenance work).<br>▶ Device search function |
| C00002/28 | 0: Off / ready | Check MasterPin<br>▶ Unlocking the controller with a MasterPin |
| C00002/29 | 0: Off / ready | Set binding ID<br>▶ Device personalisation |
| C00002/30 | 0: Off / ready | Delete binding ID<br>▶ Device personalisation |
| C00002/31 | 0: Off / ready | Set password<br>▶ Password protection |

| Parameter \| Name: **C00002 \| Device command** | | Data type: UNSIGNED_8 Index: 24573$_d$ = 5FFD$_h$ |
|---|---|---|
| C00002/32 | 0: Off / ready | Check password ▶ Password protection |
| C00002/33 | 0: Off / ready | Delete password ▶ Password protection |
| C00002/34 | 0: Off / ready | Identify pole position (360°) • This device command is serves to carry out an automatic identification of the pole position. • The device command is only carried out if the controller is in the "Switched On" status • In order to identify the pole position, the controller must be enabled after this device command. ▶ Pole position identification |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

### C00003

Parameter | Name:
**C00003 | Status of last device command**

Data type: UNSIGNED_8
Index: 24572$_d$ = 5FFC$_h$

Status of the device command executed last (C00002).

**Note:**
Before switching off the supply voltage after carrying out a device command, check whether the device command has been carried out successfully via the status display!

▶ Drive control (DCTRL): Device commands

| Selection list (read only) | | Info |
|---|---|---|
| 0 | Successful | Device command has been executed successfully. |
| 1 | Command unknown | Device command implausible or unknown to the system. |
| 2 | Password protection | Unauthorised access for requested device command. ▶ Password protection |
| 3 | Time-out | Device command could not be processed in the defined time (timeout). |
| 4 | System error | |
| 5 | Command server assigned | |
| 6 | Controller inhibit required | |
| 10 | Memory module binding error | ▶ Device personalisation |
| 11 | Password too short | ▶ Password protection |
| 12 | Wrong password | |
| 13 | Password already set | |
| 14 | Password not assigned | |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

### C00005

| Parameter \| Name: **C00005 \| Application** | | Data type: UNSIGNED_16 Index: $24570_d$ = $5FFA_h$ |
|---|---|---|
| Selection of the technology application | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | Wiring has changed | This display appears if the FB interconnection has been changed in the application level using the FB Editor. |
| **1000** | **Actuating drive speed** | This technology application is used to solve speed-controlled drive tasks, e.g. conveying belts. |
| 2000 | Table positioning | This technology application is used to solve position-controlled drive tasks which are usually controlled by a higher-level control system using a fieldbus. |
| 3000 | Switch-off positioning | This technology application is used to solve speed-controlled drive tasks which require a pre-switch off or stopping at certain positions, e.g. roller conveyors and conveying belts. This is implemented by connecting switch-off sensors. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### C00006

| Parameter \| Name: **C00006 \| Motor control** | | Data type: UNSIGNED_8 Index: $24569_d$ = $5FF9_h$ |
|---|---|---|
| Selection of the motor control mode | | ▶ Motor control (MCTRL): Select control mode |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 1 | SC: Servo control PSM | This control type is used for the servo control of a synchronous motor.<br>• The control type requires a speed feedback via an encoder mounted to the motor!<br>▶ Servo control |
| 2 | SC: Servo control ASM | This control type is used for the servo control of an asynchronous motor.<br>• The control type requires a speed feedback via an encoder mounted to the motor!<br>▶ Servo control |
| 3 | SLPSM: Sensorless PSM | This control type is used for the sensorless control of a synchronous motor.<br>▶ Sensorless control for synchronous motors |
| 4 | SLVC: Vector control | This control type is used for sensorless vector control of an asynchronous motor.<br>• The control type requires motor parameters to be set as exactly as possible!<br>▶ Sensorless vector control |
| **6** | **VFCplus: V/f linear** | This control type is used for the speed control of an asynchronous motor via a linear V/f characteristic and is the simplest control type.<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▶ V/f characteristic control |
| 7 | VFCplus: V/f linear + encoder | This control type is used for speed control of an asynchronous motor via a linear V/f characteristic.<br>• The control type requires a speed feedback via an encoder mounted to the motor!<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▶ V/f control |

| Parameter \| Name: **C00006 \| Motor control** | | Data type: UNSIGNED_8 Index: $24569_d$ = $5FF9_h$ |
|---|---|---|
| 8 | VFCplus: V/f quadr | This control type is used for speed control of an asynchronous motor via a square-law V/f characteristic.<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▸ V/f characteristic control |
| 9 | VFCplus: V/f quadr + encoder | This control type is used for speed control of an asynchronous motor via a square-law V/f characteristic.<br>• The control type requires a speed feedback via an encoder mounted to the motor!<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▸ V/f control |
| 10 | VFCplus: V/f definable | This type of control is used for the speed control of an asynchronous motor via a user-definable characteristic with several interpolation points.<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>▸ V/f characteristic control |
| 11 | VFCplusEco: V/f energy-saving | This control type is used for energy-saving speed control of an asynchronous motor via a linear V/f characteristic.<br>• For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered.<br>• Predestinated application areas of this control type are materials handling technology and pump and fan systems.<br>▸ V/f characteristic control, energy-saving |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00007**

| Parameter \| Name: **C00007 \| Control mode** | | Data type: UNSIGNED_16 Index: $24568_d$ = $5FF8_h$ |
|---|---|---|

With this parameter the control mode for the technology application selected under C00005 is defined, i.e. how the inputs and outputs of the technology application are connected to the I/Os of the controller.

- How the inputs and outputs are connected in the individual control modes is described in the corresponding technology application:
  - "Actuating drive speed" TA
  - TA "Table positioning"
  - "Switch-off positioning" TA

| Selection list (Lenze setting printed in bold) | | Info |
|---|---|---|
| 0 | Wiring has changed | This is displayed when the FB interconnection has been changed in the I/O level via the FB Editor. |
| **10** | **Terminals 0** | The technology application is controlled via the digital and analog input terminals of the controller. • For a short overview of the preconfigured terminal assignment see the following section "Terminal assignment of the control modes 10 ... 16". |
| 12 | Terminals 2 | |
| 14 | Terminals 11 | |
| 16 | Terminal 16 | |
| 20 | Keypad | The technology application is controlled via the keypad. |
| 21 | PC | The technology application is controlled via the "Free parameters" of the controller (PC control). |
| 30 | CAN | The technology application is controlled by means of CAN-PDOs via the system bus "CAN on board". ▶ System bus "CAN on board" |
| 40 | MCI | The technology application is controlled by means of MCI-PDOs via the MCI-interface of an attached communication module (e.g. PROFIBUS). |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## Terminal assignment of the control modes 10 ... 16

Note: The following tables only list the connected inputs/outputs.

| | "Actuating drive speed" TA | | | |
|---|---|---|---|---|
| **Input/output** | **10: Terminals 0** | **12: Terminals 2** | **14: Terminals 11** | **16: Terminals 16** |
| **RFR** | Controller enable / Reset of error message | | | |
| **DI1** | Fixed setpoint 1/3 | | Change of direction of rotation | Fixed setpoint 1/3 |
| **DI2** | Fixed setpoint 2/3 | | Activate manual DC-injection braking (DCB) | Fixed setpoint 2/3 |
| **DI3** | Activate manual DC-injection braking (DCB) | Quick stop | Motor potentiometer: Increase speed | CW rotation quick stop |
| **DI4** | Change of direction of rotation | | Motor potentiometer: Decrease speed | CCW rotation quick stop |
| **A1U, A1I** | Main speed setpoint (10 V ≡ 100 % reference speed) | | | |
| **O1U, O1I** | Actual speed value (10 V ≡ 100 % reference speed) | | | |
| **DO1** | Status "Drive is ready" | | | |
| **Relay output** | Status "Error is pending" | | | |

| TA "Table positioning" | | | | |
|---|---|---|---|---|
| Input/output | 10: Terminals 0 | 12: Terminals 2 | 14: Terminals 11 | 16: Terminals 16 |
| RFR | Controller enable / Reset of error message | | | |
| DI1 | - | - | Positive limit switch | - |
| DI2 | - | - | Negative limit switch | - |
| DI3 | Positive limit switch | - | - | Manual jog in positive direction |
| DI4 | Negative limit switch | Positioning profile, array bit (valency 4) | | Manual jog in negative direction |
| DI5 | Positioning profile, array bit (valency 2) | | | |
| DI6 | Positioning profile, array bit (valency 1) | | | |
| DI7 | Start of profile generation | | | |
| A1U, A1I | Main speed setpoint (10 V ≡ 100 % reference speed) | | | |
| O1U, O1I | Actual speed value (10 V ≡ 100 % reference speed) | | | |
| O2U, O2I | Current motor current (10 V ≡ 100 % $I_{max\_mot}$) | | | |
| DO1 | Status "Drive is ready" | | | |
| DO2 | Status "Target position (actual value) is in the target window" | | | |
| DO3 | Status "Home position is known" | | | |
| BD1, BD2 | Control of the holding brake | | | |
| Relay output | Status "Error is pending" | | | |

| "Switch-off positioning" TA | | | | |
|---|---|---|---|---|
| Input/output | 10: Terminals 0 | 12: Terminals 2 | 14: Terminals 11 | 16: Terminals 16 |
| RFR | Controller enable | Controller enable / Reset of error message | | |
| DI1 | Fixed setpoint 1/3 | Stop function 1 | | Fixed setpoint 1/3 |
| DI2 | Fixed setpoint 2/3 | Stop function 2 | Selection: Pre-switch off 1 | Fixed setpoint 2/3 |
| DI3 | Reset error messages | CW rotation quick stop<br>Selection: Switch-off position 1 | | CW rotation quick stop |
| DI4 | Change of direction of rotation | CCW rotation quick stop<br>Selection: Switch-off position 2 | | CCW rotation quick stop |
| DI5 | - | - | Stop function 2 | - |
| DI6 | - | - | Selection: Pre-switch off 2 | - |
| A1U | Main speed setpoint (10 V ≡ 100 % reference speed) | | | |
| BD1, BD2 | Control of the holding brake | | | |

**C00008**

| Parameter \| Name:<br>**C00008 \| Original application\|control source** | Data type: UNSIGNED_16<br>Index: $24567_d$ = $5FF7_h$ |
|---|---|

Display of the originally selected technology application and the originally selected control mode.
- This parameter shows the selection that was set with C00005 and C00007 before a change in the I/O level or the application level was carried out.
- For diagnostic purposes, this display serves to determine whether there is a standard interconnection in the controller or a change carried out by the user.

| Selection list (read only) | | Info |
|---|---|---|
| 0 | Free\|Free | Application: Interconnection has been changed.<br>I/O level: Interconnection has been changed. |
| 10 | Free\|Terminal0 | Application: Interconnection has been changed.<br>I/O level: "Terminals 0" control mode |
| 12 | Free\|Terminal2 | Application: Interconnection has been changed.<br>I/O level: "Terminal 2" control mode |
| 14 | Free\|Terminal11 | Application: Interconnection has been changed.<br>I/O level: "Terminal 11" control mode |

| Parameter \| Name: C00008 \| Original application\|control source | | Data type: UNSIGNED_16 Index: $24567_d$ = $5FF7_h$ |
|---|---|---|
| 16 | Free\|Terminal 16 | Application: Interconnection has been changed.<br>I/O level: "Terminal 16" control mode |
| 20 | Free\|Keypad | Application: Interconnection has been changed.<br>I/O level: "Keypad" control mode |
| 21 | Free\|PC | Application: Interconnection has been changed.<br>I/O level: "PC" control mode |
| 30 | Free\|CAN | Application: Interconnection has been changed.<br>I/O level: "CAN" control mode |
| 40 | Free\|MCI | Application: Interconnection has been changed.<br>I/O level: "MCI" control mode |
| 1000 | Speed\|Free | Application: Actuating drive speed<br>I/O level: Interconnection has been changed. |
| 1010 | Speed\|Terminal0 | Application: Actuating drive speed<br>I/O level: "Terminals 0" control mode |
| 1012 | Speed\|Terminal2 | Application: Actuating drive speed<br>I/O level: "Terminal 2" control mode |
| 1014 | Speed\|Terminal11 | Application: Actuating drive speed<br>I/O level: "Terminal 11" control mode |
| 1016 | Speed\|Terminal16 | Application: Actuating drive speed<br>I/O level: "Terminal 16" control mode |
| 1020 | Speed\|Keypad | Application: Actuating drive speed<br>I/O level: "Keypad" control mode |
| 1021 | Speed\|PC | Application: Actuating drive speed<br>I/O level: "PC" control mode |
| 1030 | Speed\|CAN | Application: Actuating drive speed<br>I/O level: "CAN" control mode |
| 1040 | Speed\|MCI | Application: Actuating drive speed<br>I/O level: "MCI" control mode |
| 2000 | TabPos\|Free | Application: Table positioning<br>I/O level: Interconnection has been changed. |
| 2010 | TabPos\|Terminal0 | Application: Table positioning<br>I/O level: "Terminals 0" control mode |
| 2012 | TabPos\|Terminal2 | Application: Table positioning<br>I/O level: "Terminal 2" control mode |
| 2014 | TabPos\|Terminal11 | Application: Table positioning<br>I/O level: "Terminal 11" control mode |
| 2016 | TabPos\|Terminal16 | Application: Table positioning<br>I/O level: "Terminal 16" control mode |
| 2020 | TabPos\|Keypad | Application: Table positioning<br>I/O level: "Keypad" control mode |
| 2021 | TabPos\|PC | Application: Table positioning<br>I/O level: "PC" control mode |
| 2030 | TabPos\|CAN | Application: Table positioning<br>I/O level: "CAN" control mode |
| 2040 | TabPos\|MCI | Application: Table positioning<br>I/O level: "MCI" control mode |
| 3000 | SwitchPos\|Free | Application: Switch-off positioning<br>I/O level: Interconnection has been changed. |
| 3010 | SwitchPos\|Terminal0 | Application: Switch-off positioning<br>I/O level: "Terminals 0" control mode |

| Parameter \| Name: **C00008 \| Original application\|control source** | | | Data type: UNSIGNED_16 Index: $24567_d$ = $5FF7_h$ |
|---|---|---|---|
| 3012 | SwitchPos\|Terminal 2 | | Application: Switch-off positioning<br>I/O level: "Terminal 2" control mode |
| 3014 | SwitchPos\|Terminal 11 | | Application: Switch-off positioning<br>I/O level: "Terminal 11" control mode |
| 3016 | SwitchPos\|Terminal 16 | | Application: Switch-off positioning<br>I/O level: "Terminal 16" control mode |
| 3020 | SwitchPos\|Keypad | | Application: Switch-off positioning<br>I/O level: "Keypad" control mode |
| 3021 | SwitchPos\|PC | | Application: Switch-off positioning<br>I/O level: "PC" control mode |
| 3030 | SwitchPos\|CAN | | Application: Switch-off positioning<br>I/O level: "CAN" control mode |
| 3040 | SwitchPos\|MCI | | Application: Switch-off positioning<br>I/O level: "MCI" control mode |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00010**

| Parameter \| Name: **C00010 \| AIN1: Characteristic** | | | Data type: INTEGER_16 Index: $24565_d$ = $5FF5_h$ |
|---|---|---|---|
| | | | ▸ Analog terminals: Signal adaptation via characteristic |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00010/1 | 0.00 % | | AIN1: (+y0) = min |
| C00010/2 | 0.00 % | | AIN1: (+x0) = Dead band |
| C00010/3 | 0.00 % | | AIN1: (-y0) = (-min) |
| C00010/4 | 0.00 % | | AIN1: (-x0) = (-Dead band) |
| C00010/5 | 100.00 % | | AIN1: (+ymax) |
| C00010/6 | 100.00 % | | AIN1: (+xmax) |
| C00010/7 | 100.00 % | | AIN1: (-ymax) |
| C00010/8 | 100.00 % | | AIN1: (-xmax) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00011**

Parameter | Name:
**C00011 | Appl.: Reference speed**

Data type: UNSIGNED_16
Index: $24564_d$ = $5FF4_h$

Setting the reference speed
- In the controller, all speed-related signals are processed to one reference variable in percent.
- Set a reference speed here that corresponds to 100 %.
- The frequency that corresponds to the set reference speed is displayed in C00059.

**Note:**
This is not a maximum limitation!
All values in percent in the controller may be in a range of 0 ... 199.99 %.

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 50 | rpm | 60000 | **1500 rpm** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00012**

| Parameter \| Name: **C00012 \| Accel. time - main setpoint** | | | Data type: UNSIGNED_32 Index: $24563_d$ = $5FF3_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Acceleration time of the ramp generator for the main speed setpoint<br>• Generally, this ramp generator is used for all speed-controlled technology applications. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **2.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00013**

| Parameter \| Name: **C00013 \| Decel. time - main setpoint** | | | Data type: UNSIGNED_32 Index: $24562_d$ = $5FF2_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Deceleration time of the ramp generator for the main speed setpoint<br>• Generally, this ramp generator is used for all speed-controlled technology applications. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **2.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00015**

| Parameter \| Name: **C00015 \| VFC: V/f base frequency** | | | Data type: UNSIGNED_16 Index: $24560_d$ = $5FF0_h$ |
|---|---|---|---|
| V/f base frequency for V/f characteristic control (VFCplus) and V/f control (VFCplus+encoder)<br>• The motor voltage increases linearly with the frequency until the base frequency is reached. From this value on, the motor voltage remains constant, the speed increases and the maximum torque decreases.<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 7.5 | Hz | 2600.0 | **50.0 Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00016**

| Parameter \| Name: **C00016 \| VFC: Vmin boost** | | | Data type: UNSIGNED_16 Index: $24559_d$ = $5FEF_h$ |
|---|---|---|---|
| Boost of the V/f voltage characteristic in the range of small speeds or frequencies with V/f characteristic control (VFCplus) and V/f control (VFCplus+encoder)<br>• This may increase the starting torque.<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well.<br>▸ Motor control (MCTRL): Setting the Vmin boost | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **1.60 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00018**

| Parameter \| Name: **C00018 \| Switching frequency** | Data type: UNSIGNED_8 Index: $24557_d$ = $5FED_h$ |
|---|---|

Selection of the pulse width modulated switching frequency transferred from the inverter to the motor
- Select between an ideal setting for the drive which provides smooth running, and an optimal setting with regard to the inverter which keeps its losses to a minimum (min. Pv).
- Both possibilities offer fixed and variable switching frequencies.
- When a variable switching frequency is selected, the switching frequency may change as a function of the load and rotational frequency.

▶ Selection of switching frequency

| Selection list (Lenze setting printed in bold) | |
|---|---|
| 1 | 4 kHz var./drive-optimised |
| **2** | **8 kHz var./drive-optimised** |
| 3 | 16 kHz var./drive-optimised |
| 5 | 2 kHz constant/drive-optimised |
| 6 | 4 kHz constant/drive-optimised |
| 7 | 8 kHz constant/drive-optimised |
| 8 | 16 kHz constant/drive-optimised |
| 11 | 4 kHz var./min. Pv |
| 12 | 8 kHz var./min. Pv |
| 13 | 16 kHz var./min. Pv |
| 15 | 2 kHz constant/min. Pv |
| 16 | 4 kHz constant/min. Pv |
| 17 | 8 kHz constant/min. Pv |
| 18 | 16 kHz constant/min. Pv |
| 21 | 8 kHz var./drive-opt./4 kHz min |
| 22 | 16 kHz var./drive-opt./4 kHz min |
| 23 | 16 kHz var./drive-opt./8 kHz min |
| 31 | 8 kHz var./min. Pv/4 kHz min |
| 32 | 16 kHz var./min. Pv/4 kHz min |
| 33 | 16 kHz var./min. Pv/8 kHz min |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00019**

| Parameter \| Name: **C00019 \| Auto DCB: Threshold** | Data type: UNSIGNED_16 Index: $24556_d$ = $5FEC_h$ |
|---|---|

Setpoint speed  threshold for automatic DC injection braking
- For speed setpoints with values below the thresholds a DC current is injected or the motor is not supplied with current, depending on the setting.

▶ DC-injection braking

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0 | rpm | 60000 | **3 rpm** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

### C00020

| Parameter \| Name: C00020 \| AIN2: Characteristic | | Data type: INTEGER_16 Index: $24555_d$ = $5FEB_h$ |
|---|---|---|

▶ Analog terminals: Signal adaptation via characteristic

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0.00 | % | 199.99 | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00020/1 | 0.00 % | AIN2: (+y0) = min |
| C00020/2 | 0.00 % | AIN2: (+x0) = Dead band |
| C00020/3 | 0.00 % | AIN2: (-y0) = (-min) |
| C00020/4 | 0.00 % | AIN2: (-x0) = (-Dead band) |
| C00020/5 | 100.00 % | AIN2: (+ymax) |
| C00020/6 | 100.00 % | AIN2: (+xmax) |
| C00020/7 | 100.00 % | AIN2: (-ymax) |
| C00020/8 | 100.00 % | AIN2: (-xmax) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

### C00021

| Parameter \| Name: C00021 \| Slip comp. | Data type: INTEGER_16 Index: $24554_d$ = $5FEA_h$ |
|---|---|

Slip compensation for V/f characteristic control (VFCplus) and sensorless vector control (SLVC)
- A higher slip compensation results in a higher increase in frequency and voltage when the machine is under load.
- After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well.

▶ Motor control (MCTRL): Optimising the operational performance by slip compensation

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| -100.00 | % | 100.00 | **2.67 %** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

### C00022

| Parameter \| Name: C00022 \| Imax in motor mode | Data type: UNSIGNED_16 Index: $24553_d$ = $5FE9_h$ |
|---|---|

Maximum current in motor mode for all motor control modes

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.00 | A | 655.35 | **47.00 A** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

### C00023

| Parameter \| Name: C00023 \| Imax in generator mode | Data type: INTEGER_16 Index: $24552_d$ = $5FE8_h$ |
|---|---|

Maximum current in generator mode for all motor control modes
- 100 % ≡ Imax in motor mode (C00022)

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.00 | % | 100.00 | **100.00 %** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00024**

| Parameter \| Name: **C00024 \| LS_DriveInterface: bNActCompare** | Data type: INTEGER_16 Index: $24551_d$ = $5FE7_h$ |
|---|---|

Threshold for the actual speed comparison
- This parameter serves to set a threshold that is compared with the actual speed value.
- If the value falls below this threshold, the *bNactCompare* output sets the LS_DriveInterface system block to TRUE.
- Switching hysteresis = +1 %

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0.00 | % | 199.99 | **0.00 %** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00026**

| Parameter \| Name: **C00026 \| AINx: Offset** | Data type: INTEGER_16 Index: $24549_d$ = $5FE5_h$ |
|---|---|

Offset for analog inputs

▸ Analog terminals

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00026/1 | 0.00 % | | AIN1: Offset |
| C00026/2 | 0.00 % | | AIN2: Offset |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00027**

| Parameter \| Name: **C00027 \| AINx: Gain** | Data type: INTEGER_32 Index: $24548_d$ = $5FE4_h$ |
|---|---|

Gain for analog inputs

▸ Analog terminals

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -100.0000 | | 100.0000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00027/1 | 1.0000 | | AIN1: Gain |
| C00027/2 | 1.0000 | | AIN2: Gain |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C00028**

| Parameter \| Name: **C00028 \| AINx: Input voltage** | Data type: INTEGER_16 Index: $24547_d$ = $5FE3_h$ |
|---|---|

Display of the input voltage at the analog inputs

▸ Analog terminals

| **Display range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -10.00 | V | 10.00 | |
| **Subcodes** | | | **Info** |
| C00028/1 | | | AIN1: Input voltage |
| C00028/2 | | | AIN2: Input voltage |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00029**

| Parameter \| Name: **C00029 \| AINx: Input current** | | | Data type: INTEGER_16 Index: $24546_d$ = $5FE2_h$ |
|---|---|---|---|

Display of the input current at the analog inputs
- When the corresponding analog input is configured for current measurement (C00034/x = 1 or 2).
- When C00034/x is set = 2 (4 ... 20 mA), 0 ... 16 mA is displayed.

▶ Analog terminals

| **Display range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0.00 | mA | 20.00 | |
| **Subcodes** | | | **Info** |
| C00029/1 | | | AIN1: Input current |
| C00029/2 | | | AIN2: Input current |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00033**

| Parameter \| Name: **C00033 \| AINx: Output value** | | | Data type: INTEGER_16 Index: $24542_d$ = $5FDE_h$ |
|---|---|---|---|

Display of the output value in percent of the analog input amplifier
- 100 % ≡ 16384 ≡ +10 V / +20 mA

▶ Analog terminals

| **Display range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00033/1 | | | AIN1: Output value |
| C00033/2 | | | AIN2: Output value |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00034**

| Parameter \| Name: **C00034 \| AINx: Configuration** | | | Data type: UNSIGNED_8 Index: $24541_d$ = $5FDD_h$ |
|---|---|---|---|

Configuration of the analog inputs for current or voltage measurement

▶ Analog terminals

| **Selection list** | | **Info** |
|---|---|---|
| 0 | -10V..+10V | Input signal is the voltage signal -10 V ... +10 V<br>• -10 V ... +10 V ≡ -100 % ... +100 % |
| 1 | 0mA..20mA | Input signal is the current signal 0 mA ... 20 mA<br>• 0 mA ... 20 mA ≡ 0 % ... +100 % |
| 2 | 4mA..20mA | Input signal is the current signal 4 mA ... 20 mA<br>• 4 mA ... 20 mA ≡ 0 % ... +100 %<br>• The current loop is monitored for open circuit (I < 4 mA) by the device. |
| **Subcodes** | **Lenze setting** | **Info** |
| C00034/1 | 0: -10V..+10V | AIN1: Config. |
| C00034/2 | 0: -10V..+10V | AIN2: Config. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00036**

| Parameter \| Name: **C00036 \| DC braking: Current** | | | Data type: INTEGER_16 Index: $24539_d$ = $5FDB_h$ |
|---|---|---|---|

Braking current in [%] based on rated device current (C00098)

▶ DC-injection braking

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0.00 | % | 200.00 | **50.00 %** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00039**

| Parameter \| Name:<br>**C00039 \| Fixed setpoint x (L_NSet_1 n-Fix)** | | Data type: INTEGER_16<br>Index: $24536_d$ = $5FD8_h$ |
|---|---|---|
| The L_NSet_1 FB: Fixed speed setpoints (JOG values) for the setpoint generator | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -199.99 | % | 199.99 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00039/1 | 40.00 % | Fixed setpoint 1 |
| C00039/2 | 60.00 % | Fixed setpoint 2 |
| C00039/3 | 80.00 % | Fixed setpoint 3 |
| C00039/4 | 0.00 % | Fixed setpoint 4 |
| C00039/5 | 0.00 % | Fixed setpoint 5 |
| C00039/6 | 0.00 % | Fixed setpoint 6 |
| C00039/7 | 0.00 % | Fixed setpoint 7 |
| C00039/8 | 0.00 % | Fixed setpoint 8 |
| C00039/9 | 0.00 % | Fixed setpoint 9 |
| C00039/10 | 0.00 % | Fixed setpoint 10 |
| C00039/11 | 0.00 % | Fixed setpoint 11 |
| C00039/12 | 0.00 % | Fixed setpoint 12 |
| C00039/13 | 0.00 % | Fixed setpoint 13 |
| C00039/14 | 0.00 % | Fixed setpoint 14 |
| C00039/15 | 0.00 % | Fixed setpoint 15 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00050**

| Parameter \| Name:<br>**C00050 \| MCTRL: Speed setpoint** | | Data type: INTEGER_32<br>Index: $24525_d$ = $5FCD_h$ |
|---|---|---|
| Display of the speed setpoint at the speed setpoint input of the motor control | | |
| **Display range** (min. value \| unit \| max. value) | | |
| -60000 | rpm | 60000 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00051**

| Parameter \| Name:<br>**C00051 \| MCTRL: Actual speed value** | | Data type: INTEGER_32<br>Index: $24524_d$ = $5FCC_h$ |
|---|---|---|
| Display of the actual speed value of the motor shaft<br><br>**Note:**<br>The displayed value only corresponds to the real actual speed value of the motor shaft if an encoder is connected to the motor and the evaluation of the feedback signal has been set correctly ("Closed loop" operation).<br>In case of operation without speed feedback, the signal is calculated from the motor control and thus may not correspond to the real actual speed. | | |
| **Display range** (min. value \| unit \| max. value) | | |
| -60000 | rpm | 60000 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C00052

| Parameter \| Name: **C00052 \| Motor voltage** | | | Data type: UNSIGNED_16 Index: $24523_d$ = $5FCB_h$ |
|---|---|---|---|
| Display of the current motor voltage/output voltage of the inverter | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | V | 1000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00053

| Parameter \| Name: **C00053 \| DC-bus voltage** | | | Data type: UNSIGNED_16 Index: $24522_d$ = $5FCA_h$ |
|---|---|---|---|
| Display of the current DC-bus voltage | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | V | 1000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00054

| Parameter \| Name: **C00054 \| Motor current** | | | Data type: UNSIGNED_16 Index: $24521_d$ = $5FC9_h$ |
|---|---|---|---|
| Display of the current motor current/output current of the inverter | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | A | 300.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00055

| Parameter \| Name: **C00055 \| Actual values** | | | Data type: INTEGER_16 Index: $24520_d$ = $5FC8_h$ |
|---|---|---|---|
| Actual value of the encoder/feedback system<br>**Note:**<br>When a single-track feedback has been selected for the HTL encoder (C00115 = 1 or 3), the sign of the actual speed value is created from the sign of the speed setpoint. In C00055/1 und C00055/2, a positive speed is always displayed.<br>▸ Encoder/feedback system | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -32767 | rpm | 32767 | |
| **Subcodes** | | | **Info** |
| C00055/1 | | | Actual value - HTL encoder FreqIn12 |
| C00055/2 | | | Actual value - HTL encoder FreqIn67 |
| C00055/3 | | | Actual value - MultiEncoder TTL |
| C00055/4 | | | Actual value - resolver |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00056**

| Parameter \| Name: **C00056 \| Torque** | | | Data type: INTEGER_32 Index: $24519_d$ = $5FC7_h$ |
|---|---|---|---|
| Display of the current torque | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -65000.00 | Nm | 65000.00 | |
| **Subcodes** | | | **Info** |
| C00056/1 | | | Torque setpoint<br>• Only in case of sensorless vector control (SLVC) and servo control (SC). |
| C00056/2 | | | Actual torque value<br>• Estimated actual torque for all motor control modes. |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00057**

| Parameter \| Name: **C00057 \| Maximum torque** | | | Data type: UNSIGNED_32 Index: $24518_d$ = $5FC6_h$ |
|---|---|---|---|
| Display of the maximum torque to be generated by the motor<br>• The maximum torque to be generated by the motor depends on various factors, e.g. on Imax in motor mode (C00022) and the motor type used. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | Nm | 65000.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00058**

| Parameter \| Name: **C00058 \| Output frequency** | | | Data type: INTEGER_32 Index: $24517_d$ = $5FC5_h$ |
|---|---|---|---|
| Display of the current output frequency | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -1300.00 | Hz | 1300.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00059**

| Parameter \| Name: **C00059 \| Appl.: Reference frequency C11** | | | Data type: UNSIGNED_32 Index: $24516_d$ = $5FC4_h$ |
|---|---|---|---|
| Display of the field frequency which corresponds to the reference speed set in C00011. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | Hz | 1300.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00060**

| Parameter \| Name: **C00060 \| Motor rotor position** | | | Data type: UNSIGNED_16 Index: $24515_d$ = $5FC3_h$ |
|---|---|---|---|
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 2047 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00061**

| Parameter \| Name: **C00061 \| Heatsink temperature** | | | Data type: INTEGER_16 Index: $24514_d$ = $5FC2_h$ |
|---|---|---|---|
| Display of the current heatsink temperature | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -50 | °C | 150 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00062

| Parameter \| Name: **C00062 \| Temperature inside the controller** | | | Data type: INTEGER_16 Index: $24513_d$ = $5FC1_h$ |
|---|---|---|---|
| From version 02.00.00 Display of the current temperature inside the controller | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -200 | °C | 200 | |
| **Subcodes** | | | **Info** |
| C00062/1 | | | Interior temperature of CU |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00063

| Parameter \| Name: **C00063 \| Motor temperature** | | | Data type: INTEGER_16 Index: $24512_d$ = $5FC0_h$ |
|---|---|---|---|
| Display of the current motor temperature | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -200 | °C | 250 | |
| **Subcodes** | | | **Info** |
| C00063/1 | | | Motor temperature |
| C00063/2 | | | Motor temperature via resolver |
| C00063/3 | | | Motor temperature via MultiEncoder |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00064

| Parameter \| Name: **C00064 \| Device utilisation (Ixt)** | | | Data type: INTEGER_16 Index: $24511_d$ = $5FBF_h$ |
|---|---|---|---|
| Display of the device utilisation Ixt in different time resolutions<br>• If the value displayed here exceeds the threshold set in C00123, the fault message "OC5: Device overload (Ixt)" is output and the fault response set in C00604 is executed (default setting: "Warning"). | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 250.00 | |
| **Subcodes** | | | **Info** |
| C00064/1 | | | Device utilisation (Ixt)<br>• Maximum value of the pulse utilisation (C00064/2) and permanent utilisation (C00064/3). |
| C00064/2 | | | Device utilisation (Ixt) 15s<br>• Pulse utilisation over the last 15 seconds (only for loads >160 %). |
| C00064/3 | | | Device utilisation (Ixt) 3 min<br>• Permanent utilisation over the last 3 minutes. |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00065

| Parameter \| Name: **C00065 \| Supply voltage 24V** | | | Data type: INTEGER_16 Index: $24510_d$ = $5FBE_h$ |
|---|---|---|---|
| Display of the 24V supply voltage for the supply of the control electronics<br><br>**Note:**<br>The 24 V supply for the control electronics is either provided by an external supply or by the controller itself if it is connected to the mains voltage. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.0 | V | 3276.7 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

## C00066

| Parameter \| Name: **C00066 \| Thermal motor load (I²xt)** | | | Data type: INTEGER_16 Index: $24509_d$ = $5FBD_h$ |
|---|---|---|---|
| Display of the thermal motor load, sensorlessly determined using a motor model<br>• If the value displayed here exceeds the motor overload setting (C00120), the fault message "OC6: Thermal motor overload (I2xt)" is output and the fault response set in C00606 is executed (default setting: "Warning"). | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00070

| Parameter \| Name: **C00070 \| Vp speed controller** | | | Data type: UNSIGNED_16 Index: $24505_d$ = $5FB9_h$ |
|---|---|---|---|
| Amplification factor Vp of the speed controller for different motor control modes | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | | 600.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00070/1 | 15.00 | | SLVC: Vp speed controller |
| C00070/2 | 6.00 | | SC: Vp speed controller |
| C00070/3 | 3.00 | | SLPSM: Vp speed controller |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

## C00071

| Parameter \| Name: **C00071 \| Ti speed controller** | | | Data type: UNSIGNED_16 Index: $24504_d$ = $5FB8_h$ |
|---|---|---|---|
| Reset time Ti of the speed controller for different motor control modes | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0 | ms | 6000.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00071/1 | 100.0 ms | | SLVC: Ti speed controller |
| C00071/2 | 50.0 ms | | SC: Ti speed controller |
| C00071/3 | 100.0 ms | | SLPSM: Ti speed controller |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

## C00072

| Parameter \| Name: **C00072 \| SC: Tdn speed controller** | | | Data type: UNSIGNED_16 Index: $24503_d$ = $5FB7_h$ |
|---|---|---|---|
| Differential time constant Tdn of the speed controller for servo control (SC) and sensorless control for synchronous motors (SLPSM) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | ms | 3.00 | **0.00 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

## C00073

| Parameter \| Name: **C00073 \| Imax/M controller gain** | | | Data type: UNSIGNED_16 Index: $24502_d$ = $5FB6_h$ |
|---|---|---|---|
| Amplification factor Vp of certain controllers for different motor control modes | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | | 16.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00073/1 | 0.25 | | VFC: Vp Imax controller<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. |
| C00073/2 | 1.25 | | SLVC: Vp torque controller |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

## C00074

| Parameter \| Name: **C00074 \| Reset time Imax/M controller** | | | Data type: UNSIGNED_16 Index: $24501_d$ = $5FB5_h$ |
|---|---|---|---|
| Reset time Ti of certain controllers for different motor control modes | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 9990 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00074/1 | 65 ms | | VFC: Ti Imax controller |
| C00074/2 | 30 ms | | SLVC: Ti torque controller |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

## C00075

| Parameter \| Name: **C00075 \| Vp current controller** | | | Data type: UNSIGNED_16 Index: $24500_d$ = $5FB4_h$ |
|---|---|---|---|
| Gain factor Vp of the current controller for servo control (SC) and certain inverter functions (parameter identification, flying restart circuit)<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | V/A | 500.00 | **7.00 V/A** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

## C00076

| Parameter \| Name: **C00076 \| Ti current controller** | | | Data type: UNSIGNED_16 Index: $24499_d$ = $5FB3_h$ |
|---|---|---|---|
| Reset time Ti of the current controller for servo control (SC) and certain inverter functions (parameter identification, flying restart circuit)<br>• After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | ms | 500.00 | **10.61 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00077**

| Parameter \| Name: **C00077 \| SC: Vp field controller** | | | Data type: UNSIGNED_16 Index: $24498_d$ = $5FB2_h$ |
|---|---|---|---|

Gain factor Vp of the field controller for servo control (SC)
- After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0.00 | | 500.00 | **12.80** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100

**C00078**

| Parameter \| Name: **C00078 \| SC: Tn field controller** | | | Data type: UNSIGNED_16 Index: $24497_d$ = $5FB1_h$ |
|---|---|---|---|

Reset time Tn of the field controller for servo control (SC)
- After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0.1 | ms | 6000.0 | **256.0 ms** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10

**C00079**

| Parameter \| Name: **C00079 \| SC: Settings** | | Data type: UNSIGNED_8 Index: $24496_d$ = $5FB0_h$ |
|---|---|---|

Configuration of different options for servo control (SC) and sensorless control for synchronous motors (SLPSM)

| **Selection list** | | |
|---|---|---|
| 0 | Off | |
| 1 | On | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00079/1 | 0: Off | SC: Current controller - feedforward control<br>• Feedforward control/decoupling network of the current controller. |
| C00079/2 | 1: On | SC: Adapt. field weakening controller<br>• Speed-dependent adaptive field weakening controller. |
| C00079/3 | 0: Off | SC: n-ctrl anti-wind-up<br>• "Anti-wind-up" effect of the speed controller in case of an output voltage limitation in the field weakening range. |
| C00079/4 | 0: Off | Field weakening for synchronous motors<br>• From version 02.00.00 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00080**

| Parameter \| Name: **C00080 \| Override point of field weakening** | | | Data type: INTEGER_16 Index: $24495_d$ = $5FAF_h$ |
|---|---|---|---|

Offset of the override point of field weakening
- In the V/f characteristic control mode (VFCplus), the stall protection function or the max. permissible current in the field weakening range can be adapted.

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| -500 | Hz | 500 | **0 Hz** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00081**

| Parameter \| Name:<br>**C00081 \| Rated motor power** | | | Data type: UNSIGNED_16<br>Index: $24494_d$ = $5FAE_h$ |
|---|---|---|---|
| This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.<br><br>**Note:**<br>It is mandatory to give the rated motor power for the sensorless vector control (SLVC) and the servo control (SC). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | kW | 500.00 | **11.00 kW** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00082**

| Parameter \| Name:<br>**C00082 \| Motor rotor resistance** | | | Data type: UNSIGNED_32<br>Index: $24493_d$ = $5FAD_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | mOhm | 200000 | **276 mOhm** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00083**

| Parameter \| Name:<br>**C00083 \| Motor rotor time constant** | | | Data type: UNSIGNED_16<br>Index: $24492_d$ = $5FAC_h$ |
|---|---|---|---|
| Display of the rotor time constant of the motor<br>• This value is calculated from the rotor resistance and the rotor inductance (leakage and magnetising inductance). | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 32767 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00084**

| Parameter \| Name:<br>**C00084 \| Motor stator resistance** | | | Data type: UNSIGNED_32<br>Index: $24491_d$ = $5FAB_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | mOhm | 200000 | **330 mOhm** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00085**

| Parameter \| Name:<br>**C00085 \| Motor stator leakage inductance** | | | Data type: UNSIGNED_16<br>Index: $24490_d$ = $5FAA_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | mH | 650.00 | **3.50 mH** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00087**

| Parameter \| Name: **C00087 \| Rated motor speed** | | | Data type: UNSIGNED_16 Index: $24488_d$ = $5FA8_h$ |
|---|---|---|---|

This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.

**Note:**
It is mandatory to give the rated motor speed for the sensorless vector control (SLVC) and the servo control (SC).

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 50 | rpm | 60000 | **1460 rpm** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00088**

| Parameter \| Name: **C00088 \| Rated motor current** | | | Data type: UNSIGNED_16 Index: $24487_d$ = $5FA7_h$ |
|---|---|---|---|

This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0.20 | A | 320.00 | **21.00 A** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100

**C00089**

| Parameter \| Name: **C00089 \| Rated motor frequency** | | | Data type: UNSIGNED_16 Index: $24486_d$ = $5FA6_h$ |
|---|---|---|---|

This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.

**Note:**
It is mandatory to give the rated motor frequency for the sensorless vector control (SLVC) and the servo control (SC).

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 1 | Hz | 1000 | **50 Hz** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00090**

| Parameter \| Name: **C00090 \| Rated motor voltage** | | | Data type: UNSIGNED_16 Index: $24485_d$ = $5FA5_h$ |
|---|---|---|---|

This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0 | V | 5000 | **400 V** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1

**C00091**

| Parameter \| Name: **C00091 \| Motor cosine phi** | | | Data type: UNSIGNED_8 Index: $24484_d$ = $5FA4_h$ |
|---|---|---|---|

This value can be obtained from the motor nameplate. After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically.

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0.40 | | 1.00 | **0.85** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100

**C00092**

| Parameter \| Name: **C00092 \| Motor magnetising inductance** | | | Data type: UNSIGNED_16 Index: $24483_d$ = $5FA3_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | mH | 6500.0 | **81.0 mH** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00093**

| Parameter \| Name: **C00093 \| Power section identification** | | | Data type: UNSIGNED_16 Index: $24482_d$ = $5FA2_h$ |
|---|---|---|---|
| Display of the identification of the detected power section of the controller | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00095**

| Parameter \| Name: **C00095 \| Motor magnetising current** | | | Data type: UNSIGNED_16 Index: $24480_d$ = $5FA0_h$ |
|---|---|---|---|
| After the motor to be used has been selected from the motor catalogue, the suitable value can be entered automatically. An automatic detection via the motor parameter identification is possible as well. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | A | 320.00 | **8.50 A** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00097**

| Parameter \| Name: **C00097 \| Rated motor torque** | | | Data type: UNSIGNED_32 Index: $24478_d$ = $5F9E_h$ |
|---|---|---|---|
| Display of the rated motor torque<br>• The value displayed here is calculated from different parameters, e.g. the maximum current set in C00022. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | Nm | 65535.00 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00098**

| Parameter \| Name: **C00098 \| Device rated current** | | | Data type: UNSIGNED_16 Index: $24477_d$ = $5F9D_h$ |
|---|---|---|---|
| Display of the rated inverter current which is defined by the integrated power section. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.0 | A | 6000.0 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00099**

| Parameter \| Name: **C00099 \| Firmware version** | Data type: VISIBLE_STRING Index: $24476_d$ = $5F9C_h$ |
|---|---|
| Display of the firmware version of the device as string | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 12 | |

**C00100**

| Parameter \| Name: **C00100 \| Firmware version** | | | Data type: UNSIGNED_8 Index: $24475_d$ = $5F9B_h$ |
|---|---|---|---|
| Display of the firmware version of the device, divided into subsections. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 99 | |
| **Subcodes** | | | **Info** |
| C00100/1 | | | Firmware version - main version |
| C00100/2 | | | Firmware version - subversion |
| C00100/3 | | | Firmware version - release |
| C00100/4 | | | Firmware version - build |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00101**

| Parameter \| Name: **C00101 \| Add. accel. time x** | | | Data type: UNSIGNED_32 Index: $24474_d$ = $5F9A_h$ |
|---|---|---|---|
| The L_NSet_1: FB: Additional acceleration times for the main setpoint<br>• The additional acceleration times set here can be selected via the binary inputs *bTI1* ... *bTI8* of the L_NSet_1 FB. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00101/1 | 0.000 s | | Add. accel. time 1 ... 15 |
| C00101/... | | | |
| C00101/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00103**

| Parameter \| Name: **C00103 \| Add. decel. time x** | | | Data type: UNSIGNED_32 Index: $24472_d$ = $5F98_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Additional deceleration times for the main setpoint<br>• The additional deceleration times set here can be selected via the binary inputs *bTI1* ... *bTI8* of the L_NSet_1 FB. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00103/1 | 0.000 s | | Add. decel. time 1 ... 15 |
| C00103/... | | | |
| C00103/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00104**

| Parameter \| Name: **C00104 \| Position control with quick stop** | | | Data type: UNSIGNED_16 Index: $24471_d$ = $5F97_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** (☑ = bit set) | | | |
| Bit 0 ☐ | Standstill position-controlled | | |
| Bit 1 ☐ | Ramp position-controlled | | |
| Bit 2 ☐ | Reserved | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C00104/1 | 0 | Position control at quick stop | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00105**

| Parameter \| Name: **C00105 \| Decel. time - quick stop** | | | Data type: UNSIGNED_32 Index: $24470_d$ = $5F96_h$ |
|---|---|---|---|
| The set deceleration time determines the ramp slope at quick stop | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.900 | **2.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00106**

| Parameter \| Name: **C00106 \| Auto DCB: Hold time** | | | Data type: UNSIGNED_32 Index: $24469_d$ = $5F95_h$ |
|---|---|---|---|
| Hold time of the automatic DC injection brake<br>• The DC injection brake is applied for the time set here if the value falls below the speed setpoint set in C00019. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.000 | **0.500 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00107**

| Parameter \| Name: **C00107 \| DC braking: Hold time** | | | Data type: UNSIGNED_32 Index: $24468_d$ = $5F94_h$ |
|---|---|---|---|
| Maximum hold time of the manual DC injection brake<br>• A time can be set here after which the DC injection brake is switched off automatically to prevent the motor from thermal overload. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.000 | **999.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00114**

| Parameter \| Name: **C00114 \| DigInX: Inversion** | | | Data type: UNSIGNED_16 Index: $24461_d$ = $5F8D_h$ |
|---|---|---|---|
| The polarity of each digital input of the device can be inverted via this bit field.<br>▸ Digital input terminals | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded:** (☑ = bit set) | | | **Info** |
| Bit 0 ☐ | Invert. DI1 | | Inversion of digital input 1 |
| Bit 1 ☐ | Invert. DI2 | | Inversion of digital input 2 |
| Bit 2 ☐ | Invert. DI3 | | Inversion of digital input 3 |
| Bit 3 ☐ | Invert. DI4 | | Inversion of digital input 4 |
| Bit 4 ☐ | Invert. DI5 | | Inversion of digital input 5 |
| Bit 5 ☐ | Invert. DI6 | | Inversion of digital input 6 |
| Bit 6 ☐ | Invert. DI7 | | Inversion of digital input 7 |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Invert CInh | | Inversion of digital input RFR (controller enable) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00115**

| Parameter \| Name: **C00115 \| DI 1/2 & 6/7: Fct.** | | Data type: UNSIGNED_8 Index: $24460_d$ = $5F8C_h$ |
|---|---|---|
| Function assignment of the digital terminals DI1/2 and DI6/7 | | |
| | | ▸ Digital input terminals: Function assignment |
| **Selection list** | | **Info** |
| 0 | DI1(6)=In / DI2(7)=In | DI1/6 = digital input<br>DI2/7 = digital input |
| 1 | DI1(6)=FreqIn / DI2(7)=In | DI1/6 = 1-track frequency input<br>DI2/7 = digital input |
| 2 | DI1(6)&DI2(7)=FreqIn (2-track) | DI1/6 and DI2/7 = 2-track frequency input |
| 3 | DI1(6)=FreqIn / DI2(7)=direction | DI1/6 = 1-track frequency input<br>DI2/7 = specification of direction |
| 4 | DI1(6)=CountIn / DI2(7)=In | DI1/6 = counter input<br>DI2/7 = digital input |
| **Subcodes** | **Lenze setting** | **Info** |
| C00115/1 | 0: DI1(6)=In / DI2(7)=In | Function assignment of DI1 and DI2 |
| C00115/2 | 0: DI1(6)=In / DI2(7)=In | Function assignment of DI6 and DI7 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00117**

| Parameter \| Name: **C00117 \| Status of brake output BD** | | Data type: UNSIGNED_8 Index: $24458_d$ = $5F8A_h$ |
|---|---|---|
| Status message of brake output | | |
| **Selection list** (read only) | | **Info** |
| 0 | Inactive | Brake output is inactive |
| 1 | Active | Brake output is active |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00118**

| Parameter \| Name: **C00118 \| DigOutX: Inversion** | | | Data type: UNSIGNED_8 Index: $24457_d$ = $5F89_h$ |
|---|---|---|---|
| The polarity of each digital output of the device can be inverted via this bit field. | | | |
| | | | ▸ Digital output terminals |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x00** (decimal: 0) |
| **Value is bit-coded:** (☑ = bit set) | | | **Info** |
| Bit 0 ☐ | Relay inverted | | Relay inversion |
| Bit 1 ☐ | DO1 inverted | | Inversion of digital output 1 |
| Bit 2 ☐ | DO2 inverted | | Inversion of digital output 2 |
| Bit 3 ☐ | DO3 inverted | | Inversion of digital output 3 |
| Bit 4 ☐ | High current inverted | | Inversion of high-current output |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00120**

| Parameter \| Name: **C00120 \| Setting of motor overload (I²xt)** | | | Data type: INTEGER_16 Index: $24455_d$ = $5F87_h$ |
|---|---|---|---|
| Operating threshold for the "OC6: Motor overload (I²xt)" error message<br>• The response for reaching the threshold can be selected in C00606.<br>• The current thermal motor load is displayed in C00066. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 250.00 | **100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00121**

| Parameter \| Name: **C00121 \| Warning threshold motor temperature** | | | Data type: UNSIGNED_16 Index: $24454_d$ = $5F86_h$ |
|---|---|---|---|
| Operating threshold for error message "Motor: Temperature at Res./Enc. > C121"<br>• The response for reaching the threshold can be selected in C00583.<br>• The current motor temperature is displayed in C00063.<br>▶ Encoder/feedback system: Motor temperature monitoring (KTY) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | °C | 250 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00121/1 | 120 °C | | Warning threshold motor temperature resolver |
| C00121/2 | 120 °C | | Warning threshold motor temperature MultiEncoder |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00123**

| Parameter \| Name: **C00123 \| Device utilisation threshold (Ixt)** | | | Data type: INTEGER_16 Index: $24452_d$ = $5F84_h$ |
|---|---|---|---|
| Operating threshold for the "OC5: Device overload (Ixt)" error message<br>• The response for reaching the threshold can be selected in C00604.<br>• The current device utilisation is displayed in C00064. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 200.00 | **100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00129**

| Parameter \| Name: **C00129 \| Brake resistance value** | | | Data type: UNSIGNED_16 Index: $24446_d$ = $5F7E_h$ |
|---|---|---|---|
| Resistance value of the connected brake resistor<br>• The value to be entered can be obtained from the nameplate of the brake resistor. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | Ohm | 500.0 | **39.0 Ohm** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00130**

| Parameter \| Name: **C00130 \| Rated power - brake resistor** | | | Data type: UNSIGNED_16 Index: $24445_d$ = $5F7D_h$ |
|---|---|---|---|
| Rated power of the connected brake resistor<br>• The value to be entered can be obtained from the nameplate of the brake resistor. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | W | 65535 | **100 W** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00131**

| Parameter \| Name: **C00131 \| Thermal capacity - brake resistor** | | | Data type: UNSIGNED_16 Index: $24444_d$ = $5F7C_h$ |
|---|---|---|---|
| Thermal capacity of the connected brake resistor<br>• The value to be entered can be obtained from the nameplate of the brake resistor. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | kWs | 6553.5 | **10.0 kWs** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00133**

| Parameter \| Name: **C00133 \| Brake resistor utilisation** | | | Data type: UNSIGNED_16 Index: $24442_d$ = $5F7A_h$ |
|---|---|---|---|
| Display of the utilisation of the connected brake resistor | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | % | 65535 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00134**

| Parameter \| Name: **C00134 \| L_NSet_1: Ramp smoothing** | | Data type: UNSIGNED_8 Index: $24441_d$ = $5F79_h$ |
|---|---|---|
| The L_NSet_1 FB: Configuration of the ramp rounding for the main setpoint | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Off** | Ramp rounding deactivated |
| 1 | PT1 behaviour | Ramp rounding with PT1 behaviour<br>• The corresponding S-ramp time must be set in C00182. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00136**

| Parameter \| Name: **C00136 \| Communication control words** | | | Data type: UNSIGNED_16 Index: $24439_d$ = $5F77_h$ |
|---|---|---|---|
| Control words of the communication interfaces | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | SwitchOn | | |
| Bit 1 | DisableVoltage | | |
| Bit 2 | SetQuickStop | | |
| Bit 3 | EnableOperation | | |
| Bit 4 | ModeSpecific_1 | | |
| Bit 5 | ModeSpecific_2 | | |
| Bit 6 | ModeSpecific_3 | | |
| Bit 7 | ResetFault | | |
| Bit 8 | SetHalt | | |
| Bit 9 | Reserved_1 | | |
| Bit 10 | Reserved_2 | | |
| Bit 11 | LenzeSpecific_1 | | |
| Bit 12 | LenzeSpecific_2 | | |
| Bit 13 | LenzeSpecific_3 | | |
| Bit 14 | SetFail | | |
| Bit 15 | LenzeSpecific_4 | | |
| **Subcodes** | | | **Info** |
| C00136/1 | | | MCI control word<br>• Control word of the MCI communication interface (communication module) |
| C00136/2 | | | CAN control word<br>• Control word of the CAN communication interface (CAN on board) |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00137**

| Parameter \| Name: **C00137 \| Device status** | Data type: UNSIGNED_16 Index: $24438_d$ = $5F76_h$ |
|---|---|
| Display of the current device status | |

| **Selection list** (read only) | |
|---|---|
| 0 | FirmwareUpdate |
| 1 | Init |
| 2 | MotorIdent |
| 3 | ReadyToSwitchON |
| 4 | SwitchedON |
| 5 | OperationEnable |
| 6 | Warning |
| 7 | Trouble |
| 8 | Fault |
| 9 | TroubleQSP |
| 10 | SafeTorqueOff |
| 11 | SystemFail |
| 12 | Reserved_1 |
| 13 | Reserved_2 |
| 14 | Reserved_3 |
| 15 | Reserved_4 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00138**

| Parameter \| Name:<br>**C00138 \| Internal control signals** | | | Data type: UNSIGNED_16<br>Index: $24437_d$ = $5F75_h$ |
|---|---|---|---|
| Bit coded display of internal control signals of different sources | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Reserved | | |
| Bit 1 | DisableVoltage | | |
| Bit 2 | SetQuickStop | | |
| Bit 3 | EnableOperation | | |
| Bit 4 | InitFinishedOK | | |
| Bit 5 | ModeSpecific_2 | | |
| Bit 6 | ModeSpecific_3 | | |
| Bit 7 | ResetFault | | |
| Bit 8 | SetHalt | | |
| Bit 9 | FirmwareUpdate | | |
| Bit 10 | MotorIdent | | |
| Bit 11 | SetMessage | | |
| Bit 12 | SetIMP | | |
| Bit 13 | SetSystemFail | | |
| Bit 14 | SetFail | | |
| Bit 15 | SetFailQSP | | |
| **Subcodes** | | | **Info** |
| C00138/1 | | | SYS control signals |
| C00138/2 | | | MCK control signals |
| C00138/3 | | | FWM control signals |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00142**

| Parameter \| Name:<br>**C00142 \| Auto-start option** | | | Data type: UNSIGNED_8<br>Index: $24433_d$ = $5F71_h$ |
|---|---|---|---|
| Starting performance of the controller after mains connection, undervoltage, loading of the Lenze setting as well as a reset of "Trouble" or "Fault" can be parameterised individually.<br>▶ Automatic restart after mains connection/fault... | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x19** (decimal: 25) |
| **Value is bit-coded:** (☑ = bit set) | | | |
| Bit 0 ☑ | Inhibit at power-on | | |
| Bit 1 ☐ | Inhibit at trouble | | |
| Bit 2 ☐ | Inhibit at fault | | |
| Bit 3 ☑ | Inhibit at undervoltage | | |
| Bit 4 ☑ | Inhibit at Lenze setting | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00144**

| Parameter \| Name:<br>**C00144 \| Switching freq. reduct. (Temp.)** | | Data type: UNSIGNED_8<br>Index: $24431_d$ = $5F6F_h$ |
|---|---|---|
| Activation of the automatic switching frequency reduction if the temperature is too high | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | Off | Automatic switching frequency reduction deactivated |
| **1** | **On** | Automatic switching frequency reduction activated |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00148**

| Parameter \| Name:<br>**C00148 \| LS_DriveInterface: Error message config.** | | | Data type: UNSIGNED_16<br>Index: $24427_d$ = $5F6B_h$ |
|---|---|---|---|
| Selection of the device statuses for which the *bCollectedFail* group error output of SB LS_DriveInterface is to be set to TRUE. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0030** (decimal: 48) |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | SafeTorqueOff | | |
| Bit 1 ☐ | ReadyToSwitchOn | | |
| Bit 2 ☐ | SwitchedOn | | |
| Bit 3 ☐ | TroubleQSP | | |
| Bit 4 ☑ | Trouble | | |
| Bit 5 ☑ | Fault | | |
| Bit 6 ☐ | Warning | | |
| Bit 7 ☐ | ImpIsActive | | |
| Bit 8 ☐ | CInhIsActive | | |
| Bit 9 ☐ | Fail CAN_Management | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Lock bFail at TroubleQSP | | From version 02.00.00<br>If this bit is set, the *bFail* output of the SB LS_DriveInterface is also set in the "TroubleQSP" status.<br>• Advantage: Even in the "TroubleQSP" status, an error occurred before can still be recognised. |
| Bit 15 ☐ | Use 16BitFailNo. | | If this bit is set, the short 16-bit error number (*wStateDetermFailNoShort*) is also provided at the *wStateDetermFailNoLow* output of the SB LS_DriveInterface.<br>• In this case, the *wStateDetermFailNoHigh* output is "0".<br>• Advantage: The bus transfer of the error numbers is possible via a data word without changing the interconnection of the technology application. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00150**

| Parameter \| Name: **C00150 \| Status word** | | Data type: UNSIGNED_16 Index: $24425_d$ = $5F69_h$ |
|---|---|---|
| Bit coded device status word | | |
| **Display area** (min. hex value \| max. hex value) | | |
| 0x0000 | 0xFFFF | |
| **Value is bit-coded:** | | **Info** |
| Bit 0 | FreeStatusBit0 | Free status bit 0 |
| Bit 1 | PowerDisabled | Power switched off |
| Bit 2 | FreeStatusBit2 | Free status bit 2 |
| Bit 3 | FreeStatusBit3 | Free status bit 3 |
| Bit 4 | FreeStatusBit4 | Free status bit 4 |
| Bit 5 | FreeStatusBit5 | Free status bit 5 |
| Bit 6 | ActSpeedIsZero | Current speed is 0 |
| Bit 7 | ControllerInhibit | Controller is inhibited |
| Bit 8 | StatusCodeBit0 | Status code bit 0 |
| Bit 9 | StatusCodeBit1 | Status code bit 1 |
| Bit 10 | StatusCodeBit2 | Status code bit 2 |
| Bit 11 | StatusCodeBit3 | Status code bit 3 |
| Bit 12 | Warning | Warning |
| Bit 13 | Trouble | Trouble |
| Bit 14 | FreeStatusBit14 | Free status bit 14 |
| Bit 15 | FreeStatusBit15 | Free status bit 15 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | |

**C00155**

| Parameter \| Name: **C00155 \| Extended status word** | | | Data type: UNSIGNED_16 Index: $24420_d$ = $5F64_h$ |
|---|---|---|---|
| Bit coded device status word 2 | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Fail | | Error |
| Bit 1 | M_max | | Maximum torque |
| Bit 2 | I_max | | Maximum current |
| Bit 3 | PowerDisabled | | Power switched off |
| Bit 4 | Ready | | Controller is ready for operation |
| Bit 5 | ControllerInhibit | | Controller is inhibited |
| Bit 6 | Trouble | | Trouble |
| Bit 7 | InitState | | Initialisation |
| Bit 8 | CwCcw | | CW/CCW rotation |
| Bit 9 | TroubleQSP | | Quick stop due to fault is active |
| Bit 10 | SafeTorqueOff | | Safe torque off |
| Bit 11 | AplicationRunning | | Application is running |
| Bit 12 | AplParSetBit0 | | Application parameter set - bit 0 |
| Bit 13 | AplParSetBit1 | | Application parameter set - bit 1 |
| Bit 14 | Quick stop | | Quick stop is active |
| Bit 15 | Motor parameter identification | | Motor parameter identification is active |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00158**

| Parameter \| Name:<br>**C00158 \| Cause of controller inhibit** | Data type: UNSIGNED_16<br>Index: $24417_d$ = $5F61_h$ |
|---|---|
| Bit coded display of the cause/source of the controller inhibit | |

| **Display area** (min. hex value \| max. hex value) | | |
|---|---|---|
| 0x0000 | | 0xFFFF |

| **Value is bit-coded:** | |
|---|---|
| Bit 0 | Terminal controller enable |
| Bit 1 | CAN control word |
| Bit 2 | MCI control word |
| Bit 3 | SwitchOn |
| Bit 4 | Application |
| Bit 5 | Device command |
| Bit 6 | Error response |
| Bit 7 | Internal signal |
| Bit 8 | Reserved |
| Bit 9 | Reserved |
| Bit 10 | AutoStartLock |
| Bit 11 | Motor parameter identification |
| Bit 12 | Automatic brake operation |
| Bit 13 | DCB-IMP |
| Bit 14 | Reserved |
| Bit 15 | Reserved |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

**C00159**

| Parameter \| Name:<br>**C00159 \| Cause of quick stop QSP** | | | Data type: UNSIGNED_16<br>Index: $24416_d$ = $5F60_h$ |
|---|---|---|---|
| Bit coded display of the cause/source of the quick stop | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Reserved | | |
| Bit 1 | CAN control word | | |
| Bit 2 | MCI control word | | |
| Bit 3 | Reserved | | |
| Bit 4 | Application | | |
| Bit 5 | Device command | | |
| Bit 6 | Error response | | |
| Bit 7 | Internal signal | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Operating system | | |
| Bit 11 | Reserved | | |
| Bit 12 | MCK | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00160**

| Parameter \| Name:<br>**C00160 \| Status determining error (16bit)** | | | Data type: UNSIGNED_16<br>Index: $24415_d$ = $5F5F_h$ |
|---|---|---|---|
| Display of the status determining error as short 16-bit error number | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | | | **Info** |
| C00160/1 | | | Status determining error (16-bit) |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00161**

| Parameter \| Name: **C00161 \| LS_SetError_x: Error number** | | | Data type: UNSIGNED_16 Index: $24414_d$ = $5F5E_h$ |
|---|---|---|---|
| Setting of the error number for user error messages | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00161/1 | 1 | | LS_SetError_1: Error no.1 |
| C00161/2 | 2 | | LS_SetError_1: Error no.2 |
| C00161/3 | 3 | | LS_SetError_1: Error no.3 |
| C00161/4 | 4 | | LS_SetError_1: Error no.4 |
| C00161/5 | 1 | | LS_SetError_2: Error no.1 |
| C00161/6 | 2 | | LS_SetError_2: Error no.2 |
| C00161/7 | 3 | | LS_SetError_2: Error no.3 |
| C00161/8 | 4 | | LS_SetError_2: Error no.4 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00163**

| Parameter \| Name: **C00163 \| Logbook - binary elements** | | Data type: UNSIGNED_16 Index: $24412_d$ = $5F5C_h$ |
|---|---|---|
| Selection of two binary signals to be logged in the logbook | | |
| **Selection list** | | **Info** |
| 0 | No signal | |
| 1 | DI1: Input signal | |
| 2 | DI2: Input signal | |
| 3 | DI3: Input signal | |
| 4 | DI4: Input signal | |
| 5 | Controller inhibit signal | |
| 6 | Digital counter: Comparison bit | |
| 7 | CAN1 input bit 0 | |
| 8 | CAN1 input bit 1 | |
| 9 | CAN1 input bit 2 | |
| 10 | CAN1 input bit 3 | |
| 11 | CAN1 input bit 4 | |
| 12 | CAN1 input bit 5 | |
| 13 | CAN1 input bit 6 | |
| 14 | CAN1 input bit 7 | |
| 15 | CAN1 input bit 8 | |
| 16 | CAN1 input bit 9 | |
| 17 | CAN1 input bit 10 | |
| 18 | CAN1 input bit 11 | |
| 19 | CAN1 input bit 12 | |
| 20 | CAN1 input bit 13 | |
| 21 | CAN1 input bit 14 | |
| 22 | CAN1 input bit 15 | |
| 23 | CAN2 input bit 0 | |
| 24 | CAN2 input bit 1 | |
| 25 | CAN2 input bit 2 | |

| Parameter \| Name: **C00163 \| Logbook - binary elements** | | Data type: UNSIGNED_16 Index: $24412_d$ = $5F5C_h$ |
|---|---|---|
| 26 | CAN2 input bit 3 | |
| 27 | CAN2 input bit 4 | |
| 28 | CAN2 input bit 5 | |
| 29 | CAN2 input bit 6 | |
| 30 | CAN2 input bit 7 | |
| 31 | CAN2 input bit 8 | |
| 32 | CAN2 input bit 9 | |
| 33 | CAN2 input bit 10 | |
| 34 | CAN2 input bit 11 | |
| 35 | CAN2 input bit 12 | |
| 36 | CAN2 input bit 13 | |
| 37 | CAN2 input bit 14 | |
| 38 | CAN2 input bit 15 | |
| 39 | CAN3 input bit 0 | |
| 40 | CAN3 input bit 1 | |
| 41 | CAN3 input bit 2 | |
| 42 | CAN3 input bit 3 | |
| 43 | CAN3 input bit 4 | |
| 44 | CAN3 input bit 5 | |
| 45 | CAN3 input bit 6 | |
| 46 | CAN3 input bit 7 | |
| 47 | CAN3 input bit 8 | |
| 48 | CAN3 input bit 9 | |
| 49 | CAN3 input bit 10 | |
| 50 | CAN3 input bit 11 | |
| 51 | CAN3 input bit 12 | |
| 52 | CAN3 input bit 13 | |
| 53 | CAN3 input bit 14 | |
| 54 | CAN3 input bit 15 | |
| 55 | MCI word1 input bit0 | |
| 56 | MCI word1 input bit1 | |
| 57 | MCI word1 input bit2 | |
| 58 | MCI word1 input bit3 | |
| 59 | MCI Word 1 Input bit 4 | |
| 60 | MCI word1 input bit5 | |
| 61 | MCI word1 input bit6 | |
| 62 | MCI word1 input bit7 | |
| 63 | MCI word1 input bit8 | |
| 64 | MCI word1 input bit9 | |
| 65 | MCI word1 input bit10 | |
| 66 | MCI word1 input bit11 | |
| 67 | MCI word1 input bit12 | |
| 68 | MCI word1 input bit13 | |
| 69 | MCI word1 input bit14 | |

| Parameter \| Name: C00163 \| Logbook - binary elements | | Data type: UNSIGNED_16 Index: $24412_d$ = $5F5C_h$ |
|---|---|---|
| 70 | MCI word1 input bit15 | |
| 71 | MCI word2 input bit0 | |
| 72 | MCI word2 input bit1 | |
| 73 | MCI Word 2 Input bit 2 | |
| 74 | MCI word2 input bit3 | |
| 75 | MCI word2 input bit4 | |
| 76 | MCI word2 input bit5 | |
| 77 | MCI word2 input bit6 | |
| 78 | MCI word 2 input bit 7 | |
| 79 | MCI word2 input bit8 | |
| 80 | MCI word2 input bit9 | |
| 81 | MCI word2 input bit10 | |
| 82 | MCI word2 input bit11 | |
| 83 | MCI Word 2 Input bit 12 | |
| 84 | MCI word2 input bit13 | |
| 85 | MCI word2 input bit14 | |
| 86 | MCI word2 input bit15 | |
| 87 | Position controller: Limit | |
| 88 | Speed controller: Limit | |
| 89 | Speed setpoint: Limit | |
| 90 | Torque setpoint: Limit | |
| 91 | Current setpoint: Limit | |
| 92 | DC injection brake active | |
| 93 | Quick stop is active | |
| 94 | Pulse inhibit is active | |
| 95 | Controller inhibit is active | |
| 96 | Safe status active | |
| 97 | Direction of rotation ccw | |
| 98 | Actual speed = 0 | |
| 99 | L_Or_1: Out | |
| 100 | L_DFlipFlop_1: Out | |
| 101 | L_DigitalDelay_1: Out | |
| 102 | L_Compare_1: Out | |
| 103 | L_Compare_2: Out | |
| 104 | L_NSet_1: Setpoint reached | |
| 105 | L_DigitalLogic_1: Out | |
| 106 | L_SignalMonitor_b: Out1 | |
| 107 | L_SignalMonitor_b: Out2 | |
| 108 | L_SignalMonitor_b: Out3 | |
| 109 | L_SignalMonitor_b: Out4 | |
| 110 | L_PCTRL_1: act=set | |

| Parameter \| Name: C00163 \| Logbook - binary elements | | Data type: UNSIGNED_16 Index: $24412_d$ = $5F5C_h$ |
|---|---|---|
| **Subcodes** | **Lenze setting** | **Info** |
| C00163/1 | 0: No signal | Logbook - binary element 1 |
| C00163/2 | 0: No signal | Logbook - binary element 2 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00164**

| Parameter \| Name: C00164 \| Logbook - analog elements | | Data type: UNSIGNED_16 Index: $24411_d$ = $5F5B_h$ |
|---|---|---|
| Selection of an analog signal to be logged in the logbook | | |
| **Selection list** | | **Info** |
| 0 | No signal | |
| 1 | AIN1 | |
| 2 | CAN1 control word | |
| 3 | CAN1 input word 2 | |
| 4 | CAN1 input word 3 | |
| 5 | CAN1 input word 4 | |
| 6 | CAN2 input word 1 | |
| 7 | CAN2 input word 2 | |
| 8 | CAN2 input word 3 | |
| 9 | CAN2 input word 4 | |
| 10 | CAN3 input word 1 | |
| 11 | CAN3 input word 2 | |
| 12 | CAN3 input word 3 | |
| 13 | CAN3 input word 4 | |
| 14 | Digital counter LowWord | |
| 15 | Digital counter HighWord | |
| 16 | MCI word 1 | |
| 17 | MCI word 2 | |
| 18 | MCI word 3 | |
| 19 | MCI word 4 | |
| 20 | MCI word 5 | |
| 21 | MCI word 6 | |
| 22 | MCI word 7 | |
| 23 | MCI word 8 | |
| 24 | MCI word 9 | |
| 25 | MCI word 10 | |
| 26 | MCI word 11 | |
| 27 | MCI word 12 | |
| 28 | MCI word 13 | |
| 29 | MCI word 14 | |
| 30 | MCI word 15 | |
| 31 | MCI word 16 | |
| 32 | Current motor speed | |
| 33 | Current motor torque | |
| 34 | DC-bus voltage | |

| Parameter \| Name: C00164 \| Logbook - analog elements | | Data type: UNSIGNED_16 Index: $24411_d$ = $5F5B_h$ |
|---|---|---|
| 35 | Current motor current | |
| 36 | Current motor voltage | |
| 37 | Current motor frequency | |
| 38 | Effective speed setpoint | |
| 39 | Device utilisation | |
| 40 | Motor utilisation | |
| 41 | L_OffsetGainPar_1: Out | |
| 42 | L_OffsetGainPar_2: Out | |
| 43 | L_OffsetGainPar_3: Out | |
| 44 | L_Aritmethik_1: Out | |
| 45 | L_AnalogSwitch_1: Out | |
| 46 | L_NSet_1: Out | |
| 47 | L_MotorPoti_1: Out | |
| 48 | L_PCTRL_1: Out | |
| 49 | L_SignalMonitor_a: Out1 | |
| 50 | L_SignalMonitor_a: Out2 | |
| 51 | L_SignalMonitor_a: Out3 | |
| 52 | L_SignalMonitor_a: Out4 | |
| 53 | L_MulDiv_1: Out | |
| 54 | L_NSet_1: Target setpoint | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00164/1 | 0: No signal | Logbook - analog element 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00165**

| Parameter \| Name: C00165 \| Error information | Data type: VISIBLE_STRING Index: $24410_d$ = $5F5A_h$ |
|---|---|
| Display of the error number divided into sectors in the event of an error | |
| **Subcodes** | **Info** |
| C00165/1 | Status determining error |
| C00165/2 | Current error |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 14 | |

**C00166**

| Parameter \| Name: **C00166 \| Error information text** | Data type: VISIBLE_STRING Index: $24409_d$ = $5F59_h$ |
|---|---|
| Display of details on the status determining error and on the currently pending error | |
| **Subcodes** | **Info** |
| C00166/1 | Resp. to status det. error<br>• Response to the status determining error |
| C00166/2 | Subj. - status det. error<br>• Subject area of the status determining error |
| C00166/3 | Mess. - status det. error<br>• Textual message of the status determining error |
| C00166/4 | Resp. to curr. error<br>• Response of the currently pending error |
| C00166/5 | Subj. - curr. error<br>• Subject area of the currently pending error |
| C00166/6 | Mess. - curr. error<br>• Textual message of the currently pending error |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 30 | |

**C00167**

| Parameter \| Name: **C00167 \| Logbook data** | Data type: OCTET_STRING Index: $24408_d$ = $5F58_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00168**

| Parameter \| Name: **C00168 \| Status determining error** | | | Data type: UNSIGNED_32 Index: $24407_d$ = $5F57_h$ |
|---|---|---|---|
| Display of the internal error number for the status determining error | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 4294967295 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

### C00169

| Parameter \| Name: **C00169 \| Logbook setting** | | | Data type: UNSIGNED_16 Index: $24406_d$ = $5F56_h$ |
|---|---|---|---|
| Configuration which message types are to be logged in the logbook. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x067E** (decimal: 1662) |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☐ | Reserved | | |
| Bit 1 ☑ | Log entry: Fault | | |
| Bit 2 ☑ | Log entry: Trouble | | |
| Bit 3 ☑ | Log entry: TroubleQuickstop | | |
| Bit 4 ☑ | Log entry: WarningLocked | | |
| Bit 5 ☑ | Log entry: Warning | | |
| Bit 6 ☑ | Log entry: Information | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☑ | Activation: Error counter | | |
| Bit 10 ☑ | Activation: Log line refresh | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

### C00170

| Parameter \| Name: **C00170 \| Current error** | | | Data type: UNSIGNED_32 Index: $24405_d$ = $5F55_h$ |
|---|---|---|---|
| Display of the internal error number of the currently pending error | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 4294967295 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

### C00171

| Parameter \| Name: **C00171 \| Logbook access index** | Data type: UNSIGNED_8 Index: $24404_d$ = $5F54_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

### C00173

| Parameter \| Name: **C00173 \| Mains voltage** | | Data type: UNSIGNED_8 Index: $24402_d$ = $5F52_h$ |
|---|---|---|
| Selection of the mains voltage for operating the device. | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | **3ph 400V / 1ph 230V** | 3-phase 400 V or 1-phase 230 V |
| 1 | 3ph 440V / 1ph 230V | 3-phase 440 V or 1-phase 230 V |
| 2 | 3ph 480V / 1ph 230V | 3-phase 480 V or 1-phase 230 V |
| 3 | 3ph 500V / 1ph 230V | 3-phase 500 V or 1-phase 230 V |
| 4 | 3ph 400V / 1ph 115V | 3-phase 400 V or 1-phase 115 V |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00174**

| Parameter \| Name: **C00174 \| Reduc. brake chopper threshold** | | | Data type: UNSIGNED_8 Index: $24401_d$ = $5F51_h$ |
|---|---|---|---|
| The threshold from which on the brake chopper is controlled is reduced by the voltage value set here. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | V | 150 | **0 V** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00175**

| Parameter \| Name: **C00175 \| Brake energy management** | | Data type: UNSIGNED_8 Index: $24400_d$ = $5F50_h$ |
|---|---|---|
| Selection of the braking procedure | | ▶ Select response if the brake resistor is controlled |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **R_Brems** | The brake resistor is used. When the threshold voltage (C00174) is exceeded, the brake resistor is energised. |
| 1 | RfgStop | The "Ramp function generator stop" signal (*MCTRL_bRfgStop*) is used. When the threshold voltage is exceeded (C00174), the ramp function generator is stopped. |
| 2 | R_Brems + HlgStop | The brake resistor and the "Ramp function generator stop" signal are used. When the threshold voltage is exceeded (C00174), the brake resistor is energised and the ramp function generator is stopped. |
| 3 | FI_MotBrk + RfgStop | Braking is performed by a superimposed speed setpoint vibration in conjunction with "Ramp function generator stop". |
| 4 | R_Brems + FU_MotBrk + HlgStop | Braking is performed by combining all three braking procedures. |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00176**

| Parameter \| Name: **C00176 \| Undervoltage threshold for mains OFF** | Data type: UNSIGNED_16 Index: $24399_d$ = $5F4F_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00177**

| Parameter \| Name: **C00177 \| Switching cycles** | | | Data type: UNSIGNED_32 Index: $24398_d$ = $5F4E_h$ |
|---|---|---|---|
| Counter of different switching cycles and stressful situations | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00177/1 | | | Number of mains switching cycles |
| C00177/2 | | | Number of switching cycles of the output relay |
| C00177/3 | | | Short circuit counter |
| C00177/4 | | | Earth fault counter |
| C00177/5 | | | "Clamp" counter |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00178

| Parameter \| Name: C00178 \| Elapsed-hour meter | | | Data type: UNSIGNED_32 Index: $24397_d$ = $5F4D_h$ |
|---|---|---|---|
| Display of operating hours in seconds | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | s | 2147483647 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00179

| Parameter \| Name: C00179 \| Power-on time meter | | | Data type: UNSIGNED_32 Index: $24396_d$ = $5F4C_h$ |
|---|---|---|---|
| Display of the power-on time in seconds | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | s | 2147483647 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00180

| Parameter \| Name: C00180 \| Running time | | | Data type: UNSIGNED_32 Index: $24395_d$ = $5F4B_h$ |
|---|---|---|---|
| Display of various running times in seconds | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | s | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00180/1 | | | Running time - control card |
| C00180/2 | | | Running time - heatsink fan |
| C00180/3 | | | Running time - internal fan |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00181

| Parameter \| Name: C00181 \| Time settings | | | Data type: UNSIGNED_16 Index: $24394_d$ = $5F4A_h$ |
|---|---|---|---|
| Time for device search function (optical location) | | | ▶ Device search function |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | s | 6000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00181/1 | 5 s | | Time - device search function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00182

| Parameter \| Name: C00182 \| L_NSet_1: S-ramp time PT1 | | | Data type: INTEGER_16 Index: $24393_d$ = $5F49_h$ |
|---|---|---|---|
| FB L_NSet_1: PT1 S-ramp time for the main setpoint ramp function generator<br>• Only effective with activated ramp rounding (C00134 = "1"). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.01 | s | 50.00 | **20.00 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00184**

| Parameter \| Name: **C00184 \| AutoFailReset repetition time** | | | Data type: UNSIGNED_16 Index: $24391_d$ = $5F47_h$ |
|---|---|---|---|
| After the time set here has expired, an error message of an error that has occurred will be reset automatically if "AutoFailReset" had been configured correspondingly in C00188. | | | |
| | | | ▶ AutoFailReset function |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | s | 600 | **3 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00185**

| Parameter \| Name: **C00185 \| AutoFailReset remaining time** | | | Data type: UNSIGNED_16 Index: $24390_d$ = $5F46_h$ |
|---|---|---|---|
| Display of the residual runtime of the "AutoFailReset" function | | | |
| | | | ▶ AutoFailReset function |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | s | 600 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00186**

| Parameter \| Name: **C00186 \| Max. number of AutoFailReset processes** | | | Data type: UNSIGNED_8 Index: $24389_d$ = $5F45_h$ |
|---|---|---|---|
| Maximum number of "AutoFailReset" procedures | | | |
| | | | ▶ AutoFailReset function |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | | 16 | **4** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00187**

| Parameter \| Name: **C00187 \| Current AutoFailReset processes** | | | Data type: UNSIGNED_8 Index: $24388_d$ = $5F44_h$ |
|---|---|---|---|
| Data of the current number of "AutoFailReset" procedures | | | |
| | | | ▶ AutoFailReset function |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 16 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00188**

| Parameter \| Name: **C00188 \| AutoFailReset configuration** | | Data type: UNSIGNED_8 Index: $24387_d$ = $5F43_h$ |
|---|---|---|
| Setting which error messages are to be reset automatically. | | |
| | | ▶ AutoFailReset function |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Off** | No automatic error message reset |
| 1 | Fault + TroubleQSP | Error messages with the response "Fault" and "TroubleQSP" are reset automatically |
| 2 | WarningLocked | Error messages with the response "WarningLocked" are reset automatically |
| 3 | All locking | All "locking" error messages are reset automatically |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00189**

| Parameter \| Name: **C00189 \| Resp. to too frequent AutoFailReset** | Data type: UNSIGNED_8 Index: $24386_d$ = $5F42_h$ |
|---|---|
| Response to exceeding the maximum number of "AutoFailReset" processes set in C00186. | |

▸ AutoFailReset function

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| 0 | No Reaction |
| **1** | **Fault** |
| 2 | Trouble |
| 3 | TroubleQuickStop |
| 4 | WarningLocked |
| 5 | Warning |
| 6 | Information |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00190**

| Parameter \| Name: **C00190 \| L_NSet_1: Setpoint arithmetic** | Data type: UNSIGNED_8 Index: $24385_d$ = $5F41_h$ |
|---|---|

The L_NSet_1 FB: Selection of arithmetics
- To be able to influence the main setpoint (NSet) by an additional setpoint (NAdd).

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **Out = Set** |
| 1 | Out = Set + Add |
| 2 | NOut = NSet - NAdd |
| 3 | NOut = (NSet * NAdd) / 100% |
| 4 | NOut = (NSet * 1%) / \|NAdd\| |
| 5 | Out = (Set*100%)/(100%-Add) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00199**

| Parameter \| Name: **C00199 \| Description data** | Data type: VISIBLE_STRING Index: $24376_d$ = $5F38_h$ |
|---|---|

Parameters for storing decription data for the controller

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00199/1 | | Device name |

☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ COM ☐ MOT Character length: 24

**C00200**

| Parameter \| Name: **C00200 \| Firmware product type** | Data type: VISIBLE_STRING Index: $24375_d$ = $5F37_h$ |
|---|---|

Display of the firmware product type

☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 19

**C00201**

| Parameter \| Name: **C00201 \| Firmware** | Data type: VISIBLE_STRING Index: $24374_d$ = $5F36_h$ |
|---|---|
| Display of the firmware data of the control card and the power section | |
| **Subcodes** | **Info** |
| C00201/1 | Firmware type - ctrl card |
| C00201/2 | Firmware version - ctrl card |
| C00201/3 | Firmware comp. file - ctrl card |
| C00201/4 | Firmware type - power section |
| C00201/5 | Firmware version - power sect. |
| C00201/6 | Firmw. comp. file - power sect. |
| C00201/7 | Firmware type CU2 |
| C00201/8 | Firmware version CU2 |
| C00201/9 | Firmware comp. dat. CU2 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 22 | |

**C00203**

| Parameter \| Name: **C00203 \| Product type code** | Data type: VISIBLE_STRING Index: $24372_d$ = $5F34_h$ |
|---|---|
| Display of the types of the individual device components | |
| **Subcodes** | **Info** |
| C00203/1 | Type: Control card |
| C00203/2 | Type: Power section |
| C00203/3 | Type: MCI module |
| C00203/4 | Reserved |
| C00203/5 | Type: Memory module |
| C00203/6 | Type: Safety card |
| C00203/7 | Type: Standard device |
| C00203/8 | Type: Complete device |
| C00203/9 | Reserved |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 24 | |

**C00204**

| Parameter \| Name: **C00204 \| Serial number** | Data type: VISIBLE_STRING Index: $24371_d$ = $5F33_h$ |
|---|---|
| Display of the serial numbers of the individual device components | |
| **Subcodes** | **Info** |
| C00204/1 | Serial no.: Control card |
| C00204/2 | Serial no.: Power section |
| C00204/3 | Serial no.: MCI module |
| C00204/4 | Reserved |
| C00204/5 | Serial no.: Memory module |
| C00204/6 | Serial no.: Safety card |
| C00204/7 | Serial no.: Standard device |
| C00204/8 | Serial no.: Complete device |
| C00204/9 | Reserved |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 24 | |

C00205

| Parameter \| Name:<br>**C00205 \| Info** | Data type: VISIBLE_STRING<br>Index: $24370_d$ = $5F32_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00206

| Parameter \| Name:<br>**C00206 \| Production date** | Data type: VISIBLE_STRING<br>Index: $24369_d$ = $5F31_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00210

| Parameter \| Name:<br>**C00210 \| HW version** | Data type: VISIBLE_STRING<br>Index: $24365_d$ = $5F2D_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00220

| Parameter \| Name:<br>**C00220 \| L_NSet_1: Acceleration time - add. setpoint** | | | Data type: UNSIGNED_32<br>Index: $24355_d$ = $5F23_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Acceleration time for the additional setpoint *nNAdd_a* | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **0.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

C00221

| Parameter \| Name:<br>**C00221 \| L_NSet_1: Deceleration time - add. setpoint** | | | Data type: UNSIGNED_32<br>Index: $24354_d$ = $5F22_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Deceleration time for the additional setpoint *nNAdd_a* | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **0.000 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

C00222

| Parameter \| Name:<br>**C00222 \| L_PCTRL_1: Vp** | | | Data type: INTEGER_16<br>Index: $24353_d$ = $5F21_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Gain factor Vp for the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.1 | | 500.0 | **1.0** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

C00223

| Parameter \| Name:<br>**C00223 \| L_PCTRL_1: Tn** | | | Data type: UNSIGNED_16<br>Index: $24352_d$ = $5F20_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Reset time Tn for the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 20 | ms | 6000 | **400 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

C00224

| Parameter \| Name:<br>**C00224 \| L_PCTRL_1: Kd** | | | Data type: UNSIGNED_16<br>Index: $24351_d$ = $5F1F_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Derivative-action coefficient Kd for the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | | 5.0 | **0.0** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

## C00225

| Parameter \| Name: C00225 \| L_PCTRL_1: MaxLimit | | | Data type: INTEGER_16 Index: 24350$_d$ = 5F1E$_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Maximum output value of the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **199.99 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00226

| Parameter \| Name: C00226 \| L_PCTRL_1: MinLimit | | | Data type: INTEGER_16 Index: 24349$_d$ = 5F1D$_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Minimum output value of the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **-199.99 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00227

| Parameter \| Name: C00227 \| L_PCTRL_1: Acceleration time | | | Data type: UNSIGNED_32 Index: 24348$_d$ = 5F1C$_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Acceleration time for the output value of the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **0.010 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

## C00228

| Parameter \| Name: C00228 \| L_PCTRL_1: Deceleration time | | | Data type: UNSIGNED_32 Index: 24347$_d$ = 5F1B$_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Deceleration time for the output value of the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | s | 999.999 | **0.010 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

## C00231

| Parameter \| Name: C00231 \| L_PCTRL_1: Operating range | | | Data type: INTEGER_16 Index: 24344$_d$ = 5F18$_h$ |
|---|---|---|---|
| The L_PCTRL_1 FB: Operating range for the PID process controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00231/1 | 199.99 % | | L_PCTRL_1: Pos. maximum |
| C00231/2 | 0.00 % | | L_PCTRL_1: Pos. minimum |
| C00231/3 | 0.00 % | | L_PCTRL_1: Neg. minimum |
| C00231/4 | 199.99 % | | L_PCTRL_1: Neg. maximum |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00233**

| Parameter \| Name: **C00233 \| L_PCTRL_1: Root function** | | Data type: UNSIGNED_8 Index: $24342_d$ = $5F16_h$ |
|---|---|---|
| The L_PCTRL_1 FB: Use of the root function at the actual value input | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Off** | Root function inactive<br>• The actual value *nAct_a* remains unchanged for further processing |
| 1 | On | Root function active<br>• The root is extracted of the actual value *nAct_a* for further processing |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00234**

| Parameter \| Name: **C00234 \| Oscillation damping influence** | | | Data type: UNSIGNED_16 Index: $24341_d$ = $5F15_h$ |
|---|---|---|---|
| | | | ▶ Oscillation damping |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 250.00 | **5.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00235**

| Parameter \| Name: **C00235 \| Oscillation damping filter time** | | | Data type: UNSIGNED_8 Index: $24340_d$ = $5F14_h$ |
|---|---|---|---|
| | | | ▶ Oscillation damping |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 2 | ms | 250 | **32 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00236**

| Parameter \| Name: **C00236 \| Field weakening oscillation damping** | | | Data type: UNSIGNED_8 Index: $24339_d$ = $5F13_h$ |
|---|---|---|---|
| Oscillation damping for idling machines | | | |
| | | | ▶ Oscillation damping |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 40 | **14** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00241**

| Parameter \| Name: **C00241 \| L_NSet_1: Hyst. NSet reached** | | | Data type: INTEGER_16 Index: $24334_d$ = $5F0E_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Hysteresis window for the zero detection of the speed output setpoint<br>• The speed threshold for the zero detection is 1 % | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.50 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00242

| Parameter \| Name: **C00242 \| L_PCTRL_1: Operating mode** | | | Data type: UNSIGNED_8 Index: $24333_d$ = $5F0D_h$ |
|---|---|---|---|

The L_PCTRL_1 FB: Selection of the operating mode
- Depending on the selection, the blue switches in the displayed signal flow are set accordingly in the Engineer on the **Application parameters** tab for the L_PCTRL_1 FB.

| Selection list (Lenze setting printed in bold) | | Info |
|---|---|---|
| 0 | **Off** | The input setpoint *nNSet_a* is output without any changes at the output *nOut_a*. |
| 1 | nNSet + nNSet_PID | *nNSet_a* and *nAct_a* are used as PID input values. The arriving *nNSet_a* is additively linked to the value output by the PID element. |
| 2 | nSet_PID | *nSet_a* and *nAct_a* are used as PID input values. The input *nNSet_a* is not considered. |
| 3 | nNSet_PID | *nNSet_a* and *nAct_a* are used as PID input values. The input *nSet_a* is not considered. |
| 4 | nNSet + nSet_PID | *nSet_a* and *nAct_a* are used as PID input values. The arriving *nNSet_a* setpoint is additively linked to the value output by the PID element. |
| 5 | nNSet \|\| nSet_PID | *nSet_a* and *nAct_a* are used as PID input values. The setpoint *nNSet_a* is output at the output *nOut_a* . The PID output value is output at the output *nPIDOut_a*. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C00243

| Parameter \| Name: **C00243 \| L_PCTRL_1: Acceleration time influence** | | | Data type: UNSIGNED_32 Index: $24332_d$ = $5F0C_h$ |
|---|---|---|---|

The L_PCTRL_1 FB: Acceleration time for showing the PID output value

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.000 | s | 999.999 | **5.000 s** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

## C00244

| Parameter \| Name: **C00244 \| L_PCTRL_1: Deceleration time influence** | | | Data type: UNSIGNED_32 Index: $24331_d$ = $5F0B_h$ |
|---|---|---|---|

The L_PCTRL_1 FB: Deceleration time for masking out the PID output value

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.000 | s | 999.999 | **5.000 s** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

## C00245

| Parameter \| Name: **C00245 \| L_PCTRL_1: PID output value** | | | Data type: INTEGER_16 Index: $24330_d$ = $5F0A_h$ |
|---|---|---|---|

The L_PCTRL_1 FB: Display of the output value of the PID process controller

| Display range (min. value \| unit \| max. value) | | |
|---|---|---|
| -199.99 | % | 199.99 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

## C00246

| Parameter \| Name: **C00246 \| L_PCTRL_1: nAct_a internal** | | | Data type: INTEGER_16 Index: $24329_d$ = $5F09_h$ |
|---|---|---|---|

FB L_PCTRL_1: Display of the internal actual value

| Display range (min. value \| unit \| max. value) | | |
|---|---|---|
| -199.99 | % | 199.99 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

## C00247

| Parameter \| Name: C00247 \| L_PCTRL_1: Window setpoint reached | | | Data type: INTEGER_16 Index: $24328_d$ = $5F08_h$ |
|---|---|---|---|
| FB L_PCTRL_1: Window for comparison operation "actual value = setpoint" | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **2.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00249

| Parameter \| Name: C00249 \| L_PT1_1: Time constant | | | Data type: UNSIGNED_16 Index: $24326_d$ = $5F06_h$ |
|---|---|---|---|
| FB L_PT1_1: Time constant Tn | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | ms | 5000 | **2000 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00250

| Parameter \| Name: C00250 \| L_PT1 2-3: Time constant | | | Data type: INTEGER_16 Index: $24325_d$ = $5F05_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 5000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00250/1 | 2000 ms | | L_PT1_2: Time constant |
| C00250/2 | 2000 ms | | L_PT1_3: Time constant |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00251

| Parameter \| Name: C00251 \| L_DT1_1: Time constant | | | Data type: INTEGER_16 Index: $24324_d$ = $5F04_h$ |
|---|---|---|---|
| FB L_DT1_1: Time constant Tn | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 10 | ms | 5000 | **1000 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00252

| Parameter \| Name: C00252 \| L_DT1_1: Gain | | | Data type: INTEGER_16 Index: $24323_d$ = $5F03_h$ |
|---|---|---|---|
| FB L_DT1_1: Gain factor Vp | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -320.00 | | 320.00 | **1.00** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00253

| Parameter \| Name: C00253 \| L_DT1_1: Sensitivity | | | Data type: UNSIGNED_8 Index: $24322_d$ = $5F02_h$ |
|---|---|---|---|

FB L_DT1_1: Selection of sensitivity
- Depending on the selection, the number of indicated higher-order bits is evaluated.

**Note:**
The most significant bit determines the sign of the value, the remaining bits determine the numerical value.

| Selection list (Lenze setting printed in bold) | | Info |
|---|---|---|
| 1 | **15 bits** | Bit 0 ... bit 14 are evaluated |
| 2 | 14 bits | Bit 0 ... bit 13 are evaluated |
| 3 | 13 bits | Bit 0 ... bit 12 are evaluated |
| 4 | 12 bits | Bit 0 ... bit 11 are evaluated |
| 5 | 11 Bit | Bit 0 ... bit 10 are evaluated |
| 6 | 10 Bit | Bit 0 ... bit 9 are evaluated |
| 7 | 9 Bit | Bit 0 ... bit 8 are evaluated |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C00254

| Parameter \| Name: C00254 \| Kp position controller | Data type: UNSIGNED_16 Index: $24321_d$ = $5F01_h$ |
|---|---|

Gain for following error compensation

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.00 | 1/s | 500.00 | **5.00 1/s** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

## C00265

| Parameter \| Name: C00265 \| SLVC: Tn torque controller | Data type: UNSIGNED_8 Index: $24310_d$ = $5EF6_h$ |
|---|---|

**This code is for device-internal use only and must not be written to by the user!**

## C00270

| Parameter \| Name: C00270 \| SC: Freq. current setpoint filter | Data type: UNSIGNED_16 Index: $24305_d$ = $5EF1_h$ |
|---|---|

Frequency to be inhibited by the current setpoint filter at servo control (SC) and sensorless control for synchronous motors (SLPSM).

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 40.0 | Hz | 1000.0 | **200.0 Hz** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10

## C00271

| Parameter \| Name: C00271 \| SC: Width of current setpoint filter | Data type: UNSIGNED_16 Index: $24304_d$ = $5EF0_h$ |
|---|---|

Frequency width of the current setpoint filter at servo control (SC) and sensorless control for synchronous motors (SLPSM)
- Width around the frequency to be inhibited (C00270).

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.0 | Hz | 500.0 | **0.0 Hz** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10

**C00272**

| Parameter \| Name: **C00272 \| SC: Depth of current setpoint filter** | | | Data type: UNSIGNED_16 Index: $24303_d$ = $5EEF_h$ |
|---|---|---|---|
| Damping of the current setpoint filter at servo control (SC) and sensorless control for synchronous motors (SLPSM) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | db | 100 | **0 db** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00273**

| Parameter \| Name: **C00273 \| Moment of inertia** | | | Data type: UNSIGNED_32 Index: $24302_d$ = $5EEE_h$ |
|---|---|---|---|
| Moment of inertia for setpoint feedforward control at servo control (SC) and sensorless vector control (SLVC) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | kg cm^2 | 600000.00 | **0.00 kg cm^2** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

**C00274**

| Parameter \| Name: **C00274 \| SC: Max. change in acceleration** | | | Data type: UNSIGNED_16 Index: $24301_d$ = $5EED_h$ |
|---|---|---|---|
| Limitation of the acceleration change at servo control (SC) and sensorless control for synchronous motors (SLPSM)<br>• Setting in % of M_Nenn per ms. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | %/ms | 400.0 | **400.0 %/ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00275**

| Parameter \| Name: **C00275 \| Filter setpoint feedforward control** | | | Data type: UNSIGNED_16 Index: $24300_d$ = $5EEC_h$ |
|---|---|---|---|
| Filter time of setpoint feedforward control at servo control (SC) and sensorless vector control (SLVC)<br>• The setpoint feedforward control requires the entry of the moment of inertia in C00273. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 1000.0 | **1.0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00276**

| Parameter \| Name: **C00276 \| SC: Max. output voltage** | | | Data type: UNSIGNED_8 Index: $24299_d$ = $5EEB_h$ |
|---|---|---|---|
| Maximum output voltage at servo control (SC)<br>• Regarding the current DC-bus voltage. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 80 | % | 99 | **95 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00280**

| Parameter \| Name: **C00280 \| SC: Filter time const. DC detection** | | | Data type: UNSIGNED_16 Index: $24295_d$ = $5EE7_h$ |
|---|---|---|---|
| Filter time constant for DC-bus voltage filtering<br>• The filter time constant is e.g. used for field weakening control at servo control (SC). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | ms | 1000 | **25 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00290**

| Parameter \| Name: **C00290 \| RCOM error counter** | Data type: UNSIGNED_16 Index: $24285_d$ = $5EDD_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00291**

| Parameter \| Name: **C00291 \| Error type RCOM** | Data type: UNSIGNED_8 Index: $24284_d$ = $5EDC_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00296**

| Parameter \| Name: **C00296 \| ICOM error number** | Data type: UNSIGNED_16 Index: $24279_d$ = $5ED7_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00297**

| Parameter \| Name: **C00297 \| Counter Receive Error Isr** | Data type: UNSIGNED_8 Index: $24278_d$ = $5ED6_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00301**

| Parameter \| Name: **C00301 \| DebugAccess** | Data type: UNSIGNED_16 Index: $24274_d$ = $5ED2_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00302**

| Parameter \| Name: **C00302 \| Internal Commands** | Data type: UNSIGNED_8 Index: $24273_d$ = $5ED1_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00304**

| Parameter \| Name: **C00304 \| Password1** | Data type: UNSIGNED_32 Index: $24271_d$ = $5ECF_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00305**

| Parameter \| Name: **C00305 \| Password2** | Data type: UNSIGNED_32 Index: $24270_d$ = $5ECE_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00306**

| Parameter \| Name: **C00306 \| Debug address** | Data type: UNSIGNED_32 Index: $24269_d$ = $5ECD_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00307**

| Parameter \| Name: **C00307 \| Debug value** | Data type: UNSIGNED_16 Index: $24268_d$ = $5ECC_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00308**

| Parameter \| Name: **C00308 \| PartitionOffset** | Data type: UNSIGNED_16 Index: $24267_d$ = $5ECB_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00309**

| Parameter \| Name: **C00309 \| PartitionSel** | Data type: UNSIGNED_8 Index: $24266_d$ = $5ECA_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00310

| Parameter \| Name: C00310 \| PartitionValue | Data type: UNSIGNED_16 Index: $24265_d$ = $5EC9_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00311

| Parameter \| Name: C00311 \| Runtime measurement | Data type: UNSIGNED_32 Index: $24264_d$ = $5EC8_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00312

| Parameter \| Name: C00312 \| System runtimes | | | Data type: UNSIGNED_32 Index: $24263_d$ = $5EC7_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | µs | 1310.700 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00312/1 | 0.000 µs | | System runtime reserve |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

## C00313

| Parameter \| Name: C00313 \| LS_DataAccess: Activation | Data type: UNSIGNED_8 Index: $24262_d$ = $5EC6_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00314

| Parameter \| Name: C00314 \| LS_DataAccess: Address access | Data type: UNSIGNED_32 Index: $24261_d$ = $5EC5_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00315

| Parameter \| Name: C00315 \| SystemFail-Adr | Data type: UNSIGNED_32 Index: $24260_d$ = $5EC4_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00316

| Parameter \| Name: C00316 \| SystemFail-Info | Data type: UNSIGNED_16 Index: $24259_d$ = $5EC3_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00317

| Parameter \| Name: C00317 \| WatchdogTimeMax | Data type: UNSIGNED_16 Index: $24258_d$ = $5EC2_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00320

| Parameter \| Name: C00320 \| Debug information | Data type: UNSIGNED_32 Index: $24255_d$ = $5EBF_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00321**

| Parameter \| Name: **C00321 \| Main program runtime** | | | Data type: UNSIGNED_16 Index: $24254_d$ = $5EBE_h$ |
|---|---|---|---|
| Display of the current and the maximum runtime of the main program in the controller | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00321/1 | 0 ms | | Curr. runtime of main program |
| C00321/2 | 0 ms | | Max. runtime of main program |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00322**

| Parameter \| Name: **C00322 \| Transmission mode CAN TxPDOs** | | | Data type: UNSIGNED_8 Index: $24253_d$ = $5EBD_h$ |
|---|---|---|---|

TPDO transmission type according to DS301 V4.02
- The following transmission modes are supported:
  - 0: Synchronous and acyclic
  - 1 ... 240: Synchronous and cyclic
  - 252: Synchronous - RTR only
  - 253: Asynchronous - RTR only
  - 254: Asynchronous - manufacturer-specific
  - 255: Asynchronous - device-profile specific
- The basic setting for all PDOs is "Asynchronous - manufacturer-specific" (254).
- Illustration of the CANopen objects I-1800/2 ... I-1802/2 (see DS301 V4.02).

▶ System bus "CAN on board"

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | | 255 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00322/1 | 254 | | Transmission mode CAN1 OUT |
| C00322/2 | 254 | | Transmission mode CAN2 OUT |
| C00322/3 | 254 | | Transmission mode CAN3 OUT |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00323**

| Parameter \| Name: **C00323 \| Transmission mode CAN Rx PDOs** | | | Data type: UNSIGNED_8 Index: $24252_d$ = $5EBC_h$ |
|---|---|---|---|

RPDO transmission type according to DS301 V4.02
- In the case of the RPDO serves as monitoring setting in the case of sync-controlled PDOs.
- The following transmission modes are supported:
  - 0: Synchronous and acyclic
  - 1 ... 240: Synchronous and cyclic
  - 252: Synchronous - RTR only
  - 253: Asynchronous - RTR only
  - 254: Asynchronous - manufacturer-specific
  - 255: Asynchronous - device-profile specific
- The basic setting for all PDOs is "Asynchronous - manufacturer-specific" (254).
- Illustration of the CANopen objects I-1400/2 ... I-1402/2 (see DS301 V4.02).

▶ System bus "CAN on board"

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | | 255 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00323/1 | 254 | | Transmission mode CAN1 IN |
| C00323/2 | 254 | | Transmission mode CAN2 IN |
| C00323/3 | 254 | | Transmission mode CAN3 IN |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00324**

| Parameter \| Name: **C00324 \| CAN transmission blocking time** | Data type: UNSIGNED_16 Index: $24251_d$ = $5EBB_h$ |
|---|---|

Blocking time for the transmission of the emergency telegram and the process data

**Note:**
If the "Asynchronous - manufacturer-specific/device profile-specific" transmisson type is set, the transmission cycle timer is reset to 0 if event-controlled transmission has been triggered.
Example: Cycle time (C00356/x) = 500 ms, blocking time = 100 ms, data change sporadically:
- In the case of a sporadic data change < 500 ms, due to the blocking time set, transmission takes place every 100 ms (event-controlled transmission) as quickly as possible.
- In the case of a sporadic data change > 500 ms, due to the cycle time set, transmission takes place every 500 ms (cyclic transmission).

▶ System bus "CAN on board"

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | ms | 6500 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00324/1 | 0 ms | | CAN emergency blocking time |
| C00324/2 | 0 ms | | CAN1_OUT blocking time |
| C00324/3 | 0 ms | | CAN2_OUT blocking time |
| C00324/4 | 0 ms | | CAN3_OUT blocking time |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C00338**

| Parameter \| Name: **C00338 \| L_Arithmetik_1: Function** | Data type: UNSIGNED_8 Index: $24237_d$ = $5EAD_h$ |
|---|---|

The L_Arithmetik_1 FB: Selection of internal arithmetics

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **Out = In1** |
| 1 | Out = In1 + In2 |
| 2 | nOut_a = nIn1_a - nIn2_a |
| 3 | Out = (In1 * In2) / 100% |
| 4 | nOut_a = (nIn1_a * 1%) / \|nIn2_a\| |
| 5 | nOut_a = (nIn1_a * 100%) / (100% - nIn2_a) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00339**

| Parameter \| Name: **C00339 \| L_Arithmetik_2: Function** | Data type: UNSIGNED_8 Index: $24236_d$ = $5EAC_h$ |
|---|---|

The L_Arithmetik_2 FB: Selection of internal arithmetics

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **nOut_a = nIn1_a** |
| 1 | nOut_a = nIn1_a + nIn2_a |
| 2 | nOut_a = nIn1_a - nIn2_a |
| 3 | nOut_a = (nIn1_a * nIn2_a) / 100% |
| 4 | nOut_a = (nIn1_a * 1%) / \|nIn2_a\| |
| 5 | nOut_a = (nIn1_a * 100%) / (100% - nIn2_a) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00341**

| Parameter \| Name:<br>**C00341 \| CAN management - error configuration** | | | Data type: UNSIGNED_16<br>Index: $24234_d$ = $5EAA_h$ |
|---|---|---|---|
| Selection of the events for which the *bFail* error output of the LS_CANMAnagement SB must be set to TRUE. | | | |
| **Setting range** **(min. hex value \| max. hex value)** | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☐ | BusOff_MsgErr | | |
| Bit 1 ☐ | Warning | | |
| Bit 2 ☐ | NodeStopped | | |
| Bit 3 ☐ | HeartBeatEvent | | |
| Bit 4 ☐ | CAN1_In_Überw. | | |
| Bit 5 ☐ | CAN2_In_Überw. | | |
| Bit 6 ☐ | CAN3_In_Überw. | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

## C00342

| Parameter \| Name: **C00342 \| CAN decoupling PDOInOut** | | | Data type: UNSIGNED_16 Index: $24233_d$ = $5EA9_h$ |
|---|---|---|---|
| Configuration defining the events that lead to a decoupling of the process data words. | | | |
| | | | ▸ Configuring exception handling of the CAN PDOs |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☐ | BusOff_MsgErr | | |
| Bit 1 ☐ | Warning | | |
| Bit 2 ☐ | NodeStopped | | |
| Bit 3 ☐ | HeartBeatEvent | | |
| Bit 4 ☐ | CAN1_In_Überw. | | |
| Bit 5 ☐ | CAN2_In_Überw. | | |
| Bit 6 ☐ | CAN3_In_Überw. | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Trouble | | |
| Bit 15 ☐ | Fault | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00342/1 | 0 | | CAN decoupling PDO_In from the bus |
| C00342/2 | 0 | | CAN decoupling PDO_Out from the appl. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

## C00343

| Parameter \| Name: **C00343 \| LP_CanIn decoupling value** | | Data type: UNSIGNED_16 Index: $24232_d$ = $5EA8_h$ |
|---|---|---|
| Definition of the value the process data words are to have in the decoupled state. | | |
| | | ▶ Configuring exception handling of the CAN PDOs |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | | 65535 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00343/1 | 0 | LP_CanIn1:wCtrl DiscVal |
| C00343/2 | 0 | LP_CanIn1:wIn2 DiscVal |
| C00343/3 | 0 | LP_CanIn1:wIn3 DiscVal |
| C00343/4 | 0 | LP_CanIn1:wIn4 DiscVal |
| C00343/5 | 0 | LP_CanIn2:wIn1 DiscVal |
| C00343/6 | 0 | LP_CanIn2:wIn2 DiscVal |
| C00343/7 | 0 | LP_CanIn2:wIn3 DiscVal |
| C00343/8 | 0 | LP_CanIn2:wIn4 DiscVal |
| C00343/9 | 0 | LP_CanIn3:wIn1 DiscVal |
| C00343/10 | 0 | LP_CanIn3:wIn2 DiscVal |
| C00343/11 | 0 | LP_CanIn3:wIn3 DiscVal |
| C00343/12 | 0 | LP_CanIn3:wIn4 DiscVal |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00344

| Parameter \| Name: **C00344 \| LP_CanOut decoupling value** | | Data type: UNSIGNED_16 Index: $24231_d$ = $5EA7_h$ |
|---|---|---|
| Definition of the value the process data words are to have in the decoupled state. | | |
| | | ▶ Configuring exception handling of the CAN PDOs |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | | 65535 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00344/1 | 0 | LP_CanOut1:wState DiscVal |
| C00344/2 | 0 | LP_CanOut1:wOut2 DiscVal |
| C00344/3 | 0 | LP_CanOut1:wOut3 DiscVal |
| C00344/4 | 0 | LP_CanOut1:wOut4 DiscVal |
| C00344/5 | 0 | LP_CanOut2:wOut1 DiscVal |
| C00344/6 | 0 | LP_CanOut2:wOut2 DiscVal |
| C00344/7 | 0 | LP_CanOut2:wOut3 DiscVal |
| C00344/8 | 0 | LP_CanOut2:wOut4 DiscVal |
| C00344/9 | 0 | LP_CanOut3:wOut1 DiscVal |
| C00344/10 | 0 | LP_CanOut3:wOut2 DiscVal |
| C00344/11 | 0 | LP_CanOut3:wOut3 DiscVal |
| C00344/12 | 0 | LP_CanOut3:wOut4 DiscVal |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00345**

| Parameter \| Name: **C00345 \| CAN error status** | Data type: UNSIGNED_8 Index: $24230_d$ = $5EA6_h$ |
|---|---|

System bus "CAN on board"

| **Selection list** (read only) | |
|---|---|
| 0 | No Error |
| 1 | Warning ErrActive |
| 2 | Warning ErrPassive |
| 3 | Bus off |
| 4 | Reserved |
| 5 | Reserved |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00347**

| Parameter \| Name: **C00347 \| CAN status HeartBeat producer** | Data type: UNSIGNED_8 Index: $24228_d$ = $5EA4_h$ |
|---|---|

Heartbeat protocol

| **Selection list** | |
|---|---|
| 0 | Boot-up |
| 4 | Stopped |
| 5 | Operational |
| 127 | Pre-Operat. |
| 250 | Failed |
| 255 | NoResponse |

| **Subcodes** | **Info** |
|---|---|
| C00347/1 | Status node 1 ... 15 |
| C00347/... | |
| C00347/15 | |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

### C00349

| Parameter \| Name: **C00349 \| CAN setting - DIP switch** | | | Data type: UNSIGNED_16 Index: 24226$_d$ = 5EA2$_h$ |
|---|---|---|---|
| DIP switch setting during last mains power-on | | | |
| | | | ▸ System bus "CAN on board" |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Node address 1 | | |
| Bit 1 | Node address 2 | | |
| Bit 2 | Node address 4 | | |
| Bit 3 | Node address 8 | | |
| Bit 4 | Node address 16 | | |
| Bit 5 | Node address 32 | | |
| Bit 6 | Node address 64 | | |
| Bit 7 | Baud rate 1 | | |
| Bit 8 | Baud rate 2 | | |
| Bit 9 | Baud rate 4 | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | DIP switch at 24V-ON accepted | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

### C00350

| Parameter \| Name: **C00350 \| CAN node address** | | | Data type: UNSIGNED_8 Index: 24225$_d$ = 5EA1$_h$ |
|---|---|---|---|
| Setting of the node address via parameters<br>• The node address can only be parameterised if the node address "0" is set via the DIP switches.<br>• A change in the node address will not be effective until a CAN Reset Node is performed. | | | |
| | | | ▸ System bus "CAN on board" |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | | 127 | **1** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00351**

| Parameter \| Name: **C00351 \| CAN baud rate** | Data type: UNSIGNED_8 Index: $24224_d$ = $5EA0_h$ |
|---|---|

Setting of the baud rate via parameters
- The baud rate can only be parameterised if the baud rate "0" is set via the DIP switches.
- A change in the baud rate will not be effective until a CAN Reset Node is performed.

System bus "CAN on board"

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **500 kbps** |
| 1 | 250 kbps |
| 2 | 125 kbps |
| 3 | 50 kbps |
| 4 | 1000 kbps |
| 5 | 20 kbps |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C00352**

| Parameter \| Name: **C00352 \| CAN slave/master** | Data type: UNSIGNED_8 Index: $24223_d$ = $5E9F_h$ |
|---|---|

The drive starts as CAN master after mains switching if a value of "1" has been entered and saved here.

System bus "CAN on board"

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **Slave** |
| 1 | Master |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C00353**

| Parameter \| Name: **C00353 \| CAN IN/OUT COBID source** | Data type: UNSIGNED_8 Index: $24222_d$ = $5E9E_h$ |
|---|---|

Identifier assignment procedure for the CANx In/Out process data

System bus "CAN on board"

| **Selection list** | | **Info** |
|---|---|---|
| 0 | COBID = C0350 + LenzeBaseID | COBID = device address + LenzeBaseID |
| 1 | COBID = C0350 + CANBaseID | COBID = device address + CANBaseID (C00354/x) |
| 2 | COBID = C0354/x | COBID = direct setting from C00354/x |
| **Subcodes** | **Lenze setting** | **Info** |
| C00353/1 | 1: COBID = C0350 + CANBaseID | COBID source CAN1_IN/OUT |
| C00353/2 | 1: COBID = C0350 + CANBaseID | COBID source CAN2_IN/OUT |
| C00353/3 | 1: COBID = C0350 + CANBaseID | COBID source CAN3_IN/OUT |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

## C00354

| Parameter \| Name: **C00354 \| COBID** | | | Data type: UNSIGNED_32 Index: $24221_d$ = $5E9D_h$ |
|---|---|---|---|
| Setting of the default COBID according to CANopen<br>• A change in the COBID will not be effective until a CAN reset node is performed.<br>▸ System bus "CAN on board" | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x00000000 | | 0xFFFFFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | COBID Bit0 | | • Bit 0 ... 10: COB-ID<br>• Bit 11 ... 30: Reserved<br>• Bit 31: PDO invalid (is not transmitted) |
| ... | ... | | |
| Bit 31 | PDO invalid | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00354/1 | 513 | | COBID CAN1_IN |
| C00354/2 | 385 | | COBID CAN1_OUT |
| C00354/3 | 769 | | COBID CAN2_IN |
| C00354/4 | 641 | | COBID CAN2_OUT |
| C00354/5 | 1025 | | COBID CAN3_IN |
| C00354/6 | 897 | | COBID CAN3_OUT |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT | | | |

## C00355

| Parameter \| Name: **C00355 \| Active COBID** | | | Data type: UNSIGNED_16 Index: $24220_d$ = $5E9C_h$ |
|---|---|---|---|
| Display of the COBID of the PDOs that is active in the CAN stack<br>▸ System bus "CAN on board" | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 2047 | |
| **Subcodes** | | | **Info** |
| C00355/1 | | | Active COBID CAN1_IN |
| C00355/2 | | | Active COBID CAN1_OUT |
| C00355/3 | | | Active COBID CAN2_IN |
| C00355/4 | | | Active COBID CAN2_OUT |
| C00355/5 | | | Active COBID CAN3_IN |
| C00355/6 | | | Active COBID CAN3_OUT |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

## C00356

| Parameter \| Name: **C00356 \| CAN time settings** | | Data type: UNSIGNED_16 Index: $24219_d$ = $5E9B_h$ |
|---|---|---|
| Different time settings for the CAN interface | | |
| | | System bus "CAN on board" |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | ms | 65000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00356/1 | 3000 ms | CAN delay during status change from "Boot-up" to "Operational" |
| C00356/2 | 0 ms | CAN2_OUT cycle time |
| C00356/3 | 0 ms | CAN3_OUT cycle time |
| C00356/4 | 0 ms | CANx_OUT time "Operational" until "first transmission" |
| C00356/5 | 0 ms | CAN1_OUT cycle time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

## C00357

| Parameter \| Name: **C00357 \| CAN monitoring times** | | Data type: UNSIGNED_16 Index: $24218_d$ = $5E9A_h$ |
|---|---|---|
| Mapping of the RPDO event time (see DS301 V4.02)<br>• If a value unequal to "0" is entered, the RPDO is not expected before the set time has expired.<br>• If the RPDO is not received within the expected time, the response set in C00593/1...3 will be triggered. | | |
| | | System bus "CAN on board" |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | ms | 65000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00357/1 | 3000 ms | CAN1_IN monitoring time |
| C00357/2 | 3000 ms | CAN2_IN monitoring time |
| C00357/3 | 3000 ms | CAN3_IN monitoring time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

## C00358

| Parameter \| Name: **C00358 \| CANx_OUT data length** | | Data type: UNSIGNED_8 Index: $24217_d$ = $5E99_h$ |
|---|---|---|
| Setting of the data length for TX PDOs | | |
| | | System bus "CAN on board" |
| **Setting range** (min. value \| unit \| max. value) | | |
| 1 | | 8 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00358/1 | 8 | CAN1_OUT data length |
| C00358/2 | 8 | CAN2_OUT data length |
| C00358/3 | 8 | CAN3_OUT data length |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C00359**

| Parameter \| Name: **C00359 \| CAN status** | Data type: UNSIGNED_8 Index: $24216_d$ = $5E98_h$ |
|---|---|

System bus "CAN on board"

| **Selection list** (read only) | |
|---|---|
| 0 | Operational |
| 1 | Pre-Operat. |
| 2 | Reserved |
| 3 | Reserved |
| 4 | BootUp |
| 5 | Stopped |
| 6 | Reserved |
| 7 | Reset |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C00360**

| Parameter \| Name: **C00360 \| CAN telegram counter** | Data type: UNSIGNED_16 Index: $24215_d$ = $5E97_h$ |
|---|---|

System bus "CAN on board"

| **Display range** (min. value \| unit \| max. value) | | |
|---|---|---|
| 0 | | 65535 |

| **Subcodes** | **Info** |
|---|---|
| C00360/1 | All PDO/SDO sent |
| C00360/2 | All PDO/SDO received |
| C00360/3 | Telegram counter CAN1_OUT |
| C00360/4 | Telegram counter CAN2_OUT |
| C00360/5 | Telegram counter CAN3_OUT |
| C00360/6 | Telegram counter SDO1 OUT |
| C00360/7 | Telegram counter SDO2 OUT |
| C00360/8 | Telegram counter CAN1_IN |
| C00360/9 | Telegram counter CAN2_IN |
| C00360/10 | Telegram counter CAN3_IN |
| C00360/11 | Telegram counter SDO1 IN |
| C00360/12 | Telegram counter SDO2 IN |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C00364**

| Parameter \| Name: **C00364 \| CAN MessageError** | Data type: UNSIGNED_8 Index: $24211_d$ = $5E93_h$ |
|---|---|

System bus "CAN on board"

| **Selection list** (read only) | |
|---|---|
| 0 | No Error |
| 1 | StuffError |
| 2 | FormError |
| 3 | AckError |
| 4 | Bit1Error |
| 5 | Bit0Error |
| 6 | CRCError |
| 7 | Reserved |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C00366**

| Parameter \| Name: **C00366 \| Number of CAN SDO channels** | Data type: UNSIGNED_8 Index: $24209_d$ = $5E91_h$ |
|---|---|

Selection of the number of active parameter data channels
- In the Lenze setting, only the parameter data channel 1 is activated according to CANopen.
- In order to activate both parameter data channels, set the selection "2 SDO Lenze".

System bus "CAN on board"

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **1 SDO CANOpen** |
| 1 | 2 SDO Lenze |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C00367**

| Parameter \| Name: **C00367 \| CAN Sync-Rx-Identifier** | | | Data type: UNSIGNED_16 Index: $24208_d$ = $5E90_h$ |
|---|---|---|---|
| Identifier by means of which the sync slave is to receive sync telegrams.<br>• Mapping of the CANopen object I-1005 (see DS301 V4.02).<br>System bus "CAN on board" | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0080** (decimal: 128) |
| **Value is bit-coded:** (☑ = bit set) | | | |
| Bit 0 ☐ | COBID Bit0 | | |
| Bit 1 ☐ | COBID Bit1 | | |
| Bit 2 ☐ | COBID Bit2 | | |
| Bit 3 ☐ | COBID Bit3 | | |
| Bit 4 ☐ | COBID Bit4 | | |
| Bit 5 ☐ | COBID Bit5 | | |
| Bit 6 ☐ | COBID Bit6 | | |
| Bit 7 ☑ | COBID Bit7 | | |
| Bit 8 ☐ | COBID Bit8 | | |
| Bit 9 ☐ | COBID Bit9 | | |
| Bit 10 ☐ | COBID Bit10 | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT | | | |

## C00368

| Parameter \| Name: **C00368 \| CAN Sync-Tx-Identifier** | Data type: UNSIGNED_16 Index: $24207_d$ = $5E8F_h$ |
|---|---|

Identifier by means of which the sync master is to transmit sync telegrams.
- Mapping of the CANopen object I-1005 (see DS301 V4.02).

▸ System bus "CAN on board"

| Setting range (min. hex value \| max. hex value) | | | Lenze setting |
|---|---|---|---|
| 0x0000 | | 0xFFFF | **0x0080** (decimal: 128) |

| Value is bit-coded: (☑ = bit set) | |
|---|---|
| Bit 0 ☐ | COBID Bit0 |
| Bit 1 ☐ | COBID Bit1 |
| Bit 2 ☐ | COBID Bit2 |
| Bit 3 ☐ | COBID Bit3 |
| Bit 4 ☐ | COBID Bit4 |
| Bit 5 ☐ | COBID Bit5 |
| Bit 6 ☐ | COBID Bit6 |
| Bit 7 ☑ | COBID Bit7 |
| Bit 8 ☐ | COBID Bit8 |
| Bit 9 ☐ | COBID Bit9 |
| Bit 10 ☐ | COBID Bit10 |
| Bit 11 ☐ | Reserved |
| Bit 12 ☐ | Reserved |
| Bit 13 ☐ | Reserved |
| Bit 14 ☐ | Reserved |
| Bit 15 ☐ | Sync-transmit off |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT

## C00369

| Parameter \| Name: **C00369 \| CAN sync transmission cycle time** | Data type: UNSIGNED_16 Index: $24206_d$ = $5E8E_h$ |
|---|---|

Cycle during which the sync master is to transmit sync telegrams.
- If "0 ms" is set (Lenze setting), no sync telegrams are generated.
- Mapping of the CANopen object I-1006 (see DS301 V4.02).

▸ System bus "CAN on board"

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0 | ms | 65000 | **0 ms** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

## C00370

| Parameter \| Name: **C00370 \| SyncTxRxTimes** | Data type: INTEGER_16 Index: $24205_d$ = $5E8D_h$ |
|---|---|

▸ System bus "CAN on board"

| Display range (min. value \| unit \| max. value) | | |
|---|---|---|
| -1310 | µs | 1310 |

| Subcodes | Info |
|---|---|
| C00370/1 | CAN Sync instant of transmission |
| C00370/2 | Sync instant of reception |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00372**

| Parameter \| Name: **C00372 \| CAN_Tx_Rx_Error** | | | Data type: UNSIGNED_8 Index: $24203_d$ = $5E8B_h$ |
|---|---|---|---|
| | | | System bus "CAN on board" |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |
| **Subcodes** | | | **Info** |
| C00372/1 | | | CAN Tx_Error |
| C00372/2 | | | CAN Rx_Error |
| C00372/3 | | | CAN Tx_Overflow |
| C00372/4 | | | CAN Rx_Overflow |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00381**

| Parameter \| Name: **C00381 \| CAN Heartbeat producer time** | | | Data type: UNSIGNED_16 Index: $24194_d$ = $5E82_h$ |
|---|---|---|---|
| Time interval for the transmission of the heartbeat telegram to the consumer(s).<br>• The heartbeat telegram is sent automatically as soon as a time > 0 ms is set.<br>• Mapping of the CANopen object I-1017 (see DS301 V4.02). | | | |
| | | | Heartbeat protocol |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | ms | 65535 | **0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C00385**

| Parameter \| Name: **C00385 \| CAN node addr. HeartBeat producer** | | | Data type: UNSIGNED_8 Index: $24190_d$ = $5E7E_h$ |
|---|---|---|---|
| The subcodes represent the nodes to be monitored by heartbeat. | | | |
| | | | Heartbeat protocol |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 127 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00385/1 | 0 | | CAN node address HeartBeat producer 1 ... 15 |
| C00385/... | | | |
| C00385/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00386**

| Parameter \| Name: **C00386 \| CAN HeartBeat ConsumerTime** | | | Data type: UNSIGNED_16 Index: $24189_d$ = $5E7D_h$ |
|---|---|---|---|
| The subcodes represent the nodes to be monitored by heartbeat. | | | |
| | | | Heartbeat protocol |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00386/1 | 0 ms | | ConsumerTime HeartBeat producer 1 ... 15 |
| C00386/... | | | |
| C00386/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00400**

| Parameter \| Name: **C00400 \| LS_PulseGenerator** | | Data type: UNSIGNED_16 Index: $24175_d$ = $5E6F_h$ |
|---|---|---|
| Time setting of the pulse to be output by the SB LS_PulseGenerator | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | ms | 60000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00400/1 | 1000 ms | Length of LOW level (break) |
| C00400/2 | 1000 ms | Length of HIGH level |
| C00400/3 | 100 ms | Delay time for status signal *bFirstCycleDone*<br>• The *bFirstCycleDone* status signal is set to TRUE when the first task cycle is complete and the time set here has expired. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00401**

| Parameter \| Name: **C00401 \| CANxInOut: Inversion** | | | Data type: UNSIGNED_16 Index: $24174_d$ = $5E6E_h$ |
|---|---|---|---|
| This parameter serves to invert the control/status bits of the CAN port blocks. | | | |
| | | | ▸ CAN port block |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Active | | Bit set = bit is inverted |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00401/1 | 0 | | Inversion of LP_CanIn1.bCtrl1_B0...15 |
| C00401/2 | 0 | | Inversion of LP_CanOut1.bState1_B0...15 |
| C00401/3 | 0 | | Inversion of LP_CanIn2.bIn1_B0...15 |
| C00401/4 | 0 | | Inversion of LP_CanOut2.bOut1_B0...15 |
| C00401/5 | 0 | | Inversion of LP_CanIn3.bIn1_B0...15 |
| C00401/6 | 0 | | Inversion of LP_CanOut3.bOut1_B0...15 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00408**

| Parameter \| Name: **C00408 \| LP_CanIn mapping selection** | | Data type: UNSIGNED_8 Index: $24167_d$ = $5E67_h$ |
|---|---|---|
| Selection of the mapping source for port blocks LP_CanIn1...3 | | |
| | | ▸ CAN port block |
| **Selection list** | | **Info** |
| 0 | CanIn | CanIn |
| 1 | Par.C409 | Mapping configured in C00409 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00408/1 | 0: CanIn | Mapping selection LP_CanIn1 |
| C00408/2 | 0: CanIn | Mapping selection LP_CanIn2 |
| C00408/3 | 0: CanIn | Mapping selection LP_CanIn3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00409**

| Parameter \| Name: **C00409 \| LP_CanIn Mapping** | | | Data type: UNSIGNED_16 Index: $24166_d$ = $5E66_h$ |
|---|---|---|---|
| Mapping for port blocks LP_CanIn1...3 | | | |
| | | | CAN port block |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00409/1 | 0 | | LP_CanIn1:wCtrl MapVal |
| C00409/2 | 0 | | LP_CanIn1:wIn2 MapVal |
| C00409/3 | 0 | | LP_CanIn1:wIn3 MapVal |
| C00409/4 | 0 | | LP_CanIn1:wIn4 MapVal |
| C00409/5 | 0 | | LP_CanIn2:wIn1 MapVal |
| C00409/6 | 0 | | LP_CanIn2:wIn2 MapVal |
| C00409/7 | 0 | | LP_CanIn2:wIn3 MapVal |
| C00409/8 | 0 | | LP_CanIn2:wIn4 MapVal |
| C00409/9 | 0 | | LP_CanIn3:wIn1MapVal |
| C00409/10 | 0 | | LP_CanIn3:wIn2 MapVal |
| C00409/11 | 0 | | LP_CanIn3:wIn3 MapVal |
| C00409/12 | 0 | | LP_CanIn3:wIn4 MapVal |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00410**

| Parameter \| Name: **C00410 \| L_SignalMonitor_a: Signal sources** | | Data type: UNSIGNED_16 Index: $24165_d$ = $5E65_h$ |
|---|---|---|
| The L_SignalMonitor_a FB: Selection of the signal sources | | |
| **Selection list** | | |
| See selection list - analog signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00410/1 | 0: Not connected | Signal source for output *nOut1_a* |
| C00410/2 | 0: Not connected | Signal source for output *nOut2_a* |
| C00410/3 | 0: Not connected | Signal source for output *nOut3_a* |
| C00410/4 | 0: Not connected | Signal source for output *nOut4_a* |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00411**

| Parameter \| Name: **C00411 \| L_SignalMonitor_b: Signal sources** | | Data type: UNSIGNED_16 Index: $24164_d$ = $5E64_h$ |
|---|---|---|
| The L_SignalMonitor_b FB: Selection of the signal sources | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00411/1 | 0: Not connected | Signal source for output *bOut1* |
| C00411/2 | 0: Not connected | Signal source for output *bOut2* |
| C00411/3 | 0: Not connected | Signal source for output *bOut3* |
| C00411/4 | 0: Not connected | Signal source for output *bOut4* |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00412**

| Parameter \| Name: **C00412 \| L_SignalMonitor_b: Inversion** | | | Data type: UNSIGNED_8 Index: $24163_d$ = $5E63_h$ |
|---|---|---|---|
| The L_SignalMonitor_b FB: Inversion of the binary outputs | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x00** (decimal: 0) |
| **Value is bit-coded:** (☑ = bit set) | | | **Info** |
| Bit 0 ☐ | bOut1 inverted | | Bit set = inversion active |
| Bit 1 ☐ | bOut2 inverted | | |
| Bit 2 ☐ | bOut3 inverted | | |
| Bit 3 ☐ | bOut4 inverted | | |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00413**

| Parameter \| Name: **C00413 \| L_SignalMonitor_a: Offs./gain** | | | Data type: INTEGER_16 Index: $24162_d$ = $5E62_h$ |
|---|---|---|---|
| The L_SignalMonitor_a FB: Gain and offset of the analog signals | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00413/1 | 0.00 % | | Offset for output *nOut1_a* |
| C00413/2 | 100.00 % | | Gain for output *nOut1_a* |
| C00413/3 | 0.00 % | | Offset for output *nOut2_a* |
| C00413/4 | 100.00 % | | Gain for output *nOut2_a* |
| C00413/5 | 0.00 % | | Offset for output *nOut3_a* |
| C00413/6 | 100.00 % | | Gain for output *nOut3_a* |
| C00413/7 | 0.00 % | | Offset for output *nOut4_a* |
| C00413/8 | 100.00 % | | Gain for output *nOut4_a* |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00417**

| Parameter \| Name: **C00417 \| Activate resolver error comp.** | | Data type: UNSIGNED_8 Index: $24158_d$ = $5E5E_h$ |
|---|---|---|
| | | ▸ Encoder/feedback system: Resolver |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **Resolver error comp. act.** | |
| 1 | Resolver error comp. deact. | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00420**

| Parameter \| Name: **C00420 \| Number of encoder increments** | | | Data type: UNSIGNED_16 Index: $24155_d$ = $5E5B_h$ |
|---|---|---|---|
| Indication of the encoder constant | | | |
| | | | Encoder/feedback system: Multi-Encoder |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | Incr./rev. | 32768 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00420/1 | 128 incr./rev. | | FreqIn12: Encoder increment |
| C00420/2 | 128 incr./rev. | | FreqIn67: Encoder increment |
| C00420/3 | 512 incr./rev. | | LS_Multi-Encoder: Encoder increment |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00421**

| Parameter \| Name: **C00421 \| LS_Multi-Encoder: Supply voltage** | | | Data type: UNSIGNED_16 Index: $24154_d$ = $5E5A_h$ |
|---|---|---|---|
| **Stop!** Before connecting an encoder, make sure that the encoder supply voltage is set correctly! If the set supply voltage exceeds the permissible supply voltage of the connected encoder, the encoder may be destroyed! | | | |
| | | | Encoder/feedback system: Multi-Encoder |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 5.0 | V | 12.0 | **5.0 V** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C00422**

| Parameter \| Name: **C00422 \| LS_Multi-Encoder: Encoder type** | | Data type: UNSIGNED_8 Index: $24153_d$ = $5E59_h$ |
|---|---|---|
| | | Encoder/feedback system: Multi-Encoder |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **Incremental encoder (TTL)** | |
| 1 | Sine/cosine encoder | |
| 2 | Absolute value encoder (hiperface) | |
| 4 | Absolute value encoder (SSI) | |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00423

| Parameter \| Name: **C00423 \| DOx: Delay times** | | Data type: UNSIGNED_16 Index: $24152_d$ = $5E58_h$ |
|---|---|---|
| Delay times for the digital output terminals | | |
| | | ▸ Digital output terminals |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0.000 | s | 65.000 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00423/1 | 0.000 s | Relay ON delay |
| C00423/2 | 0.000 s | Relay OFF delay |
| C00423/3 | 0.000 s | DO1 ON delay |
| C00423/4 | 0.000 s | DO1 OFF delay |
| C00423/5 | 0.000 s | DO2 ON delay |
| C00423/6 | 0.000 s | DO2 OFF delay |
| C00423/7 | 0.000 s | DO3 ON delay |
| C00423/8 | 0.000 s | DO3 OFF delay |
| C00423/9 | 0.000 s | DO "High Current" ON delay |
| C00423/10 | 0.000 s | DO "High Current" OFF delay |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

## C00424

| Parameter \| Name: **C00424 \| Pulse form TTL encoder** | Data type: UNSIGNED_8 Index: $24151_d$ = $5E57_h$ |
|---|---|
| | ▸ Encoder/feedback system: Multi-Encoder |
| **Selection list** (Lenze setting printed in bold) | |
| **0** | **4x evaluation A/B** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C00425

| Parameter \| Name: **C00425 \| Encoder scanning time** | Data type: UNSIGNED_8 Index: $24150_d$ = $5E56_h$ |
|---|---|
| Encoder sample time for the digital input terminals when configured as frequency inputs | |
| | ▸ Using DI1(6) and DI2(7) as frequency inputs |
| **Selection list** | |
| 0 | 1 ms |
| 1 | 2 ms |
| 2 | 5 ms |
| 3 | 10 ms |
| 4 | 20 ms |
| 5 | 50 ms |
| 6 | 100 ms |
| 7 | 200 ms |
| 8 | 500 ms |
| 9 | 1000 ms |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00425/1 | 3: 10 ms | FreqIn12: Encoder scanning time<br>• Only active with edge-counting procedure (C00496 = 3). |
| C00425/2 | 3: 10 ms | FreqIn67: Encoder scanning time |

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

### C00426

| Parameter \| Name: **C00426 \| SSI encoder: Data bits** | Data type: UNSIGNED_8<br>Index: $24149_d$ = $5E55_h$ |
|---|---|

Example: Setting for an SSI encode with the coding "8192x4096":
- 8192 = $2^{13}$ = 13 bits Multiturn → C00426/4 = 13
- 4096 = $2^{12}$ = 12 bits Singleturn → C00426/2 = 12
- This results in a data word length of 25 bits → C00426/1 = 25
- Start bit Singleturn = bit 0 → C00426/3 = 0
- Start bit Multiturn = bit 12 → C00426/5 = 12

▸ Encoder/feedback system: SSI encoder

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | | 32 | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00426/1 | 25 | SSI encoder: Data word length |
| C00426/2 | 13 | SSI encoder: Bits SingleTurn |
| C00426/3 | 0 | SSI encoder: Start bit Singleturn |
| C00426/4 | 12 | SSI encoder: Bits Multiturn |
| C00426/5 | 13 | SSI encoder: Start bit Multiturn |
| C00426/6 | 0 | SSI encoder: Status bit<br>• From version 02.00.00 |
| C00426/7 | 0 | SSI encoder: Left shift of raw value<br>• From version 02.00.00 |

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

### C00427

| Parameter \| Name: **C00427 \| SSI encoder: Bit rate** | Data type: UNSIGNED_8<br>Index: $24148_d$ = $5E54_h$ |
|---|---|

▸ Encoder/feedback system: SSI encoder

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **1** | **100 kbps** |
| 2 | 200 kbps |
| 3 | 300 kbps |
| 4 | 400 kbps |
| 5 | 500 kbps |
| 6 | 750 kbps |
| 7 | 1000 kbps |

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

### C00428

| Parameter \| Name: **C00428 \| SSI encoder: Coding** | Data type: UNSIGNED_8<br>Index: $24147_d$ = $5E53_h$ |
|---|---|

▸ Encoder/feedback system: SSI encoder

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **Binary code** |
| 1 | Gray code |

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C00434

| Parameter \| Name: C00434 \| OxU/I: Gain | | Data type: INTEGER_16 Index: $24141_d$ = $5E4D_h$ |
|---|---|---|

Gain of the analog outputs

▸ Analog terminals

| Setting range (min. value \| unit \| max. value) | | |
|---|---|---|
| -199.99 | % | 199.99 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00434/1 | 100.00 % | O1U: Gain |
| C00434/2 | 100.00 % | O2U: Gain |
| C00434/3 | 100.00 % | O1I: Gain |
| C00434/4 | 100.00 % | O2I: Gain |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

## C00435

| Parameter \| Name: C00435 \| OxU/I: Offset | | Data type: INTEGER_16 Index: $24140_d$ = $5E4C_h$ |
|---|---|---|

Offset of the analog outputs

▸ Analog terminals

| Setting range (min. value \| unit \| max. value) | | |
|---|---|---|
| -199.99 | % | 199.99 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00435/1 | 0.00 % | O1U: Offset |
| C00435/2 | 0.00 % | O2U: Offset |
| C00435/3 | 0.00 % | O1I: Offset |
| C00435/4 | 0.00 % | O2I: Offset |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

## C00436

| Parameter \| Name: C00436 \| OxU: Voltage | | Data type: INTEGER_16 Index: $24139_d$ = $5E4B_h$ |
|---|---|---|

Display of the voltage at the analog outputs

▸ Analog terminals

| Display range (min. value \| unit \| max. value) | | |
|---|---|---|
| 0.00 | V | 10.00 |

| Subcodes | Info |
|---|---|
| C00436/1 | O1U: Voltage |
| C00436/2 | O2U: Voltage |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100

## C00437

| Parameter \| Name: C00437 \| OxI: Current | | Data type: INTEGER_32 Index: $24138_d$ = $5E4A_h$ |
|---|---|---|

Display of the current at the analog outputs

▸ Analog terminals

| Display range (min. value \| unit \| max. value) | | |
|---|---|---|
| 0.000 | mA | 20.000 |

| Subcodes | Info |
|---|---|
| C00437/1 | O1I: Current |
| C00437/2 | O2I: Current |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C00439**

| Parameter \| Name: **C00439 \| OxU/I: Input value** | Data type: INTEGER_16 Index: 24136d = 5E48h |
|---|---|

Display of the input values for the analog outputs

Analog terminals

| Display range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00439/1 | | | O1U: Input value |
| C00439/2 | | | O2U: Input value |
| C00439/3 | | | O1I: Input value |
| C00439/4 | | | O2I: Input value |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00440**

| Parameter \| Name: **C00440 \| LS_AnalogIn1: PT1 time constant** | Data type: UNSIGNED_16 Index: 24135d = 5E47h |
|---|---|

PT1 time constant (S-ramp time) for the analog inputs

Analog terminals

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | ms | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00440/1 | 10 ms | | PT1 rounding AnalogIn1 |
| C00440/2 | 10 ms | | PT1 rounding AnalogIn2 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00441**

| Parameter \| Name: **C00441 \| Decoupling AnalogOut** | | | Data type: UNSIGNED_16 Index: $24134_d$ = $5E46_h$ |
|---|---|---|---|
| Configuration defining the events that lead to a decoupling of the analog output terminals. ▸ Configuring exception handling of the output terminals | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded:** (☑ = bit set) | | | |
| Bit 0 ☐ | SafeTorqueOff | | |
| Bit 1 ☐ | ReadyToSwitchOn | | |
| Bit 2 ☐ | SwitchedOn | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | Trouble | | |
| Bit 5 ☐ | Fault | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Fail CAN_Management | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C00442**

| Parameter \| Name: **C00442 \| AOutx: Decoupling value** | | | Data type: INTEGER_16 Index: $24133_d$ = $5E45_h$ |
|---|---|---|---|
| Definition of the value the analog output terminals are to have in the decoupled state. ▸ Configuring exception handling of the output terminals | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00442/1 | 0.00 % | | AOut1_U: Decoupling value |
| C00442/2 | 0.00 % | | AOut2_U: Decoupling value |
| C00442/3 | 0.00 % | | AOut1_I: Decoupling value |
| C00442/4 | 0.00 % | | AOut2_I: Decoupling value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00443**

| Parameter \| Name: **C00443 \| DIx: Level** | | | Data type: UNSIGNED_16<br>Index: $24132_d$ = $5E44_h$ |
|---|---|---|---|
| Bit coded display of the level of the digital inputs | | | |
| | | | ▶ Digital input terminals |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | DI1 | | Bit set = HIGH level |
| Bit 1 | DI2 | | |
| Bit 2 | DI3 | | |
| Bit 3 | DI4 | | |
| Bit 4 | DI5 | | |
| Bit 5 | DI6 | | |
| Bit 6 | DI7 | | |
| Bit 7 | Reserve | | |
| Bit 8 | Reserve | | |
| Bit 9 | Reserve | | |
| Bit 10 | Reserve | | |
| Bit 11 | Reserve | | |
| Bit 12 | Reserve | | |
| Bit 13 | Reserve | | |
| Bit 14 | Reserve | | |
| Bit 15 | CINH | | |
| **Subcodes** | | | **Info** |
| C00443/1 | | | DIx: Terminal level |
| C00443/2 | | | DIx: Output level |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

**C00444**

| Parameter \| Name: **C00444 \| DOx: Level** | | | Data type: UNSIGNED_16 Index: $24131_d$ = $5E43_h$ |
|---|---|---|---|
| Bit coded display of the level of the digital outputs | | | |
| | | | ▸ Digital output terminals |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Relay | | Bit set = HIGH level |
| Bit 1 | DO1 | | |
| Bit 2 | DO2 | | |
| Bit 3 | DO3 | | |
| Bit 4 | High current | | |
| Bit 5 | Reserve | | |
| Bit 6 | Reserve | | |
| Bit 7 | Reserve | | |
| Bit 8 | Reserve | | |
| Bit 9 | Reserve | | |
| Bit 10 | Reserve | | |
| Bit 11 | Reserve | | |
| Bit 12 | Reserve | | |
| Bit 13 | Reserve | | |
| Bit 14 | Reserve | | |
| Bit 15 | Reserve | | |
| **Subcodes** | | | **Info** |
| C00444/1 | | | DOx: Input level |
| C00444/2 | | | DOx: Terminal level |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00445**

| Parameter \| Name: **C00445 \| FreqInxx_nOut_v** | | | Data type: INTEGER_16 Index: $24130_d$ = $5E42_h$ |
|---|---|---|---|
| Display of the frequency input signals which are fed into the application. | | | |
| | | | ▸ Using DI1(6) and DI2(7) as frequency inputs |
| **Display range** (min. value \| unit \| max. value) | | | |
| -32767 | Incr/ms | 32767 | |
| **Subcodes** | | | **Info** |
| C00445/1 | | | FreqIn12_nOut_v |
| C00445/2 | | | FreqIn67_nOut_v |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00446**

| Parameter \| Name:<br>**C00446 \| FreqInxx_nOut_a** | | | Data type: INTEGER_16<br>Index: $24129_d$ = $5E41_h$ |
|---|---|---|---|
| Display of the frequency input signals which are fed into the application. | | | |
| | | | ▶ Using DI1(6) and DI2(7) as frequency inputs |
| **Display range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00446/1 | | | FreqIn12_nOut_a |
| C00446/2 | | | FreqIn67_nOut_a |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00447**

| Parameter \| Name:<br>**C00447 \| DigOut decoupling** | | | Data type: UNSIGNED_16<br>Index: $24128_d$ = $5E40_h$ |
|---|---|---|---|
| Configuration defining the events that lead to a decoupling of the digital output terminals. | | | |
| | | | ▶ Configuring exception handling of the output terminals |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded:** (☑ = bit set) | | | |
| Bit 0 ☐ | SafeTorqueOff | | |
| Bit 1 ☐ | ReadyToSwitchOn | | |
| Bit 2 ☐ | SwitchedOn | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | Trouble | | |
| Bit 5 ☐ | Fault | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Fail CAN_Management | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

## C00448

| Parameter \| Name: **C00448 \| DigOut decoupling value** | | | Data type: UNSIGNED_16 Index: $24127_d$ = $5E3F_h$ |
|---|---|---|---|
| Definition of the value the digital output terminals are to have in the decoupled state.<br>• Bit set = HIGH level | | | |
| | | | Configuring exception handling of the output terminals |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |
| **Value is bit-coded:** (☑ = bit set) | | | |
| Bit 0 ☐ | Relay_ON | | |
| Bit 1 ☐ | DigOut1_ON | | |
| Bit 2 ☐ | DigOut2_ON | | |
| Bit 3 ☐ | DigOut3_ON | | |
| Bit 4 ☐ | DigOut4_ON | | |
| Bit 5 ☐ | HighCurrent_ON | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

## C00449

| Parameter \| Name: **C00449 \| FreqInxx_dnOut_p** | | | Data type: INTEGER_32 Index: $24126_d$ = $5E3E_h$ |
|---|---|---|---|
| | | | Output of the encoder position of the DI1/DI2 frequency input |
| **Display range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00449/1 | | | FreqIn12_dnOut_p |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00455

| Parameter \| Name: **C00455 \| FB_call table** | Data type: UNSIGNED_16 Index: $24120_d$ = $5E38_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00456

| Parameter \| Name: **C00456 \| Editor level** | Data type: UNSIGNED_8 Index: $24119_d$ = $5E37_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00458

| Parameter \| Name: **C00458 \| SYS_call table** | Data type: UNSIGNED_16 Index: $24117_d$ = $5E35_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00459**

| Parameter \| Name: **C00459 \| SYS_Output table** | Data type: UNSIGNED_16 Index: $24116_d$ = $5E34_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00461**

| Parameter \| Name: **C00461 \| Remote: Acceleration/deceleration time** | | | Data type: UNSIGNED_32 Index: $24114_d$ = $5E32_h$ |
|---|---|---|---|
| | | | ▶ PC manual control |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00461/1 | 2.000 s | | Remote: Acceleration/deceleration time |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C00462**

| Parameter \| Name: **C00462 \| Remote: Control** | | | Data type: UNSIGNED_16 Index: $24113_d$ = $5E31_h$ |
|---|---|---|---|
| | | | ▶ PC manual control |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00462/1 | 0 | | Remote: Control mode |
| C00462/2 | 0 | | Remote: Monitoring counter |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00463**

| Parameter \| Name: **C00463 \| Remote: MCK control** | | | Data type: UNSIGNED_32 Index: $24112_d$ = $5E30_h$ |
|---|---|---|---|
| This parameter serves to control the functions of the Motion Control Kernel for PC manual control. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x00000000 | | 0xFFFFFFFF | |
| **Value is bit-coded:** (☑ = bit set) | | | |
| Bit 0 ☐ | OpMode_Bit0 | | |
| Bit 1 ☐ | OpMode_Bit1 | | |
| Bit 2 ☐ | OpMode_Bit2 | | |
| Bit 3 ☐ | OpMode_Bit3 | | |
| Bit 4 ☐ | ManJogPos | | |
| Bit 5 ☐ | ManJogNeg | | |
| Bit 6 ☐ | ManExecute2ndSpeed | | |
| Bit 7 ☐ | ReleaseLimitSwitch | | |
| Bit 8 ☐ | HomStartStop | | |
| Bit 9 ☐ | HomSetPos | | |
| Bit 10 ☐ | HomResetPos | | |
| Bit 11 ☐ | EnableSpeedOverride | | |
| Bit 12 ☐ | EnableAccOverride | | |
| Bit 13 ☐ | EnableSRampOverride | | |
| Bit 14 ☐ | PosTeachSetPos | | |
| Bit 15 ☐ | PosTeachActPos | | |
| Bit 16 ☐ | PosExecute | | |
| Bit 17 ☐ | PosFinishTarget | | |
| Bit 18 ☐ | PosDisableFollowProfile | | |
| Bit 19 ☐ | PosStop | | |
| Bit 20 ☐ | PosModeBit0 | | |
| Bit 21 ☐ | PosModeBit1 | | |
| Bit 22 ☐ | PosModeBit2 | | |
| Bit 23 ☐ | PosModeBit3 | | |
| Bit 24 ☐ | ProfileNo_Bit0 | | |
| Bit 25 ☐ | ProfileNo_Bit1 | | |
| Bit 26 ☐ | ProfileNo_Bit2 | | |
| Bit 27 ☐ | ProfileNo_Bit3 | | |
| Bit 28 ☐ | ProfileNo_Bit4 | | |
| Bit 29 ☐ | ProfileNo_Bit5 | | |
| Bit 30 ☐ | ProfileNo_Bit6 | | |
| Bit 31 ☐ | ProfileNo_Bit7 | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00463/1 | 0 | | Remote: MCK control |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00464**

| Parameter \| Name: **C00464 \| Remote: Monitoring timeout** | | | Data type: UNSIGNED_16 Index: 24111$_d$ = 5E2F$_h$ |
|---|---|---|---|
| | | | ▶ PC manual control |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 200 | ms | 5000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00464/1 | 2000 ms | | Remote: Monitoring timeout |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00465**

| Parameter \| Name: **C00465 \| Keypad: Timeout welcome screen** | | Data type: INTEGER_32 Index: 24110$_d$ = 5E2E$_h$ |
|---|---|---|
| Time setting for the automatic change of the keypad display to the welcome screen | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **Never display welcome screen** | |
| 5 | 5 min | |
| 15 | 15 min | |
| 30 | 30 min | |
| 60 | 60 min | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00466**

| Parameter \| Name: **C00466 \| Keypad: Default parameter** | | | Data type: INTEGER_32 Index: 24109$_d$ = 5E2D$_h$ |
|---|---|---|---|
| Setting of the default parameter for the keypad | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 65535 | **51** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00467**

| Parameter \| Name: **C00467 \| Keypad: Default welcome screen** | | Data type: INTEGER_32 Index: 24108$_d$ = 5E2C$_h$ |
|---|---|---|
| Selection of the welcome screen for the keypad | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | Main menu | |
| **1** | **Parameter list** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00468**

| Parameter \| Name: **C00468 \| Service code** | Data type: INTEGER_32 Index: 24107$_d$ = 5E2B$_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00469**

| Parameter \| Name: **C00469 \| Keypad: STOP key function** | | Data type: INTEGER_32 Index: 24106$_d$ = 5E2A$_h$ |
|---|---|---|
| Selection of the function for the STOP key on the keypad | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | No function | STOP key does not have any function |
| **1** | **Inhibit controller** | STOP key sets controller inhibit in the drive |
| 2 | Activate quick stop | STOP key sets quick stop in the drive |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00470**

| Parameter \| Name: **C00470 \| LS_ParFree_b** | | Data type: UNSIGNED_8 Index: $24105_d$ = $5E29_h$ |
|---|---|---|
| The LS_ParFree_b SB: Setting of the signal level to be output | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00470/1 | 0: FALSE | Signal level for output *bPar1* ... *bPar32* |
| C00470/... | | |
| C00470/32 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00471**

| Parameter \| Name: **C00471 \| LS_ParFree** | | | Data type: UNSIGNED_16 Index: $24104_d$ = $5E28_h$ |
|---|---|---|---|
| The LS_ParFree SB: Setting of the words to be output | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Bit0 | | |
| ... | ... | | |
| Bit 15 | Bit15 | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00471/1 | 0 | | Value for output *wPar1* ... *wPar32* |
| C00471/... | | | |
| C00471/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

## C00472

| Parameter \| Name: **C00472 \| LS_ParFree_a** | | Data type: INTEGER_16 Index: $24103_d$ = $5E27_h$ |
|---|---|---|
| The LS_ParFree_a SB: Setting of the analog signals to be output | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -199.99 % 199.99 | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00472/1 | 0.00 % | Value for output *nPar1_a* |
| C00472/2 | 0.00 % | Value for output *nPar2_a* |
| C00472/3 | 100.00 % | Value for output *nPar3_a* |
| C00472/4 | 100.00 % | Value for output *nPar4_a* |
| C00472/5 | 0.00 % | Value for output *nPar5_a* |
| C00472/6 | 0.00 % | Value for output *nPar6_a* |
| C00472/7 | 0.00 % | Value for output *nPar7_a* |
| C00472/8 | 0.00 % | Value for output *nPar8_a* |
| C00472/9 | 0.00 % | Value for output *nPar9_a* |
| C00472/10 | 0.00 % | Value for output *nPar10_a* |
| C00472/11 | 0.00 % | Value for output *nPar11_a* |
| C00472/12 | 0.00 % | Value for output *nPar12_a* |
| C00472/13 | 0.00 % | Value for output *nPar13_a* |
| C00472/14 | 0.00 % | Value for output *nPar14_a* |
| C00472/15 | 0.00 % | Value for output *nPar15_a* |
| C00472/16 | 0.00 % | Value for output *nPar16_a* |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | |

## C00473

| Parameter \| Name: **C00473 \| LS_ParFree_v** | | Data type: INTEGER_16 Index: $24102_d$ = $5E26_h$ |
|---|---|---|
| The LS_ParFree_v SB: Setting of the speed signals to be output | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -32767 Incr./ms 32767 | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00473/1 | 0 incr./ms | Values for output *nPar1_v* ... *nPar8_v* |
| C00473/... | | |
| C00473/8 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00474

| Parameter \| Name: **C00474 \| LS_ParFree_p** | | Data type: INTEGER_32 Index: $24101_d$ = $5E25_h$ |
|---|---|---|
| SB LS_ParFree_p: Setting of the position signals to be output | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -2147483647 Incr. 2147483647 | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00474/1 | 0 incr. | Values for output *dnPar1_p* ... *dnPar8_p* |
| C00474/... | | |
| C00474/8 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00475**

| Parameter \| Name: **C00475 \| LS_ParFreeUnit_1_2** | | | Data type: INTEGER_32 Index: $24100_d$ = $5E24_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| SB LS_ParFreeUnit / LS_ParFreeUnit_2: Setting of the position signals to be output | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00475/1 | 0.0000 units | | Values for output *dnC475_1 ... dnC475_32* |
| C00475/... | | | |
| C00475/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C00476**

| Parameter \| Name: **C00476 \| LS_ParFree_a_2** | | | Data type: INTEGER_16 Index: $24099_d$ = $5E23_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| SB LS_ParFree_a_2: Setting of the analog signals to be output | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00476/1 | 0.00 % | | Value for output *nC476_1_a* |
| C00476/2 | 0.00 % | | Value for output *nC476_2_a* |
| C00476/3 | 0.00 % | | Value for output *nC476_3_a* |
| C00476/4 | 0.00 % | | Value for output *nC476_4_a* |
| C00476/5 | 0.00 % | | Value for output *nC476_5_a* |
| C00476/6 | 0.00 % | | Value for output *nC476_6_a* |
| C00476/7 | 0.00 % | | Value for output *nC476_7_a* |
| C00476/8 | 0.00 % | | Value for output *nC476_8_a* |
| C00476/9 | 0.00 % | | Value for output *nC476_9_a* |
| C00476/10 | 0.00 % | | Value for output *nC476_10_a* |
| C00476/11 | 0.00 % | | Value for output *nC476_11_a* |
| C00476/12 | 0.00 % | | Value for output *nC476_12_a* |
| C00476/13 | 0.00 % | | Value for output *nC476_13_a* |
| C00476/14 | 0.00 % | | Value for output *nC476_14_a* |
| C00476/15 | 0.00 % | | Value for output *nC476_15_a* |
| C00476/16 | 0.00 % | | Value for output *nC476_16_a* |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00477**

| Parameter \| Name: **C00477 \| LS_ParFree_2** | | Data type: UNSIGNED_16 Index: 24098$_d$ = 5E22$_h$ |
|---|---|---|
| From version 02.00.00 | | |
| SB LS_ParFree_2: Setting of the words to be output | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | |
| Bit 0 | Bit0 | |
| ... | ... | |
| Bit 15 | Bit15 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00477/1 | 0 | Value for output *wC477_1 ... wC477_32* |
| C00477/... | | |
| C00477/32 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C00478**

| Parameter \| Name: **C00478 \| LS_ParFree_v_2** | | Data type: INTEGER_16 Index: 24097$_d$ = 5E21$_h$ |
|---|---|---|
| From version 02.00.00 | | |
| SB LS_ParFree_v_2: Setting of the speed signals to be output | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -32767 | Incr./ms | 32767 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00478/1 | 0 incr./ms | Values for output *nC478_1_v ... nC478_8_v* |
| C00478/... | | |
| C00478/8 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00479**

| Parameter \| Name: **C00479 \| LS_ParFree32** | | Data type: INTEGER_32 Index: 24096$_d$ = 5E20$_h$ |
|---|---|---|
| From version 02.00.00 | | |
| SB LS_ParFree32: Setting of the 32-bit values to be output | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -2147483647 | | 2147483647 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00479/1 | 0 | Values for output *dnC479_1 ... dnC479_8* |
| C00479/... | | |
| C00479/8 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00480**

| Parameter \| Name: **C00480 \| LS_DisFree_b** | | | Data type: UNSIGNED_16 Index: $24095_d$ = $5E1F_h$ |
|---|---|---|---|
| The LS_DisFree_b SB: Display of the input values | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | bDis1 | | Signal level input *bDis1 ... bDis16* |
| ... | ... | | |
| Bit 15 | bDis16 | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00481**

| Parameter \| Name: **C00481 \| LS_DisFree** | | | Data type: UNSIGNED_16 Index: $24094_d$ = $5E1E_h$ |
|---|---|---|---|
| The LS_DisFree SB: Display of the input values | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00481/1 | | | Input values *wDis1 ... wDis8* |
| C00481/... | | | |
| C00481/8 | | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00482**

| Parameter \| Name: **C00482 \| LS_DisFree_a** | | | Data type: INTEGER_16 Index: $24093_d$ = $5E1D_h$ |
|---|---|---|---|
| The LS_DisFree_a SB: Display of the input values | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00482/1 | | | Input values *nDis1_a ... nDis8_a* |
| C00482/... | | | |
| C00482/8 | | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00483**

| Parameter \| Name: **C00483 \| LS_DisFree_p** | | | Data type: INTEGER_32 Index: $24092_d$ = $5E1C_h$ |
|---|---|---|---|
| SB LS_DisFree_p: Display of the input values | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00483/1 | | | Input values *dnDis1_p ... dnDis8_p* |
| C00483/... | | | |
| C00483/8 | | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00484**

| Parameter \| Name: **C00484 \| Application units: Offset** | | Data type: INTEGER_16 Index: $24091_d$ = $5E1B_h$ |
|---|---|---|
| The LS_DisFree_a SB: Offset for display of the input variables in application unit ▸ Display of internal process factors in application units | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -199.99 | % | 199.99 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00484/1 | 0.00 % | Application unit 1: Offset |
| C00484/2 | 0.00 % | Application unit 2: Offset |
| C00484/3 | 0.00 % | Application unit 3: Offset |
| C00484/4 | 0.00 % | Application unit 4: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | |

**C00485**

| Parameter \| Name: **C00485 \| Application units - display factor** | | Data type: INTEGER_32 Index: $24090_d$ = $5E1A_h$ |
|---|---|---|
| The LS_DisFree_a SB: Display factor for display of the input variables in application unit ▸ Display of internal process factors in application units | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -65536.0000 | | 65536.0000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00485/1 | 1.0000 | Application unit 1: Display factor |
| C00485/2 | 1.0000 | Application unit 2: Display factor |
| C00485/3 | 1.0000 | Application unit 3: Display factor |
| C00485/4 | 1.0000 | Application unit 4: Display factor |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | |

**C00486**

| Parameter \| Name: **C00486 \| Application units:** | | Data type: VISIBLE_STRING Index: $24089_d$ = $5E19_h$ |
|---|---|---|
| The LS_DisFree_a SB: Text for the display of the input variables in application unit ▸ Display of internal process factors in application units | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00486/1 | | Application unit 1: Text |
| C00486/2 | | Application unit 2: Text |
| C00486/3 | | Application unit 3: Text |
| C00486/4 | | Application unit 4: Text |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ COM ☐ MOT Character length: 7 | | |

**C00487**

| Parameter \| Name: **C00487 - Application units** | | | Data type: INTEGER_32 Index: $24088_d$ = $5E18_h$ |
|---|---|---|---|
| SB LS_DisFree_a: Display of the input values in a configurable application unit | | | |
| | | | Display of internal process factors in application units |
| **Display range** (min. value \| unit \| max. value) | | | |
| -21474836.47 | units | 21474836.47 | |
| **Subcodes** | | | **Info** |
| C00487/1 | | | Application units 1 |
| C00487/2 | | | Application units 2 |
| C00487/3 | | | Application units 3 |
| C00487/4 | | | Application units 4 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00488**

| Parameter \| Name: **C00488 \| L_JogCtrlEdgeDetect** | | Data type: UNSIGNED_8 Index: $24087_d$ = $5E17_h$ |
|---|---|---|
| The L_JogCtrlExtension_1 FB: Signal methodology<br>• Selection whether the corresponding function is to be activated by edge or level. | | |
| **Selection list** | | |
| 0 | Level | |
| 1 | Edge | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00488/1 | 0: Level | InputSens.SlowDown1<br>• Selection of edge or level for starting slow-down function 1 |
| C00488/2 | 0: Level | InputSens.Stop1<br>• Selection of edge or level for stop function 1 |
| C00488/3 | 0: Level | InputSens.SlowDown2<br>• Selection of edge or level for starting slow-down function 2 |
| C00488/4 | 0: Level | InputSens.Stop2<br>• Selection of edge or level for stop function 2 |
| C00488/5 | 0: Level | InputSens.SlowDown3<br>• Selection of edge or level for starting slow-down function 3 |
| C00488/6 | 0: Level | InputSens.Stop3<br>• Selection of edge or level for stop function 3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00490**

| Parameter \| Name: **C00490 \| Position encoder selection** | | | Data type: UNSIGNED_8 Index: $24085_d$ = $5E15_h$ |
|---|---|---|---|
| Selection of the feedback system for the generation of the actual position on the load | | | |
| | | | ▸ Encoder/feedback system |
| **Selection list** (Lenze setting printed in bold) | | | **Info** |
| **0** | **No sensor** | | No encoder available on the load |
| 1 | Sensor signal FreqIn12 | | Position encoder signal is fed via digital inputs DI1 and DI2 |
| 2 | Encoder signal FreqIn67 | | Position encoder signal is fed via digital inputs DI6 and DI7 |
| 3 | Multi encoder | | Position encoder signal is supplied via multi-encoder interface |
| 4 | Resolver | | Position encoder signal is supplied via resolver interface |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00492**

| Parameter \| Name: **C00492 \| Hiperface: Detected TypCode** | | | Data type: UNSIGNED_8 Index: $24083_d$ = $5E13_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| | | | ▸ Encoder/feedback system: Absolute value encoder (sin/cos encoder with hiperface) |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00493**

| Parameter \| Name: **C00493 \| Hiperface: TypCode** | | | Data type: UNSIGNED_8 Index: $24082_d$ = $5E12_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| | | | ▸ Encoder/feedback system: Absolute value encoder (sin/cos encoder with hiperface) |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 255 | **0** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00494**

| Parameter \| Name: **C00494 \| Hiperface: Resolutions** | | | Data type: UNSIGNED_32 Index: $24081_d$ = $5E11_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| | | | ▸ Encoder/feedback system: Absolute value encoder (sin/cos encoder with hiperface) |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65536 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00494/1 | 0 | | Hiperface: Number of revolutions |
| C00494/2 | 0 | | Hiperface: Steps per revolution |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00495

| Parameter \| Name: C00495 \| Speed sensor selection | | Data type: UNSIGNED_8 Index: $24080_d$ = $5E10_h$ |
|---|---|---|

Selection of the feedback system for the actual speed for motor control and display

Encoder/feedback system

| Selection list (Lenze setting printed in bold) | | Info |
|---|---|---|
| **0** | **No sensor** | No sensor available for the actual speed detection |
| 1 | Sensor signal FreqIn12 | Speed sensor signal is fed via the digital DI1 and DI2 inputs |
| 2 | Encoder signal FreqIn67 | Speed encoder signal is fed via digital inputs DI6 and DI7 |
| 3 | Multi encoder | Speed encoder signal is supplied via multi-encoder interface |
| 4 | Resolver | Speed encoder signal is supplied via resolver interface |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C00496

| Parameter \| Name: C00496 \| Encoder evaluation method DigIn12 | | Data type: UNSIGNED_8 Index: $24079_d$ = $5E0F_h$ |
|---|---|---|

Encoder/feedback system

| Selection list (Lenze setting printed in bold) | | Info |
|---|---|---|
| 0 | High-resolution encoders | High-precision procedure for high-resolution encoders (>=512 increments) |
| 1 | Low-resolution encoder (StateLine) | High-precision procedure for low-resolution encoders (<=128 increments) |
| **2** | **Comb. encoder procedure** | Combination of the first two procedures as a function of the speed (recommended procedure) |
| 3 | Edge-counting procedure | Simple edge counting procedure with adjustable scanning time (C00425) |

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C00497

| Parameter \| Name: C00497 \| Nact filter time constant | | Data type: UNSIGNED_16 Index: $24078_d$ = $5E0E_h$ |
|---|---|---|

Encoder/feedback system

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0.0 | ms | 500.0 | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00497/1 | 1.0 ms | FreqIn12: Encoder filter time |
| C00497/2 | 1.0 ms | FreqIn67: Encoder filter time |
| C00497/3 | 1.0 ms | LS_Multi-Encoder: Encoder filter time |
| C00497/4 | 2.0 ms | LS_Resolver: Encoder filter time |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10

**C00498**

| Parameter \| Name: **C00498 \| Open-circuit monitoring** | | Data type: UNSIGNED_8 Index: $24077_d$ = $5E0D_h$ |
|---|---|---|

Configuration of open-circuit monitoring for encoder/feedback system
- We recommend using the Lenze setting.
- The settings 1 ... 3 make sense if the respective encoder is not to be used as speed encoder or position encoder but within the scope of the application. Moreover, a directed disconnection of the monitoring is possible.

▶ Encoder/feedback system

| Selection list | | Info |
|---|---|---|
| 0 | Speed and position encoder | Open-circuit monitoring is active for the speed encoder selected in C00495 and the position encoder selected in C00490.<br>• Depending on whether a speed encoder or position encoder has been selected, the monitoring mode for the resolver and/or encoder is activated.<br>• If no speed encoder and position encoder have been selected, open-circuit monitoring is deactivated. |
| 1 | Resolver only | Open-circuit monitoring is only active for the resolver, independent on the selection of the speed encoder and position encoder.<br>• Open-circuit monitoring for the encoder is deactivated.<br>Open-circuit monitoring is active for the resolver and encoder, independent on the selection of the speed encoder and position encoder. |
| 2 | Encoder only | Open-circuit monitoring is only active for the encoder, independent on the selection of the speed encoder and position encoder.<br>• Open-circuit monitoring for the resolver is deactivated. |
| 3 | Resolver and encoder | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00498/1 | 0: Speed encoder and position encoder | Open-circuit monitoring |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00505**

| Parameter \| Name: **C00505 \| Password data** | | Data type: VISIBLE_STRING Index: $24070_d$ = $5E06_h$ |
|---|---|---|

▶ Device access protection

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00505/1 | | MasterPin |
| C00505/2 | | Binding ID |
| C00505/3 | | Password |

☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ COM ☐ MOT Character length: 16

## C00507

| Parameter \| Name: **C00507 \| Current password protection** | Data type: UNSIGNED_16 Index: $24068_d$ = $5E04_h$ |
|---|---|

Display of the currently active device access protection

Device access protection

| **Display area** (min. hex value \| max. hex value) | | | |
|---|---|---|---|
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Only access to user menu | | |
| Bit 1 | Parameter write protection | | |
| Bit 2 | Parameter read protection | | |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Memory module binding on | | |

| Subcodes | Info |
|---|---|
| C00507/1 | Password protection - all communication channels |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

## C00516

| Parameter \| Name: **C00516 \| Checksums** | Data type: UNSIGNED_32 Index: $24059_d$ = $5DFB_h$ |
|---|---|

**This code is for device-internal use only and must not be written to by the user!**

**C00517**

| Parameter \| Name: **C00517 \| User menu** | | | Data type: INTEGER_32 Index: $24058_d$ = $5DFA_h$ |
|---|---|---|---|
| When a system is installed, parameters must be changed time and again until the system runs satisfactorily. The user menu of a device serves to create a selection of frequently used parameters to be able to access and change these parameters quickly.<br>• Format: <code number>,<subcode number><br>• If "0.000" is set, no entry will be displayed in the user menu. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | | 16000.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00517/1 | 51.000 | | C00051: Display of actual speed value |
| C00517/2 | 53.000 | | C00053: Display of DC-bus voltage |
| C00517/3 | 54.000 | | C00054: Display of motor current |
| C00517/4 | 61.000 | | C00061: Display of heatsink temperature |
| C00517/5 | 137.000 | | C00137: Display of device status |
| C00517/6 | 166.003 | | C00166/3: Display of current error message |
| C00517/7 | 0.000 | | User menu: Entry 7 |
| C00517/8 | 11.000 | | C00011: Reference speed |
| C00517/9 | 39.001 | | C00039/1: Fixed setpoint 1 |
| C00517/10 | 39.002 | | C00039/2: Fixed setpoint 2 |
| C00517/11 | 12.000 | | C00012: Accel. time - main setpoint |
| C00517/12 | 13.000 | | C00013: Decel. time - main setpoint |
| C00517/13 | 15.000 | | C00015: V/f base frequency |
| C00517/14 | 16.000 | | C00016: Vmin boost |
| C00517/15 | 22.000 | | C00022: Imax in motor mode |
| C00517/16 | 120.000 | | C00120: Setting of motor overload ($I^2xt$) |
| C00517/17 | 87.000 | | C00087: Rated motor speed |
| C00517/18 | 99.000 | | C00099: Display of firmware version |
| C00517/19 | 200.000 | | C00200: Display of firmware product type |
| C00517/20 | 0.000 | | User menu: Entry 20 |
| C00517/21 | 0.000 | | User menu: Entry 21 |
| C00517/22 | 0.000 | | User menu: Entry 22 |
| C00517/23 | 0.000 | | User menu: Entry 23 |
| C00517/24 | 105.000 | | C00105: Decel. time - quick stop |
| C00517/25 | 173.000 | | C00173: Mains voltage |
| C00517/26 | 0.000 | | User menu: Entry 26 |
| C00517/27 | 0.000 | | User menu: Entry 27 |
| C00517/28 | 0.000 | | User menu: Entry 28 |
| C00517/29 | 0.000 | | User menu: Entry 29 |
| C00517/30 | 0.000 | | User menu: Entry 30 |
| C00517/31 | 0.000 | | User menu: Entry 31 |
| C00517/32 | 0.000 | | User menu: Entry 32 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00560**

| Parameter \| Name: **C00560 \| Fan switching status** | Data type: UNSIGNED_8 Index: $24015_d$ = $5DCF_h$ |
|---|---|
| Display of the function status of the device fans | |
| **Selection list** | |
| 0 Off | |
| 1 On | |
| 2 No fan | |
| **Subcodes** | **Info** |
| C00560/1 | Switching status - internal fan |
| C00560/2 | Switching status - heatsink fan |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00561**

| Parameter \| Name: **C00561 \| Failure indication** | Data type: UNSIGNED_8 Index: $24014_d$ = $5DCE_h$ |
|---|---|
| Failure display of device fans and motor phases | |
| **Selection list** | |
| 0 No error | |
| 1 Error | |
| **Subcodes** | **Info** |
| C00561/1 | Internal fan |
| C00561/2 | Heatsink fan |
| C00561/3 | Motor phase U |
| C00561/4 | Motor phase V |
| C00561/5 | Motor phase W |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00565**

| Parameter \| Name: **C00565 \| Resp. to mains phase failure** | Data type: UNSIGNED_8 Index: $24010_d$ = $5DCA_h$ |
|---|---|
| Response to the failure of mains phases | |
| **Selection list** (Lenze setting printed in bold) | |
| 0 No Reaction | |
| 1 Fault | |
| **5 Warning** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00566**

| Parameter \| Name: **C00566 \| Resp. to fan failure** | Data type: UNSIGNED_8 Index: $24009_d$ = $5DC9_h$ |
|---|---|
| Response to the detection of a fan failure | |
| **Selection list** (Lenze setting printed in bold) | |
| 0 No Reaction | |
| 1 Fault | |
| **5 Warning** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

### C00567

| Parameter \| Name: **C00567 \| Resp. to speed controller limited** | | Data type: UNSIGNED_8 Index: $24008_d$ = $5DC8_h$ |
|---|---|---|
| Response if speed controller output is limited (*bLimSpeedCtrlOut* = TRUE) | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **No Reaction** | |
| 1 | Fault | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### C00569

| Parameter \| Name: **C00569 \| Resp. to peak current** | | Data type: UNSIGNED_8 Index: $24006_d$ = $5DC6_h$ |
|---|---|---|
| Configuration of monitoring of the motor control (group 1) | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 5 | Warning | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00569/1 | 0: No Reaction | Response for overcurrent detection and clamp operation |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### C00570

| Parameter \| Name: **C00570 \| Resp. to controller limitations** | | Data type: UNSIGNED_8 Index: $24005_d$ = $5DC5_h$ |
|---|---|---|
| Configuration of monitoring of the motor control (group 2) | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 5 | Warning | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00570/1 | 0: No Reaction | Response if direct-axis current controller is limited<br>• e.g. at servo control (SC) |
| C00570/2 | 0: No Reaction | Response if cross current controller is limited<br>• e.g. at servo control (SC) |
| C00570/3 | 0: No Reaction | Response if torque setpoint is limited<br>• Limitation of the speed controller output, the differential setpoint precontrol, and of the additive torque at (SC) servo control and (SLVC) sensorless vector control. |
| C00570/4 | 0: No Reaction | Response if field controller is limited |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### C00572

| Parameter \| Name: **C00572 \| Brake resistor overload threshold** | | | Data type: UNSIGNED_8 Index: $24003_d$ = $5DC3_h$ |
|---|---|---|---|
| Adjustable threshold for monitoring the brake resistor utilisation<br>• The response for reaching the threshold can be selected in C00574. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | % | 100 | **100 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

C00574

| Parameter | Name: **C00574 | Resp. to brake resist. overtemp.** | | | Data type: UNSIGNED_8 Index: $24001_d$ = $5DC1_h$ |
|---|---|---|---|
| Response which is triggered if the threshold set in C00572 for monitoring brake resistor utilisation is reached. | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| **0** | **No Reaction** | | |
| 1 | Fault | | |
| 5 | Warning | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

C00576

| Parameter | Name: **C00576 | SC: Field feedforward control** | | | Data type: UNSIGNED_16 Index: $23999_d$ = $5DBF_h$ |
|---|---|---|---|
| Point of action of the direct-axis current setpoint precontrol for an early reduction of the field current. In this way, the acceleration behaviour can be improved in the field weakening range.<br>• The entry [%] refers to the slip of the asynchronous motor. | | | |
| **Setting range** (min. value | unit | max. value) | | | **Lenze setting** |
| 0 | % | 600 | **200 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

C00577

| Parameter | Name: **C00577 | SC: Vp field weakening controller** | | | Data type: UNSIGNED_16 Index: $23998_d$ = $5DBE_h$ |
|---|---|---|---|
| Proportional gain of the field weakening controller<br>• When "0" is set, the P component of the controller is deactivated.<br>• The recommended setting is in the range between 0 and 0.0020 | | | |
| **Setting range** (min. value | unit | max. value) | | | **Lenze setting** |
| 0.0000 | | 2.0000 | **0.0010** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

C00578

| Parameter | Name: **C00578 | SC: Tn field weakening controller** | | | Data type: UNSIGNED_16 Index: $23997_d$ = $5DBD_h$ |
|---|---|---|---|
| Time constant of the field weakening controller<br>• The recommended setting is in the range between 10 and 30 ms | | | |
| **Setting range** (min. value | unit | max. value) | | | **Lenze setting** |
| 0.1 | ms | 6000.0 | **20.0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

C00579

| Parameter | Name: **C00579 | Resp. to max. speed/output freq. reached** | | | Data type: UNSIGNED_8 Index: $23996_d$ = $5DBC_h$ |
|---|---|---|---|
| Response when the max. speed limit (C00909) or output frequency limit (C00910) has been reached. | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| **0** | **No Reaction** | | |
| 1 | Fault | | |
| 5 | Warning | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00580**

| Parameter \| Name:<br>**C00580 \| Resp. to operating system error** | | Data type: UNSIGNED_8<br>Index: 23995$_d$ = 5DBB$_h$ |
|---|---|---|
| From version 02.00.00<br>Response if the required computing time of the application exceeds the available computing time. | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 5 | Warning | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00580/1 | 0: No Reaction | Resp. to runtime exceedance |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00581**

| Parameter \| Name:<br>**C00581 \| Resp. to LS_SetError_x** | | Data type: UNSIGNED_8<br>Index: 23994$_d$ = 5DBA$_h$ |
|---|---|---|
| Selection of the error responses for application error messages<br>• An application error message is tripped by a FALSE/TRUE edge at the binary inputs *bSetError1...4*. | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00581/1 | 0: No Reaction | LS_SetError_1: Resp. to bSetError1 |
| C00581/2 | 0: No Reaction | LS_SetError_1: Resp. to bSetError2 |
| C00581/3 | 0: No Reaction | LS_SetError_1: Resp. to bSetError3 |
| C00581/4 | 0: No Reaction | LS_SetError_1: Resp. to bSetError4 |
| C00581/5 | 0: No Reaction | LS_SetError_2: Resp. to bSetError1 |
| C00581/6 | 0: No Reaction | LS_SetError_2: Resp. to bSetError2 |
| C00581/7 | 0: No Reaction | LS_SetError_2: Resp. to bSetError3 |
| C00581/8 | 0: No Reaction | LS_SetError_2: Resp. to bSetError4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00582**

| Parameter \| Name:<br>**C00582 \| Resp. to heatsink temp. > shutdown temp. -5°C** | | Data type: UNSIGNED_8<br>Index: 23993$_d$ = 5DB9$_h$ |
|---|---|---|
| Response if the heatsink temperature has reached the switch-off temperature threshold. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **No Reaction** | |
| 1 | Fault | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00583**

| Parameter \| Name: **C00583 \| Resp. to motor temperature KTY** | | Data type: UNSIGNED_8 Index: $23992_d$ = $5DB8_h$ |
|---|---|---|
| Response to motor overtemperature<br>• The motor temperature is measured via the resolver or encoder cable.<br>▸ Encoder/feedback system: Motor temperature monitoring (KTY) | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00583/1 | 1: Fault | Resp. to motor overtemp. KTY resolver |
| C00583/2 | 1: Fault | Resp. to motor overtemp. KTY MultiEncoder |
| C00583/3 | 5: Warning | Resp. to motor temp. > C00121 resolver |
| C00583/4 | 5: Warning | Resp. to motor temp. > C00121 encoder |
| C00583/5 | 1: Fault | Resp. to temp. sensor error KTY resolver |
| C00583/6 | 1: Fault | Resp. to temp. sensor error KTY MultiEncoder |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00585**

| Parameter \| Name: **C00585 \| Resp. to motor overtemp. PTC** | | Data type: UNSIGNED_8 Index: $23990_d$ = $5DB6_h$ |
|---|---|---|
| Response to motor overtemperature<br>• The motor temperature is measured by means of a PTC thermistor at terminal X106. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| **1** | **Fault** | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00586**

| Parameter \| Name: **C00586 \| Resp. to encoder open circuit HTL** | | Data type: UNSIGNED_8 Index: $23989_d$ = $5DB5_h$ |
|---|---|---|
| Response to encoder feedback system failure or encoder feedback system track failure due to open circuit | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| **1** | **Fault** | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00588**

| Parameter \| Name:<br>**C00588 \| Resp. to max. speed at switching freq.** | Data type: UNSIGNED_8<br>Index: $23987_d$ = $5DB3_h$ |
|---|---|
| Response if the maximum speed for the set inverter switching frequency is reached (C00018) | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **No Reaction** |
| 1 | Fault |
| 5 | Warning |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00590**

| Parameter \| Name:<br>**C00590 \| Resp. to switching frequency red.** | Data type: UNSIGNED_8<br>Index: $23985_d$ = $5DB1_h$ |
|---|---|
| Response to reduction of the inverter switching frequency (C00018) | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **No Reaction** |
| 1 | Fault |
| 5 | Warning |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00592**

| Parameter \| Name:<br>**C00592 \| Resp. to CAN bus connection** | Data type: UNSIGNED_8<br>Index: $23983_d$ = $5DAF_h$ |
|---|---|
| Configuration of monitoring of the CAN interface (group 1)<br>▸ System bus "CAN on board" | |

| **Selection list** | |
|---|---|
| 0 | No Reaction |
| 1 | Fault |
| 2 | Trouble |
| 3 | TroubleQuickStop |
| 4 | WarningLocked |
| 5 | Warning |
| 6 | Information |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00592/1 | 0: No Reaction | Response to incorrect telegram for CAN communication |
| C00592/2 | 0: No Reaction | Response to "BusOff" (bus system switched off) |
| C00592/3 | 0: No Reaction | Response to warnings of the CAN controller |
| C00592/4 | 0: No Reaction | Response to communication stop of a CAN bus node |
| C00592/5 | 0: No Reaction | Response to an event in the case of monitoring via heartbeat protocol |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C00593**

| Parameter \| Name: **C00593 \| Resp. to CANx_IN monitoring** | | Data type: UNSIGNED_8 Index: $23982_d$ = $5DAE_h$ |
|---|---|---|
| Configuration of monitoring of the CAN interface (group 2) | | |
| | | ▸ System bus "CAN on board" |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00593/1 | 0: No Reaction | Response if the monitoring time set in C00357/1 for the reception of the PDO CAN1_IN is exceeded. |
| C00593/2 | 0: No Reaction | Response if the monitoring time set in C00357/2 for the reception of the PDO CAN2_IN is exceeded. |
| C00593/3 | 0: No Reaction | Response if the monitoring time set in C00357/3 for the reception of the PDO CAN3_IN is exceeded. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C00594**

| Parameter \| Name: **C00594 \| Resp. to control word error** | | Data type: UNSIGNED_8 Index: $23981_d$ = $5DAD_h$ |
|---|---|---|
| Configuration of device control monitoring | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 5 | Warning | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00594/1 | 1: Fault | Response if error bit 14 in the CAN control word is set. |
| C00594/2 | 1: Fault | Response if error bit 14 in the MCI control word is set. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00595

| Parameter \| Name: **C00595 \| MCK: Resp. to MCK error** | | Data type: UNSIGNED_8<br>Index: $23980_d$ = $5DAC_h$ |
|---|---|---|
| Configuration of monitoring of the Motion Control Kernel | | |
| | | ▸ Basic drive functions |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00595/1 | 3: TroubleQuickStop | Response if the input *bLimitSwitchPos* for travel range monitoring is set to FALSE (fail-safe) by the positive hardware limit switch . |
| C00595/2 | 3: TroubleQuickStop | Response if the input *bLimitSwitchNeg* for travel range monitoring is set to FALSE (fail-safe) by the negative hardware limit switch . |
| C00595/3 | 3: TroubleQuickStop | Response for detection that the position is beyond the positive software limit position (C01229/1). |
| C00595/4 | 3: TroubleQuickStop | Response for detection that the position is beyond the negative software limit position (C01229/2). |
| C00595/5 | 5: Warning | Response if following error limit 1 is exceeded (C01215/1) |
| C00595/6 | 5: Warning | Response if following error limit 2 is exceeded (C01215/2) |
| C00595/7 | 3: TroubleQuickStop | Response if the maximum travel distance (display in C01213/1) is exceeded |
| C00595/8 | 4: WarningLocked | Response to start procedures with reference condition when the reference is not set |
| C00595/9 | 4: WarningLocked | Response to a non-supported positioning mode |
| C00595/10 | 4: WarningLocked | Response to implausible profile set data |
| C00595/11 | 5: Warning | Response to the selection of an invalid operating mode of the MCK |
| C00595/12 | 4: WarningLocked | Response to indicating an invalid profile data set |
| C00595/13 | 5: Warning | Response to an error of the FB L_MckCtrlInterface_1 |
| C00595/14 | 4: WarningLocked | Response to a profile start with a target position outside the software limit positions (C01229/1 and C01229/2). |
| C00595/15 | 1: Fault | MCK: Resp. to remote timeout |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00597

| Parameter \| Name: **C00597 \| Resp. to motor phase failure** | | Data type: UNSIGNED_8<br>Index: $23978_d$ = $5DAA_h$ |
|---|---|---|
| Response to motor phase failure<br>• If a phase current does not exceed the threshold set in C00599 for more than one period, the response set here will be triggered. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **No Reaction** | |
| 1 | Fault | |
| 5 | Warning | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00598**

| Parameter \| Name: **C00598 \| Resp. to open circuit AINx** | | Data type: UNSIGNED_8 Index: $23977_d$ = $5DA9_h$ |
|---|---|---|
| Configuration of monitoring of the analog inputs | | ▸ Analog terminals |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 5 | Warning | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00598/1 | 3: TroubleQuickStop | Response to open circuit at AIN1 if configured as 4 ... 20 mA current loop |
| C00598/2 | 3: TroubleQuickStop | Response to open circuit at AIN2 when being configured as 4 ... 20 mA-current loop |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00599**

| Parameter \| Name: **C00599 \| Motor phase failure threshold** | | | Data type: INTEGER_16 Index: $23976_d$ = $5DA8_h$ |
|---|---|---|---|
| Threshold for motor phase failure monitoring<br>• 100 % ≡ rated inverter current (C00098)<br>• If a phase current does not exceed the threshold set here for more than one period, the response to motor phase failure set in C00597 will be triggered. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1.00 | % | 100.00 | **5.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00600**

| Parameter \| Name: **C00600 \| Resp. to DC bus voltage** | | Data type: UNSIGNED_8 Index: $23975_d$ = $5DA7_h$ |
|---|---|---|
| Configuration of monitoring of the motor control (group 3) | | |
| **Selection list** | | |
| 1 | Fault | |
| 2 | Trouble | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00600/1 | 2: Trouble | Response to DC bus undervoltage |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00601**

| Parameter \| Name: **C00601 \| Del. resp. to fault: DC bus overvoltage** | | | Data type: UNSIGNED_16 Index: $23974_d$ = $5DA6_h$ |
|---|---|---|---|
| Error response delay times | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 65.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00601/1 | 2.000 s | | Delay time for triggering the "DC-bus overvoltage" error<br>• If a DC-bus overvoltage occurs, an error will not be triggered until the set delay time has elapsed. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00602**

| Parameter \| Name: **C00602 \| Resp. to earth fault** | | Data type: UNSIGNED_8 Index: $23973_d$ = $5DA5_h$ |
|---|---|---|
| Response to earth fault in the motor phase(s) | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| **1** | **Fault** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00603**

| Parameter \| Name: **C00603 \| Resp. to feedback** | | Data type: UNSIGNED_8 Index: $23972_d$ = $5DA4_h$ |
|---|---|---|
| Response of different monitoring modes for the encoder/feedback system | | ▶ Encoder/feedback system |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00603/1 | 1: Fault | Resp. to MultiEncoder open circuit |
| C00603/2 | 1: Fault | Resp. to resolver open circuit |
| C00603/3 | 1: Fault | Resp. to encoder comm. error |
| C00603/4 | 1: Fault | Resp. to encoder angular drift monit. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00604**

| Parameter \| Name: **C00604 \| Resp. to device overload (Ixt)** | | Data type: UNSIGNED_8 Index: $23971_d$ = $5DA3_h$ |
|---|---|---|
| Response if the adjustable device utilisation threshold (C00123) is reached.<br>• The current device utilisation is displayed in C00064. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| 1 | Fault | |
| **5** | **Warning** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00606**

| Parameter \| Name: **C00606 \| Resp. to motor overload (I²xt)** | | Data type: UNSIGNED_8 Index: $23969_d$ = $5DA1_h$ |
|---|---|---|
| Response if the adjustable motor overload threshold (C00120) is reached.<br>• The current thermal motor load is displayed in C00066. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | No Reaction | |
| 1 | Fault | |
| **5** | **Warning** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00607**

| Parameter \| Name: **C00607 \| Resp. to max freq. feedb. DIG12/67** | Data type: UNSIGNED_8 Index: $23968_d$ = $5DA0_h$ |
|---|---|
| Response when the maximum input frequency has been reached via the digital inputs. | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| 0 | No Reaction |
| **1** | **Fault** |
| 5 | Warning |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00608**

| Parameter \| Name: **C00608 \| Resp. to maximum torque** | Data type: UNSIGNED_8 Index: $23967_d$ = $5D9F_h$ |
|---|---|
| Response if the maximum torque (C00057) is reached. | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **No Reaction** |
| 1 | Fault |
| 5 | Warning |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00609**

| Parameter \| Name: **C00609 \| Resp. to maximum current** | Data type: UNSIGNED_8 Index: $23966_d$ = $5D9E_h$ |
|---|---|
| Response if the maximum current (C00022, C00023) is reached. | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **No Reaction** |
| 1 | Fault |
| 5 | Warning |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00610**

| Parameter \| Name: **C00610 \| 16-bit connection table** | Data type: UNSIGNED_16 Index: $23965_d$ = $5D9D_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00611**

| Parameter \| Name: **C00611 \| Bool connection table** | Data type: UNSIGNED_16 Index: $23964_d$ = $5D9C_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00612**

| Parameter \| Name: **C00612 \| 32-bit connection table** | Data type: UNSIGNED_16 Index: $23963_d$ = $5D9B_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00613**

| Parameter \| Name: **C00613 \| 16-bit connection table AdditionalFBsHL** | Data type: UNSIGNED_16 Index: $23962_d$ = $5D9A_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00615**

| Parameter \| Name: **C00615 \| Bool connection table AdditionalFBsHL** | Data type: UNSIGNED_16 Index: $23960_d$ = $5D98_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00617

| Parameter \| Name: C00617 \| 32-bit connection table AdditionalFBsHL | Data type: UNSIGNED_16 Index: $23958_d$ = $5D96_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00620

| Parameter \| Name: C00620 \| System connection list: 16-bit | Data type: UNSIGNED_16 Index: $23955_d$ = $5D93_h$ |
|---|---|

Connection parameters: 16-bit inputs
- Selection of the 16 bit output signals to be connected to the 16 bit input signals
- The selection list contains all 16 bit output signals which can be assigned to the 16 bit inputs displayed by the subcodes.

| Selection list |
|---|
| See selection list - analog signals |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00620/1 | 1003: LA_nCtrl_nMotorSpeedAct_a | LS_AnalogOutput: nOut1_a (V) |
| C00620/2 | 0: Not connected | LP_CanOut1: wState |
| C00620/3 | 0: Not connected | LP_CanOut1: wOut2 |
| C00620/4 | 0: Not connected | LP_CanOut1: wOut3 |
| C00620/5 | 0: Not connected | LP_CanOut1: wOut4 |
| C00620/6 | 0: Not connected | LP_CanOut2: wOut1 |
| C00620/7 | 0: Not connected | LP_CanOut2: wOut2 |
| C00620/8 | 0: Not connected | LP_CanOut2: wOut3 |
| C00620/9 | 0: Not connected | LP_CanOut2: wOut4 |
| C00620/10 | 0: Not connected | LP_CanOut3: wOut1 |
| C00620/11 | 0: Not connected | LP_CanOut3: wOut2 |
| C00620/12 | 0: Not connected | LP_CanOut3: wOut3 |
| C00620/13 | 0: Not connected | LP_CanOut3: wOut4 |
| C00620/14 | 0: Not connected | LS_DisFree_a: nDis1_a |
| C00620/15 | 0: Not connected | LS_DisFree_a: nDis2_a |
| C00620/16 | 0: Not connected | LS_DisFree_a: nDis3_a |
| C00620/17 | 0: Not connected | LS_DisFree_a: nDis4_a |
| C00620/18 | 0: Not connected | LS_DisFree: wDis1 |
| C00620/19 | 0: Not connected | LS_DisFree: wDis2 |
| C00620/20 | 0: Not connected | LS_DisFree: wDis3 |
| C00620/21 | 0: Not connected | LS_DisFree: wDis4 |
| C00620/22 | 0: Not connected | LP_MciOut: wState |
| C00620/23 | 0: Not connected | LP_MciOut: wOut2 |
| C00620/24 | 0: Not connected | LP_MciOut: wOut3 |
| C00620/25 | 0: Not connected | LP_MciOut: wOut4 |
| C00620/26 | 0: Not connected | LP_MciOut: wOut5 |
| C00620/27 | 0: Not connected | LP_MciOut: wOut6 |
| C00620/28 | 0: Not connected | LP_MciOut: wOut7 |
| C00620/29 | 0: Not connected | LP_MciOut: wOut8 |
| C00620/30 | 0: Not connected | LP_MciOut: wOut9 |
| C00620/31 | 0: Not connected | LP_MciOut: wOut10 |
| C00620/32 | 0: Not connected | LP_MciOut: wOut11 |
| C00620/33 | 0: Not connected | LP_MciOut: wOut12 |
| C00620/34 | 0: Not connected | LP_MciOut: wOut13 |

| Parameter \| Name:<br>**C00620 \| System connection list: 16-bit** | | Data type: UNSIGNED_16<br>Index: $23955_d$ = $5D93_h$ |
|---|---|---|
| C00620/35 | 0: Not connected | LP_MciOut: wOut14 |
| C00620/36 | 0: Not connected | LP_MciOut: wOut15 |
| C00620/37 | 0: Not connected | LP_MciOut: wOut16 |
| C00620/38 | 0: Not connected | LS_AnalogOutput: nOut2_a (V) |
| C00620/39 | 0: Not connected | LS_AnalogOutput: nOut1_a (I) |
| C00620/40 | 0: Not connected | LS_AnalogOutput: nOut2_a (I) |
| C00620/41 | 0: Not connected | LS_DisFree_a: nDis5_a |
| C00620/42 | 0: Not connected | LS_DisFree_a: nDis6_a |
| C00620/43 | 0: Not connected | LS_DisFree_a: nDis7_a |
| C00620/44 | 0: Not connected | LS_DisFree_a: nDis8_a |
| C00620/45 | 0: Not connected | LS_DisFree: wDis5 |
| C00620/46 | 0: Not connected | LS_DisFree: wDis6 |
| C00620/47 | 0: Not connected | LS_DisFree: wDis7 |
| C00620/48 | 0: Not connected | LS_DisFree: wDis8 |
| C00620/49 | 0: Not connected | LS_ParReadWrite_1: wParIndex |
| C00620/50 | 0: Not connected | LS_ParReadWrite_1: wParSubindex |
| C00620/51 | 0: Not connected | LS_ParReadWrite_1: wInHWord |
| C00620/52 | 0: Not connected | LS_ParReadWrite_1: wInLWord |
| C00620/53 | 0: Not connected | LS_ParReadWrite_2: wParIndex |
| C00620/54 | 0: Not connected | LS_ParReadWrite_2: wParSubindex |
| C00620/55 | 0: Not connected | LS_ParReadWrite_2: wInHWord |
| C00620/56 | 0: Not connected | LS_ParReadWrite_2: wInLWord |
| C00620/57 | 0: Not connected | LS_ParReadWrite_3: wParIndex |
| C00620/58 | 0: Not connected | LS_ParReadWrite_3: wParSubindex |
| C00620/59 | 0: Not connected | LS_ParReadWrite_3: wInHWord |
| C00620/60 | 0: Not connected | LS_ParReadWrite_3: wInLWord |
| C00620/61 | 0: Not connected | LS_ParReadWrite_4: wParIndex |
| C00620/62 | 0: Not connected | LS_ParReadWrite_4: wParSubindex |
| C00620/63 | 0: Not connected | LS_ParReadWrite_4: wInHWord |
| C00620/64 | 0: Not connected | LS_ParReadWrite_4: wInLWord |
| C00620/65 | 0: Not connected | LS_ParReadWrite_5: wParIndex |
| C00620/66 | 0: Not connected | LS_ParReadWrite_5: wParSubindex |
| C00620/67 | 0: Not connected | LS_ParReadWrite_5: wInHWord |
| C00620/68 | 0: Not connected | LS_ParReadWrite_5: wInLWord |
| C00620/69 | 0: Not connected | LS_ParReadWrite_6: wParIndex |
| C00620/70 | 0: Not connected | LS_ParReadWrite_6: wParSubindex |
| C00620/71 | 0: Not connected | LS_ParReadWrite_6: wInHWord |
| C00620/72 | 0: Not connected | LS_ParReadWrite_6: wInLWord |
| C00620/73 | 0: Not connected | Reserved |
| C00620/74 | 0: Not connected | Reserved |
| C00620/75 | 0: Not connected | Reserved |
| C00620/76 | 0: Not connected | Reserved |
| C00620/77 | 0: Not connected | Reserved |
| C00620/78 | 0: Not connected | Reserved |

| Parameter \| Name: C00620 \| System connection list: 16-bit | | Data type: UNSIGNED_16 Index: $23955_d$ = $5D93_h$ |
|---|---|---|
| C00620/79 | 0: Not connected | Reserved |
| C00620/80 | 0: Not connected | Reserved |
| C00620/81 | 0: Not connected | Reserved |
| C00620/82 | 0: Not connected | Reserved |
| C00620/83 | 0: Not connected | Reserved |
| C00620/84 | 0: Not connected | Reserved |
| C00620/85 | 0: Not connected | Reserved |
| C00620/86 | 0: Not connected | Reserved |
| C00620/87 | 0: Not connected | Reserved |
| C00620/88 | 0: Not connected | Reserved |
| C00620/89 | 0: Not connected | LS_RetainData: wIn1 |
| C00620/90 | 0: Not connected | LS_RetainData: wIn2 |
| C00620/91 | 0: Not connected | LS_RetainData: wIn3 |
| C00620/92 | 0: Not connected | LS_RetainData: wIn4 |
| C00620/93 | 0: Not connected | LS_AxisBusIn: wLine1 |
| C00620/94 | 0: Not connected | LS_AxisBusIn: wLine2 |
| C00620/95 | 0: Not connected | LS_AxisBusIn: wLine3 |
| C00620/96 | 0: Not connected | LS_AxisBusIn: wCas1 |
| C00620/97 | 0: Not connected | LS_AxisBusIn: wCas2 |
| C00620/98 | 0: Not connected | LS_AxisBusIn: wCas3 |
| C00620/99 | 0: Not connected | LS_AxisBusIn: wCas4 |
| C00620/100 | 0: Not connected | LS_AxisBusAux: wAuxOut1 |
| C00620/101 | 0: Not connected | LS_AxisBusAux: wAuxOut2 |
| C00620/102 | 0: Not connected | LS_AxisBusAux: wAuxOut3 |
| C00620/103 | 0: Not connected | LS_AxisBusAux: wAuxOut4 |
| C00620/104 | 0: Not connected | LS_AxisBusAux: wSelectSlave |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00621**

| Parameter \| Name: C00621 \| System connection list: Bool | | Data type: UNSIGNED_16 Index: $23954_d$ = $5D92_h$ |
|---|---|---|
| Connection parameters: Binary inputs<br>• Selection of the binary output signals to be connected to the binary input signals<br>• The selection list contains all binary output signals which can be assigned to the binary inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00621/1 | 1001: LA_nCtrl_bDriveFail | LS_DigitalOutput: bRelay |
| C00621/2 | 1000: LA_nCtrl_bDriveReady | LS_DigitalOutput: bOut1 |
| C00621/3 | 0: Not connected | LS_DigitalInput: bCountIn1_Reset |
| C00621/4 | 0: Not connected | LS_DigitalInput: bCountIn1_LoadStartValue |
| C00621/5 | 0: Not connected | LP_CanOut1: bState_B0 |
| C00621/6 | 0: Not connected | LP_CanOut1: bState_B1 |
| C00621/7 | 0: Not connected | LP_CanOut1: bState_B2 |
| C00621/8 | 0: Not connected | LP_CanOut1: bState_B3 |

| Parameter \| Name: C00621 \| System connection list: Bool | | Data type: UNSIGNED_16 Index: $23954_d$ = $5D92_h$ |
|---|---|---|
| C00621/9 | 0: Not connected | LP_CanOut1: bState_B4 |
| C00621/10 | 0: Not connected | LP_CanOut1: bState_B5 |
| C00621/11 | 0: Not connected | LP_CanOut1: bState_B6 |
| C00621/12 | 0: Not connected | LP_CanOut1: bState_B7 |
| C00621/13 | 0: Not connected | LP_CanOut1: bState_B8 |
| C00621/14 | 0: Not connected | LP_CanOut1: bState_B9 |
| C00621/15 | 0: Not connected | LP_CanOut1: bState_B10 |
| C00621/16 | 0: Not connected | LP_CanOut1: bState_B11 |
| C00621/17 | 0: Not connected | LP_CanOut1: bState_B12 |
| C00621/18 | 0: Not connected | LP_CanOut1: bState_B13 |
| C00621/19 | 0: Not connected | LP_CanOut1: bState_B14 |
| C00621/20 | 0: Not connected | LP_CanOut1: bState_B15 |
| C00621/21 | 0: Not connected | LS_DisFree_b: bDis1 |
| C00621/22 | 0: Not connected | LS_DisFree_b: bDis2 |
| C00621/23 | 0: Not connected | LS_DisFree_b: bDis3 |
| C00621/24 | 0: Not connected | LS_DisFree_b: bDis4 |
| C00621/25 | 0: Not connected | LS_DisFree_b: bDis5 |
| C00621/26 | 0: Not connected | LS_DisFree_b: bDis6 |
| C00621/27 | 0: Not connected | LS_DisFree_b: bDis7 |
| C00621/28 | 0: Not connected | LS_DisFree_b: bDis8 |
| C00621/29 | 0: Not connected | LP_CanOut2: bOut1_B0 |
| C00621/30 | 0: Not connected | LP_CanOut2: bOut1_B1 |
| C00621/31 | 0: Not connected | LP_CanOut2: bOut1_B2 |
| C00621/32 | 0: Not connected | LP_CanOut2: bOut1_B3 |
| C00621/33 | 0: Not connected | LP_CanOut2: bOut1_B4 |
| C00621/34 | 0: Not connected | LP_CanOut2: bOut1_B5 |
| C00621/35 | 0: Not connected | LP_CanOut2: bOut1_B6 |
| C00621/36 | 0: Not connected | LP_CanOut2: bOut1_B7 |
| C00621/37 | 0: Not connected | LP_CanOut2: bOut1_B8 |
| C00621/38 | 0: Not connected | LP_CanOut2: bOut1_B9 |
| C00621/39 | 0: Not connected | LP_CanOut2: bOut1_B10 |
| C00621/40 | 0: Not connected | LP_CanOut2: bOut1_B11 |
| C00621/41 | 0: Not connected | LP_CanOut2: bOut1_B12 |
| C00621/42 | 0: Not connected | LP_CanOut2: bOut1_B13 |
| C00621/43 | 0: Not connected | LP_CanOut2: bOut1_B14 |
| C00621/44 | 0: Not connected | LP_CanOut2: bOut1_B15 |
| C00621/45 | 0: Not connected | LP_CanOut3: bOut1_B0 |
| C00621/46 | 0: Not connected | LP_CanOut3: bOut1_B1 |
| C00621/47 | 0: Not connected | LP_CanOut3: bOut1_B2 |
| C00621/48 | 0: Not connected | LP_CanOut3: bOut1_B3 |
| C00621/49 | 0: Not connected | LP_CanOut3: bOut1_B4 |
| C00621/50 | 0: Not connected | LP_CanOut3: bOut1_B5 |
| C00621/51 | 0: Not connected | LP_CanOut3: bOut1_B6 |
| C00621/52 | 0: Not connected | LP_CanOut3: bOut1_B7 |

| Parameter \| Name: C00621 \| System connection list: Bool | | Data type: UNSIGNED_16 Index: $23954_d$ = $5D92_h$ |
|---|---|---|
| C00621/53 | 0: Not connected | LP_CanOut3: bOut1_B8 |
| C00621/54 | 0: Not connected | LP_CanOut3: bOut1_B9 |
| C00621/55 | 0: Not connected | LP_CanOut3: bOut1_B10 |
| C00621/56 | 0: Not connected | LP_CanOut3: bOut1_B11 |
| C00621/57 | 0: Not connected | LP_CanOut3: bOut1_B12 |
| C00621/58 | 0: Not connected | LP_CanOut3: bOut1_B13 |
| C00621/59 | 0: Not connected | LP_CanOut3: bOut1_B14 |
| C00621/60 | 0: Not connected | LP_CanOut3: bOut1_B15 |
| C00621/61 | 0: Not connected | LP_MciOut: bState_B0 |
| C00621/62 | 0: Not connected | LP_MciOut: bState_B1 |
| C00621/63 | 0: Not connected | LP_MciOut: bState_B2 |
| C00621/64 | 0: Not connected | LP_MciOut: bState_B3 |
| C00621/65 | 0: Not connected | LP_MciOut: bState_B4 |
| C00621/66 | 0: Not connected | LP_MciOut: bState_B5 |
| C00621/67 | 0: Not connected | LP_MciOut: bState_B6 |
| C00621/68 | 0: Not connected | LP_MciOut: bState_B7 |
| C00621/69 | 0: Not connected | LP_MciOut: bState_B8 |
| C00621/70 | 0: Not connected | LP_MciOut: bState_B9 |
| C00621/71 | 0: Not connected | LP_MciOut: bState_B10 |
| C00621/72 | 0: Not connected | LP_MciOut: bState_B11 |
| C00621/73 | 0: Not connected | LP_MciOut: bState_B12 |
| C00621/74 | 0: Not connected | LP_MciOut: bState_B13 |
| C00621/75 | 0: Not connected | LP_MciOut: bState_B14 |
| C00621/76 | 0: Not connected | LP_MciOut: bState_B15 |
| C00621/77 | 0: Not connected | LP_MciOut: bOut2_B0 |
| C00621/78 | 0: Not connected | LP_MciOut: bOut2_B1 |
| C00621/79 | 0: Not connected | LP_MciOut: bOut2_B2 |
| C00621/80 | 0: Not connected | LP_MciOut: bOut2_B3 |
| C00621/81 | 0: Not connected | LP_MciOut: bOut2_B4 |
| C00621/82 | 0: Not connected | LP_MciOut: bOut2_B5 |
| C00621/83 | 0: Not connected | LP_MciOut: bOut2_B6 |
| C00621/84 | 0: Not connected | LP_MciOut: bOut2_B7 |
| C00621/85 | 0: Not connected | LP_MciOut: bOut2_B8 |
| C00621/86 | 0: Not connected | LP_MciOut: bOut2_B9 |
| C00621/87 | 0: Not connected | LP_MciOut: bOut2_B10 |
| C00621/88 | 0: Not connected | LP_MciOut: bOut2_B11 |
| C00621/89 | 0: Not connected | LP_MciOut: bOut2_B12 |
| C00621/90 | 0: Not connected | LP_MciOut: bOut2_B13 |
| C00621/91 | 0: Not connected | LP_MciOut: bOut2_B14 |
| C00621/92 | 0: Not connected | LP_MciOut: bOut2_B15 |
| C00621/93 | 0: Not connected | LS_SetError_1: bSetError1 |
| C00621/94 | 0: Not connected | LS_SetError_1: bSetError2 |
| C00621/95 | 0: Not connected | LS_SetError_1: bSetError3 |
| C00621/96 | 0: Not connected | LS_SetError_1: bSetError4 |

| Parameter \| Name: **C00621 \| System connection list: Bool** | | Data type: UNSIGNED_16 Index: $23954_d$ = $5D92_h$ |
|---|---|---|
| C00621/97 | 0: Not connected | LS_DigitalInput: bCountIn6_Reset |
| C00621/98 | 0: Not connected | LS_DigitalInput: bCountIn6_LoadStartValue |
| C00621/99 | 0: Not connected | LS_DigitalOutput: bOut2 |
| C00621/100 | 0: Not connected | LS_DigitalOutput: bOut3 |
| C00621/101 | 0: Not connected | LS_DigitalOutput: bOut HighCurrent |
| C00621/102 | 0: Not connected | LS_DisFree_b: bDis9 |
| C00621/103 | 0: Not connected | LS_DisFree_b: bDis10 |
| C00621/104 | 0: Not connected | LS_DisFree_b: bDis11 |
| C00621/105 | 0: Not connected | LS_DisFree_b: bDis12 |
| C00621/106 | 0: Not connected | LS_DisFree_b: bDis13 |
| C00621/107 | 0: Not connected | LS_DisFree_b: bDis14 |
| C00621/108 | 0: Not connected | LS_DisFree_b: bDis15 |
| C00621/109 | 0: Not connected | LS_DisFree_b: bDis16 |
| C00621/110 | 0: Not connected | Reserved |
| C00621/111 | 0: Not connected | LS_ParReadWrite_1: bExecute |
| C00621/112 | 0: Not connected | LS_ParReadWrite_1: bReadWrite |
| C00621/113 | 0: Not connected | LS_ParReadWrite_2: bExecute |
| C00621/114 | 0: Not connected | LS_ParReadWrite_2: bReadWrite |
| C00621/115 | 0: Not connected | LS_ParReadWrite_3: bExecute |
| C00621/116 | 0: Not connected | LS_ParReadWrite_3: bReadWrite |
| C00621/117 | 0: Not connected | LS_ParReadWrite_4: bExecute |
| C00621/118 | 0: Not connected | LS_ParReadWrite_4: bReadWrite |
| C00621/119 | 0: Not connected | LS_ParReadWrite_5: bExecute |
| C00621/120 | 0: Not connected | LS_ParReadWrite_5: bReadWrite |
| C00621/121 | 0: Not connected | LS_ParReadWrite_6: bExecute |
| C00621/122 | 0: Not connected | LS_ParReadWrite_6: bReadWrite |
| C00621/123 | 0: Not connected | LS_WriteParamList: bExecute |
| C00621/124 | 0: Not connected | LS_WriteParamList: bSelectWriteValue_1 |
| C00621/125 | 0: Not connected | LS_WriteParamList: bSelectWriteValue_2 |
| C00621/126 | 0: Not connected | LS_CANManagement: bResetNode |
| C00621/127 | 0: Not connected | LS_CANManagement: bReInitCAN |
| C00621/128 | 0: Not connected | LS_DigitalInput: bPosIn12_Load |
| C00621/129 | 0: Not connected | Reserved |
| C00621/130 | 0: Not connected | Reserved |
| C00621/131 | 0: Not connected | Reserved |
| C00621/132 | 0: Not connected | Reserved |
| C00621/133 | 0: Not connected | Reserved |
| C00621/134 | 0: Not connected | Reserved |
| C00621/135 | 0: Not connected | Reserved |
| C00621/136 | 0: Not connected | Reserved |
| C00621/137 | 0: Not connected | Reserved |
| C00621/138 | 0: Not connected | Reserved |
| C00621/139 | 0: Not connected | Reserved |
| C00621/140 | 0: Not connected | Reserved |

| Parameter \| Name: C00621 \| System connection list: Bool | | Data type: UNSIGNED_16 Index: $23954_d$ = $5D92_h$ |
|---|---|---|
| C00621/141 | 0: Not connected | Reserved |
| C00621/142 | 0: Not connected | Reserved |
| C00621/143 | 0: Not connected | Reserved |
| C00621/144 | 0: Not connected | Reserved |
| C00621/145 | 0: Not connected | Reserved |
| C00621/146 | 0: Not connected | Reserved |
| C00621/147 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn3 |
| C00621/148 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn4 |
| C00621/149 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn5 |
| C00621/150 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn6 |
| C00621/151 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn7 |
| C00621/152 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn3_Rising |
| C00621/153 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn3_Falling |
| C00621/154 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn4_Rising |
| C00621/155 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn4_Falling |
| C00621/156 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn5_Rising |
| C00621/157 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn5_Falling |
| C00621/158 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn3Window |
| C00621/159 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn4Window |
| C00621/160 | 0: Not connected | LS_TouchProbe: bDisableTPDigIn5Window |
| C00621/161 | 0: Not connected | LS_AxisBusIO: bSetFail |
| C00621/162 | 0: Not connected | LS_AxisBusIO: bResetFail |
| C00621/163 | 0: Not connected | Reserved |
| C00621/164 | 0: Not connected | LS_MultiEncoder: bSetRef |
| C00621/165 | 0: Not connected | LS_RetainData: bSetRetain_1 |
| C00621/166 | 0: Not connected | LS_RetainData: bSetRetain_2 |
| C00621/167 | 0: Not connected | LS_RetainData: bSetRetain_3 |
| C00621/168 | 0: Not connected | LS_RetainData: bLoadParams |
| C00621/169 | 0: Not connected | LS_RetainData: bIn1 |
| C00621/170 | 0: Not connected | LS_RetainData: bIn2 |
| C00621/171 | 0: Not connected | LS_RetainData: bIn3 |
| C00621/172 | 0: Not connected | LS_RetainData: bIn4 |
| C00621/173 | 0: Not connected | LS_AxisBusAux: bReadWrite |
| C00621/174 | 0: Not connected | LS_AxisBusAux: bExecute |
| C00621/175 | 0: Not connected | LS_AxisBusAux: bStop |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00622**

| Parameter \| Name: **C00622 \| System connection list: Angle** | | Data type: UNSIGNED_16 Index: $23953_d$ = $5D91_h$ |
|---|---|---|
| Connection parameters: 32-bit inputs<br>• Selection of the 32-bit output signals for connection with the 32-bit input signals.<br>• The selection list contains all 32-bit output signals which can be assigned to the 32-bit inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - angle signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00622/1 | 0: Not connected | LS_DisFree_p: dnDis1_p |
| C00622/2 | 0: Not connected | LS_DisFree_p: dnDis2_p |
| C00622/3 | 0: Not connected | LS_DisFree_p: dnDis3_p |
| C00622/4 | 0: Not connected | LS_DisFree_p: dnDis4_p |
| C00622/5 | 0: Not connected | LS_DisFree_p: dnDis5_p |
| C00622/6 | 0: Not connected | LS_DisFree_p: dnDis6_p |
| C00622/7 | 0: Not connected | LS_DisFree_p: dnDis7_p |
| C00622/8 | 0: Not connected | LS_DisFree_p: dnDis8_p |
| C00622/9 | 0: Not connected | LP_CanOut1: dnOut34_p |
| C00622/10 | 0: Not connected | LP_CanOut2: dnOut34_p |
| C00622/11 | 0: Not connected | LP_CanOut3: dnOut34_p |
| C00622/12 | 0: Not connected | LP_MciOut: dnOut34_p |
| C00622/13 | 0: Not connected | LS_DigitalInput: dnPosIn12_Set_p |
| C00622/14 | 0: Not connected | Reserved |
| C00622/15 | 0: Not connected | Reserved |
| C00622/16 | 0: Not connected | LS_RetainData: dnIn1 |
| C00622/17 | 0: Not connected | LS_RetainData: dnIn2 |
| C00622/18 | 0: Not connected | LS_RetainData: dnIn3 |
| C00622/19 | 0: Not connected | LS_RetainData: dnIn4 |
| C00622/20 | 0: Not connected | LS_AxisBusOut: dnLine12 |
| C00622/21 | 0: Not connected | LS_AxisBusOut: dnCas12 |
| C00622/22 | 0: Not connected | LS_AxisBusOut: dnCas34 |
| C00622/23 | 0: Not connected | LS_AxisBusAux: dnAuxOut12 |
| C00622/24 | 0: Not connected | LS_AxisBusAux: dnAuxOut34 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00630**

| Parameter \| Name: **C00630 \| L_Limit 1-2: Min/Max** | | | Data type: INTEGER_16 Index: $23945_d$ = $5D89_h$ |
|---|---|---|---|
| Setting the limits | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00630/1 | -199.99 % | | L_Limit_1: Min.Limit |
| C00630/2 | 199.99 % | | L_Limit_1: Max.Limit |
| C00630/3 | -199.99 % | | L_Limit_2: Min.Limit |
| C00630/4 | 199.99 % | | L_Limit_2: Max.Limit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00631**

| Parameter \| Name: **C00631 \| L_LimitPhi 1-3: Min/Max** | | | Data type: INTEGER_32 Index: $23944_d$ = $5D88_h$ |
|---|---|---|---|
| Setting the limits | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00631/1 | -2147483647 incr. | | L_LimitPhi_1: Min.Limit |
| C00631/2 | 2147483647 incr. | | L_LimitPhi_1: Max.Limit |
| C00631/3 | -2147483647 incr. | | L_LimitPhi_2: Min.Limit |
| C00631/4 | 2147483647 incr. | | L_LimitPhi_2: Max.Limit |
| C00631/5 | -2147483647 incr. | | L_LimitPhi_3: Min.Limit |
| C00631/6 | 2147483647 incr. | | L_LimitPhi_3: Max.Limit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00632**

| Parameter \| Name: **C00632 \| L_NSet_1: Max.SkipFrq.** | | | Data type: INTEGER_16 Index: $23943_d$ = $5D87_h$ |
|---|---|---|---|
| Maximum limit values for the speed blocking zones<br>• Selection of the maximum limit values for the blocking zones in which the speed must not be constant. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00632/1 | 0.00 % | | L_NSet_1: Max. skip freq.1 |
| C00632/2 | 0.00 % | | L_NSet_1: Max. skip freq.2 |
| C00632/3 | 0.00 % | | L_NSet_1: Max. skip freq.3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00633**

| Parameter \| Name: **C00633 \| L_NSet_1: Min.SkipFrq.** | | | Data type: INTEGER_16 Index: $23942_d$ = $5D86_h$ |
|---|---|---|---|
| Minimum limit values for the speed blocking zones<br>• Selection of the minimum limit values for the blocking zones in which the speed must not be constant. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00633/1 | 0.00 % | | L_NSet_1: Min. skip freq.1 |
| C00633/2 | 0.00 % | | L_NSet_1: Min. skip freq.2 |
| C00633/3 | 0.00 % | | L_NSet_1: Min. skip freq.3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00634**

| Parameter \| Name: **C00634 \| L_NSet_1: wState** | | | Data type: UNSIGNED_16 Index: $23941_d$ = $5D85_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Bit coded status display | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | No blocking zone active | | "1" ≡ No blocking zone set for constant speeds |
| Bit 1 | Blocking zone 1 active | | "1" ≡ Suppression of constant speed characteristics within the limits of blocking zone 1 |
| Bit 2 | Blocking zone 2 active | | "1" ≡ Suppression of constant speed characteristics within the limits of blocking zone 2 |
| Bit 3 | Blocking zone 3 active | | "1" ≡ Suppression of constant speed characteristics within the limits of blocking zone 3 |
| Bit 4 | Jog in blocking zone | | "1" ≡ A ramp is used to keep the speed setpoint within a speed blocking zone |
| Bit 5 | MaxLimit active | | "1" ≡ Speed setpoint is at the maximum speed limit |
| Bit 6 | MinLimit active | | "1" ≡ Speed setpoint is at the minimum speed limit |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00635**

| Parameter \| Name: **C00635 \| L_NSet_1: nMaxLimit** | | | Data type: INTEGER_16 Index: $23940_d$ = $5D84_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Maximum speed setpoint for speed setpoint limitation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **199.99 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00636**

| Parameter \| Name: **C00636 \| L_NSet_1: nMinLimit** | | | Data type: INTEGER_16 Index: $23939_d$ = $5D83_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Minimum speed setpoint for speed setpoint limitation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **-199.99 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00637**

| Parameter \| Name: **C00637 \| L_NSet_1: Output blocking zones** | | | Data type: INTEGER_16 Index: $23938_d$ = $5D82_h$ |
|---|---|---|---|
| The L_NSet_1 FB: Speed setpoint is displayed after being processed by blocking zone function | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00638**

| Parameter \| Name: **C00638 \| L_NSet_1: Output ramp rounding** | | Data type: INTEGER_16 Index: $23937_d$ = $5D81_h$ |
|---|---|---|
| The L_NSet_1 FB: Speed setpoint is displayed after being processed by PT1 filter function | | |
| **Display range** (min. value \| unit \| max. value) | | |
| -199.99 | % | 199.99 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | |

**C00639**

| Parameter \| Name: **C00639 \| L_NSet_1: Output additional value** | | Data type: INTEGER_16 Index: $23936_d$ = $5D80_h$ |
|---|---|---|
| The L_NSet_1 FB: Additional speed setpoint is displayed after being processed by the ramp generator | | |
| **Display range** (min. value \| unit \| max. value) | | |
| -199.99 | % | 199.99 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | |

**C00640**

| Parameter \| Name: **C00640 \| L_NSet_1: nNOut_a** | | Data type: INTEGER_16 Index: $23935_d$ = $5D7F_h$ |
|---|---|---|
| The L_NSet_1 FB: Display of the generated main speed setpoint at the output *nNOut_a* | | |
| **Display range** (min. value \| unit \| max. value) | | |
| -199.99 | % | 199.99 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | |

**C00643**

| Parameter \| Name: **C00643 \| Resp. to PLI monitoring** | | Data type: UNSIGNED_8 Index: $23932_d$ = $5D7C_h$ |
|---|---|---|
| | | ▸ Pole position identification |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00643/1 | 1: Fault | Response to PLI monitoring |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00644**

| Parameter \| Name: **C00644 \| PLI traversing direction** | | Data type: UNSIGNED_8 Index: $23931_d$ = $5D7B_h$ |
|---|---|---|
| | | ▸ Pole position identification |
| **Selection list** | | |
| 0 | right rotating field | |
| 1 | left rotating field | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00644/1 | 0: right rotating field | PLI 360° traversing direction |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00645**

| Parameter \| Name: C00645 \| PLI max. permissible deflection | | Data type: INTEGER_16 Index: $23930_d$ = $5D7A_h$ |
|---|---|---|
| | | ▸ Pole position identification |
| **Setting range** (min. value \| unit \| max. value) | | |
| -6.0 | ° | 90.0 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00645/1 | 0.0 ° | Reserved |
| C00645/2 | 0.0 ° | PLI 360° max. error tolerance |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | |

**C00646**

| Parameter \| Name: C00646 \| PLI current amplitude | | Data type: UNSIGNED_16 Index: $23929_d$ = $5D79_h$ |
|---|---|---|
| | | ▸ Pole position identification |
| **Setting range** (min. value \| unit \| max. value) | | |
| 1 | % | 1000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00646/1 | 1 % | Reserved |
| C00646/2 | 100 % | PLI 360° current amplitude |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00647**

| Parameter \| Name: C00647 \| PLI ramp time | | Data type: UNSIGNED_16 Index: $23928_d$ = $5D78_h$ |
|---|---|---|
| | | ▸ Pole position identification |
| **Setting range** (min. value \| unit \| max. value) | | |
| 5 | % | 1000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00647/1 | 5 % | Reserved |
| C00647/2 | 100 % | PLI 360° ramp time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C00650**

| Parameter \| Name: C00650 \| L_Arithmetik 3-5: Function | | Data type: UNSIGNED_8 Index: $23925_d$ = $5D75_h$ |
|---|---|---|
| Selection of the internal arithmetics | | |
| **Selection list** | | |
| 0 | nOut_a = nIn1_a | |
| 1 | nOut_a = nIn1_a + nIn2_a | |
| 2 | nOut_a = nIn1_a - nIn2_a | |
| 3 | nOut_a = (nIn1_a * nIn2_a) / 100% | |
| 4 | nOut_a = (nIn1_a * 1%) / \|nIn2_a\| | |
| 5 | nOut_a = (nIn1_a * 100%) / (100% - nIn2_a) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00650/1 | 0: nOut_a = nIn1_a | L_Arithmetik_3: Function |
| C00650/2 | 0: nOut_a = nIn1_a | L_Arithmetik_4: Function |
| C00650/3 | 0: nOut_a = nIn1_a | L_Arithmetik_5: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00660

| Parameter \| Name: C00660 \| L_FixSet_a_1: Analog values | | | Data type: INTEGER_16 Index: $23915_d$ = $5D6B_h$ |
|---|---|---|---|
| FB L_FixSet_a_1: Setting of the fixed values | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00660/1 | 0.00 % | | Fixed value 0 ... 15 |
| C00660/... | | | |
| C00660/16 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00661

| Parameter \| Name: C00661 \| L_FixSet_w_1: Fixed values | | | Data type: UNSIGNED_16 Index: $23914_d$ = $5D6A_h$ |
|---|---|---|---|
| FB L_FixSet_w_1: Setting of the fixed values | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00661/1 | 0 | | Fixed value 0 ... 15 |
| C00661/... | | | |
| C00661/16 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00662

| Parameter \| Name: C00662 \| L_FixSet_w_2: Fixed values | | | Data type: UNSIGNED_16 Index: $23913_d$ = $5D69_h$ |
|---|---|---|---|
| FB L_FixSet_w_2: Setting of the fixed values | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00662/1 | 0 | | Fixed value 0 ... 15 |
| C00662/... | | | |
| C00662/16 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00670

| Parameter \| Name: C00670 \| L_OffsetGainP_1: Gain | | | Data type: INTEGER_32 Index: $23905_d$ = $5D61_h$ |
|---|---|---|---|
| The L_OffsetGainP_1 FB: Gain as multiplier of the input signal + offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -100.0000 | | 100.0000 | **1.0000** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

## C00671

| Parameter \| Name: C00671 \| L_OffsetGainP_2: Gain | | | Data type: INTEGER_32 Index: $23904_d$ = $5D60_h$ |
|---|---|---|---|
| The L_OffsetGainP_2 FB: Gain as multiplier of the input signal + offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -100.0000 | | 100.0000 | **1.0000** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C00672**

| Parameter \| Name: **C00672 \| L_OffsetGainP_3: Gain** | | | Data type: INTEGER_32 Index: $23903_d$ = $5D5F_h$ |
|---|---|---|---|
| The L_OffsetGainP_3 FB: Gain as multiplier of the input signal + offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -100.0000 | | 100.0000 | **1.0000** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C00673**

| Parameter \| Name: **C00673 \| L_OffsetGainPhiP 1-2: Offset** | | | Data type: INTEGER_32 Index: $23902_d$ = $5D5E_h$ |
|---|---|---|---|
| Angular offset (is added to the angular input signal) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00673/1 | 0 incr. | | L_OffsetGainPhiP_1: Offset |
| C00673/2 | 0 incr. | | L_OffsetGainPhiP_2: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00674**

| Parameter \| Name: **C00674 \| L_OffsetGainPhiP 1-2: Gain** | | | Data type: INTEGER_32 Index: $23901_d$ = $5D5D_h$ |
|---|---|---|---|
| Angular gain as multiplier of the input signal + angular offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00674/1 | 0 | | L_OffsetGainPhiP_1: Gain |
| C00674/2 | 0 | | L_OffsetGainPhiP_2: Gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00677**

| Parameter \| Name: **C00677 \| L_GainOffsetP 1-3: Parameter** | | | Data type: INTEGER_16 Index: $23898_d$ = $5D5A_h$ |
|---|---|---|---|
| Gain and offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00677/1 | 100.00 % | | L_GainOffsetP_1: Gain |
| C00677/2 | 0.00 % | | L_GainOffsetP_1: Offset |
| C00677/3 | 100.00 % | | L_GainOffsetP_2: Gain |
| C00677/4 | 0.00 % | | L_GainOffsetP_2: Offset |
| C00677/5 | 100.00 % | | L_GainOffsetP_3: Gain |
| C00677/6 | 0.00 % | | L_GainOffsetP_3: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00678

| Parameter \| Name: **C00678 \| L_GainOffsetPhiP 1-2: Parameter** | | | Data type: INTEGER_32 Index: $23897_d$ = $5D59_h$ |
|---|---|---|---|
| Gain and offset | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00678/1 | 65536 | | L_GainOffsetPhiP_1: Gain |
| C00678/2 | 0 | | L_GainOffsetPhiP_1: Offset |
| C00678/3 | 65536 | | L_GainOffsetPhiP_2: Gain |
| C00678/4 | 0 | | L_GainOffsetPhiP_2: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00679

| Parameter \| Name: **C00679 \| L_MulDiv_2: Parameter** | | | Data type: INTEGER_16 Index: $23896_d$ = $5D58_h$ |
|---|---|---|---|
| The L_MulDiv_2 FB: Numerator and denominator | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00679/1 | 0 | | L_MulDiv_2: Numerator |
| C00679/2 | 10000 | | L_MulDiv_2: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00680

| Parameter \| Name: **C00680 \| L_Compare_1: Fct.** | | Data type: UNSIGNED_8 Index: $23895_d$ = $5D57_h$ |
|---|---|---|
| The L_Compare_1 FB: Comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 1 | In1 = In2 | |
| 2 | In1 > In2 | |
| 3 | In1 < In2 | |
| 4 | \|In1\| = \|In2\| | |
| 5 | \|In1\| > \|In2\| | |
| **6** | **\|In1\| < \|In2\|** | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00681

| Parameter \| Name: **C00681 \| L_Compare_1: Hysteresis** | | | Data type: INTEGER_16 Index: $23894_d$ = $5D56_h$ |
|---|---|---|---|
| The L_Compare_1 FB: Hysteresis for the comparison function selected in C00680 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.50 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00682**

| Parameter \| Name: **C00682 \| L_Compare_1: Window** | | | Data type: INTEGER_16 Index: $23893_d$ = $5D55_h$ |
|---|---|---|---|
| The L_Compare_1 FB: Window for the comparison function selected in C00680 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **2.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00685**

| Parameter \| Name: **C00685 \| L_Compare_2: Fct.** | | Data type: UNSIGNED_8 Index: $23890_d$ = $5D52_h$ |
|---|---|---|
| The L_Compare_2 FB: Comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 1 | In1 = In2 | |
| 2 | In1 > In2 | |
| 3 | In1 < In2 | |
| **4** | **\|In1\| = \|In2\|** | |
| 5 | \|In1\| > \|In2\| | |
| 6 | \|In1\| < \|In2\| | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00686**

| Parameter \| Name: **C00686 \| L_Compare_2: Hysteresis** | | | Data type: INTEGER_16 Index: $23889_d$ = $5D51_h$ |
|---|---|---|---|
| The L_Compare_2 FB: Hysteresis for the comparison function selected in C00685 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.50 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00687**

| Parameter \| Name: **C00687 \| L_Compare_2: Window** | | | Data type: INTEGER_16 Index: $23888_d$ = $5D50_h$ |
|---|---|---|---|
| The L_Compare_2 FB: Window for the comparison function selected in C00685 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **2.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00690**

| Parameter \| Name: **C00690 \| L_Compare_3: Function** | | Data type: UNSIGNED_8 Index: $23885_d$ = $5D4D_h$ |
|---|---|---|
| The L_Compare_3 FB: Comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **1** | **In1 = In2** | |
| 2 | In1 > In2 | |
| 3 | In1 < In2 | |
| 4 | \|In1\| = \|In2\| | |
| 5 | \|In1\| > \|In2\| | |
| 6 | \|In1\| < \|In2\| | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00691**

| Parameter \| Name: **C00691 \| L_Compare_3: Hysteresis** | | | Data type: INTEGER_16 Index: $23884_d$ = $5D4C_h$ |
|---|---|---|---|
| FB L_Compare_3: Hysteresis for the comparison operation selected in C00690 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.00 %** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00692**

| Parameter \| Name: **C00692 \| L_Compare_3: Window** | | | Data type: INTEGER_16 Index: $23883_d$ = $5D4B_h$ |
|---|---|---|---|
| FB L_Compare_3: window for the comparison operation selected in C00690 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **0.00 %** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00693**

| Parameter \| Name: **C00693 \| L_Compare 4-5: Function** | | Data type: UNSIGNED_8 Index: $23882_d$ = $5D4A_h$ |
|---|---|---|
| Comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | | |
| **Selection list** | | |
| 1 | In1 = In2 | |
| 2 | In1 > In2 | |
| 3 | In1 < In2 | |
| 4 | \|In1\| = \|In2\| | |
| 5 | \|In1\| > \|In2\| | |
| 6 | \|In1\| < \|In2\| | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00693/1 | 1: In1 = In2 | L_Compare_4: Function |
| C00693/2 | 1: In1 = In2 | L_Compare_5: Function |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00694**

| Parameter \| Name: **C00694 \| L_Compare 4-5: Hysteresis** | | | Data type: INTEGER_16 Index: $23881_d$ = $5D49_h$ |
|---|---|---|---|
| Hysteresis for the comparison operation selected in C00693 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00694/1 | 0.00 % | | L_Compare_4: Hysteresis |
| C00694/2 | 0.00 % | | L_Compare_5: Hysteresis |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00695**

| Parameter \| Name: **C00695 \| L_Compare 4-5: Window** | | | Data type: INTEGER_16 Index: $23880_d$ = $5D48_h$ |
|---|---|---|---|
| Window for the comparison operation selected in C00693 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00695/1 | 0.00 % | | L_Compare_4: Window |
| C00695/2 | 0.00 % | | L_Compare_5: Window |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00696**

| Parameter \| Name: **C00696 \| L_OffsetGainP_1: Offset** | | | Data type: INTEGER_16 Index: $23879_d$ = $5D47_h$ |
|---|---|---|---|
| The L_OffsetGainP_1 FB: Offset (additive to the input signal) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00697**

| Parameter \| Name: **C00697 \| L_OffsetGainP_2: Offset** | | | Data type: INTEGER_16 Index: $23878_d$ = $5D46_h$ |
|---|---|---|---|
| The L_OffsetGainP_2 FB: Offset (additive to the input signal) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00698**

| Parameter \| Name: **C00698 \| L_OffsetGainP_3: Offset** | | | Data type: INTEGER_16 Index: $23877_d$ = $5D45_h$ |
|---|---|---|---|
| The L_OffsetGainP_3 FB: Offset (additive to the input signal) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00699**

| Parameter \| Name: **C00699 \| L_MulDiv_1: Parameter** | | | Data type: INTEGER_16 Index: $23876_d$ = $5D44_h$ |
|---|---|---|---|
| The L_MulDiv_1 FB: Numerator and denominator | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00699/1 | 0 | | L_MulDiv_1: Numerator |
| C00699/2 | 10000 | | L_MulDiv_1: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00700**

| Parameter \| Name: **C00700 \| LA_NCtrl: Analog connection list** | | Data type: UNSIGNED_16 Index: $23875_d$ = $5D43_h$ |
|---|---|---|
| Connection parameters for "Actuating drive speed" application: 16-bit inputs<br>• Selection of the 16 bit output signals to be connected to the 16 bit input signals<br>• The selection list contains all 16 bit output signals which can be assigned to the 16 bit inputs displayed by the subcodes. | | |
| **Selection list** | | |
| See selection list - analog signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00700/1 | 20005: LS_ParFix_wDriveCtrl | LA_NCtrl: wCANDriveControl<br>(input for control word from CAN to device control) |
| C00700/2 | 20005: LS_ParFix_wDriveCtrl | LA_NCtrl: wMCIDriveControl<br>(input for control word from communication interface to device control) |
| C00700/3 | 20012: LS_ParFree_nC472_3_a | LA_NCtrl: nTorqueMotLim_a<br>(input for maximum torque in motor mode) |
| C00700/4 | 20013: LS_ParFree_nC472_4_a | LA_NCtrl: nTorqueGenLim_a<br>(input for maximum torque in generator mode) |
| C00700/5 | 0: Not connected | LA_NCtrl: nPIDVpAdapt_a<br>(input for adapting the PID controller gain) |
| C00700/6 | 0: Not connected | LA_NCtrl: nPIDActValue_a<br>(input for actual PID controller value) |
| C00700/7 | 16000: AIN1_Out | LA_NCtrl: nMainSetValue_a<br>(input for main speed setpoint) |
| C00700/8 | 0: Not connected | LA_NCtrl: nAuxSetValue_a<br>(input for additional speed setpoint) |
| C00700/9 | 0: Not connected | LA_NCtrl: nGPAnalogSwitchIn1_a<br>(input for analog-value selector analog signal 1) |
| C00700/10 | 0: Not connected | LA_NCtrl: nGPAnalogSwitchIn2_a<br>(input for analog-value selector analog signal 2) |
| C00700/11 | 0: Not connected | LA_NCtrl: nGPArithmetikIn1_a<br>(input for arithmetic function analog signal 1) |
| C00700/12 | 0: Not connected | LA_NCtrl: nGPArithmetikIn2_a<br>(input for arithmetic function analog signal 2) |
| C00700/13 | 0: Not connected | LA_NCtrl: nGPMulDivIn_a<br>(input for analog signal for multiplication/division) |
| C00700/14 | 0: Not connected | LA_NCtrl: nGPCompareIn1_a<br>(input for comparison operation analog signal 1) |
| C00700/15 | 0: Not connected | LA_NCtrl: nGPCompareIn2_a<br>(input for comparison operation analog signal 2) |
| C00700/16 | 0: Not connected | LA_NCtrl: nVoltageAdd_a<br>(input for additive voltage boost) |
| C00700/17 | 0: Not connected | LA_NCtrl: nPIDInfluence_a<br>(input for influence signal of PID controller correcting variable) |
| C00700/18 | 0: Not connected | LA_NCtrl: nPIDSetValue_a<br>(input for PID controller setpoint) |
| C00700/19 | 0: Not connected | LA_NCtrl: nPWMAngleOffset<br>(input for pulse width modulation phase offset) |
| C00700/20 | 0: Not connected | LA_NCtrl: nBoost_a<br>(input for additional setpoint for the motor voltage at speed = 0) |

| Parameter \| Name: C00700 \| LA_NCtrl: Analog connection list | | Data type: UNSIGNED_16 Index: $23875_d$ = $5D43_h$ |
|---|---|---|
| C00700/21 | 0: Not connected | LA_NCtrl: wSMCtrl (interface to optional safety system) |
| C00700/22 | 0: Not connected | Reserved |
| C00700/23 | 0: Not connected | Reserved |
| C00700/24 | 0: Not connected | Reserved |
| C00700/25 | 0: Not connected | Reserved |
| C00700/26 | 0: Not connected | LA_NCtrl: wFreeIn1 (input for user signal 1) |
| C00700/27 | 0: Not connected | LA_NCtrl: wFreeIn2 (input for user signal 2) |
| C00700/28 | 0: Not connected | LA_NCtrl: wFreeIn3 (input for user signal 3) |
| C00700/29 | 0: Not connected | LA_NCtrl: wFreeIn4 (input for user signal 4) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00701**

| Parameter \| Name: C00701 \| LA_NCtrl: Digital connection list | | Data type: UNSIGNED_16 Index: $23874_d$ = $5D42_h$ |
|---|---|---|
| Connection parameters for "Actuating drive speed" application: Binary inputs<br>• Selection of the binary output signals to be connected to the binary input signals<br>• The selection list contains all binary output signals which can be assigned to the binary inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00701/1 | 0: Not connected | LA_NCtrl: bCInh (control input for setting controller inhibit) |
| C00701/2 | 16008: DigIn_CInh | LA_NCtrl: bFailReset (control input for error acknowledgement) |
| C00701/3 | 0: Not connected | LA_NCtrl: bSetQuickstop (control input for quick stop request) |
| C00701/4 | 16002: DigIn_bIn3 | LA_NCtrl: bSetDCBrake (control input for DC-injection braking request) |
| C00701/5 | 0: Not connected | LA_NCtrl: bRFG_Stop (control input for stopping the speed ramp function generator) |
| C00701/6 | 0: Not connected | LA_NCtrl: bRFG_0 (control input for setting the speed ramp function generator to 0) |
| C00701/7 | 0: Not connected | Reserved |
| C00701/8 | 16003: DigIn_bIn4 | LA_NCtrl: bSetSpeedCcw (control input for change of direction of rotation) |
| C00701/9 | 16000: DigIn_bIn1 | LA_NCtrl: bJogSpeed1 (selection input for fixed setpoints) |
| C00701/10 | 16001: DigIn_bIn2 | LA_NCtrl: bJogSpeed2 (selection input for fixed setpoints) |
| C00701/11 | 0: Not connected | LA_NCtrl: bJogSpeed4 (selection input for fixed setpoints) |
| C00701/12 | 0: Not connected | LA_NCtrl: bJogSpeed8 (selection input for fixed setpoints) |

| Parameter \| Name: C00701 \| LA_NCtrl: Digital connection list | | Data type: UNSIGNED_16 Index: $23874_d$ = $5D42_h$ |
|---|---|---|
| C00701/13 | 0: Not connected | LA_NCtrl: bJogRamp1 (selection input for additional acceleration/deceleration times) |
| C00701/14 | 0: Not connected | LA_NCtrl: bJogRamp2 (selection input for additional acceleration/deceleration times) |
| C00701/15 | 0: Not connected | LA_NCtrl: bJogRamp4 (selection input for additional acceleration/deceleration times) |
| C00701/16 | 0: Not connected | LA_NCtrl: bJogRamp8 (selection input for additional acceleration/deceleration times) |
| C00701/17 | 0: Not connected | LA_NCtrl: bMPOTInAct (control input for deactivation of motor potentiometer) |
| C00701/18 | 0: Not connected | LA_NCtrl: bMPOTUp (control input for motor potentiometer ramp-up) |
| C00701/19 | 0: Not connected | LA_NCtrl: bMPOTDown (control input for motor potentiometer ramp-down) |
| C00701/20 | 0: Not connected | LA_NCtrl: bMBRKRelease (control input for manual holding brake release request) |
| C00701/21 | 0: Not connected | LA_NCtrl: bGPFree1 (control input for manual jog in positive direction request) |
| C00701/22 | 0: Not connected | LA_NCtrl: bGPFree2 (control input for manual jog in negative direction request) |
| C00701/23 | 0: Not connected | LA_NCtrl: bGPAnalogSwitchSet (control input for analog-value selector change-over) |
| C00701/24 | 0: Not connected | LA_NCtrl: bGPDigitalDelayIn (input for digital signal with time delay) |
| C00701/25 | 0: Not connected | LA_NCtrl: bGPLogicIn1 (input signal 1 for digital logic) |
| C00701/26 | 0: Not connected | LA_NCtrl: bGPLogicIn2 (input signal 2 for digital logic) |
| C00701/27 | 0: Not connected | LA_NCtrl: bGPLogicIn3 (input signal 3 for digital logic) |
| C00701/28 | 0: Not connected | LA_NCtrl: bGPDFlipFlopInD (control input for DFlipFlop setting signal) |
| C00701/29 | 0: Not connected | LA_NCtrl: bGPDFlipFlopInClk (control input for DFlipFlop clock signal) |
| C00701/30 | 0: Not connected | LA_NCtrl: bGPDFlipFlopInClr (control input for DFlipFlop reset signal) |
| C00701/31 | 0: Not connected | LA_NCtrl: bMPotEnable (control input for activation of motor potentiometer) |
| C00701/32 | 0: Not connected | LA_NCtrl: bPIDEnableInfluenceRamp (control input for activation of influence of output correcting variable of PID controller) |
| C00701/33 | 0: Not connected | LA_NCtrl: bPIDIOff (control input for deactivation of PID controller I component) |
| C00701/34 | 20000: LS_ParFix_True | LA_NCtrl: bRLQCw (control input for activation of CW direction of rotation of speed setpoint) |

| Parameter \| Name: C00701 \| LA_NCtrl: Digital connection list | | Data type: UNSIGNED_16 Index: 23874d = 5D42h |
|---|---|---|
| C00701/35 | 0: Not connected | LA_NCtrl: bRLQCcw (control input for activation of CCW direction of rotation of speed setpoint) |
| C00701/36 | 0: Not connected | Reserved |
| C00701/37 | 0: Not connected | Reserved |
| C00701/38 | 0: Not connected | Reserved |
| C00701/39 | 0: Not connected | Reserved |
| C00701/40 | 0: Not connected | Reserved |
| C00701/41 | 0: Not connected | LA_NCtrl: bFreeIn1 (input for binary user signal 1) |
| C00701/42 | 0: Not connected | LA_NCtrl: bFreeIn2 (input for binary user signal 2) |
| C00701/43 | 0: Not connected | LA_NCtrl: bFreeIn3 (input for binary user signal 3) |
| C00701/44 | 0: Not connected | LA_NCtrl: bFreeIn4 (input for binary user signal 4) |
| C00701/45 | 0: Not connected | LA_NCtrl: bFreeIn5 (input for binary user signal 5) |
| C00701/46 | 0: Not connected | LA_NCtrl: bFreeIn6 (input for binary user signal 6) |
| C00701/47 | 0: Not connected | LA_NCtrl: bFreeIn7 (input for binary user signal 7) |
| C00701/48 | 0: Not connected | LA_NCtrl: bFreeIn8 (input for binary user signal 8) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00705**

| Parameter \| Name: C00705 \| LA_NCtrl_Out: Analog signal list | Data type: UNSIGNED_16 Index: 23870d = 5D3Eh |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00706**

| Parameter \| Name: C00706 \| LA_NCtrl_Out digital signal list | Data type: UNSIGNED_16 Index: 23869d = 5D3Dh |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00710**

| Parameter \| Name: C00710 \| LA_TabPos: Analog connection list | Data type: UNSIGNED_16 Index: 23865d = 5D39h |
|---|---|

Connection parameters for "Table positioning" application: 16-bit inputs
- Selection of the 16 bit output signals to be connected to the 16 bit input signals
- The selection list contains all 16 bit output signals which can be assigned to the 16 bit inputs displayed by the subcodes.

| Selection list | |
|---|---|
| See selection list - analog signals | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00710/1 | 0: Not connected | LA_TabPos: wCanDriveControl (input for control word from CAN to device control) |
| C00710/2 | 0: Not connected | LA_TabPos: wMciDriveControl (input for control word from communication interface to device control) |

| Parameter \| Name: **C00710 \| LA_TabPos: Analog connection list** | | Data type: UNSIGNED_16 Index: $23865_d$ = $5D39_h$ |
|---|---|---|
| C00710/3 | 0: Not connected | LA_TabPos: nTorqueMotLim_a (input for maximum torque in motor mode) |
| C00710/4 | 0: Not connected | LA_TabPos: nTorqueGenLim_a (input for maximum torque in generator mode) |
| C00710/5 | 0: Not connected | LA_TabPos: nMainSetValue_a (input for main speed setpoint) |
| C00710/6 | 0: Not connected | LA_TabPos: nAuxSetValue_a (input for additional speed setpoint) |
| C00710/7 | 0: Not connected | LA_TabPos: wMckCtrl1 (input for MCK control word 1) |
| C00710/8 | 0: Not connected | LA_TabPos: wMckCtrl2 (input for MCK control word 2) |
| C00710/9 | 0: Not connected | LA_TabPos: wMckOperationMode (input for selection of MCK operating mode) |
| C00710/10 | 0: Not connected | LA_TabPos: wPosProfileMode (input for selection of MCK positioning mode in positioning mode) |
| C00710/11 | 0: Not connected | LA_TabPos: wPosProfileNo (input for MCK positioning profile number in positioning mode) |
| C00710/12 | 0: Not connected | LA_TabPos: nGPAnalogSwitchIn1_a (input for analog-value selector analog signal 1) |
| C00710/13 | 0: Not connected | LA_TabPos: nGPAnalogSwitchIn2_a (input for analog-value selector analog signal 2) |
| C00710/14 | 0: Not connected | LA_TabPos: nGPArithmetikIn1_a (input for arithmetic function analog signal 1) |
| C00710/15 | 0: Not connected | LA_TabPos: nGPArithmetikIn2_a (input for arithmetic function analog signal 2) |
| C00710/16 | 0: Not connected | LA_TabPos: nGPMulDivIn_a (input for analog signal for multiplication/division) |
| C00710/17 | 0: Not connected | LA_TabPos: nGPCompareIn1_a (input for comparison operation analog signal 1) |
| C00710/18 | 0: Not connected | LA_TabPos: nGPCompareIn2_a (input for comparison operation analog signal 2) |
| C00710/19 | 0: Not connected | LA_TabPos: wGPCounter1LdVal (input for load value for counter module 1) |
| C00710/20 | 0: Not connected | LA_TabPos: wGPCounter1CmpVal (input for comparison value for counter module 1) |
| C00710/21 | 0: Not connected | LA_TabPos: nSpeedOverride_a (input for speed override) |
| C00710/22 | 0: Not connected | LA_TabPos: nAccOverride_a (input for acceleration override) |
| C00710/23 | 0: Not connected | LA_TabPos: wFreeIn1 (input for user signal 1) |
| C00710/24 | 0: Not connected | LA_TabPos: wFreeIn2 (input for user signal 2) |
| C00710/25 | 0: Not connected | LA_TabPos: wFreeIn3 (input for user signal 3) |
| C00710/26 | 0: Not connected | LA_TabPos: wFreeIn4 (input for user signal 4) |

| Parameter \| Name:<br>**C00710 \| LA_TabPos: Analog connection list** | | Data type: UNSIGNED_16<br>Index: $23865_d$ = $5D39_h$ |
|---|---|---|
| C00710/27 | 0: Not connected | LA_TabPos: nPosCtrlOutLimit<br>(input for correcting variable limitation of position controller) |
| C00710/28 | 0: Not connected | LA_TabPos: nPosCtrlPAdapt<br>(input for adapting the position controller gain) |
| C00710/29 | 0: Not connected | LA_TabPos: wSMCtrl<br>(interface to optional safety system) |
| C00710/30 | 0: Not connected | LA_TabPos: wPosProfileUnitsLW<br>(input for selecting the target position in [units], Low-Word) |
| C00710/31 | 0: Not connected | LA_TabPos: wPosProfileUnitsHW<br>(input for selecting the target position in [units], High-Word) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00711**

| Parameter \| Name:<br>**C00711 \| LA_TabPos: Digital connection list** | | Data type: UNSIGNED_16<br>Index: $23864_d$ = $5D38_h$ |
|---|---|---|
| Connection parameters for "Table positioning" application: Binary inputs<br>• Selection of the binary output signals to be connected to the binary input signals<br>• The selection list contains all binary output signals which can be assigned to the binary inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00711/1 | 0: Not connected | LA_TabPos: bCInh<br>(control input for setting controller inhibit) |
| C00711/2 | 0: Not connected | LA_TabPos: bFailReset<br>(control input for error acknowledgement) |
| C00711/3 | 0: Not connected | LA_TabPos: bSetQuickstop<br>(control input for quick stop request) |
| C00711/4 | 0: Not connected | LA_TabPos: bSetSpeedCcw<br>(control input for negation of speed direction) |
| C00711/5 | 0: Not connected | LA_TabPos: bJogSpeed1<br>(control input for fixed speed selection valency 1) |
| C00711/6 | 0: Not connected | LA_TabPos: bJogSpeed2<br>(control input for fixed speed selection valency 2) |
| C00711/7 | 0: Not connected | LA_TabPos: bMPotEnable<br>(control input for activation of motor potentiometer) |
| C00711/8 | 0: Not connected | LA_TabPos: bMPotUp<br>(control input for motor potentiometer ramp-up) |
| C00711/9 | 0: Not connected | LA_TabPos: bMPotDown<br>(control input for motor potentiometer ramp-down) |
| C00711/10 | 0: Not connected | LA_TabPos: bMBrakeRelease<br>(control input for manual holding brake release request) |
| C00711/11 | 0: Not connected | LA_TabPos: bPosCtrlOn<br>(control input for activation of position controller) |
| C00711/12 | 0: Not connected | LA_TabPos: bLimitSwitchPos<br>(input for positive hardware limit switch signal) |
| C00711/13 | 0: Not connected | LA_TabPos: bLimitSwitchNeg<br>(input for negative hardware limit switch signal) |

| Parameter \| Name: **C00711 \| LA_TabPos: Digital connection list** | | Data type: UNSIGNED_16 Index: $23864_d$ = $5D38_h$ |
|---|---|---|
| C00711/14 | 0: Not connected | LA_TabPos: bReleaseLimitSwitch (control input for hardware limit switch retracting request) |
| C00711/15 | 0: Not connected | LA_TabPos: bManJogPos (control input for manual jog in positive direction request) |
| C00711/16 | 0: Not connected | LA_TabPos: bManJogNeg (control input for manual jog in negative direction request) |
| C00711/17 | 0: Not connected | LA_TabPos: bManEnable2ndSpeed (control input for activation of the second manual speed for manual jog) |
| C00711/18 | 0: Not connected | LA_TabPos: bEnableSpeedOverride (control input for activation of speed override) |
| C00711/19 | 0: Not connected | LA_TabPos: bEnableAccOverride (control input for activation of acceleration override) |
| C00711/20 | 0: Not connected | LA_TabPos: bHomeStartStop (control input for start/stop homing) |
| C00711/21 | 0: Not connected | LA_TabPos: bHomeSetPosition (control input for setting the home position) |
| C00711/22 | 0: Not connected | LA_TabPos: bHomeResetPosition (control input for reset of "Home position known") |
| C00711/23 | 0: Not connected | LA_TabPos: bHomeMark (input for pre-stop signal for homing) |
| C00711/24 | 0: Not connected | LA_TabPos: bPosSetProfilePosition (control input for accepting the profile position in the profile data set) |
| C00711/25 | 0: Not connected | LA_TabPos: bPosSetActualPosition (control input for accepting the current position in the profile data set) |
| C00711/26 | 0: Not connected | LA_TabPos: bPosExecute (control input for positioning start) |
| C00711/27 | 0: Not connected | LA_TabPos: bPosFinishTarget (control input for completion of positioning to target position) |
| C00711/28 | 0: Not connected | LA_TabPos: bPosDisableFollowProfile (control input for suppression of sequence profile linkage) |
| C00711/29 | 0: Not connected | LA_TabPos: bPosStop (control input for interruption of profile generation by ramp-down procedure) |
| C00711/30 | 0: Not connected | LA_TabPos: bGPAnalogSwitchSet (control input for analog-value selector change-over) |
| C00711/31 | 0: Not connected | LA_TabPos: bGPDigitalDelayIn (input for digital signal with time delay) |
| C00711/32 | 0: Not connected | LA_TabPos: bGPLogicIn1 (input signal 1 for digital logic) |
| C00711/33 | 0: Not connected | LA_TabPos: bGPLogicIn2 (input signal 2 for digital logic) |
| C00711/34 | 0: Not connected | LA_TabPos: bGPLogicIn3 (input signal 3 for digital logic) |
| C00711/35 | 0: Not connected | LA_TabPos: bGPDFlipFlop_InD (control input for DFlipFlop setting signal) |

| Parameter \| Name: **C00711 \| LA_TabPos: Digital connection list** | | Data type: UNSIGNED_16 Index: $23864_d$ = $5D38_h$ |
|---|---|---|
| C00711/36 | 0: Not connected | LA_TabPos: bGPDFlipFlop_InClk (control input for DFlipFlop clock signal) |
| C00711/37 | 0: Not connected | LA_TabPos: bGPDFlipFlop_InClr (control input for DFlipFlop reset signal) |
| C00711/38 | 0: Not connected | LA_TabPos: bGPCounter1ClkUp (control input for up-counting counter module 1) |
| C00711/39 | 0: Not connected | LA_TabPos: bGPCounter1ClkDown (control input for down-counting counter module 1) |
| C00711/40 | 0: Not connected | LA_TabPos: bGPCounter1Load (control input for load value acceptance in counter module 1) |
| C00711/41 | 0: Not connected | LA_TabPos: bMckOperationMode_1 (control input for MCK operating mode changeover valency 1) |
| C00711/42 | 0: Not connected | LA_TabPos: bMckOperationMode_2 (control input for MCK operating mode changeover valency 2) |
| C00711/43 | 0: Not connected | LA_TabPos: bMckOperationMode_4 (control input for MCK operating mode changeover valency 4) |
| C00711/44 | 0: Not connected | LA_TabPos: bMckOperationMode_8 (control input for MCK operating mode changeover valency 8) |
| C00711/45 | 0: Not connected | LA_TabPos: bPosProfileNo_1 (control input for selection of profile number valency 1) |
| C00711/46 | 0: Not connected | LA_TabPos: bPosProfileNo_2 (control input for selection of profile number valency 2) |
| C00711/47 | 0: Not connected | LA_TabPos: bPosProfileNo_4 (control input for selection of profile number valency 4) |
| C00711/48 | 0: Not connected | LA_TabPos: bPosProfileNo_8 (control input for selection of profile number valency 8) |
| C00711/49 | 0: Not connected | LA_TabPos: bFreeIn1 (input for binary user signal 1) |
| C00711/50 | 0: Not connected | LA_TabPos: bFreeIn2 (input for binary user signal 2) |
| C00711/51 | 0: Not connected | LA_TabPos: bFreeIn3 (input for binary user signal 3) |
| C00711/52 | 0: Not connected | LA_TabPos: bFreeIn4 (input for binary user signal 4) |
| C00711/53 | 0: Not connected | LA_TabPos: bFreeIn5 (input for binary user signal 5) |
| C00711/54 | 0: Not connected | LA_TabPos: bFreeIn6 (input for binary user signal 6) |
| C00711/55 | 0: Not connected | LA_TabPos: bFreeIn7 (input for binary user signal 7) |
| C00711/56 | 0: Not connected | LA_TabPos: bFreeIn8 (input for binary user signal 8) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00712**

| Parameter \| Name: **C00712 \| LA_TabPos: phi connection list** | | Data type: UNSIGNED_16 Index: $23863_d$ = $5D37_h$ |
|---|---|---|
| Connection parameters for "Table positioning" application: 32-bit inputs<br>• Selection of the 32-bit output signals for connection with the 32-bit input signals<br>• The selection list contains all 32-bit output signals which can be assigned to the 32-bit inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - angle signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00712/1 | 0: Not connected | LA_TabPos: dnPosProfilePosition<br>(input for selecting the target position in [increments]) |
| C00712/2 | 0: Not connected | LA_TabPos: dnFreeIn1<br>(input for 32-bit user signal 1) |
| C00712/3 | 0: Not connected | LA_TabPos: dnFreeIn2<br>(input for 32-bit user signal 2) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00715**

| Parameter \| Name: **C00715 \| LA_TabPos_Out: Analog signal list** | Data type: UNSIGNED_16 Index: $23860_d$ = $5D34_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00716**

| Parameter \| Name: **C00716 \| LA_TabPos_Out: Digital signal list** | Data type: UNSIGNED_16 Index: $23859_d$ = $5D33_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00717**

| Parameter \| Name: **C00717 \| LA_TabPos_Out: phi signal list** | Data type: UNSIGNED_16 Index: $23858_d$ = $5D32_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00720**

| Parameter \| Name: **C00720 \| L_DigitalDelay_1: Delay** | | | Data type: UNSIGNED_32 Index: $23855_d$ = $5D2F_h$ |
|---|---|---|---|
| Switch-on/off delay time | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 3600.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00720/1 | 0.000 s | | L_DigitalDelay_1: ON delay |
| C00720/2 | 0.000 s | | L_DigitalDelay_1: OFF delay |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00721**

| Parameter \| Name: **C00721 \| L_DigitalDelay 2,3: Delay** | | | Data type: UNSIGNED_32 Index: $23854_d$ = $5D2E_h$ |
|---|---|---|---|
| Switch-on/off delay time | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 3600.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00721/1 | 0.000 s | | L_DigitalDelay_2: ON delay |
| C00721/2 | 0.000 s | | L_DigitalDelay_2: OFF delay |
| C00721/3 | 0.000 s | | L_DigitalDelay_3: ON delay |
| C00721/4 | 0.000 s | | L_DigitalDelay_3: OFF delay |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C00725**

Parameter | Name: **C00725 | Current switching frequency** — Data type: UNSIGNED_8, Index: $23850_d$ = $5D2A_h$

Display of the current switching frequency
- In C00018 you can choose between a drive-optimised setting for good smooth-running characteristics and an inverter loss-optimised setting (min. Pv).
- Both possibilities offer fixed and variable switching frequencies.
- When a variable switching frequency is selected in C00018, the switching frequency may change as a function of the load and rotational frequency.

| **Selection list** (read only) | |
|---|---|
| 1 | 4 kHz var./drive-optimised |
| 2 | 8 kHz var./drive-optimised |
| 3 | 16 kHz var./drive-optimised |
| 5 | 2 kHz constant/drive-optimised |
| 6 | 4 kHz constant/drive-optimised |
| 7 | 8 kHz constant/drive-optimised |
| 8 | 16 kHz constant/drive-optimised |
| 11 | 4 kHz var./min. Pv |
| 12 | 8 kHz var./min. Pv |
| 13 | 16 kHz var./min. Pv |
| 14 | Reserved |
| 15 | 2 kHz constant/min. Pv |
| 16 | 4 kHz constant/min. Pv |
| 17 | 8 kHz constant/min. Pv |
| 18 | 16 kHz constant/min. Pv |
| 21 | 8 kHz var./drive-opt./4 kHz min |
| 22 | 16 kHz var./drive-opt./4 kHz min |
| 23 | 16 kHz var./drive-opt./8 kHz min |
| 31 | 8 kHz var./min. Pv/4 kHz min |
| 32 | 16 kHz var./min. Pv/4 kHz min |
| 33 | 16 kHz var./min. Pv/8 kHz min |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00726**

Parameter | Name: **C00726 | Current limit values** — Data type: UNSIGNED_8, Index: $23849_d$ = $5D29_h$

**This code is for device-internal use only and must not be written to by the user!**

**C00727**

| Parameter \| Name: **C00727 \| LS_Keypad digital values** | | Data type: UNSIGNED_8 Index: $23848_d$ = $5D28_h$ |
|---|---|---|
| Execution of control commands for keypad operation | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | | 1 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00727/1 | 0 | "1" ≡ request quick stop |
| C00727/2 | 0 | "1" ≡ request DC-injection braking |
| C00727/3 | 0 | "1" ≡ request change of direction of rotation |
| C00727/4 | 0 | "1" ≡ request fixed speed setpoint 1 |
| C00727/5 | 0 | "1" ≡ request fixed speed setpoint 2 |
| C00727/6 | 0 | "1" ≡ motor potentiometer: request activation |
| C00727/7 | 0 | "1" ≡ motor potentiometer: request pos. acceleration |
| C00727/8 | 0 | "1" ≡ motor potentiometer: request neg. acceleration |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00728**

| Parameter \| Name: **C00728 \| Keypad analog values** | | Data type: INTEGER_16 Index: $23847_d$ = $5D27_h$ |
|---|---|---|
| Selection of different setpoints when operating via keypad | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -199.99 | % | 199.99 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00728/1 | 100.00 % | Torque limit in motor mode |
| C00728/2 | 100.00 % | Torque limit in generator mode |
| C00728/3 | 0.00 % | Setpoint speed |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | |

**C00729**

| Parameter \| Name: **C00729 \| Remote: Setpoint selection** | | Data type: INTEGER_16 Index: $23846_d$ = $5D26_h$ |
|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | |
| -199.99 | | 199.99 |
| **Subcodes** | **Lenze setting** | **Info** |
| C00729/1 | 0.00 | Remote: Setpoint keypad |
| C00729/2 | 0.00 | Remote: Setpoint PC |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | |

**C00730**

| Parameter \| Name: **C00730 \| Oscilloscope scanning interval** | Data type: UNSIGNED_32 Index: $23845_d$ = $5D25_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00731**

| Parameter \| Name: **C00731 \| Oscilloscope recording length** | Data type: UNSIGNED_32 Index: $23844_d$ = $5D24_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00732**

| Parameter \| Name: **C00732 \| Oscilloscope command** | Data type: UNSIGNED_8 Index: $23843_d$ = $5D23_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00734

| Parameter \| Name:<br>**C00734 \| Oscilloscope trigger channel selection** | Data type: UNSIGNED_16<br>Index: $23841_d$ = $5D21_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00735

| Parameter \| Name:<br>**C00735 \| Oscilloscope channel source type** | Data type: UNSIGNED_8<br>Index: $23840_d$ = $5D20_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00736

| Parameter \| Name:<br>**C00736 \| Oscilloscope data type/data width** | Data type: UNSIGNED_16<br>Index: $23839_d$ = $5D1F_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00740

| Parameter \| Name:<br>**C00740 \| Oscilloscope offset variables** | Data type: UNSIGNED_32<br>Index: $23835_d$ = $5D1B_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00741

| Parameter \| Name:<br>**C00741 \| Oscilloscope trigger mode** | Data type: UNSIGNED_8<br>Index: $23834_d$ = $5D1A_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00742

| Parameter \| Name:<br>**C00742 \| Oscilloscope trigger delay** | Data type: INTEGER_16<br>Index: $23833_d$ = $5D19_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00743

| Parameter \| Name:<br>**C00743 \| Oscilloscope trigger level** | Data type: UNSIGNED_32<br>Index: $23832_d$ = $5D18_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00744

| Parameter \| Name:<br>**C00744 \| Oscilloscope trigger mask** | Data type: UNSIGNED_32<br>Index: $23831_d$ = $5D17_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00746

| Parameter \| Name:<br>**C00746 \| Oscilloscope trigger counter** | Data type: UNSIGNED_32<br>Index: $23829_d$ = $5D15_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00747

| Parameter \| Name:<br>**C00747 \| Oscilloscope status word** | Data type: UNSIGNED_16<br>Index: $23828_d$ = $5D14_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00748

| Parameter \| Name:<br>**C00748 \| Oscilloscope no. of measured values** | Data type: UNSIGNED_32<br>Index: $23827_d$ = $5D13_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

C00749

| Parameter \| Name:<br>**C00749 \| Oscilloscope recording** | Data type: UNSIGNED_32<br>Index: $23826_d$ = $5D12_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00750**

| Parameter \| Name: **C00750 \| Select. of BU oscillos. channels** | Data type: UNSIGNED_8 Index: $23825_d$ = $5D11_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00751**

| Parameter \| Name: **C00751 \| Oscilloscope data memory** | Data type: UNSIGNED_32 Index: $23824_d$ = $5D10_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00753**

| Parameter \| Name: **C00753 \| Oscilloscope data memory octet string** | Data type: OCTET_STRING Index: $23822_d$ = $5D0E_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00760**

| Parameter \| Name: **C00760 \| LA_SwitchPos: Analog connection list** | Data type: UNSIGNED_16 Index: $23815_d$ = $5D07_h$ |
|---|---|

Connection parameters for "Switch-off positioning" application: 16-bi inputs
- Selection of the 16 bit output signals to be connected to the 16 bit input signals
- The selection list contains all 16 bit output signals which can be assigned to the 16 bit inputs displayed by the subcodes.

| **Selection list** | |
|---|---|
| See selection list - analog signals | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C00760/1 | 0: Not connected | LA_SwitchPos: wCANDriveControl (input for control word from CAN to device control) |
| C00760/2 | 0: Not connected | LA_SwitchPos: wMCIDriveControl (input for control word from communication interface to device control) |
| C00760/3 | 0: Not connected | LA_SwitchPos: nVoltageAdd_a (input for additive voltage impression) |
| C00760/4 | 0: Not connected | LA_SwitchPos: nBoost_a (input for additional setpoint for the motor voltage at speed = 0) |
| C00760/5 | 0: Not connected | LA_SwitchPos: nPWMAngleOffset (input for additional offset for the electrical angle of rotation) |
| C00760/6 | 0: Not connected | LA_SwitchPos: nTorqueMotLim_a (input for maximum torque in motor mode) |
| C00760/7 | 0: Not connected | LA_SwitchPos: nTorqueGenLim_a (input for maximum torque in generator mode) |
| C00760/8 | 0: Not connected | LA_SwitchPos: nMainSetValue_a (input for main speed setpoint) |
| C00760/9 | 0: Not connected | LA_SwitchPos: nAuxSetValue_a (input for additional speed setpoint) |
| C00760/10 | 0: Not connected | LA_SwitchPos: nGPAnalogSwitchIn1_a (input for analog-value selector analog signal 1) |
| C00760/11 | 0: Not connected | LA_SwitchPos: nGPAnalogSwitchIn2_a (input for analog-value selector analog signal 2) |
| C00760/12 | 0: Not connected | LA_SwitchPos: nGPArithmetikIn1_a (input for arithmetic function analog signal 1) |
| C00760/13 | 0: Not connected | LA_SwitchPos: nGPArithmetikIn2_a (input for arithmetic function analog signal 2) |
| C00760/14 | 0: Not connected | LA_SwitchPos: nGPMulDivIn_a (input for analog signal for multiplication/division) |

| Parameter \| Name: **C00760 \| LA_SwitchPos: Analog connection list** | | Data type: UNSIGNED_16 Index: $23815_d$ = $5D07_h$ |
|---|---|---|
| C00760/15 | 0: Not connected | LA_SwitchPos: nGPCompareIn1_a (input for comparison operation analog signal 1) |
| C00760/16 | 0: Not connected | LA_SwitchPos: nGPCompareIn2_a (input for comparison operation analog signal 2) |
| C00760/17 | 0: Not connected | LA_SwitchPos: wSMCtrl (interface to optional safety system) |
| C00760/18 | 0: Not connected | Reserved |
| C00760/19 | 0: Not connected | Reserved |
| C00760/20 | 0: Not connected | Reserved |
| C00760/21 | 0: Not connected | Reserved |
| C00760/22 | 0: Not connected | LA_SwitchPos: wFreeIn1 (input for user signal 1) |
| C00760/23 | 0: Not connected | LA_SwitchPos: wFreeIn2 (input for user signal 2) |
| C00760/24 | 0: Not connected | LA_SwitchPos: wFreeIn3 (input for user signal 3) |
| C00760/25 | 0: Not connected | LA_SwitchPos: wFreeIn4 (input for user signal 4) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00761**

| Parameter \| Name: **C00761 \| LA_SwitchPos: Digital connection list** | | Data type: UNSIGNED_16 Index: $23814_d$ = $5D06_h$ |
|---|---|---|
| Connection parameters for "Switch-off positioning" application: Binary inputs<br>• Selection of the binary output signals to be connected to the binary input signals<br>• The selection list contains all binary output signals which can be assigned to the binary inputs mapped by the subcodes. | | |
| **Selection list** | | |
| See selection list - digital signals | | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00761/1 | 0: Not connected | LA_SwitchPos: bCInh (control input for setting controller inhibit) |
| C00761/2 | 0: Not connected | LA_SwitchPos: bFailReset (control input for error acknowledgement) |
| C00761/3 | 0: Not connected | LA_SwitchPos: bSetQuickstop (control input for quick stop request) |
| C00761/4 | 0: Not connected | LA_SwitchPos: bSetDCBrake (control input for DC-injection braking request) |
| C00761/5 | 0: Not connected | LA_SwitchPos: bRFG_Stop (control input for stopping the speed ramp function generator) |
| C00761/6 | 0: Not connected | LA_SwitchPos: bSetSpeedCcw (control input for change of direction of rotation) |
| C00761/7 | 0: Not connected | LA_SwitchPos: bRLQCw (control input for activation of CW rotation (fail-safe)) |
| C00761/8 | 0: Not connected | LA_SwitchPos: bRLQCcw (control input for activation of CCW rotation (fail-safe)) |
| C00761/9 | 0: Not connected | LA_SwitchPos: bJogCtrlInputSel1 (selection input 1 for binary coded selection of the switch-off position 1 ... 3) |

| Parameter \| Name: C00761 \| LA_SwitchPos: Digital connection list | | Data type: UNSIGNED_16 Index: $23814_d$ = $5D06_h$ |
|---|---|---|
| C00761/10 | 0: Not connected | LA_SwitchPos: bJogCtrlInputSel2 (selection input 2 for binary coded selection of the switch-off position 1 ... 3) |
| C00761/11 | 0: Not connected | LA_SwitchPos: bJogCtrlRfgIn (control input for setpoint generator ramp-down) |
| C00761/12 | 0: Not connected | LA_SwitchPos: bJogCtrlJog1 (selection input 1 for overriding fixed setpoints (JOG setpoints) for the main setpoint) |
| C00761/13 | 0: Not connected | LA_SwitchPos: bJogCtrlJog2 (selection input 2 for overriding fixed setpoints (JOG setpoints) for the main setpoint) |
| C00761/14 | 0: Not connected | LA_SwitchPos: bJogCtrlSlowDown1 (control input for selection of pre-switch off 1) |
| C00761/15 | 0: Not connected | LA_SwitchPos: bJogCtrlStop1 (control input for stop function 1) |
| C00761/16 | 0: Not connected | LA_SwitchPos: bJogCtrlSlowDown2 (control input for selection of pre-switch off 2) |
| C00761/17 | 0: Not connected | LA_SwitchPos: bJogCtrlStop2 (control input for stop function 2) |
| C00761/18 | 0: Not connected | LA_SwitchPos: bJogCtrlSlowDown3 (control input for selection of pre-switch off 3) |
| C00761/19 | 0: Not connected | LA_SwitchPos: bJogCtrlStop3 (control input for stop function 3) |
| C00761/20 | 0: Not connected | LA_SwitchPos: bJogSpeed4 (selection input for fixed setpoints) |
| C00761/21 | 0: Not connected | LA_SwitchPos: bJogSpeed8 (selection input for fixed setpoints) |
| C00761/22 | 0: Not connected | LA_SwitchPos: bJogRamp1 (selection input for additional acceleration/deceleration times) |
| C00761/23 | 0: Not connected | LA_SwitchPos: bJogRamp2 (selection input for additional acceleration/deceleration times) |
| C00761/24 | 0: Not connected | LA_SwitchPos: bJogRamp4 (selection input for additional acceleration/deceleration times) |
| C00761/25 | 0: Not connected | LA_SwitchPos: bJogRamp8 (selection input for additional acceleration/deceleration times) |
| C00761/26 | 0: Not connected | LA_SwitchPos: bMBrkRelease (control input for manual holding brake release request) |
| C00761/27 | 0: Not connected | LA_SwitchPos: bGPAnalogSwitchSet (control input for analog-value selector change-over) |
| C00761/28 | 0: Not connected | LA_SwitchPos: bGPDigitalDelayIn (input for digital signal with time delay) |
| C00761/29 | 0: Not connected | LA_SwitchPos: bGPLogicIn1 (input signal 1 for digital logic) |
| C00761/30 | 0: Not connected | LA_SwitchPos: bGPLogicIn2 (input signal 2 for digital logic) |
| C00761/31 | 0: Not connected | LA_SwitchPos: bGPLogicIn3 (input signal 3 for digital logic) |
| C00761/32 | 0: Not connected | LA_SwitchPos: bGPDFlipFlop_InD (control input for DFlipFlop setting signal) |

| Parameter \| Name: **C00761 \| LA_SwitchPos: Digital connection list** | | Data type: UNSIGNED_16 Index: $23814_d$ = $5D06_h$ |
|---|---|---|
| C00761/33 | 0: Not connected | LA_SwitchPos: bGPDFlipFlop_InClk (control input for DFlipFlop clock signal) |
| C00761/34 | 0: Not connected | LA_SwitchPos: bGPDFlipFlop_InClr (control input for DFlipFlop reset signal) |
| C00761/35 | 0: Not connected | Reserved |
| C00761/36 | 0: Not connected | Reserved |
| C00761/37 | 0: Not connected | Reserved |
| C00761/38 | 0: Not connected | Reserved |
| C00761/39 | 0: Not connected | Reserved |
| C00761/40 | 0: Not connected | LA_SwitchPos: bFreeIn1 (input for binary user signal 1) |
| C00761/41 | 0: Not connected | LA_SwitchPos: bFreeIn2 (input for binary user signal 2) |
| C00761/42 | 0: Not connected | LA_SwitchPos: bFreeIn3 (input for binary user signal 3) |
| C00761/43 | 0: Not connected | LA_SwitchPos: bFreeIn4 (input for binary user signal 4) |
| C00761/44 | 0: Not connected | LA_SwitchPos: bFreeIn5 (input for binary user signal 5) |
| C00761/45 | 0: Not connected | LA_SwitchPos: bFreeIn6 (input for binary user signal 6) |
| C00761/46 | 0: Not connected | LA_SwitchPos: bFreeIn7 (input for binary user signal 7) |
| C00761/47 | 0: Not connected | LA_SwitchPos: bFreeIn8 (input for binary user signal 8) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00762**

| Parameter \| Name: **C00762 \| LA_SwitchPos: phi connection list** | Data type: UNSIGNED_16 Index: $23813_d$ = $5D05_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00765**

| Parameter \| Name: **C00765 \| LA_SwitchPos_Out: Analog signal list** | Data type: UNSIGNED_16 Index: $23810_d$ = $5D02_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00766**

| Parameter \| Name: **C00766 \| LA_SwitchPos_Out: Digital signal list** | Data type: UNSIGNED_16 Index: $23809_d$ = $5D01_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C00767**

| Parameter \| Name: **C00767 \| LA_SwitchPos_Out: phi signal list** | Data type: UNSIGNED_16 Index: $23808_d$ = $5D00_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

## C00800

| Parameter \| Name: **C00800 \| L_MPot_1: Upper limit** | | | Data type: INTEGER_16 Index: $23775_d$ = $5CDF_h$ |
|---|---|---|---|
| The L_MPot_1 FB: Upper limit of the motor potentiometer function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00801

| Parameter \| Name: **C00801 \| L_MPot_1: Lower limit** | | | Data type: INTEGER_16 Index: $23774_d$ = $5CDE_h$ |
|---|---|---|---|
| The L_MPot_1 FB: Lower limit of the motor potentiometer function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **-100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00802

| Parameter \| Name: **C00802 \| L_MPot_1: Acceleration time** | | | Data type: UNSIGNED_16 Index: $23773_d$ = $5CDD_h$ |
|---|---|---|---|
| The L_MPot_1 FB: Acceleration time of the motor potentiometer function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.1 | s | 6000.0 | **10.0 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

## C00803

| Parameter \| Name: **C00803 \| L_MPot_1: Deceleration time** | | | Data type: UNSIGNED_16 Index: $23772_d$ = $5CDC_h$ |
|---|---|---|---|
| The L_MPot_1 FB: Deceleration time of the motor potentiometer function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.1 | s | 6000.0 | **10.0 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

## C00804

| Parameter \| Name: **C00804 \| L_MPot_1: Inactive fct.** | | Data type: UNSIGNED_8 Index: $23771_d$ = $5CDB_h$ |
|---|---|---|
| The L_MPot_1 FB: Selection of the response if the motor potentiometer is deactivated via input *bInAct* | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | **Retain value** | Retain output value |
| 1 | Deceleration to 0 | Deceleration via ramp to 0 |
| 2 | Deceleration to lower limit | Deceleration via ramp to the lower limit (C00801) |
| 3 | Without ramp to 0 | Step change to 0 |
| 4 | Without ramp to lower limit | Jump to lower limit (C00800) |
| 5 | Acceleration to upper limit | Acceleration via ramp to upper limit (C00800) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00805**

| Parameter \| Name: **C00805 \| L_MPot_1: Init fct.** | | | Data type: UNSIGNED_8 Index: $23770_d$ = $5CDA_h$ |
|---|---|---|---|
| The L_MPot_1 FB: Selection of the response at device switch-on | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| **0** | **Load last value** | | |
| 1 | Load lower limit | | |
| 2 | Load 0 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00806**

| Parameter \| Name: **C00806 \| L_MPot_1: Use** | | Data type: UNSIGNED_8 Index: $23769_d$ = $5CD9_h$ |
|---|---|---|
| The L_MPot_1 FB: Use of the motor potentiometer | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **No** | The motor potentiometer is not used.<br>• The analog value applied to the *nIn_a* input is looped through without any changes to the *nOut_a* output. |
| 1 | Yes | The motor potentiometer is used.<br>• The analog value applied at the *nIn_a* input is led via the motor potentiometer and provided at the *nOut_a* output. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00807**

| Parameter \| Name: **C00807 \| L_NLim_1: Max.SkipFrq.** | | | Data type: INTEGER_16 Index: $23768_d$ = $5CD8_h$ |
|---|---|---|---|
| FB L_NLim_1: Maximum blocking frequencies | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00807/1 | 0.00 % | | Maximum blocking frequency for zone 1 |
| C00807/2 | 0.00 % | | Maximum blocking frequency for zone 2 |
| C00807/3 | 0.00 % | | Maximum blocking frequency for zone 3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00808**

| Parameter \| Name: **C00808 \| L_NLim_1: Min.SkipFrq.** | | | Data type: INTEGER_16 Index: $23767_d$ = $5CD7_h$ |
|---|---|---|---|
| FB L_NLim_1: Minimum blocking frequencies | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00808/1 | 0.00 % | | Minimum blocking frequency for zone 1 |
| C00808/2 | 0.00 % | | Minimum blocking frequency for zone 2 |
| C00808/3 | 0.00 % | | Minimum blocking frequency for zone 2 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00809

| Parameter \| Name: **C00809 \| L_NLim_2: Max.SkipFrq.** | | | Data type: INTEGER_16 Index: 23766$_d$ = 5CD6$_h$ |
|---|---|---|---|
| FB L_NLim_2: Maximum blocking frequencies | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00809/1 | 0.00 % | | Maximum blocking frequency for zone 1 |
| C00809/2 | 0.00 % | | Maximum blocking frequency for zone 2 |
| C00809/3 | 0.00 % | | Maximum blocking frequency for zone 3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00810

| Parameter \| Name: **C00810 \| L_NLim_2: Min.SkipFrq.** | | | Data type: INTEGER_16 Index: 23765$_d$ = 5CD5$_h$ |
|---|---|---|---|
| FB L_NLim_2: Minimum blocking frequencies | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00810/1 | 0.00 % | | Minimum blocking frequency for zone 1 |
| C00810/2 | 0.00 % | | Minimum blocking frequency for zone 2 |
| C00810/3 | 0.00 % | | Minimum blocking frequency for zone 3 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00811

| Parameter \| Name: **C00811 \| L_NLim_1: Current output value** | | | Data type: INTEGER_16 Index: 23764$_d$ = 5CD4$_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00811/1 | | | L_NLim_1: Current output value |
| C00811/2 | | | L_NLim_2: Current output value |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00812**

| Parameter \| Name: **C00812 \| L_NLim: Current status** | | | Data type: UNSIGNED_16 Index: $23763_d$ = $5CD3_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | No blocking zone active | | |
| Bit 1 | Blocking zone 1 active | | |
| Bit 2 | Blocking zone 2 active | | |
| Bit 3 | Blocking zone 3 active | | |
| Bit 4 | Value in blocking zone | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| **Subcodes** | | | **Info** |
| C00812/1 | | | L_NLim_1: Current status |
| C00812/2 | | | L_NLim_2: Current status |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

**C00820**

| Parameter \| Name: **C00820 \| L_DigitalLogic_1: Function** | | Data type: UNSIGNED_8 Index: $23755_d$ = $5CCB_h$ |
|---|---|---|
| The L_DigitalLogic_1 FB: Selection of the internal logic function | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **bOut = 0** | Constant value "FALSE" |
| 1 | bOut = 1 | Constant value "TRUE" |
| 2 | bOut = bIn1 AND bIn2 AND bIn3 | AND operation |
| 3 | bOut = bIn1 OR bIn2 OR bIn3 | OR operation |
| 4 | bOut = f (truth table) | The truth table parameterised in C00821 is used. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

### C00821

| Parameter \| Name: **C00821 \| L_DigitalLogic_1: Truth table** | | Data type: UNSIGNED_8 Index: $23754_d$ = $5CCA_h$ |
|---|---|---|
| The L_DigitalLogic_1 FB: Parameterisation of the truth table | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00821/1 | 0: FALSE | L_DigitalLogic_1: bIn3...bIn1=0 0 0 |
| C00821/2 | 0: FALSE | L_DigitalLogic_1: bIn3...bIn1=0 0 1 |
| C00821/3 | 0: FALSE | L_DigitalLogic_1: bIn3...bIn1=0 1 0 |
| C00821/4 | 0: FALSE | L_DigitalLogic_1: bIn3...bIn1=0 1 1 |
| C00821/5 | 0: FALSE | L_DigitalLogic_1: bIn3...bIn1=1 0 0 |
| C00821/6 | 0: FALSE | L_DigitalLogic_1: bIn3...bIn1=1 0 1 |
| C00821/7 | 0: FALSE | L_DigitalLogic_1: bIn3...bIn1=1 1 0 |
| C00821/8 | 0: FALSE | L_DigitalLogic_1: bIn3...bIn1=1 1 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### C00822

| Parameter \| Name: **C00822 \| L_DigitalLogic_2: Function** | | Data type: UNSIGNED_8 Index: $23753_d$ = $5CC9_h$ |
|---|---|---|
| The L_DigitalLogic_2 FB: Selection of the internal logic function | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **bOut = 0** | Constant value "FALSE" |
| 1 | bOut = 1 | Constant value "TRUE" |
| 2 | bOut = bIn1 AND ... bIn3 | AND operation |
| 3 | bOut = bIn1 OR ... bIn3 | OR operation |
| 4 | bOut = f (truth table) | The truth table parameterised in C00823 is used. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### C00823

| Parameter \| Name: **C00823 \| L_DigitalLogic_2: Truth table** | | Data type: UNSIGNED_8 Index: $23752_d$ = $5CC8_h$ |
|---|---|---|
| The L_DigitalLogic_2 FB: Parameterisation of the truth table | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00823/1 | 0: FALSE | L_DigitalLogic_2: bIn3...bIn1=0 0 0 |
| C00823/2 | 0: FALSE | L_DigitalLogic_2: bIn3...bIn1=0 0 1 |
| C00823/3 | 0: FALSE | L_DigitalLogic_2: bIn3...bIn1=0 1 0 |
| C00823/4 | 0: FALSE | L_DigitalLogic_2: bIn3...bIn1=0 1 1 |
| C00823/5 | 0: FALSE | L_DigitalLogic_2: bIn3...bIn1=1 0 0 |
| C00823/6 | 0: FALSE | L_DigitalLogic_2: bIn3...bIn1=1 0 1 |
| C00823/7 | 0: FALSE | L_DigitalLogic_2: bIn3...bIn1=1 1 0 |
| C00823/8 | 0: FALSE | L_DigitalLogic_2: bIn3...bIn1=1 1 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00824**

| Parameter \| Name: **C00824 \| L_DigitalLogic5_1: Function** | | Data type: UNSIGNED_8 Index: $23751_d$ = $5CC7_h$ |
|---|---|---|
| FB L_DigitalLogic5_1: Selection of the interna logic operation | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **bOut = 0** | |
| 1 | bOut = 1 | |
| 2 | bOut = f (truth table) | The truth table parameterised in C00825 is used. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00825**

| Parameter \| Name: **C00825 \| L_DigitalLogic5_1: Truth table** | | Data type: UNSIGNED_8 Index: $23750_d$ = $5CC6_h$ |
|---|---|---|
| FB L_DigitalLogic5_1: Parameter setting of the truth table | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00825/1 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 0 0 0 |
| C00825/2 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 0 0 1 |
| C00825/3 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 0 1 0 |
| C00825/4 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 0 1 1 |
| C00825/5 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 1 0 0 |
| C00825/6 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 1 0 1 |
| C00825/7 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 1 1 0 |
| C00825/8 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 0 1 1 1 |
| C00825/9 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 0 0 0 |
| C00825/10 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 0 0 1 |
| C00825/11 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 0 1 0 |
| C00825/12 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 0 1 1 |
| C00825/13 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 1 0 0 |
| C00825/14 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 1 0 1 |
| C00825/15 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 1 1 0 |
| C00825/16 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=0 1 1 1 1 |
| C00825/17 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 0 0 0 |
| C00825/18 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 0 0 1 |
| C00825/19 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 0 1 0 |
| C00825/20 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 0 1 1 |
| C00825/21 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 1 0 0 |
| C00825/22 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 1 0 1 |
| C00825/23 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 1 1 0 |
| C00825/24 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 0 1 1 1 |
| C00825/25 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 0 0 0 |
| C00825/26 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 0 0 1 |
| C00825/27 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 0 1 0 |
| C00825/28 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 0 1 1 |

| Parameter \| Name: C00825 \| L_DigitalLogic5_1: Truth table | | Data type: UNSIGNED_8 Index: $23750_d$ = $5CC6_h$ |
|---|---|---|
| C00825/29 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 1 0 0 |
| C00825/30 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 1 0 1 |
| C00825/31 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 1 1 0 |
| C00825/32 | 0: FALSE | L_DigitalLogic5_1: bIn5...bIn1=1 1 1 1 1 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00826

| Parameter \| Name: C00826 \| L_DigitalLogic5_2: Function | | Data type: UNSIGNED_8 Index: $23749_d$ = $5CC5_h$ |
|---|---|---|
| FB L_DigitalLogic5_2: Selection of the interna logic operation | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **bOut = 0** | |
| 1 | bOut = 1 | |
| 2 | bOut = f (truth table) | The truth table parameterised in C00827 is used. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00827

| Parameter \| Name: C00827 \| L_DigitalLogic5_2: Truth table | | Data type: UNSIGNED_8 Index: $23748_d$ = $5CC4_h$ |
|---|---|---|
| FB L_DigitalLogic5_2: Parameter setting of the truth table | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00827/1 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 0 0 0 |
| C00827/2 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 0 0 1 |
| C00827/3 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 0 1 0 |
| C00827/4 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 0 1 1 |
| C00827/5 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 1 0 0 |
| C00827/6 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 1 0 1 |
| C00827/7 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 1 1 0 |
| C00827/8 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 0 1 1 1 |
| C00827/9 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 0 0 0 |
| C00827/10 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 0 0 1 |
| C00827/11 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 0 1 0 |
| C00827/12 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 0 1 1 |
| C00827/13 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 1 0 0 |
| C00827/14 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 1 0 1 |
| C00827/15 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 1 1 0 |
| C00827/16 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=0 1 1 1 1 |
| C00827/17 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 0 0 0 |
| C00827/18 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 0 0 1 |
| C00827/19 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 0 1 0 |
| C00827/20 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 0 1 1 |
| C00827/21 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 1 0 0 |
| C00827/22 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 1 0 1 |

| Parameter \| Name: **C00827 \| L_DigitalLogic5_2: Truth table** | | Data type: UNSIGNED_8 Index: $23748_d$ = $5CC4_h$ |
|---|---|---|
| C00827/23 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 1 1 0 |
| C00827/24 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 0 1 1 1 |
| C00827/25 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 0 0 0 |
| C00827/26 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 0 0 1 |
| C00827/27 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 0 1 0 |
| C00827/28 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 0 1 1 |
| C00827/29 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 1 0 0 |
| C00827/30 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 1 0 1 |
| C00827/31 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 1 1 0 |
| C00827/32 | 0: FALSE | L_DigitalLogic5_2: bIn5...bIn1=1 1 1 1 1 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00828**

| Parameter \| Name: **C00828 \| L_DigitalLogic_3: Function** | | Data type: UNSIGNED_8 Index: $23747_d$ = $5CC3_h$ |
|---|---|---|

From version 02.00.00
The L_DigitalLogic_3 FB: Selection of the internal logic function

| **Selection list** (Lenze setting printed in bold) | | **Info** |
|---|---|---|
| **0** | **bOut = 0** | Constant value "FALSE" |
| 1 | bOut = 1 | Constant value "TRUE" |
| 2 | bOut = bIn1 AND bIn2 AND bIn3 | AND operation |
| 3 | bOut = bIn1 OR bIn2 OR bIn3 | OR operation |
| 4 | bOut = f (truth table) | The truth table parameterised in C00829 is used. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00829**

| Parameter \| Name: **C00829 \| L_DigitalLogic_3: Truth table** | | Data type: UNSIGNED_8 Index: $23746_d$ = $5CC2_h$ |
|---|---|---|

From version 02.00.00
The L_DigitalLogic_3 FB: Parameterisation of the truth table

| **Selection list** | | |
|---|---|---|
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00829/1 | 0: FALSE | L_DigitalLogic_3: bIn3...bIn1=0 0 0 |
| C00829/2 | 0: FALSE | L_DigitalLogic_3: bIn3...bIn1=0 0 1 |
| C00829/3 | 0: FALSE | L_DigitalLogic_3: bIn3...bIn1=0 1 0 |
| C00829/4 | 0: FALSE | L_DigitalLogic_3: bIn3...bIn1=0 1 1 |
| C00829/5 | 0: FALSE | L_DigitalLogic_3: bIn3...bIn1=1 0 0 |
| C00829/6 | 0: FALSE | L_DigitalLogic_3: bIn3...bIn1=1 0 1 |
| C00829/7 | 0: FALSE | L_DigitalLogic_3: bIn3...bIn1=1 1 0 |
| C00829/8 | 0: FALSE | L_DigitalLogic_3: bIn3...bIn1=1 1 1 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00830**

| Parameter \| Name: **C00830 \| 16-bit inputs [%]** | | | Data type: INTEGER_16 Index: $23745_d$ = $5CC1_h$ |
|---|---|---|---|
| Display in percent of 16-bit input values of different blocks | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00830/1 | | | L_Absolut_1: nIn_a |
| C00830/2 | | | L_AddSub_1: nIn1_a |
| C00830/3 | | | L_AddSub_1: nIn2_a |
| C00830/4 | | | L_AddSub_1: nIn3_a |
| C00830/5 | | | L_OffsetGain_1: nIn_a |
| C00830/6 | | | L_OffsetGain_1: nOffset_a |
| C00830/7 | | | L_OffsetGain_1: nGain_a |
| C00830/8 | | | L_Negation_1: nIn_a |
| C00830/9 | | | L_GainOffset_1: nIn_a |
| C00830/10 | | | L_GainOffset_1: nGain_a |
| C00830/11 | | | L_GainOffset_1: nOffset_a |
| C00830/12 | | | L_Arithmetik_1: nIn1_a |
| C00830/13 | | | L_Arithmetik_1: nIn2_a |
| C00830/14 | | | L_AnalogSwitch_1: nIn1_a |
| C00830/15 | | | L_AnalogSwitch_1: nIn2_a |
| C00830/16 | | | L_Compare_1: nIn1_a |
| C00830/17 | | | L_Compare_1: nIn2_a |
| C00830/18 | | | MCTRL: nTorqueLimitAdapt_a |
| C00830/19 | | | Reserved |
| C00830/20 | | | MCTRL: nPosCtrlPAdapt_a |
| C00830/21 | | | MCTRL: nPosCtrlOutLimit_a |
| C00830/22 | | | MCTRL: nSpeedSetValue_a |
| C00830/23 | | | MCTRL: nSpeedLowLimit_a |
| C00830/24 | | | MCTRL: nSpeedCtrlI_a |
| C00830/25 | | | MCTRL: nSpeedCtrlPAdapt_a |
| C00830/26 | | | MCTRL: nBoost_a |
| C00830/27 | | | MCTRL: nTorqueSetValue_a |
| C00830/28 | | | MCTRL: nTorqueGenLimit_a |
| C00830/29 | | | MCTRL: nTorqueMotLimit_a |
| C00830/30 | | | Reserved |
| C00830/31 | | | MCTRL: nVoltageAdd_a |
| C00830/32 | | | MCTRL: nPWMAngleOffset_a |
| C00830/33 | | | L_NSet_1: nCInhVal_a |
| C00830/34 | | | L_NSet_1: nNSet_a |
| C00830/35 | | | L_NSet_1: nSet_a |
| C00830/36 | | | L_NSet_1: nNAdd_a |
| C00830/37 | | | DCTRL: wCANControl |
| C00830/38 | | | DCTRL: wCCMControl |
| C00830/39 | | | L_NLim_1: nIn_a |
| C00830/40 | | | Reserved |

| Parameter \| Name: **C00830 \| 16-bit inputs [%]** | Data type: INTEGER_16 Index: $23745_d$ = $5CC1_h$ |
|---|---|
| C00830/41 | L_Compare_2: nIn1_a |
| C00830/42 | L_Compare_2: nIn2_a |
| C00830/43 | L_Compare_3: nIn1_a |
| C00830/44 | L_Compare_3: nIn2_a |
| C00830/45 | L_AnalogSwitch_2: nIn1_a |
| C00830/46 | L_AnalogSwitch_2: nIn2_a |
| C00830/47 | L_AnalogSwitch_3: nIn1_a |
| C00830/48 | L_AnalogSwitch_3: nIn2_a |
| C00830/49 | L_Arithmetik_2: nIn1_a |
| C00830/50 | L_Arithmetik_2: nIn2_a |
| C00830/51 | Reserved |
| C00830/52 | Reserved |
| C00830/53 | L_GainOffset_2: nIn_a |
| C00830/54 | L_GainOffset_2: nGain_a |
| C00830/55 | L_GainOffset_2: nOffset_a |
| C00830/56 | L_OffsetGainP_1: nIn_a |
| C00830/57 | L_OffsetGainP_2: nIn_a |
| C00830/58 | L_OffsetGain_2: nIn_a |
| C00830/59 | L_OffsetGain_2: nOffset_a |
| C00830/60 | L_OffsetGain_2: nGain_a |
| C00830/61 | L_PCTRL_1: nAct_a |
| C00830/62 | L_PCTRL_1: nAdapt_a |
| C00830/63 | L_PCTRL_1: nSet_a |
| C00830/64 | L_PCTRL_1: nInfluence_a |
| C00830/65 | MCK: nSpeedCtrlI_a |
| C00830/66 | MCK: nPWMAngleOffset_a |
| C00830/67 | Reserved |
| C00830/68 | MCK: nBrkAddValue_a |
| C00830/69 | MCK: nTorqueSetValue_a |
| C00830/70 | MCK: nTorqueLimitAdapt_a |
| C00830/71 | MCK: nSRampOverride_a |
| C00830/72 | MCK: nSpeedSetValue_a |
| C00830/73 | MCK: wMotionCtrl2 |
| C00830/74 | MCK: wMotionCtrl1 |
| C00830/75 | MCK: nSpeedOverride_a |
| C00830/76 | MCK: nAccOverride_a |
| C00830/77 | MCK: nSpeedAdd_v |
| C00830/78 | MCK: wAuxCtrl |
| C00830/79 | MCK: wSMCtrl |
| C00830/80 | L_OffsetGainP_3: nIn_a |
| C00830/81 | L_MPot_1: nIn_a |
| C00830/82 | L_MulDiv_1: nIn_a |
| C00830/83 | LS_DataAccess: wIn1 (Lenze-internal) |

| Parameter \| Name: C00830 \| 16-bit inputs [%] | Data type: INTEGER_16 Index: 23745$_d$ = 5CC1$_h$ |
|---|---|
| C00830/84 | LS_DataAccess: wIn2 (Lenze-internal) |
| C00830/85 | LS_DataAccess: wIn3 (Lenze-internal) |
| C00830/86 | LS_DataAccess: wIn4 (Lenze-internal) |
| C00830/87 | L_PT1_1: nIn_a |
| C00830/88 | MCTRL: nSpeedHighLimit_a |
| C00830/89 | L_PCTRL_1: nNSet_a |
| C00830/90 | L_PCTRL_1: nISet_a |
| C00830/91 | L_Interpolator_1: nPhdIn_v |
| C00830/92 | L_Interpolator_1: nNIn_a |
| C00830/93 | Reserved |
| C00830/94 | Reserved |
| C00830/95 | Reserved |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | |

**C00831**

| Parameter \| Name: C00831 \| 16-bit inputs | | | Data type: UNSIGNED_16 Index: 23744$_d$ = 5CC0$_h$ |
|---|---|---|---|
| Decimal/hexadecimal/bit coded display of 16 bit input values of various blocks | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00831/1 | | | L_Absolut_1: nIn_a |
| C00831/2 | | | L_AddSub_1: nIn1_a |
| C00831/3 | | | L_AddSub_1: nIn2_a |
| C00831/4 | | | L_AddSub_1: nIn3_a |
| C00831/5 | | | L_OffsetGain_1: nIn_a |
| C00831/6 | | | L_OffsetGain_1: nOffset_a |
| C00831/7 | | | L_OffsetGain_1: nGain_a |
| C00831/8 | | | L_Negation_1: nIn_a |
| C00831/9 | | | L_GainOffset_1: nIn_a |
| C00831/10 | | | L_GainOffset_1: nGain_a |
| C00831/11 | | | L_GainOffset_1: nOffset_a |
| C00831/12 | | | L_Arithmetik_1: nIn1_a |
| C00831/13 | | | L_Arithmetik_1: nIn2_a |
| C00831/14 | | | L_AnalogSwitch_1: nIn1_a |
| C00831/15 | | | L_AnalogSwitch_1: nIn2_a |
| C00831/16 | | | L_Compare_1: nIn1_a |
| C00831/17 | | | L_Compare_1: nIn2_a |
| C00831/18 | | | MCTRL: nTorqueLimitAdapt_a |

| Parameter \| Name: C00831 \| 16-bit inputs | Data type: UNSIGNED_16 Index: $23744_d$ = $5CC0_h$ |
|---|---|
| C00831/19 | Reserved |
| C00831/20 | MCTRL: nPosCtrlPAdapt_a |
| C00831/21 | MCTRL: nPosCtrlOutLimit_a |
| C00831/22 | MCTRL: nSpeedSetValue_a |
| C00831/23 | MCTRL: nSpeedLowLimit_a |
| C00831/24 | MCTRL: nSpeedCtrlI_a |
| C00831/25 | MCTRL: nSpeedCtrlPAdapt_a |
| C00831/26 | MCTRL: nBoost_a |
| C00831/27 | MCTRL: nTorqueSetValue_a |
| C00831/28 | MCTRL: nTorqueGenLimit_a |
| C00831/29 | MCTRL: nTorqueMotLimit_a |
| C00831/30 | Reserved |
| C00831/31 | MCTRL: nVoltageAdd_a |
| C00831/32 | MCTRL: nPWMAngleOffset_a |
| C00831/33 | L_NSet_1: nCInhVal_a |
| C00831/34 | L_NSet_1: nNSet_a |
| C00831/35 | L_NSet_1: nSet_a |
| C00831/36 | L_NSet_1: nNAdd_a |
| C00831/37 | DCTRL: wCANControl |
| C00831/38 | DCTRL: wMCIControl |
| C00831/39 | L_NLim_1: nIn_a |
| C00831/40 | Reserved |
| C00831/41 | L_Compare_2: nIn1_a |
| C00831/42 | L_Compare_2: nIn2_a |
| C00831/43 | L_Compare_3: nIn1_a |
| C00831/44 | L_Compare_3: nIn2_a |
| C00831/45 | L_AnalogSwitch_2: nIn1_a |
| C00831/46 | L_AnalogSwitch_2: nIn2_a |
| C00831/47 | L_AnalogSwitch_3: nIn1_a |
| C00831/48 | L_AnalogSwitch_3: nIn2_a |
| C00831/49 | L_Arithmetik_2: nIn1_a |
| C00831/50 | L_Arithmetik_2: nIn2_a |
| C00831/51 | Reserved |
| C00831/52 | Reserved |
| C00831/53 | L_GainOffset_2: nIn_a |
| C00831/54 | L_GainOffset_2: nGain_a |
| C00831/55 | L_GainOffset_2: nOffset_a |
| C00831/56 | L_OffsetGainP_1: nIn_a |
| C00831/57 | L_OffsetGainP_2: nIn_a |
| C00831/58 | L_OffsetGain_2: nIn_a |
| C00831/59 | L_OffsetGain_2: nOffset_a |
| C00831/60 | L_OffsetGain_2: nGain_a |
| C00831/61 | L_PCTRL_1: nAct_a |
| C00831/62 | L_PCTRL_1: nAdapt_a |

| Parameter \| Name: **C00831 \| 16-bit inputs** | Data type: UNSIGNED_16 Index: $23744_d$ = $5CC0_h$ |
|---|---|
| C00831/63 | L_PCTRL_1: nSet_a |
| C00831/64 | L_PCTRL_1: nInfluence_a |
| C00831/65 | MCK: nSpeedCtrlI_a |
| C00831/66 | MCK: nPWMAngleOffset_a |
| C00831/67 | Reserved |
| C00831/68 | MCK: nBrkTorqueAdd_a |
| C00831/69 | MCK: nTorqueSetValue_a |
| C00831/70 | MCK: nTorqueLimitAdapt_a |
| C00831/71 | MCK: nSRampOverride_a |
| C00831/72 | MCK: nSpeedSetValue_a |
| C00831/73 | MCK: wMotionCtrl2 |
| C00831/74 | MCK: wMotionCtrl1 |
| C00831/75 | MCK: nSpeedOverride_a |
| C00831/76 | MCK: nAccOverride_a |
| C00831/77 | MCK: nSpeedAdd_v |
| C00831/78 | MCK: wAuxCtrl |
| C00831/79 | MCK: wSMCtrl |
| C00831/80 | L_OffsetGainP_3: nIn_a |
| C00831/81 | L_MPot_1: nIn_a |
| C00831/82 | L_MulDiv_1: nIn_a |
| C00831/83 | LS_DataAccess: wIn1 (Lenze-internal) |
| C00831/84 | LS_DataAccess: wIn2 (Lenze-internal) |
| C00831/85 | LS_DataAccess: wIn3 (Lenze-internal) |
| C00831/86 | LS_DataAccess: wIn4 (Lenze-internal) |
| C00831/87 | L_PT1_1: nIn_a |
| C00831/88 | MCTRL: nSpeedHighLimit_a |
| C00831/89 | L_PCTRL_1: nNSet_a |
| C00831/90 | L_PCTRL_1: nISet_a |
| C00831/91 | L_Interpolator_1: nPhdIn_v |
| C00831/92 | L_Interpolator_1: nNIn_a |
| C00831/93 | Reserved |
| C00831/94 | Reserved |
| C00831/95 | Reserved |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | |

**C00832**

| Parameter \| Name: **C00832 \| 16-bit inputs [rpm]** | | | Data type: INTEGER_16 Index: $23743_d$ = $5CBF_h$ |
|---|---|---|---|
| Display of 16-bit input values of different blocks in [rpm] | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -32767 | rpm | 32767 | |
| **Subcodes** | | | **Info** |
| C00832/1 | | | L_Absolut_1: nIn_a |
| C00832/2 | | | L_AddSub_1: nIn1_a |
| C00832/3 | | | L_AddSub_1: nIn2_a |
| C00832/4 | | | L_AddSub_1: nIn3_a |
| C00832/5 | | | L_OffsetGain_1: nIn_a |
| C00832/6 | | | L_OffsetGain_1: nOffset_a |
| C00832/7 | | | L_OffsetGain_1: nGain_a |
| C00832/8 | | | L_Negation_1: nIn_a |
| C00832/9 | | | L_GainOffset_1: nIn_a |
| C00832/10 | | | L_GainOffset_1: nGain_a |
| C00832/11 | | | L_GainOffset_1: nOffset_a |
| C00832/12 | | | L_Arithmetik_1: nIn1_a |
| C00832/13 | | | L_Arithmetik_1: nIn2_a |
| C00832/14 | | | L_AnalogSwitch_1: nIn1_a |
| C00832/15 | | | L_AnalogSwitch_1: nIn2_a |
| C00832/16 | | | L_Compare_1: nIn1_a |
| C00832/17 | | | L_Compare_1: nIn2_a |
| C00832/18 | | | MCTRL: nTorqueLimitAdapt_a |
| C00832/19 | | | Reserved |
| C00832/20 | | | MCTRL: nPosCtrlPAdapt_a |
| C00832/21 | | | MCTRL: nPosCtrlOutLimit_a |
| C00832/22 | | | MCTRL: nSpeedSetValue_a |
| C00832/23 | | | MCTRL: nSpeedLowLimit_a |
| C00832/24 | | | MCTRL: nSpeedCtrlI_a |
| C00832/25 | | | MCTRL: nSpeedCtrlPAdapt_a |
| C00832/26 | | | MCTRL: nBoost_a |
| C00832/27 | | | MCTRL: nTorqueSetValue_a |
| C00832/28 | | | MCTRL: nTorqueGenLimit_a |
| C00832/29 | | | MCTRL: nTorqueMotLimit_a |
| C00832/30 | | | Reserved |
| C00832/31 | | | MCTRL: nVoltageAdd_a |
| C00832/32 | | | MCTRL: nPWMAngleOffset_a |
| C00832/33 | | | L_NSet_1: nCInhVal_a |
| C00832/34 | | | L_NSet_1: nNSet_a |
| C00832/35 | | | L_NSet_1: nSet_a |
| C00832/36 | | | L_NSet_1: nNAdd_a |
| C00832/37 | | | DCTRL: wCANControl |
| C00832/38 | | | DCTRL: wCCMControl |
| C00832/39 | | | L_NLim_1: nIn_a |
| C00832/40 | | | Reserved |

| Parameter \| Name: **C00832 \| 16-bit inputs [rpm]** | Data type: INTEGER_16 Index: $23743_d$ = $5CBF_h$ |
|---|---|
| C00832/41 | L_Compare_2: nIn1_a |
| C00832/42 | L_Compare_2: nIn2_a |
| C00832/43 | L_Compare_3: nIn1_a |
| C00832/44 | L_Compare_3: nIn2_a |
| C00832/45 | L_AnalogSwitch_2: nIn1_a |
| C00832/46 | L_AnalogSwitch_2: nIn2_a |
| C00832/47 | L_AnalogSwitch_3: nIn1_a |
| C00832/48 | L_AnalogSwitch_3: nIn2_a |
| C00832/49 | L_Arithmetik_2: nIn1_a |
| C00832/50 | L_Arithmetik_2: nIn2_a |
| C00832/51 | Reserved |
| C00832/52 | Reserved |
| C00832/53 | L_GainOffset_2: nIn_a |
| C00832/54 | L_GainOffset_2: nGain_a |
| C00832/55 | L_GainOffset_2: nOffset_a |
| C00832/56 | L_OffsetGainP_1: nIn_a |
| C00832/57 | L_OffsetGainP_2: nIn_a |
| C00832/58 | L_OffsetGain_2: nIn_a |
| C00832/59 | L_OffsetGain_2: nOffset_a |
| C00832/60 | L_OffsetGain_2: nGain_a |
| C00832/61 | L_PCTRL_1: nAct_a |
| C00832/62 | L_PCTRL_1: nAdapt_a |
| C00832/63 | L_PCTRL_1: nSet_a |
| C00832/64 | L_PCTRL_1: nInfluence_a |
| C00832/65 | MCK: nSpeedCtrlI_a |
| C00832/66 | MCK: nPWMAngleOffset_a |
| C00832/67 | Reserved |
| C00832/68 | MCK: nBrkTorqueAdd_a |
| C00832/69 | MCK: nTorqueSetValue_a |
| C00832/70 | MCK: nTorqueLimitAdapt_a |
| C00832/71 | MCK: nSRampOverride_a |
| C00832/72 | MCK: nSpeedSetValue_a |
| C00832/73 | MCK: wMotionCtrl2 |
| C00832/74 | MCK: wMotionCtrl1 |
| C00832/75 | MCK: nSpeedOverride_a |
| C00832/76 | MCK: nAccOverride_a |
| C00832/77 | MCK: nSpeedAdd_v |
| C00832/78 | MCK: wAuxCtrl |
| C00832/79 | MCK: wSMCtrl |
| C00832/80 | L_OffsetGainP_3: nIn_a |
| C00832/81 | L_MPot_1: nIn_a |
| C00832/82 | L_MulDiv_1: nIn_a |
| C00832/83 | LS_DataAccess: wIn1 (Lenze-internal) |

| Parameter \| Name: **C00832 \| 16-bit inputs [rpm]** | Data type: INTEGER_16 Index: 23743$_d$ = 5CBF$_h$ |
|---|---|
| C00832/84 | LS_DataAccess: wIn2 (Lenze-internal) |
| C00832/85 | LS_DataAccess: wIn3 (Lenze-internal) |
| C00832/86 | LS_DataAccess: wIn4 (Lenze-internal) |
| C00832/87 | L_PT1_1: nIn_a |
| C00832/88 | MCTRL: nSpeedHighLimit_a |
| C00832/89 | L_PCTRL_1: nNSet_a |
| C00832/90 | L_PCTRL_1: nISet_a |
| C00832/91 | L_Interpolator_1: nPhdIn_v |
| C00832/92 | L_Interpolator_1: nNIn_a |
| C00832/93 | Reserved |
| C00832/94 | Reserved |
| C00832/95 | Reserved |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

## C00833

| Parameter \| Name: **C00833 \| Binary inputs** | Data type: UNSIGNED_8 Index: 23742$_d$ = 5CBE$_h$ |
|---|---|
| Display of the signal status of the binary inputs of different blocks | |
| **Selection list** | |
| 0 False | |
| 1 True | |
| **Subcodes** | **Info** |
| C00833/1 | L_And_1: bIn1 |
| C00833/2 | L_And_1: bIn2 |
| C00833/3 | L_And_1: bIn3 |
| C00833/4 | L_DFlipFlop_1: bD |
| C00833/5 | L_DFlipFlop_1: bClk |
| C00833/6 | L_DFlipFlop_1: bClr |
| C00833/7 | L_Not_1: bIn |
| C00833/8 | L_Or_1: bIn1 |
| C00833/9 | L_Or_1: bIn2 |
| C00833/10 | L_Or_1: bIn3 |
| C00833/11 | L_RLQ_1: bCw |
| C00833/12 | L_RLQ_1: bCcw |
| C00833/13 | L_AnalogSwitch_1: bSet |
| C00833/14 | L_NSet_1: bRfgStop |
| C00833/15 | L_NSet_1: bRfg0 |
| C00833/16 | L_NSet_1: bNSetInv |
| C00833/17 | L_NSet_1: bJog1 |
| C00833/18 | L_NSet_1: bJog2 |
| C00833/19 | L_NSet_1: bJog4 |
| C00833/20 | L_NSet_1: bJog8 |
| C00833/21 | L_NSet_1: bTi1 |

| Parameter \| Name: C00833 \| Binary inputs | Data type: UNSIGNED_8 Index: $23742_d$ = $5CBE_h$ |
|---|---|
| C00833/22 | L_NSet_1: bTi2 |
| C00833/23 | L_NSet_1: bTi4 |
| C00833/24 | L_NSet_1: bTi8 |
| C00833/25 | L_NSet_1: bLoad |
| C00833/26 | L_NSet_1: bExternalCINH |
| C00833/27 | MCTRL: bPosCtrlOn |
| C00833/28 | MCTRL: bSpeedInterpolatorOn |
| C00833/29 | MCTRL: bTorqueInterpolatorOn |
| C00833/30 | MCTRL: bTorquemodeOn |
| C00833/31 | MCTRL: bSpeedCtrlIOn |
| C00833/32 | MCTRL: bAutoBoostOn |
| C00833/33 | MCTRL: bQSPOn |
| C00833/34 | MCTRL: bDcBrakeOn |
| C00833/35 | MCTRL: bDeltaPosOn |
| C00833/36 | DCTRL: bCINH |
| C00833/37 | DCTRL: bFailReset |
| C00833/38 | DCTRL: bStatus_B0 |
| C00833/39 | DCTRL: bStatus_B2 |
| C00833/40 | DCTRL: bStatus_B3 |
| C00833/41 | DCTRL: bStatus_B4 |
| C00833/42 | DCTRL: bStatus_B5 |
| C00833/43 | DCTRL: bStatus_B14 |
| C00833/44 | DCTRL: bStatus_B15 |
| C00833/45 | DCTRL: bFree_1 |
| C00833/46 | DCTRL: bFree_2 |
| C00833/47 | DCTRL: bFree_3 |
| C00833/48 | DCTRL: bFree_4 |
| C00833/49 | L_And_2: bIn1 |
| C00833/50 | L_And_2: bIn2 |
| C00833/51 | L_And_2: bIn3 |
| C00833/52 | L_And_3: bIn1 |
| C00833/53 | L_And_3: bIn2 |
| C00833/54 | L_And_3: bIn3 |
| C00833/55 | L_Or_2: bIn1 |
| C00833/56 | L_Or_2: bIn2 |
| C00833/57 | L_Or_2: bIn3 |
| C00833/58 | L_Or_3: bIn1 |
| C00833/59 | L_Or_3: bIn2 |
| C00833/60 | L_Or_3: bIn3 |
| C00833/61 | L_Not_2: bIn |
| C00833/62 | L_Not_3: bIn |
| C00833/63 | L_DigitalLogic_1: bIn1 |
| C00833/64 | L_DigitalLogic_1: bIn2 |
| C00833/65 | L_DigitalLogic_1: bIn3 |

| Parameter \| Name: C00833 \| Binary inputs | Data type: UNSIGNED_8 Index: $23742_d$ = $5CBE_h$ |
|---|---|
| C00833/66 | L_DigitalDelay_1: bIn |
| C00833/67 | MCTRL: bPosDerivativeOn |
| C00833/68 | MCTRL: bMotorRefOffsetOn |
| C00833/69 | MCTRL: bSpeedCtrlPAdaptOn |
| C00833/70 | L_AnalogSwitch_2: bSet |
| C00833/71 | L_AnalogSwitch_3: bSet |
| C00833/72 | L_MPot_1: bUp |
| C00833/73 | L_MPot_1: bInAct |
| C00833/74 | L_MPot_1: bDown |
| C00833/75 | L_PCTRL_1: bPIDOff |
| C00833/76 | L_PCTRL_1: bInAct |
| C00833/77 | L_PCTRL_1: bIOff |
| C00833/78 | MCK: bSpeedCtrlIOn |
| C00833/79 | MCK: bDcBrakeOn |
| C00833/80 | MCK: bBrkRelease |
| C00833/81 | MCK: bBrkStartTorqueDir |
| C00833/82 | MCK: bBrkApplied |
| C00833/83 | MCK: bLimitSwitchPos |
| C00833/84 | MCK: bLimitSwitchNeg |
| C00833/85 | MCK: bPosCtrlOn |
| C00833/86 | MCK: bDeltaPosOn |
| C00833/87 | MCK: bPosDerivativeOn |
| C00833/88 | MCK: bMotorRefOffsetOn |
| C00833/89 | MCK: bQspOn |
| C00833/90 | MCK: bTorquemodeOn |
| C00833/91 | MCK: bTorqueLimitAdaptOn |
| C00833/92 | MCK: bHomMark |
| C00833/93 | L_Transient_1: bIn |
| C00833/94 | L_Transient_2: bIn |
| C00833/95 | L_Transient_3: bIn |
| C00833/96 | L_Transient_4: bIn |
| C00833/97 | Reserved |
| C00833/98 | MCTRL: bTorqueLimitAdaptOn |
| C00833/99 | L_NSet_1: bNAddInv |
| C00833/100 | L_MPot_1: bEnable |
| C00833/101 | L_NLim_1: bEnable |
| C00833/102 | LS_DataAccess: bEnableIn1 (Lenze-internal) |
| C00833/103 | LS_DataAccess: bEnableIn2 (Lenze-internal) |
| C00833/104 | LS_DataAccess: bEnableIn3 (Lenze-internal) |
| C00833/105 | LS_DataAccess: bEnableIn4 (Lenze-internal) |
| C00833/106 | L_PCTRL_1: bEnableInfluenceRamp |

| Parameter \| Name: **C00833 \| Binary inputs** | Data type: UNSIGNED_8 Index: $23742_d$ = $5CBE_h$ |
|---|---|
| C00833/107 | LS_SetError_2: bSetError1 |
| C00833/108 | LS_SetError_2: bSetError2 |
| C00833/109 | LS_SetError_2: bSetError3 |
| C00833/110 | LS_SetError_2: bSetError4 |
| C00833/111 | L_JogCtrlExtension: bInputSel1 |
| C00833/112 | L_JogCtrlExtension: bInputSel2 |
| C00833/113 | L_JogCtrlExtension: bRfgIn |
| C00833/114 | L_JogCtrlExtension: bJog1In |
| C00833/115 | L_JogCtrlExtension: bJog2In |
| C00833/116 | L_JogCtrlExtension: bSlowDown1 |
| C00833/117 | L_JogCtrlExtension: bStop1 |
| C00833/118 | L_JogCtrlExtension: bbSlowDown2 |
| C00833/119 | L_JogCtrlExtension: bStop2 |
| C00833/120 | L_JogCtrlExtension: bSlowDown3 |
| C00833/121 | L_JogCtrlExtension: bStop3 |
| C00833/122 | L_PCTRL_1: bISet |
| C00833/123 | L_Interpolator_1: bSpeedAct0 |
| C00833/124 | L_Or_4: bIn1 |
| C00833/125 | L_Or_4: bIn2 |
| C00833/126 | L_Or_4: bIn3 |
| C00833/127 | L_DigitalLogic_3: bIn1 |
| C00833/128 | L_DigitalLogic_3: bIn2 |
| C00833/129 | L_DigitalLogic_3: bIn3 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00834**

| Parameter \| Name: **C00834 \| 32-bit inputs [incr]** | | | Data type: INTEGER_32 Index: $23741_d$ = $5CBD_h$ |
|---|---|---|---|
| Display in [increments] of 32 bit input values of various blocks | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C00834/1 | | | MCK: dnPosSetValue_p |
| C00834/2 | | | MCK: dnMotorRefOffset_p |
| C00834/3 | | | MCK: dnDeltaPos_p |
| C00834/4 | | | MCTRL: dnDeltaPos_p |
| C00834/5 | | | MCTRL: dnPosSetValue_p |
| C00834/6 | | | MCTRL: dnMotorRefOffset_p |
| C00834/7 | | | MCK: dnProfilePosition_p |
| C00834/8 | | | L_Interpolator_1: dnPhiIn_p |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00835**

| Parameter \| Name: **C00835 \| 16-bit inputs [%] (Set2)** | | | Data type: INTEGER_16 Index: $23740_d$ = $5CBC_h$ |
|---|---|---|---|
| Display in percent of 16-bit input values of different blocks | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00835/1 | | | L_Absolut_2: nIn_a |
| C00835/2 | | | L_AnalogSwitch_4: nIn1_a |
| C00835/3 | | | L_AnalogSwitch_4: nIn2_a |
| C00835/4 | | | L_AnalogSwitch_5: nIn1_a |
| C00835/5 | | | L_AnalogSwitch_5: nIn2_a |
| C00835/6 | | | L_Compare_4: nIn1_a |
| C00835/7 | | | L_Compare_4: nIn2_a |
| C00835/8 | | | L_Compare_5: nIn1_a |
| C00835/9 | | | L_Compare_5: nIn2_a |
| C00835/10 | | | L_Arithmetik_3: nIn1_a |
| C00835/11 | | | L_Arithmetik_3: nIn2_a |
| C00835/12 | | | L_Arithmetik_4: nIn1_a |
| C00835/13 | | | L_Arithmetik_4: nIn2_a |
| C00835/14 | | | L_Arithmetik_5: nIn1_a |
| C00835/15 | | | L_Arithmetik_5: nIn2_a |
| C00835/16 | | | L_Counter_2: wLdVal |
| C00835/17 | | | L_Counter_2: wCmpVal |
| C00835/18 | | | L_Counter_3: wLdVal |
| C00835/19 | | | L_Counter_3: wCmpVal |
| C00835/20 | | | L_PhaseIntK_1: nIn_v |
| C00835/21 | | | L_Negation_2: nIn_a |
| C00835/22 | | | L_NLim_2: nIn_a |
| C00835/23 | | | L_OffsetGain_3: nIn_a |
| C00835/24 | | | L_OffsetGain_3: nOffset_a |
| C00835/25 | | | L_OffsetGain_3: nGain_a |
| C00835/26 | | | L_PT1_2: nIn_a |
| C00835/27 | | | L_PT1_3: nIn_a |
| C00835/28 | | | L_PhaseIntK_2: nIn_v |
| C00835/29 | | | L_SampleHold_1: nIn_a |
| C00835/30 | | | L_SampleHold_2: nIn_a |
| C00835/31 | | | L_Mux_1: wInSelect |
| C00835/32 | | | L_GainOffset_3: nIn_a |
| C00835/33 | | | L_GainOffset_3: nGain_a |
| C00835/34 | | | L_GainOffset_3: nOffset_a |
| C00835/35 | | | L_MulDiv_2: nIn_a |
| C00835/36 | | | L_DT1_1: nIn_a |
| C00835/37 | | | L_Counter_1: wLdVal |
| C00835/38 | | | L_Counter_1: wCmpVal |
| C00835/39 | | | L_GainOffsetP_1: nIn_a |
| C00835/40 | | | L_GainOffsetP_2: nIn_a |

| Parameter \| Name: C00835 \| 16-bit inputs [%] (Set2) | Data type: INTEGER_16 Index: $23740_d$ = $5CBC_h$ |
|---|---|
| C00835/41 | L_GainOffsetP_3: nIn_a |
| C00835/42 | L_Limit_1: nIn_a |
| C00835/43 | L_Limit_2: nIn_a |
| C00835/44 | L_MckCtrlInterface_1: wOperationMode |
| C00835/45 | L_MckCtrlInterface_1: wPosMode |
| C00835/46 | L_MckCtrlInterface_1: wProfileNo |
| C00835/47 | L_MckCtrlInterface_1: wInMckPosCtrl_1 |
| C00835/48 | L_MckCtrlInterface_1: wInMckPosCtrl_2 |
| C00835/49 | L_MckStateInterface_1: wInMckPosState_1 |
| C00835/50 | L_MckStateInterface_1: wInMckPosState_2 |
| C00835/51 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_1 |
| C00835/52 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_2 |
| C00835/53 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_3 |
| C00835/54 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_4 |
| C00835/55 | L_ConvWordToBits_1: wInput |
| C00835/56 | L_ConvWordToBits_2: wInput |
| C00835/57 | L_ConvWordToBits_3: wInput |
| C00835/58 | L_ConvWordsToDInt_1: wInLWord |
| C00835/59 | L_ConvWordsToDInt_1: wInHWord |
| C00835/60 | L_ConvWordsToDInt_2: wInLWord |
| C00835/61 | L_ConvWordsToDInt_2: wInHWord |
| C00835/62 | L_ConvWordsToDInt_3: wInLWord |
| C00835/63 | L_ConvWordsToDInt_3: wInHWord |
| C00835/64 | L_ConvUnitsToIncr_1: wInLWord |
| C00835/65 | L_ConvUnitsToIncr_1: wInHWord |
| C00835/66 | L_ConvUnitsToIncr_2: wInLWord |
| C00835/67 | L_ConvUnitsToIncr_2: wInHWord |
| C00835/68 | L_ConvUnitsToIncr_3: wInLWord |
| C00835/69 | L_ConvUnitsToIncr_3: wInHWord |
| C00835/70 | L_Curve_1: nIn_a |
| C00835/71 | L_ConvW_1: wIn |
| C00835/72 | L_ConvW_2: wIn |
| C00835/73 | L_ConvW_3: wIn |
| C00835/74 | L_ConvW_4: wIn |
| C00835/75 | L_MckCtrlInterface_1: wPosSetHW |
| C00835/76 | L_MckCtrlInterface_1: wPosSetLW |
| C00835/77 | L_PhaseDiff_1: nIn_v |
| C00835/78 | L_PhaseDiff_2: nIn_v |
| C00835/79 | L_SRFG_1: nIn_a |
| C00835/80 | L_SRFG_2: nIn_a |
| C00835/81 | L_SRFG_1: nSet_a |
| C00835/82 | L_SRFG_2: nSet_a |
| C00835/83 | L_SignalSwitch_1: wIn1 |
| C00835/84 | L_SignalSwitch_2: wIn1 |

| Parameter \| Name: C00835 \| 16-bit inputs [%] (Set2) | Data type: INTEGER_16 Index: $23740_d$ = $5CBC_h$ |
|---|---|
| C00835/85 | L_SignalSwitch_3: wIn1 |
| C00835/86 | L_SignalSwitch_4: wIn1 |
| C00835/87 | L_SignalSwitch_1: wIn2 |
| C00835/88 | L_SignalSwitch_2: wIn2 |
| C00835/89 | L_SignalSwitch_3: wIn2 |
| C00835/90 | L_SignalSwitch_4: wIn2 |
| C00835/91 | L_Odometer_1: nInSpeed_v |
| C00835/92 | L_CalcDiameter_1: wDMax |
| C00835/93 | L_CalcDiameter_1: wDMin |
| C00835/94 | L_CalcDiameter_1: wVMax |
| C00835/95 | L_CalcDiameter_1: nVLine_a |
| C00835/96 | L_CalcDiameter_1: nMotorSpeedAct_v |
| C00835/97 | L_CalcDiameter_1: wGearNum |
| C00835/98 | L_CalcDiameter_1: wGearDenom |
| C00835/99 | L_CalcDiameter_1: nSetD_a |
| C00835/100 | L_ProcessCtrl_1: nVpAdapt_a |
| C00835/101 | L_ProcessCtrl_1: nSet_a |
| C00835/102 | L_ProcessCtrl_1: nAct_a |
| C00835/103 | L_ProcessCtrl_1: nRTimeAdapt_a |
| C00835/104 | L_ProcessCtrl_1: nInfluence_a |
| C00835/105 | L_PhiIntegrator_1: nIn_v |
| C00835/106 | L_PhiIntegrator_1: wGearNum |
| C00835/107 | L_PhiIntegrator_1: wGearDenom |
| C00835/108 | L_SwitchPoint_1: nActSpeed_v |
| C00835/109 | L_PhiIntegrator_1: nSpeedAdd_v |
| C00835/110 | L_DFSET_1: nSpeedTrim_v |
| C00835/111 | L_DFSET_1: nSpeedTrim_a |
| C00835/112 | L_DFSET_1: wGearNum |
| C00835/113 | L_DFSET_1: wGainNum |
| C00835/114 | L_DFSET_1: nSet_v |
| C00835/115 | L_DFSET_1: wGainDenom |
| C00835/116 | L_DFSET_1: wGearDenom |
| C00835/117 | L_DFSET_1: nPositionTrimming |
| C00835/118 | L_DFSET_1: nNAct_v |
| C00835/119 | L_DFRFG_1: nIn_v |
| C00835/120 | L_GearComp_1: nTorque_a |
| C00835/121 | L_ConvAP_1: nIn_a |
| C00835/122 | L_ConvAP_2: nIn_a |
| C00835/123 | L_ConvAP_3: nIn_a |
| C00835/124 | L_ConvX_1: nIn_a |
| C00835/125 | L_ConvX_1: nNum |
| C00835/126 | L_ConvX_1: wDenom |
| C00835/127 | L_ConvX_2: nIn_a |
| C00835/128 | L_ConvX_2: nNum |

| Parameter \| Name: C00835 \| 16-bit inputs [%] (Set2) | Data type: INTEGER_16 Index: 23740$_d$ = 5CBC$_h$ |
|---|---|
| C00835/129 | L_ConvX_2: wDenom |
| C00835/130 | L_ConvX_3: nIn_a |
| C00835/131 | L_ConvX_3: nNum |
| C00835/132 | L_ConvX_3: wDenom |
| C00835/133 | L_ConvPP_1: nNum |
| C00835/134 | L_ConvPP_1: wDenom |
| C00835/135 | L_ConvPP_2: nNum |
| C00835/136 | L_ConvPP_2: wDenom |
| C00835/137 | L_ConvPP_3: nNum |
| C00835/138 | L_ConvPP_3: wDenom |
| C00835/139 | L_Curve_2: nIn_a |
| C00835/140 | L_Curve_3: nIn_a |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | |

**C00836**

| Parameter \| Name: C00836 \| 16-bit inputs (Set2) | | | Data type: UNSIGNED_16 Index: 23739$_d$ = 5CBB$_h$ |
|---|---|---|---|
| Decimal/hexadecimal/bit coded display of 16 bit input values of various blocks | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00836/1 | | | L_Absolut_2: nIn_a |
| C00836/2 | | | L_AnalogSwitch_4: nIn1_a |
| C00836/3 | | | L_AnalogSwitch_4: nIn2_a |
| C00836/4 | | | L_AnalogSwitch_5: nIn1_a |
| C00836/5 | | | L_AnalogSwitch_5: nIn2_a |
| C00836/6 | | | L_Compare_4: nIn1_a |
| C00836/7 | | | L_Compare_4: nIn2_a |
| C00836/8 | | | L_Compare_5: nIn1_a |
| C00836/9 | | | L_Compare_5: nIn2_a |
| C00836/10 | | | L_Arithmetik_3: nIn1_a |
| C00836/11 | | | L_Arithmetik_3: nIn2_a |
| C00836/12 | | | L_Arithmetik_4: nIn1_a |
| C00836/13 | | | L_Arithmetik_4: nIn2_a |
| C00836/14 | | | L_Arithmetik_5: nIn1_a |
| C00836/15 | | | L_Arithmetik_5: nIn2_a |
| C00836/16 | | | L_Counter_2: wLdVal |
| C00836/17 | | | L_Counter_2: wCmpVal |
| C00836/18 | | | L_Counter_3: wLdVal |
| C00836/19 | | | L_Counter_3: wCmpVal |
| C00836/20 | | | L_PhaseIntK_1: nIn_v |

| Parameter \| Name: C00836 \| 16-bit inputs (Set2) | Data type: UNSIGNED_16 Index: $23739_d$ = $5CBB_h$ |
|---|---|
| C00836/21 | L_Negation_2: nIn_a |
| C00836/22 | L_NLim_2: nIn_a |
| C00836/23 | L_OffsetGain_3: nIn_a |
| C00836/24 | L_OffsetGain_3: nOffset_a |
| C00836/25 | L_OffsetGain_3: nGain_a |
| C00836/26 | L_PT1_2: nIn_a |
| C00836/27 | L_PT1_3: nIn_a |
| C00836/28 | L_PhaseIntK_2: nIn_v |
| C00836/29 | L_SampleHold_1: nIn_a |
| C00836/30 | L_SampleHold_2: nIn_a |
| C00836/31 | L_Mux_1: wInSelect |
| C00836/32 | L_GainOffset_3: nIn_a |
| C00836/33 | L_GainOffset_3: nGain_a |
| C00836/34 | L_GainOffset_3: nOffset_a |
| C00836/35 | L_MulDiv_2: nIn_a |
| C00836/36 | L_DT1_1: nIn_a |
| C00836/37 | L_Counter_1: wLdVal |
| C00836/38 | L_Counter_1: wCmpVal |
| C00836/39 | L_GainOffsetP_1: nIn_a |
| C00836/40 | L_GainOffsetP_2: nIn_a |
| C00836/41 | L_GainOffsetP_3: nIn_a |
| C00836/42 | L_Limit_1: nIn_a |
| C00836/43 | L_Limit_2: nIn_a |
| C00836/44 | L_MckCtrlInterface_1: wOperationMode |
| C00836/45 | L_MckCtrlInterface_1: wPosMode |
| C00836/46 | L_MckCtrlInterface_1: wProfileNo |
| C00836/47 | L_MckCtrlInterface_1: wInMckPosCtrl_1 |
| C00836/48 | L_MckCtrlInterface_1: wInMckPosCtrl_2 |
| C00836/49 | L_MckStateInterface_1: wInMckPosState_1 |
| C00836/50 | L_MckStateInterface_1: wInMckPosState_2 |
| C00836/51 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_1 |
| C00836/52 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_2 |
| C00836/53 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_3 |
| C00836/54 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_4 |
| C00836/55 | L_ConvWordToBits_1: wInput |
| C00836/56 | L_ConvWordToBits_2: wInput |
| C00836/57 | L_ConvWordToBits_3: wInput |
| C00836/58 | L_ConvWordsToDInt_1: wInLWord |
| C00836/59 | L_ConvWordsToDInt_1: wInHWord |
| C00836/60 | L_ConvWordsToDInt_2: wInLWord |
| C00836/61 | L_ConvWordsToDInt_2: wInHWord |
| C00836/62 | L_ConvWordsToDInt_3: wInLWord |
| C00836/63 | L_ConvWordsToDInt_3: wInHWord |
| C00836/64 | L_ConvUnitsToIncr_1: wInLWord |

| Parameter \| Name: **C00836 \| 16-bit inputs (Set2)** | Data type: UNSIGNED_16 Index: $23739_d$ = $5CBB_h$ |
|---|---|
| C00836/65 | L_ConvUnitsToIncr_1: wInHWord |
| C00836/66 | L_ConvUnitsToIncr_2: wInLWord |
| C00836/67 | L_ConvUnitsToIncr_2: wInHWord |
| C00836/68 | L_ConvUnitsToIncr_3: wInLWord |
| C00836/69 | L_ConvUnitsToIncr_3: wInHWord |
| C00836/70 | L_Curve_1: nIn_a |
| C00836/71 | L_ConvW_1: wIn |
| C00836/72 | L_ConvW_2: wIn |
| C00836/73 | L_ConvW_3: wIn |
| C00836/74 | L_ConvW_4: wIn |
| C00836/75 | L_MckCtrlInterface_1: wPosSetHW |
| C00836/76 | L_MckCtrlInterface_1: wPosSetLW |
| C00836/77 | L_PhaseDiff_1: nIn_v |
| C00836/78 | L_PhaseDiff_2: nIn_v |
| C00836/79 | L_SRFG_1: nIn_a |
| C00836/80 | L_SRFG_2: nIn_a |
| C00836/81 | L_SRFG_1: nSet_a |
| C00836/82 | L_SRFG_2: nSet_a |
| C00836/83 | L_SignalSwitch_1: wIn1 |
| C00836/84 | L_SignalSwitch_2: wIn1 |
| C00836/85 | L_SignalSwitch_3: wIn1 |
| C00836/86 | L_SignalSwitch_4: wIn1 |
| C00836/87 | L_SignalSwitch_1: wIn2 |
| C00836/88 | L_SignalSwitch_2: wIn2 |
| C00836/89 | L_SignalSwitch_3: wIn2 |
| C00836/90 | L_SignalSwitch_4: wIn2 |
| C00836/91 | L_Odometer_1: nInSpeed_v |
| C00836/92 | L_CalcDiameter_1: wDMax |
| C00836/93 | L_CalcDiameter_1: wDMin |
| C00836/94 | L_CalcDiameter_1: wVMax |
| C00836/95 | L_CalcDiameter_1: nVLine_a |
| C00836/96 | L_CalcDiameter_1: nMotorSpeedAct_v |
| C00836/97 | L_CalcDiameter_1: wGearNum |
| C00836/98 | L_CalcDiameter_1: wGearDenom |
| C00836/99 | L_CalcDiameter_1: nSetD_a |
| C00836/100 | L_ProcessCtrl_1: nVpAdapt_a |
| C00836/101 | L_ProcessCtrl_1: nSet_a |
| C00836/102 | L_ProcessCtrl_1: nAct_a |
| C00836/103 | L_ProcessCtrl_1: nRTimeAdapt_a |
| C00836/104 | L_ProcessCtrl_1: nInfluence_a |
| C00836/105 | L_PhiIntegrator_1: nIn_v |
| C00836/106 | L_PhiIntegrator_1: wGearNum |
| C00836/107 | L_PhiIntegrator_1: wGearDenom |
| C00836/108 | L_SwitchPoint_1: nActSpeed_v |

| Parameter \| Name: C00836 \| 16-bit inputs (Set2) | Data type: UNSIGNED_16 Index: 23739$_d$ = 5CBB$_h$ |
|---|---|
| C00836/109 | L_PhiIntegrator_1: nSpeedAdd_v |
| C00836/110 | L_DFSET_1: nSpeedTrim_v |
| C00836/111 | L_DFSET_1: nSpeedTrim_a |
| C00836/112 | L_DFSET_1: wGearNum |
| C00836/113 | L_DFSET_1: wGainNum |
| C00836/114 | L_DFSET_1: nSet_v |
| C00836/115 | L_DFSET_1: wGainDenom |
| C00836/116 | L_DFSET_1: wGearDenom |
| C00836/117 | L_DFSET_1: nPositionTrimming |
| C00836/118 | L_DFSET_1: nNAct_v |
| C00836/119 | L_DFRFG_1: nIn_v |
| C00836/120 | L_GearComp_1: nTorque_a |
| C00836/121 | L_ConvAP_1: nIn_a |
| C00836/122 | L_ConvAP_2: nIn_a |
| C00836/123 | L_ConvAP_3: nIn_a |
| C00836/124 | L_ConvX_1: nIn_a |
| C00836/125 | L_ConvX_1: nNum |
| C00836/126 | L_ConvX_1: wDenom |
| C00836/127 | L_ConvX_2: nIn_a |
| C00836/128 | L_ConvX_2: nNum |
| C00836/129 | L_ConvX_2: wDenom |
| C00836/130 | L_ConvX_3: nIn_a |
| C00836/131 | L_ConvX_3: nNum |
| C00836/132 | L_ConvX_3: wDenom |
| C00836/133 | L_ConvPP_1: nNum |
| C00836/134 | L_ConvPP_1: wDenom |
| C00836/135 | L_ConvPP_2: nNum |
| C00836/136 | L_ConvPP_2: wDenom |
| C00836/137 | L_ConvPP_3: nNum |
| C00836/138 | L_ConvPP_3: wDenom |
| C00836/139 | L_Curve_2: nIn_a |
| C00836/140 | L_Curve_3: nIn_a |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

## C00837

| Parameter \| Name: C00837 \| 16-bit inputs [rpm] (Set2) | | | Data type: INTEGER_16 Index: 23738$_d$ = 5CBA$_h$ |
|---|---|---|---|
| Display of 16-bit input values of different blocks in [rpm] | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -32767 | rpm | 32767 | |
| **Subcodes** | | | **Info** |
| C00837/1 | | | L_Absolut_2: nIn_a |
| C00837/2 | | | L_AnalogSwitch_4: nIn1_a |
| C00837/3 | | | L_AnalogSwitch_4: nIn2_a |
| C00837/4 | | | L_AnalogSwitch_5: nIn1_a |

| Parameter \| Name: **C00837 \| 16-bit inputs [rpm] (Set2)** | Data type: INTEGER_16 Index: $23738_d$ = $5CBA_h$ |
|---|---|
| C00837/5 | L_AnalogSwitch_5: nIn2_a |
| C00837/6 | L_Compare_4: nIn1_a |
| C00837/7 | L_Compare_4: nIn2_a |
| C00837/8 | L_Compare_5: nIn1_a |
| C00837/9 | L_Compare_5: nIn2_a |
| C00837/10 | L_Arithmetik_3: nIn1_a |
| C00837/11 | L_Arithmetik_3: nIn2_a |
| C00837/12 | L_Arithmetik_4: nIn1_a |
| C00837/13 | L_Arithmetik_4: nIn2_a |
| C00837/14 | L_Arithmetik_5: nIn1_a |
| C00837/15 | L_Arithmetik_5: nIn2_a |
| C00837/16 | L_Counter_2: wLdVal |
| C00837/17 | L_Counter_2: wCmpVal |
| C00837/18 | L_Counter_3: wLdVal |
| C00837/19 | L_Counter_3: wCmpVal |
| C00837/20 | L_PhaseIntK_1: nIn_v |
| C00837/21 | L_Negation_2: nIn_a |
| C00837/22 | L_NLim_2: nIn_a |
| C00837/23 | L_OffsetGain_3: nIn_a |
| C00837/24 | L_OffsetGain_3: nOffset_a |
| C00837/25 | L_OffsetGain_3: nGain_a |
| C00837/26 | L_PT1_2: nIn_a |
| C00837/27 | L_PT1_3: nIn_a |
| C00837/28 | L_PhaseIntK_2: nIn_v |
| C00837/29 | L_SampleHold_1: nIn_a |
| C00837/30 | L_SampleHold_2: nIn_a |
| C00837/31 | L_Mux_1: wInSelect |
| C00837/32 | L_GainOffset_3: nIn_a |
| C00837/33 | L_GainOffset_3: nGain_a |
| C00837/34 | L_GainOffset_3: nOffset_a |
| C00837/35 | L_MulDiv_2: nIn_a |
| C00837/36 | L_DT1_1: nIn_a |
| C00837/37 | L_Counter_1: wLdVal |
| C00837/38 | L_Counter_1: wCmpVal |
| C00837/39 | L_GainOffsetP_1: nIn_a |
| C00837/40 | L_GainOffsetP_2: nIn_a |
| C00837/41 | L_GainOffsetP_3: nIn_a |
| C00837/42 | L_Limit_1: nIn_a |
| C00837/43 | L_Limit_2: nIn_a |
| C00837/44 | L_MckCtrlInterface_1: wOperationMode |
| C00837/45 | L_MckCtrlInterface_1: wPosMode |
| C00837/46 | L_MckCtrlInterface_1: wProfileNo |
| C00837/47 | L_MckCtrlInterface_1: wInMckPosCtrl_1 |
| C00837/48 | L_MckCtrlInterface_1: wInMckPosCtrl_2 |

| Parameter \| Name: C00837 \| 16-bit inputs [rpm] (Set2) | Data type: INTEGER_16 Index: $23738_d$ = $5CBA_h$ |
|---|---|
| C00837/49 | L_MckStateInterface_1: wInMckPosState_1 |
| C00837/50 | L_MckStateInterface_1: wInMckPosState_2 |
| C00837/51 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_1 |
| C00837/52 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_2 |
| C00837/53 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_3 |
| C00837/54 | L_PosiShaftCtrlInterface_1: wInPosiShaftCtrl_4 |
| C00837/55 | L_ConvWordToBits_1: wInput |
| C00837/56 | L_ConvWordToBits_2: wInput |
| C00837/57 | L_ConvWordToBits_3: wInput |
| C00837/58 | L_ConvWordsToDInt_1: wInLWord |
| C00837/59 | L_ConvWordsToDInt_1: wInHWord |
| C00837/60 | L_ConvWordsToDInt_2: wInLWord |
| C00837/61 | L_ConvWordsToDInt_2: wInHWord |
| C00837/62 | L_ConvWordsToDInt_3: wInLWord |
| C00837/63 | L_ConvWordsToDInt_3: wInHWord |
| C00837/64 | L_ConvUnitsToIncr_1: wInLWord |
| C00837/65 | L_ConvUnitsToIncr_1: wInHWord |
| C00837/66 | L_ConvUnitsToIncr_2: wInLWord |
| C00837/67 | L_ConvUnitsToIncr_2: wInHWord |
| C00837/68 | L_ConvUnitsToIncr_3: wInLWord |
| C00837/69 | L_ConvUnitsToIncr_3: wInHWord |
| C00837/70 | L_Curve_1: nIn_a |
| C00837/71 | L_ConvW_1: wIn |
| C00837/72 | L_ConvW_2: wIn |
| C00837/73 | L_ConvW_3: wIn |
| C00837/74 | L_ConvW_4: wIn |
| C00837/75 | L_MckCtrlInterface_1: wPosSetHW |
| C00837/76 | L_MckCtrlInterface_1: wPosSetLW |
| C00837/77 | L_PhaseDiff_1: nIn_v |
| C00837/78 | L_PhaseDiff_2: nIn_v |
| C00837/79 | L_SRFG_1: nIn_a |
| C00837/80 | L_SRFG_2: nIn_a |
| C00837/81 | L_SRFG_1: nSet_a |
| C00837/82 | L_SRFG_2: nSet_a |
| C00837/83 | L_SignalSwitch_1: wIn1 |
| C00837/84 | L_SignalSwitch_2: wIn1 |
| C00837/85 | L_SignalSwitch_3: wIn1 |
| C00837/86 | L_SignalSwitch_4: wIn1 |
| C00837/87 | L_SignalSwitch_1: wIn2 |
| C00837/88 | L_SignalSwitch_2: wIn2 |
| C00837/89 | L_SignalSwitch_3: wIn2 |
| C00837/90 | L_SignalSwitch_4: wIn2 |
| C00837/91 | L_Odometer_1: nInSpeed_v |
| C00837/92 | L_CalcDiameter_1: wDMax |

| Parameter \| Name: C00837 \| 16-bit inputs [rpm] (Set2) | Data type: INTEGER_16 Index: $23738_d$ = $5CBA_h$ |
|---|---|
| C00837/93 | L_CalcDiameter_1: wDMin |
| C00837/94 | L_CalcDiameter_1: wVMax |
| C00837/95 | L_CalcDiameter_1: nVLine_a |
| C00837/96 | L_CalcDiameter_1: nMotorSpeedAct_v |
| C00837/97 | L_CalcDiameter_1: wGearNum |
| C00837/98 | L_CalcDiameter_1: wGearDenom |
| C00837/99 | L_CalcDiameter_1: nSetD_a |
| C00837/100 | L_ProcessCtrl_1: nVpAdapt_a |
| C00837/101 | L_ProcessCtrl_1: nSet_a |
| C00837/102 | L_ProcessCtrl_1: nAct_a |
| C00837/103 | L_ProcessCtrl_1: nRTimeAdapt_a |
| C00837/104 | L_ProcessCtrl_1: nInfluence_a |
| C00837/105 | L_PhiIntegrator_1: nIn_v |
| C00837/106 | L_PhiIntegrator_1: wGearNum |
| C00837/107 | L_PhiIntegrator_1: wGearDenom |
| C00837/108 | L_SwitchPoint_1: nActSpeed_v |
| C00837/109 | L_PhiIntegrator_1: nSpeedAdd_v |
| C00837/110 | L_DFSET_1: nSpeedTrim_v |
| C00837/111 | L_DFSET_1: nSpeedTrim_a |
| C00837/112 | L_DFSET_1: wGearNum |
| C00837/113 | L_DFSET_1: wGainNum |
| C00837/114 | L_DFSET_1: nSet_v |
| C00837/115 | L_DFSET_1: wGainDenom |
| C00837/116 | L_DFSET_1: wGearDenom |
| C00837/117 | L_DFSET_1: nPositionTrimming |
| C00837/118 | L_DFSET_1: nNAct_v |
| C00837/119 | L_DFRFG_1: nIn_v |
| C00837/120 | L_GearComp_1: nTorque_a |
| C00837/121 | L_ConvAP_1: nIn_a |
| C00837/122 | L_ConvAP_2: nIn_a |
| C00837/123 | L_ConvAP_3: nIn_a |
| C00837/124 | L_ConvX_1: nIn_a |
| C00837/125 | L_ConvX_1: nNum |
| C00837/126 | L_ConvX_1: wDenom |
| C00837/127 | L_ConvX_2: nIn_a |
| C00837/128 | L_ConvX_2: nNum |
| C00837/129 | L_ConvX_2: wDenom |
| C00837/130 | L_ConvX_3: nIn_a |
| C00837/131 | L_ConvX_3: nNum |
| C00837/132 | L_ConvX_3: wDenom |
| C00837/133 | L_ConvPP_1: nNum |
| C00837/134 | L_ConvPP_1: wDenom |
| C00837/135 | L_ConvPP_2: nNum |
| C00837/136 | L_ConvPP_2: wDenom |

| Parameter \| Name: **C00837 \| 16-bit inputs [rpm] (Set2)** | Data type: INTEGER_16 Index: $23738_d$ = $5CBA_h$ |
|---|---|
| C00837/137 | L_ConvPP_3: nNum |
| C00837/138 | L_ConvPP_3: wDenom |
| C00837/139 | L_Curve_2: nIn_a |
| C00837/140 | L_Curve_3: nIn_a |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00838**

| Parameter \| Name: **C00838 \| Binary inputs (Set2)** | | Data type: UNSIGNED_8 Index: $23737_d$ = $5CB9_h$ |
|---|---|---|
| Display of the signal status of the binary inputs of different blocks<br>• In addition to the parameter C00833. | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | | **Info** |
| C00838/1 | | L_And5_1: bIn1 |
| C00838/2 | | L_And5_1: bIn2 |
| C00838/3 | | L_And5_1: bIn3 |
| C00838/4 | | L_And5_1: bIn4 |
| C00838/5 | | L_And5_1: bIn5 |
| C00838/6 | | L_And5_2: bIn1 |
| C00838/7 | | L_And5_2: bIn2 |
| C00838/8 | | L_And5_2: bIn3 |
| C00838/9 | | L_And5_2: bIn4 |
| C00838/10 | | L_And5_2: bIn5 |
| C00838/11 | | L_AnalogSwitch_4: bSet |
| C00838/12 | | L_AnalogSwitch_5: bSet |
| C00838/13 | | L_DFlipFlop_2: bD |
| C00838/14 | | L_DFlipFlop_2: bClk |
| C00838/15 | | L_DFlipFlop_2: bClr |
| C00838/16 | | L_DigitalDelay_2: bIn |
| C00838/17 | | L_DigitalDelay_3: bIn |
| C00838/18 | | L_PhaseIntK_1: bLoad |
| C00838/19 | | L_PhaseIntK_2: bLoad |
| C00838/20 | | L_DigitalLogic_2: bIn1 |
| C00838/21 | | L_DigitalLogic_2: bIn2 |
| C00838/22 | | L_DigitalLogic_2: bIn3 |
| C00838/23 | | L_DigitalLogic5_1: bIn1 |
| C00838/24 | | L_DigitalLogic5_1: bIn2 |
| C00838/25 | | L_DigitalLogic5_1: bIn3 |
| C00838/26 | | L_DigitalLogic5_1: bIn4 |
| C00838/27 | | L_DigitalLogic5_1: bIn5 |
| C00838/28 | | L_DigitalLogic5_2: bIn1 |
| C00838/29 | | L_DigitalLogic5_2: bIn2 |
| C00838/30 | | L_DigitalLogic5_2: bIn3 |

| Parameter \| Name: **C00838 \| Binary inputs (Set2)** | Data type: UNSIGNED_8 Index: $23737_d$ = $5CB9_h$ |
|---|---|
| C00838/31 | L_DigitalLogic5_2: bIn4 |
| C00838/32 | L_DigitalLogic5_2: bIn5 |
| C00838/33 | L_NLim_2: bEnable |
| C00838/34 | L_Or5_1: bIn1 |
| C00838/35 | L_Or5_1: bIn2 |
| C00838/36 | L_Or5_1: bIn3 |
| C00838/37 | L_Or5_1: bIn4 |
| C00838/38 | L_Or5_1: bIn5 |
| C00838/39 | L_Or5_2: bIn1 |
| C00838/40 | L_Or5_2: bIn2 |
| C00838/41 | L_Or5_2: bIn3 |
| C00838/42 | L_Or5_2: bIn4 |
| C00838/43 | L_Or5_2: bIn5 |
| C00838/44 | L_Not_4: bIn |
| C00838/45 | L_Not_5: bIn |
| C00838/46 | L_Not_6: bIn |
| C00838/47 | L_Not_7: bIn |
| C00838/48 | L_RSFlipFlop_1: bSet |
| C00838/49 | L_RSFlipFlop_1: bReset |
| C00838/50 | L_RSFlipFlop_2: bSet |
| C00838/51 | L_RSFlipFlop_2: bReset |
| C00838/52 | L_SampleHold_1: bLoad |
| C00838/53 | L_SampleHold_2: bLoad |
| C00838/54 | L_Counter_2: bClkUp |
| C00838/55 | L_Counter_2: bClkDown |
| C00838/56 | L_Counter_2: bLoad |
| C00838/57 | L_Counter_3: bClkUp |
| C00838/58 | L_Counter_3: bClkDown |
| C00838/59 | L_Counter_3: bLoad |
| C00838/60 | L_Transient_5: bIn |
| C00838/61 | L_Transient_6: bIn |
| C00838/62 | L_Transient_7: bIn |
| C00838/63 | L_Transient_8: bIn |
| C00838/64 | L_Counter_1: bCountUp |
| C00838/65 | L_Counter_1: bCountDown |
| C00838/66 | L_Counter_1: bLoad |
| C00838/67 | L_ConvBitsToWord_1: bBit0 |
| C00838/68 | L_ConvBitsToWord_1: bBit1 |
| C00838/69 | L_ConvBitsToWord_1: bBit2 |
| C00838/70 | L_ConvBitsToWord_1: bBit3 |
| C00838/71 | L_ConvBitsToWord_1: bBit4 |
| C00838/72 | L_ConvBitsToWord_1: bBit5 |
| C00838/73 | L_ConvBitsToWord_1: bBit6 |
| C00838/74 | L_ConvBitsToWord_1: bBit7 |

| Parameter \| Name: C00838 \| Binary inputs (Set2) | Data type: UNSIGNED_8 Index: $23737_d$ = $5CB9_h$ |
|---|---|
| C00838/75 | L_ConvBitsToWord_1: bBit8 |
| C00838/76 | L_ConvBitsToWord_1: bBit9 |
| C00838/77 | L_ConvBitsToWord_1: bBit10 |
| C00838/78 | L_ConvBitsToWord_1: bBit11 |
| C00838/79 | L_ConvBitsToWord_1: bBit12 |
| C00838/80 | L_ConvBitsToWord_1: bBit13 |
| C00838/81 | L_ConvBitsToWord_1: bBit14 |
| C00838/82 | L_ConvBitsToWord_1: bBit15 |
| C00838/83 | L_ConvBitsToWord_2: bBit0 |
| C00838/84 | L_ConvBitsToWord_2: bBit1 |
| C00838/85 | L_ConvBitsToWord_2: bBit2 |
| C00838/86 | L_ConvBitsToWord_2: bBit3 |
| C00838/87 | L_ConvBitsToWord_2: bBit4 |
| C00838/88 | L_ConvBitsToWord_2: bBit5 |
| C00838/89 | L_ConvBitsToWord_2: bBit6 |
| C00838/90 | L_ConvBitsToWord_2: bBit7 |
| C00838/91 | L_ConvBitsToWord_2: bBit8 |
| C00838/92 | L_ConvBitsToWord_2: bBit9 |
| C00838/93 | L_ConvBitsToWord_2: bBit10 |
| C00838/94 | L_ConvBitsToWord_2: bBit11 |
| C00838/95 | L_ConvBitsToWord_2: bBit12 |
| C00838/96 | L_ConvBitsToWord_2: bBit13 |
| C00838/97 | L_ConvBitsToWord_2: bBit14 |
| C00838/98 | L_ConvBitsToWord_2: bBit15 |
| C00838/99 | L_ConvBitsToWord_3: bBit0 |
| C00838/100 | L_ConvBitsToWord_3: bBit1 |
| C00838/101 | L_ConvBitsToWord_3: bBit2 |
| C00838/102 | L_ConvBitsToWord_3: bBit3 |
| C00838/103 | L_ConvBitsToWord_3: bBit4 |
| C00838/104 | L_ConvBitsToWord_3: bBit5 |
| C00838/105 | L_ConvBitsToWord_3: bBit6 |
| C00838/106 | L_ConvBitsToWord_3: bBit7 |
| C00838/107 | L_ConvBitsToWord_3: bBit8 |
| C00838/108 | L_ConvBitsToWord_3: bBit9 |
| C00838/109 | L_ConvBitsToWord_3: bBit10 |
| C00838/110 | L_ConvBitsToWord_3: bBit11 |
| C00838/111 | L_ConvBitsToWord_3: bBit12 |
| C00838/112 | L_ConvBitsToWord_3: bBit13 |
| C00838/113 | L_ConvBitsToWord_3: bBit14 |
| C00838/114 | L_ConvBitsToWord_3: bBit15 |
| C00838/115 | MckCtrlInterface_1: bManJogPos |
| C00838/116 | MckCtrlInterface_1: bManJogNeg |
| C00838/117 | MckCtrlInterface_1: bManJogExecute2ndVel |
| C00838/118 | MckCtrlInterface_1: bReleaseLimitSwitch |

| Parameter \| Name: C00838 \| Binary inputs (Set2) | Data type: UNSIGNED_8 Index: $23737_d$ = $5CB9_h$ |
|---|---|
| C00838/119 | MckCtrlInterface_1: bHomingStartStop |
| C00838/120 | MckCtrlInterface_1: bHomingSetPos |
| C00838/121 | MckCtrlInterface_1: bHomingResetPos |
| C00838/122 | MckCtrlInterface_1: bEnableVelOverride |
| C00838/123 | MckCtrlInterface_1: bEnableAccOverride |
| C00838/124 | MckCtrlInterface_1: bDisableSShaping |
| C00838/125 | MckCtrlInterface_1: bPosExecute |
| C00838/126 | MckCtrlInterface_1: bPosExecuteFinish |
| C00838/127 | MckCtrlInterface_1: bPosDisableFollowProfile |
| C00838/128 | MckCtrlInterface_1: bPosStop |
| C00838/129 | MckCtrlInterface_1: bPosTeachSetPos |
| C00838/130 | MckCtrlInterface_1: bPosTeachActPos |
| C00838/131 | MckCtrlInterface_1: bProfileNo_1 |
| C00838/132 | MckCtrlInterface_1: bProfileNo_2 |
| C00838/133 | MckCtrlInterface_1: bProfileNo_4 |
| C00838/134 | MckCtrlInterface_1: bProfileNo_8 |
| C00838/135 | MckCtrlInterface_1: bOperationMode_1 |
| C00838/136 | MckCtrlInterface_1: bOperationMode_2 |
| C00838/137 | MckCtrlInterface_1: bOperationMode_4 |
| C00838/138 | MckCtrlInterface_1: bOperationMode_8 |
| C00838/139 | L_PhaseDiff_1: bEnable |
| C00838/140 | L_PhaseDiff_2: bEnable |
| C00838/141 | L_PhaseDiff_1: bReset |
| C00838/142 | L_PhaseDiff_2: bReset |
| C00838/143 | L_SRFG_1: bLoad |
| C00838/144 | L_SRFG_2: bLoad |
| C00838/145 | L_SignalSwitch_1: bSet |
| C00838/146 | L_SignalSwitch_2: bSet |
| C00838/147 | L_SignalSwitch_3: bSet |
| C00838/148 | L_SignalSwitch_4: bSet |
| C00838/149 | L_Odometer_1: bTriggerPulse |
| C00838/150 | L_Odometer_1: bReset |
| C00838/151 | L_FixSet_a_1: bSelect1 |
| C00838/152 | L_FixSet_a_1: bSelect2 |
| C00838/153 | L_FixSet_a_1: bSelect4 |
| C00838/154 | L_FixSet_a_1: bSelect8 |
| C00838/155 | L_FixSet_w_1: bSelect1 |
| C00838/156 | L_FixSet_w_1: bSelect2 |
| C00838/157 | L_FixSet_w_1: bSelect4 |
| C00838/158 | L_FixSet_w_1: bSelect8 |
| C00838/159 | L_FixSet_w_2: bSelect1 |
| C00838/160 | L_FixSet_w_2: bSelect2 |
| C00838/161 | L_FixSet_w_2: bSelect4 |
| C00838/162 | L_FixSet_w_2: bSelect8 |

| Parameter \| Name: C00838 \| Binary inputs (Set2) | Data type: UNSIGNED_8 Index: $23737_d$ = $5CB9_h$ |
|---|---|
| C00838/163 | L_CalcDiameter_1: bResetPos |
| C00838/164 | L_CalcDiameter_1: bHoldD |
| C00838/165 | L_CalcDiameter_1: bUnidirect |
| C00838/166 | L_CalcDiameter_1: bUnwind |
| C00838/167 | L_CalcDiameter_1: bLoadDiameter |
| C00838/168 | L_CalcDiameter_1: bCalcRef |
| C00838/169 | L_ProcessCtrl_1: bLoadAct |
| C00838/170 | L_ProcessCtrl_1: bIOff |
| C00838/171 | L_ProcessCtrl_1: bReset |
| C00838/172 | L_PhiIntegrator_1: bTPReceived |
| C00838/173 | L_PhiIntegrator_1: bReset |
| C00838/174 | L_PhiIntegrator_1: bLoad |
| C00838/175 | L_PosCtrlLin_1: bExecute |
| C00838/176 | L_PosCtrlLin_1: bSetPos0 |
| C00838/177 | L_PosCtrlLin_1: bPosMode |
| C00838/178 | L_PosCtrlLin_1: bEnable |
| C00838/179 | L_PosCtrlLin_2: bExecute |
| C00838/180 | L_PosCtrlLin_2: bSetPos0 |
| C00838/181 | L_PosCtrlLin_2: bPosMode |
| C00838/182 | L_PosCtrlLin_2: bEnable |
| C00838/183 | L_SwitchPoint_1: bDisable |
| C00838/184 | L_DFSET_1: bZeroPulse |
| C00838/185 | L_DFSET_1: bSetTPReceived |
| C00838/186 | L_DFSET_1: bActTPReceived |
| C00838/187 | L_DFSET_1: bSetActIntegrator |
| C00838/188 | L_DFSET_1: bResetAllIntegrators |
| C00838/189 | L_DFRFG_1: bSetTPReceived |
| C00838/190 | L_DFRFG_1: bRfg0 |
| C00838/191 | L_DFRFG_1: bRfgStop |
| C00838/192 | L_DFRFG_1: bReset |
| C00838/193 | L_ConvX_1: bInvers |
| C00838/194 | L_ConvX_2: bInvers |
| C00838/195 | L_ConvX_3: bInvers |
| C00838/196 | L_ConvPP_1: bAct |
| C00838/197 | L_ConvPP_2: bAct |
| C00838/198 | L_ConvPP_3: bAct |
| C00838/199 | L_SignalSwitch32_1: bSet |
| C00838/200 | L_SignalSwitch32_2: bSet |
| C00838/201 | L_SignalSwitch32_3: bSet |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

## C00839

| Parameter \| Name: C00839 \| 32-bit inputs [incr] (Set2) | | | Data type: INTEGER_32 Index: $23736_d$ = $5CB8_h$ |
|---|---|---|---|
| Display in [increments] of 32 bit input values of various blocks<br>• In addition to the parameter C00834. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -2147483647 | Incr. | 2147483647 | |

| Subcodes | Info |
|---|---|
| C00839/1 | L_ComparePhi_1: dnIn1_p |
| C00839/2 | L_ComparePhi_1: dnIn2_p |
| C00839/3 | L_ComparePhi_2: dnIn1_p |
| C00839/4 | L_ComparePhi_2: dnIn2_p |
| C00839/5 | L_ComparePhi_3: dnIn1_p |
| C00839/6 | L_ComparePhi_3: dnIn2_p |
| C00839/7 | L_ComparePhi_4: dnIn1_p |
| C00839/8 | L_ComparePhi_4: dnIn2_p |
| C00839/9 | L_ComparePhi_5: dnIn1_p |
| C00839/10 | L_ComparePhi_5: dnIn2_p |
| C00839/11 | L_ArithmetikPhi_1: dnIn1_p |
| C00839/12 | L_ArithmetikPhi_1: dnIn2_p |
| C00839/13 | L_ArithmetikPhi_2: dnIn1_p |
| C00839/14 | L_ArithmetikPhi_2: dnIn2_p |
| C00839/15 | L_ArithmetikPhi_3: dnIn1_p |
| C00839/16 | L_ArithmetikPhi_3: dnIn2_p |
| C00839/17 | L_GainOffsetPhiP_1: dnIn_p |
| C00839/18 | L_GainOffsetPhiP_2: dnIn_p |
| C00839/19 | L_LimitPhi_1: dnIn_p |
| C00839/20 | L_LimitPhi_2: dnIn_p |
| C00839/21 | L_LimitPhi_3: dnIn_p |
| C00839/22 | L_OffsetGainPhiP_1: dnIn_p |
| C00839/23 | L_OffsetGainPhiP_2: dnIn_p |
| C00839/24 | L_PhaseIntK_1: dnSet_p |
| C00839/25 | L_PhaseIntK_2: dnSet_p |
| C00839/26 | L_Mux_1: dnInput1_p |
| C00839/27 | L_Mux_1: dnInput2_p |
| C00839/28 | L_Mux_1: dnInput3_p |
| C00839/29 | L_Mux_1: dnInput4_p |
| C00839/30 | L_Mux_1: dnInput5_p |
| C00839/31 | L_Mux_1: dnInput6_p |
| C00839/32 | L_Mux_1: dnInput7_p |
| C00839/33 | L_Mux_1: dnInput8_p |
| C00839/34 | L_SOrt_1: dnInput_p |
| C00839/35 | L_ConvDIntToWords_1: dnInput_p |
| C00839/36 | L_ConvDIntToWords_2: dnInput_p |
| C00839/37 | L_ConvDIntToWords_3: dnInput_p |
| C00839/38 | L_MckCtrlInterface_1: dnPosSetIn_p |
| C00839/39 | L_PhaseDiff_1: dnSet_p |

| Parameter \| Name: **C00839 \| 32-bit inputs [incr] (Set2)** | Data type: INTEGER_32 Index: $23736_d$ = $5CB8_h$ |
|---|---|
| C00839/40 | L_PhaseDiff_2: dnSet_p |
| C00839/41 | L_PhaseDiff_1: dnAdd_p |
| C00839/42 | L_PhaseDiff_2: dnAdd_p |
| C00839/43 | L_MckStateInterface_1: dnPosIn_p |
| C00839/44 | L_Odometer_1: dnInPosition_p |
| C00839/45 | L_PhiIntegrator_1: dnTPPosition_p |
| C00839/46 | L_PhiIntegrator_1: dnLoadVal_p |
| C00839/47 | L_PosCtrlLin_1: dnSet_p |
| C00839/48 | L_PosCtrlLin_1: dnAct_p |
| C00839/49 | L_PosCtrlLin_2: dnSet_p |
| C00839/50 | L_PosCtrlLin_2: dnAct_p |
| C00839/51 | L_SwitchPoint_1: dnActPos_p |
| C00839/52 | L_SwitchPoint_1: dnSwitchPoint1_p |
| C00839/53 | L_SwitchPoint_1: dn2ndPoint_Size1_p |
| C00839/54 | L_SwitchPoint_1: dnSwitchPoint2_p |
| C00839/55 | L_SwitchPoint_1: dn2ndPoint_Size2_p |
| C00839/56 | L_SwitchPoint_1: dnSwitchPoint3_p |
| C00839/57 | L_SwitchPoint_1: dn2ndPoint_Size3_p |
| C00839/58 | L_SwitchPoint_1: dnSwitchPoint4_p |
| C00839/59 | L_SwitchPoint_1: dn2ndPoint_Size4_p |
| C00839/60 | L_DFSET_1: dnSetTPPos_p |
| C00839/61 | L_DFSET_1: dnActTPPos_p |
| C00839/62 | L_DFSET_1: dnPosOffset |
| C00839/63 | L_DFRFG_1: dnSetTPPos_p |
| C00839/64 | L_GearComp_1: dnPhiIn_p |
| C00839/65 | L_ConvPA_1: dnIn_p |
| C00839/66 | L_ConvPA_2: dnIn_p |
| C00839/67 | L_ConvPA_3: dnIn_p |
| C00839/68 | L_ConvPP_1: dnIn_p |
| C00839/69 | L_ConvPP_2: dnIn_p |
| C00839/70 | L_ConvPP_3: dnIn_p |
| C00839/71 | L_SignalSwitch32_1: dnIn1 |
| C00839/72 | L_SignalSwitch32_1: dnIn2 |
| C00839/73 | L_SignalSwitch32_2: dnIn1 |
| C00839/74 | L_SignalSwitch32_2: dnIn2 |
| C00839/75 | L_SignalSwitch32_3: dnIn1 |
| C00839/76 | L_SignalSwitch32_3: dnIn2 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

**C00840**

| Parameter \| Name:<br>**C00840 \| 16-bit inputs I/O level [%]** | | | Data type: INTEGER_16<br>Index: $23735_d$ = $5CB7_h$ |
|---|---|---|---|
| Display in percent of 16 bit input values of various blocks of the I/O level | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |

| Subcodes | Info |
|---|---|
| C00840/1 | LS_AnalogOutput: nOut1_a (V) |
| C00840/2 | LP_CanOut1: wState |
| C00840/3 | LP_CanOut1: wOut2 |
| C00840/4 | LP_CanOut1: wOut3 |
| C00840/5 | LP_CanOut1: wOut4 |
| C00840/6 | LP_CanOut2: wOut1 |
| C00840/7 | LP_CanOut2: wOut2 |
| C00840/8 | LP_CanOut2: wOut3 |
| C00840/9 | LP_CanOut2: wOut4 |
| C00840/10 | LP_CanOut3: wOut1 |
| C00840/11 | LP_CanOut3: wOut2 |
| C00840/12 | LP_CanOut3: wOut3 |
| C00840/13 | LP_CanOut3: wOut4 |
| C00840/14 | LS_DisFree_a: nDis1_a |
| C00840/15 | LS_DisFree_a: nDis2_a |
| C00840/16 | LS_DisFree_a: nDis3_a |
| C00840/17 | LS_DisFree_a: nDis4_a |
| C00840/18 | LS_DisFree: wDis1 |
| C00840/19 | LS_DisFree: wDis2 |
| C00840/20 | LS_DisFree: wDis3 |
| C00840/21 | LS_DisFree: wDis4 |
| C00840/22 | LP_MciOut: wState |
| C00840/23 | LP_MciOut: wOut2 |
| C00840/24 | LP_MciOut: wOut3 |
| C00840/25 | LP_MciOut: wOut4 |
| C00840/26 | LP_MciOut: wOut5 |
| C00840/27 | LP_MciOut: wOut6 |
| C00840/28 | LP_MciOut: wOut7 |
| C00840/29 | LP_MciOut: wOut8 |
| C00840/30 | LP_MciOut: wOut9 |
| C00840/31 | LP_MciOut: wOut10 |
| C00840/32 | LP_MciOut: wOut11 |
| C00840/33 | LP_MciOut: wOut12 |
| C00840/34 | LP_MciOut: wOut13 |
| C00840/35 | LP_MciOut: wOut14 |
| C00840/36 | LP_MciOut: wOut15 |
| C00840/37 | LP_MciOut: wOut16 |
| C00840/38 | LS_AnalogOutput: nOut2_a (V) |
| C00840/39 | LS_AnalogOutput: nOut1_a (I) |
| C00840/40 | LS_AnalogOutput: nOut2_a (I) |

| Parameter \| Name: **C00840 \| 16-bit inputs I/O level [%]** | Data type: INTEGER_16 Index: $23735_d$ = $5CB7_h$ |
|---|---|
| C00840/41 | LS_DisFree_a: nDis5_a |
| C00840/42 | LS_DisFree_a: nDis6_a |
| C00840/43 | LS_DisFree_a: nDis7_a |
| C00840/44 | LS_DisFree_a: nDis8_a |
| C00840/45 | LS_DisFree: wDis5 |
| C00840/46 | LS_DisFree: wDis6 |
| C00840/47 | LS_DisFree: wDis7 |
| C00840/48 | LS_DisFree: wDis8 |
| C00840/49 | LS_ParReadWrite_1: wParIndex |
| C00840/50 | LS_ParReadWrite_1: wParSubindex |
| C00840/51 | LS_ParReadWrite_1: wInHWord |
| C00840/52 | LS_ParReadWrite_1: wInLWord |
| C00840/53 | LS_ParReadWrite_2: wParIndex |
| C00840/54 | LS_ParReadWrite_2: wParSubindex |
| C00840/55 | LS_ParReadWrite_2: wInHWord |
| C00840/56 | LS_ParReadWrite_2: wInLWord |
| C00840/57 | LS_ParReadWrite_3: wParIndex |
| C00840/58 | LS_ParReadWrite_3: wParSubindex |
| C00840/59 | LS_ParReadWrite_3: wInHWord |
| C00840/60 | LS_ParReadWrite_3: wInLWord |
| C00840/61 | LS_ParReadWrite_4: wParIndex |
| C00840/62 | LS_ParReadWrite_4: wParSubindex |
| C00840/63 | LS_ParReadWrite_4: wInHWord |
| C00840/64 | LS_ParReadWrite_4: wInLWord |
| C00840/65 | LS_ParReadWrite_5: wParIndex |
| C00840/66 | LS_ParReadWrite_5: wParSubindex |
| C00840/67 | LS_ParReadWrite_5: wInHWord |
| C00840/68 | LS_ParReadWrite_5: wInLWord |
| C00840/69 | LS_ParReadWrite_6: wParIndex |
| C00840/70 | LS_ParReadWrite_6: wParSubindex |
| C00840/71 | LS_ParReadWrite_6: wInHWord |
| C00840/72 | LS_ParReadWrite_6: wInLWord |
| C00840/73 | Reserved |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C00841**

| Parameter \| Name: **C00841 \| 16-bit inputs I/O level** | | | Data type: UNSIGNED_16 Index: $23734_d$ = $5CB6_h$ |
|---|---|---|---|
| Decimal/hexadecimal/bit coded display of 16 bit input values of various blocks of the I/O level | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |

| Parameter \| Name: C00841 \| 16-bit inputs I/O level | Data type: UNSIGNED_16 Index: $23734_d$ = $5CB6_h$ |
|---|---|
| **Subcodes** | **Info** |
| C00841/1 | LS_AnalogOutput: nOut1_a (V) |
| C00841/2 | LP_CanOut1: wState |
| C00841/3 | LP_CanOut1: wOut2 |
| C00841/4 | LP_CanOut1: wOut3 |
| C00841/5 | LP_CanOut1: wOut4 |
| C00841/6 | LP_CanOut2: wOut1 |
| C00841/7 | LP_CanOut2: wOut2 |
| C00841/8 | LP_CanOut2: wOut3 |
| C00841/9 | LP_CanOut2: wOut4 |
| C00841/10 | LP_CanOut3: wOut1 |
| C00841/11 | LP_CanOut3: wOut2 |
| C00841/12 | LP_CanOut3: wOut3 |
| C00841/13 | LP_CanOut3: wOut4 |
| C00841/14 | LS_DisFree_a: nDis1_a |
| C00841/15 | LS_DisFree_a: nDis2_a |
| C00841/16 | LS_DisFree_a: nDis3_a |
| C00841/17 | LS_DisFree_a: nDis4_a |
| C00841/18 | LS_DisFree: wDis1 |
| C00841/19 | LS_DisFree: wDis2 |
| C00841/20 | LS_DisFree: wDis3 |
| C00841/21 | LS_DisFree: wDis4 |
| C00841/22 | LP_MciOut: wState |
| C00841/23 | LP_MciOut: wOut2 |
| C00841/24 | LP_MciOut: wOut3 |
| C00841/25 | LP_MciOut: wOut4 |
| C00841/26 | LP_MciOut: wOut5 |
| C00841/27 | LP_MciOut: wOut6 |
| C00841/28 | LP_MciOut: wOut7 |
| C00841/29 | LP_MciOut: wOut8 |
| C00841/30 | LP_MciOut: wOut9 |
| C00841/31 | LP_MciOut: wOut10 |
| C00841/32 | LP_MciOut: wOut11 |
| C00841/33 | LP_MciOut: wOut12 |
| C00841/34 | LP_MciOut: wOut13 |
| C00841/35 | LP_MciOut: wOut14 |
| C00841/36 | LP_MciOut: wOut15 |
| C00841/37 | LP_MciOut: wOut16 |
| C00841/38 | LS_AnalogOutput: nOut2_a (V) |
| C00841/39 | LS_AnalogOutput: nOut1_a (I) |
| C00841/40 | LS_AnalogOutput: nOut2_a (I) |
| C00841/41 | LS_DisFree_a: nDis5_a |
| C00841/42 | LS_DisFree_a: nDis6_a |
| C00841/43 | LS_DisFree_a: nDis7_a |

| Parameter \| Name: **C00841 \| 16-bit inputs I/O level** | Data type: UNSIGNED_16 Index: $23734_d$ = $5CB6_h$ |
|---|---|
| C00841/44 | LS_DisFree_a: nDis8_a |
| C00841/45 | LS_DisFree: wDis5 |
| C00841/46 | LS_DisFree: wDis6 |
| C00841/47 | LS_DisFree: wDis7 |
| C00841/48 | LS_DisFree: wDis8 |
| C00841/49 | LS_ParReadWrite_1: wParIndex |
| C00841/50 | LS_ParReadWrite_1: wParSubindex |
| C00841/51 | LS_ParReadWrite_1: wInHWord |
| C00841/52 | LS_ParReadWrite_1: wInLWord |
| C00841/53 | LS_ParReadWrite_2: wParIndex |
| C00841/54 | LS_ParReadWrite_2: wParSubindex |
| C00841/55 | LS_ParReadWrite_2: wInHWord |
| C00841/56 | LS_ParReadWrite_2: wInLWord |
| C00841/57 | LS_ParReadWrite_3: wParIndex |
| C00841/58 | LS_ParReadWrite_3: wParSubindex |
| C00841/59 | LS_ParReadWrite_3: wInHWord |
| C00841/60 | LS_ParReadWrite_3: wInLWord |
| C00841/61 | LS_ParReadWrite_4: wParIndex |
| C00841/62 | LS_ParReadWrite_4: wParSubindex |
| C00841/63 | LS_ParReadWrite_4: wInHWord |
| C00841/64 | LS_ParReadWrite_4: wInLWord |
| C00841/65 | LS_ParReadWrite_5: wParIndex |
| C00841/66 | LS_ParReadWrite_5: wParSubindex |
| C00841/67 | LS_ParReadWrite_5: wInHWord |
| C00841/68 | LS_ParReadWrite_5: wInLWord |
| C00841/69 | LS_ParReadWrite_6: wParIndex |
| C00841/70 | LS_ParReadWrite_6: wParSubindex |
| C00841/71 | LS_ParReadWrite_6: wInHWord |
| C00841/72 | LS_ParReadWrite_6: wInLWord |
| C00841/73 | Reserved |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | |

**C00843**

| Parameter \| Name: **C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8 Index: $23732_d$ = $5CB4_h$ |
|---|---|
| Display of the signal status of the binary inputs of different I/O level blocks | |
| **Selection list** | |
| 0 False | |
| 1 True | |
| **Subcodes** | **Info** |
| C00843/1 | LS_DigitalOutput: bRelay |
| C00843/2 | LS_DigitalOutput: bOut1 |
| C00843/3 | LS_DigitalInput: bCountIn1_Reset |
| C00843/4 | LS_DigitalInput: bCountIn1_LoadStartValue |
| C00843/5 | LP_CanOut1: bState_B0 |

| Parameter \| Name: **C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8 Index: $23732_d$ = $5CB4_h$ |
|---|---|
| C00843/6 | LP_CanOut1: bState_B1 |
| C00843/7 | LP_CanOut1: bState_B2 |
| C00843/8 | LP_CanOut1: bState_B3 |
| C00843/9 | LP_CanOut1: bState_B4 |
| C00843/10 | LP_CanOut1: bState_B5 |
| C00843/11 | LP_CanOut1: bState_B6 |
| C00843/12 | LP_CanOut1: bState_B7 |
| C00843/13 | LP_CanOut1: bState_B8 |
| C00843/14 | LP_CanOut1: bState_B9 |
| C00843/15 | LP_CanOut1: bState_B10 |
| C00843/16 | LP_CanOut1: bState_B11 |
| C00843/17 | LP_CanOut1: bState_B12 |
| C00843/18 | LP_CanOut1: bState_B13 |
| C00843/19 | LP_CanOut1: bState_B14 |
| C00843/20 | LP_CanOut1: bState_B15 |
| C00843/21 | LS_DisFree_b: bDis1 |
| C00843/22 | LS_DisFree_b: bDis2 |
| C00843/23 | LS_DisFree_b: bDis3 |
| C00843/24 | LS_DisFree_b: bDis4 |
| C00843/25 | LS_DisFree_b: bDis5 |
| C00843/26 | LS_DisFree_b: bDis6 |
| C00843/27 | LS_DisFree_b: bDis7 |
| C00843/28 | LS_DisFree_b: bDis8 |
| C00843/29 | LP_CanOut2: bOut1_B0 |
| C00843/30 | LP_CanOut2: bOut1_B1 |
| C00843/31 | LP_CanOut2: bOut1_B2 |
| C00843/32 | LP_CanOut2: bOut1_B3 |
| C00843/33 | LP_CanOut2: bOut1_B4 |
| C00843/34 | LP_CanOut2: bOut1_B5 |
| C00843/35 | LP_CanOut2: bOut1_B6 |
| C00843/36 | LP_CanOut2: bOut1_B7 |
| C00843/37 | LP_CanOut2: bOut1_B8 |
| C00843/38 | LP_CanOut2: bOut1_B9 |
| C00843/39 | LP_CanOut2: bOut1_B10 |
| C00843/40 | LP_CanOut2: bOut1_B11 |
| C00843/41 | LP_CanOut2: bOut1_B12 |
| C00843/42 | LP_CanOut2: bOut1_B13 |
| C00843/43 | LP_CanOut2: bOut1_B14 |
| C00843/44 | LP_CanOut2: bOut1_B15 |
| C00843/45 | LP_CanOut3: bOut1_B0 |
| C00843/46 | LP_CanOut3: bOut1_B1 |
| C00843/47 | LP_CanOut3: bOut1_B2 |
| C00843/48 | LP_CanOut3: bOut1_B3 |
| C00843/49 | LP_CanOut3: bOut1_B4 |

| Parameter \| Name: **C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8 Index: $23732_d$ = $5CB4_h$ |
|---|---|
| C00843/50 | LP_CanOut3: bOut1_B5 |
| C00843/51 | LP_CanOut3: bOut1_B6 |
| C00843/52 | LP_CanOut3: bOut1_B7 |
| C00843/53 | LP_CanOut3: bOut1_B8 |
| C00843/54 | LP_CanOut3: bOut1_B9 |
| C00843/55 | LP_CanOut3: bOut1_B10 |
| C00843/56 | LP_CanOut3: bOut1_B11 |
| C00843/57 | LP_CanOut3: bOut1_B12 |
| C00843/58 | LP_CanOut3: bOut1_B13 |
| C00843/59 | LP_CanOut3: bOut1_B14 |
| C00843/60 | LP_CanOut3: bOut1_B15 |
| C00843/61 | LP_MciOut: bState_B0 |
| C00843/62 | LP_MciOut: bState_B1 |
| C00843/63 | LP_MciOut: bState_B2 |
| C00843/64 | LP_MciOut: bState_B3 |
| C00843/65 | LP_MciOut: bState_B4 |
| C00843/66 | LP_MciOut: bState_B5 |
| C00843/67 | LP_MciOut: bState_B6 |
| C00843/68 | LP_MciOut: bState_B7 |
| C00843/69 | LP_MciOut: bState_B8 |
| C00843/70 | LP_MciOut: bState_B9 |
| C00843/71 | LP_MciOut: bState_B10 |
| C00843/72 | LP_MciOut: bState_B11 |
| C00843/73 | LP_MciOut: bState_B12 |
| C00843/74 | LP_MciOut: bState_B13 |
| C00843/75 | LP_MciOut: bState_B14 |
| C00843/76 | LP_MciOut: bState_B15 |
| C00843/77 | LP_MciOut: bOut2_B0 |
| C00843/78 | LP_MciOut: bOut2_B1 |
| C00843/79 | LP_MciOut: bOut2_B2 |
| C00843/80 | LP_MciOut: bOut2_B3 |
| C00843/81 | LP_MciOut: bOut2_B4 |
| C00843/82 | LP_MciOut: bOut2_B5 |
| C00843/83 | LP_MciOut: bOut2_B6 |
| C00843/84 | LP_MciOut: bOut2_B7 |
| C00843/85 | LP_MciOut: bOut2_B8 |
| C00843/86 | LP_MciOut: bOut2_B9 |
| C00843/87 | LP_MciOut: bOut2_B10 |
| C00843/88 | LP_MciOut: bOut2_B11 |
| C00843/89 | LP_MciOut: bOut2_B12 |
| C00843/90 | LP_MciOut: bOut2_B13 |
| C00843/91 | LP_MciOut: bOut2_B14 |
| C00843/92 | LP_MciOut: bOut2_B15 |
| C00843/93 | LS_SetError_1: bSetError1 |

| Parameter \| Name: **C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8 Index: $23732_d$ = $5CB4_h$ |
|---|---|
| C00843/94 | LS_SetError_1: bSetError2 |
| C00843/95 | LS_SetError_1: bSetError3 |
| C00843/96 | LS_SetError_1: bSetError4 |
| C00843/97 | LS_DigitalInput: bCountIn6_Reset |
| C00843/98 | LS_DigitalInput: bCountIn6_LoadStartValue |
| C00843/99 | LS_DigitalOutput: bOut2 |
| C00843/100 | LS_DigitalOutput: bOut3 |
| C00843/101 | LS_DigitalOutput: bOut HighCurrent |
| C00843/102 | LS_DisFree_b: bDis9 |
| C00843/103 | LS_DisFree_b: bDis10 |
| C00843/104 | LS_DisFree_b: bDis11 |
| C00843/105 | LS_DisFree_b: bDis12 |
| C00843/106 | LS_DisFree_b: bDis13 |
| C00843/107 | LS_DisFree_b: bDis14 |
| C00843/108 | LS_DisFree_b: bDis15 |
| C00843/109 | LS_DisFree_b: bDis16 |
| C00843/110 | Reserved |
| C00843/111 | LS_ParReadWrite_1: bExecute |
| C00843/112 | LS_ParReadWrite_1: bReadWrite |
| C00843/113 | LS_ParReadWrite_2: bExecute |
| C00843/114 | LS_ParReadWrite_2: bReadWrite |
| C00843/115 | LS_ParReadWrite_3: bExecute |
| C00843/116 | LS_ParReadWrite_3: bReadWrite |
| C00843/117 | LS_ParReadWrite_4: bExecute |
| C00843/118 | LS_ParReadWrite_4: bReadWrite |
| C00843/119 | LS_ParReadWrite_5: bExecute |
| C00843/120 | LS_ParReadWrite_5: bReadWrite |
| C00843/121 | LS_ParReadWrite_6: bExecute |
| C00843/122 | LS_ParReadWrite_6: bReadWrite |
| C00843/123 | LS_WriteParamList: bExecute |
| C00843/124 | LS_WriteParamList: bSelectWriteValue_1 |
| C00843/125 | LS_WriteParamList: bSelectWriteValue_2 |
| C00843/126 | LS_CANManagement: bResetNode |
| C00843/127 | LS_CANManagement: bReInitCAN |
| C00843/128 | LS_DigitalInput: bPosIn12_Load |
| C00843/129 | Reserved |
| C00843/130 | Reserved |
| C00843/131 | Reserved |
| C00843/132 | Reserved |
| C00843/133 | Reserved |
| C00843/134 | Reserved |
| C00843/135 | Reserved |
| C00843/136 | Reserved |
| C00843/137 | Reserved |

| Parameter \| Name: **C00843 \| Binary inputs I/O level** | Data type: UNSIGNED_8 Index: $23732_d = 5CB4_h$ |
|---|---|
| C00843/138 | Reserved |
| C00843/139 | Reserved |
| C00843/140 | Reserved |
| C00843/141 | Reserved |
| C00843/142 | Reserved |
| C00843/143 | Reserved |
| C00843/144 | Reserved |
| C00843/145 | Reserved |
| C00843/146 | Reserved |
| C00843/147 | LS_TouchProbe: bDisableTPDigIn3 |
| C00843/148 | LS_TouchProbe: bDisableTPDigIn4 |
| C00843/149 | LS_TouchProbe: bDisableTPDigIn5 |
| C00843/150 | LS_TouchProbe: bDisableTPDigIn6 |
| C00843/151 | LS_TouchProbe: bDisableTPDigIn7 |
| C00843/152 | LS_TouchProbe: bDisableTPDigIn3_Rising |
| C00843/153 | LS_TouchProbe: bDisableTPDigIn3_Falling |
| C00843/154 | LS_TouchProbe: bDisableTPDigIn4_Rising |
| C00843/155 | LS_TouchProbe: bDisableTPDigIn4_Falling |
| C00843/156 | LS_TouchProbe: bDisableTPDigIn5_Rising |
| C00843/157 | LS_TouchProbe: bDisableTPDigIn5_Falling |
| C00843/158 | LS_TouchProbe: bDisableTPDigIn3Window |
| C00843/159 | LS_TouchProbe: bDisableTPDigIn4Window |
| C00843/160 | LS_TouchProbe: bDisableTPDigIn5Window |
| C00843/161 | LS_AxisBusIO: bSetFail |
| C00843/162 | LS_AxisBusIO: bResetFail |
| C00843/163 | Reserved |
| C00843/164 | LS_MultiEncoder: bSetRef |
| C00843/165 | LS_RetainData: bSetRetain_1 |
| C00843/166 | LS_RetainData: bSetRetain_2 |
| C00843/167 | LS_RetainData: bSetRetain_3 |
| C00843/168 | LS_RetainData: bLoadParams |
| C00843/169 | LS_RetainData: bIn1 |
| C00843/170 | LS_RetainData: bIn2 |
| C00843/171 | LS_RetainData: bIn3 |
| C00843/172 | LS_RetainData: bIn4 |
| C00843/173 | LS_AxisBusAux: bReadWrite |
| C00843/174 | LS_AxisBusAux: bExecute |
| C00843/175 | LS_AxisBusAux: bStop |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

## C00844

| Parameter \| Name: **C00844 \| 32-bit inputs I/O level [incr]** | Data type: INTEGER_32 Index: $23731_d$ = $5CB3_h$ |
|---|---|
| Display of 32-bit input values of different I/O level blocks in [increments] | |
| **Display range** (min. value \| unit \| max. value) | |
| -2147483647 Incr. 2147483647 | |
| **Subcodes** | **Info** |
| C00844/1 | LS_DisFree_p: dnDis1_p |
| C00844/2 | LS_DisFree_p: dnDis2_p |
| C00844/3 | LS_DisFree_p: dnDis3_p |
| C00844/4 | LS_DisFree_p: dnDis4_p |
| C00844/5 | LS_DisFree_p: dnDis5_p |
| C00844/6 | LS_DisFree_p: dnDis6_p |
| C00844/7 | LS_DisFree_p: dnDis7_p |
| C00844/8 | LS_DisFree_p: dnDis8_p |
| C00844/9 | LP_CanOut1: dnOut34_p |
| C00844/10 | LP_CanOut2: dnOut34_p |
| C00844/11 | LP_CanOut3: dnOut34_p |
| C00844/12 | LP_MciOut: dnOut34_p |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | |

## C00866

| Parameter \| Name: **C00866 \| CAN input words** | Data type: UNSIGNED_16 Index: $23709_d$ = $5C9D_h$ |
|---|---|
| Display of the 16 bit input values of the CAN interface | ▶ System bus "CAN on board" |
| **Display area** (min. hex value \| max. hex value) | |
| 0x0000 0xFFFF | |
| **Value is bit-coded:** | |
| Bit 0 Active | |
| ... ... | |
| Bit 15 Active | |
| **Subcodes** | **Info** |
| C00866/1 | LP_CanIn1: wCtrl |
| C00866/2 | LP_CanIn1: wIn2 |
| C00866/3 | LP_CanIn1: wIn3 |
| C00866/4 | LP_CanIn1: wIn4 |
| C00866/5 | LP_CanIn2: wIn1 |
| C00866/6 | LP_CanIn2: wIn2 |
| C00866/7 | LP_CanIn2: wIn3 |
| C00866/8 | LP_CanIn2: wIn4 |
| C00866/9 | LP_CanIn3: wIn1 |
| C00866/10 | LP_CanIn3: wIn2 |
| C00866/11 | LP_CanIn3: wIn3 |
| C00866/12 | LP_CanIn3: wIn4 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | |

## C00868

| Parameter \| Name: C00868 \| CAN output words | | | Data type: UNSIGNED_16 Index: $23707_d$ = $5C9B_h$ |
|---|---|---|---|
| Display of the 16 bit output values of the CAN interface | | | |
| | | | ▸ System bus "CAN on board" |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| Bit 1 | Active | | |
| Bit 2 | Active | | |
| Bit 3 | Active | | |
| Bit 4 | Active | | |
| Bit 5 | Active | | |
| Bit 6 | Active | | |
| Bit 7 | Active | | |
| Bit 8 | Active | | |
| Bit 9 | Active | | |
| Bit 10 | Active | | |
| Bit 11 | Active | | |
| Bit 12 | Active | | |
| Bit 13 | Active | | |
| Bit 14 | Active | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00868/1 | | | LP_CanOut1: wState |
| C00868/2 | | | LP_CanOut1: wOut2 |
| C00868/3 | | | LP_CanOut1: wOut3 |
| C00868/4 | | | LP_CanOut1: wOut4 |
| C00868/5 | | | LP_CanOut2: wOut1 |
| C00868/6 | | | LP_CanOut2: wOut2 |
| C00868/7 | | | LP_CanOut2: wOut3 |
| C00868/8 | | | LP_CanOut2: wOut4 |
| C00868/9 | | | LP_CanOut3: wOut1 |
| C00868/10 | | | LP_CanOut3: wOut2 |
| C00868/11 | | | LP_CanOut3: wOut3 |
| C00868/12 | | | LP_CanOut3: wOut4 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C00876**

| Parameter \| Name: **C00876 \| MCI input words** | | | Data type: UNSIGNED_16 Index: $23699_d$ = $5C93_h$ |
|---|---|---|---|
| Display of the 16 bit input values of the communication module | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00876/1 | | | LP_MciIn: wCtrl |
| C00876/2 | | | LP_MciIn: wIn2 |
| C00876/3 | | | LP_MciIn: wIn3 |
| C00876/4 | | | LP_MciIn: wIn4 |
| C00876/5 | | | LP_MciIn: wIn5 |
| C00876/6 | | | LP_MciIn: wIn6 |
| C00876/7 | | | LP_MciIn: wIn7 |
| C00876/8 | | | LP_MciIn: wIn8 |
| C00876/9 | | | LP_MciIn: wIn9 |
| C00876/10 | | | LP_MciIn: wIn10 |
| C00876/11 | | | LP_MciIn: wIn11 |
| C00876/12 | | | LP_MciIn: wIn12 |
| C00876/13 | | | LP_MciIn: wIn13 |
| C00876/14 | | | LP_MciIn: wIn14 |
| C00876/15 | | | LP_MciIn: wIn15 |
| C00876/16 | | | LP_MciIn: wIn16 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

## C00877

| Parameter \| Name: **C00877 \| MCI output words** | | | Data type: UNSIGNED_16 Index: $23698_d$ = $5C92_h$ |
|---|---|---|---|
| Display of the 16 bit output values of the communication module | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | |
| Bit 0 | Active | | |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | | | **Info** |
| C00877/1 | | | LP_MciOut: wState |
| C00877/2 | | | LP_MciOut: wOut2 |
| C00877/3 | | | LP_MciOut: wOut3 |
| C00877/4 | | | LP_MciOut: wOut4 |
| C00877/5 | | | LP_MciOut: wOut5 |
| C00877/6 | | | LP_MciOut: wOut6 |
| C00877/7 | | | LP_MciOut: wOut7 |
| C00877/8 | | | LP_MciOut: wOut8 |
| C00877/9 | | | LP_MciOut: wOut9 |
| C00877/10 | | | LP_MciOut: wOut10 |
| C00877/11 | | | LP_MciOut: wOut11 |
| C00877/12 | | | LP_MciOut: wOut12 |
| C00877/13 | | | LP_MciOut: wOut13 |
| C00877/14 | | | LP_MciOut: wOut14 |
| C00877/15 | | | LP_MciOut: wOut15 |
| C00877/16 | | | LP_MciOut: wOut16 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

## C00890

| Parameter \| Name: **C00890 \| MCI_InOut: Inversion** | | | Data type: UNSIGNED_16 Index: $23685_d$ = $5C85_h$ |
|---|---|---|---|
| This parameter serves to invert the control/status bits of the MCI port blocks. | | | |
| **Setting range** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Active | | Bit set = inversion active |
| ... | ... | | |
| Bit 15 | Active | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00890/1 | 0 | | Inversion of LP_MciIn.wCtrl |
| C00890/2 | 0 | | Inversion of LP_MciOut.wState |
| C00890/3 | 0 | | Inversion of LP_MciIn.bIn2_B0...15 |
| C00890/4 | 0 | | Inversion of LP_MciOut.bOut2_B0...15 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

## C00905

| Parameter \| Name: **C00905 \| Motor phase direction of rotation** | | | Data type: UNSIGNED_8 Index: $23670_d$ = $5C76_h$ |
|---|---|---|---|
| To correct such misconnected motor phases, the rotating field of the controller's output can be reversed by selecting "1: Inverted". In this case, a phase will be reversed at the output of the inverter. | | | |
| **Selection list** (Lenze setting printed in bold) | | | |
| **0** | **not inverted** | | |
| 1 | inverted | | |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00909

| Parameter \| Name: **C00909 \| Speed limitation** | | | Data type: INTEGER_16 Index: $23666_d$ = $5C72_h$ |
|---|---|---|---|
| Max. positive/negative speed for all motor control modes | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 175.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00909/1 | 120.00 % | | Max. pos. speed |
| C00909/2 | 120.00 % | | Max. neg. speed |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00910

| Parameter \| Name: **C00910 \| Frequency limitation** | | | Data type: UNSIGNED_16 Index: $23665_d$ = $5C71_h$ |
|---|---|---|---|
| Max. positive/negative output frequency for all motor control modes | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Hz | 1000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00910/1 | 1000 Hz | | Max. pos. output frequency |
| C00910/2 | 1000 Hz | | Max. neg. output frequency |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00915

| Parameter \| Name: **C00915 \| Motor cable length** | | | Data type: UNSIGNED_16 Index: $23660_d$ = $5C6C_h$ |
|---|---|---|---|
| Single motor cable length for calculating the motor cable resistance<br>• The calculated motor cable resistance is displayed in C00917. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | m | 1000.0 | **5.0 m** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

## C00916

| Parameter \| Name: **C00916 \| Motor cable cross-section** | | | Data type: UNSIGNED_16 Index: $23659_d$ = $5C6B_h$ |
|---|---|---|---|
| Motor cable cross-section of a phase/cable for calculating the motor cable resistance<br>• The calculated motor cable resistance is displayed in C00917. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.50 | mm^2 | 100.00 | **6.00 mm^2** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

### C00917

| Parameter \| Name: C00917 \| Motor cable resistance | | | Data type: UNSIGNED_16 Index: $23658_d$ = $5C6A_h$ |
|---|---|---|---|
| Display of the motor cable resistance of a motor cable phase<br>• The motor cable resistance is calculated from the motor cable length set in C00915 and the motor cable cross-section set in C00916. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | mOhm | 64000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

### C00918

| Parameter \| Name: C00918 \| SC: Start motor magnetising current | | | Data type: UNSIGNED_8 Index: $23657_d$ = $5C69_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | % | 90 | **87 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

### C00922

| Parameter \| Name: C00922 \| ICM_DiagnosticCounter | Data type: UNSIGNED_16 Index: $23653_d$ = $5C65_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

### C00925

| Parameter \| Name: C00925 \| LS_Resolver: Number of pole pairs | | | Data type: UNSIGNED_8 Index: $23650_d$ = $5C62_h$ |
|---|---|---|---|
| | | | ▶ Encoder/feedback system |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | | 10 | **1** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

### C00926

| Parameter \| Name: C00926 \| Pole position | | | Data type: INTEGER_16 Index: $23649_d$ = $5C61_h$ |
|---|---|---|---|
| | | | ▶ Encoder/feedback system |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -179.9 | ° | 179.9 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00926/1 | -90.0 ° | | LS_Resolver: Pole position |
| C00926/2 | 0.0 ° | | LS_MultiEncoder: Pole position |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

### C00927

| Parameter \| Name: C00927 \| Motor rotor position | | | Data type: UNSIGNED_16 Index: $23648_d$ = $5C60_h$ |
|---|---|---|---|
| | | | ▶ Encoder/feedback system |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 2047 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

## C00937

| Parameter \| Name: C00937 \| Field-oriented motor currents | | | Data type: INTEGER_16 Index: 23638d = 5C56h |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Field weakening for synchronous motors |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | A | 320.00 | |
| **Subcodes** | | | **Info** |
| C00937/1 | | | Field-producing current |
| C00937/2 | | | Torque-producing current |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

## C00938

| Parameter \| Name: C00938 \| PSM: Maximum motor current field weakening | | | Data type: UNSIGNED_16 Index: 23637d = 5C55h |
|---|---|---|---|
| From version 02.00.00 | | | ▸ Field weakening for synchronous motors |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 500.00 | **100.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

## C00939

| Parameter \| Name: C00939 \| Ultimate motor current | | | Data type: UNSIGNED_16 Index: 23636d = 5C54h |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | A | 3000.0 | **3000.0 A** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

## C00940

| Parameter \| Name: C00940 \| L_ConvW numerator | | | Data type: INTEGER_16 Index: 23635d = 5C53h |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00940/1 | 1 | | L_ConvW_1: Numerator |
| C00940/2 | 1 | | L_ConvW_2: Numerator |
| C00940/3 | 1 | | L_ConvW_3: Numerator |
| C00940/4 | 1 | | L_ConvW_4: Numerator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00941

| Parameter \| Name: C00941 \| L_ConvW denominator | | | Data type: INTEGER_16 Index: 23634d = 5C52h |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00941/1 | 1 | | L_ConvW_1: Denominator |
| C00941/2 | 1 | | L_ConvW_2: Denominator |
| C00941/3 | 1 | | L_ConvW_3: Denominator |
| C00941/4 | 1 | | L_ConvW_4: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00942**

| Parameter \| Name: **C00942 \| L_ConvW conversion method** | | Data type: UNSIGNED_8 Index: 23633$_d$ = 5C51$_h$ |
|---|---|---|
| **Selection list** | | |
| 0 | no conversion | |
| 1 | from [%] into [incr./ms] | |
| 2 | from [incr./ms] into [%] | |
| 3 | Factors signed | |
| 4 | Factors unsigned | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00942/1 | 0: No conversion | L_ConvW_1: Conversion method |
| C00942/2 | 0: No conversion | L_ConvW_2: Conversion method |
| C00942/3 | 0: No conversion | L_ConvW_3: Conversion method |
| C00942/4 | 0: No conversion | L_ConvW_4: Conversion method |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00950**

| Parameter \| Name: **C00950 \| L_Interpolator_1: Activation FB functions** | | Data type: UNSIGNED_8 Index: 23625$_d$ = 5C49$_h$ |
|---|---|---|
| The L_Interpolator_1 FB: Activation of signal interpolation and signal monitoring | | |
| **Selection list** | | |
| 0 | Off | |
| 1 | On | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00950/1 | 0: Off | L_Interpolator_1: Signal interpolation |
| C00950/2 | 0: Off | L_Interpolator_1: Signal monitoring |
| C00950/3 | 0: Off | L_Interpolator_1: Master value monitoring |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00951**

| Parameter \| Name: **C00951 \| L_Interpolator_1: No. of interpolation steps** | | | Data type: UNSIGNED_16 Index: 23624$_d$ = 5C48$_h$ |
|---|---|---|---|
| The L_Interpolator_1 FB: No. of interpolation steps | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 65535 | **1** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00952**

| Parameter \| Name: **C00952 \| L_Interpolator_1: Limit value - error cycles** | | | Data type: UNSIGNED_16 Index: 23623$_d$ = 5C47$_h$ |
|---|---|---|---|
| The L_Interpolator_1 FB: Limit value for missing data telegrams | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 65535 | **5** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00953**

| Parameter \| Name: **C00953 \| L_Interpolator_1: Speed-up** | | | Data type: UNSIGNED_8 Index: 23622$_d$ = 5C46$_h$ |
|---|---|---|---|
| FB L_Interpolator_1: Filter | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 100 | **0** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C00959

| Parameter \| Name: **C00959 \| L_Curve: Current output value** | | | Data type: INTEGER_16 Index: $23616_d$ = $5C40_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| FB L_Curve_1: Display of the current output value *nOut_a* | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | | | **Info** |
| C00959/1 | | | L_Curve_1: Current output value |
| C00959/2 | | | L_Curve_2: Current output value |
| C00959/3 | | | L_Curve_3: Current output value |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00960

| Parameter \| Name: **C00960 \| L_Curve_1: Selected curve type** | | Data type: UNSIGNED_8 Index: $23615_d$ = $5C3F_h$ |
|---|---|---|
| FB L_Curve_1: Selected curve type | | |
| **Selection list** | | |
| 0 | Out = 0 | |
| 1 | Out = In | |
| 2 | Out = f(In) | |
| 3 | Out = f(table) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C00960/1 | 1: Out = In | L_Curve_1: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C00961

| Parameter \| Name: **C00961 \| L_Curve_1: Input limitation** | | | Data type: INTEGER_16 Index: $23614_d$ = $5C3E_h$ |
|---|---|---|---|
| FB L_Curve_1: Upper and lower limit for input value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00961/1 | 199.99 % | | L_Curve_1: Max. input |
| C00961/2 | -199.99 % | | L_Curve_1: Min. input |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00963

| Parameter \| Name: **C00963 \| L_Curve_1: Table X-values** | | | Data type: INTEGER_16 Index: $23612_d$ = $5C3C_h$ |
|---|---|---|---|
| FB L_Curve_1: X-values for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00963/1 | 0 | | X values 1 ... 32 for characteristic function |
| C00963/... | | | |
| C00963/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00964**

| Parameter \| Name: **C00964 \| L_Curve_1: Table Y-values** | | | Data type: INTEGER_16 Index: $23611_d$ = $5C3B_h$ |
|---|---|---|---|
| FB L_Curve_1: Y-value for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00964/1 | 0 | | Y values 1 ... 32 for characteristic function |
| C00964/... | | | |
| C00964/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00965**

| Parameter \| Name: **C00965 \| Max. motor speed** | | | Data type: UNSIGNED_16 Index: $23610_d$ = $5C3A_h$ |
|---|---|---|---|
| When the drive reaches the motor speed set here:<br>• The "Fault" error response takes place, i.e. the motor is shut down immediately.<br>• The error message "OS2: Max. motor speed reached" is entered into the logbook. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 50 | rpm | 60000 | **60000 rpm** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00966**

| Parameter \| Name: **C00966 \| VFC: Time const. slip comp.** | | | Data type: UNSIGNED_16 Index: $23609_d$ = $5C39_h$ |
|---|---|---|---|
| Filter time constant of the slip compensation for V/f characteristic control (VFCplus)<br>• The time constant of slip compensation serves to specify the dynamics of slip compensation for V/f characteristic control without feedback.<br>• The lower the selected time constant, the higher the dynamic performance of the slip compensation. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 1 | ms | 6000 | **100 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

## C00967

| Parameter \| Name: C00967 \| VFC: Frequency interpol. point n | | | Data type: INTEGER_16 Index: $23608_d$ = $5C38_h$ |
|---|---|---|---|
| Selection of the interpolation points (frequency values) for the V/f characteristic control (VFCplus) with user-definable V/f characteristic (C00006 = "10") | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2600.0 | Hz | 2600.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00967/1 | -50.0 Hz | | VFC: Frequency interpol. point 1 |
| C00967/2 | -40.0 Hz | | VFC: Frequency interpol. point 2 |
| C00967/3 | -30.0 Hz | | VFC: Frequency interpol. point 3 |
| C00967/4 | -20.0 Hz | | VFC: Frequency interpol. point 4 |
| C00967/5 | -10.0 Hz | | VFC: Frequency interpol. point 5 |
| C00967/6 | 0.0 Hz | | VFC: Frequency interpol. point 6 |
| C00967/7 | 10.0 Hz | | VFC: Frequency interpol. point 7 |
| C00967/8 | 20.0 Hz | | VFC: Frequency interpol. point 8 |
| C00967/9 | 30.0 Hz | | VFC: Frequency interpol. point 9 |
| C00967/10 | 40.0 Hz | | VFC: Frequency interpol. point 10 |
| C00967/11 | 50.0 Hz | | VFC: Frequency interpol. point 11 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

## C00968

| Parameter \| Name: C00968 \| VFC: Voltage interpol. point n | | | Data type: UNSIGNED_16 Index: $23607_d$ = $5C37_h$ |
|---|---|---|---|
| Selection of the interpolation points (voltage values) for the V/f characteristic control (VFCplus) with user-definable V/f characteristic (C00006 = "10") | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | V | 600.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00968/1 | 400.00 V | | VFC: Voltage interpol. point 1 |
| C00968/2 | 320.00 V | | VFC: Voltage interpol. point 2 |
| C00968/3 | 240.00 V | | VFC: Voltage interpol. point 3 |
| C00968/4 | 160.00 V | | VFC: Voltage interpol. point 4 |
| C00968/5 | 80.00 V | | VFC: Voltage interpol. point 5 |
| C00968/6 | 0.00 V | | VFC: Voltage interpol. point 6 |
| C00968/7 | 80.00 V | | VFC: Voltage interpol. point 7 |
| C00968/8 | 160.00 V | | VFC: Voltage interpol. point 8 |
| C00968/9 | 240.00 V | | VFC: Voltage interpol. point 9 |
| C00968/10 | 320.00 V | | VFC: Voltage interpol. point 10 |
| C00968/11 | 400.00 V | | VFC: Voltage interpol. point 11 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00971**

| Parameter \| Name: **C00971 \| VFC: Limitation V/f +encoder** | | | Data type: UNSIGNED_16 Index: $23604_d$ = $5C34_h$ |
|---|---|---|---|
| Limitation of the output frequency of the slip regulator and limitation of the injected stator frequency for the V/f control (VFCplus+encoder) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | Hz | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00971/1 | 10.00 Hz | | Maximum output / correcting variable of the slip regulator<br>• The slip regulator output is limited to the value set here in motor/generator mode.<br>• It is recommended to select 1 to 3 times the slip frequency of the motor as limit value. |
| C00971/2 | 100.00 Hz | | Maximum frequency deviation between the rotational frequency (speed) measured mechanically by the encoder and the injected stator frequency.<br>• A limitation may e.g. avoid overcurrent interruption when traversing to a fixed limit stop. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00972**

| Parameter \| Name: **C00972 \| VFC: Vp V/f +encoder** | | | Data type: UNSIGNED_16 Index: $23603_d$ = $5C33_h$ |
|---|---|---|---|
| Proportional gain of the slip regulator for V/f control (VFCplus+encoder)<br>• The gain must be selected depending on the drive system and the sensor resolution (range: 0.005 ... 5).<br>• A high gain requires a high number of increments. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | Hz/Hz | 64.000 | **0.100 Hz/Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1000 | | | |

**C00973**

| Parameter \| Name: **C00973 \| VFC: Ti V/f +encoder** | | | Data type: UNSIGNED_16 Index: $23602_d$ = $5C32_h$ |
|---|---|---|---|
| Integral time constant of the slip regulator for V/f control (VFCplus+encoder)<br>• In general, the time constant should be selected in a range of 20 ms (high dynamics) to 200 (low dynamics). | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 6000.0 | **100.0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C00975**

| Parameter \| Name: **C00975 \| VFC-ECO: Vp CosPhi controller** | | | Data type: UNSIGNED_16 Index: $23600_d$ = $5C30_h$ |
|---|---|---|---|
| Proportional gain of the Cos-Phi controller for energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.000 | Hz/Hz | 64.000 | **0.500 Hz/Hz** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00976**

| Parameter \| Name: **C00976 \| VFC-ECO: Ti CosPhi controller** | | | Data type: UNSIGNED_16 Index: $23599_d$ = $5C2F_h$ |
|---|---|---|---|
| Reset time of the Cos-Phi controller for energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 6000.0 | **200.0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C00977**

| Parameter \| Name: **C00977 \| VFC-ECO: Minimum voltage V/f** | | | Data type: INTEGER_16 Index: $23598_d$ = $5C2E_h$ |
|---|---|---|---|
| Minimum voltage V/f of the Cos-Phi controller for energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 20.00 | % | 100.00 | **20.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00978**

| Parameter \| Name: **C00978 \| VFC-ECO: Voltage reduction** | | | Data type: INTEGER_16 Index: $23597_d$ = $5C2D_h$ |
|---|---|---|---|
| Display of the voltage reduction with energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -1000 | V | 1000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C00979**

| Parameter \| Name: **C00979 \| Cosine phi** | | | Data type: INTEGER_16 Index: $23596_d$ = $5C2C_h$ |
|---|---|---|---|
| Display of the cosφ setpoint and actual value with energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -1.00 | | 1.00 | |
| **Subcodes** | | | **Info** |
| C00979/1 | | | Cosine phi act |
| C00979/2 | | | Cosine phi set |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00980**

| Parameter \| Name: **C00980 \| Output power** | | | Data type: INTEGER_32 Index: $23595_d$ = $5C2B_h$ |
|---|---|---|---|
| Display parameter for an energy analysis in the prevailing application. From this, decisions can be deduced whether a measure for energy optimisation is economic. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.000 | kW | 32.000 | |
| **Subcodes** | | | **Info** |
| C00980/1 | | | Active output power |
| C00980/2 | | | Apparent output power |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C00981**

| Parameter \| Name: **C00981 \| Energy display** | | | Data type: INTEGER_32 Index: $23594_d$ = $5C2A_h$ |
|---|---|---|---|
| Display parameter for an energy analysis in the prevailing application. From this, decisions can be deduced whether a measure for energy optimisation is economic.<br>• The values are saved to the device by switching off the mains and cannot be reset. | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | kWh | 21474836.47 | |
| **Subcodes** | | | **Info** |
| C00981/1 | | | Output energy in motor mode |
| C00981/2 | | | Output energy in generator mode |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00982**

| Parameter \| Name: **C00982 \| VFC-ECO: Voltage reduction ramp** | | | Data type: UNSIGNED_8 Index: $23593_d$ = $5C29_h$ |
|---|---|---|---|
| Voltage ramp for cancelling V-Sub with energy-saving V/f characteristic control (VFCplusEco) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.1 | s | 5.0 | **0.8 s** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C00985**

| Parameter \| Name: **C00985 \| SLVC: Field current controller gain** | | | Data type: INTEGER_16 Index: $23590_d$ = $5C26_h$ |
|---|---|---|---|
| Gain of the direct-axis current difference (Id) between setpoint and actual current for the voltage model of the sensorless vector control (SLVC)<br>• The gain should be selected in a range of 0 ...1 %. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 10.00 | **0.50 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00986**

| Parameter \| Name: **C00986 \| SLVC: Cross current controller gain** | | | Data type: INTEGER_16 Index: $23589_d$ = $5C25_h$ |
|---|---|---|---|
| Gain of the  cross current difference for the voltage model of the sensorless vector control (SLVC) | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 10.00 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C00987**

| Parameter \| Name: **C00987 \| Inverter motor brake: nAdd** | | | Data type: INTEGER_16 Index: $23588_d$ = $5C24_h$ |
|---|---|---|---|
| Speed lift which is connected in pulses to the brake rampe when the motor is braked.<br>▸ Inverter motor brake | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | rpm | 1000 | **80 rpm** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C00988**

| Parameter \| Name: **C00988 \| Inverter motor brake: PT1 filter time** | | | Data type: INTEGER_16 Index: $23587_d$ = $5C23_h$ |
|---|---|---|---|
| PT1 filter time for smoothing the speed lift which is added in pulses (C00987)<br>▸ Inverter motor brake | | | |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | ms | 100.0 | **0.0 ms** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10 | | | |

**C00990**

| Parameter \| Name: **C00990 \| Flying restart fct.: Activation** | | Data type: UNSIGNED_8 Index: $23585_d$ = $5C21_h$ |
|---|---|---|
| Switch on/activate flying restart circuit for non-feedback drive systems<br>▸ Flying restart fct. | | |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **Off** | |
| 1 | On | |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C00991**

| Parameter \| Name: **C00991 \| Flying restart fct.: Process** | | | Data type: UNSIGNED_16 Index: $23584_d$ = $5C20_h$ |
|---|---|---|---|
| Selection of the starting value and the speed search range for the flying restart function | | | |

▶ Flying restart fct.

| Selection list (Lenze setting printed in bold) | | Info |
|---|---|---|
| 0 | 0...+n \| Start: +10 Hz | Search positive speed range (0 ... +n) with a start frequency of +10 Hz |
| 1 | -n...0 \| Start: -10 Hz | Search negative speed range (-n ... 0) with a start frequency of -10 Hz |
| **2** | **-n...+n \| Start: +10 Hz** | Search negative and positive speed range (-n ... n) with a start frequency of +10 Hz |
| 3 | -n...+n \| Start: -10 Hz | Search negative and positive speed range (-n ... n) with a start frequency of -10 Hz |
| 4 | -n...+n \| Start: Cx992 | Search the negative and positive speed range (-n ... n) with the start frequency set in C00992 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00992**

| Parameter \| Name: **C00992 \| Flying restart fct.: Start frequency** | Data type: INTEGER_16 Index: $23583_d$ = $5C1F_h$ |
|---|---|

Manual selection of the starting value for the flying restart function
- Only active if C00991 = 4

▶ Flying restart fct.

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| -200 | Hz | 200 | **10 Hz** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C00993**

| Parameter \| Name: **C00993 \| Flying restart fct: Int. time** | Data type: UNSIGNED_16 Index: $23582_d$ = $5C1E_h$ |
|---|---|

Time constant of the angular difference controller of the flying restart function
- The time constant is to amount between 60 ... 300 ms.

▶ Flying restart fct.

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.0 | ms | 6000.0 | **300.0 ms** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10

**C00994**

| Parameter \| Name: **C00994 \| Flying restart fct.: Current** | Data type: INTEGER_16 Index: $23581_d$ = $5C1D_h$ |
|---|---|

Current to be injected during the flying restart process
- 100 % ≡ rated motor current (C00088).
- The flying restart current should amount to 10 ... 25 % of the rated motor current.

▶ Flying restart fct.

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.00 | % | 100.00 | **25.00 %** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

## C00995

| Parameter \| Name: C00995 \| SLPSM: Controlled current setpoint | | | Data type: UNSIGNED_16 Index: $23580_d$ = $5C1C_h$ |
|---|---|---|---|
| | | | ▸ Sensorless control for synchronous motor |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 5.00 | % | 400.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00995/1 | 100.00 % | | SLPSM: Controlled accelerating current |
| C00995/2 | 20.00 % | | SLPSM: Controlled standstill current |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100 | | | |

## C00996

| Parameter \| Name: C00996 \| SLPSM: Switching speed | | | Data type: INTEGER_16 Index: $23579_d$ = $5C1B_h$ |
|---|---|---|---|
| | | | ▸ Sensorless control for synchronous motor |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00996/1 | 13.00 % | | SLPSM: Switching speed, closed-loop control |
| C00996/2 | 8.00 % | | SLPSM: Switching speed, open-loop control |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00997

| Parameter \| Name: C00997 \| SLPSM: Filter cutoff frequency | | | Data type: INTEGER_16 Index: $23578_d$ = $5C1A_h$ |
|---|---|---|---|
| | | | ▸ Sensorless control for synchronous motor |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.00 | % | 100.00 | **5.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C00998

| Parameter \| Name: C00998 \| SLPSM: Filter time rotor position | | | Data type: INTEGER_16 Index: $23577_d$ = $5C19_h$ |
|---|---|---|---|
| | | | ▸ Sensorless control for synchronous motor |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.5 | ms | 20.0 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C00998/1 | 3.0 ms | | SLPSM: Filter time rotor position |
| C00998/2 | 5.0 ms | | SLPSM: Filter time actual speed value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

## C00999

| Parameter \| Name: C00999 \| SLPSM: PLL gain | | | Data type: INTEGER_16 Index: $23576_d$ = $5C18_h$ |
|---|---|---|---|
| | | | ▸ Sensorless control for synchronous motor |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | % | 1000 | **400 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C01000**

| Parameter \| Name: **C01000 \| MCTRL: Status** | | Data type: UNSIGNED_16 Index: $23575_d$ = $5C17_h$ |
|---|---|---|
| **Display area** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** | | |
| Bit 0 | SL PSM: Mode | |
| Bit 1 | Reserved | |
| Bit 2 | Reserved | |
| Bit 3 | Reserved | |
| Bit 4 | Reserved | |
| Bit 5 | Reserved | |
| Bit 6 | Reserved | |
| Bit 7 | Reserved | |
| Bit 8 | Reserved | |
| Bit 9 | Reserved | |
| Bit 10 | Reserved | |
| Bit 11 | Reserved | |
| Bit 12 | Reserved | |
| Bit 13 | Reserved | |
| Bit 14 | Reserved | |
| Bit 15 | Reserved | |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT

**C01010**

| Parameter \| Name: **C01010 \| L_ArithmetikPhi 1-3: Function** | | Data type: UNSIGNED_8 Index: $23565_d$ = $5C0D_h$ |
|---|---|---|
| Selection of the internal arithmetics | | |
| **Selection list** | | |
| 0 | dnOut_p = dnIn1_p | |
| 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| 4 | dnOut_p = dnIn1_p / dnIn2_p | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01010/1 | 0: dnOut_p = dnIn1_p | L_ArithmetikPhi_1: Function |
| C01010/2 | 0: dnOut_p = dnIn1_p | L_ArithmetikPhi_2: Function |
| C01010/3 | 0: dnOut_p = dnIn1_p | L_ArithmetikPhi_3: Function |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01020

| Parameter \| Name: **C01020 \| L_Odometer_1: Memory length** | Data type: UNSIGNED_8 Index: $23555_d$ = $5C03_h$ |
|---|---|
| FB L_Odometer_1: No. of measurements | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| 1 | 1 measurement |
| 2 | 2 measurements |
| 3 | 3 measurements |
| 4 | 4 measurements |
| 5 | 5 measurements |
| 6 | 6 measurements |
| **7** | **7 measurements** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01021

| Parameter \| Name: **C01021 \| L_Odometer_1: Memory type** | Data type: UNSIGNED_8 Index: $23554_d$ = $5C02_h$ |
|---|---|
| FB L_Odometer_1: If "ring buffer" is selected, it is started again after the measurements set in C01020 and the old values are overwritten. Otherwise, the measurement stops. | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **No ring buffer** |
| 1 | Ring buffer |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01022

| Parameter \| Name: **C01022 \| L_Odometer_1: Input selection** | Data type: UNSIGNED_8 Index: $23553_d$ = $5C01_h$ |
|---|---|
| FB L_Odometer_1: Selection of position or speed input | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **Pos input** |
| 1 | V input |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01023

| Parameter \| Name: **C01023 \| L_Odometer_1: Edge selection** | Data type: UNSIGNED_8 Index: $23552_d$ = $5C00_h$ |
|---|---|
| FB L_Odometer_1: No. of the edge triggering the measurement | |

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| **0** | **High edge** |
| 1 | Low edge |
| 2 | High and low edge |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01025

| Parameter \| Name: **C01025 \| L_Curve_2: Selected curve type** | | Data type: UNSIGNED_8 Index: $23550_d$ = $5BFE_h$ |
|---|---|---|
| From version 02.00.00<br>FB L_Curve_2: Selected curve type | | |
| **Selection list** | | |
| 0 | Out = 0 | |
| 1 | Out = In | |
| 2 | Out = f(In) | |
| 3 | Out = f(table) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01025/1 | 1: Out = In | L_Curve_2: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C01026

| Parameter \| Name: **C01026 \| L_Curve_2: Input limitation** | | | Data type: INTEGER_16 Index: $23549_d$ = $5BFD_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_2: Upper and lower limit for input value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01026/1 | 199.99 % | | L_Curve_2: Max. input |
| C01026/2 | -199.99 % | | L_Curve_2: Min. input |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C01028

| Parameter \| Name: **C01028 \| L_Curve_2: Table X-values** | | | Data type: INTEGER_16 Index: $23547_d$ = $5BFB_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_2: X-values for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01028/1 | 0 | | X values 1 ... 32 for characteristic function |
| C01028/... | | | |
| C01028/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01029

| Parameter \| Name: **C01029 \| L_Curve_2: Table Y-values** | | | Data type: INTEGER_16 Index: $23546_d$ = $5BFA_h$ |
|---|---|---|---|
| From version 02.00.00<br>FB L_Curve_2: Y-value for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01029/1 | 0 | | Y values 1 ... 32 for characteristic function |
| C01029/... | | | |
| C01029/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01030**

| Parameter \| Name: **C01030 \| L_Curve_3: Selected curve type** | | | Data type: UNSIGNED_8 Index: $23545_d$ = $5BF9_h$ |
|---|---|---|---|
| From version 02.00.00 FB L_Curve_3: Selected curve type | | | |
| **Selection list** | | | |
| 0 | Out = 0 | | |
| 1 | Out = In | | |
| 2 | Out = f(In) | | |
| 3 | Out = f(table) | | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C01030/1 | 1: Out = In | L_Curve_3: Function | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01031**

| Parameter \| Name: **C01031 \| L_Curve_3: Input limitation** | | | Data type: INTEGER_16 Index: $23544_d$ = $5BF8_h$ |
|---|---|---|---|
| From version 02.00.00 FB L_Curve_3: Upper and lower limit for input value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C01031/1 | 199.99 % | L_Curve_3: Max. input | |
| C01031/2 | -199.99 % | L_Curve_3: Min. input | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C01033**

| Parameter \| Name: **C01033 \| L_Curve_3: Table X-values** | | | Data type: INTEGER_16 Index: $23542_d$ = $5BF6_h$ |
|---|---|---|---|
| From version 02.00.00 FB L_Curve_3: X-values for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C01033/1 | 0 | X values 1 ... 32 for characteristic function | |
| C01033/... | | | |
| C01033/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01034**

| Parameter \| Name: **C01034 \| L_Curve_3: Table Y-values** | | | Data type: INTEGER_16 Index: $23541_d$ = $5BF5_h$ |
|---|---|---|---|
| From version 02.00.00 FB L_Curve_3: Y-value for characteristic function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C01034/1 | 0 | Y values 1 ... 32 for characteristic function | |
| C01034/... | | | |
| C01034/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01040

| Parameter \| Name: **C01040 \| L_SRFG_1..2 linear ramp time** | | | Data type: UNSIGNED_32 Index: $23535_d$ = $5BEF_h$ |
|---|---|---|---|
| Symmetrical acceleration/deceleration time for the ramp function generator | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.001 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01040/1 | 100.000 s | | L_SRFG_1: linear ramp time |
| C01040/2 | 100.000 s | | L_SRFG_2: linear ramp time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

## C01041

| Parameter \| Name: **C01041 \| L_SRFG_1..2 S-ramp time** | | | Data type: UNSIGNED_32 Index: $23534_d$ = $5BEE_h$ |
|---|---|---|---|
| S-ramp time for jerk-free acceleration | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.001 | s | 50.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01041/1 | 0.200 s | | L_SRFG_1: S-ramp time |
| C01041/2 | 0.200 s | | L_SRFG_2: S-ramp time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

## C01042

| Parameter \| Name: **C01042 \| L_SRFG_1..2 limitations of output values** | | | Data type: INTEGER_16 Index: $23533_d$ = $5BED_h$ |
|---|---|---|---|
| Limitation of the output values | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01042/1 | 100.00 % | | L_SRFG_1: Pos. Limit |
| C01042/2 | -100.00 % | | L_SRFG_1: Neg. Limit |
| C01042/3 | 100.00 % | | L_SRFG_2: Pos. Limit |
| C01042/4 | -100.00 % | | L_SRFG_2: Neg. Limit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

## C01045

| Parameter \| Name: **C01045 \| L_ConvAP 1-3: Numerator/denominator** | | | Data type: INTEGER_16 Index: $23530_d$ = $5BEA_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01045/1 | 1 | | L_ConvAP_1: Numerator |
| C01045/2 | 1 | | L_ConvAP_1: Numerator |
| C01045/3 | 1 | | L_ConvAP_2: Numerator |
| C01045/4 | 1 | | L_ConvAP_2: Numerator |
| C01045/5 | 1 | | L_ConvAP_3: Numerator |
| C01045/6 | 1 | | L_ConvAP_3: Numerator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01046

| Parameter \| Name: C01046 \| L_ConvPA 1-3: byDivision | | | Data type: INTEGER_8 Index: $23529_d$ = $5BE9_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| Setting range (min. value \| unit \| max. value) | | | |
| 0 | | 31 | |
| Subcodes | Lenze setting | | Info |
| C01046/1 | 1 | | L_ConvPA_1: Division factor |
| C01046/2 | 1 | | L_ConvPA_2: Division factor |
| C01046/3 | 1 | | L_ConvPA_3: Division factor |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01047

| Parameter \| Name: C01047 \| L_GearComp: Offset | | | Data type: INTEGER_16 Index: $23528_d$ = $5BE8_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| Setting range (min. value \| unit \| max. value) | | | |
| -16383 | | 16383 | |
| Subcodes | Lenze setting | | Info |
| C01047/1 | 0 | | L_GearComp_1: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01048

| Parameter \| Name: C01048 \| L_GearComp: Num_Denom | | | Data type: INTEGER_16 Index: $23527_d$ = $5BE7_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| Setting range (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| Subcodes | Lenze setting | | Info |
| C01048/1 | 1 | | L_GearComp_1: Numerator |
| C01048/2 | 1 | | L_GearComp_1: Denominator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01049

| Parameter \| Name: C01049 \| L_CalcDiameter: Status | | | Data type: INTEGER_16 Index: $23526_d$ = $5BE6_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| Display range (min. value \| unit \| max. value) | | | |
| -10 | | 30 | |
| Subcodes | | | Info |
| C01049/1 | | | L_CalcDiameter_1: Status |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01050

| Parameter \| Name: C01050 \| L_CalcDiameter_1: Diameter recalculation | | | Data type: UNSIGNED_32 Index: $23525_d$ = $5BE5_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.001 | Rev. | 1000.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01050/1 | 1.000 rev. | | L_CalcDiameter_1: Diameter recalculation 0 |
| C01050/2 | 0.100 rev. | | L_CalcDiameter_1: Diameter recalculation 1 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

## C01051

| Parameter \| Name: C01051 \| L_CalcDiameter: Filter time constant | | | Data type: UNSIGNED_16 Index: $23524_d$ = $5BE4_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.010 | s | 3.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01051/1 | 1.000 s | | L_CalcDiameter_1: Filter time constant |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

## C01052

| Parameter \| Name: C01052 \| L_CalcDiameter: Web break monitoring | | | Data type: INTEGER_16 Index: $23523_d$ = $5BE3_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 100.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01052/1 | 10.00 % | | L_CalcDiameter_1: Permissible diameter change |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

## C01053

| Parameter \| Name: C01053 \| L_ProcessCtrl: Controller times | | | Data type: UNSIGNED_16 Index: $23522_d$ = $5BE2_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 30.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01053/1 | 0.000 s | | L_ProcessCtrl_1: Acceleration/deceleration time |
| C01053/2 | 0.000 s | | L_ProcessCtrl_1: Filter time constant |
| C01053/3 | 0.000 s | | L_ProcessCtrl_1: Rate time |
| C01053/4 | 1.000 s | | L_ProcessCtrl_1: Reset time |
| C01053/5 | 0.000 s | | L_ProcessCtrl_1: Rate action |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

## C01054

| Parameter \| Name: **C01054 \| L_ProcessCtrl: System deviation** | | Data type: INTEGER_16 Index: $23521_d$ = $5BE1_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0.00 | % | 199.99 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01054/1 | 100.00 % | L_ProcessCtrl_1: Gain of system deviation |
| C01054/2 | 0.00 % | L_ProcessCtrl_1: Area of system deviation |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | |

## C01055

| Parameter \| Name: **C01055 \| L_ProcessCtrl: Correcting value limitation** | | Data type: UNSIGNED_8 Index: $23520_d$ = $5BE0_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01055/1 | 0: FALSE | L_ProcessCtrl_1: Correcting variable limitation |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C01056

| Parameter \| Name: **C01056 \| L_ProcessCtrl: Controller gain** | | Data type: UNSIGNED_16 Index: $23519_d$ = $5BDF_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0.00 | | 100.00 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01056/1 | 0.10 | L_ProcessCtrl_1: Controller gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | |

## C01057

| Parameter \| Name: **C01057 \| L_CalcDiameter: Current diameter** | | Data type: UNSIGNED_32 Index: $23518_d$ = $5BDE_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Display range** (min. value \| unit \| max. value) | | |
| 0.000 | mm | 10000.000 |
| **Subcodes** | | **Info** |
| C01057/1 | | L_CalcDiameter_1: Current diameter |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | |

## C01058

| Parameter \| Name: C01058 \| L_PosCtrlLin 1-2: Limit stop | | Data type: UNSIGNED_8 Index: 23517d = 5BDDh |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | deactivated | |
| 1 | activated | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01058/1 | 0: Deactivated | L_PosCtrlLin_1: Limit stop |
| C01058/2 | 0: Deactivated | L_PosCtrlLin_2: Limit stop |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C01059

| Parameter \| Name: C01059 \| L_PosCtrlLin 1-2: Positioning behaviour | | Data type: UNSIGNED_8 Index: 23516d = 5BDCh |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | dnOut_p = 0 | |
| 1 | dnOut_p/nOut_v follow dnAct_p | |
| 2 | dnOut_p/nOut_v follow dnSet_p | |
| 3 | dnOut_p/nOut_v follow dnAct_p (without limitation) | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01059/1 | 0: dnOut_p = 0 | L_PosCtrlLin_1: Positioning behaviour |
| C01059/2 | 0: dnOut_p = 0 | L_PosCtrlLin_2: Positioning behaviour |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C01060

| Parameter \| Name: C01060 \| L_PosCtrlLin 1-2: Ramps | | | Data type: INTEGER_32 Index: 23515d = 5BDBh |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.010 | s | 130.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01060/1 | 1.000 s | | L_PosCtrlLin_1: Deceleration of set position |
| C01060/2 | 1.000 s | | L_PosCtrlLin_1: Acceleration ramp |
| C01060/3 | 1.000 s | | L_PosCtrlLin_1: Deceleration ramp |
| C01060/4 | 1.000 s | | L_PosCtrlLin_2: Deceleration of set position |
| C01060/5 | 1.000 s | | L_PosCtrlLin_2: Acceleration ramp |
| C01060/6 | 1.000 s | | L_PosCtrlLin_2: Deceleration ramp |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01061**

| Parameter \| Name:<br>**C01061 \| L_PosCtrlLin 1-2: Traversing speeds** | | | Data type: INTEGER_16<br>Index: $23514_d$ = $5BDA_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -15000 | rpm | 15000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01061/1 | 199 rpm | | L_PosCtrlLin_1: Forward motion |
| C01061/2 | 199 rpm | | L_PosCtrlLin_1: Return motion |
| C01061/3 | 199 rpm | | L_PosCtrlLin_2: Forward motion |
| C01061/4 | 199 rpm | | L_PosCtrlLin_2: Return motion |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01062**

| Parameter \| Name:<br>**C01062 \| L_SwitchPoint: Dead time** | | | Data type: UNSIGNED_16<br>Index: $23513_d$ = $5BD9_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | µs | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01062/1 | 0 µs | | L_SwitchPoint_1: Dead time 1 |
| C01062/2 | 0 µs | | L_SwitchPoint_1: Dead time 2 |
| C01062/3 | 0 µs | | L_SwitchPoint_1: Dead time 3 |
| C01062/4 | 0 µs | | L_SwitchPoint_1: Dead time 4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01063**

| Parameter \| Name:<br>**C01063 \| L_SwitchPoint: Hysteresis** | | | Data type: UNSIGNED_16<br>Index: $23512_d$ = $5BD8_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01063/1 | 0 incr. | | L_SwitchPoint_1: Hysteresis 1 |
| C01063/2 | 0 incr. | | L_SwitchPoint_1: Hysteresis 2 |
| C01063/3 | 0 incr. | | L_SwitchPoint_1: Hysteresis 3 |
| C01063/4 | 0 incr. | | L_SwitchPoint_1: Hysteresis 4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01064**

| Parameter \| Name: **C01064 \| L_SwitchPoint: CenterMode** | | Data type: UNSIGNED_8 Index: $23511_d$ = $5BD7_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01064/1 | 0: FALSE | L_SwitchPoint_1: CenterMode 1 |
| C01064/2 | 0: FALSE | L_SwitchPoint_1: CenterMode 2 |
| C01064/3 | 0: FALSE | L_SwitchPoint_1: CenterMode 3 |
| C01064/4 | 0: FALSE | L_SwitchPoint_1: CenterMode 4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01065**

| Parameter \| Name: **C01065 \| L_SwitchPoint: Running time** | | | Data type: UNSIGNED_16 Index: $23510_d$ = $5BD6_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01065/1 | 0 ms | | L_SwitchPoint_1: Running time 1 |
| C01065/2 | 0 ms | | L_SwitchPoint_1: Running time 2 |
| C01065/3 | 0 ms | | L_SwitchPoint_1: Running time 3 |
| C01065/4 | 0 ms | | L_SwitchPoint_1: Running time 4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01066**

| Parameter \| Name: **C01066 \| L_SwitchPoint: Status** | | Data type: INTEGER_16 Index: $23509_d$ = $5BD5_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| -10 | Switching points not plausible | |
| 0 | OK | |
| 10 | FB not active | |
| **Subcodes** | | **Info** |
| C01066/1 | | L_SwitchPoint_1: Status 1 |
| C01066/2 | | L_SwitchPoint_1: Status 2 |
| C01066/3 | | L_SwitchPoint_1: Status 3 |
| C01066/4 | | L_SwitchPoint_1: Status 4 |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C01067

| Parameter \| Name: C01067 \| Inversion of gearbox stages | | Data type: UNSIGNED_8 Index: $23508_d$ = $5BD4_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | not inverted | |
| 1 | inverted | |
| 2 | Automatically from MCK | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01067/1 | 0: Not inverted | L_PhiIntegrator_1: Invert. gearbox nSet_v |
| C01067/2 | 0: Not inverted | L_DFSET_1: Invert. gearbox nSet_v |
| C01067/3 | 0: Not inverted | L_CalcDiameter_1: Invert. gearbox nMotorSpeedAct_v |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C01069

| Parameter \| Name: C01069 \| L_DFSET: Ramp settings | | | Data type: UNSIGNED_16 Index: $23506_d$ = $5BD2_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | Incr./ms | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01069/1 | 10 incr./ms | | L_DFSET_1: Ramp SpeedTrim |
| C01069/2 | 100 incr./ms | | L_DFSET_1: Ramp angle compensation |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01070

| Parameter \| Name: C01070 \| L_DFSET: Angular trimming | | | Data type: INTEGER_32 Index: $23505_d$ = $5BD1_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -134217728 | Incr. | 134217728 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01070/1 | 0 incr. | | L_DFSET_1: Speed-dependent angle adjustment |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01071

| Parameter \| Name: C01071 \| L_DFSET: Following error limit | | | Data type: UNSIGNED_32 Index: $23504_d$ = $5BD0_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 10 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01071/1 | 32768 incr. | | L_DFSET_1: Following error limit |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01072**

| Parameter \| Name:<br>**C01072 \| L_DFSET: Multiplier - angular trimming** | | | Data type: INTEGER_16<br>Index: $23503_d$ = $5BCF_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -20000 | | 20000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01072/1 | 1 | | L_DFSET_1: Multiplier - angular trimming |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01073**

| Parameter \| Name:<br>**C01073 \| L_DFSET: Evaluation - setpoint angle integrator** | | Data type: UNSIGNED_8<br>Index: $23502_d$ = $5BCE_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | Evaluation with gearbox factor | |
| 1 | Evaluation without gearbox factor | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01073/1 | 0: Evaluation with gearbox factor | L_DFSET_1: Evaluation - setpoint angle integrator |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01074**

| Parameter \| Name:<br>**C01074 \| L_DFSET: Zero pulse divider** | | | Data type: UNSIGNED_16<br>Index: $23501_d$ = $5BCD_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 16384 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01074/1 | 0 | | L_DFSET_1: Divider for actual value zero pulse |
| C01074/2 | 0 | | L_DFSET_1: Divider for setpoint zero pulse |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01075**

| Parameter \| Name:<br>**C01075 \| L_DFSET: Synchronisation mode** | | Data type: UNSIGNED_8<br>Index: $23500_d$ = $5BCC_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | Inactive | |
| 1 | Permanent sync. without enable (bZeroPulse) | |
| 2 | Permanent sync. with enable (bZeroPulse) | |
| 10 | 1x sync. - Angular diff. shortest path setpoint | |
| 11 | 1x Sync. - Angular diff. Cw | |
| 12 | 1x Sync. - Angular diff. Ccw | |
| 13 | 1x Sync. - Angular diff. shortest path act. value | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01075/1 | 0: Inactive | L_DFSET_1: Synchronisation mode |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C01076

| Parameter \| Name:<br>**C01076 \| L_DFRFG: Times** | | Data type: UNSIGNED_32<br>Index: $23499_d$ = $5BCB_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0.000 | s | 999.900 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01076/1 | 1.000 s | L_DFRFG_1: Acceleration and deceleration time |
| C01076/2 | 0.000 s | L_DFRFG_1: Deceleration time for quick stop |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

## C01077

| Parameter \| Name:<br>**C01077 \| L_DFRFG: Max. speed-up** | | Data type: INTEGER_16<br>Index: $23498_d$ = $5BCA_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 1 | rpm | 15000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01077/1 | 3000 rpm | L_DFRFG_1: Max. speed-up |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01078

| Parameter \| Name:<br>**C01078 \| L_DFRFG: Following error** | | Data type: UNSIGNED_32<br>Index: $23497_d$ = $5BC9_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 10 | Incr. | 2000000000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01078/1 | 2000000000 incr. | L_DFRFG_1: Following error limit |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01079

| Parameter \| Name:<br>**C01079 \| L_DFRFG: Synchronisation window** | | Data type: UNSIGNED_16<br>Index: $23496_d$ = $5BC8_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | Incr. | 65535 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01079/1 | 100 incr. | L_DFRFG_1: Synchronisation window (position) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01080

| Parameter \| Name:<br>**C01080 \| L_DFRFG: Offset** | | Data type: INTEGER_32<br>Index: $23495_d$ = $5BC7_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -2147483647 | Incr. | 2147483647 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01080/1 | 0 incr. | L_DFRFG_1: Offset |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01081**

| Parameter \| Name: **C01081 \| L_DFRFG: Sync. direction / TP function** | | Data type: UNSIGNED_8 Index: $23494_d$ = $5BC6_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 1 | cw/ccw - without TP | |
| 2 | cw - without TP | |
| 3 | ccw - without TP | |
| 4 | cw/ccw - with TP | |
| 5 | cw - with TP | |
| 6 | ccw - with TP | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01081/1 | 1: cw/ccw - without TP | L_DFRFG_1: Sync. direction / TP function |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01082**

| Parameter \| Name: **C01082 \| LS_WriteParamList: Execute Mode** | | Data type: UNSIGNED_8 Index: $23493_d$ = $5BC5_h$ |
|---|---|---|
| Parameter change-over: Selection of the activation method | | |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| 0 | **by Execute** | The writing of the parameter list is activated by a FALSE/TRUE edge at the *bExecute* input. |
| 1 | by Input Select | The writing of the parameter list is carried out if a change is made at the select inputs and if the controller is initialised. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01083**

Parameter | Name: **C01083 | LS_WriteParamList: Error status** — Data type: UNSIGNED_16, Index: $23492_d$ = $5BC4_h$

Parameter change-over: Error status:

- 0 = no error
- 33803 = Invalid data type (e.g. STRING)
- 33804 = limit violation
- 33806 = invalid code
- 33813 = no element of the selection list
- 33815 = writing of the parameter not permitted
- 33816 = writing of the parameter only permitted if controller is inhibited
- 33829 = invalid subcode
- 33865 = no parameter with subcodes

| **Display range** (min. value \| unit \| max. value) | | |
|---|---|---|
| 0 | | 34000 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01084**

Parameter | Name: **C01084 | LS_WriteParamList: Error line** — Data type: UNSIGNED_8, Index: $23491_d$ = $5BC3_h$

Parameter change-over: Display of the number of list entry where the error occurred (in connection with the value set selected via *bSelectWriteValue_1* and *bSelectWriteValue_2*).

| **Display range** (min. value \| unit \| max. value) | | |
|---|---|---|
| 0 | | 32 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01085**

| Parameter \| Name: **C01085 \| LS_WriteParamList: Index** | | Data type: INTEGER_32 Index: $23490_d$ = $5BC2_h$ |
|---|---|---|
| Parameter change-over: Parameter for entry 1 ... 32 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0.000 | | 16000.000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01085/1 | 0.000 | Parameter for entries 1 ... 32<br>• Format: <code number>.<subcode number><br>• Examples: "12.000" = C00012; "26.001" = C00026/1 |
| C01085/... | | |
| C01085/32 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | |

**C01086**

| Parameter \| Name: **C01086 \| LS_WriteParamList: WriteValue_1** | | Data type: INTEGER_32 Index: $23489_d$ = $5BC1_h$ |
|---|---|---|
| Parameter change-over: Parameter values - value set 1 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -2147483647 | | 2147483647 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01086/1 | 0 | Parameter values - value set 1<br>• Parameter values for the parameters defined in C01085/1 ... 32. |
| C01086/... | | |
| C01086/32 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01087**

| Parameter \| Name: **C01087 \| LS_WriteParamList: WriteValue_2** | | Data type: INTEGER_32 Index: $23488_d$ = $5BC0_h$ |
|---|---|---|
| Parameter change-over: Parameter values - value set 2 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -2147483647 | | 2147483647 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01087/1 | 0 | Parameter values - value set 2<br>• Parameter values for the parameters defined in C01085/1 ... 32. |
| C01087/... | | |
| C01087/32 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01088**

| Parameter \| Name: **C01088 \| LS_WriteParamList: WriteValue_3** | | Data type: INTEGER_32 Index: $23487_d$ = $5BBF_h$ |
|---|---|---|
| Parameter change-over: Parameter values - value set 3 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -2147483647 | | 2147483647 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01088/1 | 0 | Parameter values - value set 3<br>• Parameter values for the parameters defined in C01085/1 ... 32. |
| C01088/... | | |
| C01088/32 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01089**

| Parameter \| Name: **C01089 \| LS_WriteParamList: WriteValue_4** | | | Data type: INTEGER_32 Index: $23486_d$ = $5BBE_h$ |
|---|---|---|---|
| Parameter change-over: Parameter values - value set 4 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -2147483647 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01089/1 | 0 | | Parameter values - value set 4<br>• Parameter values for the parameters defined in C01085/1 ... 32. |
| C01089/... | | | |
| C01089/32 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01090**

| Parameter \| Name: **C01090 \| LS_ParReadWrite 1-6: Index** | | | Data type: INTEGER_32 Index: $23485_d$ = $5BBD_h$ |
|---|---|---|---|
| Parameter to be read or written.<br>• Format: <code number>,<subcode number><br>• For a setting of "0,000", inputs *wParIndex* and *wParSubindex* are effective for addressing purposes instead. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | | 16000.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01090/1 | 0.000 | | LS_ParReadWrite_1: Index |
| C01090/2 | 0.000 | | LS_ParReadWrite_2: Index |
| C01090/3 | 0.000 | | LS_ParReadWrite_3: Index |
| C01090/4 | 0.000 | | LS_ParReadWrite_4: Index |
| C01090/5 | 0.000 | | LS_ParReadWrite_5: Index |
| C01090/6 | 0.000 | | LS_ParReadWrite_6: Index |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01091**

| Parameter \| Name: **C01091 \| LS_ParReadWrite 1-6: Cycle time** | | Data type: UNSIGNED_16 Index: $23484_d$ = $5BBC_h$ |
|---|---|---|
| Time interval for cyclic reading/writing | | |
| **Selection list** | | |
| 0 | 0 (by Execute) | |
| 20 | 20 ms | |
| 50 | 50 ms | |
| 100 | 100 ms | |
| 200 | 200 ms | |
| 500 | 500 ms | |
| 1000 | 1 s | |
| 2000 | 2 s | |
| 5000 | 5 s | |
| 10000 | 10 s | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01091/1 | 0: 0 (by Execute) | LS_ParReadWrite_1: Cycle time |
| C01091/2 | 0: 0 (by Execute) | LS_ParReadWrite_2: Cycle time |
| C01091/3 | 0: 0 (by Execute) | LS_ParReadWrite_3: Cycle time |
| C01091/4 | 0: 0 (by Execute) | LS_ParReadWrite_4: Cycle time |
| C01091/5 | 0: 0 (by Execute) | LS_ParReadWrite_5: Cycle time |
| C01091/6 | 0: 0 (by Execute) | LS_ParReadWrite_6: Cycle time |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01092**

| Parameter \| Name: **C01092 \| LS_ParReadWrite 1-6: FailState** | | | Data type: UNSIGNED_16 Index: $23483_d$ = $5BBB_h$ |
|---|---|---|---|
| Error status:<br>• 0 = no error<br>• 33803 = Invalid data type (e.g. STRING)<br>• 33804 = limit violation<br>• 33806 = invalid code<br>• 33813 = no element of the selection list<br>• 33815 = writing of the parameter not permitted<br>• 33816 = writing of the parameter only permitted if controller is inhibited<br>• 33829 = invalid subcode<br>• 33865 = no parameter with subcodes | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 34000 | |
| **Subcodes** | | | **Info** |
| C01092/1 | | | LS_ParReadWrite_1: Error status |
| C01092/2 | | | LS_ParReadWrite_2: Error status |
| C01092/3 | | | LS_ParReadWrite_3: Error status |
| C01092/4 | | | LS_ParReadWrite_4: Error status |
| C01092/5 | | | LS_ParReadWrite_5: Error status |
| C01092/6 | | | LS_ParReadWrite_6: Error status |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01093

| Parameter \| Name: C01093 - LS_ParReadWrite 1-6: Arithmetic mode | | Data type: UNSIGNED_8 Index: $23482_d$ = $5BBA_h$ |
|---|---|---|
| The integrated arithmetic function allows for easy arithmetic conversion of the process value to be written or which was read into the format of the target parameter via parameterisable factors and without the need for an additional arithmetic FB. | | |
| **Selection list** | | |
| 0 | No arithmetic | |
| 1 | In16Bit: LW=+/-32767 | |
| 2 | In16Bit: HW=+/-; LW=0..65535 | |
| 3 | In32Bit: HW_LW=+/-2147483647 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01093/1 | 0: no arithmetic | LS_ParReadWrite_1: Arithmetic mode |
| C01093/2 | 0: no arithmetic | LS_ParReadWrite_2: Arithmetic mode |
| C01093/3 | 0: no arithmetic | LS_ParReadWrite_3: Arithmetic mode |
| C01093/4 | 0: no arithmetic | LS_ParReadWrite_4: Arithmetic mode |
| C01093/5 | 0: no arithmetic | LS_ParReadWrite_5: Arithmetic mode |
| C01093/6 | 0: no arithmetic | LS_ParReadWrite_6: Arithmetic mode |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01094

| Parameter \| Name: C01094 - LS_ParReadWrite 1-6: Numerator | | | Data type: INTEGER_16 Index: $23481_d$ = $5BB9_h$ |
|---|---|---|---|
| Arithmetic function: Factor (numerator) for internal conversion in arithmetic modes 1 ... 3. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -32767 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01094/1 | 1 | | LS_ParReadWrite_1: Numerator |
| C01094/2 | 1 | | LS_ParReadWrite_2: Numerator |
| C01094/3 | 1 | | LS_ParReadWrite_3: Numerator |
| C01094/4 | 1 | | LS_ParReadWrite_4: Numerator |
| C01094/5 | 1 | | LS_ParReadWrite_5: Numerator |
| C01094/6 | 1 | | LS_ParReadWrite_6: Numerator |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01095

| Parameter \| Name: C01095 - LS_ParReadWrite 1-6: Denominator | | | Data type: INTEGER_16 Index: $23480_d$ = $5BB8_h$ |
|---|---|---|---|
| Arithmetic function: Factor (denominator) for internal conversion in arithmetic modes 1 ... 3. | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | | 32767 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01095/1 | 1 | | LS_ParReadWrite_1: Denominator |
| C01095/2 | 1 | | LS_ParReadWrite_2: Denominator |
| C01095/3 | 1 | | LS_ParReadWrite_3: Denominator |
| C01095/4 | 1 | | LS_ParReadWrite_4: Denominator |
| C01095/5 | 1 | | LS_ParReadWrite_5: Denominator |
| C01095/6 | 1 | | LS_ParReadWrite_6: Denominator |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01100

| Parameter \| Name:<br>**C01100 \| Function L_Counter 1-3** | | Data type: UNSIGNED_8<br>Index: $23475_d$ = $5BB3_h$ |
|---|---|---|
| Selection of reset function | | |
| **Selection list** | | |
| 0 | Normal counting | |
| 1 | Auto reset | |
| 2 | Manual reset | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01100/1 | 0: Normal counting | L_Counter_1: Function |
| C01100/2 | 0: Normal counting | L_Counter_2: Function |
| C01100/3 | 0: Normal counting | L_Counter_3: Function |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01101

| Parameter \| Name:<br>**C01101 \| Comparison L_Counter 1-3** | | Data type: UNSIGNED_8<br>Index: $23474_d$ = $5BB2_h$ |
|---|---|---|
| Selection of comparison operation | | |
| **Selection list** | | |
| 0 | Greater than or equal to | |
| 1 | Less than or equal to | |
| 2 | equal to | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01101/1 | 0: Greater than or equal to | L_Counter_1: Comparison |
| C01101/2 | 0: Greater than or equal to | L_Counter_2: Comparison |
| C01101/3 | 0: Greater than or equal to | L_Counter_3: Comparison |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01110

| Parameter \| Name:<br>**C01110 \| LS_MultiEncoder: Solid measure** | | Data type: UNSIGNED_8<br>Index: $23465_d$ = $5BA9_h$ |
|---|---|---|
| From version 02.00.00 | | |
| | | ▶ Encoder/feedback system: SSI encoder |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **rotatively unipolar** | |
| 1 | linearly unipolar | |
| 2 | rotatively bipolar | |
| 3 | linearly bipolar | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01111

| Parameter \| Name:<br>**C01111 \| SSI: Encoder constant** | | | Data type: INTEGER_32<br>Index: $23464_d$ = $5BA8_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| | | | ▶ Encoder/feedback system: SSI encoder |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 1 | | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01111/1 | 1 | | LS_MultiEncoder: Difference - traverse path |
| C01111/2 | 1 | | LS_MultiEncoder: Difference - encoder value |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01112

| Parameter \| Name: **C01112 \| LS_MultiEncoder: PosValues** | | | Data type: INTEGER_32<br>Index: $23463_d$ = $5BA7_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01112/1 | 0.0000 units | | LS_MultiEncoder: Position offset |
| C01112/2 | -214748.3647 units | | LS_MultiEncoder: Data area min |
| C01112/3 | 214748.3647 units | | LS_MultiEncoder: Data area max |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

## C01119

| Parameter \| Name: **C01119 \| LS_MultiEncoder: Current position** | | | Data type: INTEGER_32<br>Index: $23456_d$ = $5BA0_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| | | | ▸ Encoder/feedback system: Multi-Encoder |
| **Display range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01119/1 | | | LS_MultiEncoder: Current position |
| C01119/2 | | | LS_MultiEncoder: Maximum travel distance |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000

## C01120

Parameter | Name: **C01120 | Sync signal source** — Data type: UNSIGNED_8, Index: $23455_d$ = $5B9F_h$

Selection of the signal source for device synchronisation
- Basically, only one source is allowed to synchronise the drive.

▸ Synchronisation of the internal time base

| Selection list (Lenze setting printed in bold) | | Info |
|---|---|---|
| **0** | **Off** | Synchronisation off |
| 1 | CAN on board | Synchronisation via CAN bus<br>▸ Sync telegram |
| 2 | AxisBusIO | Synchronisation via axis bus<br>▸ Axis bus |
| 4 | MCI | Synchronisation via MCI (communication module) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01121

Parameter | Name: **C01121 | Sync cycle time setpoint** — Data type: UNSIGNED_16, Index: $23454_d$ = $5B9E_h$

Cycle time setpoint for device synchronisation
- Time at which the phase-locking loop (PLL) in the controller expects the synchronisation signals to arrive.
- The cycle time setpoint must be set according to the cycle of the respective synchronisation source.

**Note:**
- Only integer multiples of 1000 μs can be set.
- Intelligent communication modules usually define the cycle time setpoint derived from the bus cycle. In this case, a manual change is not possible.

Example: For the CAN bus, a distance of 2 ms has been set between two synchronisation signals. If the CAN bus is to be used as synchronisation source, a synchronisation cycle of 2000 μs must be set in C01121.

▸ Synchronisation of the internal time base

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 1000 | μs | 20000 | **1000 μs** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01122**

| Parameter \| Name:<br>**C01122 \| Sync phase position** | Data type: UNSIGNED_16<br>Index: $23453_d$ = $5B9D_h$ |
|---|---|

Phase position for device synchronisation

- The phase position determines the zero-time of the internal system cycle with regard to the synchronisation signal (bus cycle). Since PDO processing is an inherent part of the system part of the application, the instant of acceptance of the PDOs is postponed as well by a changed phase position.
- With a setting = 0, the system cycle starts simultaneously with the synchronisation signal.
- With a setting > 0, the internal system cycle starts earlier by the set time with regard to the synchronisation signal (the phase position acts negatively).
- Intelligent communication modules define the optimal time with activated synchronisation by themselves. In this case, a manual change is not possible.
- The decisive factor for defining C01122 is the time where all nodes are provided with valid PDOs.

Example: If the phase position is set to 550 µs, the system part of the application starts 550 µs before the arrival of the synchronisation signal.

▸ Synchronisation of the internal time base

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0 | µs | 1000 | **0 µs** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01123**

| Parameter \| Name:<br>**C01123 \| Sync window** | Data type: UNSIGNED_16<br>Index: $23452_d$ = $5B9C_h$ |
|---|---|

Time slot for monitoring the synchronisation signal or the phase position

- The synchronisation signal or the current phase position must be within this time slot around the corresponding expected value (C01122).
- With the setting "1000 µs" there will be no monitoring.

▸ Synchronisation of the internal time base

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0 | µs | 10000 | **100 µs** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01124**

| Parameter \| Name:<br>**C01124 \| Sync correction width** | Data type: UNSIGNED_8<br>Index: $23451_d$ = $5B9B_h$ |
|---|---|

Correction increment for device synchronisation

- If the cycle times of the synchronisation signal differs and phase-locken loop (PLL) differ from each other, this setting defines the measure the phase-locking loop is reset with.
- If synchronisation is not reached, select a higher correction constant.
- The optimum setting depends on quartz precision and must be determined empirically if required.

▸ Synchronisation of the internal time base

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| 1 | 80ns |
| 2 | 160ns |
| 3 | 240ns |
| **4** | **320ns** |
| 5 | 400ns |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01130**

| Parameter \| Name: **C01130 \| LS_RetainData: Selection** | | Data type: UNSIGNED_16 Index: $23445_d$ = $5B95_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. hex value \| max. hex value) | | |
| 0x0000 | | 0xFFFF |
| **Value is bit-coded:** (☑ = bit set) | | |
| Bit 0 ☐ | wIn1 | |
| Bit 1 ☐ | wIn2 | |
| Bit 2 ☐ | wIn3 | |
| Bit 3 ☐ | wIn4 | |
| Bit 4 ☐ | dnIn1 | |
| Bit 5 ☐ | dnIn2 | |
| Bit 6 ☐ | dnIn3 | |
| Bit 7 ☐ | dnIn4 | |
| Bit 8 ☐ | bIn1 | |
| Bit 9 ☐ | bIn2 | |
| Bit 10 ☐ | bIn3 | |
| Bit 11 ☐ | bIn4 | |
| Bit 12 ☐ | Reserved | |
| Bit 13 ☐ | Reserved | |
| Bit 14 ☐ | Reserved | |
| Bit 15 ☐ | Reserved | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01130/1 | 0 | LS_RetainData: Selection bSetRetain_1 |
| C01130/2 | 0 | LS_RetainData: Selection bSetRetain_2 |
| C01130/3 | 0 | LS_RetainData: Selection bSetRetain_3 |
| C01130/4 | 0 | LS_RetainData: Selection bLoadParams |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C01131**

| Parameter \| Name: **C01131 \| LS_RetainData: 16Bit data** | | Data type: UNSIGNED_16 Index: $23444_d$ = $5B94_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | | 65535 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01131/1 | 0 | LS_RetainData: wValue1 |
| C01131/2 | 0 | LS_RetainData: wValue2 |
| C01131/3 | 0 | LS_RetainData: wValue3 |
| C01131/4 | 0 | LS_RetainData: wValue4 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C01132

| Parameter \| Name: **C01132 \| LS_RetainData: 32Bit data** | | Data type: INTEGER_32 Index: $23443_d$ = $5B93_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Setting range** (min. value \| unit \| max. value) | | |
| -2147483647 | | 2147483647 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01132/1 | 0 | LS_RetainData: dnValue1 |
| C01132/2 | 0 | LS_RetainData: dnValue2 |
| C01132/3 | 0 | LS_RetainData: dnValue3 |
| C01132/4 | 0 | LS_RetainData: dnValue4 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01133

| Parameter \| Name: **C01133 \| LS_RetainData: Bool data** | | Data type: UNSIGNED_8 Index: $23442_d$ = $5B92_h$ |
|---|---|---|
| From version 02.00.00 | | |
| **Selection list** | | |
| 0 | False | |
| 1 | True | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01133/1 | 0: FALSE | LS_RetainData: bValue1 |
| C01133/2 | 0: FALSE | LS_RetainData: bValue2 |
| C01133/3 | 0: FALSE | LS_RetainData: bValue3 |
| C01133/4 | 0: FALSE | LS_RetainData: bValue4 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01138

| Parameter \| Name: **C01138 \| L_Transient 1-4: Function** | | Data type: UNSIGNED_8 Index: $23437_d$ = $5B8D_h$ |
|---|---|---|
| Selection of edge evaluation | | |
| **Selection list** | | |
| 0 | High edge | |
| 1 | Low edge | |
| 2 | High and low edge | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01138/1 | 0: High edge | L_Transient_1: Function |
| C01138/2 | 0: High edge | L_Transient_2: Function |
| C01138/3 | 0: High edge | L_Transient_3: Function |
| C01138/4 | 0: High edge | L_Transient_4: Function |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01139**

| Parameter \| Name: **C01139 \| L_Transient 1-4: Pulse duration** | | | Data type: UNSIGNED_16 Index: $23436_d$ = $5B8C_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 60.000 | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C01139/1 | 0.000 s | L_Transient_1: Pulse duration | |
| C01139/2 | 0.000 s | L_Transient_2: Pulse duration | |
| C01139/3 | 0.000 s | L_Transient_3: Pulse duration | |
| C01139/4 | 0.000 s | L_Transient_4: Pulse duration | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C01140**

| Parameter \| Name: **C01140 \| L_Transient 5-8: Function** | | Data type: UNSIGNED_8 Index: $23435_d$ = $5B8B_h$ |
|---|---|---|
| Selection of edge evaluation | | |
| **Selection list** | | |
| 0 | High edge | |
| 1 | Low edge | |
| 2 | High and low edge | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01140/1 | 0: High edge | L_Transient_5: Function |
| C01140/2 | 0: High edge | L_Transient_6: Function |
| C01140/3 | 0: High edge | L_Transient_7: Function |
| C01140/4 | 0: High edge | L_Transient_8: Function |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01141**

| Parameter \| Name: **C01141 \| L_Transient 5-8 pulse duration** | | | Data type: UNSIGNED_16 Index: $23434_d$ = $5B8A_h$ |
|---|---|---|---|
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 60.000 | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C01141/1 | 0.000 s | L_Transient_5: Pulse duration | |
| C01141/2 | 0.000 s | L_Transient_6: Pulse duration | |
| C01141/3 | 0.000 s | L_Transient_7: Pulse duration | |
| C01141/4 | 0.000 s | L_Transient_8: Pulse duration | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C01150**

| Parameter \| Name: **C01150 \| L_PhaseIntK: Function** | | | Data type: UNSIGNED_8 Index: $23425_d$ = $5B81_h$ |
|---|---|---|---|
| Loading behaviour of the integrator | | | |
| **Selection list** | | **Info** | |
| 0 | Loading with level | Load integrator with TRUE level at the input *bLoad* | |
| 1 | Loading with edge | Load integrator with FALSE/TRUE edge at the input *bLoad* | |
| 2 | Loading with level + reset | Load integrator when the comparison value (C01151) is reached or with TRUE level at the *bLoad* input | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C01150/1 | 0: Loading with level | L_PhaseIntK_1: Function | |
| C01150/2 | 0: Loading with level | L_PhaseIntK_2: Function | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01151**

| Parameter \| Name: **C01151 \| L_PhaseIntK: Compare** | | | Data type: INTEGER_32 Index: $23424_d$ = $5B80_h$ |
|---|---|---|---|
| Comparison value | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 2147418112 | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C01151/1 | 0 | L_PhaseIntK_1: Comparison value | |
| C01151/2 | 0 | L_PhaseIntK_2: Comparison value | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01190**

| Parameter \| Name: **C01190 \| Motor thermal sensor** | | | Data type: UNSIGNED_8 Index: $23385_d$ = $5B59_h$ |
|---|---|---|---|
| Selection of the thermal sensor for monitoring the motor temperature ▸ Encoder/feedback system: Motor temperature monitoring (KTY) | | | |
| **Selection list** | | **Info** | |
| 0 | KTY83-110 | Lenze standard KTY83-110 (MDSKX, MCS06) | |
| 1 | Spec. characteristic | Characteristic defined via C01191 and C01192 | |
| 2 | KTY83-110 + 2 x PTC | Lenze standard KTY83-110 + 2 x PTC 150°C (MCS09-MCS19) | |
| 4 | KTY84-130 | Lenze standard KTY83-130 | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C01190/1 | 0: KTY83-110 | Type of motor temperature sensor resolver | |
| C01190/2 | 0: KTY83-110 | Type of motor temperature sensor MultiEncoder | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C01191

| Parameter \| Name: **C01191 \| PTC characteristic: Temperature 1/2** | | Data type: UNSIGNED_8 Index: $23384_d$ = $5B58_h$ |
|---|---|---|

The spec. thermal sensor characteristic is selected through the setting C01190="1"

▶ Encoder/feedback system: Motor temperature monitoring (KTY)

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | °C | 255 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01191/1 | 100 °C | | PTC characteristic: Temperature 1 |
| C01191/2 | 150 °C | | PTC characteristic: Temperature 2 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01192

| Parameter \| Name: **C01192 \| PTC characteristic: Resistance 1/2** | | Data type: INTEGER_32 Index: $23383_d$ = $5B57_h$ |
|---|---|---|

The spec. thermal sensor characteristic is selected through the setting C01190="1"

▶ Encoder/feedback system: Motor temperature monitoring (KTY)

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | Ohm | 30000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01192/1 | 1070 Ohms | | PTC characteristic: Resistance 1 |
| C01192/2 | 2225 Ohms | | PTC characteristic: Resistance 2 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01193

| Parameter \| Name: **C01193 \| Feedback system motor temperature** | | Data type: UNSIGNED_8 Index: $23382_d$ = $5B56_h$ |
|---|---|---|

Selection of the feedback system for monitoring the motor temperature

▶ Encoder/feedback system: Motor temperature monitoring (KTY)

| Selection list (Lenze setting printed in bold) | |
|---|---|
| **0** | **Speed feedback** |
| 1 | Resolver input |
| 2 | Encoder input |
| 5 | Res. and enc. parallel |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01201

| Parameter \| Name: **C01201 \| MCK: Cycle** | | Data type: INTEGER_32 Index: $23374_d$ = $5B4E_h$ |
|---|---|---|

Cycle for Modulo measuring system

- When the cycle is set to 0 units (Lenze setting), the traversing range is unlimited (classical measuring system).

▶ Activation of the Modulo measuring system

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0.0000 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01201/1 | 0.0000 units | | MCK: Cycle |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

C01202

| Parameter \| Name: **C01202 \| MCK: iM motor/process** | | Data type: UNSIGNED_16 Index: 23373$_d$ = 5B4D$_h$ |
|---|---|---|

Gearbox factor - motor
- Entry of the gearbox factor as numerator/denominator ratio (numerator = motor speed and denominator = output speed of gearbox) or from the number of teeth of the gearbox arrangement.

Machine parameter

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 1 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01202/1 | 1 | | MCK: iM: Numerator gearbox factor Z2 |
| C01202/2 | 1 | | MCK: iM: Denominator gearbox factor Z1 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

C01203

| Parameter \| Name: **C01203 \| MCK: iG motor/position encoder** | | Data type: UNSIGNED_16 Index: 23372$_d$ = 5B4C$_h$ |
|---|---|---|

Gearbox factor - position encoder
- Entry of the gearbox factor as numerator/denominator ratio, with numerator = motor speed and denominator = position encoder speed.

Machine parameter

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 1 | | 65535 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01203/1 | 1 | | MCK: iG: Numerator (motor speed) |
| C01203/2 | 1 | | MCK: iG: Denominator (encoder speed) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

C01204

| Parameter \| Name: **C01204 \| MCK: Feed constant** | | Data type: INTEGER_32 Index: 23371$_d$ = 5B4B$_h$ |
|---|---|---|

The feed constant corresponds to the machine movement for one revolution of the gearbox output shaft.
- The value is entered in application units referred to one revolution.

Machine parameter

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.0001 | units/rev. | 214748.3647 | **360.0000 units/rev.** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

C01205

| Parameter \| Name: **C01205 \| MCK: Position resolution** | | Data type: INTEGER_32 Index: 23370$_d$ = 5B4A$_h$ |
|---|---|---|

Display of the number of increments corresponding to one unit.
- 1 motor revolution ≡ 65536 increments

Machine parameter

| Display range (min. value \| unit \| max. value) | | |
|---|---|---|
| 0.0000 | Incr./unit | 214748.3647 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

## C01206

| Parameter \| Name: **C01206 \| MCK: Mounting direction** | | Data type: UNSIGNED_8 Index: $23369_d$ = $5B49_h$ |
|---|---|---|
| Inversion for mirrored motor and encoder mounting | | |
| | | ▸ Machine parameter |
| **Selection list** | | |
| 0 | not inverted | |
| 1 | inverted | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01206/1 | 0: Not inverted | Motor mounting direction<br>• Setting for motor mounting turned by 180°. |
| C01206/2 | 0: Not inverted | Position encoder mounting direction<br>• Setting for position encoder system mounting turned by 180°. |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C01210

| Parameter \| Name: **C01210 \| MCK: Current positions** | | | Data type: INTEGER_32 Index: $23365_d$ = $5B45_h$ |
|---|---|---|---|
| Display of current position data of the Motion Control Kernel | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01210/1 | | | MCK: Feed<br>• Display of the current feed for positioning profiles as a relative distance. |
| C01210/2 | | | MCK: Set position<br>• Display of the current setpoint position calculated by the MCK. |
| C01210/3 | | | MCK: Actual position<br>• Display of the current actual position calculated by an optional encoder system. |
| C01210/4 | | | MCK: Following error<br>• Display of the current following error as a difference between setpoint position and actual position. |
| C01210/5 | | | MCK: Positioning accuracy<br>• Display of the current positioning accuracy referred to the actual number of increments of the position encoder. |
| C01210/6 | | | MCK: Target position |
| C01210/7 | | | MCK: Modulo position |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

## C01211

| Parameter \| Name: **C01211 \| MCK: Speed** | | | Data type: INTEGER_32 Index: $23364_d$ = $5B44_h$ |
|---|---|---|---|
| Display of the current speed data | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s | 214748.3647 | |
| **Subcodes** | | | **Info** |
| C01211/1 | | | MCK: Max. traversing speed 100%_C11<br>• Display of the maximum traversing speed based on the reference speed set in C00011. |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

## C01213

| Parameter \| Name: **C01213 \| MCK: Max. traversing distance** | | | Data type: INTEGER_32 Index: $23362_d$ = $5B42_h$ |
|---|---|---|---|
| Display of current position limits | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| -2147480000 | units | 2147480000 | |
| **Subcodes** | | | **Info** |
| C01213/1 | | | MCK: Max. traversing distance<br>• Display of the maximum distance to be traversed referred to the 32-bit display area (max. 2147483647 increments). |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01215

| Parameter \| Name: **C01215 \| MCK: Following error** | | | Data type: INTEGER_32 Index: $23360_d$ = $5B40_h$ |
|---|---|---|---|
| Limits for following error monitoring system<br>▶ Following error monitoring system | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0000 | units | 214748.0000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01215/1 | 0.0000 units | | MCK: Following error limit 1<br>• First limit of the maximum following error for monitoring and response activation. |
| C01215/2 | 0.0000 units | | MCK: Following error limit 2<br>• Second limit of the maximum following error for monitoring and response activation. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

## C01216

| Parameter \| Name: **C01216 \| MCK: Positioning setting** | | | Data type: UNSIGNED_8 Index: $23359_d$ = $5B3F_h$ |
|---|---|---|---|
| ▶ Positioning | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x01** (decimal: 1) |
| **Value is bit-coded:** (☑ = bit set) | | | |
| Bit 0 ☑ | PosAbort at PosInit | | |
| Bit 1 ☐ | PosExecute active at PosInit | | |
| Bit 2 ☐ | Reserved | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | ProfilStart at PosInit | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

### C01218

| Parameter \| Name: **C01218 \| PosFollower: Setting** | | | Data type: UNSIGNED_8 Index: $23357_d$ = $5B3D_h$ |
|---|---|---|---|
| Settings for "Position follower" mode | | | |
| | | | ▸ Position follower |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x0C** (decimal: 12) |
| **Value is bit-coded:** (☑ = bit set) | | | **Info** |
| Bit 0 ☐ | Speed FF control.: nSpeedSetValue_a | | "1" ≡ speed feedforward control value comes from main setpoint *nSpeedSetValue_a* |
| Bit 1 ☐ | Speed FF control: nSpeedAddValue_v | | "1" ≡ speed feedforward control value comes from additive speed value *nSpeedAddValue_v* |
| Bit 2 ☑ | HW limit switch on | | "1" ≡ Travel range monitoring via hardware limit switch is active. The error response can be parameterised in C00595/1 and C00595/2. |
| Bit 3 ☑ | SW limit switch on | | "1" ≡ Travel range monitoring via parameterised software limit positions is active. The error response can be parameterised in C00595/3 and C00595/4. |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

### C01219

| Parameter \| Name: **C01219 \| MCK: Speed follower setting** | | | Data type: UNSIGNED_8 Index: $23356_d$ = $5B3C_h$ |
|---|---|---|---|
| Settings for "Speed follower" mode | | | |
| | | | ▸ Speed follower |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x0C** (decimal: 12) |
| **Value is bit-coded:** (☑ = bit set) | | | **Info** |
| Bit 0 ☐ | Reserved | | |
| Bit 1 ☐ | Reserved | | |
| Bit 2 ☑ | HW limit switch on | | "1" ≡ Travel range monitoring via hardware limit switch is active. The error response can be parameterised in C00595/1 and C00595/2. |
| Bit 3 ☑ | SW limit switch on | | "1" ≡ Travel range monitoring via parameterised software limit positions is active. The error response can be parameterised in C00595/3 and C00595/4. |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Position controller off | | "1" ≡ Position controller is deactivated. Thus, the compensation of the following error is switched off. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01221**

| Parameter \| Name: **C01221 \| MCK: Ref. mode** | Data type: UNSIGNED_8 Index: $23354_d$ = $5B3A_h$ |
|---|---|

Selection of the homing mode.

▸ Homing

| **Selection list** (Lenze setting printed in bold) | | **Info** |
|---|---|---|
| 6 | >_Rn_>_TP | Positive direction - via home mark (neg. edge) - continuing in positive direction - to TP |
| 7 | <_Rn_<_TP | Negative direction - via home mark (neg. edge) - continuing in negative direction - to TP |
| 8 | >_TP | Positive direction - to TP |
| 9 | <_TP | Negative direction - to TP |
| 10 | >_Lp_<_TP | Positive direction - reversing on pos. limit switch - to TP |
| 11 | <_Ln_>_TP | Negative direction - reversing on neg. limit switch - to TP |
| **12** | **>_Lp** | Positive direction - to pos. limit switch |
| 13 | <_Ln | Negative direction - to neg. limit switch |
| 14 | >_Mlim | Positive direction towards torque limit (C01222) |
| 15 | <_Mlim | Negative direction towards torque limit (C01222) |
| 100 | SetRef | Direct acceptance and setting of the home position<br>• The measuring system is set based on the home position parameterised in C01227/2 when the drive is at standstill. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01222**

| Parameter \| Name: **C01222 \| MCK: Ref. M limit mode 14/15** | Data type: INTEGER_16 Index: $23353_d$ = $5B39_h$ |
|---|---|

Torque limit for homing modes 14 and 15 (homing towards positive stop)

- 100 % ≡ maximum torque (C00057)

▸ Homing

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0.00 | % | 199.99 | **10.00 %** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C01223**

| Parameter \| Name: **C01223 \| MCK: Ref. waiting time mode 14/15** | Data type: UNSIGNED_16 Index: $23352_d$ = $5B38_h$ |
|---|---|

Blocking time for homing modes 14 and 15 (homing towards positive stop)

- The reference is set if an excess of the torque limit set in C01222 has been deteced over a time period defined here.

▸ Homing

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0 | ms | 65000 | **100 ms** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C01224

| Parameter \| Name: **C01224 \| MCK: Ref. speeds** | | | Data type: INTEGER_32<br>Index: $23351_d$ = $5B37_h$ |
|---|---|---|---|
| Speeds for reference search/homing | | | |
| | | | Homing |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0000 | unit/s | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01224/1 | 720.0000 unit/s | | MCK: Ref. initial speed<br>• Starting speed for approaching the pre-stop signals. |
| C01224/2 | 180.0000 unit/s | | MCK: Ref. search speed<br>• Search speed for detecting the reference initiator. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

## C01225

| Parameter \| Name: **C01225 \| MCK: Ref. accelerations** | | | Data type: INTEGER_32<br>Index: $23350_d$ = $5B36_h$ |
|---|---|---|---|
| Accelerations for reference search/homing | | | |
| | | | Homing |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0000 | unit/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01225/1 | 720.0000 unit/s2 | | MCK: Ref. initial acceleration<br>• Starting acceleration for the starting speed ramps. |
| C01225/2 | 720.0000 unit/s2 | | MCK: Ref. search acceleration<br>• Search acceleration for the search speed ramps. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

## C01226

| Parameter \| Name: **C01226 \| MCK: Ref. S-ramp time** | | | Data type: UNSIGNED_16<br>Index: $23349_d$ = $5B35_h$ |
|---|---|---|---|
| S-ramp time for reference search/homing<br>• Setting "0 s" ≡ no rounding | | | |
| | | | Homing |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 10.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01226/1 | 0.000 s | | MCK: Ref. S-ramp time<br>• S-ramp time for the starting and search speed ramps. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C01227**

| Parameter \| Name: **C01227 \| MCK: Ref. positions** | | Data type: INTEGER_32 Index: $23348_d$ = $5B34_h$ |
|---|---|---|
| Positions for determining the zero position of the reference measuring system | | |
| | | ▸ Homing |
| **Setting range** (min. value \| unit \| max. value) | | |
| -214748.3647 | unit | 214748.3647 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01227/1 | 0.0000 unit | MCK: Ref. offset reference degree<br>• Relative traverse path by which the drive traverses after detection of the reference initiator. |
| C01227/2 | 0.0000 unit | MCK: Ref. home position<br>• Position with which setpoint and actual position are loaded after completion of homing. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C01228**

| Parameter \| Name: **C01228 \| MCK: Ref. sequence profile** | | | Data type: UNSIGNED_8 Index: $23347_d$ = $5B33_h$ |
|---|---|---|---|
| Number of the sequence profile the absolute position of which will be approached after homing. | | | |
| | | | ▸ Homing |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | | 15 | **0** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01229**

| Parameter \| Name: **C01229 \| MCK: Position limiting values** | | Data type: INTEGER_32 Index: $23346_d$ = $5B32_h$ |
|---|---|---|

Software limit position for limiting the valid traversing range
- The error response to leaving the valid traversing range can be parameterised in C00595/3 and C00595/4.

**Note:**
For limiting the traversing range by means of software limit positions, the home position must be known and the positive software limit position must be higher than the negative software limit position!

▸ Limit position monitoring

| **Setting range** (min. value \| unit \| max. value) | | |
|---|---|---|
| -214748.3647 | units | 214748.3647 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01229/1 | 0.0000 units | MCK: Positive SW limit position (positive travel range limit) |
| C01229/2 | 0.0000 units | MCK: Negative SW limit position (negative travel range limit) |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C01230**

| Parameter \| Name: **C01230 \| Manual jog: Setting** | | | Data type: UNSIGNED_8 Index: 23345$_d$ = 5B31$_h$ |
|---|---|---|---|
| Settings for "Manual jog" mode | | | |
| | | | ▸ Manual jog |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x00** (decimal: 0) |
| **Value is bit-coded:** (☑ = bit set) | | | **Info** |
| Bit 0 ☐ | Breakpoints 1..4 on | | "1" ≡ approach of the parameterised breakpoint in manual jog direction |
| Bit 1 ☐ | Time-based start of 2nd speed | | "1" ≡ Automatic change-over to second manual speed after the wating time set in C01235/1. |
| Bit 2 ☐ | HW limit switch on | | "1" ≡ Travel range monitoring via hardware limit switch is active. The error response can be parameterised in C00595/1 and C00595/2. |
| Bit 3 ☐ | SW limit switch on | | "1" ≡ Travel range monitoring via parameterised software limit positions is active. The error response can be parameterised in C00595/3 and C00595/4. |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Position controller off | | "1" ≡ Position controller is deactivated. Thus, the compensation of the following error is switched off. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01231**

| Parameter \| Name: **C01231 \| MCK: Manual jog speeds** | | | Data type: INTEGER_32 Index: 23344$_d$ = 5B30$_h$ |
|---|---|---|---|
| Manual jog speeds | | | |
| | | | ▸ Manual jog |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01231/1 | 360.0000 units/s | | Manual jog: speed 1<br>• Regular manual speed |
| C01231/2 | 720.0000 units/s | | Manual jog: Speed 2<br>• Second manual speed (can be activated via MCK control word) |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01232**

| Parameter \| Name: **C01232 \| MCK: Manual jog accelerations** | | | Data type: INTEGER_32 Index: 23343$_d$ = 5B2F$_h$ |
|---|---|---|---|
| Acceleration/deceleration for manual jog | | | |
| | | | ▸ Manual jog |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01232/1 | 720.0000 units/s2 | | Manual jog: Acceleration<br>• Acceleration for ramp-up to manual speed. |
| C01232/2 | 720.0000 units/s2 | | Manual jog: Deceleration<br>• Deceleration for manual speed ramp-down to standstill. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01233**

| Parameter \| Name: **C01233 \| MCK: Manual jog S-ramp time** | | Data type: UNSIGNED_16 Index: $23342_d$ = $5B2E_h$ |
|---|---|---|

S-ramp time for manual jog
- Setting "0 s" ≡ no rounding

▶ Manual jog

| **Setting range** (min. value \| unit \| max. value) | | |
|---|---|---|
| 0.000 | s | 10.000 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01233/1 | 0.000 s | Manual jog: S-ramp time<br>• S-ramp time for the manual speed ramps. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C01234**

| Parameter \| Name: **C01234 \| MCK: Manual jog breakpoints** | | Data type: INTEGER_32 Index: $23341_d$ = $5B2D_h$ |
|---|---|---|

Breakpoint positions for manual jog
- The drive stops at the parameterised positions if these are located in the manual jog direction and the approaching of the breakpoints is switched on in C01230 via bit 0.

▶ Manual jog

| **Setting range** (min. value \| unit \| max. value) | | |
|---|---|---|
| -214748.3647 | unit | 214748.3647 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01234/1 | 0.0000 unit | Manual jog: Breakpoint 1 |
| C01234/2 | 0.0000 unit | Manual jog: Breakpoint 2 |
| C01234/3 | 0.0000 unit | Manual jog: Breakpoint 3 |
| C01234/4 | 0.0000 unit | Manual jog: Breakpoint 4 |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C01235**

| Parameter \| Name: **C01235 \| MCK: Manual jog waiting times** | | Data type: UNSIGNED_16 Index: $23340_d$ = $5B2C_h$ |
|---|---|---|

Waiting times for manual jog

▶ Manual jog

| **Setting range** (min. value \| unit \| max. value) | | |
|---|---|---|
| 0.000 | s | 65.000 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01235/1 | 5.000 s | Manual jog: Waiting time 2nd speed<br>• Time after which the first manual speed (C01231/1) is accelerated to the second manual speed (C01231/2). |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C01236**

| Parameter \| Name: **C01236 \| MCK: Speed follower** | | Data type: INTEGER_32 Index: $23339_d$ = $5B2B_h$ |
|---|---|---|

Speeds for position follower

▶ Position follower

| **Setting range** (min. value \| unit \| max. value) | | |
|---|---|---|
| -214748.3647 | units/s | 214748.3647 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01236/1 | 360.0000 units/s | PosFollower: Sync. speed<br>• Synchronisation speed for approaching the setpoint position of the master. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C01237**

| Parameter \| Name: **C01237 \| MCK: Acceleration follower** | | | Data type: INTEGER_32 Index: $23338_d$ = $5B2A_h$ |
|---|---|---|---|

Accelerations for position follower

▸ Position follower

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -214748.3647 | units/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01237/1 | 720.0000 units/s2 | | Pos follower: Sync. accel.<br>• Acceleration for ramp-up to synchronisation speed. |
| C01237/2 | 720.0000 units/s2 | | Pos follower: Sync. decel.<br>• Deceleration for synchronisation speed ramp-down to standstill (to the setpoint position of the master). |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C01238**

| Parameter \| Name: **C01238 \| MCK: S-ramp time follower** | | | Data type: UNSIGNED_16 Index: $23337_d$ = $5B29_h$ |
|---|---|---|---|

S-ramp times for position follower
- Setting "0 s" ≡ no rounding

▸ Position follower

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0.000 | s | 10.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01238/1 | 0.000 s | | Pos follower: Sync. S-ramp time<br>• S-ramp time for the synchronisation speed ramps. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C01239**

| Parameter \| Name: **C01239 \| Reserved** | | | Data type: UNSIGNED_32 Index: $23336_d$ = $5B28_h$ |
|---|---|---|---|

Interpolator cycle for position follower
- Cycle time for the position interpolation of the position from the master.

▸ Position follower

| Setting range (min. value \| unit \| max. value) | | | Lenze setting |
|---|---|---|---|
| 0.000 | ms | 60.000 | **0.000 ms** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C01240**

| Parameter \| Name: **C01240 \| MCK: Control word** | | | Data type: UNSIGNED_32 Index: $23335_d$ = $5B27_h$ |
|---|---|---|---|

Bit-coded status of the MCK control word
- The MCK control word is used to control the functions of the Motion Control Kernel.

▸ MCK control word

| Display area (min. hex value \| max. hex value) | | | |
|---|---|---|---|
| 0x00000000 | | 0xFFFFFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | OpMode_Bit0 | | Active operating mode - control bit 0 |
| Bit 1 | OpMode_Bit1 | | Active operating mode - control bit 1 |
| Bit 2 | OpMode_Bit2 | | Active operating mode - control bit 2 |
| Bit 3 | OpMode_Bit3 | | Active operating mode - control bit 4 |
| Bit 4 | ManJogPos | | Control bit for manual jog in positive direction |
| Bit 5 | ManJogNeg | | Control bit for manual jog in negative direction |
| Bit 6 | ManExecute2ndSpeed | | Control bit for activation of second manual speed |

| Parameter \| Name: C01240 \| MCK: Control word | | Data type: UNSIGNED_32 Index: $23335_d = 5B27_h$ |
|---|---|---|
| Bit 7 | ReleaseLimitSwitch | Control bit for retracting the hardware limit positions |
| Bit 8 | HomStartStop | Control bit for start/stop homing |
| Bit 9 | HomSetPos | Control bit for setting of home position<br>• "0->1" ≡ sets position setpoint and actual position value to the home position at the MCK.<br>• This function is independent of the operating modes. |
| Bit 10 | HomResetPos | Control bit for reset of home position<br>• "0->1" ≡ resets the information bit for the status "Home position known" (home position not known).<br>• The position displays are not influenced by this. |
| Bit 11 | EnableSpeedOverride | Control bit for activation of speed override<br>• "1" ≡ activation of the override speed at the MCK for acceptance in a running speed profile. |
| Bit 12 | EnableAccOverride | Control bit for activation of acceleration override<br>• "1" ≡ activation of the override acceleration at the MCK for acceptance in a running speed profile. |
| Bit 13 | EnableSRampOverride | Control bit for deactivation of S-shaping for speed profiles<br>• "1" ≡ deactivates S-shaping for speed profiles |
| Bit 14 | PosTeachSetPos | Control bit for accepting the selected setpoint position in the selected profile<br>• "0->1" ≡ accepts the setpoint position at the MCK in the profile with the profile number defined via the MCK control word. |
| Bit 15 | PosTeachActPos | Control bit for accepting the current actual position in the selected profile<br>• "0->1" ≡ accepts the current actual position in the profile with the profile number defined via the MCK control word. |
| Bit 16 | PosExecute | Control bit for starting a positioning profile<br>• "0->1" ≡ starts a positioning profile selected via the control word in the "Positioning" mode. |
| Bit 17 | PosFinishTarget | Control bit for completing processing of an interrupted positioning profile<br>• "0->1" ≡ processing of a positioning profile which has already been started and has been interrupted because of a cancellation condition or a change of the operating mode is completed in the "Positioning" mode when the home position is known.<br>• **Note:** Completion of profile processing through this control is based on the target position detected at the profile start! |
| Bit 18 | PosDisableFollowProfile | Control bit for suppression of the sequence profile<br>• "1" ≡ positioning profiles whose profile numbers are entered in profile data sets for sequence profiles are not started after completion of the corresponding profile. |
| Bit 19 | PosStop | Control bit for stopping the positioning profile<br>• "0->1" ≡ ("Positioning" mode): Interrupts a running profile through deceleration along the set deceleration ramp to standstill.<br>• "0->1" ≡ ("Stop" mode): Starts a new deceleration to standstill with the parameterised stop ramp. |
| Bit 20 | PosModeBit0 | Active positioning mode - control bit 0 |
| Bit 21 | PosModeBit1 | Active positioning mode - control bit 1 |
| Bit 22 | PosModeBit2 | Active positioning mode - control bit 2 |
| Bit 23 | PosModeBit3 | Active positioning mode - control bit 3 |

| Parameter \| Name: C01240 \| MCK: Control word | | Data type: UNSIGNED_32 Index: $23335_d$ = $5B27_h$ |
|---|---|---|
| Bit 24 | ProfileNo_Bit0 | Active positioning profile number - control bit 0 |
| Bit 25 | ProfileNo_Bit1 | Active positioning profile number - control bit 1 |
| Bit 26 | ProfileNo_Bit2 | Active positioning profile number - control bit 2 |
| Bit 27 | ProfileNo_Bit3 | Active positioning profile number - control bit 3 |
| Bit 28 | ProfileNo_Bit4 | Active positioning profile number - control bit 4 |
| Bit 29 | ProfileNo_Bit5 | Active positioning profile number - control bit 5 |
| Bit 30 | ProfileNo_Bit6 | Active positioning profile number - control bit 6 |
| Bit 31 | ProfileNo_Bit7 | Active positioning profile number - control bit 7 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

### C01241

Parameter | Name:
**C01241 | MCK: Status word**

Data type: UNSIGNED_32
Index: $23334_d$ = $5B26_h$

Bit-coded status of the MCK status word
- The MCK status word is used to output status messages of the MotionControlKernel.

▶ MCK status word

| **Display area** (min. hex value \| max. hex value) | | |
|---|---|---|
| 0x00000000 | | 0xFFFFFFFF |

| **Value is bit-coded:** | | **Info** |
|---|---|---|
| Bit 0 | ActOpModeBit00 | Active operating mode - status bit 0 |
| Bit 1 | ActOpModeBit01 | Active operating mode - status bit 1 |
| Bit 2 | ActOpModeBit02 | Active operating mode - status bit 2 |
| Bit 3 | ActOpModeBit03 | Active operating mode - status bit 4 |
| Bit 4 | Busy | Status bit - profile generation active<br>• "1" ≡ Internal profile generation is active. A speed profile is being generated. |
| Bit 5 | Done | Status bit - profile generation completed<br>• "1" ≡ Generation of a speed profile with the selected position has been completed. |
| Bit 6 | AcceleratingActive | Status bit - acceleration process for profile generation active<br>• "1" ≡ Profile generation phase is in the acceleration process. |
| Bit 7 | ConstSpeedDuty | Status bit - constant speed for profile generation active<br>• "1" ≡ Profile generation phase at constant speed active. |
| Bit 8 | DeceleratingActive | Status bit - deceleration process for profile generation active<br>• "1" ≡ Profile generation phase is in the deceleration process. |
| Bit 9 | S_ShapingActive | Status bit - rounding for profile generation active<br>• "1" ≡ Rounding during acceleration/deceleration active. |
| Bit 10 | Pos. HW-Limit Detected | Status bit - positive hardware limit detected<br>• "1" ≡ Positive limit switch has triggered.<br>• Reset only possible via"Manual jog" mode! |
| Bit 11 | Neg. HW-Limit Detected | Status bit - negative hardware limit detected<br>• "1" ≡ Negative limit switch has triggered.<br>• Reset only possible via"Manual jog" mode! |
| Bit 12 | HomPosDone | Status bit - homing completed<br>• "1" ≡ Homing has been completed. |

| Parameter \| Name: C01241 \| MCK: Status word | | Data type: UNSIGNED_32 Index: $23334_d$ = $5B26_h$ |
|---|---|---|
| Bit 13 | HomPosAvailable | Status bit - home position known<br>• "1" ≡ The home position has been detected and is known in the drive. |
| Bit 14 | Pos. SW-Limit Detected | Status bit - positive software limit detected<br>• "1" ≡ Positive software limit position overtravelled. |
| Bit 15 | Neg. SW-Limit Detected | Status bit - negative software limit detected<br>• "1" ≡ Negative software limit position overtravelled. |
| Bit 16 | DwellTime | Status bit - transient effects in target position active<br>• "1" ≡ Dwell time after reaching the setpoint position is active. |
| Bit 17 | InTarget | Status bit - actual position is in the target window<br>• "1" ≡ Dwell time has expired and current actual position is in the set target window. |
| Bit 18 | PosDone | Status bit - positioning process completed<br>• "1" ≡ Positioning profile has been completed in the "Positioning" mode. Setpoint position is in target. |
| Bit 19 | Reserved | |
| Bit 20 | ActPosMode_Bit00 | Active positioning mode - status bit 0 |
| Bit 21 | ActPosMode_Bit01 | Active positioning mode - status bit 1 |
| Bit 22 | ActPosMode_Bit02 | Active positioning mode - status bit 2 |
| Bit 23 | ActPosMode_Bit03 | Active positioning mode - status bit 3 |
| Bit 24 | ActProfileNo_Bit00 | Active positioning profile - status bit 0 |
| Bit 25 | ActProfileNo_Bit01 | Active positioning profile - status bit 1 |
| Bit 26 | ActProfileNo_Bit02 | Active positioning profile - status bit 2 |
| Bit 27 | ActProfileNo_Bit03 | Active positioning profile - status bit 3 |
| Bit 28 | ActProfileNo_Bit04 | Active positioning profile - status bit 4 |
| Bit 29 | ActProfileNo_Bit05 | Active positioning profile - status bit 5 |
| Bit 30 | ActProfileNo_Bit06 | Active positioning profile - status bit 6 |
| Bit 31 | ActProfileNo_Bit07 | Active positioning profile - status bit 7 |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT

### C01242

| Parameter \| Name: C01242 \| MCK: Current pos profile number | | | Data type: UNSIGNED_8 Index: $23333_d$ = $5B25_h$ |
|---|---|---|---|
| Display of the current profile number of the active profile in the "Positioning" mode | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 255 | |

▸ Positioning

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01243**

| Parameter \| Name: **C01243 \| MCK: Current operating mode** | | Data type: UNSIGNED_8 Index: $23332_d$ = $5B24_h$ |
|---|---|---|
| Display of the active operating mode of the Motion Control Kernel | | |
| | | ▶ Basic drive functions |
| **Selection list** (read only) | | **Info** |
| 0 | Speed follower | Drive traverses according to a preselected speed setpoint |
| 1 | Homing | Drive finds its reference measuring system by setting the home position or homing |
| 2 | ManualJog | Drive can be traversed manually via initiators, e.g. for cleaning purposes or tool changes. |
| 3 | Positioning | Drive traverses according to defined travel profiles and motion processes |
| 4 | Stop | Drive is decelerated to standstill along a parameterised deceleration ramp |
| 5 | Position follower | Drive traverses according to a preselected position setpoint |
| 15 | StandBy | Internal operating mode for quick stop and pulse inhibit |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01244**

| Parameter \| Name: **C01244 \| MCK: Target detection - times** | | | Data type: UNSIGNED_16 Index: $23331_d$ = $5B23_h$ |
|---|---|---|---|
| Timing for various MCK functions | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | ms | 60000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01244/1 | 100 ms | | MCK: Dwell time - target position<br>• Time expiring after reaching the setpoint position for positioning processes and enabling the actual position detection in the target position window.<br>▶ Target position monitoring |
| C01244/2 | 0 ms | | MCK: Waiting time following error 1<br>From version 02.00.00<br>▶ Following error monitoring system |
| C01244/3 | 0 ms | | MCK: Waiting time following error 2<br>From version 02.00.00<br>▶ Following error monitoring system |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01245**

| Parameter \| Name: **C01245 \| MCK: Target detection - positions** | | | Data type: INTEGER_32 Index: $23330_d$ = $5B22_h$ |
|---|---|---|---|
| Position settings for various MCK functions | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.0000 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01245/1 | 1.0000 units | | MCK: Window target position<br>• Window around the target position for comparison with the actual position to see whether the drive is in target.<br>▶ Target position monitoring |
| C01245/2 | 1.0000 units | | MCK: Blocking zone Modulo<br>▶ Activation of the Modulo measuring system |
| C01245/3 | 0.0000 units | | MCKI: Hysteresis position change |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

**C01246**

| Parameter \| Name: **C01246 \| MCK: Select signal source** | | Data type: UNSIGNED_8 Index: $23329_d$ = $5B21_h$ |
|---|---|---|
| Selection of the touch probe signal source for reference search with touch probe detection<br>• If the reference signal is to follow a reach touch probe, the Touch-Probe-Interface must be configured accordingly.<br>▶ Homing | | |
| **Selection list** | | |
| 0 | No TP | |
| 3 | TP-DigIn3 | |
| 4 | TP-DigIn4 | |
| 5 | TP-DigIn5 | |
| 6 | TP-DigIn6 | |
| 7 | TP-DigIn7 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01246/1 | 0: No TP | MCK: Ref. TP signal source |
| C01246/2 | 0: No TP | MCK: Set.Ref. signal source |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01251**

| Parameter \| Name: **C01251 \| MCK: Acceleration stop** | | | Data type: INTEGER_32 Index: $23324_d$ = $5B1C_h$ |
|---|---|---|---|
| Acceleration parameter for stop<br>▶ Stop | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | units/s2 | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01251/1 | 720.0000 units/s2 | | MCK: Stop: Decel.<br>• Deceleration for setpoint speed ramp-down to standstill. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

### C01252

| Parameter \| Name: **C01252 \| MCK: S-ramp times stop** | | | Data type: UNSIGNED_16 Index: $23323_d$ = $5B1B_h$ |
|---|---|---|---|
| S-ramp times for stop<br>• Setting "0 s" ≡ no rounding | | | ▸ Stop |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 10.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01252/1 | 0.000 s | | MCK: Stop: S-ramp time<br>• S-ramp time for setpoint speed ramp-down to standstill. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

### C01295

| Parameter \| Name: **C01295 \| L_MCKStateInterface_1: Pos. selection** | | Data type: UNSIGNED_8 Index: $23280_d$ = $5AF0_h$ |
|---|---|---|
| Selection of the position to be provided at the *dnPosOut_p* output of the FB L_MckStateInterface | | |
| **Selection list** (Lenze setting printed in bold) | | |
| 0 | dnPosIn_p | |
| 1 | Current feed | |
| 2 | dnSetPos_p | |
| **3** | **dnActPos_p** | |
| 4 | dnDeltaPos_p | |
| 5 | dnTargetPos_p | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

### C01296

| Parameter \| Name: **C01296 \| Mode: Position calculation** | | Data type: UNSIGNED_8 Index: $23279_d$ = $5AEF_h$ |
|---|---|---|
| | | ▸ MCKInterface |
| **Selection list** | | |
| 0 | dnPosOut_p=dnPosIn_p | |
| 1 | 16 bits: LW=+/-32767 | |
| 2 | 16 bits: HW=+/-; LW=0..65535 | |
| 3 | 32 bits: HW_LW=+/-214748_3647 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01296/1 | 0: dnPosOut_p=dnPosIn_p | L_MckCtrlInterface_1: PosCalcMode |
| C01296/2 | 1: 16Bit: LW=+/-32767 | L_MckCtrlInterface_1: PosDisplayMode |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01297**

| Parameter \| Name:<br>**C01297 \| L_MckStateInterface_1: Alternative function** | | | Data type: UNSIGNED_8<br>Index: $23278_d$ = $5AEE_h$ |
|---|---|---|---|
| Selection of the alternative function for bit 16 (PosExecute) in the MCK control word<br>Alternative functions for "PosExecute" control bit | | | |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x03** (decimal: 3) |
| **Value is bit-coded:** (☑ = bit set) | | | **Info** |
| Bit 0 ☑ | PosStop with PosExecute = FALSE | | "1" ≡ Stop positioning with *PosExecute*<br>• PosExecute="0" stops a running positioning process by ramp-down to standstill. |
| Bit 1 ☑ | HomingStartStop with PosExecute | | "1" ≡ Start/stop homing with *PosExecute*<br>• PosExecute="1" starts homing, PosExecute="0" stops homing in the "Homing" mode. |
| Bit 2 ☐ | SetProfilPosition with PosExecute | | "1" ≡ Teach set position with *PosExecute*<br>• PosExecute="0->1" accepts the setpoint position at the MCK in the profile with the specified profile number. |
| Bit 3 ☐ | SetProfilPosition at position change | | "1" ≡ Automatic acceptance of set position<br>• The set position pending at the MCKInterface is automatically transferred to the profile with the preset profile number if a data change at the corresponding input for the setpoijnt is detected. |
| Bit 4 ☐ | Profile start at position change | | From version 02.00.00<br>"1" ≡ Automatic profile start at position change<br>• An automatic "PosExecute" takes place if the incremental position selection changes and this change is higher than set in the hysteresis for position change (C01245/3). |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | | |

**C01298**

| Parameter \| Name: **C01298 \| MCK: Operating mode change with profile no.** | | | Data type: UNSIGNED_8 Index: $23277_d$ = $5AED_h$ |
|---|---|---|---|
| FB L_MckCtrlInterface_1: Assignment of an operating mode to a profile | | | |
| **Selection list** | | | **Info** |
| 0 | No change | | There will be no changed operating mode when the profile is changed. Instead, the operating mode requested "from outside" via the MCK control word applies. |
| 1 | Follower | | Change to the "Speed follower" mode |
| 2 | Homing | | Change to the "Homing" mode |
| 3 | ManualJog | | Change to the "Manual jog" mode |
| 4 | Positioning | | Change to the "Positioning" mode |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01298/1 | 1: Follower | | MCK operating mode at profile no. 0<br>• Selection of the operating mode for profile 0 at L_MckCtrlInterface_1. |
| C01298/2 | 2: Homing | | MCK operating mode at profile no. 1<br>• Selection of the operating mode for profile 1 at L_MckCtrlInterface_1. |
| C01298/3 | 3: ManualJog | | MCK operating mode at profile no. 2<br>• Selection of the operating mode for profile 2 at L_MckCtrlInterface_1. |
| C01298/4 | 4: Positioning | | MCK operating mode at profile no. 3...15<br>• Selection of the operating mode for profile 3 ... 15 at L_MckCtrlInterface_1. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01299**

| Parameter \| Name: **C01299 \| MCKI: Status MCKInterface** | | | Data type: UNSIGNED_8 Index: $23276_d$ = $5AEC_h$ |
|---|---|---|---|
| FB L_MckCtrlInterface_1: Status messages | | | |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x00 | | 0xFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | InvalidOperationMode | | Invalid operating mode selection<br>• "1" ≡ The selected operating mode is not defined/invalid. |
| Bit 1 | InvalidPosMode | | Invalid positioning mode selection<br>• "1" ≡ The selected positioning mode is not defined/invalid. |
| Bit 2 | InvalidProfileNo | | Invalid profile number selection<br>• "1" ≡ The selected profile number refers to a profile data set that does not exist. |
| Bit 3 | Reserved | | |
| Bit 4 | Reserved | | |
| Bit 5 | Reserved | | |
| Bit 6 | Reserved | | |
| Bit 7 | Reserved | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C01300**

| Parameter \| Name:<br>**C01300 \| Profile data: Positioning mode** | | Data type: UNSIGNED_8<br>Index: $23275_d$ = $5AEB_h$ |
|---|---|---|
| Selection of the positioning mode for profiles 1 ... 15 | | |
| | | ▸ Profile entry |
| **Selection list** | | **Info** |
| 1 | Absolute (shortest path) | Absolute positioning<br>• The profile position corresponds to the target position. |
| 2 | Continuous | Continuous positioning without approaching a target position |
| 3 | Relative | Relative positioning<br>• The profile position determines the distance to be traversed. |
| 4 | absolute (Cw) | Clockwise absolute positioning<br>• The zero position of the axis can be exceed in this direction.<br>• The profile position corresponds to the target position. |
| 5 | absolute (Ccw) | Counter-clockwise absolute positioning<br>• The zero position of the axis can be exceed in this direction.<br>• The profile position corresponds to the target position. |
| 8 | Absolute (shortest path) to TP | Absolute positioning<br>• The profile position corresponds to the target position.<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |
| 9 | Continuous to TP | Continuous positioning without approaching a target position<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |
| 10 | Relative to TP | Relative positioning<br>• The profile position determines the distance to be traversed.<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |
| 11 | Absolute (Cw) on TP | Clockwise absolute positioning<br>• The zero position of the axis can be exceed in this direction.<br>• The profile position corresponds to the target position.<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |
| 12 | Absolute (Ccw) on TP | Counter-clockwise absolute positioning<br>• The zero position of the axis can be exceed in this direction.<br>• The profile position corresponds to the target position.<br>• After a touch probe is detected, the TP profile (C01308/1...15) is executed. |
| **Subcodes** | **Lenze setting** | **Info** |
| C01300/1 | 1: absolute (beeline) | Profiles 1 ... 15: Positioning mode |
| C01300/... | | |
| C01300/15 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01301**

| Parameter \| Name:<br>**C01301 \| Profile data: Position** | | Data type: INTEGER_32<br>Index: $23274_d$ = $5AEA_h$ |
|---|---|---|
| Selection of the positions for profiles 1 ... 15 | | |
| | | ▸ Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | |
| -214748.3647 | unit | 214748.3647 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01301/1 | 360.0000 unit | Profiles 1 ... 15: Position |
| C01301/... | | |
| C01301/15 | | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C01302**

| Parameter \| Name:<br>**C01302 \| Profile data: Speed** | | Data type: INTEGER_32<br>Index: $23273_d$ = $5AE9_h$ |
|---|---|---|
| Selection of the maximum speeds for profiles 1 ... 15 | | |
| | | ▸ Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | |
| -214748.3647 | unit/s | 214748.3647 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01302/1 | 360.0000 unit/s | Profiles 1 ... 15: Speed |
| C01302/... | | |
| C01302/15 | | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C01303**

| Parameter \| Name:<br>**C01303 \| Profile data: Acceleration** | | Data type: INTEGER_32<br>Index: $23272_d$ = $5AE8_h$ |
|---|---|---|
| Selection of the accelerations for profiles 1 ... 15 | | |
| | | ▸ Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | |
| -214748.3647 | unit/s2 | 214748.3647 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01303/1 | 720.0000 unit/s2 | Profiles 1 ... 15: Acceleration |
| C01303/... | | |
| C01303/15 | | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C01304**

| Parameter \| Name:<br>**C01304 \| Profile data: Deceleration** | | Data type: INTEGER_32<br>Index: $23271_d$ = $5AE7_h$ |
|---|---|---|
| Selection of the decelerations for profiles 1 ... 15 | | |
| | | ▸ Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | |
| -214748.3647 | unit/s2 | 214748.3647 |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C01304/1 | 720.0000 unit/s2 | Profiles 1 ... 15: Deceleration |
| C01304/... | | |
| C01304/15 | | |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

## C01305

| Parameter \| Name: **C01305 \| Profile data: Final speed** | | | Data type: INTEGER_32 Index: $23270_d$ = $5AE6_h$ |
|---|---|---|---|
| Selection of the final speeds for profiles 1 ... 15<br>• For profile linkage with overchange. | | | |
| | | | Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -214748.3647 | unit/s | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01305/1 | 0.0000 unit/s | | Profiles 1 ... 15: Final speed |
| C01305/... | | | |
| C01305/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000 | | | |

## C01306

| Parameter \| Name: **C01306 \| Profile data: S-ramp time** | | | Data type: UNSIGNED_16 Index: $23269_d$ = $5AE5_h$ |
|---|---|---|---|
| Selection of the S-ramp times for profiles 1 ... 15<br>• The maximum S-ramp time is the time in which the maximum acceleration or the maximum deceleration is reached along a ramp (jerk limitation). | | | |
| | | | Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 10.000 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01306/1 | 0.000 s | | Profiles 1 ... 15: S-ramp time |
| C01306/... | | | |
| C01306/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

## C01307

| Parameter \| Name: **C01307 \| Profile data: Sequence profile** | | | Data type: UNSIGNED_8 Index: $23268_d$ = $5AE4_h$ |
|---|---|---|---|
| Selection of the sequence profile numbers for profiles 1 ... 15<br>• For profile linkage | | | |
| | | | Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | | 15 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01307/1 | 0 | | Profiles 1 ... 15: Sequence profile |
| C01307/... | | | |
| C01307/15 | | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01308**

| Parameter \| Name: **C01308 \| Profile data: TP profile** | | Data type: UNSIGNED_8 Index: $23267_d$ = $5AE3_h$ |
|---|---|---|
| Selection of the touch-probe profile numbers for profile 1 ... 15<br>• Only relevant for positioning modes with touch-probe. | | |
| | | ▸ Profile entry |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | | 15 |
| **Subcodes** | **Lenze setting** | **Info** |
| C01308/1 | 0 | Profile 1 ... 15: TP profile |
| C01308/... | | |
| C01308/15 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01309**

| Parameter \| Name: **C01309 \| Profile data: TP signal source** | | Data type: UNSIGNED_8 Index: $23266_d$ = $5AE2_h$ |
|---|---|---|
| Selection of the touch-probe signal sources for profile 1 ... 15<br>• Only relevant for positioning modes with touch-probe. | | |
| | | ▸ Profile entry |
| **Selection list** | | |
| 3 | TP-DigIn3 | |
| 4 | TP-DigIn4 | |
| 5 | TP-DigIn5 | |
| 6 | TP-DigIn6 | |
| 7 | TP-DigIn7 | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01309/1 | 3: TP-DigIn3 | Profiles 1 ... 15: Sequence profile |
| C01309/... | | |
| C01309/15 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01501**

| Parameter \| Name: **C01501 \| Resp. to communication error with MCI** | | Data type: UNSIGNED_8 Index: $23074_d$ = $5A22_h$ |
|---|---|---|
| Configuration of monitoring functions for the communication module | | |
| **Selection list** | | |
| 0 | No Reaction | |
| 1 | Fault | |
| 2 | Trouble | |
| 3 | TroubleQuickStop | |
| 4 | WarningLocked | |
| 5 | Warning | |
| 6 | Information | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01501/1 | 0: No Reaction | Resp. to MCI connection error<br>• Response to a communication error of the attached communication module. |
| C01501/2 | 0: No Reaction | Resp. to MCI invalid module<br>• Response to an unplugged or incompatible communication module |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01670**

| Parameter \| Name: **C01670 \| L_ComparePhi 1-5: Function** | | Data type: UNSIGNED_8 Index: $22905_d$ = $5979_h$ |
|---|---|---|
| Selection of comparison operation<br>• If the statement of the selected comparison operation is true, the binary *bOut* output will be set to TRUE. | | |
| **Selection list** | | |
| 1 | In1 = In2 | |
| 2 | In1 > In2 | |
| 3 | In1 < In2 | |
| 4 | \|In1\| = \|In2\| | |
| 5 | \|In1\| > \|In2\| | |
| 6 | \|In1\| < \|In2\| | |
| **Subcodes** | **Lenze setting** | **Info** |
| C01670/1 | 1: In1 = In2 | L_ComparePhi_1: Function |
| C01670/2 | 1: In1 = In2 | L_ComparePhi_2: Function |
| C01670/3 | 1: In1 = In2 | L_ComparePhi_3: Function |
| C01670/4 | 1: In1 = In2 | L_ComparePhi_4: Function |
| C01670/5 | 1: In1 = In2 | L_ComparePhi_5: Function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C01671**

| Parameter \| Name: **C01671 \| L_ComparePhi 1-5: Hysteresis** | | | Data type: INTEGER_32 Index: $22904_d$ = $5978_h$ |
|---|---|---|---|
| Hysteresis for the comparison function | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 1073741824 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01671/1 | 0 incr. | | L_ComparePhi_1: Hysteresis |
| C01671/2 | 0 incr. | | L_ComparePhi_2: Hysteresis |
| C01671/3 | 0 incr. | | L_ComparePhi_3: Hysteresis |
| C01671/4 | 0 incr. | | L_ComparePhi_4: Hysteresis |
| C01671/5 | 0 incr. | | L_ComparePhi_5: Hysteresis |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01672**

| Parameter \| Name: **C01672 \| L_ComparePhi 1-5: Window** | | | Data type: INTEGER_32 Index: $22903_d$ = $5977_h$ |
|---|---|---|---|
| Window for the comparison operation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 1073741824 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C01672/1 | 0 incr. | | L_ComparePhi_1: Window |
| C01672/2 | 0 incr. | | L_ComparePhi_2: Window |
| C01672/3 | 0 incr. | | L_ComparePhi_3: Window |
| C01672/4 | 0 incr. | | L_ComparePhi_4: Window |
| C01672/5 | 0 incr. | | L_ComparePhi_5: Window |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C01751**

| Parameter \| Name: **C01751 \| Service code inverter characteristic** | Data type: UNSIGNED_8 Index: $22824_d$ = $5928_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01752**

| Parameter \| Name: **C01752 \| Service par. inverter charact. function** | Data type: UNSIGNED_8 Index: $22823_d$ = $5927_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01755**

| Parameter \| Name: **C01755 \| Service par. inverter charact. factor** | Data type: INTEGER_16 Index: $22820_d$ = $5924_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01763**

| Parameter \| Name: **C01763 \| Service code -clamp threshold** | Data type: INTEGER_16 Index: $22812_d$ = $591C_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01764**

| Parameter \| Name: **C01764 \| Service par. clamp time** | Data type: UNSIGNED_8 Index: $22811_d$ = $591B_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01765**

| Parameter \| Name: **C01765 \| Service code - difference threshold UG** | Data type: UNSIGNED_16 Index: $22810_d$ = $591A_h$ |
|---|---|
| **This code is for device-internal use only and must not be written to by the user!** | |

**C01770**

Parameter | Name: **C01770 | Filter time - earth-fault detect. is running** — Data type: UNSIGNED_8, Index: $22805_d$ = $5915_h$

| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
|---|---|---|---|
| 0 | ms | 250 | **2 ms** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C01902**

Parameter | Name: **C01902 | Diagnostics X6: Max. baud rate** — Data type: UNSIGNED_16, Index: $22673_d$ = $5891_h$

Maximally permissible baud rate in the standard device after determination of the baud rate at the diagnostic interface X6

| **Selection list** (Lenze setting printed in bold) | |
|---|---|
| 192 | 19.200 Bd |
| 384 | 38.400 Bd |
| 576 | 57.600 Bd |
| 750 | 75.000 Bd |
| 1152 | 115.200 Bd |
| 1500 | 150.000 Bd |
| 2500 | 250.000 Bd |
| 3750 | 375.000 Bd |
| **7500** | **750.000 Bd** |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C01903**

| Parameter \| Name:<br>**C01903 \| Diagnostics X6: Change baud rate** | Data type: UNSIGNED_8<br>Index: $22672_d$ = $5890_h$ |
|---|---|
| New baud rate determination at the diagnostic interface X6 | |
| **Selection list** (Lenze setting printed in bold) | |
| **0** **Ignore changes** | |
| 1 Negotiate baud rate | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | |

**C01905**

| Parameter \| Name:<br>**C01905 \| Diagnostics X6: Current baud rate** | | | Data type: UNSIGNED_32<br>Index: $22670_d$ = $588E_h$ |
|---|---|---|---|
| Current baud rate at the diagnostic interface X6 | | | |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | Bd | 3000000 | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | | |

**C02430**

| Parameter \| Name:<br>**C02430 \| Axis bus address and no. of nodes** | | Data type: UNSIGNED_8<br>Index: $22145_d$ = $5681_h$ |
|---|---|---|
| From version 02.00.00 | | ▸ Axis bus |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | | 62 |
| **Subcodes** | **Lenze setting** | **Info** |
| C02430/1 | 1 | Axis bus address<br>• 1 ≡ Master<br>• 2 ... 62 ≡ slave 1 ... slave 61 |
| C02430/2 | 2 | Axis bus no. of nodes<br>• Number of slaves connected to the axis bus.<br>• Setting only required for the master. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C02431**

| Parameter \| Name:<br>**C02431 \| Axis bus time settings** | | Data type: UNSIGNED_16<br>Index: $22144_d$ = $5680_h$ |
|---|---|---|
| From version 02.00.00 | | ▸ Axis bus |
| **Setting range** (min. value \| unit \| max. value) | | |
| 0 | ms | 65000 |
| **Subcodes** | **Lenze setting** | **Info** |
| C02431/1 | 3000 ms | Axis bus decel. boot-up operat.<br>• Delay during status change from "Boot-up" to "Operational". |
| C02431/2 | 1000 ms | Axis bus decel. operat.- 1.transmit<br>• "Operational" time until "first transmission". |
| C02431/3 | 1000 ms | Axis bus decel. BusOff recovery |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

## C02435

| Parameter \| Name: **C02435 \| Axis bus status** | Data type: UNSIGNED_8 Index: $22140_d$ = $567C_h$ |
|---|---|

From version 02.00.00

▶ Axis bus

| **Selection list** (read only) | |
|---|---|
| 0 | PDO active |
| 1 | SDO active |
| 2 | Reserved |
| 3 | Reserved |
| 4 | BootUp |
| 5 | Stopped |
| 6 | Reserved |
| 7 | Reset |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1

## C02436

| Parameter \| Name: **C02436 \| Axis bus error status** | Data type: UNSIGNED_8 Index: $22139_d$ = $567B_h$ |
|---|---|

From version 02.00.00

▶ Axis bus

| **Selection list** (read only) | |
|---|---|
| 0 | No Error |
| 1 | Warning ErrActive |
| 2 | Warning ErrPassive |
| 3 | Bus off |
| 4 | Reserved |
| 5 | Reserved |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1

## C02437

| Parameter \| Name: **C02437 \| Axis bus MessageError** | Data type: UNSIGNED_8 Index: $22138_d$ = $567A_h$ |
|---|---|

From version 02.00.00

▶ Axis bus

| **Selection list** (read only) | |
|---|---|
| 0 | No Error |
| 1 | StuffError |
| 2 | FormError |
| 3 | AckError |
| 4 | Bit1Error |
| 5 | Bit0Error |
| 6 | CRCError |
| 7 | Reserved |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☑ COM ☐ MOT Scaling factor: 1

**C02440**

| Parameter \| Name: **C02440 \| AxisBusIO slave/master** | | Data type: UNSIGNED_8 Index: $22135_d$ = $5677_h$ |
|---|---|---|
| From version 02.00.00 | | |
| | | ▶ Axis bus |
| **Selection list** | | |
| 0 | Off | |
| 1 | Master | |
| 2 | Slave | |
| 3 | IO | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02440/1 | 0: Off | Axis bus IO function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☑ COM ☐ MOT Scaling factor: 1 | | |

**C02580**

| Parameter \| Name: **C02580 \| Holding brake: Operating mode** | | Data type: UNSIGNED_8 Index: $21995_d$ = $55EB_h$ |
|---|---|---|
| Selection of the operating mode for holding brake control | | |
| | | ▶ Holding brake control |
| **Selection list** (Lenze setting printed in bold) | | **Info** |
| **0** | **Brake control off** | No holding brake is used. Internal control is switched off. |
| 11 | Manually controlled | The holding brake is released and closed via a control bit in the MCK control word. |
| 12 | Autom. controlled | The holding brake is automatically released and closed via speed setpoint comparisons. |
| 13 | Semi-automat. controlled | The holding brake is released and closed via a control bit in the MCK control word.<br>• In contrast to the manual operation (mode 11)<br>–the feedforward control is active in this mode, preventing a sagging e.g. in case of a hoist.<br>–the brake in this mode also closes when the controller is inhibited in order to prevent the axis in a hoist from falling. |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02581**

| Parameter \| Name: **C02581 \| Holding brake: Speed thresholds** | | | Data type: INTEGER_16 Index: $21994_d$ = $55EA_h$ |
|---|---|---|---|
| Speed setpoint threshold and hysteresis for automatic holding brake control | | | |
| | | | ▸ Holding brake control |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02581/1 | 5.00 % | | Holding brake: Switching threshold<br>• Switching threshold of the speed setpoint from which on the holding brake is released/applied automatically. |
| C02581/2 | 1.00 % | | Holding brake: Hyst.release<br>• Hysteresis for holding brake release.<br>• Release threshold = switching threshold + release hysteresis |
| C02581/3 | 1.00 % | | Holding brake: Hyst. close<br>• Hysteresis for holding brake application.<br>• Application threshold = switching threshold - application hysteresis |
| C02581/4 | 0.00 % | | Holding brake: FF control starting value 1 |
| C02581/5 | 0.00 % | | Holding brake: FF control starting value 2 |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02582**

| Parameter \| Name: **C02582 \| Holding brake: Setting** | | | Data type: UNSIGNED_8 Index: $21993_d$ = $55E9_h$ |
|---|---|---|---|
| Activation of functional holding brake control options | | | |
| | | | ▸ Holding brake control |
| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
| 0x00 | | 0xFF | **0x40** (decimal: 64) |
| **Value is bit-coded:** (☑ = bit set) | | | **Info** |
| Bit 0 ☐ | bBrkReleaseOut invert. | | Activation of inverted control<br>• "1" ≡ Inverted logic of the control signal for the holding brake control switching element. |
| Bit 1 ☐ | Horizontal brake protection | | Brake response in case of pulse inhibit<br>• "1" ≡ In the case of a pulse inhibit, the actual speed value is monitored which must reach the "Close" threshold value to cause the holding brake to be applied.<br>**Note:**<br>• This function is only active if bit 3 (horizontal/winding technology) is set as well. The function is used in order that, when the controller is inhibited, the holding brake of a drive with horizontal traverse path does not wear out during rotation.<br>• With vertical motion (bit 3 = 0), this function is not active. Especially with hoists and activated pulse inhibit of the controller, an immediate application of the brake is essential for safety-related reasons! |
| Bit 2 ☐ | with hoist inv. feedfwd. control | | Direction of feedforward control with vertical/hoist technology:<br>• "0" ≡ Positive direction<br>• "1" ≡ Negative direction<br>**Note:**<br>Reversal (Ccw) is then considered. |

| Parameter \| Name: **C02582 \| Holding brake: Setting** | | Data type: UNSIGNED_8 Index: $21993_d$ = $55E9_h$ |
|---|---|---|
| Bit 3 ☐ | Horizontal application | Direction of movement of the axis<br>• "0" ≡ The axis performs vertical movements. Gravitational acceleration causes movements.<br>• "1" ≡ The direction of the axis is horizontal or rotary. The gravitational acceleration does not cause any movement. |
| Bit 4 ☐ | Feedforward control C2581 | Selection of the feedforward control value<br>• "0" ≡ Automatic selection.<br>– The torque saved at the last stop is used.<br>• "1" ≡ Manual selection.<br>– *bBrkStartValue* = FALSE: The feedforward control value 1 set in C02581/4 is used.<br>– *bBrkStartValue* = TRUE: The feedforward control value 2 set in C02581/5 is used. |
| Bit 5 ☐ | Feedback monitoring | Activation of status monitoring<br>• "1" ≡ The *bBrkApplied* input for status detection of the brake (via a switching contact at the brake) is monitored after the waiting time set in C02589/3 has expired.<br>Activation of status monitoring<br>• "1" ≡ The *bBrkApplied* input for status detection of the brake (via a switching contact at the brake) is monitored after the waiting time set in C02589/3 has expired. |
| Bit 6 ☑ | SyncRampe L_NSet_1 | From version 02.00.00<br>Selection of the ramp time for the synchronisation process to setpoint speed after the brake opening time has elapsed<br>**Revised behaviour from version 02.00.00:**<br>• "1" ≡ The ramp time of the effective acceleration of the ramp function generator (L_NSet_1) is used (Lenze setting).<br>• "0" ≡ As before, the ramp time set in C02610/1 is used.<br>**Note:**<br>The changeover can be dynamically both via the ramp parameter and via bit 6. |
| Bit 7 ☐ | Reserved | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT | | |

**C02589**

| Parameter \| Name:<br>**C02589 \| Holding brake: Time system** | | Data type: UNSIGNED_16<br>Index: $21986_d$ = $55E2_h$ |
|---|---|---|

Operating times of the holding brake
- The electromechanical delay times of the holding brake are specified in the data sheets or on the holding brake nameplate.

▶ Holding brake control

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | ms | 60000 | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C02589/1 | 100 ms | Holding brake: Application time<br>• Time in which the holding brake is completely applied from the beginning of control and in which the controller is inhibited. |
| C02589/2 | 100 ms | Holding brake: Release time<br>• Time in which the holding brake is completely released from the beginning of control. |
| C02589/3 | 100 ms | Holding brake: Waiting time status<br>• Time after which all transient reactions are completed and the switching status of the holding brake is stable. Beginning of monitoring the feedback signal for the switching status of the holding brake. |
| C02589/4 | 0 ms | Holding brake: Ramp time FF ctrl |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C02593**

| Parameter \| Name:<br>**C02593 \| Holding brake: Activation time** | | Data type: UNSIGNED_32<br>Index: $21982_d$ = $55DE_h$ |
|---|---|---|

Time parameter for the delay of trigger signals of the holding brake control

▶ Holding brake control

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0.000 | s | 3600.000 | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C02593/1 | 0.000 s | Holding brake: Actual value monitoring<br>• Time in which the actual value is supposed to reach the threshold for closing the brake if the setpoint has already reached the threshold.<br>• Time > 0 s: If the actual speed value has not reached the threshold within the time for brake application, the holding brake is applied by control.<br>• Time = 0 s: The brake is only applied by control when the actual speed has reached the application threshold. |
| C02593/2 | 0.000 s | Holding brake: Application delay<br>• Time by which the control process for holding brake application is delayed. The time expires when the speed setpoint has reached the switching threshold for application.<br>• With positioning processes, a continuous application and release of the holding brake can thus be suppressed for the set time. |
| C02593/3 | 0.000 s | Holding brake: Reserved |
| C02593/4 | 0.000 s | Holding brake: Reserved |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000

**C02607**

| Parameter \| Name: **C02607 \| Holding brake: Status** | | | Data type: UNSIGNED_16 Index: $21968_d$ = $55D0_h$ |
|---|---|---|---|
| Switching status of the holding brake control | | | ▶ Holding brake control |
| **Display area** (min. hex value \| max. hex value) | | | |
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** | | | **Info** |
| Bit 0 | Brake applied | | Holding brake is completely applied |
| Bit 1 | Brake released | | Holding brake is completely released |
| Bit 2 | Feedforward control active | | Feedforward control for holding of the load via the motor is active before the holding brake releases. |
| Bit 3 | Closing active | | The brake closing time (C02589/1) expires |
| Bit 4 | Forced release active | | In case of automatic operation of the holding brake control, the brake is directly released via the MCK input *bBrkRelease* = TRUE |
| Bit 5 | Release active | | The brake release time (C02589/2) expires |
| Bit 6 | Setpoint synchronisation active | | A speed setpoint at the MCK is approached along a defined ramp after brake release |
| Bit 7 | Signalling contact error | | The status monitoring has tripped. |
| Bit 8 | Reserved | | |
| Bit 9 | Reserved | | |
| Bit 10 | Reserved | | |
| Bit 11 | Reserved | | |
| Bit 12 | Reserved | | |
| Bit 13 | Reserved | | |
| Bit 14 | Reserved | | |
| Bit 15 | Reserved | | |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT | | | |

**C02610**

| Parameter \| Name: **C02610 \| MCK: Accel./decel. times** | | | Data type: UNSIGNED_32 Index: $21965_d$ = $55CD_h$ |
|---|---|---|---|
| Ramp times for speed setpoint synchronisation | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.000 | s | 999.999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02610/1 | 2.000 s | | Holding brake: ramp time synchr.<br>• Ramp time for the synchronisation process to setpoint speed after the brake opening time has elapsed<br>**Revised behaviour from version 02.00.00:**<br>• The setting made here is only effective if bit 6 "SyncRampe L_NSet_1" in C02582 is set to "0".<br>• In the Lenze setting of C02582 (Bit 6 = "1"), the ramp time of the effective acceleration of the ramp function generator (L_NSet_1) is used.<br>▶ Holding brake control |
| C02610/2 | 2.000 s | | MCK: Ramp time synchr. setpoint<br>• Time for synchronisation ramps between setpoint jumps occurring through the exceedance of minimum and maximum speed setpoint limit ranges. |
| C02610/3 | 2.000 s | | MCK: SM stopping ramp |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000 | | | |

**C02611**

| Parameter \| Name: **C02611 \| MCK: Limitations** | Data type: INTEGER_16 Index: $21964_d$ = $55CC_h$ |
|---|---|

Speed setpoint limits for the determination of limited validity ranges

**Note:**
Traversing with setpoints through resulting blocking zones is executed with the ramp set in C02610/2.
▶ Min/Max speed

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0.00 | % | 199.99 | |

| Subcodes | Lenze setting | Info |
|---|---|---|
| C02611/1 | 199.99 % | MCK: Pos. max. speed<br>• Upper limit of the speed setpoint limitation in positive direction of rotation. |
| C02611/2 | 0.00 % | MCK: Pos. min. speed<br>• Lower limit of the speed setpoint limitation in positive direction of rotation. |
| C02611/3 | 0.00 % | MCK: Neg. min. speed<br>• Lower limit of the speed setpoint limitation in negative direction of rotation. |
| C02611/4 | 199.99 % | MCK: Neg. max. speed<br>• Upper limit of the speed setpoint limitation in negative direction of rotation. |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100

**C02652**

| Parameter \| Name: **C02652 \| Memorise positions at power-off** | Data type: UNSIGNED_16 Index: $21923_d$ = $55A3_h$ |
|---|---|

From version 02.00.00

| **Setting range** (min. hex value \| max. hex value) | | | **Lenze setting** |
|---|---|---|---|
| 0x0000 | | 0xFFFF | **0x0000** (decimal: 0) |

| **Value is bit-coded:** (☑ = bit set) | |
|---|---|
| Bit 0 ☐ | Actual MCTRL position |
| Bit 1 ☐ | Reserved |
| Bit 2 ☐ | Reserved |
| Bit 3 ☐ | Reserved |
| Bit 4 ☐ | Reserved |
| Bit 5 ☐ | Reserved |
| Bit 6 ☐ | Reserved |
| Bit 7 ☐ | Reserved |
| Bit 8 ☐ | Reserved |
| Bit 9 ☐ | Reserved |
| Bit 10 ☐ | Reserved |
| Bit 11 ☐ | Reserved |
| Bit 12 ☐ | Reserved |
| Bit 13 ☐ | Reserved |
| Bit 14 ☐ | Reserved |
| Bit 15 ☐ | Reserved |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

**C02810**

| Parameter \| Name: **C02810 \| TP: Edge selection** | | Data type: UNSIGNED_8 Index: $21765_d$ = $5505_h$ |
|---|---|---|

▶ Touch-probe detection

| Selection list | | |
|---|---|---|
| 0 | Off | |
| 1 | Rising | |
| 2 | Falling | |
| 3 | Rising and falling | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02810/1 | 0: Off | Reserved |
| C02810/2 | 0: Off | Reserved |
| C02810/3 | 0: Off | TPDigIn3: Edge selection |
| C02810/4 | 0: Off | TPDigIn4: Edge selection |
| C02810/5 | 0: Off | TPDigIn5: Edge selection |
| C02810/6 | 0: Off | TPDigIn6: Edge selection |
| C02810/7 | 0: Off | TPDigIn7: Edge selection |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C02811**

| Parameter \| Name: **C02811 \| TP: Sensor delay** | | Data type: UNSIGNED_16 Index: $21764_d$ = $5504_h$ |
|---|---|---|

▶ Touch-probe detection

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | µs | 6999 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02811/1 | 0 µs | | Reserved |
| C02811/2 | 0 µs | | Reserved |
| C02811/3 | 0 µs | | TPDigIn3: Sensor delay |
| C02811/4 | 0 µs | | TPDigIn4: Sensor delay |
| C02811/5 | 0 µs | | TPDigIn5: Sensor delay |
| C02811/6 | 0 µs | | TPDigIn6: Sensor delay |
| C02811/7 | 0 µs | | TPDigIn7: Sensor delay |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C02812**

| Parameter \| Name: **C02812 \| TP: Position offset** | | | Data type: INTEGER_32 Index: $21763_d$ = $5503_h$ |
|---|---|---|---|

▸ Touch-probe detection

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02812/1 | 0.0000 units | | Reserved |
| C02812/2 | 0.0000 units | | Reserved |
| C02812/3 | 0.0000 units | | TPDigIn3: Pos offset |
| C02812/4 | 0.0000 units | | TPDigIn4: Pos offset |
| C02812/5 | 0.0000 units | | TPDigIn5: Pos offset |
| C02812/6 | 0.0000 units | | TPDigIn6: Pos offset |
| C02812/7 | 0.0000 units | | TPDigIn7: Pos offset |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C02813**

| Parameter \| Name: **C02813 \| TP: Pos. window start** | | | Data type: INTEGER_32 Index: $21762_d$ = $5502_h$ |
|---|---|---|---|

▸ Touch-probe detection

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02813/1 | -214748.3647 units | | TPDigIn3: Window start |
| C02813/2 | -214748.3647 units | | TPDigIn4: Window start |
| C02813/3 | -214748.3647 units | | TPDigIn5: Window start |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C02814**

| Parameter \| Name: **C02814 \| TP: Pos. window end** | | | Data type: INTEGER_32 Index: $21761_d$ = $5501_h$ |
|---|---|---|---|

▸ Touch-probe detection

| Setting range (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -214748.3647 | units | 214748.3647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02814/1 | 214748.3647 units | | TPDigIn3: Window end |
| C02814/2 | 214748.3647 units | | TPDigIn4: Window end |
| C02814/3 | 214748.3647 units | | TPDigIn5: Window end |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

## C02815

| Parameter \| Name: C02815 \| TP: Position source | | Data type: UNSIGNED_8 Index: $21760_d$ = $5500_h$ |
|---|---|---|
| | | ▶ Touch-probe detection |
| **Selection list** | | |
| 0 | Position encoder actual value | |
| 1 | Pos DigIn1/2 | |
| 2 | Pos resolver | |
| 3 | Pos MultiEncoder | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02815/1 | 0: Position encoder actual value | Reserved |
| C02815/2 | 0: Position encoder actual value | Reserved |
| C02815/3 | 0: Position encoder actual value | TPDigIn3: Position source |
| C02815/4 | 0: Position encoder actual value | TPDigIn4: Position source |
| C02815/5 | 0: Position encoder actual value | TPDigIn5: Position source |
| C02815/6 | 0: Position encoder actual value | TPDigIn6: Position source |
| C02815/7 | 0: Position encoder actual value | TPDigIn7: Position source |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

## C02816

| Parameter \| Name: C02816 \| TP: Signal counter | | | Data type: UNSIGNED_16 Index: $21759_d$ = $54FF_h$ |
|---|---|---|---|
| | | | ▶ Touch-probe detection |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | | 65535 | |
| **Subcodes** | | | **Info** |
| C02816/1 | | | Reserved |
| C02816/2 | | | Reserved |
| C02816/3 | | | TPDigIn3: Signal counter |
| C02816/4 | | | TPDigIn4: Signal counter |
| C02816/5 | | | TPDigIn5: Signal counter |
| C02816/6 | | | TPDigIn6: Signal counter |
| C02816/7 | | | TPDigIn7: Signal counter |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

## C02817

| Parameter \| Name: C02817 \| TP: TouchProbe position | | | Data type: INTEGER_32 Index: $21758_d$ = $54FE_h$ |
|---|---|---|---|

Touch-probe detection

| Display range (min. value \| unit \| max. value) | | |
|---|---|---|
| -214748.3647 | units | 214748.3647 |

| Subcodes | Info |
|---|---|
| C02817/1 | Reserved |
| C02817/2 | Reserved |
| C02817/3 | TPDigIn3: TP position |
| C02817/4 | TPDigIn4: TP position |
| C02817/5 | TPDigIn5: TP position |
| C02817/6 | TPDigIn6: TP position |
| C02817/7 | TPDigIn7: TP position |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10000

**C02830**

| Parameter \| Name: **C02830 \| DIx: Debounce time** | | Data type: UNSIGNED_8 Index: $21745_d$ = $54F1_h$ |
|---|---|---|
| Debounce times for the digital inputs | | |
| | | Digital input terminals |
| **Selection list** | | |
| 0 | 0.00 ms | |
| 1 | 0.25 ms | |
| 2 | 0.50 ms | |
| 3 | 0.75 ms | |
| 4 | 1.00 ms | |
| 5 | 1.25 ms | |
| 6 | 1.50 ms | |
| 7 | 1.75 ms | |
| 8 | 2.00 ms | |
| 10 | 2.50 ms | |
| 12 | 3.00 ms | |
| 14 | 3.50 ms | |
| 16 | 4.00 ms | |
| 18 | 4.50 ms | |
| 20 | 5.00 ms | |
| 22 | 5.50 ms | |
| 24 | 6.00 ms | |
| 28 | 7.00 ms | |
| 32 | 8.00 ms | |
| 36 | 9.00 ms | |
| 40 | 10.0 ms | |
| 44 | 11.0 ms | |
| 48 | 12.0 ms | |
| 52 | 13.0 ms | |
| 56 | 14.0 ms | |
| 64 | 16.0 ms | |
| 72 | 18.0 ms | |
| 80 | 20.0 ms | |
| 88 | 22.0 ms | |
| 96 | 24.0 ms | |
| 104 | 26.0 ms | |
| 112 | 28.0 ms | |
| 120 | 30.0 ms | |
| 128 | 32.0 ms | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02830/1 | 1: 0.25 ms | DI1 ... DI7: Debounce time |
| C02830/... | | |
| C02830/7 | | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02840**

| Parameter \| Name: **C02840 \| CountInx: Parameter** | | | Data type: UNSIGNED_32 Index: $21735_d$ = $54E7_h$ |
|---|---|---|---|
| Starting and comparison values for digital count inputs | | | |
| | | | ▸ Use DI1(6) as counting input |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 2147483647 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02840/1 | 0 incr. | | CountIn1: Starting value |
| C02840/2 | 65535 incr. | | CountIn1: Comparison value |
| C02840/3 | 0 incr. | | CountIn6: Starting value |
| C02840/4 | 65535 incr. | | CountIn6: Comparison value |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02841**

| Parameter \| Name: **C02841 \| CountInx: Counter content** | | | Data type: UNSIGNED_32 Index: $21734_d$ = $54E6_h$ |
|---|---|---|---|
| Display of the current counter content of the digital count inputs | | | |
| | | | ▸ Use DI1(6) as counting input |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0 | Incr. | 2147483647 | |
| **Subcodes** | | | **Info** |
| C02841/1 | | | CountIn1: Counter content |
| C02841/2 | | | CountIn6: Counter content |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02842**

| Parameter \| Name: **C02842 \| FreqInxx: Offset** | | | Data type: INTEGER_16 Index: $21733_d$ = $54E5_h$ |
|---|---|---|---|
| Offset for digital frequency inputs | | | |
| | | | ▸ Using DI1(6) and DI2(7) as frequency inputs |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02842/1 | 0.00 % | | FreqIn12: Offset |
| C02842/2 | 0.00 % | | FreqIn67: Offset |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02843**

| Parameter \| Name: **C02843 \| FreqInxx: Gain** | | | Data type: INTEGER_16 Index: $21732_d$ = $54E4_h$ |
|---|---|---|---|
| Gain for digital frequency inputs | | | |
| | | | ▸ Using DI1(6) and DI2(7) as frequency inputs |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -199.99 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02843/1 | 100.00 % | | FreqIn12: Gain |
| C02843/2 | 100.00 % | | FreqIn67: Gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02844**

| Parameter \| Name: **C02844 \| FreqIn12: Function** | | Data type: UNSIGNED_8 Index: $21731_d$ = $54E3_h$ |
|---|---|---|
| | | ▸ Output of the encoder position of the DI1/DI2 frequency input |
| **Selection list** | | |
| 0 | Loading with level | |
| 1 | Loading with edge | |
| 2 | Loading with level + reset | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02844/1 | 0: Loading with level | FreqIn12: PosIn function |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02845**

| Parameter \| Name: **C02845 \| FreqIn12: PosIn comparison value** | | | Data type: INTEGER_32 Index: $21730_d$ = $54E2_h$ |
|---|---|---|---|
| | | | ▸ Output of the encoder position of the DI1/DI2 frequency input |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0 | Incr. | 2147418112 | **0 incr.** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02853**

| Parameter \| Name: **C02853 \| PSM: Lss saturation characteristic** | | | Data type: UNSIGNED_8 Index: $21722_d$ = $54DA_h$ |
|---|---|---|---|
| | | | ▸ Correction via saturation characteristic |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0 | % | 255 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02853/1 | 100 % | | PSM: Lss saturation characteristic |
| C02853/2 | 100 % | | PSM: Lss saturation characteristic |
| C02853/3 | 100 % | | PSM: Lss saturation characteristic |
| C02853/4 | 100 % | | PSM: Lss saturation characteristic |
| C02853/5 | 100 % | | PSM: Lss saturation characteristic |
| C02853/6 | 100 % | | PSM: Lss saturation characteristic |
| C02853/7 | 100 % | | PSM: Lss saturation characteristic |
| C02853/8 | 100 % | | PSM: Lss saturation characteristic |
| C02853/9 | 100 % | | PSM: Lss saturation characteristic |
| C02853/10 | 100 % | | PSM: Lss saturation characteristic |
| C02853/11 | 100 % | | PSM: Lss saturation characteristic |
| C02853/12 | 100 % | | PSM: Lss saturation characteristic |
| C02853/13 | 100 % | | PSM: Lss saturation characteristic |
| C02853/14 | 100 % | | PSM: Lss saturation characteristic |
| C02853/15 | 100 % | | PSM: Lss saturation characteristic |
| C02853/16 | 100 % | | PSM: Lss saturation characteristic |
| C02853/17 | 100 % | | PSM: Lss saturation characteristic |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02855**

| Parameter \| Name: **C02855 \| PSM: Imax Lss saturation characteristic** | | | Data type: UNSIGNED_16 Index: $21720_d$ = $54D8_h$ |
|---|---|---|---|
| | | | ▶ Correction via saturation characteristic |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| 0.0 | A | 3000.0 | **3000.0 A** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10 | | | |

**C02859**

| Parameter \| Name: **C02859 \| PSM: Activate Lss saturation char.** | | Data type: UNSIGNED_8 Index: $21716_d$ = $54D4_h$ |
|---|---|---|
| | | ▶ Correction via saturation characteristic |
| **Selection list** (Lenze setting printed in bold) | | |
| **0** | **Off** | |
| 1 | On | |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | |

**C02862**

| Parameter \| Name: **C02862 \| Resolver gain** | | | Data type: UNSIGNED_16 Index: $21713_d$ = $54D1_h$ |
|---|---|---|---|
| | | | ▶ Encoder/feedback system: Resolver |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 199.99 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02862/1 | 100.00 % | | Resolver: Cos gain |
| C02862/2 | 100.00 % | | Resolver: Sin gain |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02863**

| Parameter \| Name: **C02863 \| Resolver: Phase error** | | | Data type: INTEGER_16 Index: $21712_d$ = $54D0_h$ |
|---|---|---|---|
| | | | ▶ Encoder/feedback system: Resolver |
| **Setting range** (min. value \| unit \| max. value) | | | **Lenze setting** |
| -199.99 | % | 199.99 | **0.00 %** |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02866**

| Parameter \| Name: **C02866 \| MCTRL: Special settings** | | Data type: UNSIGNED_8 Index: $21709_d$ = $54CD_h$ |
|---|---|---|
| Activation of special functions of the internal motor control | | |
| **Selection list** | | |
| 0 | No | |
| 1 | Yes | |
| **Subcodes** | **Lenze setting** | **Info** |
| C02866/1 | 1: Yes | Motor ident.: Current controller par. C075 C076 ▶ Automatic motor data identification |
| C02866/2 | 0: No | From version 02.00.00 ▶ Motor phase error monitoring before operation |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | |

**C02867**

| Parameter \| Name: **C02867 - Identification procedure** | | | Data type: UNSIGNED_8 Index: $21708_d$ = $54CC_h$ |
|---|---|---|---|
| Selection of the identification procedure for motor parameter identification | | | |
| | | | ▶ Automatic motor parameter identification |
| **Selection list** | | **Info** | |
| 0 | automatic | Automatic selection of the optimum identification procedure:<br>• For synchronous motors, the extended identification procedure is always used.<br>• For asynchronous motors with a rated motor power of up to 11 kW, the basic identification procedure is used.<br>• For asynchronous motors with a rated motor power of more than 11 kW, the extended identification procedure is used. | |
| 1 | standard identification | • Only for asynchronous motors.<br>• Duration approx. 30 s | |
| 2 | extended identification | • Stands out due to increased accuracy of the determined motor parameters.<br>• Also supports synchronous motors and asynchronous motors with a power of more than 11 kW.<br>• Duration approx. 80 s | |
| **Subcodes** | **Lenze setting** | **Info** | |
| C02867/1 | 0: automatic | Motor parameter identification: Process | |
| ☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 1 | | | |

**C02870**

| Parameter \| Name: **C02870 \| PLI without motion: Optimisation factor** | | | Data type: INTEGER_16 Index: $21705_d$ = $54C9_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| | | | ▶ Pole position identification without motion |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | % | 300.00 | |
| **Subcodes** | | | **Info** |
| C02870/1 | | | PLI without motion: Optimisation factor |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02871**

| Parameter \| Name: **C02871 \| PLI without motion: Running time** | | | Data type: INTEGER_16 Index: $21704_d$ = $54C8_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| | | | ▶ Pole position identification without motion |
| **Display range** (min. value \| unit \| max. value) | | | |
| 0.00 | ms | 300.00 | |
| **Subcodes** | | | **Info** |
| C02871/1 | | | PLI without motion: Running time |
| ☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02872**

| Parameter \| Name: **C02872 \| PLI without motion: Adaptation of time duration** | | Data type: INTEGER_8 Index: $21703_d$ = $54C7_h$ |
|---|---|---|

From version 02.00.00

▸ Pole position identification without motion

| **Setting range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| -10 | | 10 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02872/1 | 0 | | PLI without motion: Adaptation of time duration |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C02873**

| Parameter \| Name: **C02873 \| PLI without motion: Ident. el. rotor displ. angle** | | Data type: INTEGER_16 Index: $21702_d$ = $54C6_h$ |
|---|---|---|

From version 02.00.00

▸ Pole position identification without motion

| **Display range** (min. value \| unit \| max. value) | | | |
|---|---|---|---|
| 0 | ° | 360 | |
| **Subcodes** | | | **Info** |
| C02873/1 | | | PLI without motion: Ident. el. rotor displ. angle |

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1

**C02874**

| Parameter \| Name: **C02874 \| PLI without motion** | | Data type: UNSIGNED_16 Index: $21701_d$ = $54C5_h$ |
|---|---|---|

From version 02.00.00

▸ Pole position identification without motion

| **Setting range** (min. hex value \| max. hex value) | | | |
|---|---|---|---|
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded:** (☑ = bit set) | | | |
| Bit 0 ☐ | for SLPSM with controller enable | | |
| Bit 1 ☐ | for SC PSM with mains on | | |
| Bit 2 ☐ | for SC PSM with controller enable | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02874/1 | 1 | | PLI without motion |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

**C02875**

| Parameter \| Name: **C02875 \| PLI without motion: Adaptation of ident angle** | | | Data type: INTEGER_8 Index: $21700_d$ = $54C4_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| | | | Pole position identification without motion |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -100 | ° | 100 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02875/1 | 0 ° | | PLI without motion: Adaptation of ident angle |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02876**

| Parameter \| Name: **C02876 \| PSM: Max. motor temperature** | | | Data type: UNSIGNED_8 Index: $21699_d$ = $54C3_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| 90 | °C | 200 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02876/1 | 150 °C | | PSM: Max. motor temperature |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1 | | | |

**C02877**

| Parameter \| Name: **C02877 \| PSM temperature coefficient** | | | Data type: INTEGER_16 Index: $21698_d$ = $54C2_h$ |
|---|---|---|---|
| From version 02.00.00 | | | |
| **Setting range** (min. value \| unit \| max. value) | | | |
| -0.30 | %/°C | 0.00 | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02877/1 | -0.11 %/°C | | PSM temperature coefficient |
| ☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100 | | | |

**C02878**

| Parameter \| Name: **C02878 \| KTY motor temperature compensation** | Data type: UNSIGNED_16 Index: $21697_d$ = $54C1_h$ |
|---|---|

From version 02.00.00

A temperature compensation over the detected motor temperature (display in C00063/1) serves to compensate inaccuracies in the output torque within the motor control in case of temperature changes of the asynchronous or synchronous motor.

**Note!**

In the Lenze setting, the temperature compensation within the motor control is activated. The temperature compensation, however, is only active with speed encoder selection "3: Multi encoder" or "4: Resolver" in C00495 as well as error-free KTY temperature detection (display in C00063/1 ≠ 255 °C).

▶ Temperature compensation within the motor control

| **Setting range** (min. hex value \| max. hex value) | | | |
|---|---|---|---|
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded: (☑ = bit set)** | | | |
| Bit 0 ☐ | for SC PSM | | |
| Bit 1 ☐ | for SC ASM | | |
| Bit 2 ☐ | Reserved | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |

| **Subcodes** | **Lenze setting** | **Info** |
|---|---|---|
| C02878/1 | 3 | KTY motor temperature compensation |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

**C02879**

| Parameter \| Name: **C02879 \| Slip calculation from equivalent circuit diagram** | | | Data type: UNSIGNED_16 Index: $21696_d$ = $54C0_h$ |
|---|---|---|---|

From version 02.00.00
In order to achieve a better speed stability and torque accuracy, the slip calculation can be either derived from the motor nameplate data (e.g. rated motor speed) or the motor equivalent circuit diagram data (stator resistance, rotor resistance etc.).

| **Setting range** (min. hex value \| max. hex value) | | | |
|---|---|---|---|
| 0x0000 | | 0xFFFF | |
| **Value is bit-coded: (☑ = bit set)** | | | **Info** |
| Bit 0 ☐ | SLVC | | ▸ Slip calculation for SLVC |
| Bit 1 ☐ | SC ASM | | ▸ Slip calculation for SC |
| Bit 2 ☐ | Reserved | | |
| Bit 3 ☐ | Reserved | | |
| Bit 4 ☐ | Reserved | | |
| Bit 5 ☐ | Reserved | | |
| Bit 6 ☐ | Reserved | | |
| Bit 7 ☐ | Reserved | | |
| Bit 8 ☐ | Reserved | | |
| Bit 9 ☐ | Reserved | | |
| Bit 10 ☐ | Reserved | | |
| Bit 11 ☐ | Reserved | | |
| Bit 12 ☐ | Reserved | | |
| Bit 13 ☐ | Reserved | | |
| Bit 14 ☐ | Reserved | | |
| Bit 15 ☐ | Reserved | | |
| **Subcodes** | **Lenze setting** | | **Info** |
| C02879/1 | 2 | | Slip calculation from equivalent circuit diagram |

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT

**C02994**

| Parameter \| Name: **C02994 \| FB xy position** | Data type: UNSIGNED_32 Index: $21581_d$ = $544D_h$ |
|---|---|

**This code is for device-internal use only and must not be written to by the user!**

**C02995**

| Parameter \| Name: **C02995 \| FB display InputOutput** | Data type: UNSIGNED_32 Index: $21580_d$ = $544C_h$ |
|---|---|

**This code is for device-internal use only and must not be written to by the user!**

**C02996**

| Parameter \| Name: **C02996 \| FB display InputOutput2** | Data type: UNSIGNED_32 Index: $21579_d$ = $544B_h$ |
|---|---|

**This code is for device-internal use only and must not be written to by the user!**

## 16.2.1 Selection lists for connection parameters

### 16.2.1.1 Selection list - analog signals

This selection list is relevant for the following parameters:

| Parameter | |
|---|---|
| C00410 | L_SignalMonitor_a: Signal sources |
| C00620 | System connection list: 16-bit |
| C00700 | LA_NCtrl: Analog connection list |
| C00710 | LA_TabPos: Analog connection list |
| C00760 | LA_SwitchPos: Analog connection list |

| Selection list - analog signals | |
|---|---|
| 0 | Not connected |
| 1000 | LA_nCtrl_wDriveControlStatus |
| 1001 | LA_nCtrl_wFailNumber |
| 1002 | LA_nCtrl_nMotorCurrent_a |
| 1003 | LA_nCtrl_nMotorSpeedAct_a |
| 1006 | LA_nCtrl_nGPAnalogSwitchOut_a |
| 1007 | LA_nCtrl_nGPArithmetikOut_a |
| 1008 | LA_nCtrl_nGPMulDivOut_a |
| 1009 | LA_nCtrl_nGPSignalOut1_a |
| 1010 | LA_nCtrl_nGPSignalOut2_a |
| 1011 | LA_nCtrl_nGPSignalOut3_a |
| 1012 | LA_nCtrl_nGPSignalOut4_a |
| 1013 | LA_nCtrl_nMotorTorqueAct_a |
| 1014 | LA_nCtrl_nDCVoltage_a |
| 1015 | LA_nCtrl_nMotorVoltage_a |
| 1016 | LA_nCtrl_nMotorSpeedSet_a |
| 1017 | LA_nCtrl_wFailTypeDomain |
| 1023 | LA_nCtrl_wFreeOut1 |
| 1024 | LA_nCtrl_wFreeOut2 |
| 1025 | LA_nCtrl_wFreeOut3 |
| 1026 | LA_nCtrl_wFreeOut4 |
| 1100 | LA_TabPos_wDriveControlStatus |
| 1101 | LA_TabPos_wFailNoLow |
| 1102 | LA_TabPos_wFailNoHigh |
| 1103 | LA_TabPos_nMotorCurrent_a |
| 1104 | LA_TabPos_nMotorSpeedSet_a |
| 1105 | LA_TabPos_nMotorSpeedAct_a |
| 1106 | LA_TabPos_nMotorTorqueAct_a |
| 1107 | LA_TabPos_nDCVoltage_a |
| 1108 | LA_TabPos_nMotorVoltage_a |
| 1109 | LA_TabPos_wMckState1 |
| 1110 | LA_TabPos_wMckState2 |
| 1111 | LA_TabPos_wMckActOperationMode |
| 1112 | LA_TabPos_wActProfileNo |
| 1113 | LA_TabPos_wActPosMode |
| 1114 | LA_TabPos_nGPAnalogSwitchOut_a |
| 1115 | LA_TabPos_nGPArithmetikOut_a |
| 1116 | LA_TabPos_nGPMulDivOut_a |
| 1117 | LA_TabPos_nGPSignalOut1_a |
| 1118 | LA_TabPos_nGPSignalOut2_a |
| 1119 | LA_TabPos_nGPSignalOut3_a |
| 1120 | LA_TabPos_nGPSignalOut4_a |
| 1121 | LA_TabPos_wGPCounter1Out |
| 1122 | LA_TabPos_wFreeOut1 |
| 1123 | LA_TabPos_wFreeOut2 |
| 1124 | LA_TabPos_wFreeOut3 |
| 1125 | LA_TabPos_wFreeOut4 |
| 1126 | LA_TabPos_nPosCtrlOutLimit_a |
| 1127 | LA_TabPos_nPosCtrlPAdapt_a |
| 1128 | LA_TabPos_wPosOutUnitsLW |
| 1129 | LA_TabPos_wPosOutUnitsHW |
| 1200 | LA_SwitchPos_wDriveControlStatus |
| 1201 | LA_SwitchPos_wFailNoLow |
| 1202 | LA_SwitchPos_wFailNoHigh |
| 1203 | LA_SwitchPos_nMotorCurrent_a |
| 1204 | LA_SwitchPos_nMotorSpeedSet_a |
| 1205 | LA_SwitchPos_nMotorSpeedAct_a |
| 1206 | LA_SwitchPos_nMotorTorqueAct_a |
| 1207 | LA_SwitchPos_nDCVoltage_a |
| 1208 | LA_SwitchPos_nMotorVoltage_a |
| 1209 | LA_SwitchPos_nGPAnalogSwitchOut_a |
| 1210 | LA_SwitchPos_nGPArithmetikOut_a |
| 1211 | LA_SwitchPos_nGPMulDivOut_a |
| 1212 | LA_SwitchPos_nGPSignalOut1_a |
| 1213 | LA_SwitchPos_nGPSignalOut2_a |
| 1214 | LA_SwitchPos_nGPSignalOut3_a |
| 1215 | LA_SwitchPos_nGPSignalOut4_a |
| 1221 | LA_SwitchPos_wFreeOut1 |
| 1222 | LA_SwitchPos_wFreeOut2 |
| 1223 | LA_SwitchPos_wFreeOut3 |
| 1224 | LA_SwitchPos_wFreeOut4 |
| 16000 | AIN1_Out |
| 16001 | AIN2_Out |
| 16002 | CAN1_wCtrl |
| 16003 | CAN1_wIn2 |
| 16004 | CAN1_wIn3 |
| 16005 | CAN1_wIn4 |
| 16006 | CAN2_wIn1 |
| 16007 | CAN2_wIn2 |
| 16008 | CAN2_wIn3 |
| 16009 | CAN2_wIn4 |
| 16010 | CAN3_wIn1 |
| 16011 | CAN3_wIn2 |
| 16012 | CAN3_wIn3 |
| 16013 | CAN3_wIn4 |
| 16014 | DIGIN_wCountIn1_LW |

| Selection list - analog signals | |
|---|---|
| 16015 | DIGIN_wCountIn1_HW |
| 16016 | DIGIN_nFreqIn12_a |
| 16017 | DIGIN_nFreqIn12_v |
| 16018 | DIGIN_wCountIn6_LW |
| 16019 | DIGIN_wCountIn6_HW |
| 16020 | DIGIN_nFreqIn67_a |
| 16021 | DIGIN_nFreqIn67_v |
| 16100 | LS_DataAccess_Out1 |
| 16101 | LS_DataAccess_Out2 |
| 16102 | LS_DataAccess_Out3 |
| 16103 | LS_DataAccess_Out4 |
| 16104 | LP_MciIn_wCtrl |
| 16105 | LP_MciIn_wIn2 |
| 16106 | LP_MciIn_wIn3 |
| 16107 | LP_MciIn_wIn4 |
| 16108 | LP_MciIn_wIn5 |
| 16109 | LP_MciIn_wIn6 |
| 16110 | LP_MciIn_wIn7 |
| 16111 | LP_MciIn_wIn8 |
| 16112 | LP_MciIn_wIn9 |
| 16113 | LP_MciIn_wIn10 |
| 16114 | LP_MciIn_wIn11 |
| 16115 | LP_MciIn_wIn12 |
| 16116 | LP_MciIn_wIn13 |
| 16117 | LP_MciIn_wIn14 |
| 16118 | LP_MciIn_wIn15 |
| 16119 | LP_MciIn_wIn16 |
| 16120 | LS_Keypad_nTorqueMotLim_a |
| 16121 | LS_Keypad_nTorqueGenLim_a |
| 16122 | LS_Keypad_nMainSetValue_a |
| 16123 | LS_CANManagement_wNodeID |
| 16130 | LS_ParReadWrite_1_wOutHWord |
| 16131 | LS_ParReadWrite_1_wOutLWord |
| 16132 | LS_ParReadWrite_2_wOutHWord |
| 16133 | LS_ParReadWrite_2_wOutLWord |
| 16134 | LS_ParReadWrite_3_wOutHWord |
| 16135 | LS_ParReadWrite_3_wOutLWord |
| 16136 | LS_ParReadWrite_4_wOutHWord |
| 16137 | LS_ParReadWrite_4_wOutLWord |
| 16138 | LS_ParReadWrite_5_wOutHWord |
| 16139 | LS_ParReadWrite_5_wOutLWord |
| 16140 | LS_ParReadWrite_6_wOutHWord |
| 16141 | LS_ParReadWrite_6_wOutLWord |
| 16170 | LS_AxisBusIn_wLine1 |
| 16171 | LS_AxisBusIn_wLine2 |
| 16172 | LS_AxisBusIn_wLine3 |
| 16173 | LS_AxisBusIn_wCas1 |
| 16174 | LS_AxisBusIn_wCas2 |
| 16175 | LS_AxisBusIn_wCas3 |
| 16176 | LS_AxisBusIn_wCas4 |
| 16177 | LS_AxisBusAux_wAuxIn1 |
| 16178 | LS_AxisBusAux_wAuxIn2 |

| Selection list - analog signals | |
|---|---|
| 16179 | LS_AxisBusAux_wAuxIn3 |
| 16180 | LS_AxisBusAux_wAuxIn4 |
| 16181 | LS_AxisBusAux_wSlaveNo |
| 16300 | LS_MultiEncoder_nActSpeed_v |
| 16301 | LS_MultiEncoder_nActSpeed_a |
| 16302 | LS_MultiEncoder_wHighWord |
| 16303 | LS_MultiEncoder_wLowWord |
| 16320 | LS_BUSEncoder_nActSpeed_v |
| 16321 | LS_BUSEncoder_nActSpeed_a |
| 16322 | LS_BUSEncoder_wHighWord |
| 16323 | LS_BUSEncoder_wLowWord |
| 16340 | LS_Resolver_nActSpeed_v |
| 16341 | LS_Resolver_nActSpeed_a |
| 16350 | LS_RetainData_wOut1 |
| 16351 | LS_RetainData_wOut2 |
| 16352 | LS_RetainData_wOut3 |
| 16353 | LS_RetainData_wOut4 |
| 20000 | LS_ParFix_Pos100_a |
| 20001 | LS_ParFix_Neg100_a |
| 20002 | LS_ParFix_Pos199_99_a |
| 20003 | LS_ParFix_Neg199_99_a |
| 20004 | LS_ParFix_65535 |
| 20005 | LS_ParFix_wDriveCtrl |
| 20010 | LS_ParFree_nC472_1_a |
| 20011 | LS_ParFree_nC472_2_a |
| 20012 | LS_ParFree_nC472_3_a |
| 20013 | LS_ParFree_nC472_4_a |
| 20014 | LS_ParFree_nC472_5_a |
| 20015 | LS_ParFree_nC472_6_a |
| 20016 | LS_ParFree_nC472_7_a |
| 20017 | LS_ParFree_nC472_8_a |
| 20018 | LS_ParFree_nC473_1_v |
| 20019 | LS_ParFree_nC473_2_v |
| 20020 | LS_ParFree_nC473_3_v |
| 20021 | LS_ParFree_nC473_4_v |
| 20022 | LS_ParFree_nC473_5_v |
| 20023 | LS_ParFree_nC473_6_v |
| 20024 | LS_ParFree_nC473_7_v |
| 20025 | LS_ParFree_nC473_8_v |
| 20026 | LS_ParFree_wC471_1 |
| 20027 | LS_ParFree_wC471_2 |
| 20028 | LS_ParFree_wC471_3 |
| 20029 | LS_ParFree_wC471_4 |
| 20030 | LS_ParFree_wC471_5 |
| 20031 | LS_ParFree_wC471_6 |
| 20032 | LS_ParFree_wC471_7 |
| 20033 | LS_ParFree_wC471_8 |
| 20034 | LS_ParFree_wC471_9 |
| 20035 | LS_ParFree_wC471_10 |
| 20036 | LS_ParFree_wC471_11 |
| 20037 | LS_ParFree_wC471_12 |
| 20038 | LS_ParFree_wC471_13 |

| Selection list - analog signals | |
|---|---|
| 20039 | LS_ParFree_wC471_14 |
| 20040 | LS_ParFree_wC471_15 |
| 20041 | LS_ParFree_wC471_16 |
| 20042 | LS_ParFree_wC471_17 |
| 20043 | LS_ParFree_wC471_18 |
| 20044 | LS_ParFree_wC471_19 |
| 20045 | LS_ParFree_wC471_20 |
| 20046 | LS_ParFree_wC471_21 |
| 20047 | LS_ParFree_wC471_22 |
| 20048 | LS_ParFree_wC471_23 |
| 20049 | LS_ParFree_wC471_24 |
| 20050 | LS_ParFree_wC471_25 |
| 20051 | LS_ParFree_wC471_26 |
| 20052 | LS_ParFree_wC471_27 |
| 20053 | LS_ParFree_wC471_28 |
| 20054 | LS_ParFree_wC471_29 |
| 20055 | LS_ParFree_wC471_30 |
| 20056 | LS_ParFree_wC471_31 |
| 20057 | LS_ParFree_wC471_32 |
| 20058 | LS_ParFree_nC472_9_a |
| 20059 | LS_ParFree_nC472_10_a |
| 20060 | LS_ParFree_nC472_11_a |
| 20061 | LS_ParFree_nC472_12_a |
| 20062 | LS_ParFree_nC472_13_a |
| 20063 | LS_ParFree_nC472_14_a |
| 20064 | LS_ParFree_nC472_15_a |
| 20065 | LS_ParFree_nC472_16_a |
| 20066 | LS_ParFree_nC476_1_a |
| 20067 | LS_ParFree_nC476_2_a |
| 20068 | LS_ParFree_nC476_3_a |
| 20069 | LS_ParFree_nC476_4_a |
| 20070 | LS_ParFree_nC476_5_a |
| 20071 | LS_ParFree_nC476_6_a |
| 20072 | LS_ParFree_nC476_7_a |
| 20073 | LS_ParFree_nC476_8_a |
| 20074 | LS_ParFree_nC476_9_a |
| 20075 | LS_ParFree_nC476_10_a |
| 20076 | LS_ParFree_nC476_11_a |
| 20077 | LS_ParFree_nC476_12_a |
| 20078 | LS_ParFree_nC476_13_a |
| 20079 | LS_ParFree_nC476_14_a |
| 20080 | LS_ParFree_nC476_15_a |
| 20081 | LS_ParFree_nC476_16_a |
| 20082 | LS_ParFix2_Pos100_a |
| 20083 | LS_ParFix2_Neg100_a |
| 20084 | LS_ParFix2_Pos199_99_a |
| 20085 | LS_ParFix2_Neg199_99_a |
| 20086 | LS_ParFix2_65535 |
| 20087 | LS_ParFix2_wDriveCtrl |
| 20088 | LS_ParFree_wC477_1 |
| 20089 | LS_ParFree_wC477_2 |
| 20090 | LS_ParFree_wC477_3 |

| Selection list - analog signals | |
|---|---|
| 20091 | LS_ParFree_wC477_4 |
| 20092 | LS_ParFree_wC477_5 |
| 20093 | LS_ParFree_wC477_6 |
| 20094 | LS_ParFree_wC477_7 |
| 20095 | LS_ParFree_wC477_8 |
| 20096 | LS_ParFree_wC477_9 |
| 20097 | LS_ParFree_wC477_10 |
| 20098 | LS_ParFree_wC477_11 |
| 20099 | LS_ParFree_wC477_12 |
| 20100 | LS_ParFree_wC477_13 |
| 20101 | LS_ParFree_wC477_14 |
| 20102 | LS_ParFree_wC477_15 |
| 20103 | LS_ParFree_wC477_16 |
| 20104 | LS_ParFree_wC477_17 |
| 20105 | LS_ParFree_wC477_18 |
| 20106 | LS_ParFree_wC477_19 |
| 20107 | LS_ParFree_wC477_20 |
| 20108 | LS_ParFree_wC477_21 |
| 20109 | LS_ParFree_wC477_22 |
| 20110 | LS_ParFree_wC477_23 |
| 20111 | LS_ParFree_wC477_24 |
| 20112 | LS_ParFree_wC477_25 |
| 20113 | LS_ParFree_wC477_26 |
| 20114 | LS_ParFree_wC477_27 |
| 20115 | LS_ParFree_wC477_28 |
| 20116 | LS_ParFree_wC477_29 |
| 20117 | LS_ParFree_wC477_30 |
| 20118 | LS_ParFree_wC477_31 |
| 20119 | LS_ParFree_wC477_32 |
| 20120 | LS_ParFree_nC478_1_v |
| 20121 | LS_ParFree_nC478_2_v |
| 20122 | LS_ParFree_nC478_3_v |
| 20123 | LS_ParFree_nC478_4_v |
| 20124 | LS_ParFree_nC478_5_v |
| 20125 | LS_ParFree_nC478_6_v |
| 20126 | LS_ParFree_nC478_7_v |
| 20127 | LS_ParFree_nC478_8_v |
| 32000 | MCTRL_nMotorSpeedAct_a |
| 32001 | MCTRL_nOutputSpeedCtrl_a |
| 32002 | MCTRL_nInputJerkCtrl_a |
| 32003 | MCTRL_nInputTorqueCtrl_a |
| 32004 | MCTRL_nMotorTorqueAct_a |
| 32005 | MCTRL_nActualFluxx_a |
| 32006 | MCTRL_nDCVoltage_a |
| 32007 | MCTRL_nStatorCurrentIS_a |
| 32008 | MCTRL_nEffCurrentIq_a |
| 32009 | MCTRL_nReaktCurrentId_a |
| 32010 | MCTRL_wMaxMotorSpeed |
| 32011 | MCTRL_wMaxMotorTorque |
| 32012 | MCTRL_nMotorVoltage_a |
| 32013 | MCTRL_nMotorFreqAct_a |
| 32014 | MCTRL_nEffSpeedSetValue_a |

| Selection list - analog signals | |
|---|---|
| 32015 | LS_DeviceMonitor_MCTRL_nIxTRate_a |
| 32016 | LS_DeviceMonitor_MCTRL_nI2xTRate_a |
| 32017 | MCTRL_nOutputPosCtrl_a |
| 32018 | MCTRL_nHlgSetValue_a |
| 32019 | MCTRL_nMotorSpeedAct_v |
| 32020 | MCTRL_nSpeedCtrlIAct_a |
| 32100 | DCTRL_wDeviceStateWord |
| 32101 | DCTRL_wDeviceAuxStateWord |
| 32102 | DCTRL_wDetermFailNoLow |
| 32103 | DCTRL_wDetermFailNoHigh |
| 32104 | DCTRL_wDetermFailNoShort |
| 32200 | MCK_nSpeedSet_v |
| 32201 | MCK_nSpeedCtrlI_a |
| 32202 | MCK_nSpeedSetValue_a |
| 32203 | MCK_nTorqueSetValue_a |
| 32204 | MCK_wActProfileNo |
| 32205 | MCK_wFollowProfileNo |
| 32206 | MCK_wMotionState1 |
| 32207 | MCK_wMotionState2 |
| 32208 | MCK_wAuxState |
| 32209 | MCK_nPWMAngleOffset |
| 32210 | MCK_nTorqueLimitAdapt_a |
| 32211 | Reserved |
| 34900 | MCTRL_OszCh1 |
| 34901 | MCTRL_OszCh2 |
| 34902 | MCTRL_OszCh3 |
| 34903 | MCTRL_OszCh4 |
| 34904 | MCTRL_Status1 |
| 34905 | MCTRL_Status2 |
| 34906 | MCTRL_Status3 |
| 34907 | LS_DeviceMonitor_wUB_24V |
| 36000 | L_Absolut _Out_1 |
| 36001 | L_AddSub_Out_1 |
| 36002 | L_OffsetGain_Out_1 |
| 36003 | L_OffsetGain_Out_2 |
| 36004 | L_OffsetGainP_1 |
| 36005 | L_OffsetGainP_2 |
| 36006 | L_GainOffset_Out_1 |
| 36007 | L_GainOffset_Out_2 |
| 36008 | L_GainOffsetP_1 |
| 36009 | L_GainOffsetP_2 |
| 36010 | L_Negation_Out_1 |
| 36011 | L_Aritmethik_Out_1 |
| 36012 | L_Aritmethik_Out_2 |
| 36013 | L_AnalogSwitch_Out_1 |
| 36014 | L_AnalogSwitch_Out_2 |
| 36015 | L_AnalogSwitch_Out_3 |
| 36016 | L_Limit_Out_1 |
| 36017 | L_Limit_Out_2 |
| 36018 | L_NSet_NOut_1 |
| 36019 | L_MotorPoti_1_Out |
| 36020 | L_PCTRL_1_Out |

| Selection list - analog signals | |
|---|---|
| 36021 | L_SigMonitor_a_Out1 |
| 36022 | L_SigMonitor_a_Out2 |
| 36023 | L_NLim_1_nOut_a |
| 36024 | L_Counter_1_wOut |
| 36025 | L_OffsetGainP_3 |
| 36026 | L_GainOffsetP_3 |
| 36027 | L_SigMonitor_a_Out3 |
| 36028 | L_SigMonitor_a_Out4 |
| 36029 | L_MulDiv_1_nOut_a |
| 36030 | L_NLim_1_wState |
| 36031 | L_NSet_1_wState |
| 36032 | L_NSet_1_nSetValue |
| 36033 | L_PT1_1_nOut_a |
| 36034 | L_Absolut _Out_2 |
| 36035 | L_AnalogSwitch_4_Out |
| 36036 | L_AnalogSwitch_5_Out |
| 36037 | L_Aritmethik_3_Out |
| 36038 | L_Aritmethik_4_Out |
| 36039 | L_Aritmethik_5_Out |
| 36045 | L_GainOffset_Out_3 |
| 36053 | L_MulDiv_2_nOut_a |
| 36054 | L_Negation_2_Out |
| 36055 | L_NLim_2_nOut_a |
| 36056 | L_NLim_2_wState |
| 36057 | L_OffsetGain_Out_3 |
| 36058 | L_PT1_2_nOut_a |
| 36059 | L_PT1_3_nOut_a |
| 36064 | L_SampleHold_1_nOut_a |
| 36065 | L_SampleHold_2_nOut_a |
| 36068 | L_Counter_2_wOut |
| 36069 | L_Counter_3_wOut |
| 36073 | L_DT1_1_nOut_a |
| 36074 | L_ConvBitsToWord_1_wOut |
| 36075 | L_ConvBitsToWord_2_wOut |
| 36076 | L_ConvBitsToWord_3_wOut |
| 36077 | L_ConvDIntToWords_1_wOutLWord |
| 36078 | L_ConvDIntToWords_1_wOutHWord |
| 36079 | L_ConvDIntToWords_2_wOutLWord |
| 36080 | L_ConvDIntToWords_2_wOutHWord |
| 36081 | L_ConvDIntToWords_3_wOutLWord |
| 36082 | L_ConvDIntToWords_3_wOutHWord |
| 36083 | L_MckCtrlInterface_1_wOutMckPosCtrl_1 |
| 36084 | L_MckCtrlInterface_1_wOutMckPosCtrl_2 |
| 36085 | L_MckCtrlInterface_1_wFailState |
| 36086 | L_MckStateInterface_1_wOperationMode |
| 36087 | L_MckStateInterface_1_wActProfileNo |
| 36088 | L_MckStateInterface_1_wActPosMode |
| 36089 | L_PosiShaftCtrlInterface_1_wOutMckPosCtrl_1 |
| 36090 | L_PosiShaftCtrlInterface_1_wOutMckPosCtrl_2 |
| 36091 | L_PCTRL_1_nPIDOut1_a |
| 36092 | L_PCTRL_1_nPIDOut2_a |
| 36093 | L_PCTRL_1_nInfluenceOut_a |

| Selection list - analog signals | |
|---|---|
| 36094 | L_Curve_1_nOut_a |
| 36095 | L_Interpolator_1_nPhdOut_v |
| 36096 | L_Interpolator_1_nNOut_a |
| 36097 | L_ConvW_1_wOut |
| 36098 | L_ConvW_2_wOut |
| 36099 | L_ConvW_3_wOut |
| 36100 | L_ConvW_4_wOut |
| 36101 | L_SRFG_1_nOut_a |
| 36102 | L_SRFG_2_nOut_a |
| 36103 | L_MckStateInterface_1_wPosUnitsLW |
| 36104 | L_MckStateInterface_1_wPosUnitsHW |
| 36105 | L_SignalSwitch_1_wOut |
| 36106 | L_SignalSwitch_2_wOut |
| 36107 | L_SignalSwitch_3_wOut |
| 36108 | L_SignalSwitch_4_wOut |
| 36109 | L_SRFG_1_nDeltaOut_a |
| 36110 | L_SRFG_2_nDeltaOut_a |
| 36111 | L_Odometer_1_wLastMeasure |
| 36112 | L_FixSet_a_1_nOut_a |
| 36113 | L_FixSet_a_1_wSelect |
| 36114 | L_FixSet_w_1_wOut |
| 36115 | L_FixSet_w_1_wSelect |
| 36116 | L_FixSet_w_2_wOut |
| 36117 | L_FixSet_w_2_wSelect |
| 36120 | L_CalcDiameter_nDiameter_a |
| 36121 | L_CalcDiameter_nReziprDiameter_a |
| 36122 | L_CalcDiameter_nNReel_v |
| 36123 | L_CalcDiameter_nDMin_a |
| 36124 | L_ProcessCtrl_nOut_a |
| 36125 | L_ProcessCtrl_nOutRed_a |
| 36126 | L_ProcessCtrl_nDeviation_a |
| 36127 | L_ProcessCtrl_nDComponent |
| 36128 | L_PhiIntegrator_nOut16 |
| 36129 | L_PhiIntegrator_nSpeedGearAdd_v |
| 36130 | L_PosCtrlLin_1_nNOut_v |
| 36131 | L_PosCtrlLin_2_nNOut_v |
| 36132 | L_PhiIntegrator_nSpeedGear_v |
| 36133 | L_DFSet_nSetGain_v |
| 36134 | L_DFSet_nSetGearTrim_v |
| 36135 | L_DFSet_nSetGearTrim_a |
| 36136 | L_DFRFG_nOut_v |
| 36137 | L_ConvPA_1_nOut |
| 36138 | L_ConvPA_2_nOut |
| 36139 | L_ConvPA_3_nOut |
| 36140 | L_ConvX_1_nOut |
| 36141 | L_ConvX_2_nOut |
| 36142 | L_ConvX_3_nOut |
| 36143 | MCK_wGearNum |
| 36144 | MCK_wGearDenom |
| 36145 | L_DFSet_nSpeedSetOut_v |
| 36146 | L_Curve_2_nOut_a |
| 36147 | L_Curve_3_nOut_a |

| Selection list - analog signals | |
|---|---|
| 42000 | LA_nCtrl_In_wCANDriveControl |
| 42001 | LA_nCtrl_In_wCCMDriveControl |
| 42002 | LA_nCtrl_In_nTorqueMotLim |
| 42003 | LA_nCtrl_In_nTorqueGenLim |
| 42004 | LA_nCtrl_In_nPIDVpAdapt_a |
| 42005 | LA_nCtrl_In_nPIDActValue_a |
| 42006 | LA_nCtrl_In_nMainSetValue |
| 42007 | LA_nCtrl_In_nAuxSetValue |
| 42008 | LA_nCtrl_In_nGPAnalogSwitchIn1_a |
| 42009 | LA_nCtrl_In_nGPAnalogSwitchIn2_a |
| 42010 | LA_nCtrl_In_nGPArithmetikIn1_a |
| 42011 | LA_nCtrl_In_nGPArithmetikIn2_a |
| 42012 | LA_nCtrl_In_nGPMulDivIn_a |
| 42013 | LA_nCtrl_In_nGPCompareIn1_a |
| 42014 | LA_nCtrl_In_nGPCompareIn2_a |
| 42015 | LA_nCtrl_In_nVoltageAdd_a |
| 42016 | LA_nCtrl_In_nPIDInfluence_a |
| 42017 | LA_nCtrl_In_nPIDSetValue_a |
| 42018 | LA_nCtrl_In_nPWMAngleOffset |
| 42019 | LA_nCtrl_In_nBoost_a |
| 42020 | LA_NCtrl_In_wSMControl |
| 42025 | LA_nCtrl_In_wFreeIn1 |
| 42026 | LA_nCtrl_In_wFreeIn2 |
| 42027 | LA_nCtrl_In_wFreeIn3 |
| 42028 | LA_nCtrl_In_wFreeIn4 |
| 42100 | LA_TabPos_In_wCanDriveControl |
| 42101 | LA_TabPos_In_wMciDriveControl |
| 42102 | LA_TabPos_In_nTorqueMotLim_a |
| 42103 | LA_TabPos_In_nTorqueGenLim_a |
| 42104 | LA_TabPos_In_nMainSetValue_a |
| 42105 | LA_TabPos_In_nAuxSetValue_a |
| 42106 | LA_TabPos_In_wMckCtrl1 |
| 42107 | LA_TabPos_In_wMckCtrl2 |
| 42108 | LA_TabPos_In_wMckOperationMode |
| 42109 | LA_TabPos_In_wPosProfileMode |
| 42110 | LA_TabPos_In_wPosProfileNo |
| 42111 | LA_TabPos_In_nGPAnalogSwitchIn1_a |
| 42112 | LA_TabPos_In_nGPAnalogSwitchIn2_a |
| 42113 | LA_TabPos_In_nGPArithmetikIn1_a |
| 42114 | LA_TabPos_In_nGPArithmetikIn2_a |
| 42115 | LA_TabPos_In_nGPMulDivIn_a |
| 42116 | LA_TabPos_In_nGPCompareIn1_a |
| 42117 | LA_TabPos_In_nGPCompareIn2_a |
| 42118 | LA_TabPos_In_wGPCounter1LdVal |
| 42119 | LA_TabPos_In_wGPCounter1CmpVal |
| 42120 | LA_TabPos_In_nSpeedOverride_a |
| 42121 | LA_TabPos_In_nAccOverride_a |
| 42122 | LA_TabPos_In_wFreeIn1 |
| 42123 | LA_TabPos_In_wFreeIn2 |
| 42124 | LA_TabPos_In_wFreeIn3 |
| 42125 | LA_TabPos_In_wFreeIn4 |
| 42126 | LA_TabPos_In_wSMControl |

| Selection list - analog signals | |
|---|---|
| 42127 | LA_TabPos_In_wPosProfileUnitsLW |
| 42128 | LA_TabPos_In_wPosProfileUnitsHW |
| 42200 | LA_SwitchPos_In_wCANDriveControl |
| 42201 | LA_SwitchPos_In_wMCIDriveControl |
| 42202 | LA_SwitchPos_In_nVoltageAdd_a |
| 42203 | LA_SwitchPos_In_nBoost_a |
| 42204 | LA_SwitchPos_In_nPWMAngleOffset |
| 42205 | LA_SwitchPos_In_nTorqueMotLim_a |
| 42206 | LA_SwitchPos_In_nTorqueGenLim_a |
| 42207 | LA_SwitchPos_In_nMainSetValue_a |
| 42208 | LA_SwitchPos_In_nAuxSetValue_a |
| 42209 | LA_SwitchPos_In_nGPAnalogSwitchIn1_a |
| 42210 | LA_SwitchPos_In_nGPAnalogSwitchIn2_a |
| 42211 | LA_SwitchPos_In_nGPArithmetikIn1_a |
| 42212 | LA_SwitchPos_In_nGPArithmetikIn2_a |
| 42213 | LA_SwitchPos_In_nGPMulDivIn_a |
| 42214 | LA_SwitchPos_In_nGPCompareIn1_a |
| 42215 | LA_SwitchPos_In_nGPCompareIn2_a |
| 42216 | LA_SwitchPos_In_wSMControl |
| 42221 | LA_SwitchPos_In_wFreeIn1 |
| 42222 | LA_SwitchPos_In_wFreeIn2 |
| 42223 | LA_SwitchPos_In_wFreeIn3 |
| 42224 | LA_SwitchPos_In_wFreeIn4 |

### 16.2.1.2 Selection list - digital signals

This selection list is relevant for the following parameters:

| Parameter | |
|---|---|
| C00411 | L_SignalMonitor_b: Signal sources |
| C00621 | System connection list: Bool |
| C00701 | LA_NCtrl: Digital connection list |
| C00711 | LA_TabPos: Digital connection list |
| C00761 | LA_SwitchPos: Digital connection list |

| Selection list - digital signals | |
|---|---|
| 0 | Not connected |
| 1000 | LA_nCtrl_bDriveReady |
| 1001 | LA_nCtrl_bDriveFail |
| 1002 | LA_nCtrl_bCInhActive |
| 1003 | LA_nCtrl_bQSPIsActive |
| 1004 | LA_nCtrl_bSpeedCcw |
| 1005 | LA_nCtrl_bSpeedActCompare |
| 1008 | LA_nCtrl_bGPDigitalDelayOut |
| 1009 | LA_nCtrl_bGPLogicOut |
| 1010 | LA_nCtrl_bGPSignalOut1 |
| 1011 | LA_nCtrl_bGPSignalOut2 |
| 1012 | LA_nCtrl_bGPSignalOut3 |
| 1013 | LA_nCtrl_bGPSignalOut4 |
| 1014 | LA_nCtrl_bOverLoadActive |
| 1015 | LA_nCtrl_bBrakeReleaseOut |
| 1016 | LA_nCtrl_bBrakeReleased |
| 1017 | LA_nCtrl_bGPCompareOut |
| 1018 | LA_nCtrl_bUnderLoadActive |
| 1019 | LA_nCtrl_bImaxActive |
| 1020 | LA_nCtrl_bSpeedSetReached |
| 1021 | LA_nCtrl_bSpeedActEqSet |
| 1022 | LA_nCtrl_bGPDFlipFlopOut |
| 1023 | LA_nCtrl_bGPDFlipFlopNegOut |
| 1029 | LA_nCtrl_bFreeOut1 |
| 1030 | LA_nCtrl_bFreeOut2 |
| 1031 | LA_nCtrl_bFreeOut3 |
| 1032 | LA_nCtrl_bFreeOut4 |
| 1033 | LA_nCtrl_bFreeOut5 |
| 1034 | LA_nCtrl_bFreeOut6 |
| 1035 | LA_nCtrl_bFreeOut7 |
| 1036 | LA_nCtrl_bFreeOut8 |
| 1100 | LA_TabPos_bDriveFail |
| 1101 | LA_TabPos_bDriveReady |
| 1102 | LA_TabPos_bCInhActive |
| 1103 | LA_TabPos_bQSPIsActive |
| 1104 | LA_TabPos_bSpeedCcw |
| 1105 | LA_TabPos_bSpeedActCompare |
| 1106 | LA_TabPos_bImaxActive |
| 1107 | LA_TabPos_bSpeedSetReached |
| 1108 | LA_TabPos_bMBrakeReleaseOut |
| 1109 | LA_TabPos_bMBrakeReleased |
| 1110 | LA_TabPos_bHomeDone |
| 1111 | LA_TabPos_bHomePosAvailable |
| 1112 | LA_TabPos_bProfileDone |
| 1113 | LA_TabPos_bProfileBusy |
| 1114 | LA_TabPos_bAccelerating |
| 1115 | LA_TabPos_bConstantDuty |
| 1116 | LA_TabPos_bDecelerating |
| 1117 | LA_TabPos_bDwellTime |
| 1118 | LA_TabPos_bInTarget |
| 1119 | LA_TabPos_bGPDigitalDelayOut |
| 1120 | LA_TabPos_bGPLogicOut |
| 1121 | LA_TabPos_bGPCompareOut |
| 1122 | LA_TabPos_bGPSignalOut1 |
| 1123 | LA_TabPos_bGPSignalOut2 |
| 1124 | LA_TabPos_bGPSignalOut3 |
| 1125 | LA_TabPos_bGPSignalOut4 |
| 1126 | LA_TabPos_bGPDFlipFlop_Out |
| 1127 | LA_TabPos_bGPDFlipFlop_NegOut |
| 1128 | LA_TabPos_bGPCounter1Equal |
| 1129 | LA_TabPos_bFreeOut1 |
| 1130 | LA_TabPos_bFreeOut2 |
| 1131 | LA_TabPos_bFreeOut3 |
| 1132 | LA_TabPos_bFreeOut4 |
| 1133 | LA_TabPos_bFreeOut5 |
| 1134 | LA_TabPos_bFreeOut6 |
| 1135 | LA_TabPos_bFreeOut7 |
| 1136 | LA_TabPos_bFreeOut8 |
| 1200 | LA_SwitchPos_bDriveFail |
| 1201 | LA_SwitchPos_bWarningActive |
| 1202 | LA_SwitchPos_bSafeTorqueOff |
| 1203 | LA_SwitchPos_bDriveReady |
| 1204 | LA_SwitchPos_bCInhActive |
| 1205 | LA_SwitchPos_bImplsActive |
| 1206 | LA_SwitchPos_bQSPIsActive |
| 1207 | LA_SwitchPos_bSpeedCcw |
| 1208 | LA_SwitchPos_bSpeedActCompare |
| 1209 | LA_SwitchPos_bImaxActive |
| 1210 | LA_SwitchPos_bSpeedSetReached |
| 1211 | LA_SwitchPos_bSpeedActEqSet |
| 1212 | LA_SwitchPos_bBrakeReleaseOut |
| 1213 | LA_SwitchPos_bBrakeReleased |
| 1214 | LA_SwitchPos_bGPDigitalDelayOut |
| 1215 | LA_SwitchPos_bGPLogicOut |
| 1216 | LA_SwitchPos_bGPCompareOut |
| 1217 | LA_SwitchPos_bGPDFlipFlop_Out |
| 1218 | LA_SwitchPos_bGPDFlipFlop_NegOut |
| 1219 | LA_SwitchPos_bGPSignalOut1 |
| 1220 | LA_SwitchPos_bGPSignalOut2 |
| 1221 | LA_SwitchPos_bGPSignalOut3 |

| Selection list - digital signals | |
|---|---|
| 1222 | LA_SwitchPos_bGPSignalOut4 |
| 1228 | LA_SwitchPos_bFreeOut1 |
| 1229 | LA_SwitchPos_bFreeOut2 |
| 1230 | LA_SwitchPos_bFreeOut3 |
| 1231 | LA_SwitchPos_bFreeOut4 |
| 1232 | LA_SwitchPos_bFreeOut5 |
| 1233 | LA_SwitchPos_bFreeOut6 |
| 1234 | LA_SwitchPos_bFreeOut7 |
| 1235 | LA_SwitchPos_bFreeOut8 |
| 16000 | DigIn_bIn1 |
| 16001 | DigIn_bIn2 |
| 16002 | DigIn_bIn3 |
| 16003 | DigIn_bIn4 |
| 16004 | DigIn_bIn5 |
| 16005 | DigIn_bIn6 |
| 16006 | DigIn_bIn7 |
| 16008 | DigIn_CInh |
| 16009 | DigIn_bCountIn1_Compare |
| 16010 | DigIn_bCountIn6_Compare |
| 16011 | Ain_bCurrentErrorIn1 |
| 16012 | Ain_bCurrentErrorIn2 |
| 16013 | CAN1_bCtrl1_B0 |
| 16014 | CAN1_bCtrl1_B1 |
| 16015 | CAN1_bCtrl1_B2 |
| 16016 | CAN1_bCtrl1_B3 |
| 16017 | CAN1_bCtrl1_B4 |
| 16018 | CAN1_bCtrl1_B5 |
| 16019 | CAN1_bCtrl1_B6 |
| 16020 | CAN1_bCtrl1_B7 |
| 16021 | CAN1_bCtrl1_B8 |
| 16022 | CAN1_bCtrl1_B9 |
| 16023 | CAN1_bCtrl1_B10 |
| 16024 | CAN1_bCtrl1_B11 |
| 16025 | CAN1_bCtrl1_B12 |
| 16026 | CAN1_bCtrl1_B13 |
| 16027 | CAN1_bCtrl1_B14 |
| 16028 | CAN1_bCtrl1_B15 |
| 16029 | CAN2_bIn1_B0 |
| 16030 | CAN2_bIn1_B1 |
| 16031 | CAN2_bIn1_B2 |
| 16032 | CAN2_bIn1_B3 |
| 16033 | CAN2_bIn1_B4 |
| 16034 | CAN2_bIn1_B5 |
| 16035 | CAN2_bIn1_B6 |
| 16036 | CAN2_bIn1_B7 |
| 16037 | CAN2_bIn1_B8 |
| 16038 | CAN2_bIn1_B9 |
| 16039 | CAN2_bIn1_B10 |
| 16040 | CAN2_bIn1_B11 |
| 16041 | CAN2_bIn1_B12 |
| 16042 | CAN2_bIn1_B13 |
| 16043 | CAN2_bIn1_B14 |

| Selection list - digital signals | |
|---|---|
| 16044 | CAN2_bIn1_B15 |
| 16045 | CAN3_bIn1_B0 |
| 16046 | CAN3_bIn1_B1 |
| 16047 | CAN3_bIn1_B2 |
| 16048 | CAN3_bIn1_B3 |
| 16049 | CAN3_bIn1_B4 |
| 16050 | CAN3_bIn1_B5 |
| 16051 | CAN3_bIn1_B6 |
| 16052 | CAN3_bIn1_B7 |
| 16053 | CAN3_bIn1_B8 |
| 16054 | CAN3_bIn1_B9 |
| 16055 | CAN3_bIn1_B10 |
| 16056 | CAN3_bIn1_B11 |
| 16057 | CAN3_bIn1_B12 |
| 16058 | CAN3_bIn1_B13 |
| 16059 | CAN3_bIn1_B14 |
| 16060 | CAN3_bIn1_B15 |
| 16061 | MciIn_bCtrl_B0 |
| 16062 | MciIn_bCtrl_B1 |
| 16063 | MciIn_bCtrl_B2 |
| 16064 | MciIn_bCtrl_B3 |
| 16065 | MciIn_bCtrl_B4 |
| 16066 | MciIn_bCtrl_B5 |
| 16067 | MciIn_bCtrl_B6 |
| 16068 | MciIn_bCtrl_B7 |
| 16069 | MciIn_bCtrl_B8 |
| 16070 | MciIn_bCtrl_B9 |
| 16071 | MciIn_bCtrl_B10 |
| 16072 | MciIn_bCtrl_B11 |
| 16073 | MciIn_bCtrl_B12 |
| 16074 | MciIn_bCtrl_B13 |
| 16075 | MciIn_bCtrl_B14 |
| 16076 | MciIn_bCtrl_B15 |
| 16077 | MciIn_bIn2_B0 |
| 16078 | MciIn_bIn2_B1 |
| 16079 | MciIn_bIn2_B2 |
| 16080 | MciIn_bIn2_B3 |
| 16081 | MciIn_bIn2_B4 |
| 16082 | MciIn_bIn2_B5 |
| 16083 | MciIn_bIn2_B6 |
| 16084 | MciIn_bIn2_B7 |
| 16085 | MciIn_bIn2_B8 |
| 16086 | MciIn_bIn2_B9 |
| 16087 | MciIn_bIn2_B10 |
| 16088 | MciIn_bIn2_B11 |
| 16089 | MciIn_bIn2_B12 |
| 16090 | MciIn_bIn2_B13 |
| 16091 | MciIn_bIn2_B14 |
| 16092 | MciIn_bIn2_B15 |
| 16093 | LS_Keypad_bSetQuickstop |
| 16094 | LS_Keypad_bSetDCBrake |
| 16095 | LS_Keypad_bSetSpeedCcw |

| Selection list - digital signals | |
|---|---|
| 16096 | LS_Keypad_bJogSpeed1 |
| 16097 | LS_Keypad_bJogSpeed2 |
| 16098 | LS_Keypad_bMPotEnable |
| 16099 | LS_Keypad_bMPotUp |
| 16100 | LS_Keypad_bMPotDown |
| 16101 | DigIn_bPosIn12_State |
| 16110 | LS_ParReadWrite_1_bDone |
| 16111 | LS_ParReadWrite_1_bFail |
| 16112 | LS_ParReadWrite_2_bDone |
| 16113 | LS_ParReadWrite_2_bFail |
| 16114 | LS_ParReadWrite_3_bDone |
| 16115 | LS_ParReadWrite_3_bFail |
| 16116 | LS_ParReadWrite_4_bDone |
| 16117 | LS_ParReadWrite_4_bFail |
| 16118 | LS_ParReadWrite_5_bDone |
| 16119 | LS_ParReadWrite_5_bFail |
| 16120 | LS_ParReadWrite_6_bDone |
| 16121 | LS_ParReadWrite_6_bFail |
| 16122 | LS_WriteParamList_bDone |
| 16123 | LS_WriteParamList_bFail |
| 16161 | LS_CANManagement_bFail |
| 16162 | LS_CANManagement_bOperational |
| 16165 | LS_MultiEncoder_bFail |
| 16166 | LS_Resolver_bFail |
| 16200 | LS_SyncManagement_bSyncSignalOK |
| 16201 | LS_SyncManagement_bSyncPhaseOK |
| 16202 | LS_MultiEncoder_bState |
| 16303 | LS_TouchProbe_bTP_DigIn3_Received |
| 16304 | LS_TouchProbe_bTP_DigIn4_Received |
| 16305 | LS_TouchProbe_bTP_DigIn5_Received |
| 16306 | LS_TouchProbe_bTP_DigIn6_Received |
| 16307 | LS_TouchProbe_bTP_DigIn7_Received |
| 16308 | LS_AxisBusIO_bFail_DigIn |
| 16309 | LS_AxisBusIO_bResetFail_In |
| 16310 | LS_RetainData_bOut1 |
| 16311 | LS_RetainData_bOut2 |
| 16312 | LS_RetainData_bOut3 |
| 16313 | LS_RetainData_bOut4 |
| 16320 | LS_AxisBusAux_bDone |
| 16321 | LS_AxisBusAux_bFail |
| 20000 | LS_ParFix_True |
| 20001 | bC470_1 |
| 20002 | bC470_2 |
| 20003 | bC470_3 |
| 20004 | bC470_4 |
| 20005 | bC470_5 |
| 20006 | bC470_6 |
| 20007 | bC470_7 |
| 20008 | bC470_8 |
| 20009 | bC470_9 |
| 20010 | bC470_10 |
| 20011 | bC470_11 |

| Selection list - digital signals | |
|---|---|
| 20012 | bC470_12 |
| 20013 | bC470_13 |
| 20014 | bC470_14 |
| 20015 | bC470_15 |
| 20016 | bC470_16 |
| 20017 | bC470_17 |
| 20018 | bC470_18 |
| 20019 | bC470_19 |
| 20020 | bC470_20 |
| 20021 | bC470_21 |
| 20022 | bC470_22 |
| 20023 | bC470_23 |
| 20024 | bC470_24 |
| 20025 | bC470_25 |
| 20026 | bC470_26 |
| 20027 | bC470_27 |
| 20028 | bC470_28 |
| 20029 | bC470_29 |
| 20030 | bC470_30 |
| 20031 | bC470_31 |
| 20032 | bC470_32 |
| 20033 | b100Hz |
| 20034 | b10Hz |
| 20035 | b2Hz |
| 20036 | b1Hz |
| 20037 | b1HzFlash |
| 20038 | b2HzFlash |
| 20039 | bSingleFlash1 |
| 20040 | bSingleFlash2 |
| 20041 | bDoubleFlash |
| 20042 | bSquareWave |
| 20043 | bFirstCycle |
| 20044 | LS_ParFix2_True |
| 20045 | LS_ParFreeUnit_bDataValid |
| 20046 | LS_ParFreeUnit2_bDataValid |
| 32000 | LS_DeviceMonitor_MCTRL_bFanFault |
| 32001 | LS_DeviceMonitor_MCTRL_bHeatSinkTemp |
| 32002 | MCTRL_bLimPosCtrlOut |
| 32003 | MCTRL_bLimSpeedCtrlOut |
| 32004 | MCTRL_bLimSpeedSetVal |
| 32005 | MCTRL_bLimTorqueSetVal |
| 32006 | MCTRL_bLimCurrentSetVal |
| 32007 | LS_DeviceMonitor_MCTRL_bUVDetected |
| 32008 | LS_DeviceMonitor_MCTRL_bOVDetected |
| 32009 | LS_DeviceMonitor_MCTRL_bMotorPhaseFault |
| 32010 | LS_DeviceMonitor_MCTRL_bEncoderComFault |
| 32011 | LS_DeviceMonitor_MCTRL_bIxtOverload |
| 32012 | LS_DeviceMonitor_MCTRL_bI2xtOverload |
| 32013 | MCTRL_bIdentificationActive |
| 32014 | MCTRL_bFlyingSyncActive |
| 32015 | LS_DeviceMonitor_MCTRL_bTorqueMax |
| 32016 | LS_DeviceMonitor_MCTRL_bNMax |

| Selection list - digital signals | |
|---|---|
| 32017 | LS_DeviceMonitor_MCTRL_bFChopReduced |
| 32018 | LS_DeviceMonitor_MCTRL_bMotorPTC |
| 32019 | LS_DeviceMonitor_MCTRL_bMotorTemp |
| 32020 | MCTRL_bAutoGSBIsActive |
| 32021 | LS_DeviceMonitor_MCTRL_bBrakeChopper |
| 32022 | MCTRL_bQspIsActive |
| 32023 | MCTRL_bHlgLoad |
| 32024 | MCTRL_bHlgStop |
| 32025 | LS_DeviceMonitor_MCTRL_bImpActive |
| 32026 | LS_DeviceMonitor_MCTRL_bClampActive |
| 32027 | LS_DeviceMonitor_MCTRL_bMainsFault |
| 32028 | LS_DeviceMonitor_MCTRL_bNmaxForFChop |
| 32029 | LS_DeviceMonitor_MCTRL_bShortCircuit |
| 32030 | LS_DeviceMonitor_MCTRL_bEarthFault |
| 32100 | DCTRL_bInit |
| 32101 | DCTRL_bReady |
| 32102 | DCTRL_bReadyToSwitchOn |
| 32103 | DCTRL_bOperationEnable |
| 32104 | DCTRL_bWarning |
| 32105 | DCTRL_bTrouble |
| 32106 | DCTRL_bFail |
| 32107 | DCTRL_bCollectedFail |
| 32108 | DCTRL_bSafeTorqueOff |
| 32109 | DCTRL_bIMPIsActive |
| 32110 | DCTRL_bCINHIsActive |
| 32111 | DCTRL_bSafetyIsActive |
| 32112 | DCTRL_bCwCcw |
| 32113 | DCTRL_bNactCompare |
| 32200 | MCK_bPosCtrlOn |
| 32201 | MCK_bSpeedCtrlIOn |
| 32202 | MCK_bTorquemodeOn |
| 32203 | MCK_bDcBrakeOn |
| 32204 | MCK_bBrkReleaseOut |
| 32205 | MCK_bBrkReleased |
| 32206 | MCK_bDeltaPosOn |
| 32207 | MCK_bPosDerivativeOn |
| 32208 | MCK_bMotorRefOffsetOn |
| 32209 | MCK_bQspOn |
| 32210 | MCK_bPosBusy |
| 32211 | MCK_bPosDone |
| 32212 | MCK_bHomDone |
| 32213 | MCK_bHomAvailable |
| 32214 | MCK_bTorqueLimitAdaptOn |
| 32215 | Reserved |
| 32216 | Reserved |
| 32217 | Reserved |
| 32218 | Reserved |
| 36000 | L_AND_Out_1 |
| 36001 | L_AND_Out_2 |
| 36002 | L_AND_Out_3 |
| 36003 | L_OR_Out_1 |
| 36004 | L_OR_Out_2 |

| Selection list - digital signals | |
|---|---|
| 36005 | L_OR_Out_3 |
| 36006 | L_NOT_Out_1 |
| 36007 | L_NOT_Out_2 |
| 36008 | L_NOT_Out_3 |
| 36009 | L_DFlipFlop_Out_1 |
| 36010 | L_RLQ_1_Qsp |
| 36011 | L_RLQ_1_Ccw |
| 36012 | L_DigDelay_Out_1 |
| 36013 | L_Compare_Out_1 |
| 36014 | L_Compare_Out_2 |
| 36015 | L_Compare_Out_3 |
| 36016 | L_Nset_RfgEq0_1 |
| 36017 | L_DigitalLogic_1_bOut |
| 36018 | L_Counter_1_bEqual |
| 36019 | L_SigMonitor_b_bOut1 |
| 36020 | L_SigMonitor_b_bOut2 |
| 36021 | L_SigMonitor_b_bOut3 |
| 36022 | L_SigMonitor_b_bOut4 |
| 36023 | L_PCTRL_1_bActEqSet |
| 36024 | L_NLim_1_bLimitActive |
| 36025 | L_DFlipFlop_1_NegOut |
| 36026 | L_AND5_1_Out |
| 36027 | L_AND5_2_Out |
| 36028 | L_Compare_4_bOut |
| 36029 | L_Compare_5_bOut |
| 36030 | L_Compare_6_bOut |
| 36031 | L_Compare_7_bOut |
| 36032 | L_Compare_8_bOut |
| 36033 | L_Compare_9_bOut |
| 36034 | L_Compare_10_bOut |
| 36035 | L_DFlipFlop_2_bOut |
| 36036 | L_DFlipFlop_2_bNegOut |
| 36037 | L_DigDelay_2_Out |
| 36038 | L_DigDelay_3_Out |
| 36039 | L_OR_Out_4 |
| 36040 | L_DigitalLogic_3_bOut |
| 36041 | L_DigitalLogic_2_bOut |
| 36042 | L_DigitalLogic5_1_bOut |
| 36043 | L_DigitalLogic5_2_bOut |
| 36044 | L_NLim_2_bLimitActive |
| 36045 | L_OR5_1_Out |
| 36046 | L_OR5_2_Out |
| 36047 | L_NOT_4_Out |
| 36048 | L_NOT_5_Out |
| 36049 | L_NOT_6_Out |
| 36050 | L_NOT_7_Out |
| 36055 | L_PhaseIntK_1_bState |
| 36056 | L_PhaseIntK_2_bState |
| 36057 | L_Counter_2_bEqual |
| 36058 | L_Counter_3_bEqual |
| 36059 | L_RSFlipflop_1_bOut |
| 36060 | L_RSFlipflop_1_bNegOut |

| Selection list - digital signals | |
|---|---|
| 36061 | L_RSFlipflop_2_bOut |
| 36062 | L_RSFlipflop_2_bNegOut |
| 36063 | L_Transient_5_bOut |
| 36064 | L_Transient_6_bOut |
| 36065 | L_Transient_7_bOut |
| 36066 | L_Transient_8_bOut |
| 36067 | L_ConvWordToBits_1_bBit0 |
| 36068 | L_ConvWordToBits_1_bBit1 |
| 36069 | L_ConvWordToBits_1_bBit2 |
| 36070 | L_ConvWordToBits_1_bBit3 |
| 36071 | L_ConvWordToBits_1_bBit4 |
| 36072 | L_ConvWordToBits_1_bBit5 |
| 36073 | L_ConvWordToBits_1_bBit6 |
| 36074 | L_ConvWordToBits_1_bBit7 |
| 36075 | L_ConvWordToBits_1_bBit8 |
| 36076 | L_ConvWordToBits_1_bBit9 |
| 36077 | L_ConvWordToBits_1_bBit10 |
| 36078 | L_ConvWordToBits_1_bBit11 |
| 36079 | L_ConvWordToBits_1_bBit12 |
| 36080 | L_ConvWordToBits_1_bBit13 |
| 36081 | L_ConvWordToBits_1_bBit14 |
| 36082 | L_ConvWordToBits_1_bBit15 |
| 36083 | L_ConvWordToBits_2_bBit0 |
| 36084 | L_ConvWordToBits_2_bBit1 |
| 36085 | L_ConvWordToBits_2_bBit2 |
| 36086 | L_ConvWordToBits_2_bBit3 |
| 36087 | L_ConvWordToBits_2_bBit4 |
| 36088 | L_ConvWordToBits_2_bBit5 |
| 36089 | L_ConvWordToBits_2_bBit6 |
| 36090 | L_ConvWordToBits_2_bBit7 |
| 36091 | L_ConvWordToBits_2_bBit8 |
| 36092 | L_ConvWordToBits_2_bBit9 |
| 36093 | L_ConvWordToBits_2_bBit10 |
| 36094 | L_ConvWordToBits_2_bBit11 |
| 36095 | L_ConvWordToBits_2_bBit12 |
| 36096 | L_ConvWordToBits_2_bBit13 |
| 36097 | L_ConvWordToBits_2_bBit14 |
| 36098 | L_ConvWordToBits_2_bBit15 |
| 36099 | L_ConvWordToBits_3_bBit0 |
| 36100 | L_ConvWordToBits_3_bBit1 |
| 36101 | L_ConvWordToBits_3_bBit2 |
| 36102 | L_ConvWordToBits_3_bBit3 |
| 36103 | L_ConvWordToBits_3_bBit4 |
| 36104 | L_ConvWordToBits_3_bBit5 |
| 36105 | L_ConvWordToBits_3_bBit6 |
| 36106 | L_ConvWordToBits_3_bBit7 |
| 36107 | L_ConvWordToBits_3_bBit8 |
| 36108 | L_ConvWordToBits_3_bBit9 |
| 36109 | L_ConvWordToBits_3_bBit10 |
| 36110 | L_ConvWordToBits_3_bBit11 |
| 36111 | L_ConvWordToBits_3_bBit12 |
| 36112 | L_ConvWordToBits_3_bBit13 |

| Selection list - digital signals | |
|---|---|
| 36113 | L_ConvWordToBits_3_bBit14 |
| 36114 | L_ConvWordToBits_3_bBit15 |
| 36115 | L_MckCtrlInterface_1_bFail |
| 36116 | L_MckStateInterface_1_bAccelerating |
| 36117 | L_MckStateInterface_1_bDecelerating |
| 36118 | L_MckStateInterface_1_bConstantSpeedDuty |
| 36119 | L_MckStateInterface_1_bSShaping |
| 36120 | L_MckStateInterface_1_bBusy |
| 36121 | L_MckStateInterface_1_bDone |
| 36122 | L_MckStateInterface_1_bHomingPosAvailable |
| 36123 | L_MckStateInterface_1_bHomingDone |
| 36124 | L_MckStateInterface_1_bDwellTime |
| 36125 | L_MckStateInterface_1_bInTarget |
| 36126 | L_MckStateInterface_1_bPosDone |
| 36127 | L_PosiShaftCtrlInterface_1_bQsp |
| 36128 | L_PosiShaftCtrlInterface_1_bRsp |
| 36129 | L_PosiShaftCtrlInterface_1_bTripSet |
| 36130 | L_PosiShaftCtrlInterface_1_bTripReset |
| 36131 | L_JogCtrlExtension_bRfgOut |
| 36132 | L_JogCtrlExtension_bJog1Out |
| 36133 | L_JogCtrlExtension_bJog2Out |
| 36134 | L_Curve_1_bLimit |
| 36135 | L_Interpolator_1_bIPulse |
| 36136 | L_Interpolator_1_bSignalError |
| 36137 | L_MckCtrlInterface_1_bPosSetDataValid |
| 36138 | L_Transient_1_bOut |
| 36139 | L_Transient_2_bOut |
| 36140 | L_Transient_3_bOut |
| 36141 | L_Transient_4_bOut |
| 36142 | L_CalcDiameter_bDMaxLimit |
| 36143 | L_CalcDiameter_bDMinLimit |
| 36144 | L_CalcDiameter_bWebBreak |
| 36145 | L_PhiIntegrator_bOvfl32 |
| 36146 | L_PhiIntegrator_bOvfl16 |
| 36147 | L_PhiIntegrator_bInitDone |
| 36148 | L_PosCtrlLin_1_bInTarget |
| 36149 | L_PosCtrlLin_2_bInTarget |
| 36150 | L_SwitchPoint_bOut1 |
| 36151 | L_SwitchPoint_bOut2 |
| 36152 | L_SwitchPoint_bOut3 |
| 36153 | L_SwitchPoint_bOut4 |
| 36154 | L_DFSet_bAck_b |
| 36155 | L_DFSet_bFollowingErr_b |
| 36156 | L_DFSet_bPosOverflow_b |
| 36157 | L_DFRFG_bSync_b |
| 36158 | L_DFRFG_bFail_b |
| 36159 | L_Curve_2_bLimit |
| 36160 | L_Curve_3_bLimit |
| 42000 | LA_NCtrl_In_bCInh |
| 42001 | LA_NCtrl_In_bFailReset |
| 42002 | LA_NCtrl_In_bSetQuickstop |
| 42003 | LA_NCtrl_In_bSetDCBrake |

| Selection list - digital signals | |
|---|---|
| 42004 | LA_NCtrl_In_bRFG_Stop |
| 42005 | LA_NCtrl_In_bRFG_0 |
| 42007 | LA_NCtrl_In_bSetSpeedCcw |
| 42008 | LA_NCtrl_bJogSpeed1 |
| 42009 | LA_NCtrl_bJogSpeed2 |
| 42010 | LA_NCtrl_bJogSpeed4 |
| 42011 | LA_NCtrl_bJogSpeed8 |
| 42012 | LA_NCtrl_bJogRamp1 |
| 42013 | LA_NCtrl_bJogRamp2 |
| 42014 | LA_NCtrl_bJogRamp4 |
| 42015 | LA_NCtrl_bJogRamp8 |
| 42017 | LA_NCtrl_bMPOTInAct |
| 42018 | LA_NCtrl_bMPOTUp |
| 42019 | LA_NCtrl_bMPOTDown |
| 42020 | LA_NCtrl_bMBRKRelease |
| 42021 | LA_NCtrl_bMANJogPos |
| 42022 | LA_NCtrl_bMANJogNeg |
| 42023 | LA_NCtrl_bGPAnalogSwitchSet |
| 42024 | LA_NCtrl_bGPDigitalDelayIn |
| 42025 | LA_NCtrl_bGPLogicIn1 |
| 42026 | LA_NCtrl_bGPLogicIn2 |
| 42027 | LA_NCtrl_bGPLogicIn3 |
| 42028 | LA_NCtrl_bGPDFlipFlopInD |
| 42029 | LA_NCtrl_bGPDFlipFlopInClk |
| 42030 | LA_NCtrl_bGPDFlipFlopInClr |
| 42031 | LA_NCtrl_bMPotEnable |
| 42032 | LA_NCtrl_bPIDEnableInfluenceRamp |
| 42033 | LA_NCtrl_bPIDIOff |
| 42034 | LA_NCtrl_bRLQCw |
| 42035 | LA_NCtrl_bRLQCcw |
| 42041 | LA_NCtrl_bFreeIn1 |
| 42042 | LA_NCtrl_bFreeIn2 |
| 42043 | LA_NCtrl_bFreeIn3 |
| 42044 | LA_NCtrl_bFreeIn4 |
| 42045 | LA_NCtrl_bFreeIn5 |
| 42046 | LA_NCtrl_bFreeIn6 |
| 42047 | LA_NCtrl_bFreeIn7 |
| 42048 | LA_NCtrl_bFreeIn8 |
| 42100 | LA_TabPos_In_bCInh |
| 42101 | LA_TabPos_In_bFailReset |
| 42102 | LA_TabPos_In_bSetQuickstop |
| 42103 | LA_TabPos_In_bSetSpeedCcw |
| 42104 | LA_TabPos_In_bJogSpeed1 |
| 42105 | LA_TabPos_In_bJogSpeed2 |
| 42106 | LA_TabPos_In_bMPotEnable |
| 42107 | LA_TabPos_In_bMPotUp |
| 42108 | LA_TabPos_In_bMPotDown |
| 42109 | LA_TabPos_In_bMBrakeRelease |
| 42110 | LA_TabPos_In_bPosCtrlOn |
| 42111 | LA_TabPos_In_bLimitSwitchPos |
| 42112 | LA_TabPos_In_bLimitSwitchNeg |
| 42113 | LA_TabPos_In_bReleaseLimitSwitch |

| Selection list - digital signals | |
|---|---|
| 42114 | LA_TabPos_In_bManJogPos |
| 42115 | LA_TabPos_In_bManJogNeg |
| 42116 | LA_TabPos_In_bManEnable2ndSpeed |
| 42117 | LA_TabPos_In_bEnableSpeedOverride |
| 42118 | LA_TabPos_In_bEnableAccOverride |
| 42119 | LA_TabPos_In_bHomeStartStop |
| 42120 | LA_TabPos_In_bHomeSetPosition |
| 42121 | LA_TabPos_In_bHomeResetPosition |
| 42122 | LA_TabPos_In_bHomeMark |
| 42123 | LA_TabPos_In_bPosSetProfilePosition |
| 42124 | LA_TabPos_In_bPosSetActualPosition |
| 42125 | LA_TabPos_In_bPosExecute |
| 42126 | LA_TabPos_In_bPosFinishTarget |
| 42127 | LA_TabPos_In_bPosDisableFollowProfile |
| 42128 | LA_TabPos_In_bPosStop |
| 42129 | LA_TabPos_In_bGPAnalogSwitchSet |
| 42130 | LA_TabPos_In_bGPDigitalDelayIn |
| 42131 | LA_TabPos_In_bGPLogicIn1 |
| 42132 | LA_TabPos_In_bGPLogicIn2 |
| 42133 | LA_TabPos_In_bGPLogicIn3 |
| 42134 | LA_TabPos_In_bGPDFlipFlop_InD |
| 42135 | LA_TabPos_In_bGPDFlipFlop_InClk |
| 42136 | LA_TabPos_In_bGPDFlipFlop_InClr |
| 42137 | LA_TabPos_In_bGPCounter1ClkUp |
| 42138 | LA_TabPos_In_bGPCounter1ClkDown |
| 42139 | LA_TabPos_In_bGPCounter1Load |
| 42140 | LA_TabPos_In_bMckOperationMode_1 |
| 42141 | LA_TabPos_In_bMckOperationMode_2 |
| 42142 | LA_TabPos_In_bMckOperationMode_4 |
| 42143 | LA_TabPos_In_bMckOperationMode_8 |
| 42144 | LA_TabPos_In_bPosProfileNo_1 |
| 42145 | LA_TabPos_In_bPosProfileNo_2 |
| 42146 | LA_TabPos_In_bPosProfileNo_4 |
| 42147 | LA_TabPos_In_bPosProfileNo_8 |
| 42148 | LA_TabPos_In_bFreeIn1 |
| 42149 | LA_TabPos_In_bFreeIn2 |
| 42150 | LA_TabPos_In_bFreeIn3 |
| 42151 | LA_TabPos_In_bFreeIn4 |
| 42152 | LA_TabPos_In_bFreeIn5 |
| 42153 | LA_TabPos_In_bFreeIn6 |
| 42154 | LA_TabPos_In_bFreeIn7 |
| 42155 | LA_TabPos_In_bFreeIn8 |
| 42200 | LA_SwitchPos_bCInh |
| 42201 | LA_SwitchPos_bFailReset |
| 42202 | LA_SwitchPos_bSetQuickstop |
| 42203 | LA_SwitchPos_bSetDCBrake |
| 42204 | LA_SwitchPos_bRFG_Stop |
| 42205 | LA_SwitchPos_bSetSpeedCcw |
| 42206 | LA_SwitchPos_bRLQCw |
| 42207 | LA_SwitchPos_bRLQCcw |
| 42208 | LA_SwitchPos_bJogCtrlInputSel1 |
| 42209 | LA_SwitchPos_bJogCtrlInputSel2 |

| Selection list - digital signals | |
|---|---|
| 42210 | LA_SwitchPos_bJogCtrlRfgIn |
| 42211 | LA_SwitchPos_bJogCtrlJog1 |
| 42212 | LA_SwitchPos_bJogCtrlJog2 |
| 42213 | LA_SwitchPos_bJogCtrlSlowDown1 |
| 42214 | LA_SwitchPos_bJogCtrlStop1 |
| 42215 | LA_SwitchPos_bJogCtrlSlowDown2 |
| 42216 | LA_SwitchPos_bJogCtrlStop2 |
| 42217 | LA_SwitchPos_bJogCtrlSlowDown3 |
| 42218 | LA_SwitchPos_bJogCtrlStop3 |
| 42219 | LA_SwitchPos_bJogSpeed4 |
| 42220 | LA_SwitchPos_bJogSpeed8 |
| 42221 | LA_SwitchPos_bJogRamp1 |
| 42222 | LA_SwitchPos_bJogRamp2 |
| 42223 | LA_SwitchPos_bJogRamp4 |
| 42224 | LA_SwitchPos_bJogRamp8 |
| 42225 | LA_SwitchPos_bMBrkRelease |
| 42226 | LA_SwitchPos_bGPAnalogSwitchSet |
| 42227 | LA_SwitchPos_bGPDigitalDelayIn |
| 42228 | LA_SwitchPos_bGPLogicIn1 |
| 42229 | LA_SwitchPos_bGPLogicIn2 |
| 42230 | LA_SwitchPos_bGPLogicIn3 |
| 42231 | LA_SwitchPos_bGPDFlipFlop_InD |
| 42232 | LA_SwitchPos_bGPDFlipFlop_InClk |
| 42233 | LA_SwitchPos_bGPDFlipFlop_InClr |
| 42239 | LA_SwitchPos_bFreeIn1 |
| 42240 | LA_SwitchPos_bFreeIn2 |
| 42241 | LA_SwitchPos_bFreeIn3 |
| 42242 | LA_SwitchPos_bFreeIn4 |
| 42243 | LA_SwitchPos_bFreeIn5 |
| 42244 | LA_SwitchPos_bFreeIn6 |
| 42245 | LA_SwitchPos_bFreeIn7 |
| 42246 | LA_SwitchPos_bFreeIn8 |

### 16.2.1.3 Selection list - angle signals

This selection list is relevant for the following parameters:

| Parameter | |
|---|---|
| C00622 | System connection list: Angle |
| C00712 | LA_TabPos: phi connection list |

| Selection list - angle signals | |
|---|---|
| 0 | Not connected |
| 1005 | LA_NCtrl_dnFreeOut1_p |
| 1006 | LA_NCtrl_dnFreeOut1_p |
| 1100 | LA_TabPos_dnTargetPos_p |
| 1101 | LA_TabPos_dnSetPos_p |
| 1102 | LA_TabPos_dnFreeOut1_p |
| 1103 | LA_TabPos_dnFreeOut2_p |
| 1104 | LA_TabPos_dnPosAct_p |
| 1105 | LA_TabPos_dnDeltaPosAct_p |
| 1205 | LA_SwitchPos_dnFreeOut1_p |
| 1206 | LA_SwitchPos_dnFreeOut2_p |
| 16000 | CAN1_dnIn34_p |
| 16001 | CAN2_dnIn34_p |
| 16002 | CAN3_dnIn34_p |
| 16003 | LP_MciIn_dnIn34_p |
| 16006 | CAN1_dnIn12_p |
| 16007 | CAN2_dnIn12_p |
| 16008 | CAN3_dnIn12_p |
| 16009 | LP_MciIn_dnIn56_p |
| 16010 | LP_MciIn_dnIn78_p |
| 17000 | LS_MultiEncoder_dnPosition_p |
| 17001 | LS_Resolver_dnPosition_p |
| 17010 | LS_BusEncoder_dnPosition_p |
| 17020 | DigIn_dnPosIn12_p |
| 17101 | LS_TouchProbe_TPDigIn1_dnPosition_p |
| 17102 | LS_TouchProbe_TPDigIn2_dnPosition_p |
| 17103 | LS_TouchProbe_TPDigIn3_dnPosition_p |
| 17104 | LS_TouchProbe_TPDigIn4_dnPosition_p |
| 17105 | LS_TouchProbe_TPDigIn5_dnPosition_p |
| 17106 | LS_TouchProbe_TPDigIn6_dnPosition_p |
| 17107 | LS_TouchProbe_TPDigIn7_dnPosition_p |
| 17120 | LS_RetainData_dnOut1 |
| 17121 | LS_RetainData_dnOut2 |
| 17122 | LS_RetainData_dnOut3 |
| 17123 | LS_RetainData_dnOut4 |
| 17130 | LS_AxisBusIn_dnLine12 |
| 17131 | LS_AxisBusIn_dnCas12 |
| 17132 | LS_AxisBusIn_dnCas34 |
| 17133 | LS_AxisBusAux_dnAuxIn12 |
| 17134 | LS_AxisBusAux_dnAuxIn34 |
| 20000 | dnC474_1_p |
| 20001 | dnC474_2_p |
| 20002 | dnC474_3_p |

| Selection list - angle signals | |
|---|---|
| 20003 | dnC474_4_p |
| 20004 | dnC474_5_p |
| 20005 | dnC474_6_p |
| 20006 | dnC474_7_p |
| 20007 | dnC474_8_p |
| 20008 | dnC479_1 |
| 20009 | dnC479_2 |
| 20010 | dnC479_3 |
| 20011 | dnC479_4 |
| 20012 | dnC479_5 |
| 20013 | dnC479_6 |
| 20014 | dnC479_7 |
| 20015 | dnC479_8 |
| 20016 | dnC475_1 |
| 20017 | dnC475_2 |
| 20018 | dnC475_3 |
| 20019 | dnC475_4 |
| 20020 | dnC475_5 |
| 20021 | dnC475_6 |
| 20022 | dnC475_7 |
| 20023 | dnC475_8 |
| 20024 | dnC475_9 |
| 20025 | dnC475_10 |
| 20026 | dnC475_11 |
| 20027 | dnC475_12 |
| 20028 | dnC475_13 |
| 20029 | dnC475_14 |
| 20030 | dnC475_15 |
| 20031 | dnC475_16 |
| 20032 | dnC475_17 |
| 20033 | dnC475_18 |
| 20034 | dnC475_19 |
| 20035 | dnC475_20 |
| 20036 | dnC475_21 |
| 20037 | dnC475_22 |
| 20038 | dnC475_23 |
| 20039 | dnC475_24 |
| 20040 | dnC475_25 |
| 20041 | dnC475_26 |
| 20042 | dnC475_27 |
| 20043 | dnC475_28 |
| 20044 | dnC475_29 |
| 20045 | dnC475_30 |
| 20046 | dnC475_31 |
| 20047 | dnC475_32 |
| 32000 | MCTRL_dnMotorPosAct_p |
| 32001 | MCTRL_dnMotorDeltaPosAct_p |
| 32200 | MCK_dnPosTarget_p |
| 32201 | MCK_dnPosSetValue_p |
| 32202 | MCK_dnDeltaPos_p |

| Selection list - angle signals | |
|---|---|
| 32203 | MCK_dnMotorRefOffset_p |
| 32204 | MCK_dnPosSet_p |
| 32205 | MCK_dnPosSetRelative_p |
| 36040 | L_ArithmetikPhi_1_Out_p |
| 36046 | L_GainOffsetParPhi_1_Out_p |
| 36047 | L_GainOffsetParPhi_2_Out_p |
| 36050 | L_LimitPhi_1_Out_p |
| 36051 | L_LimitPhi_2_Out_p |
| 36052 | L_LimitPhi_3_Out_p |
| 36060 | L_OffsetGainParPhi_1_Out_p |
| 36061 | L_OffsetGainParPhi_2_Out_p |
| 36066 | L_PhaseIntK_1_dnOut_p |
| 36067 | L_PhaseIntK_2_dnOut_p |
| 36070 | L_ArithmetikPhi_2_Out_p |
| 36071 | L_ArithmetikPhi_3_Out_p |
| 36072 | L_Sqrt_1_Out_p |
| 36073 | L_Mux_1_Out_p |
| 36074 | L_ConvWordsToDInt_1_dnOut_p |
| 36075 | L_ConvWordsToDInt_2_dnOut_p |
| 36076 | L_ConvWordsToDInt_3_dnOut_p |
| 36077 | L_ConvUnitsToIncr_1_dnOut_p |
| 36078 | L_ConvUnitsToIncr_2_dnOut_p |
| 36079 | L_ConvUnitsToIncr_3_dnOut_p |
| 36080 | L_Interpolator_1_dnPhiOut_p |
| 36081 | L_MckCtrlInterface_1_dnPosSetOut_p |
| 36082 | L_PhaseDiff_1_dnOut_p |
| 36083 | L_PhaseDiff_2_dnOut_p |
| 36086 | L_MckStateInterface_1_dnPosOut_p |
| 36087 | L_Odometer_1_dnPosOut_1_p |
| 36088 | L_Odometer_1_dnPosOut_2_p |
| 36089 | L_Odometer_1_dnPosOut_3_p |
| 36090 | L_Odometer_1_dnPosOut_4_p |
| 36091 | L_Odometer_1_dnPosOut_5_p |
| 36092 | L_Odometer_1_dnPosOut_6_p |
| 36093 | L_Odometer_1_dnPosOut_7_p |
| 36094 | L_Odometer_1_dnPosOut_8_p |
| 36095 | L_Odometer_1_dnDeltaPos_12_p |
| 36096 | L_Odometer_1_dnDeltaPos_23_p |
| 36097 | L_Odometer_1_dnDeltaPos_34_p |
| 36098 | L_Odometer_1_dnDeltaPos_45_p |
| 36099 | L_Odometer_1_dnDeltaPos_56_p |
| 36100 | L_Odometer_1_dnDeltaPos_67_p |
| 36101 | L_Odometer_1_dnDeltaPos_78_p |
| 36102 | L_Odometer_1_dnDeltaPos_18_p |
| 36103 | L_CalcDiameter_dnPos_p |
| 36104 | L_PhiIntegrator_dnOut32_p |
| 36105 | L_PosCtrlLin_1_dnPosOut_p |
| 36106 | L_PosCtrlLin_2_dnPosOut_p |
| 36107 | L_DFSet_dnPosDiffOut_p |
| 36108 | L_DFSet_dnPosSetOut_p |
| 36109 | L_GearComp_dnOut_p |
| 36110 | L_ConvAP_1_dnOut_p |

| Selection list - angle signals | |
|---|---|
| 36111 | L_ConvAP_2_dnOut_p |
| 36112 | L_ConvAP_3_dnOut_p |
| 36113 | L_ConvPP_1_dnOut_p |
| 36114 | L_ConvPP_2_dnOut_p |
| 36115 | L_ConvPP_3_dnOut_p |
| 36116 | L_SignalSwitch32_1_dnOut |
| 36117 | L_SignalSwitch32_2_dnOut |
| 36118 | L_SignalSwitch32_3_dnOut |
| 36119 | L_CalcDiameter_dwDiameter |
| 42005 | LA_NCtrl_dnFreeIn1_p |
| 42006 | LA_NCtrl_dnFreeIn2_p |
| 42100 | LA_TabPos_In_dnPosProfilePosition |
| 42101 | LA_TabPos_In_dnFreeIn1_p |
| 42102 | LA_TabPos_In_dnFreeIn2_p |
| 42205 | LA_SwitchPos_dnFreeIn1_p |
| 42206 | LA_SwitchPos_dnFreeIn2_p |

## 16.3 Table of attributes

The table of attributes contains information required for a communication with the controller via parameters.

### How to read the table of attributes:

| Column | | Meaning | Entry | |
|---|---|---|---|---|
| Code | | Parameter name | Cxxxxx | |
| Name | | Parameter short text (display text) | *Text* | |
| Index | dec | Index under which the parameter is addressed. The subindex for array variables corresponds to the Lenze subcode number. | 24575 - Lenze code number | Is only required for access via a bus system. |
| | hex | | $5FFF_h$ - Lenze code number | |
| Data | DS | Data structure | E | Single variable (only one parameter element) |
| | | | A | Array variable (several parameter elements) |
| | DA | Number of array elements (subcodes) | *Number* | |
| | DT | Data type | INTEGER_16 | 2 bytes with sign |
| | | | INTEGER_32 | 4 bytes with sign |
| | | | UNSIGNED_8 | 1 byte without sign |
| | | | UNSIGNED_16 | 2 bytes without sign |
| | | | UNSIGNED_32 | 4 bytes without sign |
| | | | VISIBLE_STRING [xx] | ASCII string (with character length xx) |
| | Factor | Factor for data transmission via a bus system, depending on the number of decimal positions | *Factor* | 1 ≡ no decimal positions<br>10 ≡ 1 decimal position<br>100 ≡ 2 decimal positions<br>1000 ≡ 3 decimal positions<br>10000 ≡ 4 decimal positions |
| Access | R | Read access | ☑ Reading permitted | |
| | W | Write access | ☑ Writing permitted | |
| | CINH | Controller inhibit required | ☑ Writing is only possible if the controller is inhibited | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00002 | Device commands | 24573 | 5FFD | A | 34 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00003 | Status of the last device command | 24572 | 5FFC | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C00005 | Application | 24570 | 5FFA | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00006 | Motor control | 24569 | 5FF9 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00007 | Control mode | 24568 | 5FF8 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00008 | Original application\|control source | 24567 | 5FF7 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00010 | AIN1: Characteristic | 24565 | 5FF5 | A | 8 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00011 | Appl.: Reference speed | 24564 | 5FF4 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00012 | Accel. time - main setpoint | 24563 | 5FF3 | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00013 | Decel. time - main setpoint | 24562 | 5FF2 | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00015 | VFC: V/f base frequency | 24560 | 5FF0 | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00016 | VFC: Vmin boost | 24559 | 5FEF | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00018 | Switching frequency | 24557 | 5FED | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00019 | Auto DCB: Threshold | 24556 | 5FEC | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00020 | AIN2: Characteristic | 24555 | 5FEB | A | 8 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00021 | Slip comp. | 24554 | 5FEA | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00022 | Imax in motor mode | 24553 | 5FE9 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00023 | Imax in generator mode | 24552 | 5FE8 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00024 | LS_DriveInterface: bNActCompare | 24551 | 5FE7 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00026 | AINx: Offset | 24549 | 5FE5 | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00027 | AINx: Gain | 24548 | 5FE4 | A | 2 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C00028 | AINx: Input voltage | 24547 | 5FE3 | A | 2 | INTEGER_16 | 100 | ☑ | | |
| C00029 | AINx: Input current | 24546 | 5FE2 | A | 2 | INTEGER_16 | 100 | ☑ | | |
| C00033 | AINx: Output value | 24542 | 5FDE | A | 2 | INTEGER_16 | 100 | ☑ | | |
| C00034 | AINx: Configuration | 24541 | 5FDD | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00036 | DC braking: Current | 24539 | 5FDB | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00039 | Fixed setpoint x (L_NSet_1 n-Fix) | 24536 | 5FD8 | A | 15 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00050 | MCTRL: Speed setpoint | 24525 | 5FCD | E | 1 | INTEGER_32 | 1 | ☑ | | |
| C00051 | MCTRL: Actual speed value | 24524 | 5FCC | E | 1 | INTEGER_32 | 1 | ☑ | | |
| C00052 | Motor voltage | 24523 | 5FCB | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00053 | DC-bus voltage | 24522 | 5FCA | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00054 | Motor current | 24521 | 5FC9 | E | 1 | UNSIGNED_16 | 100 | ☑ | | |
| C00055 | Actual values | 24520 | 5FC8 | A | 4 | INTEGER_16 | 1 | ☑ | | |
| C00056 | Torque | 24519 | 5FC7 | A | 2 | INTEGER_32 | 100 | ☑ | | |
| C00057 | Maximum torque | 24518 | 5FC6 | E | 1 | UNSIGNED_32 | 100 | ☑ | | |
| C00058 | Output frequency | 24517 | 5FC5 | E | 1 | INTEGER_32 | 100 | ☑ | | |
| C00059 | Appl.: Reference frequency C11 | 24516 | 5FC4 | E | 1 | UNSIGNED_32 | 100 | ☑ | | |
| C00060 | Motor rotor position | 24515 | 5FC3 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00061 | Heatsink temperature | 24514 | 5FC2 | E | 1 | INTEGER_16 | 1 | ☑ | | |
| C00062 | Interior temperature | 24513 | 5FC1 | A | 1 | INTEGER_16 | 1 | ☑ | | |
| C00063 | Motor temperature | 24512 | 5FC0 | A | 3 | INTEGER_16 | 1 | ☑ | | |
| C00064 | Device utilisation (Ixt) | 24511 | 5FBF | A | 3 | INTEGER_16 | 100 | ☑ | | |
| C00065 | Supply voltage 24V | 24510 | 5FBE | E | 1 | INTEGER_16 | 10 | ☑ | | |
| C00066 | Thermal motor load (I²xt) | 24509 | 5FBD | E | 1 | INTEGER_16 | 100 | ☑ | | |
| C00070 | Vp speed controller | 24505 | 5FB9 | A | 3 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00071 | Ti speed controller | 24504 | 5FB8 | A | 3 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00072 | SC: Tdn speed controller | 24503 | 5FB7 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00073 | Imax/M controller gain | 24502 | 5FB6 | A | 2 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00074 | Reset time Imax/M controller | 24501 | 5FB5 | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00075 | Vp current controller | 24500 | 5FB4 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00076 | Ti current controller | 24499 | 5FB3 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00077 | SC: Vp field controller | 24498 | 5FB2 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00078 | SC: Tn field controller | 24497 | 5FB1 | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00079 | SC: Settings | 24496 | 5FB0 | A | 4 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00080 | Override point of field weakening | 24495 | 5FAF | E | 1 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00081 | Rated motor power | 24494 | 5FAE | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00082 | Motor rotor resistance | 24493 | 5FAD | E | 1 | UNSIGNED_32 | 1 | ☑ | ☑ | |
| C00083 | Motor rotor time constant | 24492 | 5FAC | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00084 | Motor stator resistance | 24491 | 5FAB | E | 1 | UNSIGNED_32 | 1 | ☑ | ☑ | |
| C00085 | Motor stator leakage inductance | 24490 | 5FAA | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00087 | Rated motor speed | 24488 | 5FA8 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00088 | Rated motor current | 24487 | 5FA7 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00089 | Rated motor frequency | 24486 | 5FA6 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00090 | Rated motor voltage | 24485 | 5FA5 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00091 | Motor cosine phi | 24484 | 5FA4 | E | 1 | UNSIGNED_8 | 100 | ☑ | ☑ | |
| C00092 | Motor magnetising inductance | 24483 | 5FA3 | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00093 | Power section ID | 24482 | 5FA2 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00095 | Motor magnetising current | 24480 | 5FA0 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00097 | Rated motor torque | 24478 | 5F9E | E | 1 | UNSIGNED_32 | 100 | ☑ | | |
| C00098 | Rated device current | 24477 | 5F9D | E | 1 | UNSIGNED_16 | 10 | ☑ | | |
| C00099 | Firmware version | 24476 | 5F9C | E | 1 | VISIBLE_STRING [12] | | ☑ | | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00100 | Firmware version | 24475 | 5F9B | A | 4 | UNSIGNED_8 | 1 | ☑ | | |
| C00101 | Add. acceleration time x | 24474 | 5F9A | A | 15 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00103 | Add. deceleration time x | 24472 | 5F98 | A | 15 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00104 | Position control at quick stop | 24471 | 5F97 | A | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00105 | Decel. time - quick stop | 24470 | 5F96 | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00106 | Auto DCB: Hold time | 24469 | 5F95 | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00107 | DC braking: Hold time | 24468 | 5F94 | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00114 | DigInX: Inversion | 24461 | 5F8D | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00115 | DI 1/2 & 6/7: Function | 24460 | 5F8C | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00117 | Status of brake output BD | 24458 | 5F8A | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C00118 | DigOutX: Inversion | 24457 | 5F89 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00120 | Setting of motor overload (I²xt) | 24455 | 5F87 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00121 | Warning threshold motor temperature | 24454 | 5F86 | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00123 | Device utilisat. threshold (Ixt) | 24452 | 5F84 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00129 | Brake resistance value | 24446 | 5F7E | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00130 | Rated brake resistor power | 24445 | 5F7D | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00131 | Thermal capacity - brake resistor | 24444 | 5F7C | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00133 | Brake resistor utilisation | 24442 | 5F7A | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00134 | L_NSet_1: Ramp smoothing | 24441 | 5F79 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00136 | Communication control words | 24439 | 5F77 | A | 2 | UNSIGNED_16 | 1 | ☑ | | |
| C00137 | Device status | 24438 | 5F76 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00138 | Internal control signals | 24437 | 5F75 | A | 3 | UNSIGNED_16 | 1 | ☑ | | |
| C00142 | Auto-start option | 24433 | 5F71 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00144 | Thermal switching frequency reduction | 24431 | 5F6F | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00148 | LS_DriveInterface: Error message config. | 24427 | 5F6B | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00150 | Status word | 24425 | 5F69 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00155 | Extended status word | 24420 | 5F64 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00158 | Cause of controller inhibit | 24417 | 5F61 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00159 | Cause of quick stop QSP | 24416 | 5F60 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00160 | Status determining error (16-bit) | 24415 | 5F5F | A | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00161 | LS_SetError_x: Error number | 24414 | 5F5E | A | 8 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00163 | Logbook - binary elements | 24412 | 5F5C | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00164 | Logbook - analog elements | 24411 | 5F5B | A | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00165 | Error information | 24410 | 5F5A | A | 2 | VISIBLE_STRING [14] | | ☑ | | |
| C00166 | Error information text | 24409 | 5F59 | A | 6 | VISIBLE_STRING [30] | | ☑ | | |
| C00168 | Status determining error | 24407 | 5F57 | E | 1 | UNSIGNED_32 | 1 | ☑ | | |
| C00169 | Logbook setting | 24406 | 5F56 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00170 | Current error | 24405 | 5F55 | E | 1 | UNSIGNED_32 | 1 | ☑ | | |
| C00173 | Mains voltage | 24402 | 5F52 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00174 | Reduc. brake chopper threshold | 24401 | 5F51 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00175 | Brake energy management | 24400 | 5F50 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00177 | Switching cycles | 24398 | 5F4E | A | 5 | UNSIGNED_32 | 1 | ☑ | | |
| C00178 | Elapsed-hour meter | 24397 | 5F4D | E | 1 | UNSIGNED_32 | 1 | ☑ | | |
| C00179 | Power-on time meter | 24396 | 5F4C | E | 1 | UNSIGNED_32 | 1 | ☑ | | |
| C00180 | Running time | 24395 | 5F4B | A | 3 | UNSIGNED_32 | 1 | ☑ | | |
| C00181 | Time settings | 24394 | 5F4A | A | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00182 | L_NSet_1: S-ramp time PT1 | 24393 | 5F49 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00184 | AutoFailReset repetition time | 24391 | 5F47 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00185 | AutoFailReset residual runtime | 24390 | 5F46 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00186 | Max. number of AutoFailReset processes | 24389 | 5F45 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00187 | Current AutoFailReset processes | 24388 | 5F44 | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C00188 | AutoFailReset configuration | 24387 | 5F43 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00189 | Resp. to too frequent AutoFailReset | 24386 | 5F42 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00190 | L_NSet_1: Setpoint arithmetic | 24385 | 5F41 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00199 | Description data | 24376 | 5F38 | A | 1 | VISIBLE_STRING [24] | | ☑ | ☑ | |
| C00200 | Firmware product type | 24375 | 5F37 | E | 1 | VISIBLE_STRING [19] | | ☑ | | |
| C00201 | Firmware | 24374 | 5F36 | A | 9 | VISIBLE_STRING [22] | | ☑ | | |
| C00203 | Product type code | 24372 | 5F34 | A | 9 | VISIBLE_STRING [24] | | ☑ | | |
| C00204 | Serial number | 24371 | 5F33 | A | 9 | VISIBLE_STRING [24] | | ☑ | | |
| C00220 | L_NSet_1: Acceleration time - add. setpoint | 24355 | 5F23 | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00221 | L_NSet_1: Deceleration time - add. setpoint | 24354 | 5F22 | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00222 | L_PCTRL_1: Vp | 24353 | 5F21 | E | 1 | INTEGER_16 | 10 | ☑ | ☑ | |
| C00223 | L_PCTRL_1: Tn | 24352 | 5F20 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00224 | L_PCTRL_1: Kd | 24351 | 5F1F | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00225 | L_PCTRL_1: MaxLimit | 24350 | 5F1E | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00226 | L_PCTRL_1: MinLimit | 24349 | 5F1D | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00227 | L_PCTRL_1: Accel. time | 24348 | 5F1C | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00228 | L_PCTRL_1: Deceleration time | 24347 | 5F1B | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00231 | L_PCTRL_1: Operating range | 24344 | 5F18 | A | 4 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00233 | L_PCTRL_1: Root function | 24342 | 5F16 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00234 | Oscillation damping influence | 24341 | 5F15 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00235 | Oscillation damping filter time | 24340 | 5F14 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00236 | Oscillation damping field weakening | 24339 | 5F13 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00241 | L_NSet_1: Hyst. NSet reached | 24334 | 5F0E | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00242 | L_PCTRL_1: Operating mode | 24333 | 5F0D | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00243 | L_PCTRL_1: Accel. time influence | 24332 | 5F0C | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00244 | L_PCTRL_1: Decel. time influence | 24331 | 5F0B | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00245 | L_PCTRL_1: PID output value | 24330 | 5F0A | E | 1 | INTEGER_16 | 100 | ☑ | | |
| C00246 | L_PCTRL_1: nAct_a internal | 24329 | 5F09 | E | 1 | INTEGER_16 | 100 | ☑ | | |
| C00247 | L_PCTRL_1: Window setpoint reached | 24328 | 5F08 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00249 | L_PT1_1: Time constant | 24326 | 5F06 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00250 | L_PT1 2-3: Time constant | 24325 | 5F05 | A | 2 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00251 | L_DT1_1: Time constant | 24324 | 5F04 | E | 1 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00252 | L_DT1_1: Gain | 24323 | 5F03 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00253 | L_DT1_1: Sensitivity | 24322 | 5F02 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00254 | Kp position controller | 24321 | 5F01 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00270 | SC: Freq. current setpoint filter | 24305 | 5EF1 | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00271 | SC: Current setpoint filter width | 24304 | 5EF0 | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00272 | SC: Current setpoint filter depth | 24303 | 5EEF | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00273 | Moment of inertia | 24302 | 5EEE | E | 1 | UNSIGNED_32 | 100 | ☑ | ☑ | |
| C00274 | SC: Max. change in acceleration | 24301 | 5EED | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00275 | Setpoint feedforward control filtering | 24300 | 5EEC | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00276 | SC: max. output voltage | 24299 | 5EEB | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00280 | SC: Filter time const. DC detection | 24295 | 5EE7 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00312 | System runtimes | 24263 | 5EC7 | A | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00321 | Main program runtime | 24254 | 5EBE | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00322 | Transmission mode CAN TxPDOs | 24253 | 5EBD | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00323 | Transmission mode CAN Rx PDOs | 24252 | 5EBC | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00324 | CAN transmit blocking time | 24251 | 5EBB | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00338 | L_Arithmetik_1: Function | 24237 | 5EAD | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00339 | L_Arithmetik_2: Function | 24236 | 5EAC | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00341 | CAN management - error configuration | 24234 | 5EAA | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00342 | CAN decoupling PDOInOut | 24233 | 5EA9 | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00343 | LP_CanIn decoupling value | 24232 | 5EA8 | A | 12 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00344 | LP_CanOut decoupling value | 24231 | 5EA7 | A | 12 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00345 | CAN error status | 24230 | 5EA6 | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C00347 | CAN status HeartBeat producer | 24228 | 5EA4 | A | 15 | UNSIGNED_8 | 1 | ☑ | | |
| C00349 | CAN setting - DIP switch | 24226 | 5EA2 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00350 | CAN node address | 24225 | 5EA1 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00351 | CAN baud rate | 24224 | 5EA0 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00352 | CAN Slave/Master | 24223 | 5E9F | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00353 | CAN IN/OUT COBID source | 24222 | 5E9E | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00354 | COBID | 24221 | 5E9D | A | 6 | UNSIGNED_32 | 1 | ☑ | ☑ | |
| C00355 | Active COBID | 24220 | 5E9C | A | 6 | UNSIGNED_16 | 1 | ☑ | | |
| C00356 | CAN time settings | 24219 | 5E9B | A | 5 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00357 | CAN monitoring times | 24218 | 5E9A | A | 3 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00358 | CANx_OUT data length | 24217 | 5E99 | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00359 | CAN status | 24216 | 5E98 | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C00360 | CAN telegram counter | 24215 | 5E97 | A | 12 | UNSIGNED_16 | 1 | ☑ | | |
| C00364 | CAN MessageError | 24211 | 5E93 | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C00366 | Number of CAN SDO channels | 24209 | 5E91 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00367 | CAN Sync-Rx-Identifier | 24208 | 5E90 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00368 | CAN Sync-Tx-Identifier | 24207 | 5E8F | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00369 | CAN Sync transmission cycle time | 24206 | 5E8E | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00370 | SyncTxRxTimes | 24205 | 5E8D | A | 2 | INTEGER_16 | 1 | ☑ | | |
| C00372 | CAN_Tx_Rx_Error | 24203 | 5E8B | A | 4 | UNSIGNED_8 | 1 | ☑ | | |
| C00381 | CAN Heartbeat producer time | 24194 | 5E82 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00385 | CAN node addr. HeartBeat producer | 24190 | 5E7E | A | 15 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00386 | CAN HeartBeat-ConsumerTime | 24189 | 5E7D | A | 15 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00400 | LS_PulseGenerator | 24175 | 5E6F | A | 3 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00401 | CANxInOut: Inversion | 24174 | 5E6E | A | 6 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00408 | LP_CanIn mapping selection | 24167 | 5E67 | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00409 | LP_CanIn mapping | 24166 | 5E66 | A | 12 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00410 | L_SignalMonitor_a: Signal sources | 24165 | 5E65 | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00411 | L_SignalMonitor_b: Signal sources | 24164 | 5E64 | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00412 | L_SignalMonitor_b: Inversion | 24163 | 5E63 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00413 | L_SignalMonitor_a: Offs./gain | 24162 | 5E62 | A | 8 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00417 | Activate resolver error comp. | 24158 | 5E5E | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00420 | Encoder number of increments | 24155 | 5E5B | A | 3 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00421 | LS_Multi-Encoder: Supply voltage | 24154 | 5E5A | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00422 | LS_Multi-Encoder: Encoder type | 24153 | 5E59 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00423 | DOx: Delay times | 24152 | 5E58 | A | 10 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C00424 | Pulse form TTL encoder | 24151 | 5E57 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00425 | Encoder scanning time | 24150 | 5E56 | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00426 | SSI encoder: Data bits | 24149 | 5E55 | A | 7 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00427 | SSI encoder: Bit rate | 24148 | 5E54 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00428 | SSI encoder: Coding | 24147 | 5E53 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00434 | OxU/I: Gain | 24141 | 5E4D | A | 4 | INTEGER_16 | 100 | ☑ | ☑ | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00435 | OxU/I: Offset | 24140 | 5E4C | A | 4 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00436 | OxU: Voltage | 24139 | 5E4B | A | 2 | INTEGER_16 | 100 | ☑ | | |
| C00437 | OxI: Current | 24138 | 5E4A | A | 2 | INTEGER_32 | 1000 | ☑ | | |
| C00439 | OxU/I: Input value | 24136 | 5E48 | A | 4 | INTEGER_16 | 100 | ☑ | | |
| C00440 | LS_AnalogIn1: PT1 time constant | 24135 | 5E47 | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00441 | Decoupling AnalogOut | 24134 | 5E46 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00442 | AOutx: Decoupling value | 24133 | 5E45 | A | 4 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00443 | DIx: Level | 24132 | 5E44 | A | 2 | UNSIGNED_16 | 1 | ☑ | | |
| C00444 | DOx: Level | 24131 | 5E43 | A | 2 | UNSIGNED_16 | 1 | ☑ | | |
| C00445 | FreqInxx_nOut_v | 24130 | 5E42 | A | 2 | INTEGER_16 | 1 | ☑ | | |
| C00446 | FreqInxx_nOut_a | 24129 | 5E41 | A | 2 | INTEGER_16 | 100 | ☑ | | |
| C00447 | DigOut decoupling | 24128 | 5E40 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00448 | DigOut decoupling value | 24127 | 5E3F | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00449 | FreqInxx_dnOut_p | 24126 | 5E3E | A | 1 | INTEGER_32 | 1 | ☑ | | |
| C00461 | Remote: Acceleration/deceleration time | 24114 | 5E32 | A | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00462 | Remote: Control | 24113 | 5E31 | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00463 | Remote: MCK control | 24112 | 5E30 | A | 1 | UNSIGNED_32 | 1 | ☑ | ☑ | |
| C00464 | Remote: Monitoring timeout | 24111 | 5E2F | A | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00465 | Keypad: Time-out welcome screen | 24110 | 5E2E | E | 1 | INTEGER_32 | 1 | ☑ | ☑ | |
| C00466 | Keypad: Default parameter | 24109 | 5E2D | E | 1 | INTEGER_32 | 1 | ☑ | ☑ | |
| C00467 | Keypad: Default welcome screen | 24108 | 5E2C | E | 1 | INTEGER_32 | 1 | ☑ | ☑ | |
| C00469 | Keypad: STOP key function | 24106 | 5E2A | E | 1 | INTEGER_32 | 1 | ☑ | ☑ | ☑ |
| C00470 | LS_ParFree_b | 24105 | 5E29 | A | 32 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00471 | LS_ParFree | 24104 | 5E28 | A | 32 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00472 | LS_ParFree_a | 24103 | 5E27 | A | 16 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00473 | LS_ParFree_v | 24102 | 5E26 | A | 8 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00474 | LS_ParFree_p | 24101 | 5E25 | A | 8 | INTEGER_32 | 1 | ☑ | ☑ | |
| C00475 | LS_ParFreeUnit_1_2 | 24100 | 5E24 | A | 32 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C00476 | LS_ParFree_a_2 | 24099 | 5E23 | A | 16 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00477 | LS_ParFree_2 | 24098 | 5E22 | A | 32 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00478 | LS_ParFree_v_2 | 24097 | 5E21 | A | 8 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00479 | LS_ParFree32 | 24096 | 5E20 | A | 8 | INTEGER_32 | 1 | ☑ | ☑ | |
| C00480 | LS_DisFree_b | 24095 | 5E1F | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00481 | LS_DisFree | 24094 | 5E1E | A | 8 | UNSIGNED_16 | 1 | ☑ | | |
| C00482 | LS_DisFree_a | 24093 | 5E1D | A | 8 | INTEGER_16 | 100 | ☑ | | |
| C00483 | LS_DisFree_p | 24092 | 5E1C | A | 8 | INTEGER_32 | 1 | ☑ | | |
| C00484 | Application units: Offset | 24091 | 5E1B | A | 4 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00485 | Application units display factor | 24090 | 5E1A | A | 4 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C00486 | Application units: | 24089 | 5E19 | A | 4 | VISIBLE_STRING [7] | | ☑ | ☑ | |
| C00487 | Application units | 24088 | 5E18 | A | 4 | INTEGER_32 | 100 | ☑ | | |
| C00488 | L_JogCtrlEdgeDetect_1 | 24087 | 5E17 | A | 6 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00490 | Position encoder selection | 24085 | 5E15 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00492 | Hiperface: Detected TypCode | 24083 | 5E13 | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C00493 | Hiperface: TypCode | 24082 | 5E12 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00494 | Hiperface: Resolutions | 24081 | 5E11 | A | 2 | UNSIGNED_32 | 1 | ☑ | ☑ | |
| C00495 | Speed sensor selection | 24080 | 5E10 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00496 | Encoder evaluation method DigIn12 | 24079 | 5E0F | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00497 | Nact filter time constant | 24078 | 5E0E | A | 4 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00498 | Open-circuit monitoring | 24077 | 5E0D | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00505 | Password data | 24070 | 5E06 | A | 3 | VISIBLE_STRING [16] | | ☑ | ☑ | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00507 | Current password protection | 24068 | 5E04 | A | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00517 | User menu | 24058 | 5DFA | A | 32 | INTEGER_32 | 1000 | ☑ | ☑ | |
| C00560 | Fan switching status | 24015 | 5DCF | A | 2 | UNSIGNED_8 | 1 | ☑ | | |
| C00561 | Failure indication | 24014 | 5DCE | A | 5 | UNSIGNED_8 | 1 | ☑ | | |
| C00565 | Resp. to mains phase failure | 24010 | 5DCA | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00566 | Resp. to fan failure | 24009 | 5DC9 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00567 | Resp. to speed controller limited | 24008 | 5DC8 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00569 | Resp. to peak current | 24006 | 5DC6 | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00570 | Resp. to controller limitations | 24005 | 5DC5 | A | 4 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00572 | Brake resistor overload threshold | 24003 | 5DC3 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00574 | Resp. to brake resist. overtemp. | 24001 | 5DC1 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00576 | SC: Field feedforward control | 23999 | 5DBF | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00577 | SC: Vp field weakening controller | 23998 | 5DBE | E | 1 | UNSIGNED_16 | 10000 | ☑ | ☑ | |
| C00578 | SC: Tn field weakening controller | 23997 | 5DBD | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00579 | Resp. to max. speed/output freq. reached | 23996 | 5DBC | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00580 | Resp. to operating system error | 23995 | 5DBB | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00581 | Resp. to LS_SetError_x | 23994 | 5DBA | A | 8 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00582 | Resp. to heatsink temp.> shutdown temp. -5°C | 23993 | 5DB9 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00583 | Resp. to motor temperature KTY | 23992 | 5DB8 | A | 6 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00585 | Resp. to motor overtemp. PTC | 23990 | 5DB6 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00586 | Resp. to encoder open circuit HTL | 23989 | 5DB5 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00588 | Resp. to max. speed at switching freq. | 23987 | 5DB3 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00590 | Resp. to switch. frequency red. | 23985 | 5DB1 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00592 | Resp. to CAN bus connection | 23983 | 5DAF | A | 5 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00593 | Resp. to CANx_IN monitoring | 23982 | 5DAE | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00594 | Resp. to control word error | 23981 | 5DAD | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00595 | MCK: Resp. to MCK error | 23980 | 5DAC | A | 15 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00597 | Resp. to motor phase failure | 23978 | 5DAA | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00598 | Resp. to open circuit AINx | 23977 | 5DA9 | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00599 | Motor phase failure threshold | 23976 | 5DA8 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00600 | Resp. to DC bus voltage | 23975 | 5DA7 | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00601 | Delayed resp. to fault: DC bus overvoltage | 23974 | 5DA6 | A | 1 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C00602 | Resp. to earth fault | 23973 | 5DA5 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00603 | Resp. to feedback | 23972 | 5DA4 | A | 4 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00604 | Resp. to device overload (Ixt) | 23971 | 5DA3 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00606 | Resp. to motor overload ($I^2xt$) | 23969 | 5DA1 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00607 | Resp. to max. freq. feedb. DIG12/67 | 23968 | 5DA0 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00608 | Resp. to maximum torque | 23967 | 5D9F | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00609 | Resp. to maximum current | 23966 | 5D9E | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00620 | System connection list: 16-bit | 23955 | 5D93 | A | 104 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00621 | System connection list: Bool | 23954 | 5D92 | A | 175 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00622 | System connection list: Angle | 23953 | 5D91 | A | 24 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00630 | L_Limit 1-2: Min/Max | 23945 | 5D89 | A | 4 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00631 | L_LimitPhi 1-3: Min/Max | 23944 | 5D88 | A | 6 | INTEGER_32 | 1 | ☑ | ☑ | |
| C00632 | L_NSet_1: Max. skip freq. | 23943 | 5D87 | A | 3 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00633 | L_NSet_1: Min. skip freq. | 23942 | 5D86 | A | 3 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00634 | L_NSet_1: wState | 23941 | 5D85 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00635 | L_NSet_1: nMaxLimit | 23940 | 5D84 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00636 | L_NSet_1: nMinLimit | 23939 | 5D83 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00637 | L_NSet_1: Output blocking zones | 23938 | 5D82 | E | 1 | INTEGER_16 | 100 | ☑ | | |
| C00638 | L_NSet_1: Output ramp rounding | 23937 | 5D81 | E | 1 | INTEGER_16 | 100 | ☑ | | |
| C00639 | L_NSet_1: Output add.value | 23936 | 5D80 | E | 1 | INTEGER_16 | 100 | ☑ | | |
| C00640 | L_NSet_1: nNOut_a | 23935 | 5D7F | E | 1 | INTEGER_16 | 100 | ☑ | | |
| C00643 | Resp. to PLI monitoring | 23932 | 5D7C | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00644 | PLI traversing direction | 23931 | 5D7B | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00645 | PLI max. permissible deflection | 23930 | 5D7A | A | 2 | INTEGER_16 | 10 | ☑ | ☑ | |
| C00646 | PLI current amplitude | 23929 | 5D79 | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00647 | PLI ramp time | 23928 | 5D78 | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00650 | L_Arithmetik 3-5: Function | 23925 | 5D75 | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00660 | L_FixSet_a_1: Analog values | 23915 | 5D6B | A | 16 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00661 | L_FixSet_w_1: Fixed values | 23914 | 5D6A | A | 16 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00662 | L_FixSet_w_2: Fixed values | 23913 | 5D69 | A | 16 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00670 | L_OffsetGainP_1: Gain | 23905 | 5D61 | E | 1 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C00671 | L_OffsetGainP_2: Gain | 23904 | 5D60 | E | 1 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C00672 | L_OffsetGainP_3: Gain | 23903 | 5D5F | E | 1 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C00673 | L_OffsetGainPhiP 1-2: Offset | 23902 | 5D5E | A | 2 | INTEGER_32 | 1 | ☑ | ☑ | |
| C00674 | L_OffsetGainPhiP 1-2: Gain | 23901 | 5D5D | A | 2 | INTEGER_32 | 1 | ☑ | ☑ | |
| C00677 | L_GainOffsetP 1-3: Parameter | 23898 | 5D5A | A | 6 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00678 | L_GainOffsetPhiP 1-2: Parameter | 23897 | 5D59 | A | 4 | INTEGER_32 | 1 | ☑ | ☑ | |
| C00679 | L_MulDiv_2: Parameter | 23896 | 5D58 | A | 2 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00680 | L_Compare_1: Fct. | 23895 | 5D57 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00681 | L_Compare_1: Hysteresis | 23894 | 5D56 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00682 | L_Compare_1: Window | 23893 | 5D55 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00685 | L_Compare_2: Fct. | 23890 | 5D52 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00686 | L_Compare_2: Hysteresis | 23889 | 5D51 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00687 | L_Compare_2: Window | 23888 | 5D50 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00690 | L_Compare_3: Fct. | 23885 | 5D4D | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00691 | L_Compare_3: Hysteresis | 23884 | 5D4C | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00692 | L_Compare_3: Window | 23883 | 5D4B | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00693 | L_Compare 4-5: Fct. | 23882 | 5D4A | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00694 | L_Compare 4-5: Hysteresis | 23881 | 5D49 | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00695 | L_Compare 4-5: Window | 23880 | 5D48 | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00696 | L_OffsetGainP_1: Offset | 23879 | 5D47 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00697 | L_OffsetGainP_2: Offset | 23878 | 5D46 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00698 | L_OffsetGainP_3: Offset | 23877 | 5D45 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00699 | L_MulDiv_1: Parameter | 23876 | 5D44 | A | 2 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00700 | LA_NCtrl: Analog connection list | 23875 | 5D43 | A | 29 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00701 | LA_NCtrl: Digital connection list | 23874 | 5D42 | A | 48 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00710 | LA_TabPos: Analog connection list | 23865 | 5D39 | A | 31 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00711 | LA_TabPos: Digital connection list | 23864 | 5D38 | A | 56 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00712 | LA_TabPos: phi connection list | 23863 | 5D37 | A | 3 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00720 | L_DigitalDelay_1: Delay | 23855 | 5D2F | A | 2 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00721 | L_DigitalDelay 2,3: Delay | 23854 | 5D2E | A | 4 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C00725 | Current switching frequency | 23850 | 5D2A | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C00727 | LS_Keypad digital values | 23848 | 5D28 | A | 8 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00728 | Analog values - keypad | 23847 | 5D27 | A | 3 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00729 | Remote: Setpoint selection | 23846 | 5D26 | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00760 | LA_SwitchPos: Analog connection list | 23815 | 5D07 | A | 25 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00761 | LA_SwitchPos: Digital connection list | 23814 | 5D06 | A | 47 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00800 | L_MPot_1: Upper limit | 23775 | 5CDF | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00801 | L_MPot_1: Lower limit | 23774 | 5CDE | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00802 | L_MPot_1: Accel. time | 23773 | 5CDD | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00803 | L_MPot_1: Decel. time | 23772 | 5CDC | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00804 | L_MPot_1: Inactive fct. | 23771 | 5CDB | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00805 | L_MPot_1: Init fct. | 23770 | 5CDA | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00806 | L_MPot_1: Use | 23769 | 5CD9 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00807 | L_NLim_1: Max.SkipFrq. | 23768 | 5CD8 | A | 3 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00808 | L_NLim_1: Min.SkipFrq. | 23767 | 5CD7 | A | 3 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00809 | L_NLim_2: Max.SkipFrq. | 23766 | 5CD6 | A | 3 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00810 | L_NLim_2: Min.SkipFrq. | 23765 | 5CD5 | A | 3 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00811 | L_NLim_1: Current output value | 23764 | 5CD4 | A | 2 | INTEGER_16 | 100 | ☑ | | |
| C00812 | L_NLim: Current status | 23763 | 5CD3 | A | 2 | UNSIGNED_16 | 1 | ☑ | | |
| C00820 | L_DigitalLogic_1: Function | 23755 | 5CCB | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00821 | L_DigitalLogic_1: Truth table | 23754 | 5CCA | A | 8 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00822 | L_DigitalLogic_2: Function | 23753 | 5CC9 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00823 | L_DigitalLogic_2: Truth table | 23752 | 5CC8 | A | 8 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00824 | L_DigitalLogic5_1: Function | 23751 | 5CC7 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00825 | L_DigitalLogic5_1: Truth table | 23750 | 5CC6 | A | 32 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00826 | L_DigitalLogic5_2: Function | 23749 | 5CC5 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00827 | L_DigitalLogic5_2: Truth table | 23748 | 5CC4 | A | 32 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00828 | L_DigitalLogic_3: Function | 23747 | 5CC3 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00829 | L_DigitalLogic_3: truth table | 23746 | 5CC2 | A | 8 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00830 | 16-bit inputs [%] | 23745 | 5CC1 | A | 95 | INTEGER_16 | 100 | ☑ | | |
| C00831 | 16-bit inputs | 23744 | 5CC0 | A | 95 | UNSIGNED_16 | 1 | ☑ | | |
| C00832 | 16-bit inputs [rpm] | 23743 | 5CBF | A | 95 | INTEGER_16 | 1 | ☑ | | |
| C00833 | Binary inputs | 23742 | 5CBE | A | 129 | UNSIGNED_8 | 1 | ☑ | | |
| C00834 | 32-bit inputs [incr] | 23741 | 5CBD | A | 8 | INTEGER_32 | 1 | ☑ | | |
| C00835 | 16-bit inputs [%] (Set2) | 23740 | 5CBC | A | 140 | INTEGER_16 | 100 | ☑ | | |
| C00836 | 16-bit inputs (Set2) | 23739 | 5CBB | A | 140 | UNSIGNED_16 | 1 | ☑ | | |
| C00837 | 16-bit inputs [rpm] (Set2) | 23738 | 5CBA | A | 140 | INTEGER_16 | 1 | ☑ | | |
| C00838 | Binary inputs (Set2) | 23737 | 5CB9 | A | 201 | UNSIGNED_8 | 1 | ☑ | | |
| C00839 | 32-bit inputs [incr] (Set2) | 23736 | 5CB8 | A | 76 | INTEGER_32 | 1 | ☑ | | |
| C00840 | 16-bit inputs I/O level [%] | 23735 | 5CB7 | A | 73 | INTEGER_16 | 100 | ☑ | | |
| C00841 | 16-bit inputs I/O level | 23734 | 5CB6 | A | 73 | UNSIGNED_16 | 1 | ☑ | | |
| C00843 | Binary inputs I/O level | 23732 | 5CB4 | A | 175 | UNSIGNED_8 | 1 | ☑ | | |
| C00844 | 32-bit inputs I/O level [incr] | 23731 | 5CB3 | A | 12 | INTEGER_32 | 1 | ☑ | | |
| C00866 | CAN input words | 23709 | 5C9D | A | 12 | UNSIGNED_16 | 1 | ☑ | | |
| C00868 | CAN output words | 23707 | 5C9B | A | 12 | UNSIGNED_16 | 1 | ☑ | | |
| C00876 | MCI input words | 23699 | 5C93 | A | 16 | UNSIGNED_16 | 1 | ☑ | | |
| C00877 | MCI output words | 23698 | 5C92 | A | 16 | UNSIGNED_16 | 1 | ☑ | | |
| C00890 | MCI_InOut: Inversion | 23685 | 5C85 | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00905 | Motor phase direction of rotation | 23670 | 5C76 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00909 | Speed limitation | 23666 | 5C72 | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00910 | Frequency limitation | 23665 | 5C71 | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00915 | Motor cable length | 23660 | 5C6C | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00916 | Motor cable cross-section | 23659 | 5C6B | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00917 | Motor cable resistance | 23658 | 5C6A | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C00918 | SC: Start motor magnetising current | 23657 | 5C69 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00925 | LS_Resolver: Number of pole pairs | 23650 | 5C62 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00926 | Pole position | 23649 | 5C61 | A | 2 | INTEGER_16 | 10 | ☑ | ☑ | |
| C00927 | Motor rotor position | 23648 | 5C60 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C00937 | Field-oriented motor currents | 23638 | 5C56 | A | 2 | INTEGER_16 | 100 | ☑ | | |
| C00938 | PSM: Maximum motor current field weakening | 23637 | 5C55 | E | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00939 | Ultimate motor current | 23636 | 5C54 | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00940 | L_ConvW numerator | 23635 | 5C53 | A | 4 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00941 | L_ConvW denominator | 23634 | 5C52 | A | 4 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00942 | L_ConvW conversion method | 23633 | 5C51 | A | 4 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00950 | L_Interpolator_1: Activation FB functions | 23625 | 5C49 | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00951 | L_Interpolator_1: No. of interpolation steps | 23624 | 5C48 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00952 | L_Interpolator_1: Limit value - error cycles | 23623 | 5C47 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00953 | L_Interpolator_1: Speed-up | 23622 | 5C46 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00959 | L_Curve: Current output value | 23616 | 5C40 | A | 3 | INTEGER_16 | 100 | ☑ | | |
| C00960 | L_Curve_1: Selected curve type | 23615 | 5C3F | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C00961 | L_Curve_1: Input limitation | 23614 | 5C3E | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00963 | L_Curve_1: Table X-values | 23612 | 5C3C | A | 32 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00964 | L_Curve_1: Table Y-values | 23611 | 5C3B | A | 32 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00965 | Max. motor speed | 23610 | 5C3A | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00966 | VFC: Time const. slip comp. | 23609 | 5C39 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00967 | VFC: Frequency interpol. point n | 23608 | 5C38 | A | 11 | INTEGER_16 | 10 | ☑ | ☑ | |
| C00968 | VFC: Voltage interpol. point n | 23607 | 5C37 | A | 11 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00971 | VFC: V/f +encoder limitation | 23604 | 5C34 | A | 2 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00972 | VFC: Vp V/f +encoder | 23603 | 5C33 | E | 1 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C00973 | VFC: Ti V/f +encoder | 23602 | 5C32 | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00975 | VFC-ECO: Vp CosPhi controller | 23600 | 5C30 | E | 1 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C00976 | VFC-ECO: Ti CosPhi controller | 23599 | 5C2F | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00977 | VFC-ECO: Minimum voltage V/f | 23598 | 5C2E | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00978 | VFC-ECO: Voltage reduction | 23597 | 5C2D | E | 1 | INTEGER_16 | 1 | ☑ | | |
| C00979 | Cosine phi | 23596 | 5C2C | A | 2 | INTEGER_16 | 100 | ☑ | | |
| C00980 | Output power | 23595 | 5C2B | A | 2 | INTEGER_32 | 1000 | ☑ | | |
| C00981 | Energy display | 23594 | 5C2A | A | 2 | INTEGER_32 | 100 | ☑ | | |
| C00982 | VFC-ECO: Voltage reduction ramp | 23593 | 5C29 | E | 1 | UNSIGNED_8 | 10 | ☑ | ☑ | |
| C00985 | SLVC: Field current controller gain | 23590 | 5C26 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00986 | SLVC: Cross current controller gain | 23589 | 5C25 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00987 | Inverter motor brake: nAdd | 23588 | 5C24 | E | 1 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00988 | Inverter motor brake: PT1 filter time | 23587 | 5C23 | E | 1 | INTEGER_16 | 10 | ☑ | ☑ | |
| C00990 | Flying restart fct.: Activate | 23585 | 5C21 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C00991 | Flying restart fct.: Process | 23584 | 5C20 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C00992 | Flying restart fct.: Start frequency | 23583 | 5C1F | E | 1 | INTEGER_16 | 1 | ☑ | ☑ | |
| C00993 | Flying restart fct: Int. time | 23582 | 5C1E | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C00994 | Flying restart fct.: Current | 23581 | 5C1D | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00995 | SLPSM: Controlled current setpoint | 23580 | 5C1C | A | 2 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C00996 | SLPSM: Switching speed | 23579 | 5C1B | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00997 | SLPSM: Filter cutoff frequency | 23578 | 5C1A | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C00998 | SLPSM: Filter time rotor position | 23577 | 5C19 | A | 2 | INTEGER_16 | 10 | ☑ | ☑ | |
| C00999 | SLPSM: PLL gain | 23576 | 5C18 | E | 1 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01000 | MCTRL: Status | 23575 | 5C17 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C01010 | L_ArithmetikPhi 1-3: Function | 23565 | 5C0D | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01020 | L_Odometer_1: Memory length | 23555 | 5C03 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01021 | L_Odometer_1: Memory type | 23554 | 5C02 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C01022 | L_Odometer_1: Input selection | 23553 | 5C01 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01023 | L_Odometer_1: Edge selection | 23552 | 5C00 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01025 | L_Curve_2: Selected curve type | 23550 | 5BFE | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01026 | L_Curve_2: Input limitation | 23549 | 5BFD | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C01028 | L_Curve_2: Table X-values | 23547 | 5BFB | A | 32 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01029 | L_Curve_2: Table Y-values | 23546 | 5BFA | A | 32 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01030 | L_Curve_3: Selected curve type | 23545 | 5BF9 | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01031 | L_Curve_3: Input limitation | 23544 | 5BF8 | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C01033 | L_Curve_3: Table X-values | 23542 | 5BF6 | A | 32 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01034 | L_Curve_3: Table Y-values | 23541 | 5BF5 | A | 32 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01040 | L_SRFG_1..2 linear ramp time | 23535 | 5BEF | A | 2 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C01041 | L_SRFG_1..2 S-ramp time | 23534 | 5BEE | A | 2 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C01042 | L_SRFG_1..2 limitations of output values | 23533 | 5BED | A | 4 | INTEGER_16 | 100 | ☑ | ☑ | |
| C01045 | L_ConvAP 1-3: numerator/denominator | 23530 | 5BEA | A | 6 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01046 | L_ConvPA 1-3: byDivision | 23529 | 5BE9 | A | 3 | INTEGER_8 | 1 | ☑ | ☑ | |
| C01047 | L_GearComp: Offset | 23528 | 5BE8 | A | 1 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01048 | L_GearComp: Num_Denom | 23527 | 5BE7 | A | 2 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01049 | L_CalcDiameter: Status | 23526 | 5BE6 | A | 1 | INTEGER_16 | 1 | ☑ | | |
| C01050 | L_CalcDiameter_1: Diameter recalculation | 23525 | 5BE5 | A | 2 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C01051 | L_CalcDiameter: Filter time constant | 23524 | 5BE4 | A | 1 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01052 | L_CalcDiameter: Web break monitoring | 23523 | 5BE3 | A | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C01053 | L_ProcessCtrl: Controller times | 23522 | 5BE2 | A | 5 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01054 | L_ProcessCtrl: System deviation | 23521 | 5BE1 | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C01055 | L_ProcessCtrl: Correcting variable limitation | 23520 | 5BE0 | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01056 | L_ProcessCtrl: Controller gain | 23519 | 5BDF | A | 1 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C01057 | L_CalcDiameter: Current diameter | 23518 | 5BDE | A | 1 | UNSIGNED_32 | 1000 | ☑ | | |
| C01058 | L_PosCtrlLin_1-2: bDisable | 23517 | 5BDD | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01059 | L_PosCtrlLin 1-2: Positioning behaviour | 23516 | 5BDC | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01060 | L_PosCtrlLin 1-2: Ramps | 23515 | 5BDB | A | 6 | INTEGER_32 | 1000 | ☑ | ☑ | |
| C01061 | L_PosCtrlLin 1-2: Traversing speed | 23514 | 5BDA | A | 4 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01062 | L_SwitchPoint: Dead time | 23513 | 5BD9 | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01063 | L_SwitchPoint: Hysteresis | 23512 | 5BD8 | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01064 | L_SwitchPoint: CenterMode | 23511 | 5BD7 | A | 4 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01065 | L_SwitchPoint: Running time | 23510 | 5BD6 | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01066 | L_SwitchPoint: Status | 23509 | 5BD5 | A | 4 | INTEGER_16 | 1 | ☑ | | |
| C01067 | Inversion of gearbox stages | 23508 | 5BD4 | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01069 | L_DFSET: Ramp settings | 23506 | 5BD2 | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01070 | L_DFSET: Angular trimming | 23505 | 5BD1 | A | 1 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01071 | L_DFSET: Following error limit | 23504 | 5BD0 | A | 1 | UNSIGNED_32 | 1 | ☑ | ☑ | |
| C01072 | L_DFSET: Multiplier - angular trimming | 23503 | 5BCF | A | 1 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01073 | L_DFSET: Evaluation - setpoint angle integrator | 23502 | 5BCE | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01074 | L_DFSET: Zero pulse divider | 23501 | 5BCD | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01075 | L_DFSET: Synchronisation mode | 23500 | 5BCC | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01076 | L_DFRFG: Times | 23499 | 5BCB | A | 2 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C01077 | L_DFRFG: Max. speed-up | 23498 | 5BCA | A | 1 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01078 | L_DFRFG: Following error | 23497 | 5BC9 | A | 1 | UNSIGNED_32 | 1 | ☑ | ☑ | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C01079 | L_DFRFG: Synchronisation window | 23496 | 5BC8 | A | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01080 | L_DFRFG: Offset | 23495 | 5BC7 | A | 1 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01081 | L_DFRFG: Sync. direction / TP function | 23494 | 5BC6 | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01082 | LS_WriteParamList: Execute Mode | 23493 | 5BC5 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01083 | LS_WriteParamList: Error status | 23492 | 5BC4 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C01084 | LS_WriteParamList: Error line | 23491 | 5BC3 | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C01085 | LS_WriteParamList: Index | 23490 | 5BC2 | A | 32 | INTEGER_32 | 1000 | ☑ | ☑ | |
| C01086 | LS_WriteParamList: WriteValue_1 | 23489 | 5BC1 | A | 32 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01087 | LS_WriteParamList: WriteValue_2 | 23488 | 5BC0 | A | 32 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01088 | LS_WriteParamList: WriteValue_3 | 23487 | 5BBF | A | 32 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01089 | LS_WriteParamList: WriteValue_4 | 23486 | 5BBE | A | 32 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01090 | LS_ParReadWrite 1-6: Index | 23485 | 5BBD | A | 6 | INTEGER_32 | 1000 | ☑ | ☑ | |
| C01091 | LS_ParReadWrite 1-6: Cycle time | 23484 | 5BBC | A | 6 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01092 | LS_ParReadWrite 1-6: FailState | 23483 | 5BBB | A | 6 | UNSIGNED_16 | 1 | ☑ | | |
| C01093 | LS_ParReadWrite 1-6: Arithmetic mode | 23482 | 5BBA | A | 6 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01094 | LS_ParReadWrite 1-6: Numerator | 23481 | 5BB9 | A | 6 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01095 | LS_ParReadWrite 1-6: Denominator | 23480 | 5BB8 | A | 6 | INTEGER_16 | 1 | ☑ | ☑ | |
| C01100 | Function L_Counter 1-3 | 23475 | 5BB3 | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01101 | Comparison L_Counter 1-3 | 23474 | 5BB2 | A | 3 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01110 | LS_MultiEncoder: Solid measure | 23465 | 5BA9 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01111 | SSI: Encoder constant | 23464 | 5BA8 | A | 2 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01112 | LS_MultiEncoder: PosValues | 23463 | 5BA7 | A | 3 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01119 | LS_MultiEncoder: Current position | 23456 | 5BA0 | A | 2 | INTEGER_32 | 10000 | ☑ | | |
| C01120 | Sync signal source | 23455 | 5B9F | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01121 | Sync cycle time setpoint | 23454 | 5B9E | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01122 | Sync phase position | 23453 | 5B9D | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01123 | Sync window | 23452 | 5B9C | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01124 | Sync correction width | 23451 | 5B9B | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01130 | LS_RetainData: Selection | 23445 | 5B95 | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01131 | LS_RetainData: 16Bit data | 23444 | 5B94 | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01132 | LS_RetainData: 32Bit data | 23443 | 5B93 | A | 4 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01133 | LS_RetainData: Bool data | 23442 | 5B92 | A | 4 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01138 | L_Transient 1-4: Function | 23437 | 5B8D | A | 4 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01139 | L_Transient 1-4: Pulse duration | 23436 | 5B8C | A | 4 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01140 | L_Transient 5-8: Function | 23435 | 5B8B | A | 4 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01141 | L_Transient 5-8 pulse duration | 23434 | 5B8A | A | 4 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01150 | L_PhaseIntK: Function | 23425 | 5B81 | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01151 | L_PhaseIntK: Compare | 23424 | 5B80 | A | 2 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01190 | Motor thermal sensor | 23385 | 5B59 | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01191 | PTC characteristic: Temperature 1/2 | 23384 | 5B58 | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01192 | PTC characteristic: Resistance 1/2 | 23383 | 5B57 | A | 2 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01193 | Feedback system motor temperature | 23382 | 5B56 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01201 | MCK: Cycle | 23374 | 5B4E | A | 1 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01202 | MCK: iM motor/process | 23373 | 5B4D | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01203 | MCK: iG motor/position encoder | 23372 | 5B4C | A | 2 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01204 | MCK: Feed constant | 23371 | 5B4B | E | 1 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01205 | MCK: Position resolution | 23370 | 5B4A | E | 1 | INTEGER_32 | 10000 | ☑ | | |
| C01206 | MCK: Mounting direction | 23369 | 5B49 | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C01210 | MCK: Current positions | 23365 | 5B45 | A | 7 | INTEGER_32 | 10000 | ☑ | | |
| C01211 | MCK: Speed | 23364 | 5B44 | A | 1 | INTEGER_32 | 10000 | ☑ | | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C01213 | MCK: Max. traversing distance | 23362 | 5B42 | A | 1 | INTEGER_32 | 1 | ☑ | | |
| C01215 | MCK: Following error | 23360 | 5B40 | A | 2 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01216 | MCK: Positioning setting | 23359 | 5B3F | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01218 | PosFolger: Setting | 23357 | 5B3D | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01219 | MCK: Speed follower setting | 23356 | 5B3C | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01221 | MCK: Ref. mode | 23354 | 5B3A | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01222 | MCK: Ref. M limit mode 14/15 | 23353 | 5B39 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C01223 | MCK: Ref. waiting time mode 14/15 | 23352 | 5B38 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01224 | MCK: Ref. speeds | 23351 | 5B37 | A | 2 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01225 | MCK: Ref. accelerations | 23350 | 5B36 | A | 2 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01226 | MCK: Ref. S-ramp time | 23349 | 5B35 | A | 1 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01227 | MCK: Ref. positions | 23348 | 5B34 | A | 2 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01228 | MCK: Ref. sequence profile | 23347 | 5B33 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01229 | MCK: Position limiting values | 23346 | 5B32 | A | 2 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01230 | Manual jog: Setting | 23345 | 5B31 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01231 | MCK: Manual jog speeds | 23344 | 5B30 | A | 2 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01232 | MCK: Manual jog accelerations | 23343 | 5B2F | A | 2 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01233 | MCK: Manual jog S-ramp time | 23342 | 5B2E | A | 1 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01234 | MCK: Manual jog breakpoints | 23341 | 5B2D | A | 4 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01235 | MCK: Manual jog waiting times | 23340 | 5B2C | A | 1 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01236 | MCK: Speed follower | 23339 | 5B2B | A | 1 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01237 | MCK: Acceleration follower | 23338 | 5B2A | A | 2 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01238 | MCK: S-ramp time follower | 23337 | 5B29 | A | 1 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01239 | Reserved | 23336 | 5B28 | E | 1 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C01240 | MCK: Control word | 23335 | 5B27 | E | 1 | UNSIGNED_32 | 1 | ☑ | | |
| C01241 | MCK: Status word | 23334 | 5B26 | E | 1 | UNSIGNED_32 | 1 | ☑ | | |
| C01242 | MCK: Current pos profile number | 23333 | 5B25 | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C01243 | MCK: Current operating mode | 23332 | 5B24 | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C01244 | MCK: Target detection - times | 23331 | 5B23 | A | 3 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01245 | MCK: Target detection - positions | 23330 | 5B22 | A | 3 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01246 | MCK: Select signal source | 23329 | 5B21 | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01251 | MCK: Acceleration stop | 23324 | 5B1C | A | 1 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01252 | MCK: S-ramp times stop | 23323 | 5B1B | A | 1 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01295 | L_MCKStateInterface_1: Pos. selection | 23280 | 5AF0 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01296 | Mode: Position calculation | 23279 | 5AEF | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01297 | L_MckStateInterface_1: Alternative function | 23278 | 5AEE | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01298 | MCK: Operating mode change with profile no. | 23277 | 5AED | A | 4 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01299 | MCKI: Status MCKInterface | 23276 | 5AEC | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C01300 | Profile data: Positioning mode | 23275 | 5AEB | A | 15 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01301 | Profile data: Position | 23274 | 5AEA | A | 15 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01302 | Profile data: Speed | 23273 | 5AE9 | A | 15 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01303 | Profile data: Acceleration | 23272 | 5AE8 | A | 15 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01304 | Profile data: Deceleration | 23271 | 5AE7 | A | 15 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01305 | Profile data: Final speed | 23270 | 5AE6 | A | 15 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C01306 | Profile data: S-ramp time | 23269 | 5AE5 | A | 15 | UNSIGNED_16 | 1000 | ☑ | ☑ | |
| C01307 | Profile data: Sequence profile | 23268 | 5AE4 | A | 15 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01308 | Profile data: TP profile | 23267 | 5AE3 | A | 15 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01309 | Profile data: TP signal source | 23266 | 5AE2 | A | 15 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01501 | Resp. to communication error with MCI | 23074 | 5A22 | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C01670 | L_ComparePhi 1-5: Function | 22905 | 5979 | A | 5 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01671 | L_ComparePhi 1-5: Hysteresis | 22904 | 5978 | A | 5 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01672 | L_ComparePhi 1-5: Window | 22903 | 5977 | A | 5 | INTEGER_32 | 1 | ☑ | ☑ | |
| C01770 | Filter time - earth-fault detect. is running | 22805 | 5915 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01902 | Diagnostics X6: Max. baud rate | 22673 | 5891 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C01903 | Diagnostics X6: Change baud rate | 22672 | 5890 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C01905 | Diagnostics X6: Current baud rate | 22670 | 588E | E | 1 | UNSIGNED_32 | 1 | ☑ | | |
| C02430 | Axis bus address and no. of nodes | 22145 | 5681 | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02431 | Axis bus time settings | 22144 | 5680 | A | 3 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C02435 | Axis bus status | 22140 | 567C | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C02436 | Axis bus error status | 22139 | 567B | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C02437 | Axis bus MessageError | 22138 | 567A | E | 1 | UNSIGNED_8 | 1 | ☑ | | |
| C02440 | AxisBusIO slave/master | 22135 | 5677 | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02580 | Holding brake: Operating mode | 21995 | 55EB | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02581 | Holding brake: Speed thresholds | 21994 | 55EA | A | 5 | INTEGER_16 | 100 | ☑ | ☑ | |
| C02582 | Holding brake: Setting | 21993 | 55E9 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02589 | Holding brake: Time system | 21986 | 55E2 | A | 4 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C02593 | Holding brake: Activation time | 21982 | 55DE | A | 4 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C02607 | Holding brake: Status | 21968 | 55D0 | E | 1 | UNSIGNED_16 | 1 | ☑ | | |
| C02610 | MCK: Accel./decel. times | 21965 | 55CD | A | 3 | UNSIGNED_32 | 1000 | ☑ | ☑ | |
| C02611 | MCK: Limitations | 21964 | 55CC | A | 4 | INTEGER_16 | 100 | ☑ | ☑ | |
| C02652 | Memorise positions at power-off | 21923 | 55A3 | E | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C02810 | TP: Edge selection | 21765 | 5505 | A | 7 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02811 | TP: Sensor delay | 21764 | 5504 | A | 7 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C02812 | TP: Position offset | 21763 | 5503 | A | 7 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C02813 | TP: Pos window start | 21762 | 5502 | A | 3 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C02814 | TP: Pos window end | 21761 | 5501 | A | 3 | INTEGER_32 | 10000 | ☑ | ☑ | |
| C02815 | TP: Position source | 21760 | 5500 | A | 7 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02816 | TP: Signal counter | 21759 | 54FF | A | 7 | UNSIGNED_16 | 1 | ☑ | | |
| C02817 | TP: TouchProbe position | 21758 | 54FE | A | 7 | INTEGER_32 | 10000 | ☑ | | |
| C02830 | DIx: Debounce time | 21745 | 54F1 | A | 7 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02840 | CountInx: Parameter | 21735 | 54E7 | A | 4 | UNSIGNED_32 | 1 | ☑ | ☑ | |
| C02841 | CountInx: Counter content | 21734 | 54E6 | A | 2 | UNSIGNED_32 | 1 | ☑ | | |
| C02842 | FreqInxx: Offset | 21733 | 54E5 | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C02843 | FreqInxx: Gain | 21732 | 54E4 | A | 2 | INTEGER_16 | 100 | ☑ | ☑ | |
| C02844 | FreqIn12: Function | 21731 | 54E3 | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02845 | FreqIn12: PosIn comparison value | 21730 | 54E2 | E | 1 | INTEGER_32 | 1 | ☑ | ☑ | |
| C02853 | PSM: Lss saturation characteristic | 21722 | 54DA | A | 17 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02855 | PSM: Imax Lss saturation characteristic | 21720 | 54D8 | E | 1 | UNSIGNED_16 | 10 | ☑ | ☑ | |
| C02859 | PSM: Activate Ppp saturation char. | 21716 | 54D4 | E | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02862 | Resolver gain | 21713 | 54D1 | A | 2 | UNSIGNED_16 | 100 | ☑ | ☑ | |
| C02863 | Resolver: Phase error | 21712 | 54D0 | E | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C02866 | MCTRL: Special settings | 21709 | 54CD | A | 2 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C02867 | Identification process | 21708 | 54CC | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | ☑ |
| C02870 | PLI without motion: Optimisation factor | 21705 | 54C9 | A | 1 | INTEGER_16 | 100 | ☑ | | |
| C02871 | PLI without motion: Running time | 21704 | 54C8 | A | 1 | INTEGER_16 | 100 | ☑ | | |
| C02872 | PLI without motion: Adaptation of time duration | 21703 | 54C7 | A | 1 | INTEGER_8 | 1 | ☑ | ☑ | |
| C02873 | PLI without motion: Ident. el. rotor displ. angle | 21702 | 54C6 | A | 1 | INTEGER_16 | 1 | ☑ | | |

| Code | Name | Index | | Data | | | | Access | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | dec | hex | DS | DA | DT | Factor | R | W | CINH |
| C02874 | PLI without motion | 21701 | 54C5 | A | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C02875 | PLI without motion: Adaptation of ident angle | 21700 | 54C4 | A | 1 | INTEGER_8 | 1 | ☑ | ☑ | |
| C02876 | PSM: Max. motor temperature | 21699 | 54C3 | A | 1 | UNSIGNED_8 | 1 | ☑ | ☑ | |
| C02877 | PSM temperature coefficient | 21698 | 54C2 | A | 1 | INTEGER_16 | 100 | ☑ | ☑ | |
| C02878 | KTY motor temperature compensation | 21697 | 54C1 | A | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |
| C02879 | Slip calculation from equivalent circuit diagram | 21696 | 54C0 | A | 1 | UNSIGNED_16 | 1 | ☑ | ☑ | |

# 17 Working with the FB Editor

The function block editor (in the following called "FB Editor") is available in the »Engineer« from the "StateLine" device version.

The FB Editor serves, for example, to:

- execute an online monitoring of the technology application running in the device (e.g. for diagnostic purposes).
- reconfigure the I/O interconnection of the technology application.
- implement an individual drive solution (from the "HighLine" version).

**Note!**

The illustrations of the FB Editor user interface and the dialog boxes in this documentation are based on the »Engineer« V2.10.

## 17.1 Basics

Using the function block interconnection, any signal interconnection can be implemented. Various FBs are available for digital signal processing, signal conversion and logic modules.

For special tasks it has proved of value to use the integrated technology applications as a basis for modifications or extensions of the available FB interconnections. Moreover, from the HighLine device version experienced users are offered the opportunity to implement their own drive solutions independent of the predefined technology applications by using so-called "free interconnections".

For this purpose, the FB Editor provides the following functions:

- Copying & pasting of interconnection elements (also device-independent)
- Export & import of the interconnection
- Comparison of two interconnections (also online <-> offline comparison)
- Overview window and zoom functions
- Comments on the signal flow
- Online monitoring

The option to mask out non-used inputs and outputs of modules is also sensible to minimise the complexity of the FB interconnection and to adapt the clarity of the interconnection to the customers need.

All graphical information of the FB interconnection view (positions of the FBs, line or flag presentation of the connection, visibility of the inputs/outputs) are saved with the parameter set in the memory module of the controller and can be uploaded anytime into the FB Editor of the »Engineer« even if the Engineer project is not available.

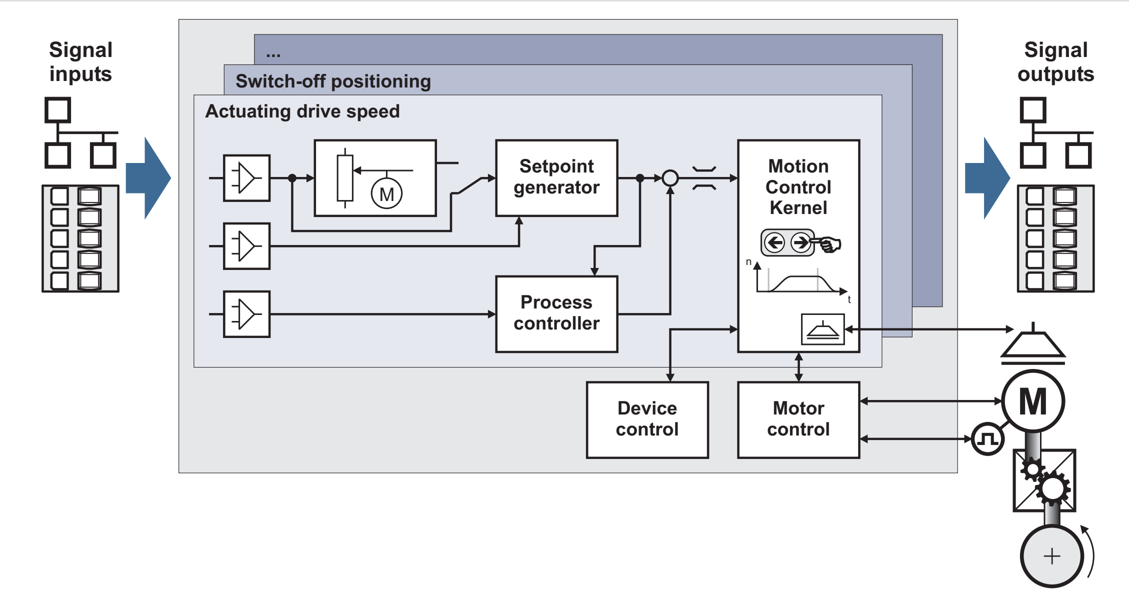
**Note!**

With the "StateLine" version, the interconnection shown in the application level cannot be edited.

### 17.1.1 Basic components of a drive solution

A drive solution consists of the following basic components:

- Signal inputs (for control and setpoint signals)
- Signal flow of the technology application
- Signal outputs (for status and actual value signals)




[17-1] Basic components of a drive solution

Regarding the 8400 device series, these three components are available for the FB interconnection and classified as follows:

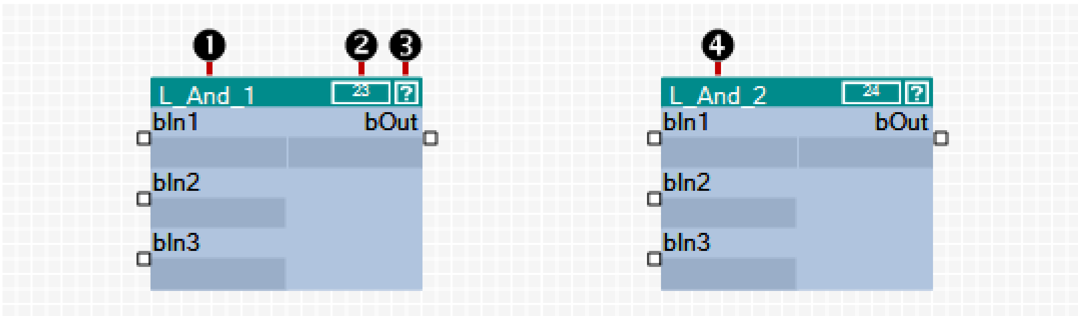
| Module type | Name | Task | Example |
|---|---|---|---|
| Function block | L_*name* | General function block for free interconnection (only HighLine) | L_Compare_1<br>L_PCTRL_1 |
| System block | LS_*name* | Signal interface to inverter-internal functions | LS_DigitalInput<br>LS_DriveInterface |
| Port block | LP_*name* | • Process data communication via a fieldbus using a communication module<br>• Process data communication via CAN on board | LP_CanIn1<br>LP_CanOut1<br>LP_MciIn<br>LP_MciOut |
| Application block | LA_*name* | Block for a technology application | LA_NCtrl<br>LA_SwitchPos |

Further information on the individual modules can be obtained from the following subchapters!

### 17.1.1.1 What is a function block?

A function block (FB) can be compared with an integrated circuit that contains a specific control logic and delivers one or several values when being executed.

- The function blocks are classified alphabetically in a "function library".
- Each function block has a unique identifier and a processing number which defines the position at which the function block is calculated during runtime.




❶ Name of the function block
❷ Processing number
❸ Button for opening the online help for the function block
❹ Another available function block of the same type (numbering in ascending order)

[17-2] Information on a function block in the FB Editor

**Tip!**

A detailed description of all available function blocks can be found in the main chapter "Function library". (📖 1110)

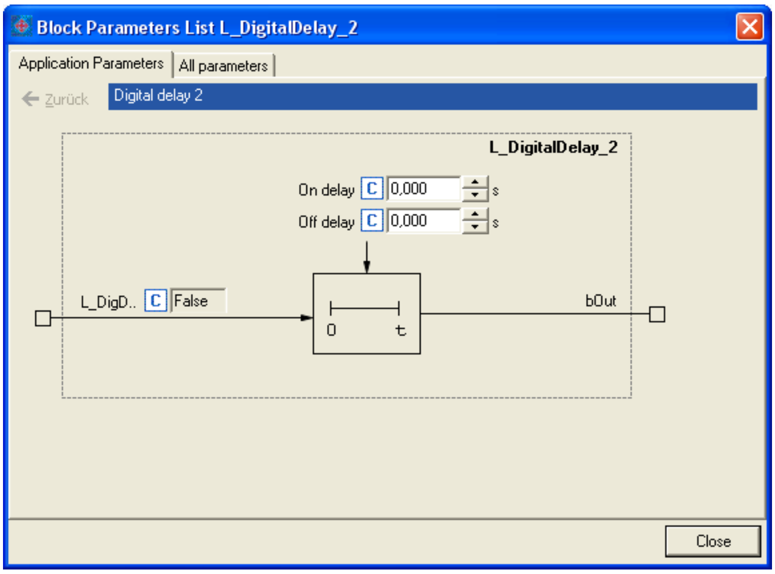
### 17.1.1.2 Parameterisable function blocks

Some function blocks have parameters which serve to change particular settings during operation, if required, or which display actual values & status information.

- The icon in the head of the module, a double-click on the module, or the **Parameter...** command in the *Context menu* of the module serve to open the parameterisation dialog or the parameter list for the module.

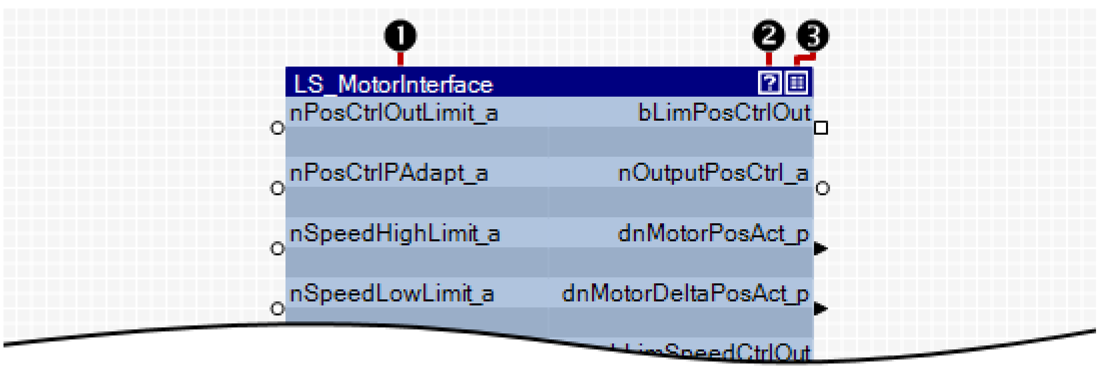
#### Example

Parameterisation dialog for the FB **L_DigitalDelay_2**:




### 17.1.1.3 What is a system block?

System blocks are a special variant of a function block. They partly activate real hardware, e. g. the digital and analog inputs/outputs and the motor control.




❶ Name of the system block
❷ Button for opening the online help for the system block
❸ Button for opening the parameterisation dialog or the parameter list for the system block

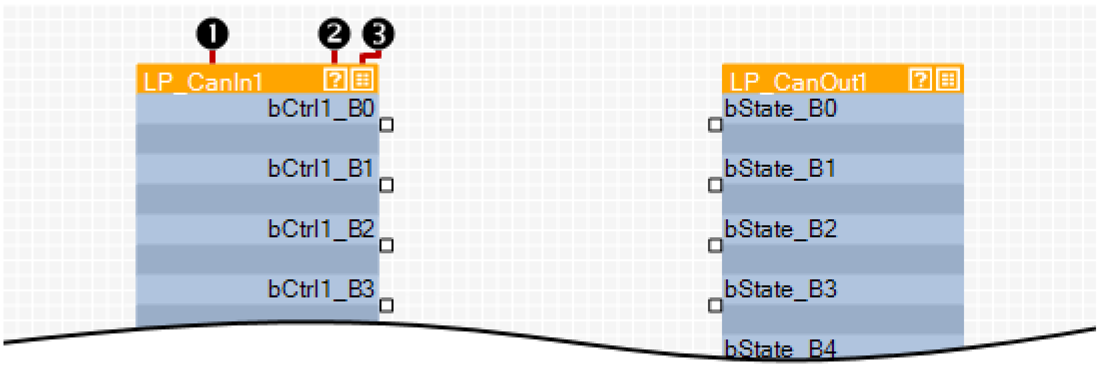
[17-3] Example: System block "LS_MotorInterface" for mapping the motor control

### 17.1.1.4 What is a port block?

A port block is a signal interface to a fieldbus. Input/output ports represent the input and output process data of the fieldbus.

- Port blocks LP_CanIn/LP_CanOut: Signal interface to the CAN bus
- Port blocks LP_MciIn/LP_MciOut: Signal interface to a plugged fieldbus module

If, for instance, the controller is to be controlled via CAN bus or a fieldbus module, the input/output ports are connected to the application block (device-internal signal processing) in the I/O level of the FB Editor.




❶ Name of the port block
❷ Button for opening the online help for the port block
❸ Button for opening the parameterisation dialog or the parameter list for the port block

[17-4] Example: Input port "LP_CanIn1" and output port "LP_CanOut1"

### 17.1.1.5 What is an application block?

The application/technology function set in C00005 is shown as application block in the I/O level of the FB Editor.

The application block comprises the signal flow processing generated via function block interconnection for the selected application in each case (e.g. "actuating drive speed" or "switch-off positioning"). The function block interconnection is shown in detail on the application level.

### 17.1.2 Conventions used for input/output identifiers

This chapter describes the conventions used for the identifiers of the inputs/outputs of the blocks. The conventions ensure a uniform and consistent terminology and make reading and comprehending the interconnection and application easier.

**Tip!**

The conventions used by Lenze are based on the "Hungarian Notation". This ensures that the most significant characteristics of the corresponding input/output (e.g. the data type) can be instantly recognised from its identifier.

An identifier consists of

- a data type entry
- an identifier (the "proper" name of the input/output)
- an (optional) signal type specification

#### Data type entry

The data type entry provides information about the data type of the corresponding input/output:

| Data type entry | Meaning | Resolution | Value range |
| --- | --- | --- | --- |
| b | BOOL | 1bit | 0 ≡ FALSE / 1 ≡ TRUE |
| dn | DINT | 32 bits | -2147483647 ... 2147483647 |
| n | INT | 16 bits | -32767 ... 32767 |
| w | WORD | 16 bits | 0 ... 65535 |

#### Identifier

The identifier is the proper name of the input/output and should indicate the application or function.

- Identifiers always start with a capital letter.
- If an identifier consists of several "words", then each "word" must start with a capital letter.
- All other letters are written in lower case.

### Signal type entry

In general, it is possible to assign a certain signal type to the inputs and outputs of the Lenze function blocks. There are e.g. digital, scaled, position, acceleration and speed signals.

- ▶ A corresponding ending (preceded by an underscore) is added to the identifier of the corresponding input/output to indicate the signal type.

| **Signal type entry** & port symbol in the FB Editor | | **Meaning** | **Resolution** | **Value range** |
|---|---|---|---|---|
| _a | ○ | Analog/scaled | 16 bits | ± 199.99 % |
| _v | ◁ / ▷ | Angular velocity | 16 bits | ± 30000.0 rpm |
| _p | ◀ / ▶ | Position | 32 bits | $-2^{31}$ ... $2^{31}$-1 increments |
| | □ | Digital (BOOL) | 8 bits | 0 ≡ FALSE; 1 ≡ TRUE |
| | ▬ | Other (WORD) | 16 bits | 0 ... 65535 |
| | ■ | Other (DINT) | 32 bits | -2147483647 ... 2147483647 |

## 17.1.3 Scaling of physical units

With regard to the parameter setting & configuration of the controller it is very helpful to know the signal types and their scaling listed in the following table, which are used to process physical values (e.g. an angular velocity or position) in the function block interconnection.

| **Signal type entry** & port symbol in the FB Editor | | **Meaning** | **Scaling** external value | ≡ internal value |
|---|---|---|---|---|
| _a | ○ | Analog/scaled | 100 % | ≡ $2^{14}$ ≡ 16384 |
| _v | ◁ / ▷ | Angular velocity | 15000 rpm | ≡ $2^{14}$ ≡ 16384 |
| _p | ◀ / ▶ | Position | 1 encoder revolution | ≡ $2^{16}$ increments |

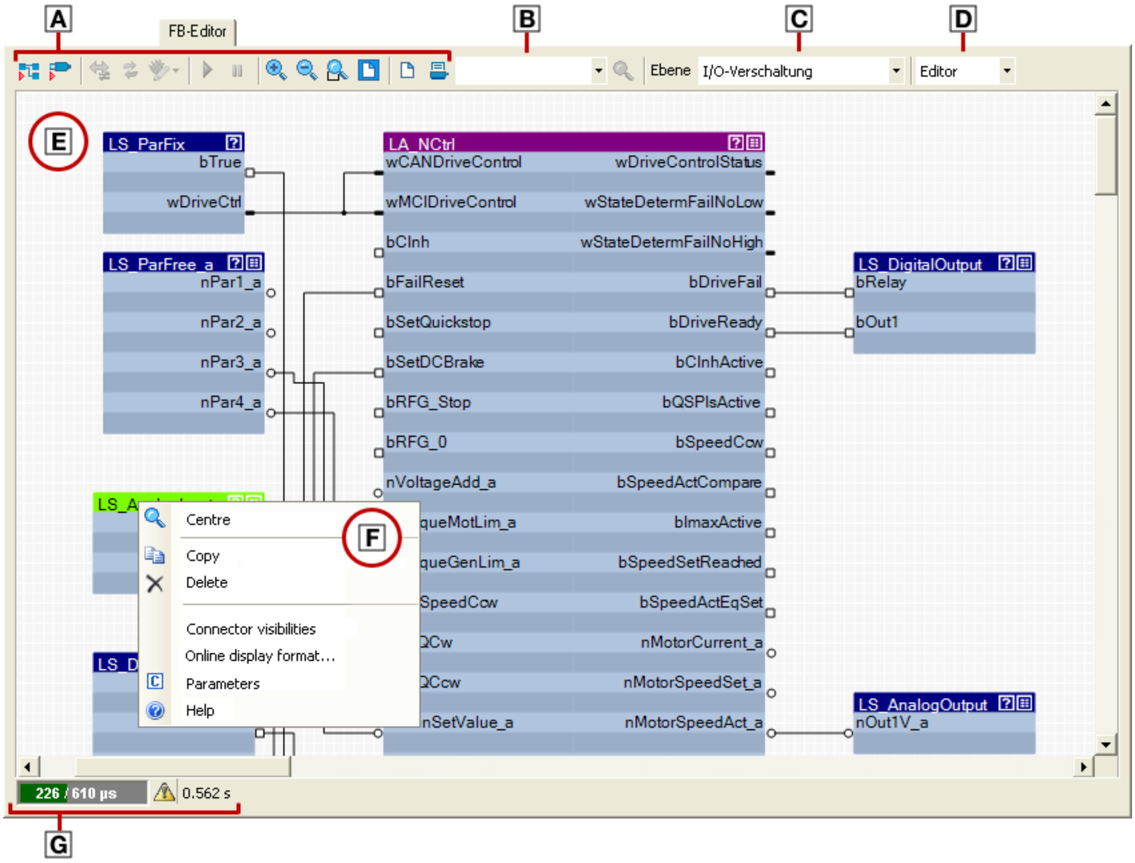
## 17.2 User interface

**How to access the FB Editor:**

1. Go to *Project View* and select the 8400 controller.
2. Go to *Workspace* and select the **FB Editor** tab.

   The FB Editor displays the wiring of the technology function selected in C00005. The interconnection of the I/Os of the controller depend on the control mode selected in C00007.

The user interface of the FB Editor includes the following control and function elements:




| | | | |
|---|---|---|---|
| A Toolbar | B Search function | C Level selection | D Editor view/overview |
| E Drawing area | F Context menu | G Status bar | Not shown: Overview window |

**Tip!**

Go to the *»Engineer« toolbar* and click the icon to hide the *Project View* and the *Message Window*. This increases the *Workspace* available for the FB Editor. A renewed click on the symbol shows the *Project View* and the *Message Window* again.

### 17.2.1 Toolbar

The FB Editor is provided with an individual toolbar in the upper position which in the following text is called *FB Editor toolbar*.

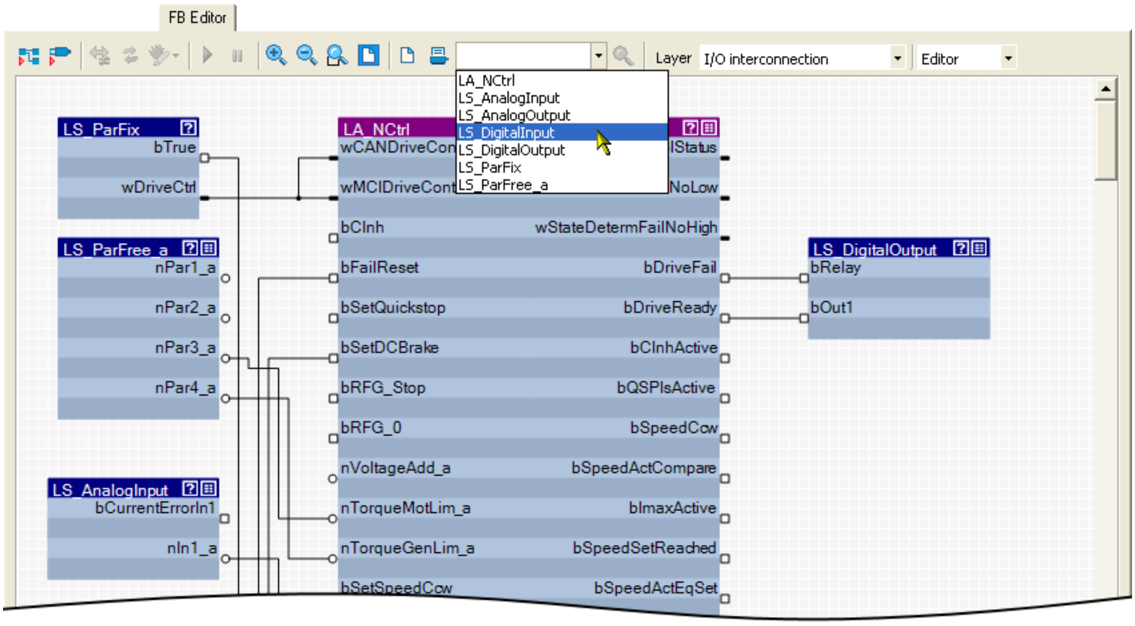
▶ Click an icon to execute the corresponding function.

| Symbol | Function |
|---|---|
| | Insert function block or system block<br>▸ Inserting a function block (🕮 1080)<br>▸ Inserting a system block (🕮 1082) |
| | Inserting a port block (🕮 1084) |
| | Adjusting online and offline interconnection (🕮 1103) |
| | Acknowledge error in the interconnection / reload interconnection |
| | Correct interconnection |
| | Start online monitoring |
| | Interrupt online monitoring |
| | Close online monitoring |
| | Enlarge view of interconnection |
| | Reduce view of interconnection |
| | Enlarge cutout of interconnection |
| | Show total interconnection in the drawing area |
| | Show print view |
| | Printing the interconnection (🕮 1104) |
| | Search function (🕮 1067) |

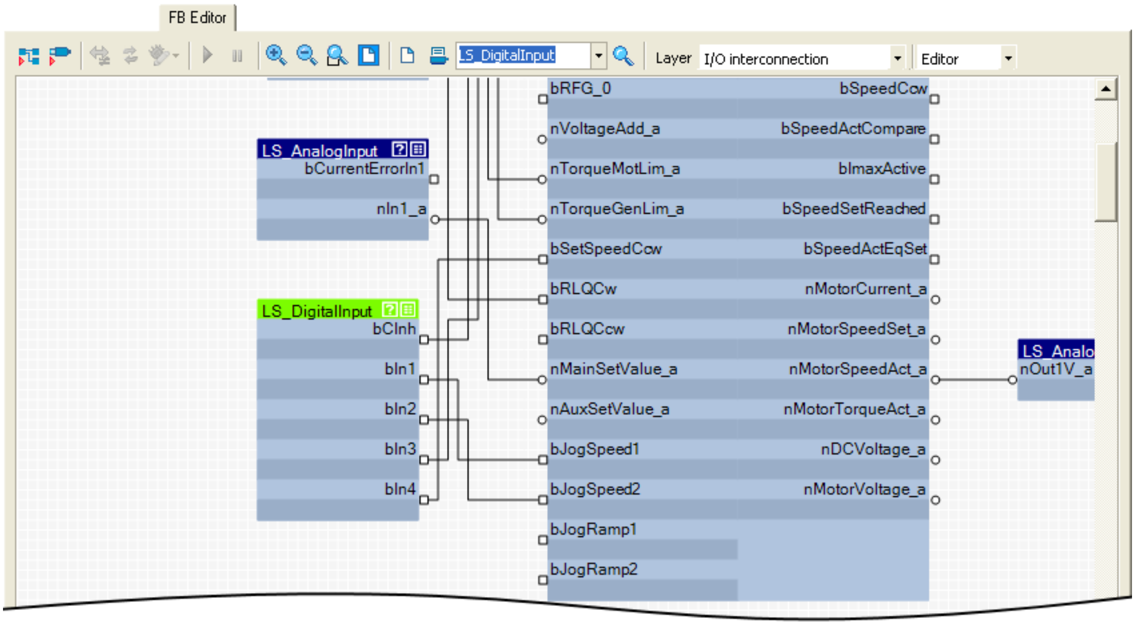
### 17.2.2 Search function

Use the search function to get quickly to a certain module of the interconnection.

▶ The list field of the search function contains all function blocks, system blocks, and port blocks of the interconnection:




▶ When you select a module in the list field, this module is zoomed in and selected at the same time (the following example shows the **LS_DigitalInput** system block):



**Tip!**

You can also enter any search text in the input field.

- If you click the icon, the cutout is moved to the object which contains this search text.
- Another click on the icon leads to a new search. Thus, you can navigate successively to all objects which contain the entered search text.
- The search text does not consider case sensitivity.

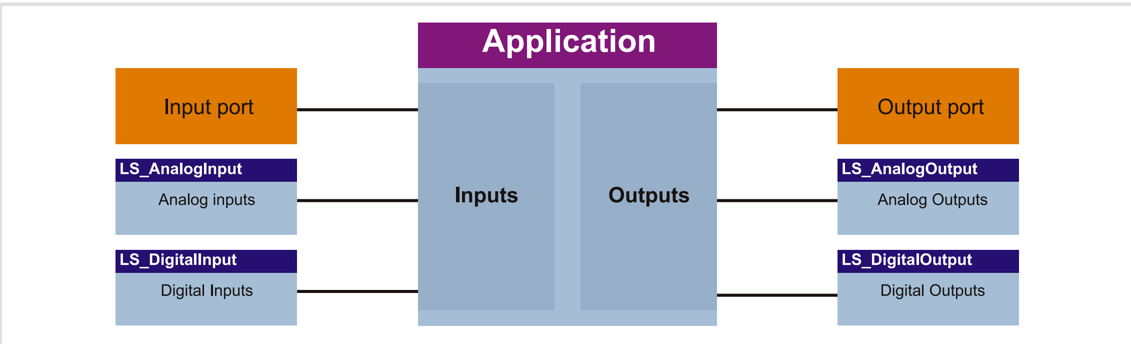
## 17.2.3 Level selection

Go to the **Level selection** list field and select the interconnection level to be displayed.

### "I/O interconnection" level

This level displays only the I/O interconnection of the currently selected technology application for a better overview.

- Details of the application are masked out in this level.
- The interconnection of the I/Os of the controller with the inputs and outputs of the application in detail depends on the control mode selected in C00007.
- The parameterisation dialogs on the **Application parameter** tab correspond to the application block displayed in this level.

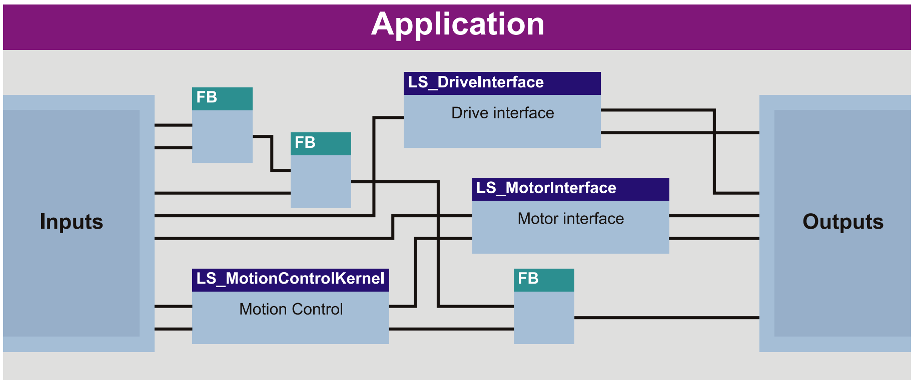



[17-5] Schematic diagram of "I/O interconnection"

### "Application interconnection" level

This level displays the interconnection of the application selected in C00005 in detail. All function blocks used in the application and the system blocks which provide the interfaces to the drive and motor interface and to the MotionControlKernel (MCK) are displayed with their connections.

▶ The interconnection of the I/Os of the controller with the inputs and outputs of the application is masked out in this level.




[17-6] Schematic diagram of "Application interconnection"

**Note!**

With the "StateLine" version, the interconnection shown in the application level cannot be edited.

**Tip!**

Every application block features so-called "free inputs and outputs" which you can use to transfer signals from the I/O level to the application level and vice versa.

- In the Lenze setting, these connectors are hidden in the function block editor.
- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

#### "Free interconnection" level

This level serves to implement an individual drive solution for the "HighLine" version.

**Note!**

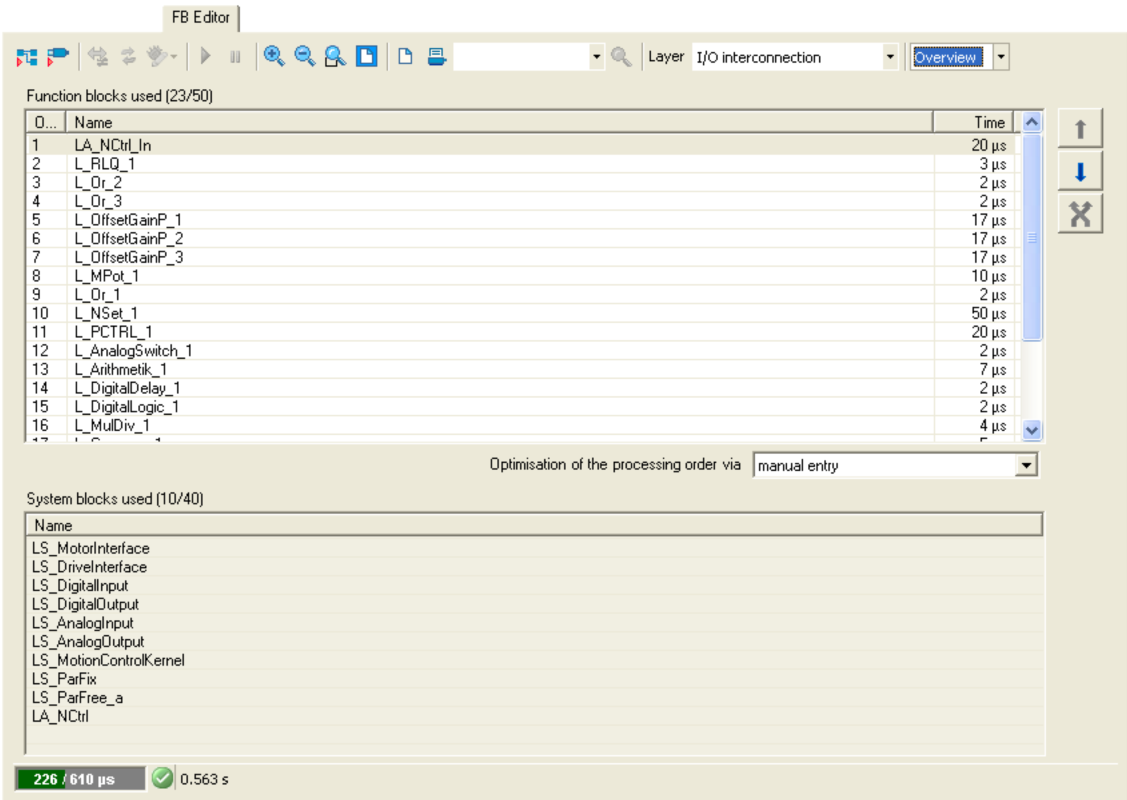
When you select the "Free interconnection" level for the first time, you are prompted to confirm whether the interconnection from the I/O level and the application level are to be combined and copied into this level.

When you confirm this confirmation prompt with **Yes**, the I/O level and the application level are not available anymore. This action can only be undone by resetting the application to a predefined Lenze application! ▸ Resetting changed interconnection (📖 1102)

### 17.2.4 Editor view/overview

Use the list field at the top right to change from the Editor to the overview and vice versa.

The overview shows all function blocks used of the interconnection in the upper list field in the order of their processing. The lower list field shows all used system blocks.

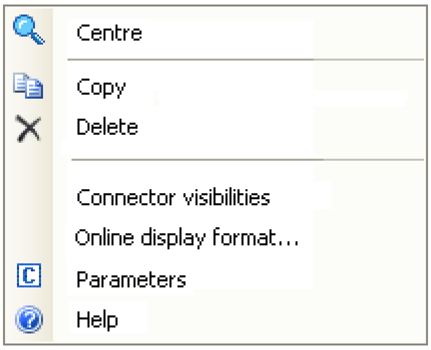



- The processing order of the function blocks can be optimised manually or according to an automatically generated selection. ▸ Changing the processing order (📖 1097)

### 17.2.5 Context menu

You can open a *context menu* via the right mouse button for each object (function block, system block, line, comment, etc.) and for the drawing area:

- The contents of the *context menu* depend on the type of object you click on.
- Example: *Context menu* for a function block:




### 17.2.6 Status bar

The status bar of the FB Editor shows, among other things, information about the system load and the error status of the interconnection:


| Symbol | Meaning |
|---|---|
| **A System load** | |
|  | Here: out of the available computing time of 610 µs, 226 µs are required by the application. |
| **B Error status of the interconnection** | |
|  | The interconnection has no errors and no warnings |
|  | The interconnection has errors and/or warnings |
| **C Communication status** | |
|  | Offline |
|  | Online |
|  | Communication error |
| **D Adjustment status** | |
| = | Offline and online interconnection match |
| ≠ | Offline and online interconnection are different |
| **E Update rate for monitoring values** | |

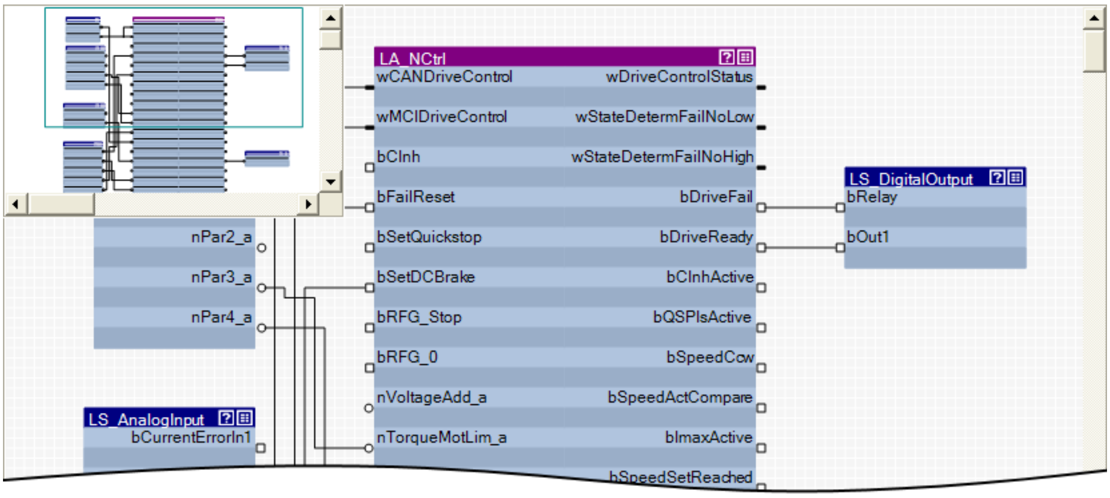
### 17.2.7 Overview window

The overview window shows the drawing area in a reduced view. The overview window serves to e.g. move quickly through a more complex interconnection.

**How to show the monitor window:**

Go to the *Context Menu* of the drawing area and select the **Overview Window**.

- If you execute this command again, the overview window is hidden again.




▶ The green frame in the overview window indicates the interconnection cutout that is currently displayed in the drawing area.

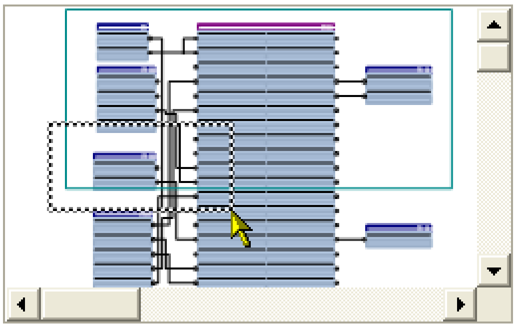
▶ Use the mouse pointer to shift and resize the cutout to be displayed.

**How to shift the cutout presented in the drawing area:**

1. Position the mouse pointer to the green frame in the overview window.
   - The mouse pointer symbol becomes a positioning cross.
2. Click left mouse button and shift the green frame to its new position by keeping the mouse button pressed, so that the desired cutout of the interconnection is displayed in the drawing area.

**How to redefine the cutout to be presented:**

In the overview window draw a frame around the area of the interconnection which is to be presented in the drawing window by keeping the left mouse button pressed:



- The aspect ratio of the frame is automatically adapted to the aspect ratio of the drawing area.
- According to the size of the frame that is drawn, also the presentation size of the objects in the drawing area changes.

**Tip!**

Go to the *FB Editor toolbar* and click the icon to adapt the view size so that all objects included in the interconnection are visible in the drawing area.

**Automatic scroll ("AutoScroll function")**
If you reach a window limitation in the drawing area when shifting an object or in the overview window when shifting the green frame, and if you then shortly hold the mouse pointer in this position, an automatic scrolling into the corresponding direction is carried out:

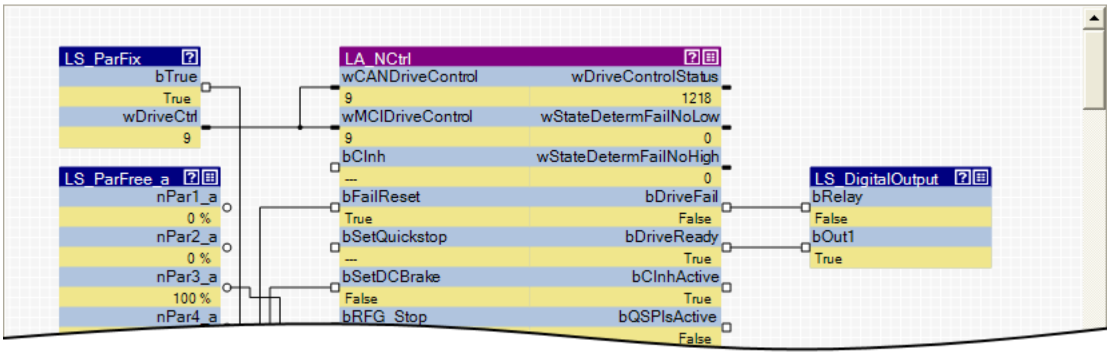
## 17.3 Using the FB Editor as "Viewer"

The main purpose of the FB Editor is the individual configuration of the selected technology application. However, you can also use the FB Editor to

- make a diagnosis of the application (when an online connection has been established),
- get a better understanding for the operating mode of the application,
- use the interconnection as an alternative parameterisation access.

### Diagnostics of the application

When an online connection to the controller has been established, the current values are displayed at the inputs and outputs of the objects.




- Process-scaled signals can be scaled in a "user-defined" way for easy diagnostics in the FB Editor. Change online display format (1077)

### Getting a better understanding for the operating mode of the application

Make yourself familiar with the signal flow of the interconnection to get a better understanding of the operating mode of the application or individual functional areas.

- The ? symbol in the head of the block or the **Help** command in the *context menu* for the block serve to open the online help for the block.

### Using the interconnection as an alternative parameterisation access

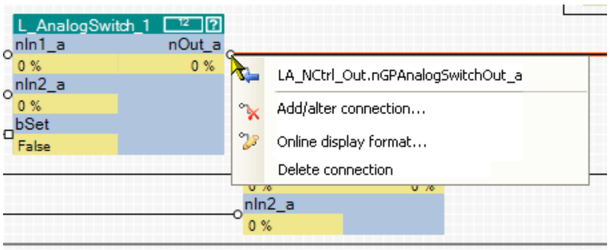
- The ▤ icon in the head of the module, a double-click on the module, or the **Parameter...** command in the *Context menu* of the module serve to open the parameterisation dialog or the parameter list for the module.

### 17.3.1 Following connections of inputs and outputs

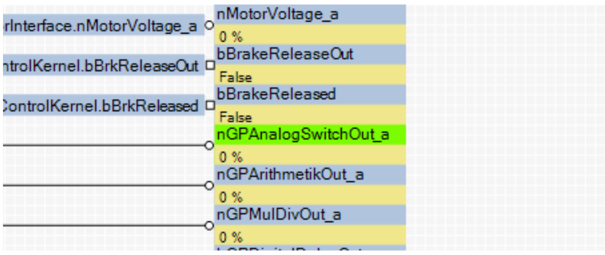
In addition to the Search function you can use the *context menu* of inputs and outputs to follow connections and quickly reach certain signals.

**How to navigate from one output to another connected input:**

1. Open the *context menu* (right mouse button) of the port symbol at the output.
   - The *context menu* for the port symbol contains all inputs which are connected to the output:



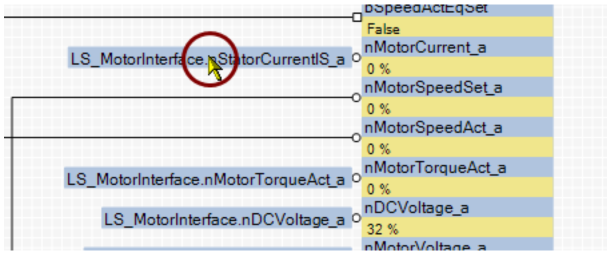

2. Select input in the *context menu* to which you want to navigate.
   - As a result, the selected input is displayed in the centre of the drawing area (in this example: nGPAnalogSwitchOut_a):



**How to navigate from one input to another connected output:**

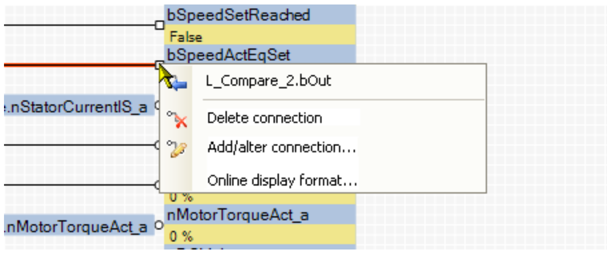
If the input is connected to a flag:

- Double-click the flag:




If the input is connected to a line:

1. Open the *context menu* (right mouse button) of the port symbol at the output:




2. Select output in the *context menu*.
   - Since an output can only be connected to an input, the *context menu* contains only an output.

The output is displayed in the centre of the drawing area.

## 17.3.2 Keyboard commands for navigation

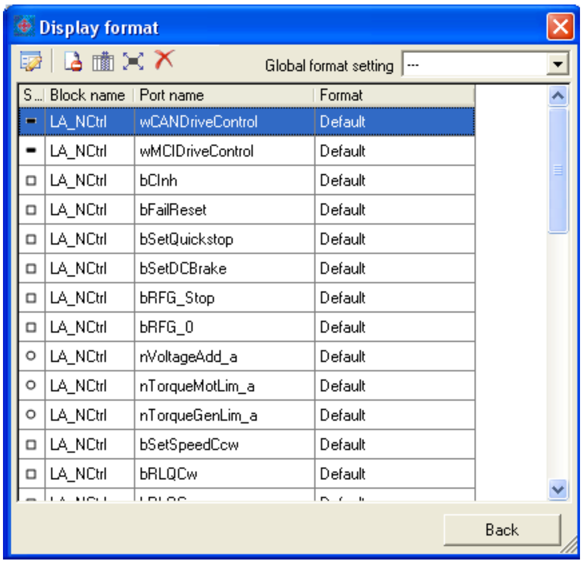
| Keyboard command | Function |
|---|---|
| **<Picture ▲ >** | Scroll up |
| **<Picture ▼ >** | Scroll down |
| **<Shift> + <picture ▲ >** | Scroll to the left |
| **<Shift> + <picture ▼ >** | Scroll to the right |
| **<POS1>** | Scroll to the left edge of the interconnection |
| **<END>** | Scroll to the right edge of the interconnection |
| **<Ctrl> + <Pos1>** | Scroll to the left upper corner of the interconnection |
| **<Ctrl> + <End>** | Scroll to the right lower corner of the interconnection |

### 17.3.3 Change online display format

For online monitoring in the FB Editor the display format of the input and output data of a block can be adapted individually. Process-scaled signals can be scaled in a "user-defined" way for easy diagnostics in the FB Editor. Thus, the display of these signals gets a process reference.

**How to change the data display format of block inputs/outputs:**

1. Go to the *context menu* of the block and select the **Online display format** command.
   - **Tip**: You can call the *context menu* of a block by clicking with the right mouse button on the header of the block.
   - The *Display format* dialog box is displayed:




2. Select the inputs/outputs from the list the display format of which is to be changed.
   - **Note**: In the **Global format setting** list field the "---" entry must be selected so that the display format can be changed.
   - If you click further inputs/outputs while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - The **<Shift>** key serves to select a related area of inputs/outputs.
   - More functions:

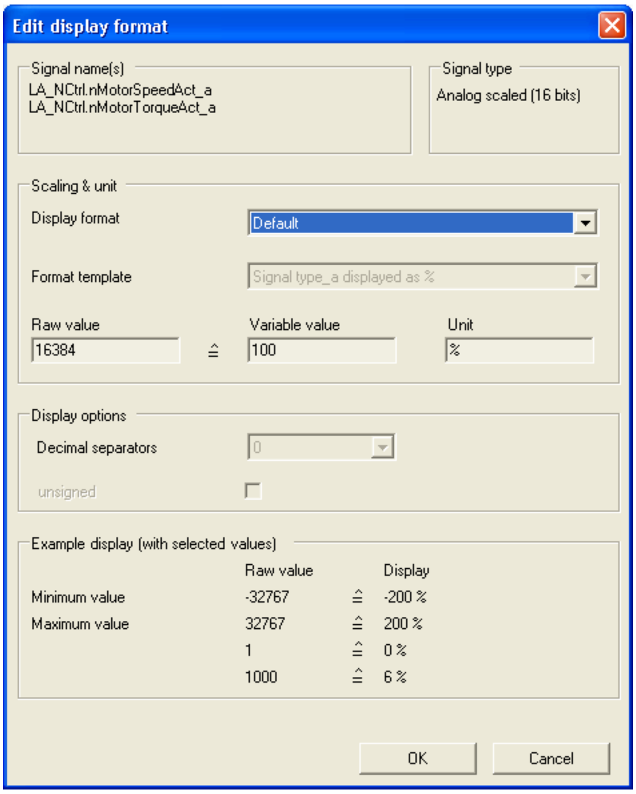
     Display masked out connections

     Display additional information

     Select all inputs/outputs

     Reset all format information

3. Click the symbol to edit the display format of the selected inputs/outputs.
   - The *Edit display format* dialog box is displayed:




4. Go to the **Display format** list field and select the "User-defined" entry.
5. Go to the **Format template** list field and select "No template".
6. Select the required scaling, unit, number of decimal positions, and sign handling.
7. Click **OK** to accept the settings and close the *Edit display format* dialog box.
   - The *Display format* dialog box now displays the text "User-defined" for the changed inputs/outputs in the **Format** column.

   After all required formats have been changed:

8. Click **Back** to close the *Display format* dialog box.
   - For online monitoring, the changed format is used.

## 17.4 Reconfiguring the predefined interconnection

How to proceed:

1. Insert additionally required objects into the interconnection.
2. Hide unneeded inputs/outputs of function blocks and system blocks to obtain a clearly arranged interconnection.
3. Arrange the objects in the drawing area in a reasonable manner.
4. Establish the connections required for the desired function.
5. If required, change (optimise) the processing order of the function blocks.

**Tip!**

Detailed information on the individual steps can be obtained from the following subchapters!

**Note!**

With the "StateLine" version, the interconnection shown in the application level cannot be edited.

### 17.4.1 Inserting/Deleting objects

Objects can be inserted in the interconnection via the *FB Editor toolbar* and the *context menu* of the drawing area. The following subchapters provide detailed information on how to insert/delete the different objects.

| Symbol | Function |
| --- | --- |
| | Inserting a function block (🕮 1080) |
| | Inserting a system block (🕮 1082) |
| | Inserting a port block (🕮 1084) |
| | Inserting a comment (🕮 1086) |

**Tip!**

Use the *context menu* of the drawing area to insert a function block, system block, port block or comment directly to the current position of the mouse pointer in the drawing area.

If you insert an object via the corresponding icon in the *FB Editor toolbar*, the object is always placed at the top left corner in the drawing area.

Interconnection elements cannot only be copied within the same interconnection but also across all devices within the same project, as long as the devices stem from the same product family. ▸ Copying interconnection elements (across all devices) (🕮 1099)

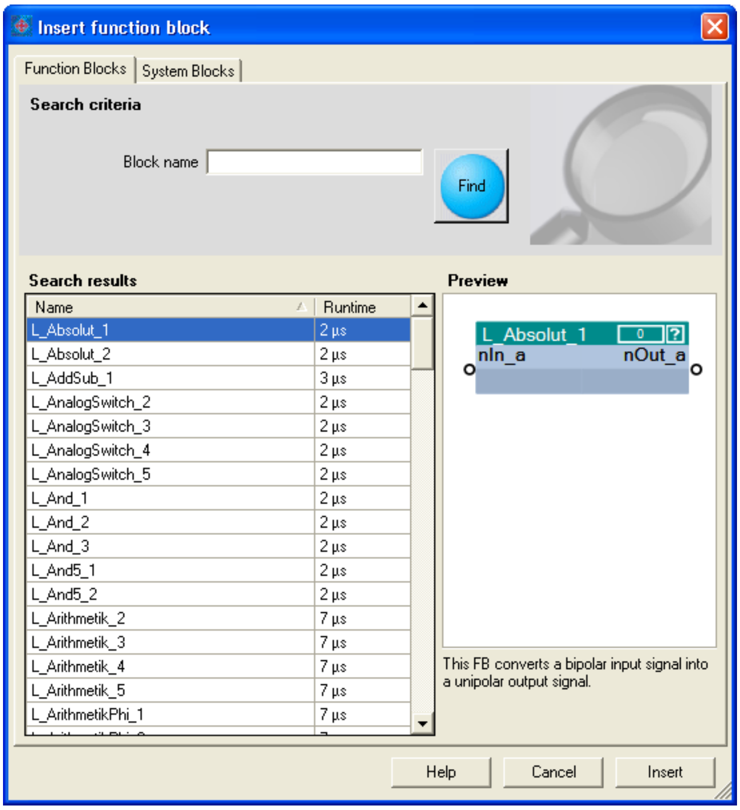
### 17.4.1.1 Inserting a function block

**Note!**

In the FB Editor, function blocks are only available in the "Application interconnection" level!

**How to insert a function block into the interconnection:**

1. In the *FB Editor toolbar*, click the icon.
   - The *Insert Function Block* dialog box appears:
2. Unless it is already displayed, select the **Function Blocks** tab.
   - All function blocks available are displayed in the **Search results** list field.




   - A preview of the selected function block is displayed.
   - A detailed description of all available function blocks can be found in the main chapter "Function library". (📖 1110)
3. If required, define **Search criteria** to narrow down the available function blocks:
   - **Block name**:
     String which must be contained in the name of the function block.
4. After changing the search criteria, press the **Find** button to update the selection.
   - Then, only the function blocks complying with the features set in the search criteria are shown in the **Search Results** list field.
   - If no search criteria are set, all function blocks available are shown.

5. Select the function block to be inserted in the **Search results** list field.
6. Press **Insert** button.
   - The dialog box is closed and the selected function block is inserted into the interconnection.

### Context menu for the function block

If you right-click on the header of a function block, a *context menu* opens via which you can execute the following functions in addition to the general processing functions (Copy, Insert, Delete):

| Command | | Function |
|---|---|---|
| | **Center** | Move the visible cutout of the drawing area so that the block is centred. |
| | **Connector visibilities...** | Define visible inputs and outputs of the block.<br>▶ Changing connector visibilities (📖 1089) |
| | **Online display format...** | Adapt the display format of the input and output data of the block individually for online monitoring.<br>▶ Change online display format (📖 1077) |
| | **Parameter...** | Open the parameter list/parameterisation dialog for the block.<br>• Only if function block is parameterisable. |
| | **Help** | Show online help for the block. |

### Related topics

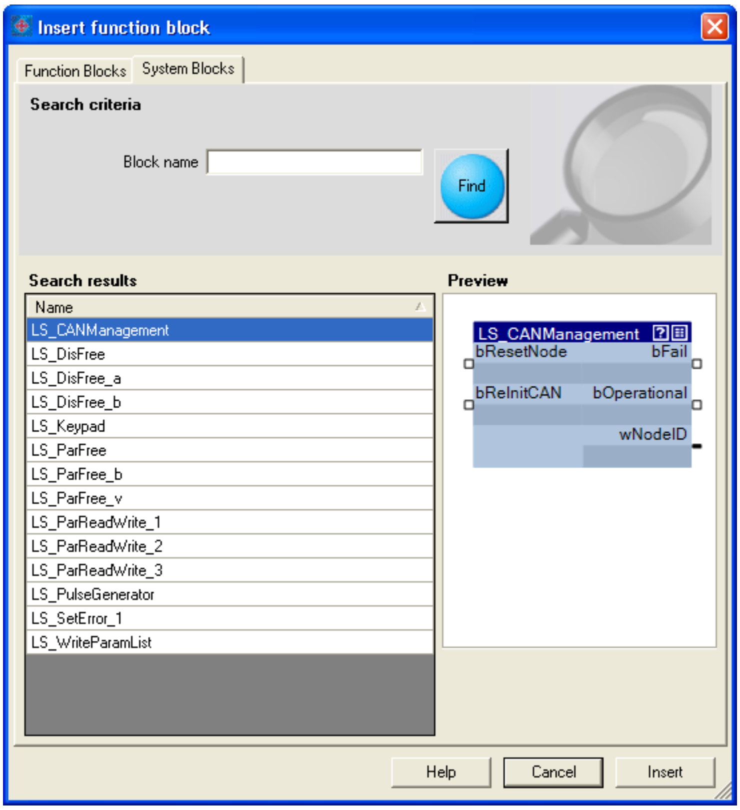
▶ Deleting objects that are no longer required (📖 1088)
▶ Changing connector visibilities (📖 1089)
▶ Arranging objects in the drawing area (📖 1090)
▶ Creating/deleting connections (📖 1091)
▶ Changing the processing order (📖 1097)

### 17.4.1.2 Inserting a system block

A system block is inserted similarly to the way a function block is inserted.

**How to insert a system block into the interconnection:**

1. In the *FB Editor toolbar*, click the icon.
   - The *Insert Function Block* dialog box appears:
2. Unless it is already displayed, select the **System Blocks** tab.
   - All system blocks available are displayed in the **Search results** list field.




   - A preview of the selected function block is displayed.
3. If required, define **Search criteria** to accordingly narrow down the system blocks available:
   - **Block name**:
     String which must be contained in the name of the system block.
4. After changing the search criteria, press the **Find** button to update the selection.
   - Then, only the system blocks complying with the features set in the search criteria are shown in the **Search Results** list field.
   - If no search criteria are set, all system blocks available are shown.

5. Select the system block to be inserted in the **Search results** list field.
6. Press **Insert** button.
   - The dialog box is closed and the selected system block is inserted into the interconnection.

### Context menu for the system block

If you right-click on the header of a system block, a *context menu* opens via which you can execute the following functions in addition to the general processing functions (Copy, Insert, Delete):

| Command | | Function |
|---|---|---|
| | **Center** | Move the visible cutout of the drawing area so that the block is centred. |
| | **Connector visibilities...** | Define visible inputs and outputs of the block.<br>▸ Changing connector visibilities (📖 1089) |
| | **Online display format...** | Adapt the display format of the input and output data of the block individually for online monitoring.<br>▸ Change online display format (📖 1077) |
| | **Parameter...** | Open the parameter list/parameterisation dialog for the block. |
| | **Help** | Show online help for the block. |

### Related topics

▸ Deleting objects that are no longer required (📖 1088)
▸ Changing connector visibilities (📖 1089)
▸ Arranging objects in the drawing area (📖 1090)
▸ Creating/deleting connections (📖 1091)

### 17.4.1.3 Inserting a port block

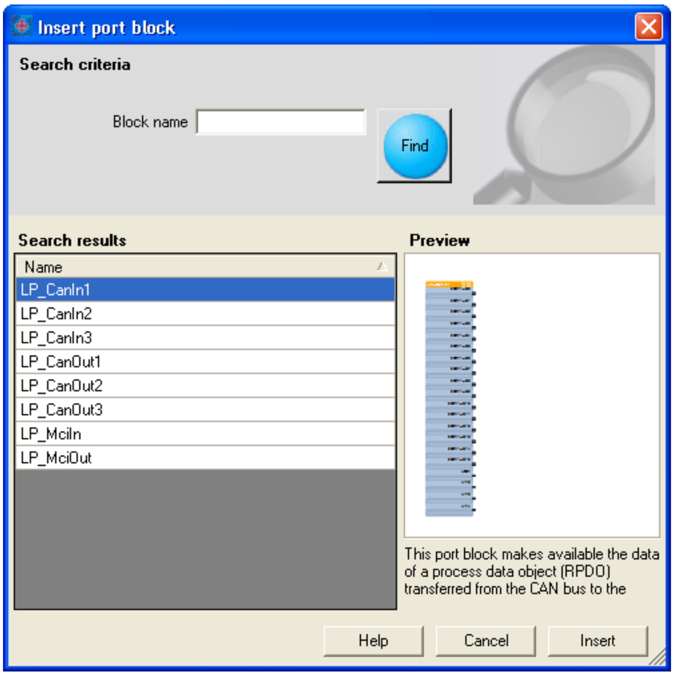
All input/output ports defined for the application on the **Ports** tab can be inserted into the interconnection in the form of port blocks in order to get access to the associated element variables.

**Tip!**

You can change between the **Ports** and **FB Editor** tabs at any time to define new ports and afterwards insert them into the interconnection.

**How to insert a port block into the interconnection:**

1. In the *FB Editor toolbar*, click the icon.
   - The *Insert port block* dialog box appears.
   - All port blocks available are displayed in the **Search results** list field.




   - A preview of the selected port block is displayed.
2. If required, define **search criteria** to accordingly narrow down the port blocks available:
   - **Block name**:
     String which must be contained in the name of the port block.
3. After changing the search criteria, press the **Find** button to update the selection.
   - Then, only the port blocks complying with the features set in the search criteria are shown in the **Search Results** list field.
   - If no search criteria are set, all port blocks available are shown.
4. Select the port block to be inserted in the **Search results** list field.

5. Press **Insert** button.
   - The dialog box is closed and the selected port block is inserted into the interconnection.

### Context menu for the port block

If you right-click on the header of a port block, a *context menu* opens via which you can execute the following functions in addition to the general processing functions (Copy, Insert, Delete):

| Command | | Function |
|---|---|---|
| | **Center** | Move the visible cutout of the drawing area so that the block is centred. |
| | **Connector visibilities...** | Define visible inputs and outputs of the block.<br>▸ Changing connector visibilities (📖 1089) |
| | **Online display format...** | Adapt the display format of the input and output data of the block individually for online monitoring.<br>▸ Change online display format (📖 1077) |
| | **Parameter...** | Open the parameter list/parameterisation dialog for the block. |
| | **Help** | Show online help for the block. |

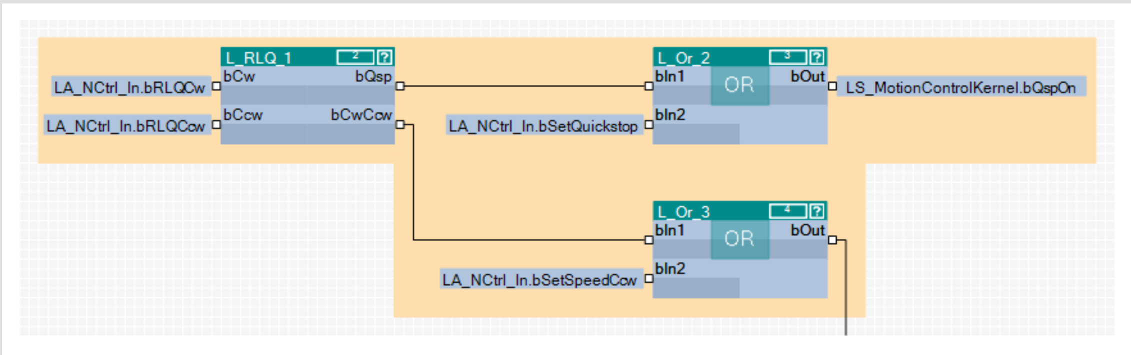
### Related topics

▸ Deleting objects that are no longer required (📖 1088)
▸ Changing connector visibilities (📖 1089)
▸ Arranging objects in the drawing area (📖 1090)
▸ Creating/deleting connections (📖 1091)

### 17.4.1.4 Inserting a comment

Comments can be inserted at any position in the drawing area.

As of the »Engineer« V2.10, the interior colour and text alignment of a comment can be changed via a properties dialog. Now the sizes of comments can also be changed using the mouse pointer. When using different interior colours you can use comments to graphically arrange areas that belong together in terms of function or separate them from other areas:



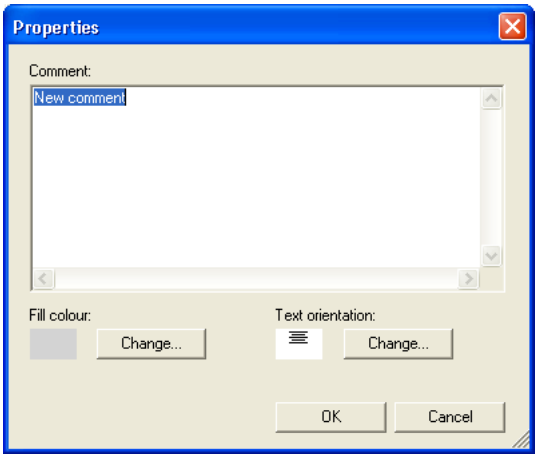

[17-7] Example: Graphical arrangement of FBs by means of two comments that overlap.

**Note!**

The term "Arrangement" does not mean a logical arrangement of the function blocks. The comments are only graphical presentation elements of the FB Editor.

**How to insert a new comment into the interconnection:**

1. Move the mouse pointer to the (free) position in the drawing area where the comment is to be inserted.
2. Go to the *Context menu* (right mouse key) and select the **New comment** command.
   - The *Properties* dialog box is displayed:




3. Enter the required comment into the text field.

4. Optional: Change preset interior colour.
   - For this purpose, click the left **Change...** button to open the *Colour* dialog box to select another interior colour.
5. Optional: Change preset text alignment.
   - For this purpose, click the right **Change...** button to open the *Text alignment* dialog box to select another text alignment.
6. Press **OK** to close the *Properties* dialog box and insert the comment.
   - After being inserted, the corner points of the comment are shown:


7. Optional: Change size of the comment.
   - For this purpose click one of the corner points with the left mouse button and enlarge the comment to the required size with the mouse button pressed.


8. Optional: Drag comment.
   - For this purpose click the comment with the left mouse button and move the comment to the required position with the mouse button pressed.

**Tip!**

The *Properties* dialog box for a comment already available can be opened by double-clicking the comment.

**Related topics**

- Deleting objects that are no longer required (1088)
- Arranging objects in the drawing area (1090)
- Creating/deleting connections (1091)

### 17.4.1.5 Deleting objects that are no longer required

Objects that are no longer required can be easily deleted again. "Delete" only means that the object is removed from the drawing area. If you have deleted an object from the drawing area, you can reinsert it any time into the interconnection.

**Note!**

Deleting an object cannot be undone.

Together with the object, all available connections to this object are deleted.

**How to delete objects that are no longer required:**

1. Select objects to be deleted.
   - You can select a single object by clicking the header of the object.
   - You can select objects that are placed together by drawing a frame around these objects while keeping the mouse button pressed.
   - If you click the header of further objects while pressing **<Ctrl>**, these will be added to an already existing selection (multi-selection).
   - All selected objects are highlighted by a light green header.
2. Press **<Del>**.

**Related topics**

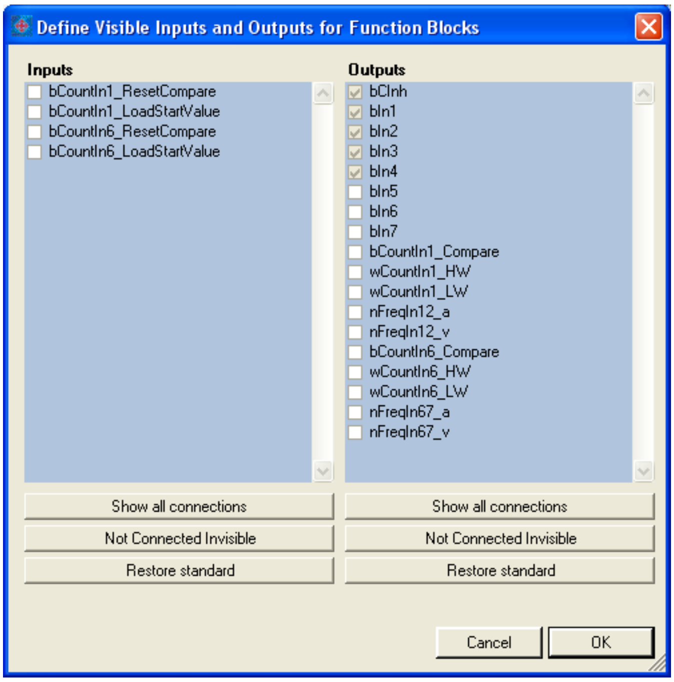
▸ Deleting connections that are no longer required (📖 1096)

### 17.4.2 Changing connector visibilities

Inputs and outputs that are not connected can be hidden for each block. This serves to reduce the dimension of the block. The interconnection becomes clearer.

**How to define the visible inputs and outputs:**

1. Go to the context menu of the block and select the **Connector visibilities** command.
   - The *Define Visible Inputs and Outputs for Function Blocks* is displayed:




   - All visible connections have a checkmark.
   - In case of a block that is inserted anew, all inputs and outputs are visible at first.
   - Inputs and outputs with a light grey checkbox are already connected and thus cannot be hidden.
2. By setting/removing the checkmarks or via the buttons you can define the visible inputs and outputs.
3. Press **OK** to accept the selected definition and close the dialog box.

### 17.4.3 Arranging objects in the drawing area

All objects can be freely arranged in the drawing area by dragging with the mouse.

We recommend to make an arrangement in which the required connections between the inputs and outputs can be created easily. A division into functional areas may also be sensible to get a better understanding of the application.

Objects which are already connected, can also be dragged to another (free) position in the drawing area. The available connections will be automatically re-routed after dragging.

**How to drag an object:**

1. Click the header of the object (and keep the button pressed).
2. Keep the button pressed and drag the object to the required position in the drawing area.
   - Via **<Esc>** you can cancel this action.

**How to drag several objects at the same time:**

1. Select the objects to be dragged.
   - You can select a single object by clicking the header of the object.
   - If you click the header of further objects while pressing **<Ctrl>**, these will be added to an already existing selection (multi-selection).
   - You can easily select objects that are placed together by drawing a frame around these objects while keeping the mouse button pressed.
   - All selected objects are highlighted by a light green header.
2. Keep the mouse button pressed on the header of one of the selected objects and drag it to the required position in the drawing area.
   - Via **<Esc>** you can cancel this action.

**Note!**

A red header indicates that the object overlaps with other objects in the drawing area!

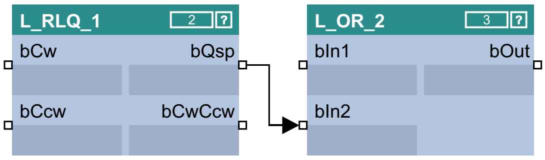
Arrange the objects so that no overlap occurs.

## 17.4.4 Creating/deleting connections

After adding objects and arranging them in a reasonable manner within the drawing area, you can create the connections between the available objects which are required for the desired function.

A connection always has a direction and therefore always has a source and a target.

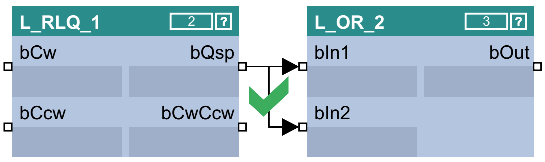
- An output represents a possible source in the interconnection.
- An input represents a possible target in the interconnection.




### Permissible/impermissible connections

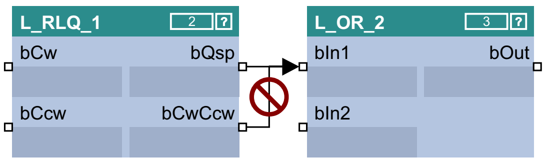
Several connections can lead from one output.

- Therefore it is always possible to start a new connection from an output.



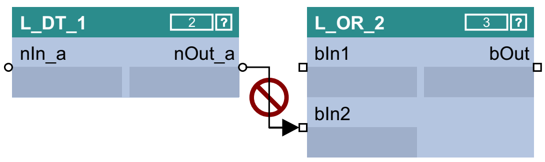

However, maximally one connection may end in an input.

- Therefore it is only possible to start a new connection from an input if there is no connection already ending in this input.



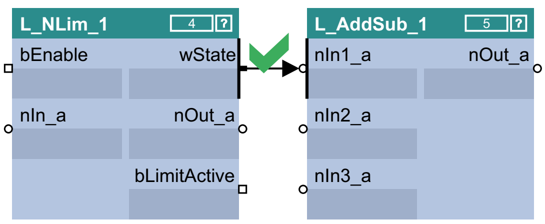

Only inputs/outputs of the same signal type can be connected.

- Thus, a connection between different port symbol cannot be established.




From the »Engineer« V2.12 "Analog/scaled" (_a) and "Miscellaneous (WORD)" signal types can also be interconnected.

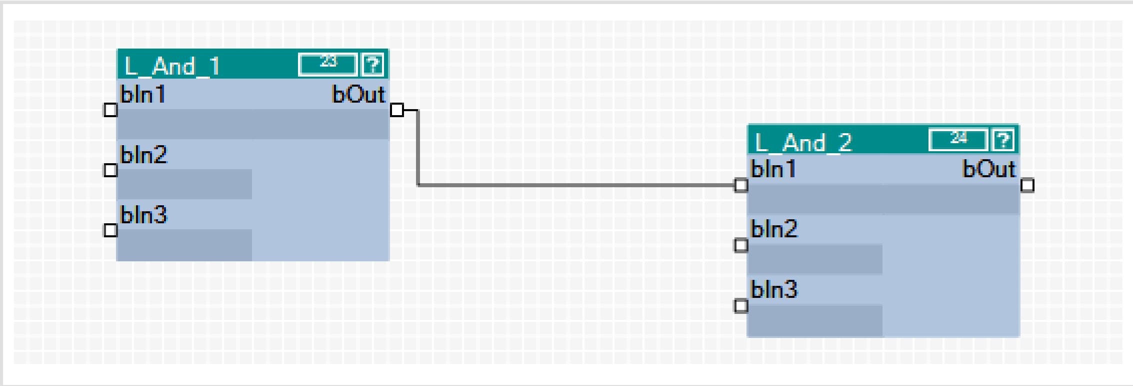
- The implicit type conversion is indicated by a vertical black bar at the port symbol.



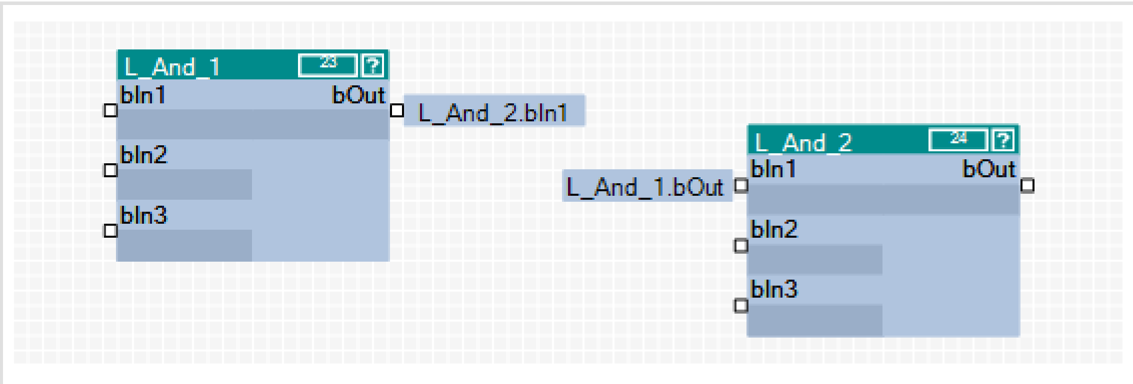

From the »Engineer« V2.13 "Analog/scaled" (_a) and "Angular velocity" (_v) signal types can also be interconnected.

#### Connection types

Connections can either be created by means of connection lines or port identifiers ("flags")




[17-8] Example 1: Connection via connection line




[17-9] Example 2: Connection via flags

**Tip!**

The commands **Show as flag** or **Show as line** in the *context menu* of a connection serve to change the representation of the connection at any time.

When an output is connected to several inputs via flags, three points are displayed ("...") at the output instead of the concrete input identifier. The *context menu* of the port symbol shows all inputs which are connected to the output.

#### 17.4.4.1 Creating a connection using the connection line

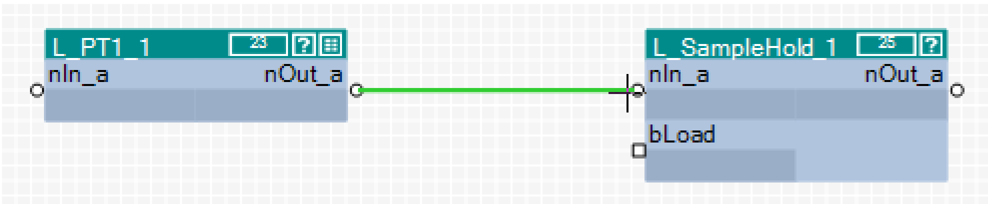
**How to create a connection using the connection line:**

1. Click the port symbol from which the new connection is to be started.
   - It is only possible to start a new connection from an input if there is no connection already ending in this input.
   - If you then move the mouse pointer away from the port symbol, a new connection is "drawn" from this port symbol.
   - Via **<Esc>** you can cancel this action.
2. Click the port symbol where the connection is to end.
   - Thereupon the corresponding connection is routed automatically if the connection is permissible.

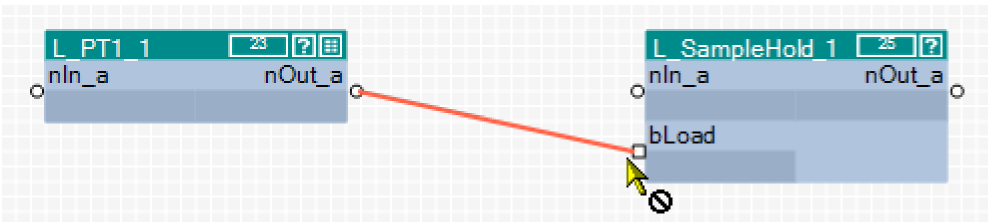
**Tip!**

If you move the mouse pointer across the port symbol while drawing a new connection, you can see whether the connection is permissible or not from the colour of the drawn line and from the mouse pointer symbol.

- Permissible connection:




- Impermissible connection (different port symbol):

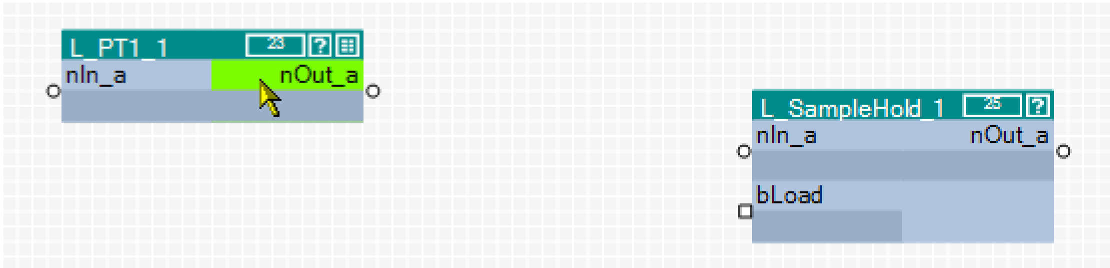



The command **Show as flag** in the *context menu* of a line serves to change the representation of the connection at any time.

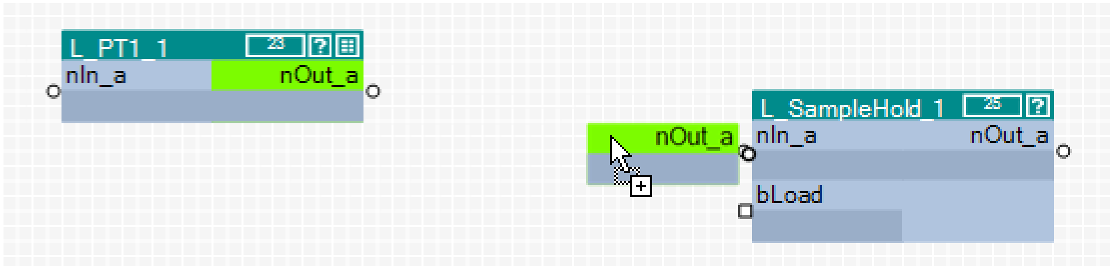
### 17.4.4.2 Creating a connection using port identifiers

**How to create a connection with port identifiers:**

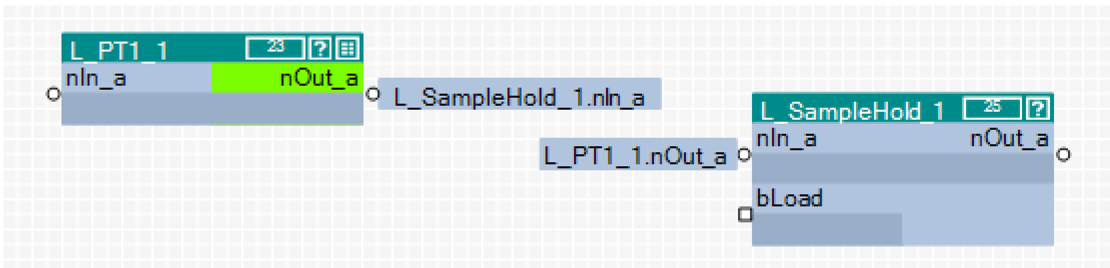
1. Click the port identifier.
   - The selected port is highlighted in light green:




2. Drag the port segment to the required port while keeping the left mouse button pressed:




After releasing the mouse button, the connection via port identifiers (flags) is created. The corresponding port identifier consists of the block name and the name of the input/output:




**Tip!**

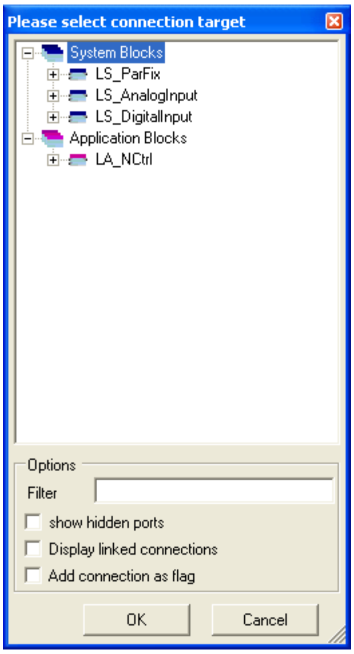
The command **Show as line** in the *context menu* of a flag serves to change the representation of the connection at any time.

### 17.4.4.3 Creating a connection via connection dialog

You can also create connections by means of a selection dialog instead of dragging by mouse. This especially makes sense if there is a great distance between the ports to be connected in the drawing area.

**How to create a connection using the selection dialog:**

1. Right-click the port identifier or click the port symbol from which the connection is to start.
   - The *context menu* for the port is displayed.
2. Go to the *context menu* for the port and select the **Add/change connection...** command.
   - The *Add/change connection* dialog box is displayed:




   - In a tree structure all inputs and outputs of the application are shown to which a connection is permissible.
   - You can enter an optional text into the **Filter** input field to reduce the selection to the blocks or ports which contain the entered text.
   - If you activate the **Show hidden ports** control field, the hidden ports for system and function blocks are shown as well.
3. Select the port where the connection is to end from the tree structure.
4. Activate the **Add connection as flag** control field if a port identifier (flag) is to be inserted instead of a connection line.
5. Press **OK** to create the connection to the selected port and close the dialog box.

### 17.4.4.4 Deleting connections that are no longer required

**How to delete connection lines:**

1. Select connection lines to be deleted.
   - Select a single connection line by directly clicking on the connection line with the right mouse button.
   - If you click further connection lines while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - All connection lines are highlighted in red.
2. Press **<Del>**.

**How to delete port identifiers/flags:**

1. Select the port identifiers to be deleted.
   - Select a single port identifier by directly clicking on the port identifier with the left mouse button.
   - If you click further port identifiers while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - All selected port identifiers are highlighted by a light green header.
2. Press **<Del>**.

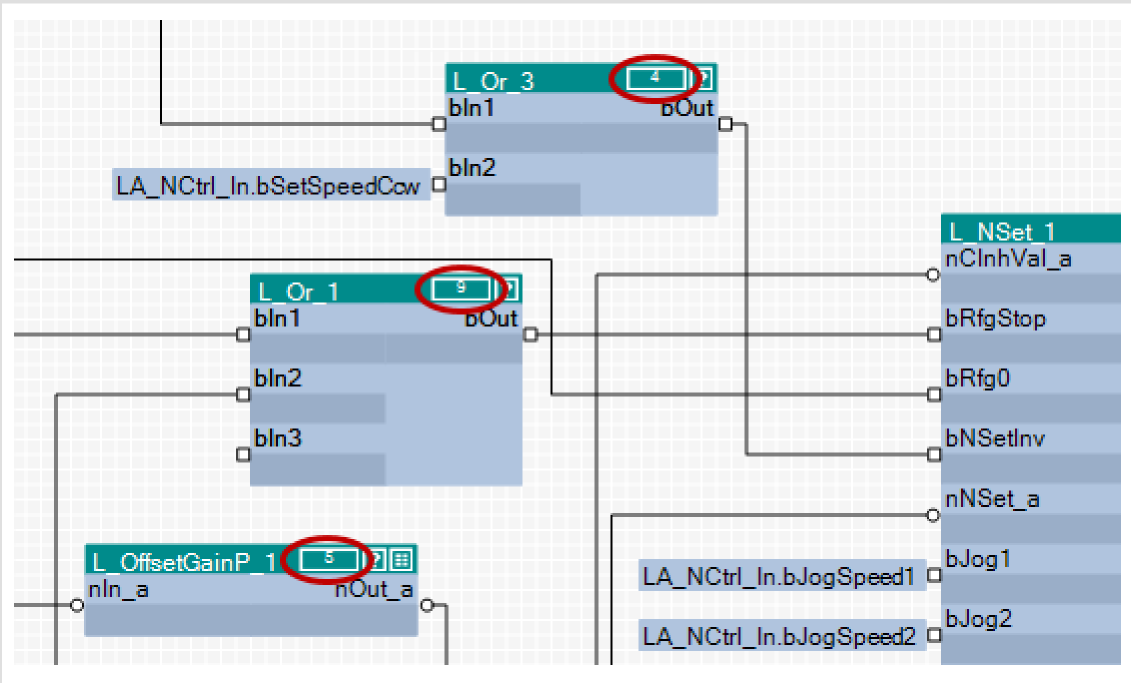
**Related topics**

▸ Deleting objects that are no longer required (1088)

### 17.4.5 Changing the processing order

If you insert a function block into the interconnection, an order index is automatically assigned to this function block. By means of this order index it is defined in which order the individual function blocks are calculated at runtime.

- ▶ The first function block inserted contains the order index "1", the next function block inserted contains the order index "2", etc.
- ▶ The respective order index is displayed in the header of the function block in the rectangle after the block name.




[17-10] Example: Function blocks with order index

**Note!**

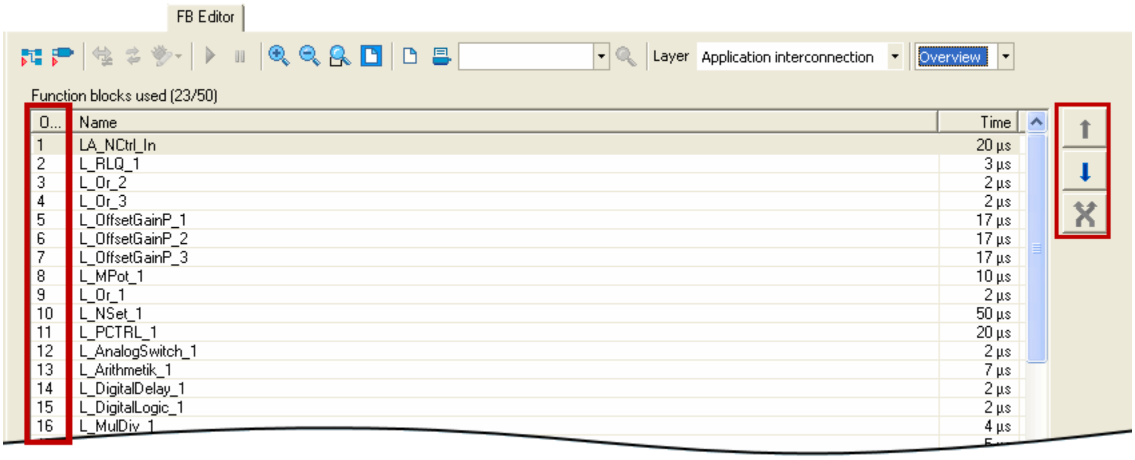
When a function block is shifted, its order index is maintained.

The processing order influences the result!

- In certain cases it may be sensible to change the processing order, but if you select an unfavourable processing order, errors may arise!

**How to change the processing order manually:**

1. Use the list field at the top right to change from the Editor to the overview.
   - The overview displays all function blocks of the interconnection in the order of their processing
   - In the first "Order" column the order index of each function block is listed.
2. Unless already selected, select the entry "Manual selection" in the **Optimisation...** list field.




3. Select the function block which is to receive a different position within the processing order.
   - If you click further function blocks while pressing **<Ctrl>** they are added to an already existing selection (multi-selection).
   - The **<Shift>** key serves to select a related area of function blocks.
4. Move the function block(s) to the desired position using the  and  buttons.
   - The  button serves to exchange two selected function blocks with regard to their order.
5. Repeat steps 3 and 4 until the required processing order has been established.

**Changing the processing order according to an automatically generated selection**

In addition to the manual selection, the **Optimisation...** list field also offers two options for an automatic adaptation of the processing order:

- **Signal flow**: The processing order is optimised according to the signal flow.
- **Topology**: The processing order is optimised according to the x/y arrangement of the function blocks in the FB Editor.

As long as an automatic adaptation has been selected, a manual change of the processing order is not possible.

### 17.4.6 Copying interconnection elements (across all devices)

Interconnection elements can be copied across the devices within the project if the devices belong to the same product family (e.g. Inverter Drives 8400).

All types of blocks and comments can be copied to the clipboard via the **Copy** command or the **<Ctrl>+<c>** shortcut and then be inserted into the FB interconnection of the same or another project device of the same product family using the **Paste** command or the **<Ctrl>+<v>** shortcut.

- During the copy process into the clipboard, existing connections between copied blocks are copied as well, and the layout is kept too. Moreover, the separate technical objects (e.g port definition) are copied. Selected connections cannot be copied on their own.
- The **Paste** command is available if the clipboard is not empty and if it was copied from a device of the same product family. Within this product family, all device types (e.g. 8400 xxxxLine Vxx.xx) are permitted.
- After the **Paste** command has been selected, a dialog box is displayed which serves to select which elements are to be inserted from the clipboard and how to solve name conflicts, if any.
- After inserting the elements, they are marked in the target interconnection in order to be repositioned or deleted again to undo the insertion.
- Inserting from the clipboard can be repeated. The originally copied contents of the clipboard remains unchanged when it is inserted.

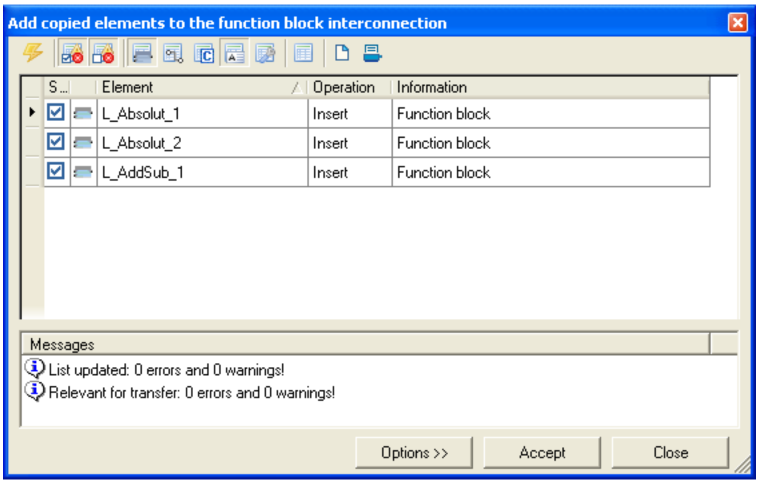
**How to copy one or several interconnection elements:**

1. Select the objects to be copied.
   - You can select a single object by clicking the header of the object.
   - If you click the header of further objects while pressing **<Ctrl>**, these will be added to an already existing selection (multi-selection).
   - You can easily select elements that are placed together by drawing a frame around these elements while keeping the mouse button pressed.
   - All selected objects are highlighted by a light green header.
2. Go to the *context menu* and select the **Copy** command (or **<Ctrl>+<c>**).
   - The selected elements are copied into the clipboard of the FB Editor.

3. If the elements are to be copied into a function block interconnection of another project device, change to the corresponding interconnection via the *project view*.
4. Go to the *context menu* and select the **Paste** command (or **<Ctrl>+<v>**).
5. Go to the *Insert FB interconnection* dialog box and select the elements to be inserted from the clipboard.
   - Detailed information on this dialog box can be obtained from the following subchapter "Insert options for copied elements". (📖 1101)
6. Click **Insert** to insert the selected elements into the target interconnection as defined.
   - Only possible if at least one element in the list has been selected for insertion.
   - Insertion is also possible via the **<Enter>** button if at least one element is selected from the list for insertion.
   - The original layout and the relative position of the inserted blocks to each other are maintained.
   - When copying across the devices, you also insert the corresponding separate technical objects (e.g. port definition).
   - The inserted elements are deleted from the list. If the list is empty, the dialog box is closed and the connections are inserted depending on the selected option.
7. If there are still elements to be entered in the list, repeat steps 5 and 6 until all elements are inserted as intended.
8. Press **Close** to stop the insertion and close the dialog box.
   - You can also use **<Esc>** or **<Enter>** to close the dialog box if "Insert" is not active.
   - The elements inserted into the target interconnection so far are maintained.
   - The connections for the blocks inserted so far are inserted depending on the selected option.

### 17.4.6.1 Insert options for copied elements

If interconnection elements have been copied to the clipboard, the »Engineer« will display a list of all elements contained in the clipboard when selecting the command **Insert** in the *Insert FB interconnection* dialog box:




The list shows the elements which can be added to the target interconnection, and the elements which cannot be added.

- In the "Selection" column, you can check/uncheck the elements to be added.
- Connections are only inserted when the dialog box is closed, which applies to all modules inserted so far. They are displayed as lines or flags, like in the original, but re-routed.
- The symbols in the *Toolbar* serve to execute the following functions:

| Symbol | Function |
|---|---|
|  | Add the selected elements to the interconnection |
|  | Show the elements to be added but are marked with an error or warning. |
|  | Show the elements not to be added and marked with an error or warning. |
|  | Show blocks |
|  | Show connections |
|  | Show parameters |
|  | Show comments |
|  | Show system elements |
|  | Show all |
|  | Print view |
|  | Print list |

▶ The buttons serve to execute the following functions:

| Button | Function |
|---|---|
| **Insert** | Add elements selected in the list to the target interconnection<br>• Only possible if at least one element in the list has been selected for insertion.<br>• Insertion is also possible via the **<Enter>** button if at least one element is selected from the list for insertion.<br>• The original layout and the relative position of the inserted blocks to each other are maintained.<br>• When copying across the devices, you also insert the corresponding separate technical objects (e.g. port definition).<br>• The added elements are simultaneously deleted from the list. The connections are added depending on the selected option. |
| **Close** | Close dialog box.<br>• You can also use **<Esc>** or **<Enter>** to close the dialog box if "Insert" is not active.<br>• The elements inserted into the target interconnection so far are maintained.<br>• The connections for the blocks inserted so far are inserted depending on the selected option. |

## 17.4.7 Resetting changed interconnection

If you only made changes on the I/O level, you can reset them by selecting a predefined control scheme in C00007. If you have also made changes on the application level, you must first reset the changed application to a predefined application in C00005.

**How to reset the application interconnection to a predefined application:**

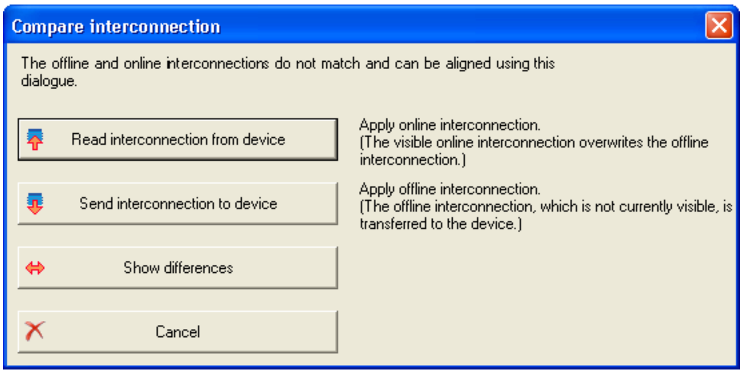
1. Go to the **Application parameters** tab.
2. Select the required application in the **Application** list field.

**How to reset the I/O interconnection to a predefined control scheme:**

1. Go to the **Application parameters** tab.
2. Select the required control scheme in the **Control source** list field.

## 17.5 Adjusting online and offline interconnection

If the »Engineer« detects that online and offline interconnection differ from each other, the *Compare interconnection* dialog box is displayed with various options for the adjustment:




**Tip!**

The dialog box can also be opened via the symbol in the *FB Editor toolbar*.

| Button | Function |
|---|---|
| **Accepting the interconnection from the device** | Add the interconnection in the device to the FB Editor. The interconnection existing in the FB Editor will be overwritten by this action. |
| **Transferring the interconnection to the device** | Transfer the offline interconnection which is currently not visible in the FB Editor to the device. The interconnection existing in the device will be overwritten by this action. |
| **Showing differences** | Showing differences between online and offline interconnection. |
| **Cancel** | Close the *Adjust interconnection* dialog box without making an adjustment. |

## 17.6 Printing the interconnection

The interconnection can be printed for documentation purposes, optionally on one page, on four pages, or not scaled.

**Tip!**

By clicking the icon in the *FB Editor toolbar*, you can get a print view before printing.

**How to print the interconnection:**

1. In the *FB Editor toolbar*, click the icon.
   - The *Circuit print size* dialog box is displayed.
2. Select the desired size and press **OK**.
   - The standard dialog box *Print* appears.
3. Press **OK** to start the printing process.

## 17.7 Comparing interconnections

The comparison operation serves to compare FB interconnections of 8400 devices within the project. An offline<>online comparison and the comparison of two online devices are possible.

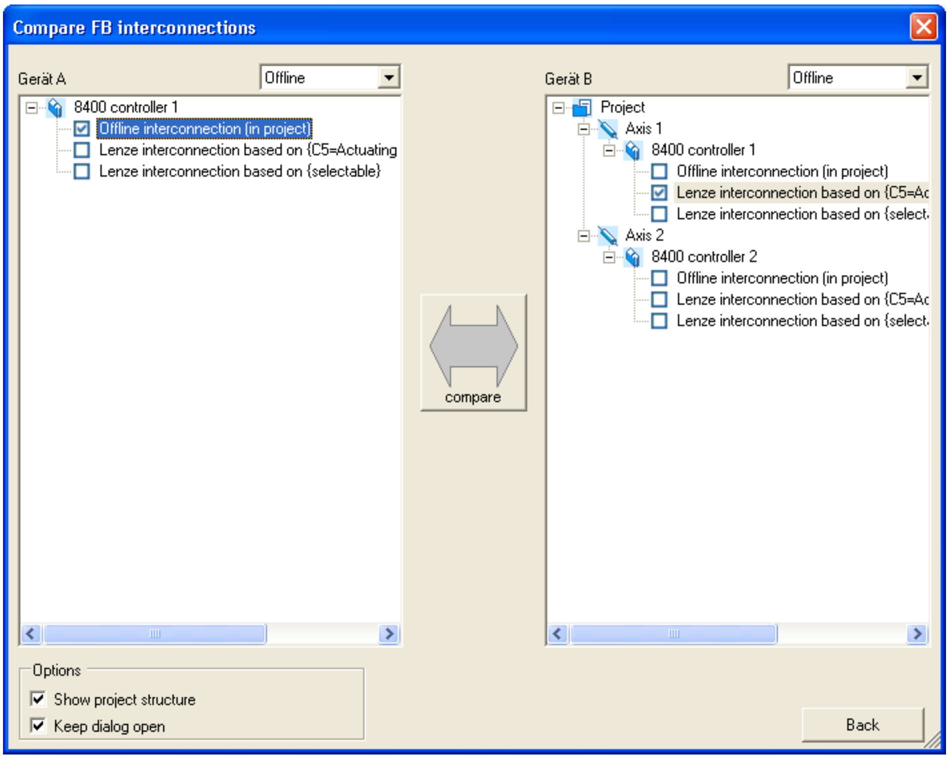
**Note!**

Only applications can be compared which have been enabled in the FB Editor!

Block positions, line representations, and connector visibilities are not compared.

**How to compare two FB interconnections:**

1. Select the command **Application data→Compare FB interconnections...**.
   - The *Compare FB interconnections* dialog box is displayed:

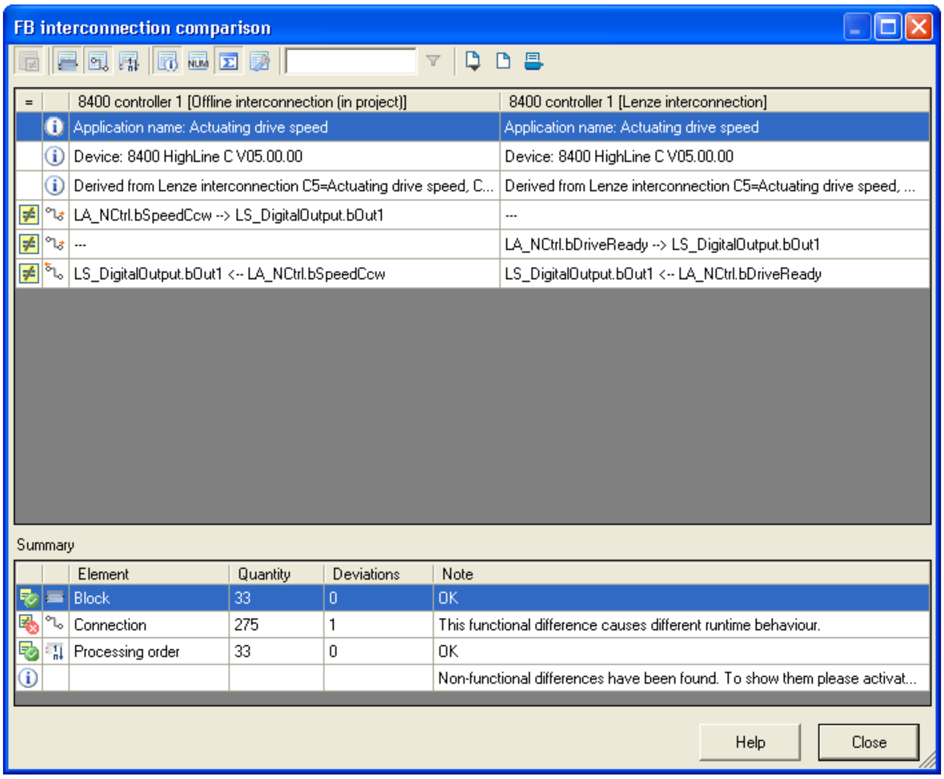



2. Select the interconnections to be compared in the project view represented on the left and right.
   - In order to execute a comparison with an online device, select "Online" in one of the two upper list fields. Then all available online devices are displayed for selection.
   - If you select "Online" in one of the two upper list fields, you can also compare the interconnections of two available online devices.

3. Click **Compare**.
   - If the comparison was executed successfully, the comparison result is displayed as a list (see the following section).
   - If a comparison of the selected interconnections is not possible, a corresponding message is displayed.
4. In order to stop the comparison operation and close the dialog box: Press **Back**.

### Representation of the comparison result

The comparison result is displayed in the form of a list in the *FB interconnection comparison* dialog box:




▶ The symbols in the *Toolbar* serve to show or hide different details and export and print the shown list.

| Symbol | Function |
|---|---|
|  | Only show differences<br>• Button can only be activated in expert mode. |
|  | Show blocks |
|  | Show connections |
|  | Show processing order |
|  | Show properties of blocks and connections<br>• Function is only available in expert mode. |
|  | Show comments<br>• Function is only available in expert mode. |

| Symbol | Function |
|---|---|
|  | Show block parameters<br>• Function is only available in expert mode. |
|  | Show application parameters<br>• Function is only available in expert mode. |
|  | Show all<br>• Function is only available in expert mode. |
|  | Show general information |
|  | Show summary |
|  | Activate expert mode<br>• In the expert mode, also non-functional differences are shown. |
|  | Apply filter<br>• Only show list entries which contain the text entered in the input field.<br>• Function is only available in expert mode. |
|  | Export shown list as comma-separated list (*.csv) |
|  | Print view |
|  | Print list |

## 17.8 Copying an interconnection

In contrast to copying/inserting selected interconnection elements via the clipboard, the function described in this chapter serves to replace the current FB interconnection of a device completely by the FB interconnection of another project device.

**Note!**

The complete FB interconnection can only be copied between devices of the same device type and version (e.g. 8400 HighLine C V1.0).

A complete interconnection comprises:

- Function blocks (use and parameter values)
- System blocks (application and parameter values)
- Port blocks (use and parameter values)
- Connections
- Comments
- Interconnection layout (arrangement of the modules)
- Port definition of the ports used in the FB interconnection

**How to copy the complete interconnection into another project device:**

1. Select the application with the FB interconnection to be copied in the *project view*.
2. Select the command **Application data→Copy FB interconnections...**.
3. Go to *project view* and select the application which is to be inserted into the copied FB interconnection.
4. Select the command **Application data→Add FB interconnection...**.
   - The command can only be activated if an FB interconnection has been copied from a device of the same device type and version.
   - After the command has been executed, the module assembly is compared. If there are relevant deviations, the insertion is refused and a corresponding message is displayed.
   - If an insertion is possible, you are asked if the FB interconnection is to be inserted.
5. Confirm the question if the copied FB interconnection is to be inserted with **Yes**.
   - After the insertion, an update of the project is required.

## 17.9 Exporting/Importing an interconnection

The interconnection existing in the project can be exported to a file for reuse/transfer to other devices.

**Note!**

The file can only be imported to devices of the same device type and version (e.g. 8400 HighLine C V1.0).

**How to export the interconnection from the project to a file:**

1. Go to the *Project view* in the *context menu* of the controller and select the **Export FB interconnection...** command.
2. Enter the memory location and the file name for the interconnection to be exported in the *Export FB interconnection* dialog box.
3. Click **Save** to export the interconnection and close the dialog box.

**How to import the interconnection from a file to the project:**

1. Go to the *Project view* in the *context menu* of the controller and select the **Import FB interconnection...** command.
2. Select the file with the interconnection to be imported in the *Import FB interconnection* dialog box.
3. Click **Open** to import the interconnection and close the dialog box.

# 18 Function library

## 18.1 Function blocks

This chapter describes the function blocks which are available for the controller in the FB Editor.

The system blocks are described in the chapter "System blocks". (1382)

**Note!**

A maximum of 75 function blocks can be used for a function block interconnection. The maximum calculating time is approx. 300 µs.

### Overview of function blocks available

| Function block | Runtime | Function |
|---|---|---|
| L_Absolute_1<br>L_Absolute_2 | 2 µs | ... converts a bipolar input signal into a unipolar output signal. |
| L_AddSub_1 | 3 µs | ... adds / subtracts analog input signals. |
| L_AnalogSwitch_1<br>L_AnalogSwitch_2<br>L_AnalogSwitch_3<br>L_AnalogSwitch_4<br>L_AnalogSwitch_5 | 2 µs | ... switches between two analog input signals. |
| L_And_1<br>L_And_2<br>L_And_3 | 2 µs | ... ANDs three binary signals. |
| L_And5_1<br>L_And5_2 | 2 µs | ... ANDs five binary signals. |
| L_Arithmetik_1<br>L_Arithmetik_2<br>L_Arithmetik_3<br>L_Arithmetik_4<br>L_Arithmetik_5 | 7 µs | ... combines two analog signals arithmetically. |
| L_ArithmetikPhi_1<br>L_ArithmetikPhi_2<br>L_ArithmetikPhi_3 | 7 µs | ... combines two angle signals arithmetically. |
| L_CalcDiameter_1 | 2 µs | ... can calculate the reel diameter from line speed and winding speed.<br>• This FB is available from version 02.00.00. |
| L_Compare_1<br>L_Compare_2<br>L_Compare_3<br>L_Compare_4<br>L_Compare_5 | 5 µs | ... compares two analog signals and can be used e.g. to implement a trigger. |
| L_ComparePhi_1<br>L_ComparePhi_2<br>L_ComparePhi_3<br>L_ComparePhi_4<br>L_ComparePhi_5 | 5 µs | ... compares two angle signals. |

| Function block | Runtime | Function |
|---|---|---|
| L_ConvAP_1<br>L_ConvAP_2<br>L_ConvAP_3 | 0 µs | ... converts an analog value into a position.<br>• This FB is available from version 02.00.00. |
| L_ConvBitsToWord_1<br>L_ConvBitsToWord_2<br>L_ConvBitsToWord_3 | 3 µs | ... converts 16 bit input values of the type "BOOL" into an output value of the type "WORD". |
| L_ConvDIntToWords_1<br>L_ConvDIntToWords_2<br>L_ConvDIntToWords_3 | 3 µs | ... converts an input value of the type "DINT" into two output values of the type "WORD". |
| L_ConvPA_1<br>L_ConvPA_2<br>L_ConvPA_3 | 3 µs | ... converts a position into an analog value.<br>• This FB is available from version 02.00.00. |
| L_ConvPP_1<br>L_ConvPP_2<br>L_ConvPP_3 | 3 µs | ... converts a position with dynamic fraction.<br>• This FB is available from version 02.00.00. |
| L_ConvUnitsToIncr_1<br>L_ConvUnitsToIncr_2<br>L_ConvUnitsToIncr_3 | 3 µs | FB in preparation!<br>... converts a position value provided in the real unit of the machine into an internal 32-bit position value. |
| L_ConvW_1<br>L_ConvW_2<br>L_ConvW_3<br>L_ConvW_4 | 2 µs | ... enables parameterisable conversion between analog signal forms.<br>• Division is remainder considered. |
| L_ConvWordsToDInt_1<br>L_ConvWordsToDInt_2<br>L_ConvWordsToDInt_3 | 3 µs | ... converts two inputs values of the type "WORD" into an output value of the type "DINT". |
| L_ConvWordToBits_1<br>L_ConvWordToBits_2<br>L_ConvWordToBits_3 | 3 µs | ... converts an input value of "WORD" type into 16 individual binary signals. |
| L_ConvX_1<br>L_ConvX_2<br>L_ConvX_3 | 0 µs | ... scales an analog value.<br>• This FB is available from version 02.00.00. |
| L_Counter_1<br>L_Counter_2<br>L_Counter_3 | 3 µs | ... is a digital upcounter and downcounter. |
| L_Curve_1<br>L_Curve_2<br>L_Curve_3 | 4 µs | ... can optionally display a characteristic function or a curve function y=f(x), the X axis being the input signal and the Y axis being the output signal.<br>• L_Curve_2 and L_Curve_3 are available from version 02.00.00. |
| L_DFlipFlop_1<br>L_DFlipFlop_2 | 3 µs | ... provides two stable states depending on the input signals. |
| L_DFRFG_1 | 3 µs | ... synchronises a drive (slave) to the master value of a master drive and then executes a angular synchronism with regard to the digital frequency.<br>• This FB is available from version 02.00.00. |
| L_DFSET_1 | 3 µs | ... prepares the master value for a slave drive. This FB enables the controller to follow the master drive true to speed and angle.<br>• This FB is available from version 02.00.00. |
| L_DigitalDelay_1<br>L_DigitalDelay_2<br>L_DigitalDelay_3 | 2 µs | ... delays binary signals. |
| L_DigitalLogic_1<br>L_DigitalLogic_2<br>L_DigitalLogic_3 | 2 µs | ... provides a binary output signal which is generated by the logic combination of three input signals.<br>• L_DigitalLogic_3 is available from version 02.00.00. |
| L_DigitalLogic5_1<br>L_DigitalLogic5_2 | 2 µs | ... provides a binary output signal which is generated by the logic combination of five input signals. |

| Function block | Runtime | Function |
|---|---|---|
| L_DT1_1 | 3 µs | ...differentiates signals. The function block can, for instance, be used to apply an acceleration (dv/dt). |
| L_FixSet_a_1 | 2 µs | ... outputs one of 16 parameterisable analog signals. |
| L_FixSet_w_1<br>L_FixSet_w_2 | 2 µs | ... outputs one of 16 parameterisable data words. |
| L_GainOffset_1<br>L_GainOffset_2<br>L_GainOffset_3 | 3 µs | ... can amplify an analog input signal and then add an offset to it.<br>• Gain and offset can be set via FB inputs. |
| L_GainOffsetP_1<br>L_GainOffsetP_2<br>L_GainOffsetP_3 | 3 µs | ... can amplify an analog input signal and then add an offset to it.<br>• Gain and offset can be set via parameters. |
| L_GainOffsetPhiP_1<br>L_GainOffsetPhiP_2 | 3 µs | ... can amplify an angle signal and then add an offset to it.<br>• Gain and offset can be set via parameters. |
| L_GearComp_1 | 3 µs | ... compensates elasticities in the drive train (gearbox compensation).<br>• This FB is available from version 02.00.00. |
| L_Interpolator_1 | 5 µs | ... can interpolate a position setpoint and/or an analog value e.g. to compensate for larger bus transmission cycles or to continue signal characteristics if data telegrams are missing. |
| L_JogCtrlExtension_1 | 5 µs | ... can be connected upstream to the L_NSet ramp generator to implement a switch-off positioning at limit switch. |
| L_Limit_1<br>L_Limit_2 | 3 µs | ... limits an analog input signal to an adjustable value range. |
| L_LimitPhi_1<br>L_LimitPhi_2<br>L_LimitPhi_3 | 3 µs | ... limits an angle signal to an adjustable value range. |
| L_MckCtrlInterface_1 | 5 µs | ... provides the application with process inputs for controlling various basic functions of the Motion Control Kernel. ▸ MCKInterface (📖 487) |
| L_MckStateInterface_1 | 5 µs | ... provides the application with various items of status information of the Motion Control via process outputs. ▸ MCKInterface (📖 487) |
| L_MPot_1 | 10 µs | ... replaces a hardware motor potentiometer as setpoint source. |
| L_MulDiv_1<br>L_MulDiv_2 | 4 µs | ... multiplies the analog input signal with a factor.<br>• Not remainder considered. |
| L_Mux_1 | 3 µs | ... switches one of eight selectable input signals to the output. |
| L_Negation_1<br>L_Negation_2 | 2 µs | ... negates an analog input signal. |
| L_NLim_1<br>L_NLim_2 | 3 µs | ... can suppress up to three parameterisable blocking zones within a continuous signal characteristic of an analog input signal. |
| L_Not_1<br>L_Not_2<br>L_Not_3<br>L_Not_4<br>L_Not_5<br>L_Not_6<br>L_Not_7 | 2 µs | ... inverts a digital input signal. |
| L_NSet_1 | 50 µs | ... contains a ramp generator with comprehensive parameterisation and control options to condition a setpoint signal. |
| L_Odometer_1 | 2 µs | ... detects positions and calculates distances. |
| L_OffsetGain_1<br>L_OffsetGain_2<br>L_OffsetGain_3 | 4 µs | ... can add an offset to an analog input signal and amplify it afterwards.<br>• Offset and gain can be set via FB inputs. |
| L_OffsetGainP_1<br>L_OffsetGainP_2<br>L_OffsetGainP_3 | 17 µs | ... can add an offset to an analog input signal and amplify it afterwards.<br>• Offset and gain can be set via parameters. |

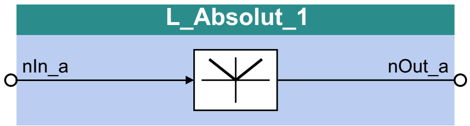
| Function block | Runtime | Function |
|---|---|---|
| L_OffsetGainPhiP_1<br>L_OffsetGainPhiP_2 | 17 µs | ... can add an offset to an angle signal and amplify it afterwards.<br>• Offset and gain can be set via parameters. |
| L_Or_1<br>L_Or_2<br>L_Or_3<br>L_Or_4 | 2 µs | ... ORs three binary signals.<br>• L_Or_4 is available from version 02.00.00. |
| L_Or5_1<br>L_Or5_2 | 2 µs | ... ORs five binary signals. |
| L_PCTRL_1 | 20 µs | ... is a PID controller and can be used for various control tasks. |
| L_PhaseDiff_1<br>L_PhaseDiff_2 | 2 µs | ... generates a position difference for the defined position setpoint from a position value and a speed signal. |
| L_PhaseIntK_1<br>L_PhaseIntK_2 | 5 µs | ... integrates a speed to an angle. |
| L_PhiIntegrator_1 | 2 µs | ... evaluates a speed with a gearbox factor and adds them in an integrator.<br>• This FB is available from version 02.00.00. |
| L_PosCtrlLin_1<br>L_PosCtrlLin_2 | 2 µs | ... is a simple linear profile generator.<br>• This FB is available from version 02.00.00. |
| L_PosiShaftCtrlInterface_1 | 5 µs | FB in preparation! |
| L_ProcessCtrl_1 | 2 µs | ... can realise a dancer position or tension control.<br>• This FB is available from version 02.00.00. |
| L_PT1_1<br>L_PT1_2<br>L_PT1_3 | 3 µs | ... filters and delays analog signals. |
| L_RLQ_1 | 3 µs | ... links a selected direction of rotation to the QSP function with wire-break protection. |
| L_RSFlipFlop_1<br>L_RSFlipFlop_2 | 3 µs | ... saves a binary input information element and resets it on command.. |
| L_SampleHold_1<br>L_SampleHold_2 | 3 µs | ... saves a value. |
| L_SignalMonitor_a | 15 µs | ... serves to output analog output signals of other FBs, SBs or LAs. |
| L_SignalMonitor_b | 3 µs | ... serves to output binary output signals of other FBs, SBs or LAs. |
| L_SignalSwitch_1<br>L_SignalSwitch_2<br>L_SignalSwitch_3<br>L_SignalSwitch_4 | 2 µs | ... switches between two input signals of the "WORD" data type. |
| L_SignalSwitch32_1<br>L_SignalSwitch32_2<br>L_SignalSwitch32_3 | 2 µs | ... .switches between two input signals of "DINT" data type.<br>• This FB is available from version 02.00.00. |
| L_SQrt_1 | 3 µs | ... outputs the square root for a DINT input value. |
| L_SRFG_1<br>L_SRFG_2 | 2 µs | ... is a ramp function generator with S-shaped ramps for limiting the temporal rise of analog signals. |
| L_SwitchPoint_1 | 3 µs | ... provides four position switch points, i.e. digital switches the binary statuses (FALSE/TRUE) of which depend on the actual position.<br>• This FB is available from version 02.00.00. |
| L_Transient_1<br>L_Transient_2<br>L_Transient_3<br>L_Transient_4<br>L_Transient_5<br>L_Transient_6<br>L_Transient_7<br>L_Transient_8 | 3 µs | ... evaluates digital signal edges and converts them into timed pulses. |

**Related topics:**

- Overview of system blocks available (📖 1382)
- Working with the FB Editor (📖 1058)

### 18.1.1 L_Absolute_1

This FB converts a bipolar input signal into a unipolar output signal.




#### Inputs

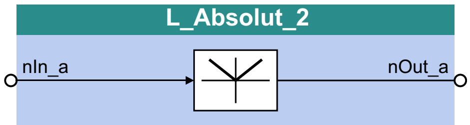
| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 18.1.2 L_Absolute_2

This FB converts a bipolar input signal into a unipolar output signal.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

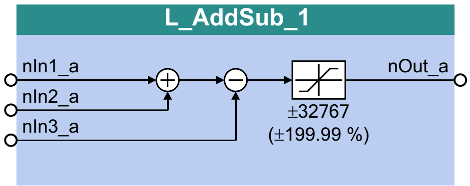
#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 18.1.3 L_AddSub_1

This FB is provided with two adding inputs and one subtracting input.

▶ The value provided at the *nOut_a* output is internally limited to ±32767.




#### Inputs

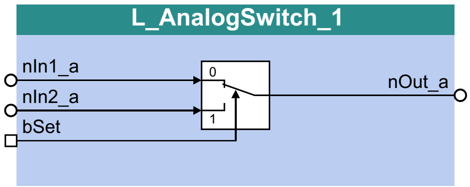
| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1<br>• This input is added |
| nIn2_a<br>INT | Input signal 2<br>• This input is added |
| nIn3_a<br>INT | Input signal 3<br>• This input is subtracted |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• *nOut_a = nIn1_a + nIn2_a - nIn3_a*<br>• Internal limitation to ±32767 (±199.99 %) |

## 18.1.4 L_AnalogSwitch_1

This function block switches between two analog input signals. The switching is controlled via a boolean input signal.




### Inputs

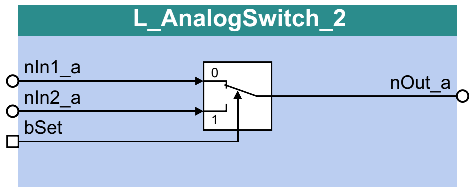
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

## 18.1.5 L_AnalogSwitch_2

This function block switches between two analog input signals. The switching is controlled via a boolean input signal.




### Inputs

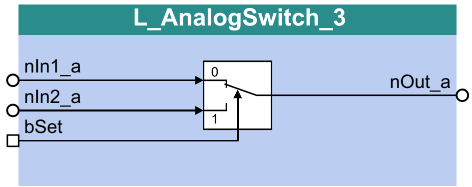
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

## 18.1.6 L_AnalogSwitch_3

This function block switches between two analog input signals. The switching is controlled via a boolean input signal.




### Inputs

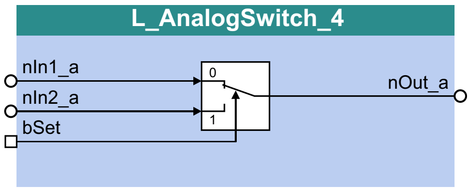
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

## 18.1.7 L_AnalogSwitch_4

This function block switches between two analog input signals. The switching is controlled via a boolean input signal.




### Inputs

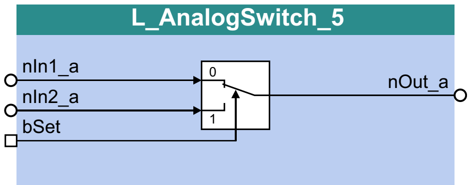
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

## 18.1.8 L_AnalogSwitch_5

This function block switches between two analog input signals. The switching is controlled via a boolean input signal.




### Inputs

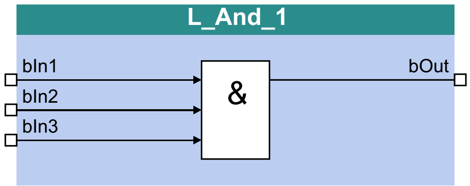
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn1_a<br>INT | Input signal 1 | |
| nIn2_a<br>INT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output to *nOut_a* | |
| | FALSE | *nIn1_a* |
| | TRUE | *nIn2_a* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

## 18.1.9 L_And_1

This FB implements the ANDing of the input signals.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

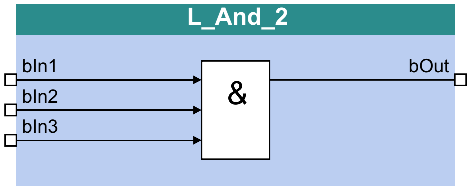
### Function

| Inputs | | | Output |
|---|---|---|---|
| bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE |

[18-1] Truth table of the FB L_And_1

## 18.1.10 L_And_2

This FB implements the ANDing of the input signals.




### Inputs

| Identifier Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

### Outputs

| Identifier Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

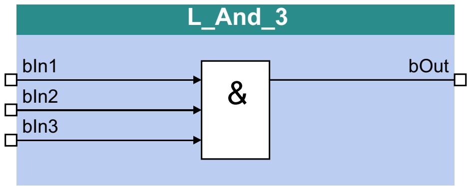
### Function

| Inputs | | | Output |
|---|---|---|---|
| **bIn3** | **bIn2** | **bIn1** | **bOut** |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE |

[18-2] Truth table of the FB L_And_2

## 18.1.11 L_And_3

This FB implements the ANDing of the input signals.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

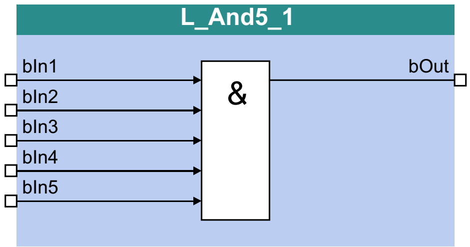
### Function

| Inputs | | | Output |
|---|---|---|---|
| bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE |

[18-3] Truth table of the FB L_And_3

## 18.1.12 L_And5_1

This FB implements the ANDing of the input signals.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

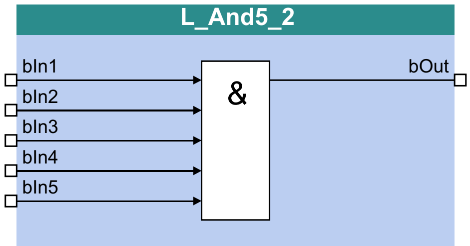
### Function

| Inputs | | | | | Output |
|---|---|---|---|---|---|
| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | FALSE | FALSE | TRUE | |
| FALSE | FALSE | FALSE | TRUE | FALSE | |
| FALSE | FALSE | FALSE | TRUE | TRUE | |
| FALSE | FALSE | TRUE | FALSE | FALSE | |
| | | ... | | | |
| TRUE | TRUE | TRUE | FALSE | TRUE | |
| TRUE | TRUE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE | TRUE | TRUE |

[18-4] Truth table of the FB L_And5_1

## 18.1.13 L_And5_2

This FB implements the ANDing of the input signals.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

### Function

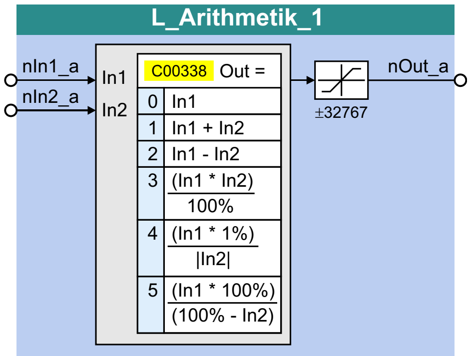
| Inputs | | | | | Output |
|---|---|---|---|---|---|
| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | FALSE | FALSE | TRUE | |
| FALSE | FALSE | FALSE | TRUE | FALSE | |
| FALSE | FALSE | FALSE | TRUE | TRUE | |
| FALSE | FALSE | TRUE | FALSE | FALSE | |
| | | ... | | | |
| TRUE | TRUE | TRUE | FALSE | TRUE | |
| TRUE | TRUE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE | TRUE | TRUE |

[18-5] Truth table of the FB L_And5_2

### 18.1.14 L_Arithmetik_1

This FB can combine two analog signals arithmetically.

- The arithmetic function is selected in C00338.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

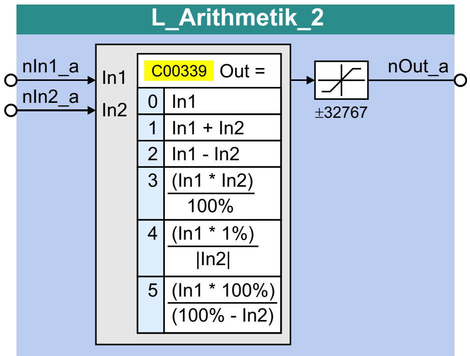
### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00338 | | | Function selection |
| | 0 | $nOut\_a = nIn1\_a$ | |
| | 1 | $nOut\_a = nIn1\_a + nIn2\_a$ | |
| | 2 | $nOut\_a = nIn1\_a - nIn2\_a$ | |
| | 3 | $nOut\_a = \frac{nIn1\_a \cdot nIn2\_a}{16384}$ | |
| | 4 | $nOut\_a = \frac{nIn1\_a}{\lvert nIn2\_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut\_a = \frac{nIn1\_a}{16384 - nIn2\_a} \cdot 16384$ | |

## 18.1.15 L_Arithmetik_2

This FB can combine two analog signals arithmetically.

- The arithmetic function is selected in C00339.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

### Outputs

| Identifier / Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

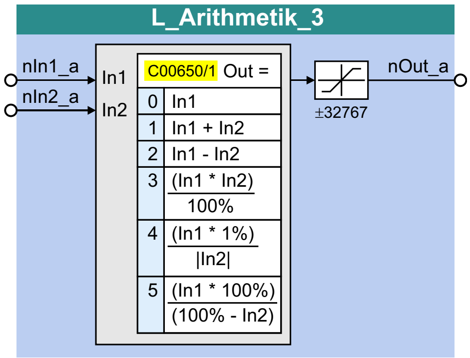
### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00339 | | | Function selection |
| | 0 | $nOut\_a = nIn1\_a$ | |
| | 1 | $nOut\_a = nIn1\_a + nIn2\_a$ | |
| | 2 | $nOut\_a = nIn1\_a - nIn2\_a$ | |
| | 3 | $nOut\_a = \frac{nIn1\_a \cdot nIn2\_a}{16384}$ | |
| | 4 | $nOut\_a = \frac{nIn1\_a}{\lvert nIn2\_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut\_a = \frac{nIn1\_a}{16384 - nIn2\_a} \cdot 16384$ | |

### 18.1.16 L_Arithmetik_3

This FB can combine two analog signals arithmetically.

- The arithmetic function is selected in C00650/1.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

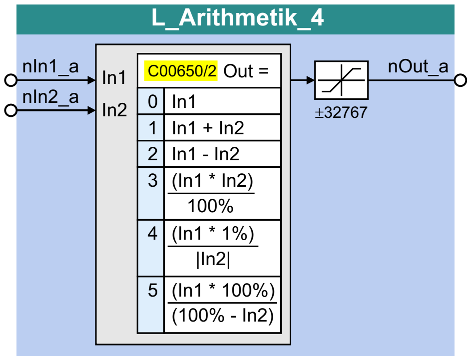
### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00650/1 | | | Function selection |
| | 0 | $nOut\_a = nIn1\_a$ | |
| | 1 | $nOut\_a = nIn1\_a + nIn2\_a$ | |
| | 2 | $nOut\_a = nIn1\_a - nIn2\_a$ | |
| | 3 | $nOut\_a = \frac{nIn1\_a \cdot nIn2\_a}{16384}$ | |
| | 4 | $nOut\_a = \frac{nIn1\_a}{\lvert nIn2\_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut\_a = \frac{nIn1\_a}{16384 - nIn2\_a} \cdot 16384$ | |

## 18.1.17 L_Arithmetik_4

This FB can combine two analog signals arithmetically.

- The arithmetic function is selected in C00650/2.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| nIn1_a / INT | Input signal 1 |
| nIn2_a / INT | Input signal 2 |

### Outputs

| Identifier / Data type | Value/meaning |
|---|---|
| nOut_a / INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

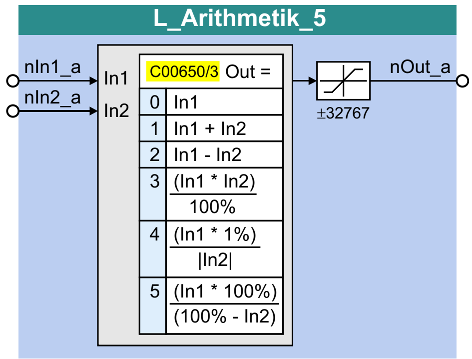
### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00650/2 | | | Function selection |
| | 0 | $nOut\_a = nIn1\_a$ | |
| | 1 | $nOut\_a = nIn1\_a + nIn2\_a$ | |
| | 2 | $nOut\_a = nIn1\_a - nIn2\_a$ | |
| | 3 | $nOut\_a = \frac{nIn1\_a \cdot nIn2\_a}{16384}$ | |
| | 4 | $nOut\_a = \frac{nIn1\_a}{\lvert nIn2\_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut\_a = \frac{nIn1\_a}{16384 - nIn2\_a} \cdot 16384$ | |

## 18.1.18 L_Arithmetik_5

This FB can combine two analog signals arithmetically.

- The arithmetic function is selected in C00650/3.
- All internal intermediate results and the value output at the *nOut_a* output are internally limited to ±32767.
- Division is not remainder considered.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %) |

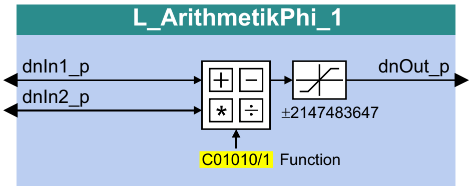
### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00650/3 | | | Function selection |
| | 0 | $nOut\_a = nIn1\_a$ | |
| | 1 | $nOut\_a = nIn1\_a + nIn2\_a$ | |
| | 2 | $nOut\_a = nIn1\_a - nIn2\_a$ | |
| | 3 | $nOut\_a = \frac{nIn1\_a \cdot nIn2\_a}{16384}$ | |
| | 4 | $nOut\_a = \frac{nIn1\_a}{\lvert nIn2\_a \rvert} \cdot 164$ | When the denominator has the value "0", it will be set to "1". |
| | 5 | $nOut\_a = \frac{nIn1\_a}{16384 - nIn2\_a} \cdot 16384$ | |

## 18.1.19 L_ArithmetikPhi_1

This FB can combine two angle signals arithmetically.

- The arithmetic function is selected in C01010/1.
- All internal intermediate results and the value provided at the *dnOut_p* output are internally limited to ±2147483647 (±$2^{31}$ - 1).
- Division is not remainder considered.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

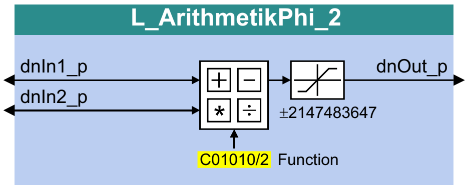
### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01010/1 | | | Function selection |
| | 0 | dnOut_p = dnIn1_p | |
| | 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| | 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| | 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| | 4 | dnOut_p = dnIn1_p / dnIn2_p | |

## 18.1.20 L_ArithmetikPhi_2

This FB can combine two angle signals arithmetically.

- The arithmetic function is selected in C01010/2.
- All internal intermediate results and the value provided at the *dnOut_p* output are internally limited to ±2147483647 (±$2^{31}$ - 1).
- Division is not remainder considered.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

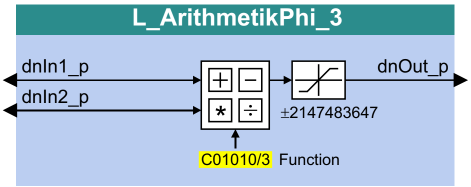
### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01010/2 | | | Function selection |
| | 0 | dnOut_p = dnIn1_p | |
| | 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| | 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| | 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| | 4 | dnOut_p = dnIn1_p / dnIn2_p | |

### 18.1.21 L_ArithmetikPhi_3

This FB can combine two angle signals arithmetically.

- The arithmetic function is selected in C01010/3.
- All internal intermediate results and the value provided at the *dnOut_p* output are internally limited to ±2147483647 (±$2^{31}$ - 1).
- Division is not remainder considered.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

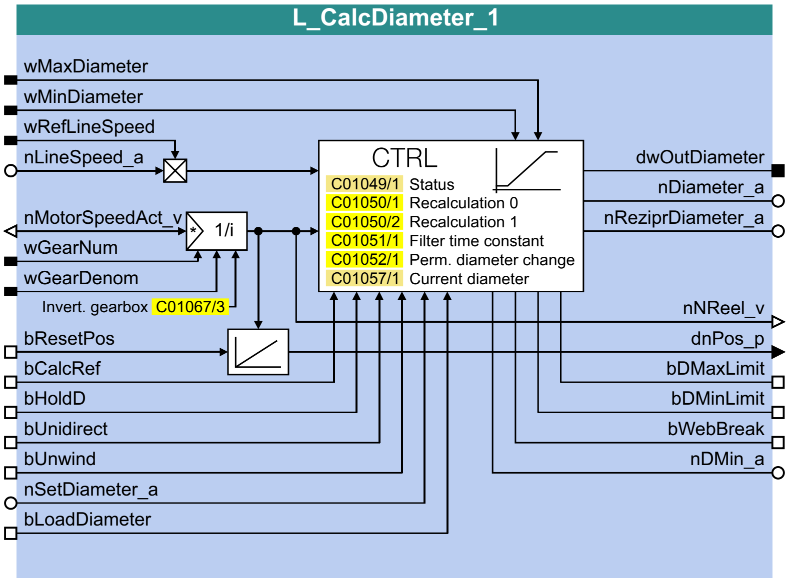
| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to ±2147483647 |

#### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01010/3 | | | Function selection |
| | 0 | dnOut_p = dnIn1_p | |
| | 1 | dnOut_p = dnIn1_p + dnIn2_p | |
| | 2 | dnOut_p = dnIn1_p - dnIn2_p | |
| | 3 | dnOut_p = dnIn1_p * dnIn2_p | |
| | 4 | dnOut_p = dnIn1_p / dnIn2_p | |

### 18.1.22 L_CalcDiameter_1

This FB serves to calculate the reel diameter from line speed and winding speed.




#### Inputs

| Identifier / Data type | Information/possible settings | |
|---|---|---|
| wMaxDiameter WORD | Maximum diameter in [mm]<br>• Internal limitation to 10000 mm (≡ 10 m) | |
| wMinDiameter WORD | Minimum diameter in [mm]<br>• Internal limitation to 10000 mm (≡ 10 m) | |
| wRefLineSpeed WORD | Maximum material speed in [0.1 m/min]<br>• Scaling: 2500 ≡ 250.0 m/min<br>• Internal limitation to 3000 m/min | |
| nLineSpeed_a INT | Material speed setpoint<br>• Scaling: 16384 ≡ maximum material speed (*wRefLineSpeed*) | |
| nMotorSpeedAct_v INT | Current motor speed in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm | |
| wGearNum WORD | Gearbox factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wGearDenom WORD | Gearbox factor (denominator)<br>• Internal limitation to 1 ... 32767 | |
| bResetPos BOOL | Reset angle of rotation *dnPos_p* | |
| | TRUE | The angle of rotation *dnPos_p* is reset. |
| bCalcRef BOOL | Selection of the calculation cycle | |
| | FALSE | Use diameter recalculation 0 (C01050/1). |
| | TRUE | Use diameter recalculation 1 (C01050/2). |
| bHoldD BOOL | Hold last diameter value | |
| | TRUE | The diameter value output last is not overwritten by the new values. |
| bUnidirect BOOL | Enable of only one change of direction and activation of the web break monitoring | |
| | TRUE | Only the change of direction defined via *bUnwind* is enabled. |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bUnwind<br>BOOL | Selection of the enabled change or direction<br>• Only when *bUnidirect* = TRUE | |
| | FALSE | Clockwise rotation (Cw) |
| | TRUE | Counter-clockwise rotation (Ccw) |
| nSetDiameter_a<br>DINT | Selection of an initial value/external diameter signal<br>• Scaling: 16384 ≡ maximum diameter (*wMaxDiameter*)<br>• The value is accepted by setting *bLoadDiameter* to TRUE. | |
| bLoadDiameter<br>BOOL | Accept initial value *nSetDiameter_a*<br>• This input has a higher priority than the *bHoldD*input. | |
| | TRUE | Accept the value at *nSetDiameter_a*. |

## Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| dwOutDiameter<br>DWORD | Current diameter in [µm]<br>• Internal limitation to 10 m<br>• Unfiltered | |
| nDiameter_a<br>INT | Current diameter in [%]<br>• 100 % ≡ Maximum diameter (*wMaxDiameter*)<br>• filtered via PT1 element (filter time constant can be set in C01051/1) | |
| nReziprDiameter_a<br>INT | Reciprocal value of the current diameter in [%]<br>• 100 % ≡ Minimum diameter (*wMinDiameter*) | |
| nNReel_v<br>INT | Current winding speed in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm | |
| bDMaxLimit<br>BOOL | Limit value monitoring | |
| | TRUE | Upper limit value *wMaxDiameter* reached. |
| bDMinLimit<br>BOOL | Limit value monitoring | |
| | TRUE | Lower limit value *wMinDiameter* reached. |
| bWebBreak<br>BOOL | Web break monitoring | |
| | TRUE | Web break after passing the monitor window set in C01052/1. |
| nDMin_a<br>INT | Minimum diameter in [%]<br>• 100 % ≡ Maximum diameter (*wMaxDiameter*) | |

## Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01049/1 | | | Status |
| | -10 | Line speed overflow | |
| | -1 | *wMinDiameter* > *wMaxDiameter* | |
| | 0 | OK - diameter has been recalculated | |
| | 10 | Diameter has been initialised with minimum diameter (*wMinDiameter)* | |
| | 20 | Diameter is loaded (*bLoadDiameter* active) | |
| | 30 | Diameter is held (*bHoldD* active) | |

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01050/1 | 0.001 | Rev. | 2.000 | Diameter recalculation 0<br>• Number of revolutions after which a diameter calculation is executed.<br>• Setting is effective when *bCalcRef* = FALSE.<br>• Initialisation: 1.000 |
| C01050/2 | 0.001 | Rev. | 2.000 | Diameter recalculation 1<br>• Number of revolutions after which a diameter calculation is executed.<br>• Setting is effective when *bCalcRef* = TRUE.<br>• Initialisation: 0.100 |
| C01051/1 | 0.010 | s | 3.000 | Filter time constant for calculated diameter values<br>• Initialisation: 1.000 s |
| C01052/1 | 0.00 | % | 100.00 | Permissible diameter change in opposite direction (web break monitoring)<br>• Only relevant when *bUnidirect* = TRUE.<br>• Referring to the maximum diameter (*wMaxDiameter*).<br>• Initialisation: 10.00 % |
| C01057/1 | 0.000 | mm | 10000.000 | Current diameter<br>• Read only |
| C01067/3 | | | | Invert. gearbox nMotorSpeedAct_v |
| | 0 | not inverted | | |
| | 1 | inverted | | |
| | 2 | Automatically from MCK | | |

### 18.1.22.1 Set initial value

An initial value or an external diameter signal can be defined at the *nSetDiameter_a* input.

- this value is accepted if *bLoadDiameter* is set to TRUE.
- In case of acceptance, the diameter calculation will be reset and the filter for diameter calculation is loaded with the defined initial value.

### 18.1.22.2 Calculate diameter

For reel diameter calculation, the material speed and the current winding speed are integrated cyclically.

- The material speed is defined via *nLineSpeed_a*.
- The winding speed is calculated from the *nMotorSpeedAct_v* motor speed and the current *wGearNum/wGearDenom* gearbox factor.
- C01050/1...2 can be used to define two different calculation cycles. The setting to the used is selected via *bCalcRef*.
- At the end of each integration interval, a new diameter value results from the division of the integrator values.
- An absolute value generation and smoothing via a first order low pass is carried out. The time constant of this filter can be set in C01051/1. The *dwOutDiameter* output is not filtered.

### 18.1.22.3 Select change direction, web break monitoring

By setting *bUnidirect* to TRUE, you only enable one change direction for diameter calculation and simultaneously activate the web break monitoring.

- The permissible change direction can be defined via *bUnwind*.
- A recalculated value is only accepted if it exceeds the value saved last in the permissible direction.

After a web break, the diameter values calculated successively mostly run very quickly in the opposite direction.

- C01052/1 serves to define the maximally permissible deviation in the opposite direction. If it is exceeded, the *bWebBreak* output is set to TRUE.
- If *bUnidirect* is set to FALSE, both change directions are enabled and the internal memory is always overwritten with the recalculated value which switches off the web break monitoring.

### 18.1.22.4 Holding the current value

By setting *bHoldD* to TRUE, you fix the diameter value *nDiameter_a* output last.

- The diameter calculation is reset and the filter for diameter calculation is loaded with the internally held diameter value.

### 18.1.22.5 Limit value monitoring

*wMinDiameter* and *wMaxDiameter* serve to define the limit values for the given diameter values. If they are reached/exceeded, the corresponding ouput (*bDMaxLimit*/*bDminLimit*) is set to TRUE.

- The unfiltered diameter value *dwOutDiameter* and the filtered diameter value *nDiameter_a* are limited.
- The limit values are entered in [mm].
- The hysteresis for resetting *bDMaxLimit*/*bDminLimit* is permanently set to 1 % of *wMaxDiameter*.
- For parameter setting of further FBs, the value in *wMinDiameter* with regard to *wMaxDiameter* is provided via the *nDMin_a* output.

### 18.1.22.6 Converting diameter in 1/D

For evaluating the material speed with 1/D (speed-controlled winder), the reciprocal value of the reel diameter is provided via *nReziprDiameter_a*.

- This value refers to the limitation value *wMinDiameter*.

### 18.1.22.7 Physical state variables

The *dwOutDiameter* output provides the current diameter as physical quantity (1 LSB ≡ 1 m).

*nNReel_v* outputs the winding speed which results from the *nMotorSpeedAct_v* motor speed and the current (*wGearNum*/*wGearDenom*) gearbox factor.

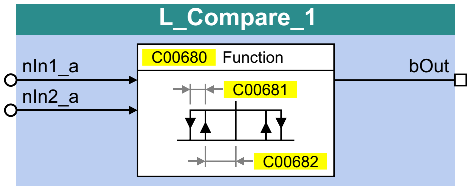
For supporting further functions (e.g. a traversing control), the winding speed is integrated additionally.

- The angle of rotation is output via *dnPos_p*.
- By setting *bResetPos* to TRUE, the angle of rotation is reset.

## 18.1.23 L_Compare_1

This FB compares two analog signals and can be used e.g. to implement a trigger.

- Comparison operation, hysteresis and window size can be parameterised.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

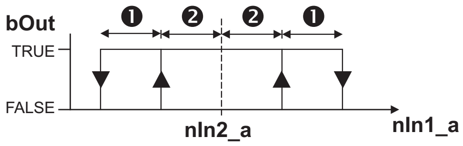
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00680 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00681 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.50 % |
| C00682 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 2.00 % |

### 18.1.23.1 Function 1: nIn1 = nIn2

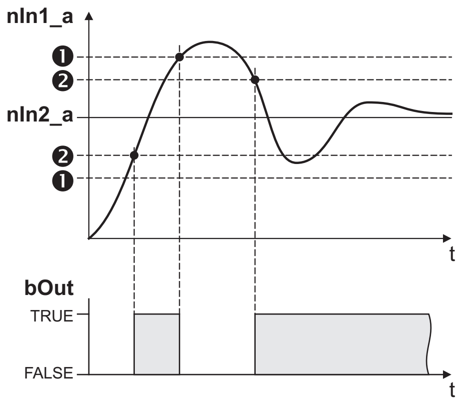
This function compares two signals with regard to equality. It can, for instance, provide the comparison "actual speed equals setpoint speed " ($n_{act} = n_{set}$).

- Use C00682 to set the window within which the equality is to apply.
- Use C00681 to set a hysteresis if the input signals are not stable and the output oscillates.




❶ C00681: Hysteresis
❷ C00682: Window

[18-6] Function 1: Switching performance

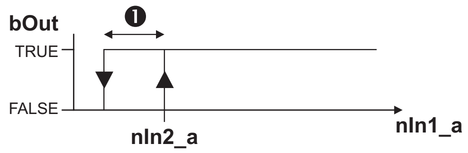



❶ C00681: Hysteresis
❷ C00682: Window

[18-7] Function 1: Example

### 18.1.23.2 Function 2: nIn1 > nIn2

This function serves, for instance, to implement the comparison "actual speed is higher than a limit value" ($n_{act}$ > $n_x$) for one direction of rotation.

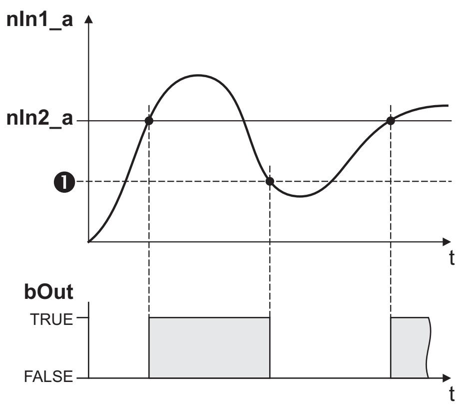



❶ C00681: Hysteresis

[18-8] Function 2: Switching performance

#### Functional sequence

1. If the value at *nIn1_a* exceeds the value *nIn2_a*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *nIn1_a* falls below the value of *nIn2_a* - *hysteresis* again, *bOut* changes back from TRUE to FALSE.

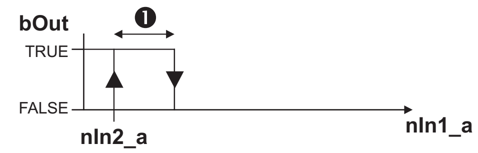



❶ C00681: Hysteresis

[18-9] Function 2: Example

### 18.1.23.3 Function 3: nIn1 < nIn2

This function serves, for instance, to implement the comparison "actual speed is lower than a limit value" ($n_{act} < n_x$) for one direction of rotation.

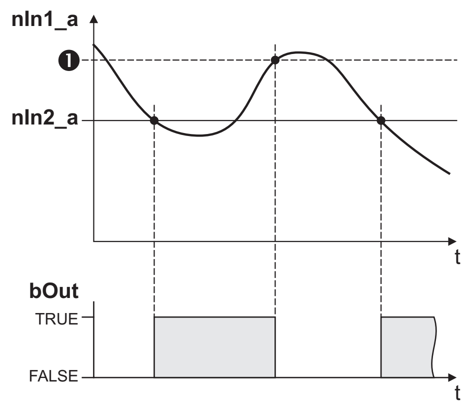



❶ C00681: Hysteresis

[18-10] Function 3: Switching performance

#### Functional sequence

1. If the value at *nIn1_a* falls below the value at *nIn2_a*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *nIn1_a* exceeds the value of *nIn2_a - hysteresis* again, *bOut* changes back from TRUE to FALSE.




❶ C00681: Hysteresis

[18-11] Function 3: Example

### 18.1.23.4 Function 4: |nIn1| = |nIn2|

This function serves to implement e.g. the comparison "$n_{act}$ = 0". This function is similar to function 1. However, the amount is generated by the input signals before signal processing (without sign).

▸ Function 1: nIn1 = nIn2

### 18.1.23.5 Function 5: |nIn1| > |nIn2|

This function serves to implement e.g. the comparison "$|n_{act}| > |n_x|$" irrespective of the direction of rotation. This function is similar to function 2. However, the amount is generated by the input signals before signal processing (without sign).

▸ Function 2: nIn1 > nIn2

### 18.1.23.6 Function 6: |nIn1| < |nIn2|

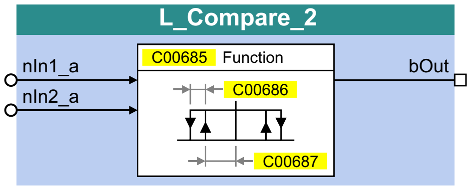
This function serves to implement the comparison "$|n_{act}| < |n_x|$" independent of the direction of rotation. This function is similar to function 3. However, the amount is generated by the input signals before signal processing (without sign).

▸ Function 3: nIn1 < nIn2

### 18.1.24 L_Compare_2

This FB compares two analog signals and can be used e.g. to implement a trigger.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameter

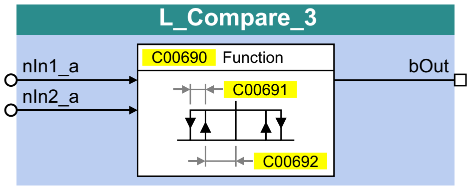
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00685 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00686 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.50 % |
| C00687 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 2.00 % |

For a detailed functional description see L_Compare_1.

## 18.1.25 L_Compare_3

This FB compares two analog signals and can be used e.g. to implement a trigger.

▶ Comparison operation, hysteresis and window size can be parameterised.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

### Parameter

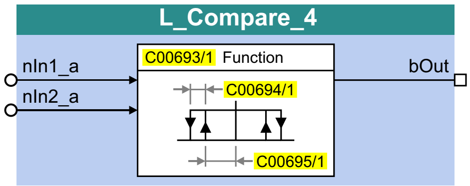
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00690 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00691 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.50 % |
| C00692 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 2.00 % |

For a detailed functional description see L_Compare_1.

### 18.1.26 L_Compare_4

This FB compares two analog signals and can be used e.g. to implement a trigger.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameter

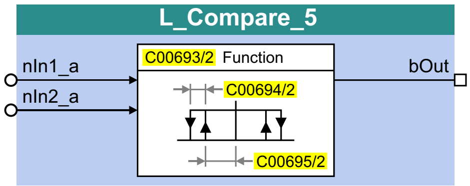
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00693/1 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00694/1 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.00 % |
| C00695/1 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 0.00 % |

For a detailed functional description see L_Compare_1.

### 18.1.27 L_Compare_5

This FB compares two analog signals and can be used e.g. to implement a trigger.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| nIn1_a<br>INT | Input signal 1 |
| nIn2_a<br>INT | Input signal 2 |

#### Outputs

| Identifier / Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

#### Parameter

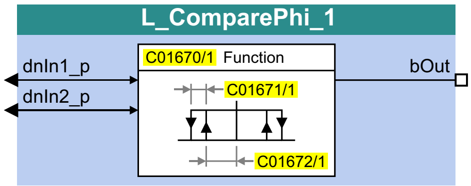
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00693/2 | | | | Function selection |
| | 1 | nIn1 = nIn2 | | |
| | 2 | nIn1 > nIn2 | | |
| | 3 | nIn1 < nIn2 | | |
| | 4 | \|nIn1\| = \|nIn2\| | | |
| | 5 | \|nIn1\| > \|nIn2\| | | |
| | 6 | \|nIn1\| < \|nIn2\| | | |
| C00694/2 | 0.00 | % | 100.00 | Hysteresis<br>• Lenze setting: 0.00 % |
| C00695/2 | 0.00 | % | 100.00 | Window<br>• Lenze setting: 0.00 % |

For a detailed functional description see L_Compare_1.

### 18.1.28 L_ComparePhi_1

This FB compares two angle signals.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

#### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

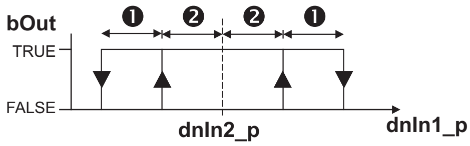
#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/1 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/1 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/1 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

### 18.1.28.1 Function 1: dnIn1 = dnIn2

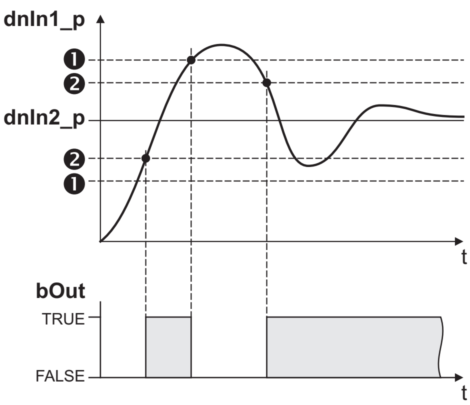
This function compares two signals with regard to equality. It can, for instance, provide the comparison "actual speed equals setpoint speed " ($n_{act} = n_{set}$).

- Use C01672 to set the window within which the equality is to apply.
- Use C01671 to set a hysteresis if the input signals are not stable and the output oscillates.




❶ C001671: Hysteresis
❷ C001672: Window

[18-12] Function 1: Switching performance

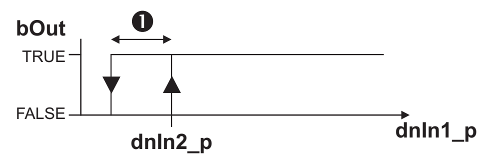



❶ C001671: Hysteresis
❷ C001672: Window

[18-13] Function 1: Example

### 18.1.28.2 Function 2: dnIn1 > dnIn2

This function serves, for instance, to implement the comparison "actual speed is higher than a limit value" ($n_{act} > n_x$) for one direction of rotation.

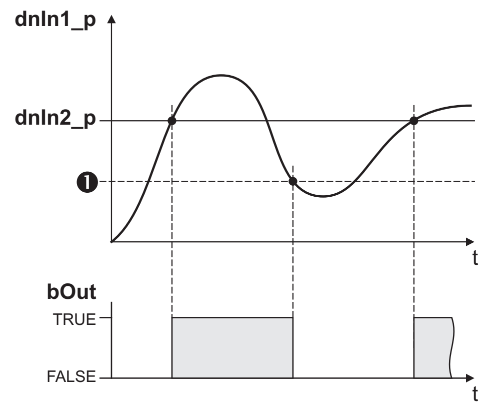



❶ C001671: Hysteresis

[18-14] Function 2: Switching performance

#### Functional sequence

1. If the value at *dnIn1_p* exceeds the value at *dnIn2_p*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *dnIn1_p* falls below the value of *dnIn2_p* - *hysteresis* again, *bOut* changes back from TRUE to FALSE.

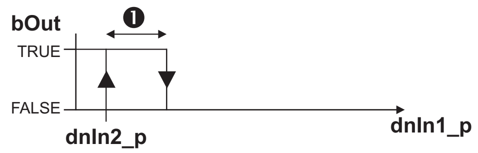



❶ C001671: Hysteresis

[18-15] Function 2: Example

### 18.1.28.3 Function 3: dnIn1 < dnIn2

This function serves, for instance, to implement the comparison "actual speed is lower than a limit value" ($n_{act} < n_x$) for one direction of rotation.

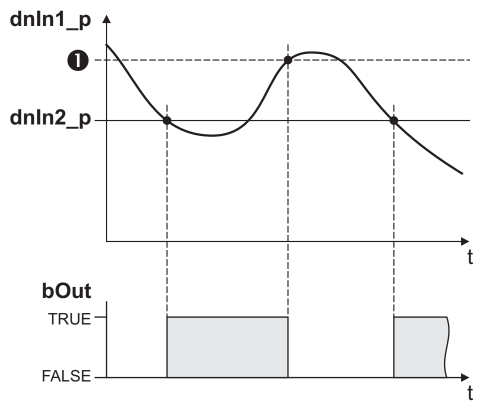



❶ C001671: Hysteresis

[18-16] Function 3: Switching performance

#### Functional sequence

1. If the value at *dnIn1_p* falls below the value at *dnIn2_p*, *bOut* changes from FALSE to TRUE.
2. Only if the signal at *dnIn1_p* exceeds the value of *dnIn2_p* - *hysteresis* again, *bOut* changes back from TRUE to FALSE.




❶ C001671: Hysteresis

[18-17] Function 3: Example

### 18.1.28.4 Function 4: |dnIn1| = |dnIn2|

This function serves to implement e.g. the comparison "$n_{act}$ = 0". This function is similar to function 1. However, the amount is generated by the input signals before signal processing (without sign).

Function 1: dnIn1 = dnIn2

### 18.1.28.5 Function 5: |dnIn1| > |dnIn2|

This function serves to implement e.g. the comparison "$|n_{act}| > |n_x|$" irrespective of the direction of rotation. This function is similar to function 2. However, the amount is generated by the input signals before signal processing (without sign).

Function 2: dnIn1 > dnIn2

### 18.1.28.6 Function 6: |dnIn1| < |dnIn2|

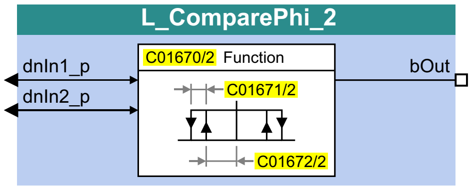
This function serves to implement the comparison "$|n_{act}| < |n_x|$" independent of the direction of rotation. This function is similar to function 3. However, the amount is generated by the input signals before signal processing (without sign).

Function 3: dnIn1 < dnIn2

## 18.1.29 L_ComparePhi_2

This FB compares two angle signals.

- Comparison operation, hysteresis and window size can be parameterised.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

### Parameter

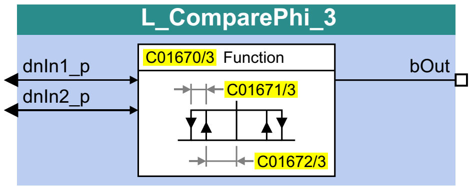
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/2 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/2 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/2 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

For a detailed functional description see L_ComparePhi_1.

## 18.1.30 L_ComparePhi_3

This FB compares two angle signals.

- Comparison operation, hysteresis and window size can be parameterised.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

### Parameter

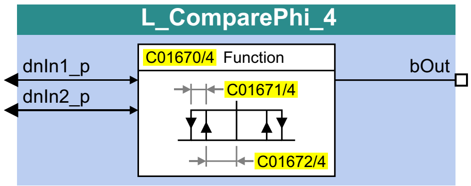
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/3 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/3 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/3 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

For a detailed functional description see L_ComparePhi_1.

## 18.1.31 L_ComparePhi_4

This FB compares two angle signals.

▶ Comparison operation, hysteresis and window size can be parameterised.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn1_p<br>DINT | Input signal 1 |
| dnIn2_p<br>DINT | Input signal 2 |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bOut<br>BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

### Parameter

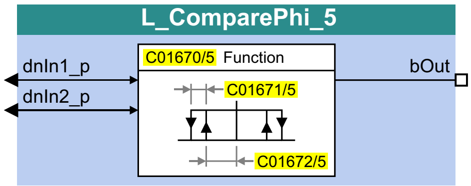
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/4 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/4 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/4 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

For a detailed functional description see L_ComparePhi_1.

### 18.1.32 L_ComparePhi_5

This FB compares two angle signals.

- Comparison operation, hysteresis and window size can be parameterised.




#### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| dnIn1_p / DINT | Input signal 1 |
| dnIn2_p / DINT | Input signal 2 |

#### Outputs

| Identifier / Data type | Value/meaning | |
|---|---|---|
| bOut / BOOL | Status signal "Comparison statement is true" | |
| | TRUE | The statement of the selected comparison mode is true. |

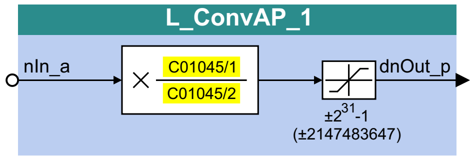
#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01670/5 | | | | Function selection |
| | 1 | dnIn1 = dnIn2 | | |
| | 2 | dnIn1 > dnIn2 | | |
| | 3 | dnIn1 < dnIn2 | | |
| | 4 | \|dnIn1\| = \|dnIn2\| | | |
| | 5 | \|dnIn1\| > \|dnIn2\| | | |
| | 6 | \|dnIn1\| < \|dnIn2\| | | |
| C01671/5 | 0 | Incr. | 1073741824 | Hysteresis<br>• Lenze setting: 0 incr. |
| C01672/5 | 0 | Incr. | 1073741824 | Window<br>• Lenze setting: 0 incr. |

For a detailed functional description see L_ComparePhi_1.

## 18.1.33 L_ConvAP_1

This FB converts an analog value into a position.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

### Parameter

| Parameter | Possible settings | | | | Info |
|---|---|---|---|---|---|
| C01045/1 | -32767 | | | 32767 | Numerator<br>• Lenze setting: 1 |
| C01045/2 | -32767 | | | 32767 | Denominator<br>• Lenze setting: 1 |

### Function

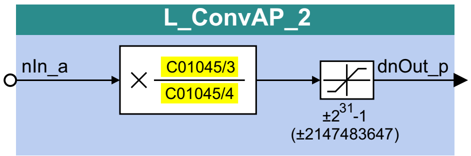
**Note!**

Conversion is remainder considered.

$$dnOut\_p = nIn\_a \cdot \frac{C01045/1}{C01045/2}$$

### 18.1.34 L_ConvAP_2

This FB converts an analog value into a position.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value<br>• Scaling: 16384 ≡ 100 % |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01045/3 | -32767 | | 32767 | Numerator<br>• Lenze setting: 1 |
| C01045/4 | -32767 | | 32767 | Denominator<br>• Lenze setting: 1 |

#### Function

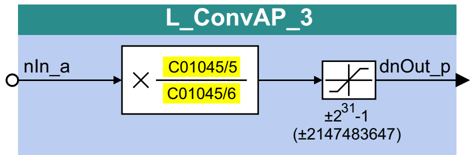
**Note!**

Conversion is remainder considered.

$$dnOut\_p = nIn\_a \cdot \frac{C01045/3}{C01045/4}$$

## 18.1.35 L_ConvAP_3

This FB converts an analog value into a position.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01045/5 | -32767 | | 32767 | Numerator<br>• Lenze setting: 1 |
| C01045/6 | -32767 | | 32767 | Denominator<br>• Lenze setting: 1 |

### Function

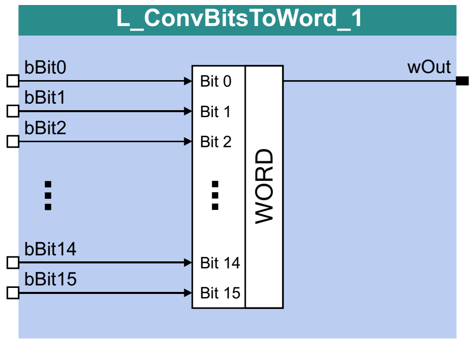
**Note!**

Conversion is remainder considered.

$$dnOut\_p = nIn\_a \cdot \frac{C01045/5}{C01045/6}$$

### 18.1.36 L_ConvBitsToWord_1

This FB converts 16 bit input values of the type "BOOL" into an output value of the type "WORD".




#### Inputs

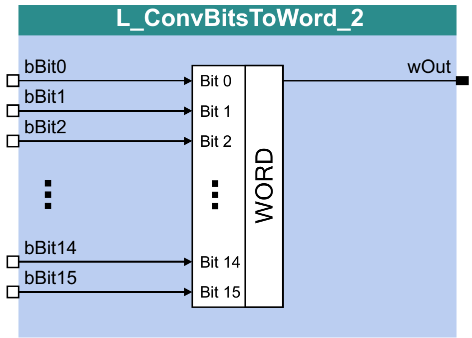
| Identifier<br>Data type | Information/possible settings |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

## 18.1.37 L_ConvBitsToWord_2

This FB converts 16 bit input values of the type "BOOL" into an output value of the type "WORD".




### Inputs

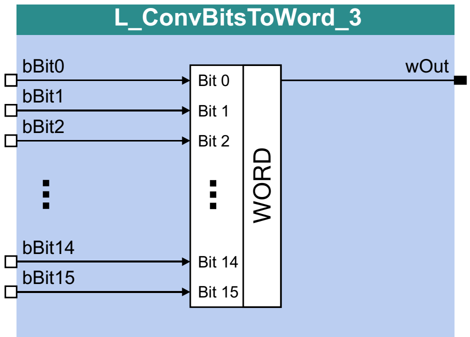
| Identifier<br>Data type | Information/possible settings |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### 18.1.38 L_ConvBitsToWord_3

This FB converts 16 bit input values of the type "BOOL" into an output value of the type "WORD".




#### Inputs

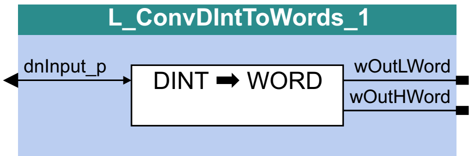
| Identifier<br>Data type | Information/possible settings |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### 18.1.39 L_ConvDIntToWords_1

This FB converts an input value of the type "DINT" into two output values of the type "WORD".




#### Inputs

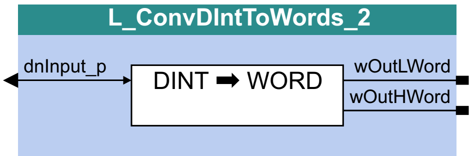
| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnInput_p<br>DINT | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOutLWord<br>WORD | Output signal Low Word |
| wOutHWord<br>WORD | Output signal High Word |

## 18.1.40 L_ConvDIntToWords_2

This FB converts an input value of the type "DINT" into two output values of the type "WORD".




### Inputs

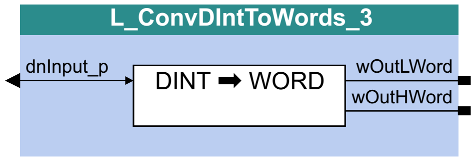
| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnInput_p<br>DINT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOutLWord<br>WORD | Output signal Low Word |
| wOutHWord<br>WORD | Output signal High Word |

## 18.1.41 L_ConvDIntToWords_3

This FB converts an input value of the type "DINT" into two output values of the type "WORD".




### Inputs

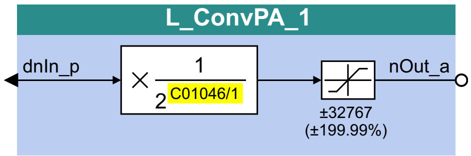
| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnInput_p<br>DINT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOutLWord<br>WORD | Output signal Low Word |
| wOutHWord<br>WORD | Output signal High Word |

## 18.1.42 L_ConvPA_1

This FB converts a position into an analog value.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01046/1 | 0 | | 31 | Division factor<br>• Lenze setting: 1 |

### Function

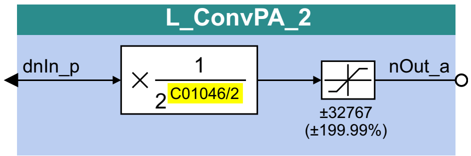
**Note!**

Conversion is remainder considered.

$$nOut\_a = dnIn\_p \cdot \frac{1}{2^{C01046/1}}$$

## 18.1.43 L_ConvPA_2

This FB converts a position into an analog value.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01046/2 | 0 | | 31 | Division factor<br>• Lenze setting: 1 |

### Function

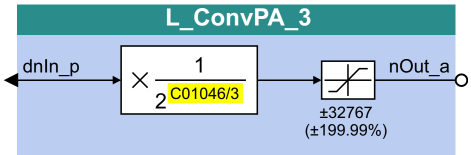
**Note!**

Conversion is remainder considered.

$$nOut\_a = dnIn\_p \cdot \frac{1}{2^{C01046/2}}$$

## 18.1.44 L_ConvPA_3

This FB converts a position into an analog value.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01046/3 | 0 | | 31 | Division factor<br>• Lenze setting: 1 |

### Function

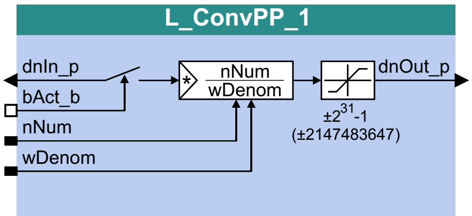
**Note!**

Conversion is remainder considered.

$$nOut\_a = dnIn\_p \cdot \frac{1}{2^{C01046/3}}$$

## 18.1.45 L_ConvPP_1

This FB converts a position with dynamic fraction.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn_p<br>DINT | Input signal | |
| bAct_b<br>BOOL | Conversion mode | |
| | FALSE | $dnOut\_p = Remainder \cdot \frac{nNum}{wDenom}$ |
| | TRUE | $dnOut\_p = dnIn\_p \cdot \frac{nNum}{wDenom}$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

### Outputs

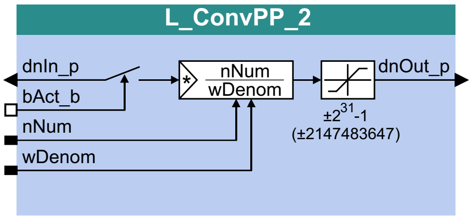
| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

**Note!**

Conversion is remainder considered.

### 18.1.46 L_ConvPP_2

This FB converts a position with dynamic fraction.




#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn_p<br>DINT | Input signal | |
| bAct_b<br>BOOL | Conversion mode | |
| | FALSE | $dnOut\_p = Remainder \cdot \frac{nNum}{wDenom}$ |
| | TRUE | $dnOut\_p = dnIn\_p \cdot \frac{nNum}{wDenom}$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

#### Outputs

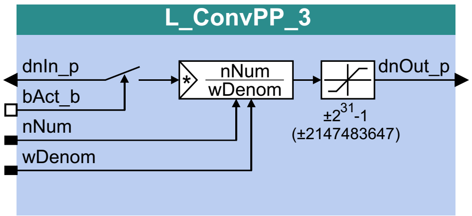
| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

**Note!**

Conversion <u>is</u> remainder considered.

### 18.1.47 L_ConvPP_3

This FB converts a position with dynamic fraction.




#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn_p<br>DINT | Input signal | |
| bAct_b<br>BOOL | Conversion mode | |
| | FALSE | $\mathrm{dnOut\_p} = \mathrm{Remainder} \cdot \dfrac{\mathrm{nNum}}{\mathrm{wDenom}}$ |
| | TRUE | $\mathrm{dnOut\_p} = \mathrm{dnIn\_p} \cdot \dfrac{\mathrm{nNum}}{\mathrm{wDenom}}$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

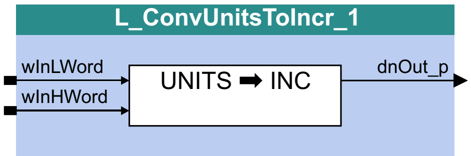
**Note!**

Conversion is remainder considered.

## 18.1.48 L_ConvUnitsToIncr_1

FB in preparation!

Taking into account the machine parameters, this FB converts a position value provided in the real unit of the machine into an internal 32-bit position value.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| wInLWord<br>WORD | Input signal Low Word |
| wInHWord<br>WORD | Input signal High Word |

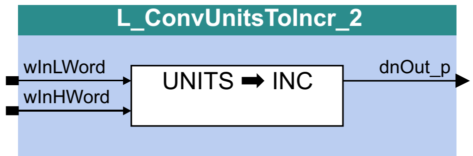
### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

## 18.1.49 L_ConvUnitsToIncr_2

FB in preparation!

Taking into account the machine parameters, this FB converts a position value provided in the real unit of the machine into an internal 32-bit position value.




### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| wInLWord<br>WORD | Input signal Low Word |
| wInHWord<br>WORD | Input signal High Word |

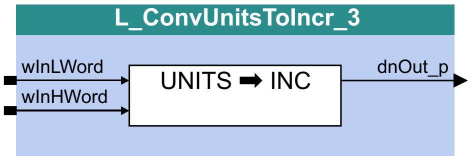
### Outputs

| Identifier / Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

### 18.1.50 L_ConvUnitsToIncr_3

FB in preparation!

Taking into account the machine parameters, this FB converts a position value provided in the real unit of the machine into an internal 32-bit position value.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| wInLWord<br>WORD | Input signal Low Word |
| wInHWord<br>WORD | Input signal High Word |

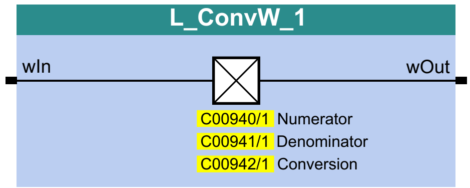
#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

### 18.1.51 L_ConvW_1

This FB serves to convert analog signal forms. The following conversions per parameter can be selected:

- [%] ➞ [incr/ms]
- [incr/ms] ➞ [%]
- Conversion via parameterisable factors
- Input signal is passed through without conversion




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| wIn<br>WORD | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00940/1 | -2147483647 | | 2147483647 | Numerator<br>• Lenze setting: 1 |
| C00941/1 | -2147483647 | | 2147483647 | Denominator<br>• Lenze setting: 1 |
| C00942/1 | | | | Selection of the conversion |
| | 0 | *wOut* = *wIn* (no conversion) | | |
| | 1 | [%] ➞ [incr/ms] | | |
| | 2 | [incr/ms] ➞ [%] | | |
| | 3 | *wOut* = *wIn* * C00940 / C00941 | | |

### Conversion formulae

**Note!**

Division is remainder considered.

$$\text{wOut [incr/ms]} = \frac{\text{wIn [\%]} \cdot \text{C00011 [rpm]} \cdot \text{65536 [incr/rev.]}}{\text{100 \%} \cdot \text{60 [s/min]} \cdot \text{1000 [ms/s]}}$$

[18-18] Conversion formula for selection 1: [%] ➞ [incr/ms]

$$\text{wOut [\%]} = \frac{\text{wIn [incr/ms]} \cdot \text{100 \%} \cdot \text{60 [s/min]} \cdot \text{1000 [ms/s]}}{\text{C00011 [rpm]} \cdot \text{65536 [incr/rev.]}}$$

[18-19] Conversion formula for selection 2: [incr/ms] ➞ [%]

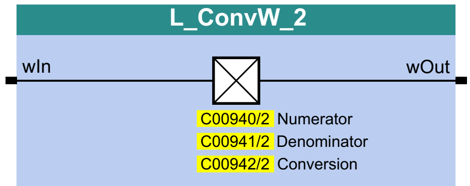
$$\text{wOut} = \text{wIn} \cdot \frac{\text{C00940}}{\text{C00941}}$$

[18-20] Conversion formula for selection 3: Parameterisable factors

## 18.1.52 L_ConvW_2

This FB serves to convert analog signal forms. The following conversions per parameter can be selected:

- [%] ➞ [incr/ms]
- [incr/ms] ➞ [%]
- Conversion via parameterisable factors
- Input signal is passed through without conversion




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| wIn<br>WORD | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

### Parameter

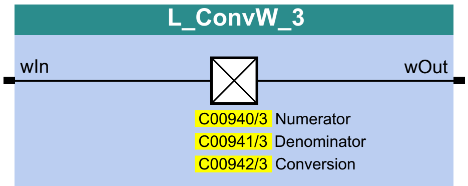
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00940/2 | -2147483647 | | 2147483647 | Numerator<br>• Lenze setting: 1 |
| C00941/2 | -2147483647 | | 2147483647 | Denominator<br>• Lenze setting: 1 |
| C00942/2 | | | | Selection of the conversion |
| | 0 | *wOut* = *wIn* (no conversion) | | |
| | 1 | [%] ➞ [incr/ms] | | |
| | 2 | [incr/ms] ➞ [%] | | |
| | 3 | *wOut* = *wIn* * C00940 / C00941 | | |

For conversion formulae see L_ConvW_1.

### 18.1.53 L_ConvW_3

This FB serves to convert analog signal forms. The following conversions per parameter can be selected:

- [%] ➞ [incr/ms]
- [incr/ms] ➞ [%]
- Conversion via parameterisable factors
- Input signal is passed through without conversion




#### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| wIn / WORD | Input signal |

#### Outputs

| Identifier / Data type | Value/meaning |
|---|---|
| wOut / WORD | Output signal |

#### Parameter

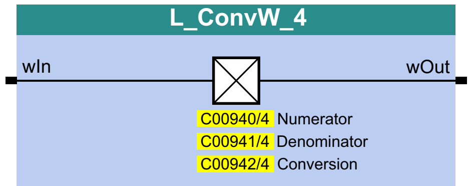
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00940/3 | -2147483647 | | 2147483647 | Numerator<br>• Lenze setting: 1 |
| C00941/3 | -2147483647 | | 2147483647 | Denominator<br>• Lenze setting: 1 |
| C00942/3 | | | | Selection of the conversion |
| | 0 | *wOut* = *wIn* (no conversion) | | |
| | 1 | [%] ➞ [incr/ms] | | |
| | 2 | [incr/ms] ➞ [%] | | |
| | 3 | *wOut* = *wIn* * C00940 / C00941 | | |

For conversion formulae see L_ConvW_1.

## 18.1.54 L_ConvW_4

This FB serves to convert analog signal forms. The following conversions per parameter can be selected:

- [%] ➞ [incr/ms]
- [incr/ms] ➞ [%]
- Conversion via parameterisable factors
- Input signal is passed through without conversion




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| wIn<br>WORD | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

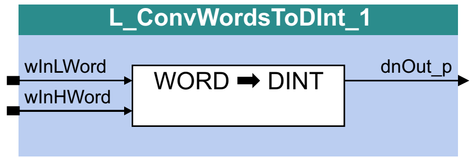
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00940/4 | -2147483647 | | 2147483647 | Numerator<br>• Lenze setting: 1 |
| C00941/4 | -2147483647 | | 2147483647 | Denominator<br>• Lenze setting: 1 |
| C00942/4 | | | | Selection of the conversion |
| | 0 | *wOut* = *wIn* (no conversion) | | |
| | 1 | [%] ➞ [incr/ms] | | |
| | 2 | [incr/ms] ➞ [%] | | |
| | 3 | *wOut* = *wIn* * C00940 / C00941 | | |

For conversion formulae see L_ConvW_1.

### 18.1.55 L_ConvWordsToDInt_1

This FB converts two input values of the type "WORD" into one output value of the type "DINT".




#### Inputs

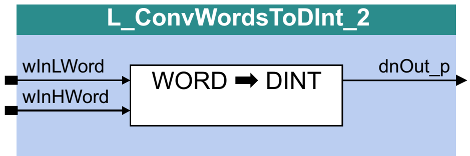
| Identifier<br>Data type | Information/possible settings |
|---|---|
| wInLWord<br>WORD | Input signal Low Word |
| wInHWord<br>WORD | Input signal High Word |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

### 18.1.56 L_ConvWordsToDInt_2

This FB converts two input values of the type "WORD" into one output value of the type "DINT".




#### Inputs

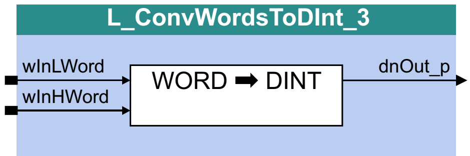
| Identifier<br>Data type | Information/possible settings |
|---|---|
| wInLWord<br>WORD | Input signal Low Word |
| wInHWord<br>WORD | Input signal High Word |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

## 18.1.57 L_ConvWordsToDInt_3

This FB converts two input values of the type "WORD" into one output value of the type "DINT".




### Inputs

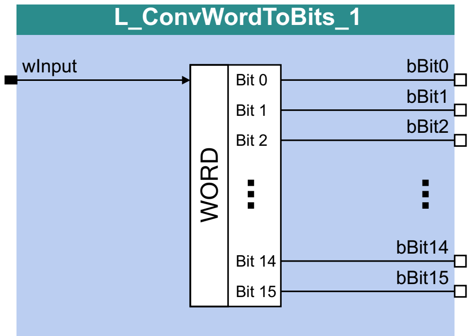
| Identifier<br>Data type | Information/possible settings |
|---|---|
| wInLWord<br>WORD | Input signal Low Word |
| wInHWord<br>WORD | Input signal High Word |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

### 18.1.58 L_ConvWordToBits_1

This FB converts an input value of "WORD" type into 16 individual binary signals.




#### Inputs

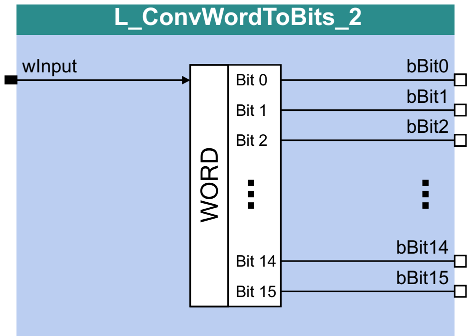
| Identifier<br>Data type | Information/possible settings |
|---|---|
| wInput<br>WORD | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Output signal |

## 18.1.59 L_ConvWordToBits_2

This FB converts an input value of "WORD" type into 16 individual binary signals.




### Inputs

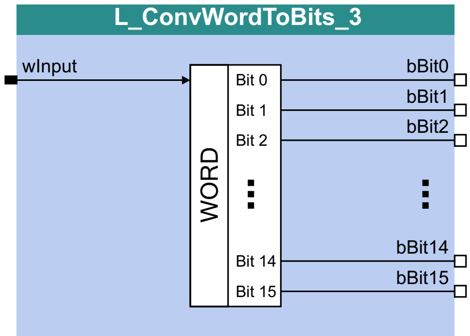
| Identifier<br>Data type | Information/possible settings |
|---|---|
| wInput<br>WORD | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Output signal |

## 18.1.60 L_ConvWordToBits_3

This FB converts an input value of "WORD" type into 16 individual binary signals.




### Inputs

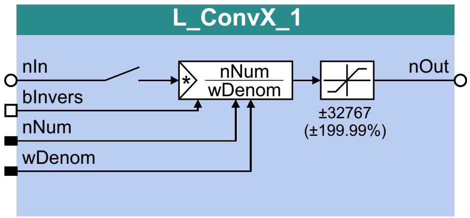
| Identifier<br>Data type | Information/possible settings |
|---|---|
| wInput<br>WORD | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bBit0<br>...<br>bBit15<br>BOOL | Output signal |

## 18.1.61 L_ConvX_1

This FB scales an analog value.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn<br>INT | Input signal | |
| bInvers<br>BOOL | Invert sign of the output signal | |
| | FALSE | $\text{nOut} = \text{nIn} \cdot \frac{\text{nNum}}{\text{wDenom}}$ |
| | TRUE | $\text{nOut} = \text{nIn} \cdot \frac{\text{nNum}}{\text{wDenom}} \cdot (-1)$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

### Outputs

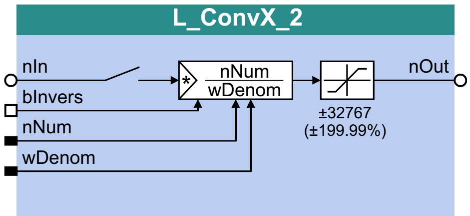
| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

**Note!**

Conversion is remainder considered.

## 18.1.62 L_ConvX_2

This FB scales an analog value.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn<br>INT | Input signal | |
| bInvers<br>BOOL | Invert sign of the output signal | |
| | FALSE | $nOut = nIn \cdot \frac{nNum}{wDenom}$ |
| | TRUE | $nOut = nIn \cdot \frac{nNum}{wDenom} \cdot (-1)$ |
| nNum<br>INT | Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767 | |
| wDenom<br>WORD | Factor (denominator)<br>• Internal limitation to 1 ... 32767 | |

### Outputs

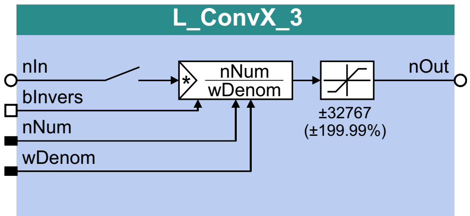
| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

**Note!**

Conversion is remainder considered.

## 18.1.63 L_ConvX_3

This FB scales an analog value.




### Inputs

<table>
<tr><th>Identifier<br>Data type</th><th colspan="2">Information/possible settings</th></tr>
<tr><td>nIn<br>INT</td><td colspan="2">Input signal</td></tr>
<tr><td rowspan="3">bInvers<br>BOOL</td><td colspan="2">Invert sign of the output signal</td></tr>
<tr><td>FALSE</td><td>$\text{nOut} = \text{nIn} \cdot \frac{\text{nNum}}{\text{wDenom}}$</td></tr>
<tr><td>TRUE</td><td>$\text{nOut} = \text{nIn} \cdot \frac{\text{nNum}}{\text{wDenom}} \cdot (-1)$</td></tr>
<tr><td>nNum<br>INT</td><td colspan="2">Factor (numerator)<br>• Internal limitation to -32767 ... -1 / 1 ... 32767</td></tr>
<tr><td>wDenom<br>WORD</td><td colspan="2">Factor (denominator)<br>• Internal limitation to 1 ... 32767</td></tr>
</table>

### Outputs

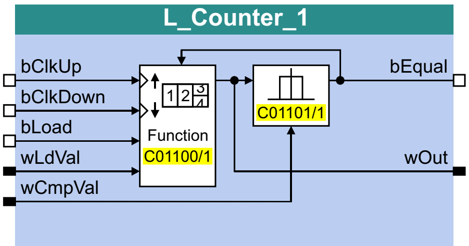
| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |

**Note!**

Conversion is remainder considered.

## 18.1.64 L_Counter_1

This FB is a digital upcounter and downcounter with a parameterisable comparison operation.




### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| bClkUp<br>BOOL | Clock input<br>• With each edge, the module counts up by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bClkDown<br>BOOL | Clock input<br>• With each edge, the module counts down by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bLoad<br>BOOL | Load input<br>• The input has the highest priority.<br>TRUE: Accept starting value *wLdVal*. |
| wLdVal<br>WORD | Starting value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |
| wCmpVal<br>WORD | Comparison value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |

### Outputs

| Identifier / Data type | Value/meaning |
|---|---|
| bEqual<br>BOOL | Status signal "Comparison statement is true"<br>• The TRUE output is active in the Lenze setting if the current counter content is greater than or equal to the comparison value *wCmpVal*.<br>TRUE: The statement of the comparison mode selected in C01101/1 is true. |
| wOut<br>WORD | Counter content<br>• Internal limitation to ± 32767<br>• The most significant bit determines the sign! |

### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01100/1 | | | Function selection<br>• Lenze setting: Normal counting |
| | 0 | Normal counting | |
| | 1 | Auto reset | |
| | 2 | Manual reset | |
| C01101/1 | | | Selection of comparison operation<br>• Lenze setting: Counter content ≥ comparison value |
| | 0 | Counter content ≥ comparison value | |
| | 1 | Counter content ≤ comparison value | |
| | 2 | Counter content = comparison value | |

### General function

- Every FALSE/TRUE edge at the *bClkUp* input causes the block to count upwards by "1".
- Every FALSE/TRUE edge at the *bClkDown* input causes the block to count downwards by "1".

### Function "Normal counting"

If the statement of the comparison mode selected in C01101/1 is true, the *bCompare* output is set to TRUE.

### Function "Auto reset"

If the statement of the comparison mode selected in C01101/1 is true, the *bCompare* output is set to TRUE for 1 ms and the counter is reset to the *wLdVal* starting value.

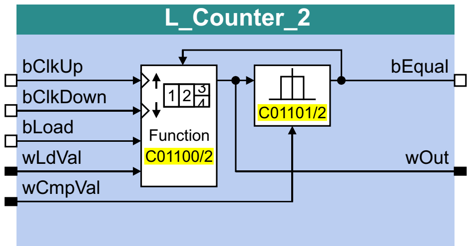
### Function "Manual reset"

If the statement of the comparison mode selected in C01101/1 is true, the *bCompare* output is set to TRUE and the counter stops.

- Edges at *bClkUp* and *bClkDown* are ignored.
- The counter must be reset via the *bLoad* input.

### 18.1.65 L_Counter_2

This FB is a digital upcounter and downcounter with a parameterisable comparison operation.




#### Inputs

| Identifier / Data type | | Information/possible settings | |
|---|---|---|---|
| bClkUp | BOOL | Clock input<br>• With each edge, the module counts up by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. | |
| bClkDown | BOOL | Clock input<br>• With each edge, the module counts down by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. | |
| bLoad | BOOL | Load input<br>• The input has the highest priority. | |
| | | TRUE | Accept starting value *wLdVal*. |
| wLdVal | WORD | Starting value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. | |
| wCmpVal | WORD | Comparison value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. | |

#### Outputs

| Identifier / Data type | | Value/meaning | |
|---|---|---|---|
| bEqual | BOOL | Status signal "Comparison statement is true"<br>• The TRUE output is active in the Lenze setting if the current counter content is greater than or equal to the comparison value *wCmpVal*. | |
| | | TRUE | The statement of the comparison mode selected in C01101/2 is true. |
| wOut | WORD | Counter content<br>• Internal limitation to ± 32767<br>• The most significant bit determines the sign! | |

### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01100/2 | | | Function selection<br>• Lenze setting: Normal counting |
| | 0 | Normal counting | |
| | 1 | Auto reset | |
| | 2 | Manual reset | |
| C01101/2 | | | Selection of comparison operation<br>• Lenze setting: Counter content ≥ comparison value |
| | 0 | Counter content ≥ comparison value | |
| | 1 | Counter content ≤ comparison value | |
| | 2 | Counter content = comparison value | |

### General function

- Every FALSE/TRUE edge at the *bClkUp* input causes the block to count upwards by "1".
- Every FALSE/TRUE edge at the *bClkDown* input causes the block to count downwards by "1".

### Function "Normal counting"

If the statement of the comparison mode selected in C01101/2 is true, the *bCompare* output is set to TRUE.

### Function "Auto reset"

If the statement of the comparison mode selected in C01101/2 is true, the *bCompare* output is set to TRUE for 1 ms and the counter is reset to the *wLdVal* starting value.

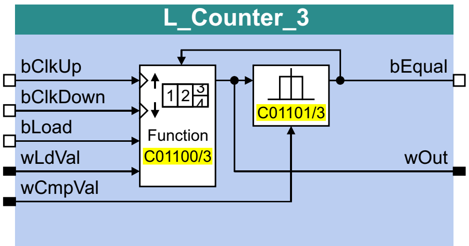
### Function "Manual reset"

If the statement of the comparison mode selected in C01101/2 is true, the *bCompare* output is set to TRUE and the counter stops.

- Edges at *bClkUp* and *bClkDown* are ignored.
- The counter must be reset via the *bLoad* input.

## 18.1.66 L_Counter_3

This FB is a digital upcounter and downcounter with a parameterisable comparison operation.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bClkUp<br>BOOL | Clock input<br>• With each edge, the module counts up by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bClkDown<br>BOOL | Clock input<br>• With each edge, the module counts down by "1".<br>• Only FALSE-TRUE edges are evaluated.<br>Note: The static state "1" is not permissible at this input. |
| bLoad<br>BOOL | Load input<br>• The input has the highest priority.<br>TRUE: Accept starting value *wLdVal*. |
| wLdVal<br>WORD | Starting value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |
| wCmpVal<br>WORD | Comparison value<br>• Assigned value is internally interpreted as "INT" data type (-32767 ... +32767), i.e. the most significant bit determines the sign. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bEqual<br>BOOL | Status signal "Comparison statement is true"<br>• The TRUE output is active in the Lenze setting if the current counter content is greater than or equal to the comparison value *wCmpVal*.<br>TRUE: The statement of the comparison mode selected in C01101/3 is true. |
| wOut<br>WORD | Counter content<br>• Internal limitation to ± 32767<br>• The most significant bit determines the sign! |

### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01100/3 | | | Function selection<br>• Lenze setting: Normal counting |
| | 0 | Normal counting | |
| | 1 | Auto reset | |
| | 2 | Manual reset | |
| C01101/3 | | | Selection of comparison operation<br>• Lenze setting: Counter content ≥ comparison value |
| | 0 | Counter content ≥ comparison value | |
| | 1 | Counter content ≤ comparison value | |
| | 2 | Counter content = comparison value | |

### General function

- Every FALSE/TRUE edge at the *bClkUp* input causes the block to count upwards by "1".
- Every FALSE/TRUE edge at the *bClkDown* input causes the block to count downwards by "1".

### Function "Normal counting"

If the statement of the comparison mode selected in C01101/3 is true, the *bCompare* output is set to TRUE.

### Function "Auto reset"

If the statement of the comparison mode selected in C01101/3 is true, the *bCompare* output is set to TRUE for 1 ms and the counter is reset to the *wLdVal* starting value.

### Function "Manual reset"

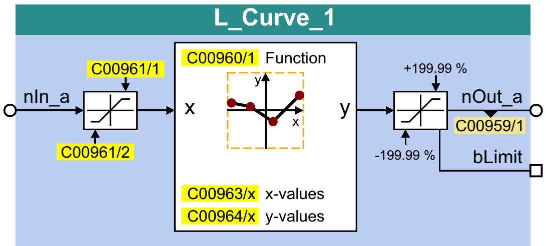
If the statement of the comparison mode selected in C01101/3 is true, the *bCompare* output is set to TRUE and the counter stops.

- Edges at *bClkUp* and *bClkDown* are ignored.
- The counter must be reset via the *bLoad* input.

### 18.1.67 L_Curve_1

This FB can optionally display a characteristic function or a curve function y=f(x), the X axis being the input signal and the Y axis being the output signal.

- A limitation of the input value can be set in C00961/1 (upper limit) and C00961/2 (lower limit).
- The output value is internally limited to ±199.99 %. If a limitation of the output value is active, *bLimit* is set to TRUE.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value (x)<br>• ±16384 ≡ ±100 %<br>• A limitation of the input value can be set in C00961/1 (upper limit) and C00961/2 (lower limit). |

#### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| nOut_a<br>INT | Output value (y)<br>• ±16384 ≡ ±100 %<br>• Internal limitation to ±32767 (±199.99 %) | |
| bLimit<br>BOOL | "Limitation active" status signal | |
| | TRUE | The output value is limited to ±199.99 %. |

#### Parameter

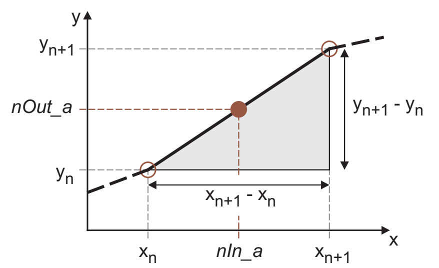
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00959/1<br>From version 02.00.00 | -199.99 | % | 199.99 | Output value<br>• Read only |
| C00960/1 | | | | Function selection |
| | 0 | *nOut_a* = 0 (FB deactivated) | | Lenze setting |
| | 1 | *nOut_a* = *nIn_a* | | |
| | 2 | *nOut_a* = f(x) | | In preparation! |
| | 3 | *nOut_a* = f(table) | | ▸ Function 3: nOut_a = f(table) |
| C00961/1 | -199.99 | % | 199.99 | Upper limit for input value<br>• Lenze setting: 199.99 % |

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00961/2 | -199.99 | % | 199.99 | Lower limit for input value<br>• Lenze setting: -199.99 % |
| C00963/1...32 | -32767 | | 32767 | X values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values X1 ... X32.<br>• Lenze setting: 0<br>▶ Function 3: nOut_a = f(table) |
| C00964/1...32 | -32767 | | 32767 | Y values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values Y1 ... Y32.<br>• Lenze setting: 0<br>▶ Function 3: nOut_a = f(table) |

## 18.1.67.1 Function 3: nOut_a = f(table)

If function 3 has been selected in C00960, the output value is calculated according to a characteristic.

- The characteristic may comprise up to 32 interpolation points which are defined by parameters.
- If the *nIn_a* input value is equal to one of the X interpolation points, the corresponding Y interpolation point will be output to *nOut_a*.
- If the *nIn_a* input value lies between two X interpolation points, the *nOut_a* output value will be interpolated linearly:




As per the equation,

$$\frac{x_{n+1} - x_n}{y_{n+1} - y_n} = \frac{nIn\_a - x_n}{nOut\_a - y_n}$$

output value *nOut_a* for *nIn_a* is calculated as follows:

$$nOut\_a = \frac{(nIn\_a - x_n) \cdot (y_{n+1} - y_n)}{x_{n+1} - x_n} + y_n$$

[18-21] Linear interpolation between two interpolation points

### Selecting the characteristic

The max. 32 interpolation points of the characteristic are selected via the subcodes of C00963 and C00964.

- The same subcodes of C00963 and C00964 correspond to one pair of variates/interpolation point ($x_n$, $y_n$).

| | 1 | 2 | ... | 32 |
|---|---|---|---|---|
| **x** | C00963/**1** | C00963/**2** | C00963/... | C00963/**32** |
| **y** | C00964/**1** | C00964/**2** | C00964/... | C00964/**32** |

- The first pair of variates (C00963/1 and C00964/1) is always valid.
- The X values of the characteristic must be entered in ascending order (x1 < x2 < ... < x32).

- An interruption of the ascending order of the X values represents the end of the characteristic (in our example: x5 < x4):

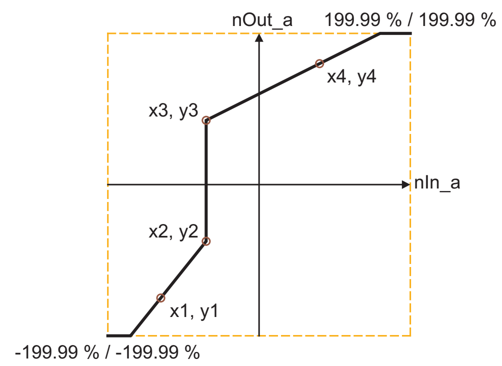
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... | 32 |
|---|---|---|---|---|---|---|---|---|---|
| **X** | -140 % | -90 % | 70 % | 145 % | 0 % | 0 % | 0 % | 0 % | 0 % |
| **y** | -140 % | -40 % | 115 % | 160 % | 0 % | 0 % | 0 % | 0 % | 0 % |

- Starting at the last valid pairs of variates, extrapolation is performed on both sides until the end of the X value range (±199.99 %).

### Exceptions

- Step changes/discontinuities can be displayed by using the same parameter setting for two consecutive X values. A step change is only valid if it has a valid pair of variates. Otherwise, extrapolation continues, starting at the last valid pair of variates.

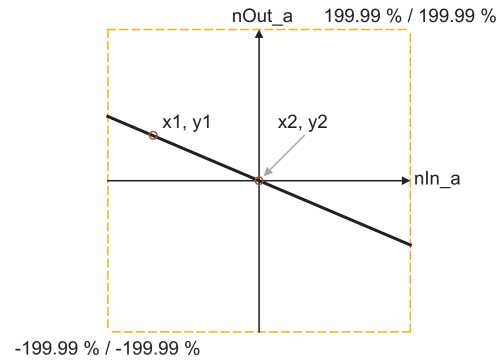
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... | 32 |
|---|---|---|---|---|---|---|---|---|---|
| **X** | -130 % | -70 % | -70 % | 80 % | 0 % | 0 % | 0 % | 0 % | 0 % |
| **y(x)** | -150 % | -75 % | 85 % | 160 % | 0 % | 0 % | 0 % | 0 % | 0 % |




[18-22] Example: Characteristic with step change

▶ If only one interpolation point, x1, in the negative range is defined, x2 exists with the "0 %" Lenze setting being the valid interpolation point (x2 > x1), and a line is drawn through both interpolation points:

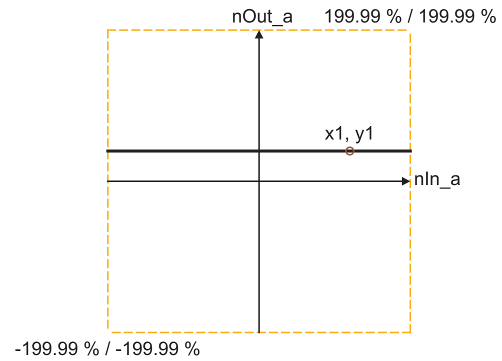
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| x | -130 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % |
| y(x) | 40 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % |




[18-23] Example: Characteristic with only one defined interpolation point and x1 < 0

▶ If only one interpolation point, x1, is defined in the positive range, a line is extrapolated:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| x | 120 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % |
| y(x) | 40 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % | 0 % |

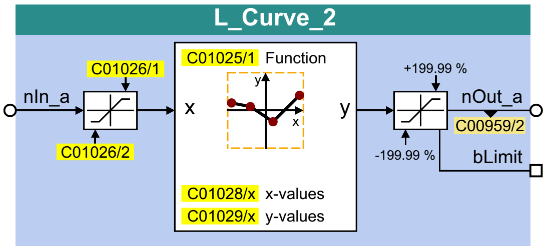



[18-24] Example: Characteristic with only one defined interpolation point and x1 < 0

## 18.1.68 L_Curve_2

This FB can optionally display a characteristic function or a curve function y=f(x), the X axis being the input signal and the Y axis being the output signal.

- A limitation of the input value can be set in C01026/1 (upper limit) and C01026/2 (lower limit).
- The output value is internally limited to ±199.99 %. If a limitation of the output value is active, *bLimit* is set to TRUE.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value (x)<br>• ±16384 ≡ ±100 %<br>• A limitation of the input value can be set in C01026/1 (upper limit) and C01026/2 (lower limit). |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| nOut_a<br>INT | Output value (y)<br>• ±16384 ≡ ±100 %<br>• Internal limitation to ±32767 (±199.99 %) | |
| bLimit<br>BOOL | "Limitation active" status signal | |
| | TRUE | The output value is limited to ±199.99 %. |

### Parameter

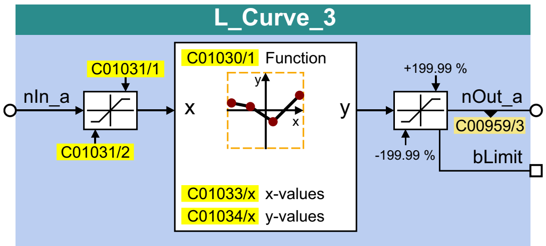
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00959/2 | -199.99 | % | 199.99 | Output value<br>• Read only |
| C001025/1 | | | | Function selection |
| | 0 | *nOut_a* = 0 (FB deactivated) | | Lenze setting |
| | 1 | *nOut_a* = *nIn_a* | | |
| | 2 | *nOut_a* = f(x) | | In preparation! |
| | 3 | *nOut_a* = f(table) | | See description of L_Curve_1:<br>Function 3: nOut_a = f(table) |
| C01026/1 | -199.99 | % | 199.99 | Upper limit for input value<br>• Lenze setting: 199.99 % |

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01026/2 | -199.99 | % | 199.99 | Lower limit for input value<br>• Lenze setting: -199.99 % |
| C01028/1...32 | -32767 | | 32767 | X values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values X1 ... X32.<br>• Lenze setting: 0<br>See description of L_Curve_1:<br>Function 3: nOut_a = f(table) |
| C01029/1...32 | -32767 | | 32767 | Y values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values Y1 ... Y32.<br>• Lenze setting: 0<br>See description of L_Curve_1:<br>Function 3: nOut_a = f(table) |

## 18.1.69 L_Curve_3

This FB can optionally display a characteristic function or a curve function y=f(x), the X axis being the input signal and the Y axis being the output signal.

- A limitation of the input value can be set in C01031/1 (upper limit) and C01031/2 (lower limit).
- The output value is internally limited to ±199.99 %. If a limitation of the output value is active, *bLimit* is set to TRUE.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input value (x)<br>• ±16384 ≡ ±100 %<br>• A limitation of the input value can be set in C01031/1 (upper limit) and C01031/2 (lower limit). |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| nOut_a<br>INT | Output value (y)<br>• ±16384 ≡ ±100 %<br>• Internal limitation to ±32767 (±199.99 %) | |
| bLimit<br>BOOL | "Limitation active" status signal | |
| | TRUE | The output value is limited to ±199.99 %. |

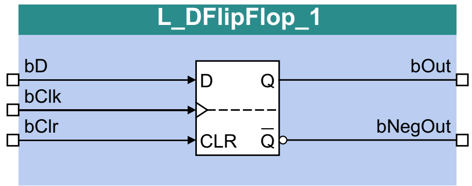
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00959/3 | -199.99 | % | 199.99 | Output value<br>• Read only |
| C01030/1 | | | | Function selection |
| | 0 | *nOut_a* = 0 (FB deactivated) | | Lenze setting |
| | 1 | *nOut_a* = *nIn_a* | | |
| | 2 | *nOut_a* = f(x) | | In preparation! |
| | 3 | *nOut_a* = f(table) | | See description of L_Curve_1:<br>Function 3: nOut_a = f(table) |
| C01031/1 | -199.99 | % | 199.99 | Upper limit for input value<br>• Lenze setting: 199.99 % |

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01031/2 | -199.99 | % | 199.99 | Lower limit for input value<br>• Lenze setting: -199.99 % |
| C01033/1...32 | -32767 | | 32767 | X values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values X1 ... X32.<br>• Lenze setting: 0<br>See description of L_Curve_1:<br>▶ Function 3: nOut_a = f(table) |
| C01034/1...32 | -32767 | | 32767 | Y values for characteristic function<br>• Subcodes 1 ... 32 correspond to interpolation point values Y1 ... Y32.<br>• Lenze setting: 0<br>See description of L_Curve_1:<br>▶ Function 3: nOut_a = f(table) |

## 18.1.70 L_DFlipFlop_1

The FB saves binary signals (DFlipFlop) in a clock-controlled way.




### Inputs

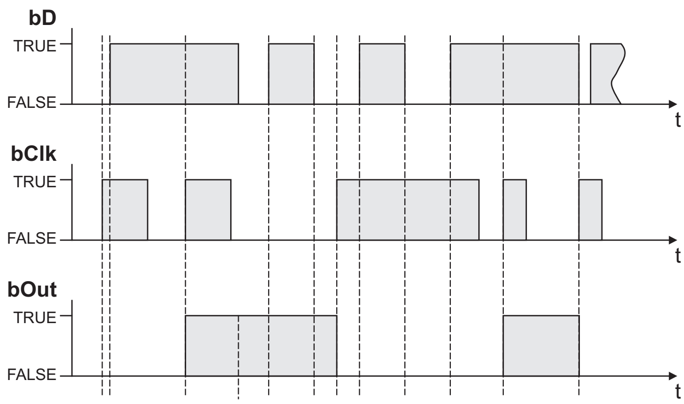
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bD<br>BOOL | Data input | |
| bClk<br>BOOL | Clock input<br>• Only FALSE/TRUE edges are evaluated | |
| bClr<br>BOOL | Reset input | |
| | TRUE | • The *bOut* output is set to FALSE.<br>• The *bNegOut* output is set to TRUE. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |
| bNegOut<br>BOOL | Output signal, inverted |

### Function

If the *bClr* input = FALSE, a signal edge at the *bClk* input switches the static input signal *bD* to the *bOut* output, where it is retained:

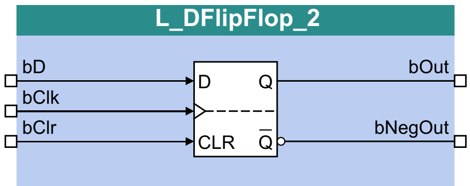


If the *bClr* input = TRUE:

- Due to the priority *bClr* > *bClk*, *bD* the *bOut* output signal can be set any time to the *FALSE* status by the *bClr* input signal = TRUE.
- The output signal is kept in this status independent of the other input signals.

## 18.1.71 L_DFlipFlop_2

The FB saves binary signals (DFlipFlop) in a clock-controlled way.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bD<br>BOOL | Data input |
| bClk<br>BOOL | Clock input<br>• Only FALSE/TRUE edges are evaluated |
| bClr<br>BOOL | Reset input<br>TRUE: • The *bOut* output is set to FALSE.<br>• The *bNegOut* output is set to TRUE. |

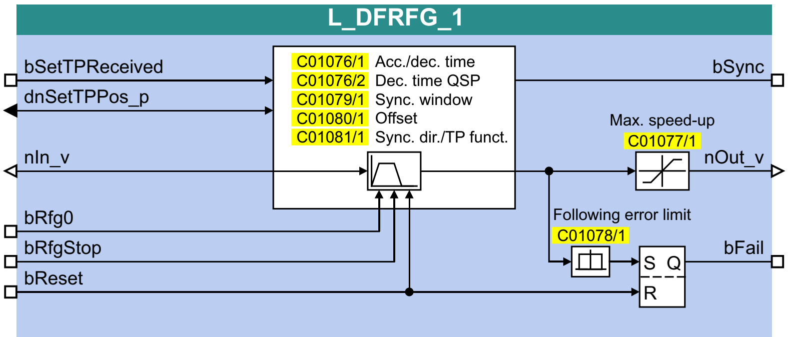
### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |
| bNegOut<br>BOOL | Output signal, inverted |

For a detailed functional description see L_DFlipFlop_1.

## 18.1.72 L_DFRFG_1

This FB synchronises a drive (slave) to the master value of a master drive and then executes a angular synchronism with regard to the digital frequency.




### Inputs

| Identifier / Data type | Information/possible settings | |
|---|---|---|
| bSetTPReceived (BOOL) | Input for status signal "Touch probe detected" | |
| | FALSE↗TRUE | Touch probe or zero pulse received. |
| dnSetTPPos_p (DINT) | Input for accepting the position measured via touch probe | |
| nIn_v (INT) | Speed setpoint of the master<br>• Scaling: 16384 ≡ 15000 rpm | |
| bRfg0 (BOOL) | Ramp down ramp function generator to "0" (quick stop function)<br>• This input can, for instance, be connected to the status signal "quick stop active". | |
| | TRUE | Ramp down the ramp function generator to "0"<br>• The drive is brought to a standstill with the deceleration time set in C01076/2. |
| bRfgStop (BOOL) | Stop ramp function generator | |
| | TRUE | Stop ramp function generator<br>• The last status is output to *nOut_v*.<br>• The speed/angle setpoint *nIn_v* is saved.<br>• After *bStop* is reset to FALSE, the setpoint angle is approached via the ramp function generator. |
| bReset (BOOL) | Reset added angle setpoint | |
| | TRUE | The internally added angle setpoint is reset and the ramp function generator is activated. If the status signal *bFail* is set, it will also be reset. |
| | TRUE↘FALSE | Detect speed/angle setpoint. |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bSync<br>BOOL | Status signal "Drive is running synchronously" | |
| | TRUE | The drive is running synchronously, the speed input id directly applied to the speed output. |
| nOut_v<br>INT | Speed/angle setpoint<br>• Scaling: 16384 ≡ 15000 rpm | |
| bFail<br>BOOL | Status signal "Angular difference exceeded" | |
| | TRUE | The angular difference set in C01078/1 has been exceeded. |

### Parameter

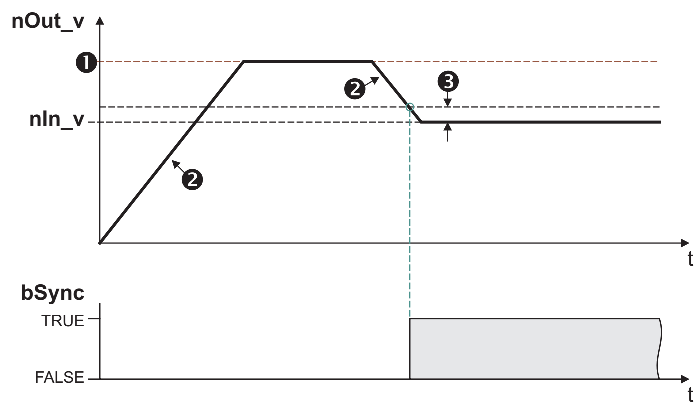
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01076/1 | 0.000 | s | 999.900 | Acceleration and deceleration time<br>• Relating to the reference speed (constant acceleration and deceleration) set in C00011.<br>• It is a maximum value which limits the acceleration of the ramp function generator.<br>• Lenze setting: 1.000 s |
| C01076/2 | 0.000 | s | 999.900 | Deceleration time for quick stop of the slave drive<br>• Relating to the reference speed (constant acceleration and deceleration) set in C00011.<br>• It is a maximum value which limits the deceleration of the ramp function generator.<br>• Lenze setting: 0.000 s |
| C01077/1 | 1 | rpm | 15000 | Max. speed-up<br>• Lenze setting: 3000 rpm |
| C01078/1 | 10 | Incr. | 2000000000 | Following error limit<br>• Scaling: a revolution is displayed with 65536 increments or steps.<br>• Lenze setting: 2000000000 incr. |
| C01079/1 | 0 | Incr. | 65535 | Synchronisation window (position)<br>• If the difference between the master and the slave is lower than the synchronisation window, the *bSync* status signal is set to TRUE.<br>• Lenze setting: 100 incr. |
| C01080/1 | -2147483647 | Incr. | 2147483647 | Offset<br>• Angular offset for the internal actual position of the FB<br>• Scaling: a revolution is displayed with 65536 increments or steps.<br>• Lenze setting: 0 incr. |

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01081/1 | | | Sync. direction/TP function<br>• Selection of the direction of rotation for synchronising<br>• With high offset values and low input speed, the drive may change its direction of rotation. In order to prevent this, a direction of rotation can be permanently defined via this parameter. |
| | 1 | cw/ccw - without TP (Lenze setting) | |
| | 2 | cw - without TP | |
| | 3 | ccw - without TP | |
| | 4 | cw/ccw - with TP | |
| | 5 | cw - with TP | |
| | 6 | ccw - with TP | |

### 18.1.72.1 Ramp function generator (profile generator)

The ramp function generator accelerates the slave drive to the input master speed.

- Synchronising starts when the FB receives a touch probe edge from the master axis (e.g. a conveying belt) at the *bSetTPReceived* input.
- When the slave reaches the setpoint speed of the master and the made up path difference is lower than the synchronisation window set in C01079/1, the *bSync* output is set to TRUE. At the same time, the FB switches the profile generator to inactive and outputs the *nIn_v* input speed to *nOut_v*.

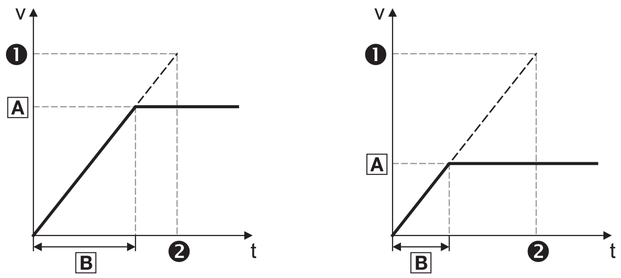



❶ C01077/1: Maximum speed
❷ C01076/1: Acceleration/deceleration time
❸ C01079/1: Synchronisation window

[18-25] Example: Synchronisation

- C01077/1 serves to set a maximum speed which limits the speed of the slave.

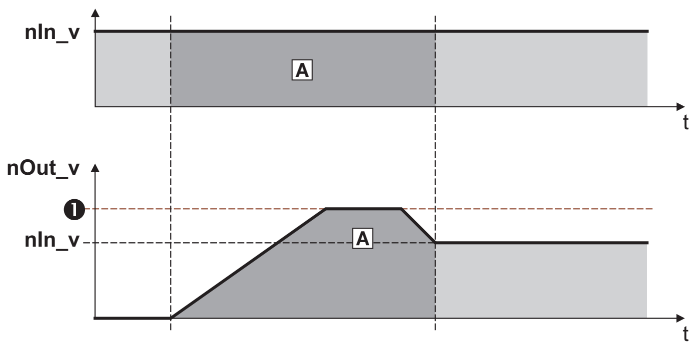
- The acceleration or deceleration in the synchronous point results from the acceleration/deceleration time set in C01076/1.
  - Reference for the acceleration/deceleration time is the reference speed (C00011):




❶ C00011: Reference speed
❷ C01076/1: Acceleration/deceleration time
A Effective target speed
B Required time

[18-26] Connection between acceleration time and acceleration

- Based on the input speed of the master axis, a setpoint angle is calculated from the starting time onwards which leads the actual angle of the slave.
- Dependent on the master speed and the settings for acceleration and offset, the FB may travel oversynchronously for reducing the angular difference, i.e. *nOut_v* is higher than *nIn_v*:




❶ C01077/1: Maximum speed
A Identical surface below the speed profile that represents the covered path.

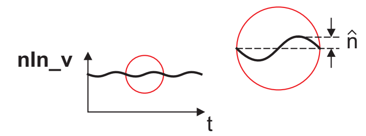
[18-27] Speed/time diagram

## Note!

- Set the maximum speed in C01077/1 higher than the master speed to be expected. The speed is selected on the motor side and is independent of the reference speed (C00011). The higher the difference between maximum speed and master speed, the less time to the synchronous time is required.
- In case of a heavily oscillating input speed it may occur that directly after setting the *bSync* status signal to TRUE the FB still executes slight angle corrections.
- Generally avoid acceleration or deceleration processes of the master axis while the slave axes are synchronising.

**Tip!**

Set the synchronisation window in C01079/1 higher than the amplitude of beat available on the input signal:

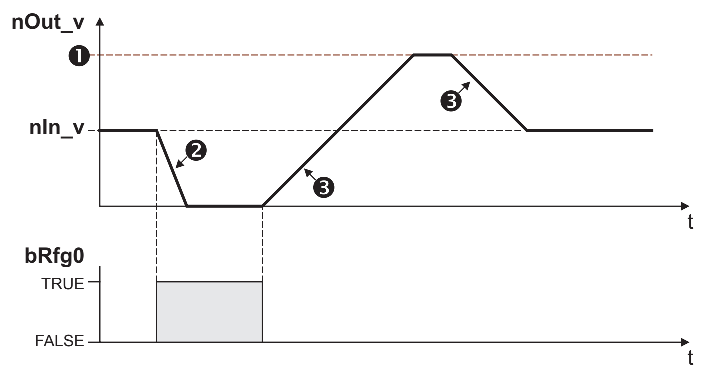



[18-28] Example: Beat on the input signal

### 18.1.72.2 Quick stop

The quick stop function removes the drive from the interconnection and leads it to standstill.

- Quick stop is activated by setting *bRfg0* to TRUE.
- C01076/2 is used to set the delay time in [s] which refers as the acceleration/deceleration time to the reference speed (C00011).
- The angle setpoint created internally on the basis of the input signal *nIn_v* is saved.
- After *bRfg0* is reset to FALSE, the angle setpoint is approached via the ramp function generator again.



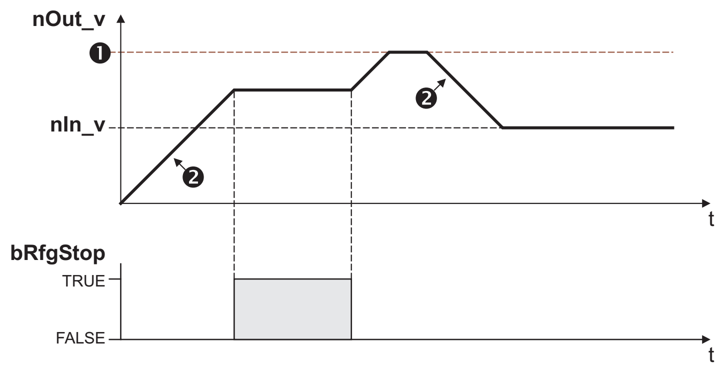

❶ C01077/1: Maximum speed
❷ C01076/2: Deceleration time for quick stop
❸ C01076/1: Acceleration/deceleration time

[18-29] Example: Quick stop function (QSP)

### 18.1.72.3 Ramp function generator stop

The stop function holds the status of the ramp function generator during operation.

- The stop function is activated by setting *bRfgStop* to TRUE.
- The last status is output to *nOut_v* (16384 ≡ 15000 rpm).
- The angle setpoint created internally on the basis of the input signal *nIn_v* is saved, thus the monitoring function of the angular difference between the created setpoint angle and actual angle cannot be activated.
- After *bRfgStop* is reset to FALSE, the angle setpoint is approached via the ramp function generator again.




❶ C01077/1: Maximum speed
❷ C01076/1: Acceleration/deceleration time

[18-30] Example: Stop function

### 18.1.72.4 Reset angle setpoint

By setting *bReset* to TRUE, the internally created angle setpoint is set to "0" and the ramp function generator is activated.

### 18.1.72.5 Detecting the angular difference

In C01078/1 , a limit value for monitoring the angular difference between the created setpoint and actual angles can be set.

- Scaling: a revolution is displayed with 65536 increments or steps.
- The ramp function generator can accept a angular difference of up to ±2140000000 increments (≈32000 revolutions).
- If monitoring responds, the status signal *bFail* is set to TRUE.
- If the internally added angle setpoint is reset by setting *bReset* to TRUE, the *bFail* status signal is reset to FALSE.

### 18.1.72.6 Offset setting

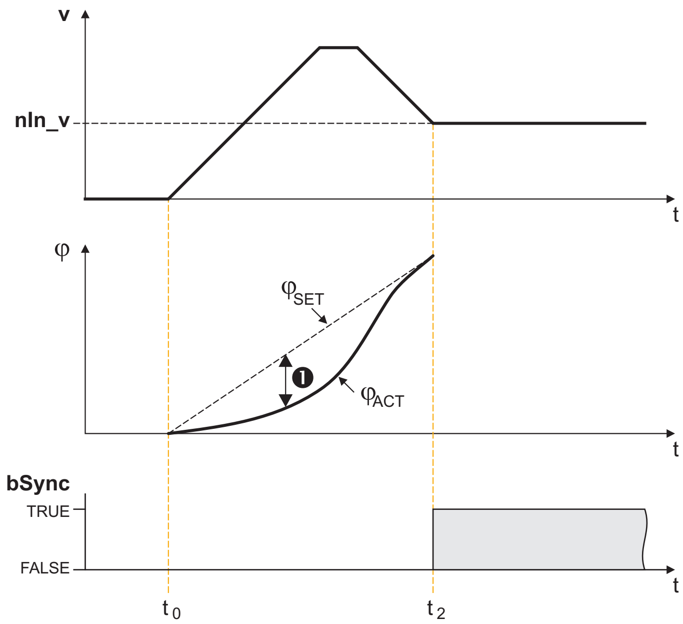
If the time to synchronicity is too long or the slave drive must not be operated oversynchronously, either the acceleration can be adapted or the touch probe initiator can be shifted. Since this is only possible to a limited extent due to construction conditions, in this case the better solution is selecting a virtual angular offset.

- In order to reach angle synchronicity, the difference between created setpoint and actual angle must be zero:

$$\varphi_{DIFF} = \varphi_{SET} - \varphi_{ACT} = 0$$

#### Behaviour without offset

If the actual angle starts with an offset = 0, a higher difference between setpoint angle and actual angle develops:

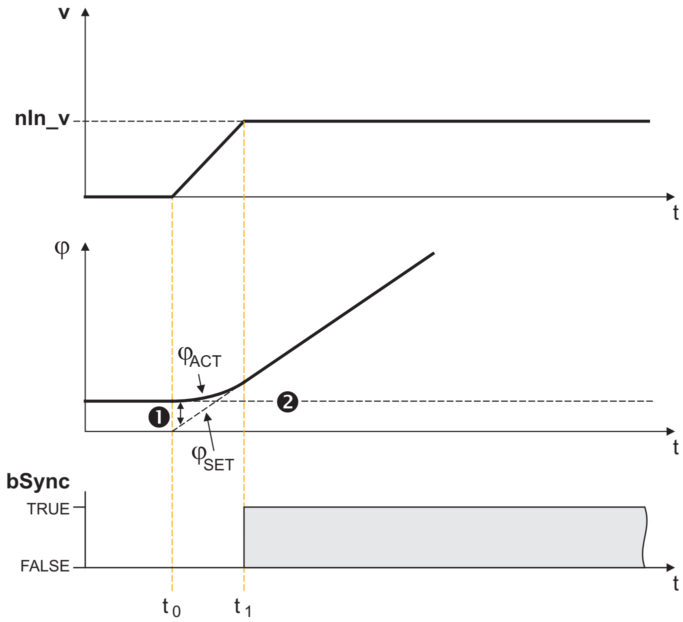



❶ Difference between setpoint angle and actual angle ($\varphi_{DIFF}$)
$t_0$ = touch probe event
$t_2$ = synchronicity reached

[18-31] Example: Without offset selection (offset = 0)

### Behaviour with offset

By selecting an offset, the actual position is preloaded with a non-zero value. This reduces the distance and the time to synchronicity.




❶ Difference between setpoint angle and actual angle ($\varphi_{DIFF}$)
❷ Offset [inc]
$t_0$ = touch probe event
$t_1$ = synchronicity achieved

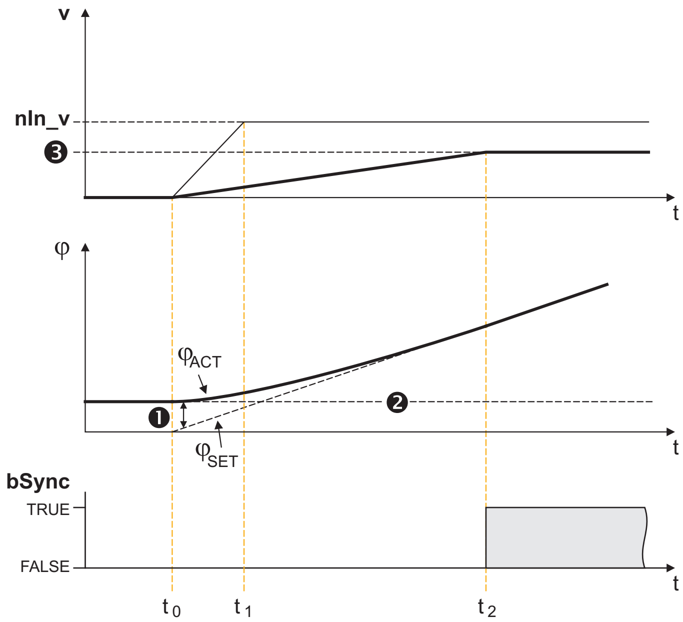
[18-32] Example 1: With offset selection

- When a positive offset is selected, the sign of the distance between setpoint and actual value changes. This distance is reduced due to the acceleration of the slave from the input of the touch probe pulse. Thus, the time to synchronicity is shorter.
- The offset refers to the master value selection and is scaled with an encoder revolution (≡ 65536 increments).
- The offset can be determined empirically, but it is sensible to calculate the required acceleration distance and select this value as offset:

$$\varphi_{ACC} = \frac{1}{2} \cdot \frac{(V_{SET})^2}{\text{C00011 [rpm]}} \cdot \text{C01076/1 [ms]} \cdot \frac{\text{16384 [incr./ms]}}{\text{15000 [rpm]}}$$

$\varphi_{ACC}$ = acceleration distance in [increments]
$v_{SET}$ = setpoint speed in [rpm]
C00011 = reference speed of the motor in [rpm]
C01076/1 = acceleration/deceleration time of the ramp generator in [ms]

The acceleration calculated from the acceleration/deceleration time (C01076/1) is a maximum acceleration that will not be reached if the input speed is reduced and the FB has to reach its target with a too high offset.




❶ Difference between setpoint angle and actual angle ($\varphi_{DIFF}$)
❷ Offset [inc]
❸ Reduced input speed
$t_0$ = touch probe event
$t_1$ = reaching synchronicity at original input speed
$t_2$ = synchronicity reached

[18-33] Example 2: With offset selection and reduced input speed

**Note!**

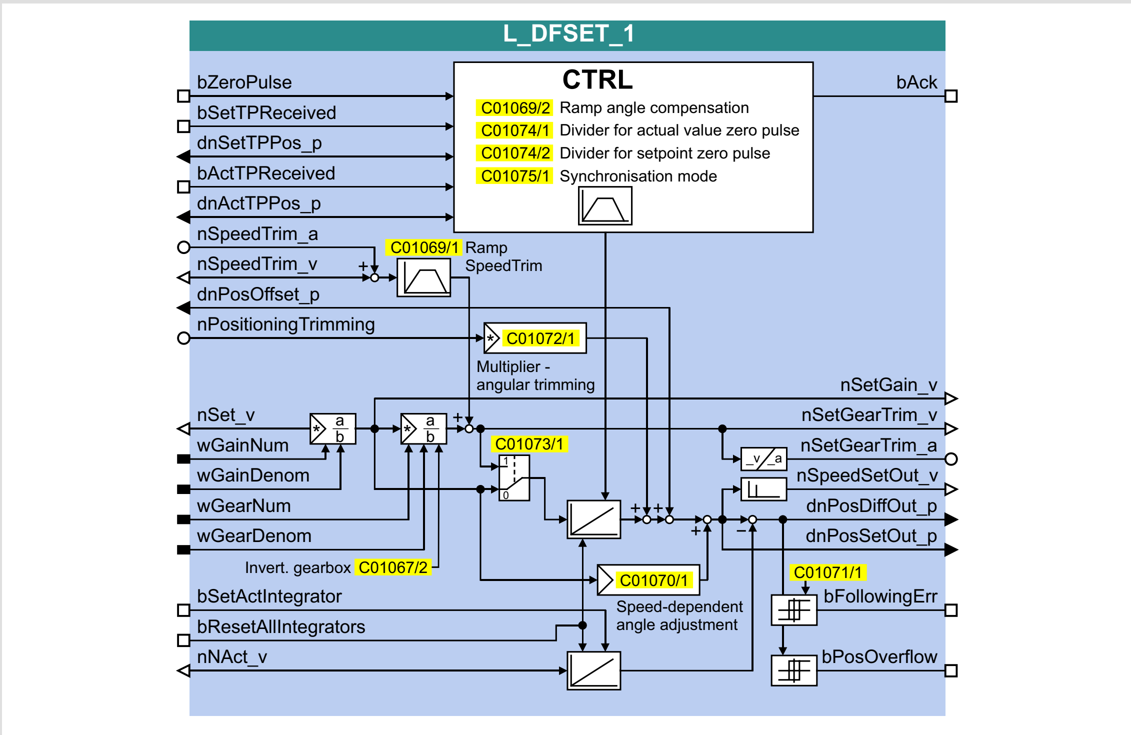
With a very low input speed, the acceleration can also get negative which causes the slave drive to change its direction of rotation.

In order to prevent a change of the direction of rotation, C01069/1 serves to select a permanent direction of rotation.

## 18.1.73 L_DFSET_1

This FB prepares the master value for a slave drive. This FB enables the controller to follow the master drive true to speed and angle.

- You can select values for the stretch factor and gearbox factor of the slave.
- A speed trimming or angular trimming is possible.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bZeroPulse<br>BOOL | Enable of the zero pulse-/touch probe synchronisation for the modes 2, 10 ... 13<br>• The mode is selected in C01075/1. | |
| | TRUE | Synchronisation enabled. |
| bSetTPReceived<br>BOOL | Input for status signal "Touch probe detected" | |
| | FALSE⇗TRUE | Touch probe or zero pulse received. |
| dnSetTPPos_p<br>DINT | Input for accepting the position measured via touch probe | |
| bActTPReceived<br>BOOL | Input for detecting an actual value touch probe or zero pulse | |
| | FALSE⇗TRUE | Actual value touch probe or zero pulse received. |
| dnActTPPos_p<br>DINT | Difference in [increments] between the occurrence of the actual value touch probe and the start of the task | |
| nSpeedTrim_a<br>INT | Speed trimming in [%]<br>• Scaling: 16384 ≡ 100 % | |
| nSpeedTrim_v<br>INT | Speed trimming in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm<br>• Speed trimming via this input is more precise. | |
| dnPosOffset_p<br>DINT | Angular offset in [increments]<br>• Scaling: a revolution is displayed with 65536 increments or steps. | |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nPositionTrimming<br>INT | Angular trimming in [increments]<br>• Scaling: a revolution is displayed with 65536 increments or steps.<br>• When analog values are selected: 100 % ≡ 1/4 revolution ≡ 16384 increments<br>• The setting range can be extended via C01072/1. | |
| nSet_v<br>INT | Speed setpoint<br>• Scaling: 16384 ≡ 15000 rpm | |
| wGainNum<br>WORD | Stretch factor (numerator) | |
| wGainDenom<br>WORD | Stretch factor (denominator) | |
| wGearNum<br>WORD | Gearbox factor (numerator) | |
| wGearDenom<br>WORD | Gearbox factor (denominator) | |
| bSetActIntegrator<br>BOOL | Equalise angle integrators (current position = set position)<br>• This input has a higher priority than the *bResetAllIntegrators* input. | |
| | FALSE↗TRUE | *dnPosDiffOut_p* = 0 |
| | TRUE | Equalise angle integrators. |
| bResetAllIntegrators<br>BOOL | Reset angle integrators<br>• This input has a higher priority than the *bResetAllIntegrators* input. | |
| | TRUE | Positional deviation, *dnPosSetOut_p* and *dnPosDiffOut_p* are set to "0". |
| nNAct_v<br>INT | Actual value in [increments/ms] for calculating the actual position<br>• Scaling: 16384 ≡ 15000 rpm | |

## Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bAck<br>BOOL | Status signal "Synchronising is executed" | |
| | TRUE | Synchronising is executed. |
| nSetGain_v<br>INT | Speed setpoint in [increments/ms] evaluated with stretch factor<br>• Scaling: 16384 ≡ 15000 rpm | |
| nSetGearTrim_v<br>INT | Speed setpoint in [increments/ms] evaluated with stretch factor and gearbox factor<br>• Scaling: 16384 ≡ 15000 rpm | |
| nSetGearTrim_a<br>INT | Speed setpoint in [%] evaluated with stretch factor and gearbox factor<br>• Scaling: 16384 ≡ 100 % reference speed (C00011) | |
| nSpeedSetOut_v<br>INT | Speed setpoint<br>• Derived from angle setpoint *dnPosSetOut_p*.<br>• Can change abruptly if e.g. the angular offset changes abruptly. | |
| dnPosDiffOut_p<br>DINT | Following error between set position and actual position<br>• This output can be used for display purposes. | |
| dnPosSetOut_p<br>DINT | Angle setpoint<br>• Scaling: a revolution is displayed with 65536 increments or steps. | |
| bFollowingErr<br>BOOL | Status signal "Following error" | |
| | TRUE | Following error occurred.<br>• The limit value set in C01071/1 has been exceeded. |
| bPosOverflow<br>BOOL | Status signal "angle controller overflow" | |
| | TRUE | Angle controller overflow occurred. |

### Parameter

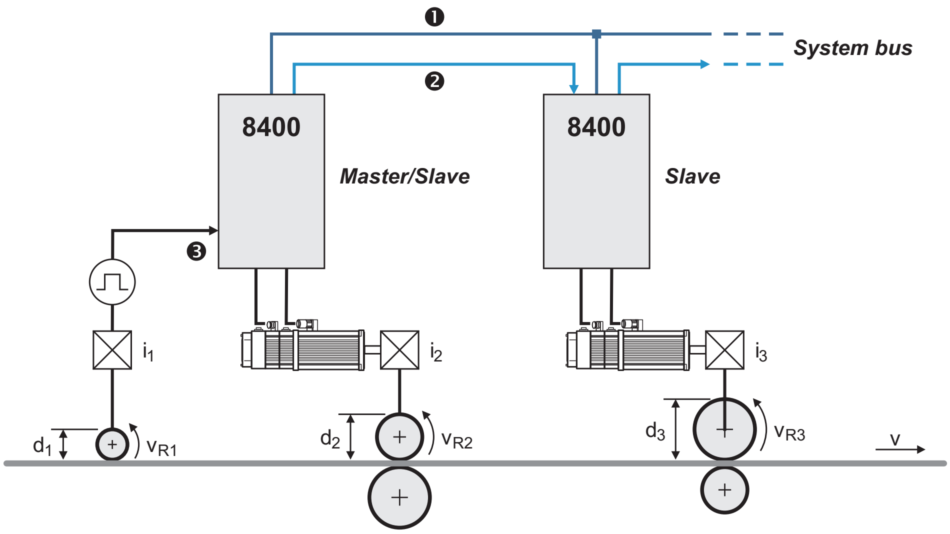
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01067/2 | | | | Invert. gearbox nSet_v |
| | 0 | not inverted | | |
| | 1 | inverted | | |
| | 2 | Automatically from MCK | | |
| C01069/1 | 1 | Incr./ms | 32767 | Ramp SpeedTrim<br>• Lenze setting: 10 incr./ms |
| C01069/2 | 1 | Incr./ms | 32767 | Ramp angle compensation<br>• Lenze setting: 100 incr./ms |
| C01070/1 | -134217728 | Incr. | 134217728 | Speed-dependent angle adjustment<br>• Scaling: a revolution is displayed with 65536 increments or steps.<br>• Lenze setting: 0 incr. |
| C01071/1 | 10 | Incr. | 2147483647 | Following error limit<br>• Scaling: a revolution is displayed with 65536 increments or steps.<br>• Lenze setting: 32768 incr. |
| C01072/1 | -20000 | | 20000 | Multiplier - angular trimming<br>• Lenze setting: 1 |
| C01073/1 | | | | Evaluation - setpoint integrator |
| | 0 | Evaluation with gearbox factor | | Lenze setting |
| | 1 | Evaluation without gearbox factor | | |
| C01074/1 | 0 | | 16384 | Divisor for actual value zero pulse<br>• Number of signals to be received at least until synchronisation gets active.<br>• Initialisation: 0 |
| C01074/2 | 0 | | 16384 | Divisor for setpoint zero pulse<br>• Number of signals to be received at least until synchronisation gets active.<br>• Initialisation: 0 |

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01075/1 | | | Synchronisation mode |
| | 0 | Synchronisation inactive | Lenze setting |
| | 1 | Permanent synchronisation without enable via the *bZeroPulse* input<br>• Correction is executed over the shortest possible path. | Synchronisation is only executed after 2 zero pulses have been received. |
| | 2 | Permanent synchronisation only with enable via the *bZeroPulse* input<br>• Correction is executed over the shortest possible path. | When *bZeroPulse* is set to TRUE, a permanent zero pulse/touch probe synchronisation is executed.<br>Synchronisation is only executed after 2 setpoint signals have been received. |
| | 10 | One-time synchronisation<br>• An angular difference is compensated over the shortest possible path. | Synchronisation is only executed after 2 setpoint signals have been received. |
| | 11 | One-time synchronisation<br>• An angle difference is compensated in CW direction. | Synchronisation is only executed after 2 setpoint signals have been received. |
| | 12 | One-time synchronisation<br>• An angular difference is compensated in CCW direction. | Synchronisation is only executed after 2 setpoint signals have been received. |
| | 13 | One-time synchronisation<br>• An angular difference is compensated over the shortest possible path. | Synchronisation is only executed after the 1. actual value signal has been received |

### 18.1.73.1 Master value rail/slave cascade

If there is only one master speed that is transmitted to all slave drives involved, e.g. via system bus (CAN), it is called master value rail.

If a slave drive takes over the master value generation for the following slave, it is called slave cascade.




❶ Master value rail
❷ Slave cascade
❸ Master value encoder

[18-34] Example: Master value rail/slave cascade

### 18.1.73.2 Setpoint conditioning with stretch factor and gearbox factor

#### Stretch factor

The stretch factor is required for the "speed synchronism via master value cascade" mode. It defines the ratio the slave drive is to be running with regard to its master value.

- The stretch factor evaluates the setpoint at the *nSet_v* input.
- The stretch factor must be selected via the *wGainNum* and *wGainDenom* inputs in the form of numerators and denominators.
- The result is provided at the *nSetGain_v* output.
  - Scaling: 16384 ≡ 15000 rpm

$$\text{nSetGain\_v} = \text{nSet\_v} \cdot \frac{\text{wGainNum}}{\text{wGainDenom}}$$

- If the stretch factor is 1 and the gearbox factors are selected correctly, the circumferential speeds of the rolls for master and slave 1 are identical in the example shown in the illustration [18-34].

#### Gearbox factor

The gearbox factor defines the gearbox ratio of the drive. Enter the ratio of the drive.

- The gearbox factor evaluates the setpoint at the *nSet_v* input multiplied by the stretch factor.
- The gearbox factor has to be selected via the inputs *wGearNum* and *wGearDenom* in the form of numerators and denominators.

- C01073/1 serves to simply deactivate the evaluation with the gearbox factor (does not apply to the speed outputs *nSetGain_v*, *nSetGearTrim_v* and *nSetGearTrim_a*).
- The result is provided at the outputs *nSetGearTrim_v* and *nSetGearTrim_a*.
  - Scaling *nSetGearTrim_v*: 16384 ≡ 15000 rpm
  - Scaling *nSetGearTrim_a*: 16384 ≡ 100 % reference speed (C00011)

$$\text{nSetGearTrim\_v} = \text{Stretch factor} \cdot \frac{\text{wGearNum}}{\text{wGearDenom}}$$

$$\text{nSetGearTrim\_v} = \text{nSet\_v} \cdot \frac{\text{wGainNum}}{\text{wGainDenom}} \cdot \frac{\text{wGearNum}}{\text{wGearDenom}}$$

$$\text{nSetGearTrim\_a} = \text{nSet\_v} \cdot \frac{\text{C00011}}{15000} \cdot \frac{\text{wGainNum}}{\text{wGainDenom}} \cdot \frac{\text{wGearNum}}{\text{wGearDenom}}$$

**Note!**

The intermediate results and the result of the evaluation are limited to ±32767 increments (16 bits).

If a limitation is active, increments and thus the real drive position will get lost!

When stretch factors are higher than "1", check if the FB gets in the limitation of calculation at maximum speed. If so, assign the required numerator and denominator values to the stretch and gearbox factors so that neither a limitation takes place in the end result nor in the intermediate result.

### 18.1.73.3 Processing correction values

#### Speed trimming

Speed trimming enables the connection of correction values, e.g. from a higher-level control loop. This permits an acceleration or deceleration of the drive.

The correction value can either be transmitted as an analog value via the *nSpeedTrim_a* input or for exact speed trimming as a speed value via the *nSpeedTrim_v* input.

- Scaling *nSpeedTrim_a*: 16384 ≡ 100 % reference speed (C00011)
- Scaling *nSpeedTrim_v*: 16384 ≡ 15000 rpm

**Stop!**

Change the speed trimming only step by step as the FB does not contain a ram generator for preventing torque impulses.

#### Angular trimming

Angular trimming enables the rotor position to be put forward or back with regard to the setpoint (the drive is leading or lagging).

The correction value must has to be defined in [increments] via the *nPositionTrimming* input. It is internally added to the angle setpoint.

- Scaling: a revolution is displayed with 65536 increments or steps.
- The maximum angular trimming amounts to ±½ motor revolution (±32767 increments).
- When analog values are selected: 100 % ≡ ¼ motor revolution ≡ 16384 increments.
- The setting range can be extended with the multiplier to be set in C00072/1.

### Angular offset

The *dnPosOffset_p* input serves to define a permanent angular offset for the setpoint of the drive.

- The maximum angular offset amounts to ±3750 motor revolutions.
- The adjustment is exclusively made via the following error (output *dnPosDiffOut_p*).
- Example: A angular offset of 90° is to be set on the load side with a gearbox ratio of 3.8147:

$$\text{dnPosOffset\_p} = \frac{90^\circ}{360^\circ} \cdot 65535\,[\text{increments}] \cdot 3.8147 = 62500\,[\text{increments}]$$

**Stop!**

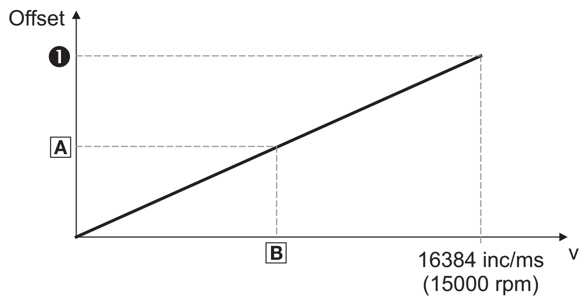
Change the angular offset only step by step as the FB does not contain a ram generator for preventing torque impulses.

Otherwise limit the jump by limiting the position controller output: Enter a maximum compensating speed in [%] at the *nPosCtrlOutLimit_a* input of the SB LS_MotorInterface with regard to the reference speed (C00011).

### Speed-proportional angle adjustment

The speed-proportional angle adjustment enables a leading or lagging of the angle with increasing speed and serves to compensate dead times and transmission times.

The correction value selected in C01070/1. in [increments] refers to a speed of 15000 rpm (linear relationship):




❶ C01070/1: Speed-proportional angle adjustment
A Effective offset
B Current speed

[18-35] Connection between offset/speed

#### 18.1.73.4 Synchronising slave drive to master drive

If a angular synchronism of the drives is required (e.g. in printing units), the slave drives must be synchronised to the master position since the FB **L_DFSET** as a master value can only receive and process on relative signal.

The FB **L_DFSET** receives the master value either via a real-time capable MotionBus as e.g. system bus (CAN) or via the master frequency channel that requires an additional module.

**Tip!**

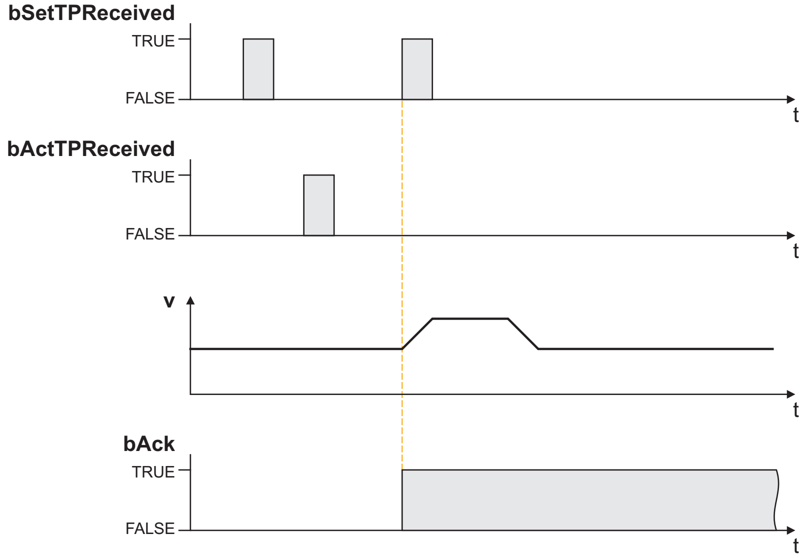
A master value cascade can be realised very easily via a master frequency connection of the drives.

For synchronisation purposes, the slave drive is informed cyclically or once about its drive position by a pulse generated by the master drive. This pulse is either the zero pulse of the master feedback or the edge of a touch probe sensor. Moreover, a pulse must be generated by the slave drive. Only if the FB **L_DFSET** receives both signals, it can execute a synchronisation.

**Synchronisation mode**

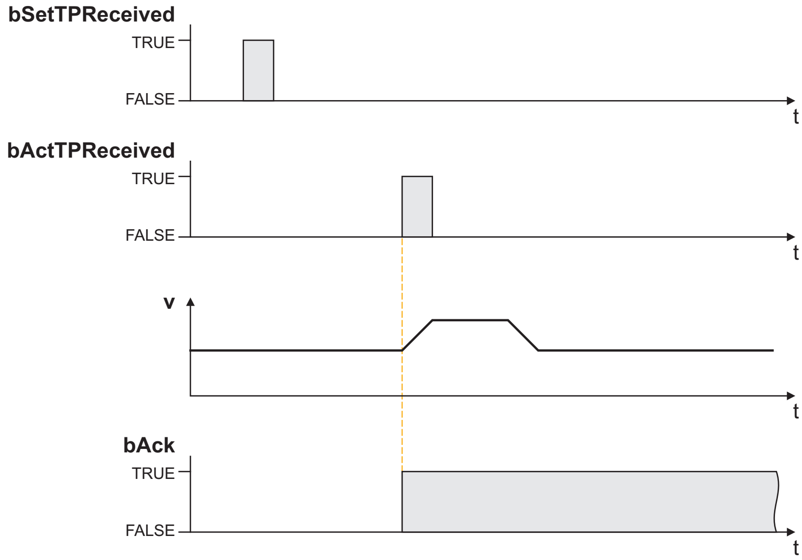
For synchronisation, the following modes are available in C01075:

| Mode | | Info |
|---|---|---|
| 0 | Synchronisation inactive | |
| 1 | Permanent synchronisation without enable via the *bZeroPulse* input<br>• Correction is executed over the shortest possible path. | The synchronisation starts at the earliest after two touch probe pulses of the master drive and one touch probe pulse of the slave drive have been received.<br>• See illustration [18-36]. (📖 1228) |
| 2 | Permanent synchronisation only with enable via the *bZeroPulse* input<br>• Correction is executed over the shortest possible path. | |
| 10 | One-time synchronisation only with enable via the *bZeroPulse* input<br>• An angular difference is compensated over the shortest possible path. | |
| 11 | One-time synchronisation only with enable via the *bZeroPulse* input<br>• An angle difference is compensated in CW direction. | |
| 12 | One-time synchronisation only with enable via the *bZeroPulse* input<br>• An angular difference is compensated in CCW direction. | |
| 13 | One-time synchronisation only with enable via the *bZeroPulse* input<br>• An angular difference is compensated over the shortest possible path. | The synchronisation starts at the earliest after two touch probe pulses of the master drive and one touch probe pulse of the master drive have been received.<br>• See illustration [18-37]. (📖 1228) |




Synchronisation mode (C01075) = 1, 2, 10, 11, 12
Divisor for actual value zero pulse (C01074/1) = 0
Divisor for setpoint zero pulse (C01074/2) = 0

[18-36] Synchronisation process in the modes 1, 2, 10, 11, 12




Synchronisation mode (C01075) = 13
Divisor for actual value zero pulse (C01074/1) = 0
Divisor for setpoint zero pulse (C01074/2) = 0

[18-37] Synchronisation process in mode 13

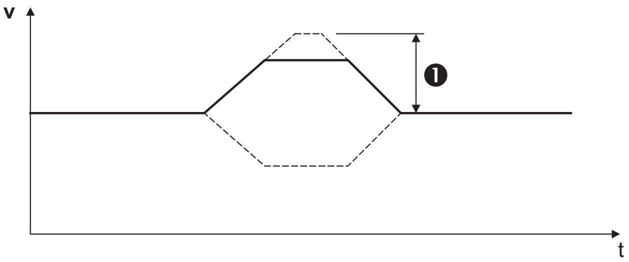
### Compensating movement

When the second set or act touch probe pulse is received, the difference between master and slave position is detected which is then communicated via the following error output to the motor control and then compensated via the position controller.

**Stop!**

The compensating movement is executed abruptly (with jerk) as the FB has no ramp generator for preventing torque pulses!

Limit the jump by limiting the position controller output: Enter a maximum compensating speed in [%] at the *nPosCtrlOutLimit_a* input of the SB LS_MotorInterface with regard to the reference speed (C00011).



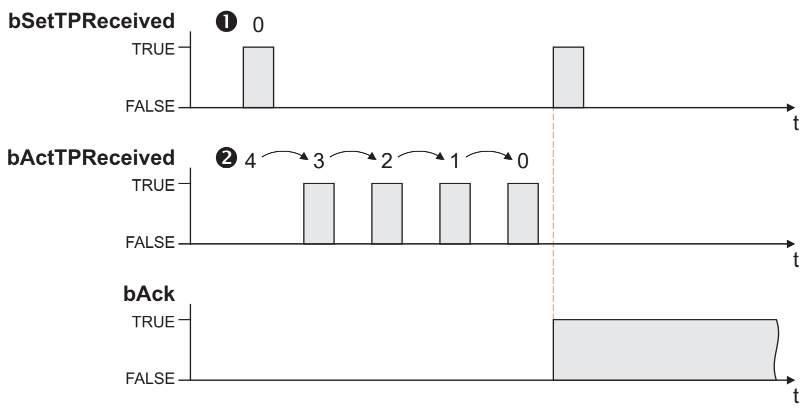

❶ h = f(Vp position, limitation of position controller output)

[18-38] Compensating movement

### 18.1.73.5 Masking out touch probe signals

When passing-through material is used, e.g. printed foil, touch probe initiators may respond several times per cycle. For suppressing such "interference pulses", count values can be selected in C01074/1 and C01074/2 that are decremented when the touch probe pulse has been received. Only when the counter content is "0", the synchronisation will be enabled.

Synchronisation mode = 1, 2, 10, 11, 12




❶ Divisor for setpoint zero pulse (C01074/2) = 0
❷ Divisor for actual value zero pulse (C01074/1) = 4

[18-39] Synchronisation process in the modes 1, 2, 10, 11, 12 with a masking out of the touch probe pulses

### 18.1.73.6 Process monitoring functions

#### Following error

The *bFollowingErr* status output is set to TRUE if the drive cannot follow its setpoint angle.

- Possible causes:
  - The centrifugal mass is too high for the set acceleration or deceleration time.
  - The torque limit has been reached (load torque > drive torque).
- Remedy: Unload drive or increase torque limit at the servo controller (if the power limits of the controller have not yet been reached).

The following error is derived from the angular difference of the setpoint angle integrator minus the actual angle integrator. The comparison value (following error limit) can be set in C01071/1.

#### Angle controller overflow (bPosOverflow = TRUE)

The *bPosOverflow* status output is set to TRUE if the angular difference that can be displayed device-internally has been exceeded. Home positions get lost here.

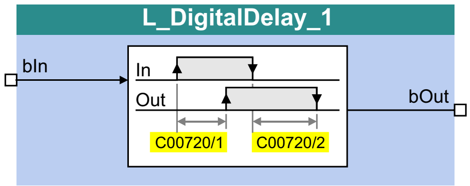
**Tip!**

If an error response is to be triggered in case of a following error and/or angle controller overflow, connect the corresponding status output with a free *bSetError* input of the SB LS_SetError_2 and parameterise the requested error response for this input in C00581.

## 18.1.74 L_DigitalDelay_1

This FB delays binary signals.

- The ON and OFF delays can be parameterised independently of one another.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input signal |

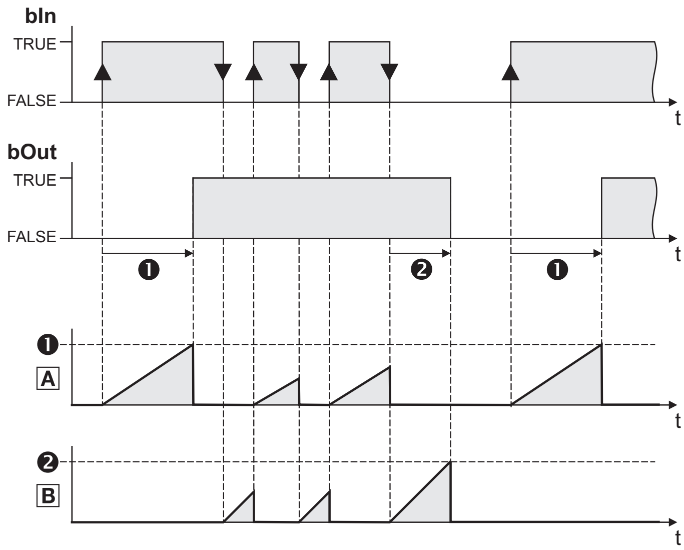
### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal (time-delayed input signal) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00720/1 | 0.000 | s | 3600.000 | ON delay<br>• Lenze setting: 0.000 s |
| C00720/2 | 0.000 | s | 3600.000 | OFF delay<br>• Lenze setting: 0.000 s |

### Function



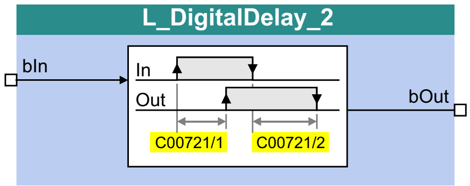

❶ ON delay
❷ OFF delay
A Internal timing element for ON delay (retriggerable)
B Internal timing element for OFF delay (retriggerable)

1. A FALSE-TRUE edge at b*In* starts the internal timing element for the ON delay.
2. After the defined ON delay, the input signal *bIn* is output at *bOut*.
3. A TRUE-FALSE edge at b*In* starts the internal timing element for the OFF delay.
4. After the defined OFF delay, the input signal *bIn* is output at *bOut*.

## 18.1.75 L_DigitalDelay_2

This FB delays binary signals.

▶ The ON and OFF delays can be parameterised independently of one another.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal (time-delayed input signal) |

### Parameter

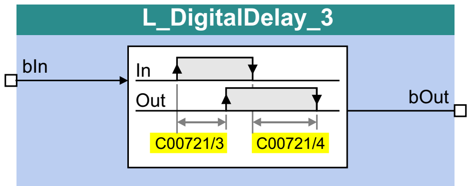
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00721/1 | 0.000 | s | 3600.000 | ON delay<br>• Lenze setting: 0.000 s |
| C00721/2 | 0.000 | s | 3600.000 | OFF delay<br>• Lenze setting: 0.000 s |

For a detailed functional description see L_DigitalDelay_1.

### 18.1.76 L_DigitalDelay_3

This FB delays binary signals.

- The ON and OFF delays can be parameterised independently of one another.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal (time-delayed input signal) |

#### Parameter

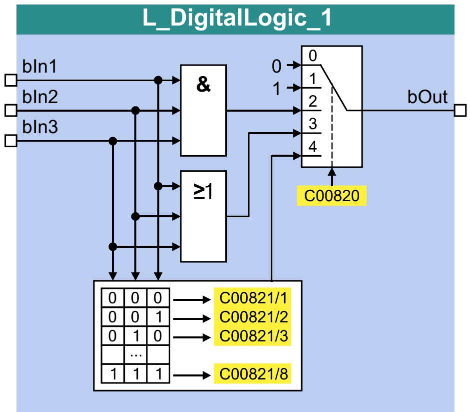
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00721/3 | 0.000 | s | 3600.000 | ON delay<br>• Lenze setting: 0.000 s |
| C00721/4 | 0.000 | s | 3600.000 | OFF delay<br>• Lenze setting: 0.000 s |

For a detailed functional description see L_DigitalDelay_1.

## 18.1.77 L_DigitalLogic_1

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- Logical ANDing of the inputs
- Logical ORing of the inputs
- Output depending on the combination of the input signals




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>BOOL | Input signal 1 |
| bIn2<br>BOOL | Input signal 2 |
| bIn3<br>BOOL | Input signal 3 |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00820 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | $bOut = bIn1 \wedge bIn2 \wedge bIn3$ | AND operation |
| | 3 | $bOut = bIn1 \vee bIn2 \vee bIn3$ | OR operation |
| | 4 | bOut = f (truth table) | The output value depends on the parameterised truth table |
| C00821 | see truth table | | Truth table<br>Each of the 8 possible input combinations can be assigned to the output value FALSE or TRUE. |

### Truth table for C00820 = 4

| bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|
| FALSE | FALSE | FALSE | C00821/1 (FALSE or TRUE) |
| FALSE | FALSE | TRUE | C00821/2 (FALSE or TRUE) |
| FALSE | TRUE | FALSE | C00821/3 (FALSE or TRUE) |
| FALSE | TRUE | TRUE | C00821/4 (FALSE or TRUE) |
| TRUE | FALSE | FALSE | C00821/5 (FALSE or TRUE) |
| TRUE | FALSE | TRUE | C00821/6 (FALSE or TRUE) |
| TRUE | TRUE | FALSE | C00821/7 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | C00821/8 (FALSE or TRUE) |

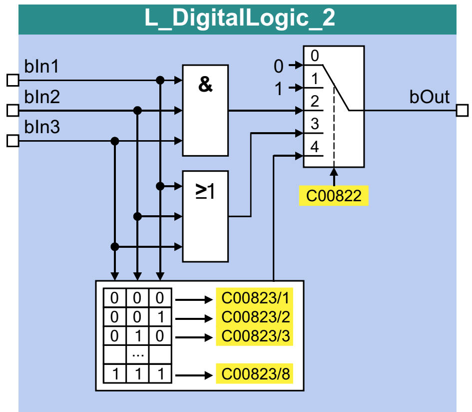
Example: If in case of the signal combination *bIn1* = FALSE, *bIn2* = FALSE and *bIn3* = TRUE, the output signal *bOut* is to be = TRUE, C00821/5 must be set to "TRUE":

| bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|
| TRUE | FALSE | FALSE | C00821/5 (TRUE) |

## 18.1.78 L_DigitalLogic_2

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- Logical ANDing of the inputs
- Logical ORing of the inputs
- Output depending on the combination of the input signals




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>BOOL | Input signal 1 |
| bIn2<br>BOOL | Input signal 2 |
| bIn3<br>BOOL | Input signal 3 |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

## Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00822 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | $bOut = bIn1 \wedge bIn2 \wedge bIn3$ | AND operation |
| | 3 | $bOut = bIn1 \vee bIn2 \vee bIn3$ | OR operation |
| | 4 | bOut = f (truth table) | The output value depends on the parameterised truth table |
| C00823 | see truth table | | Truth table<br>Each of the 8 possible input combinations can be assigned to the output value FALSE or TRUE. |

## Truth table for C00822 = 4

| bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|
| FALSE | FALSE | FALSE | C00823/1 (FALSE or TRUE) |
| FALSE | FALSE | TRUE | C00823/2 (FALSE or TRUE) |
| FALSE | TRUE | FALSE | C00823/3 (FALSE or TRUE) |
| FALSE | TRUE | TRUE | C00823/4 (FALSE or TRUE) |
| TRUE | FALSE | FALSE | C00823/5 (FALSE or TRUE) |
| TRUE | FALSE | TRUE | C00823/6 (FALSE or TRUE) |
| TRUE | TRUE | FALSE | C00823/7 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | C00823/8 (FALSE or TRUE) |

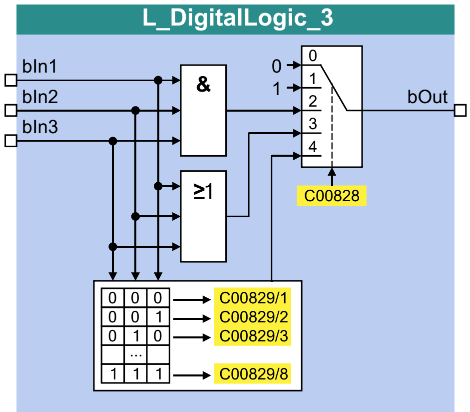
Example: If in case of the signal combination *bIn1* = FALSE, *bIn2* = FALSE and *bIn3* = TRUE, the output signal *bOut* is to be = TRUE, C00823/5 must be set to "TRUE":

| bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|
| TRUE | FALSE | FALSE | C00823/5 (TRUE) |

## 18.1.79 L_DigitalLogic_3

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- Logical ANDing of the inputs
- Logical ORing of the inputs
- Output depending on the combination of the input signals




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>BOOL | Input signal 1 |
| bIn2<br>BOOL | Input signal 2 |
| bIn3<br>BOOL | Input signal 3 |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00828 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | $bOut = bIn1 \wedge bIn2 \wedge bIn3$ | AND operation |
| | 3 | $bOut = bIn1 \vee bIn2 \vee bIn3$ | OR operation |
| | 4 | bOut = f (truth table) | The output value depends on the parameterised truth table |
| C00829 | see truth table | | Truth table<br>Each of the 8 possible input combinations can be assigned to the output value FALSE or TRUE. |

### Truth table for C00822 = 4

| bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|
| FALSE | FALSE | FALSE | C00829/1 (FALSE or TRUE) |
| FALSE | FALSE | TRUE | C00829/2 (FALSE or TRUE) |
| FALSE | TRUE | FALSE | C00829/3 (FALSE or TRUE) |
| FALSE | TRUE | TRUE | C00829/4 (FALSE or TRUE) |
| TRUE | FALSE | FALSE | C00829/5 (FALSE or TRUE) |
| TRUE | FALSE | TRUE | C00829/6 (FALSE or TRUE) |
| TRUE | TRUE | FALSE | C00829/7 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | C00829/8 (FALSE or TRUE) |

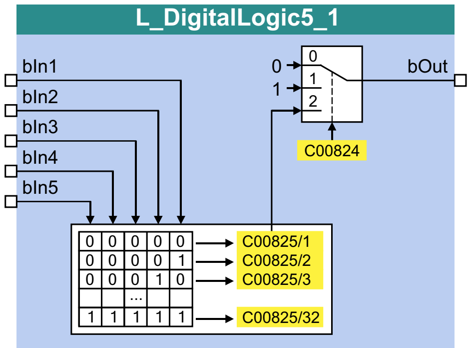
Example: If in case of the signal combination *bIn1* = FALSE, *bIn2* = FALSE and *bIn3* = TRUE, the output signal *bOut* is to be = TRUE, C00829/5 must be set to "TRUE":

| bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|
| TRUE | FALSE | FALSE | C00829/5 (TRUE) |

## 18.1.80 L_DigitalLogic5_1

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- Output depending on the combination of the input signals




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal 1 ... 5 |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00824 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | bOut = f (truth table) | The output value depends on the parameterised truth table |
| C00825 | see truth table | | Truth table<br>Each of the 32 possible input combinations can be assigned to the output value FALSE or TRUE. |

### Truth table for C00824 = 4

| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | FALSE | C00825/1 (FALSE or TRUE) |
| FALSE | FALSE | FALSE | FALSE | TRUE | C00825/2 (FALSE or TRUE) |
| FALSE | FALSE | FALSE | TRUE | FALSE | C00825/3 (FALSE or TRUE) |
| | | ... | | | C00825/... (FALSE or TRUE) |
| TRUE | TRUE | TRUE | FALSE | TRUE | C00825/30 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | TRUE | FALSE | C00825/31 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | TRUE | TRUE | C00825/32 (FALSE or TRUE) |

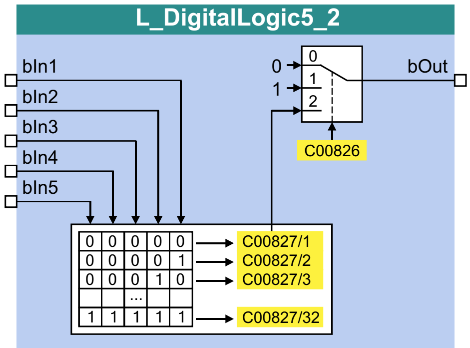
Example: If in case of the signal combination *bIn1* = TRUE, *bIn2* = FALSE, *bIn3* = TRUE, *bIn4* = TRUE and *bIn5* = TRUE, the output signal *bOut* is to be = TRUE, C00825/30 must be set to "TRUE":

| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| TRUE | TRUE | TRUE | FALSE | TRUE | C00825/30 (TRUE) |

## 18.1.81 L_DigitalLogic5_2

This FB provides a binary output signal created by a logic operation of the input signals. Optionally, one of the constant binary values independent from the input signals can be output.

- Output of a constant binary value
- Output depending on the combination of the input signals




### Inputs

| Identifier (Data type) | Information/possible settings |
|---|---|
| bIn1 ... bIn5 (BOOL) | Input signal 1 ... 5 |

### Outputs

| Identifier (Data type) | Value/meaning |
|---|---|
| bOut (BOOL) | Output signal |

### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00826 | | | Function selection |
| | 0 | "0" | Constant value "FALSE" |
| | 1 | "1" | Constant value "TRUE" |
| | 2 | bOut = f (truth table) | The output value depends on the parameterised truth table |
| C00827 | see truth table | | Truth table<br>Each of the 32 possible input combinations can be assigned to the output value FALSE or TRUE. |

### Truth table for C00826 = 4

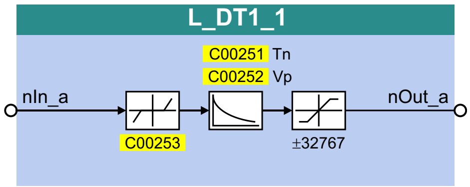
| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | FALSE | C00827/1 (FALSE or TRUE) |
| FALSE | FALSE | FALSE | FALSE | TRUE | C00827/2 (FALSE or TRUE) |
| FALSE | FALSE | FALSE | TRUE | FALSE | C00827/3 (FALSE or TRUE) |
| | | ... | | | C00827/... (FALSE or TRUE) |
| TRUE | TRUE | TRUE | FALSE | TRUE | C00827/30 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | TRUE | FALSE | C00827/31 (FALSE or TRUE) |
| TRUE | TRUE | TRUE | TRUE | TRUE | C00827/32 (FALSE or TRUE) |

Example: If in case of the signal combination *bIn1* = TRUE, *bIn2* = FALSE, *bIn3* = TRUE, *bIn4* = TRUE and *bIn5* = TRUE, the output signal *bOut* is to be = TRUE, C00827/30 must be set to "TRUE":

| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | Output signal bOut |
|---|---|---|---|---|---|
| TRUE | TRUE | TRUE | FALSE | TRUE | C00827/30 (TRUE) |

## 18.1.82 L_DT1_1

This FB differentiates signals. The function block can, for instance, be used to apply an acceleration (dv/dt).




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

### Outputs

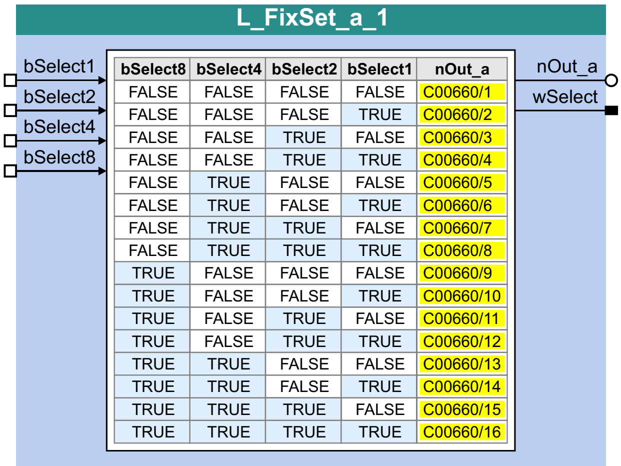
| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00251 | 10 | ms | 5000 | Time constant Tn<br>• Lenze setting: 1000 ms |
| C00252 | -320.00 | | 320.00 | Gain factor Vp<br>• Lenze setting: 1.00 |
| C00253 | **Note:** The most significant bit determines the sign of the value, the remaining bits determine the numerical value. | | | Selection of sensitivity<br>• Depending on the selection, the number of indicated higher-order bits is evaluated.<br>• Lenze setting: 15 bit |
| | 1 | 15 bits | | Bit 0 ... bit 14 are evaluated |
| | 2 | 14 bits | | Bit 0 ... bit 13 are evaluated |
| | 3 | 13 bits | | Bit 0 ... bit 12 are evaluated |
| | 4 | 12 bits | | Bit 0 ... bit 11 are evaluated |
| | 5 | 11 bits | | Bit 0 ... bit 10 are evaluated |
| | 6 | 10 bits | | Bit 0 ... bit 9 are evaluated |
| | 7 | 9 bits | | Bit 0 ... bit 8 are evaluated |

## 18.1.83 L_FixSet_a_1

This FB outputs one of 16 parameterisable analog signals. Binary coded selection of the "fixed value" to be output via the four selection inputs.




| bSelect8 | bSelect4 | bSelect2 | bSelect1 | nOut_a |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | C00660/1 |
| FALSE | FALSE | FALSE | TRUE | C00660/2 |
| FALSE | FALSE | TRUE | FALSE | C00660/3 |
| FALSE | FALSE | TRUE | TRUE | C00660/4 |
| FALSE | TRUE | FALSE | FALSE | C00660/5 |
| FALSE | TRUE | FALSE | TRUE | C00660/6 |
| FALSE | TRUE | TRUE | FALSE | C00660/7 |
| FALSE | TRUE | TRUE | TRUE | C00660/8 |
| TRUE | FALSE | FALSE | FALSE | C00660/9 |
| TRUE | FALSE | FALSE | TRUE | C00660/10 |
| TRUE | FALSE | TRUE | FALSE | C00660/11 |
| TRUE | FALSE | TRUE | TRUE | C00660/12 |
| TRUE | TRUE | FALSE | FALSE | C00660/13 |
| TRUE | TRUE | FALSE | TRUE | C00660/14 |
| TRUE | TRUE | TRUE | FALSE | C00660/15 |
| TRUE | TRUE | TRUE | TRUE | C00660/16 |

### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bSelect1 ... bSelect8<br>BOOL | Binary coded selection of the fixed value to be output<br>• See truth table displayed in the FB above. |

### Outputs

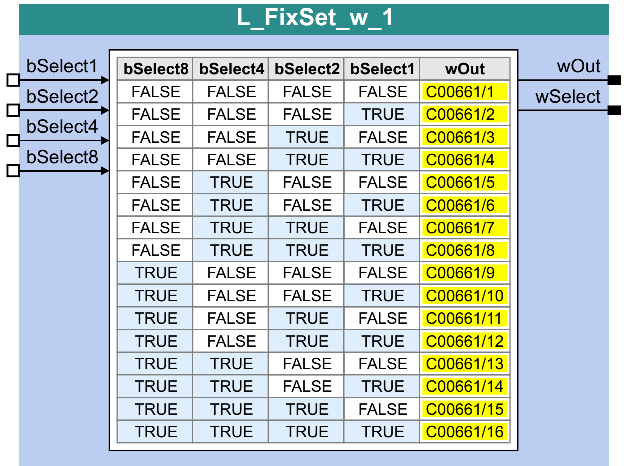
| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output of the selected fixed value |
| wSelect<br>WORD | Current selection (0 ... 15) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00660/1...16 | -199.99 | % | 199.99 | Fixed value 0 ... 15<br>• Lenze setting: 0.00 % |

## 18.1.84 L_FixSet_w_1

This FB outputs one of 16 parameterisable data words. Binary coded selection of the "fixed value" to be output via the four selection inputs.




| bSelect8 | bSelect4 | bSelect2 | bSelect1 | wOut |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | C00661/1 |
| FALSE | FALSE | FALSE | TRUE | C00661/2 |
| FALSE | FALSE | TRUE | FALSE | C00661/3 |
| FALSE | FALSE | TRUE | TRUE | C00661/4 |
| FALSE | TRUE | FALSE | FALSE | C00661/5 |
| FALSE | TRUE | FALSE | TRUE | C00661/6 |
| FALSE | TRUE | TRUE | FALSE | C00661/7 |
| FALSE | TRUE | TRUE | TRUE | C00661/8 |
| TRUE | FALSE | FALSE | FALSE | C00661/9 |
| TRUE | FALSE | FALSE | TRUE | C00661/10 |
| TRUE | FALSE | TRUE | FALSE | C00661/11 |
| TRUE | FALSE | TRUE | TRUE | C00661/12 |
| TRUE | TRUE | FALSE | FALSE | C00661/13 |
| TRUE | TRUE | FALSE | TRUE | C00661/14 |
| TRUE | TRUE | TRUE | FALSE | C00661/15 |
| TRUE | TRUE | TRUE | TRUE | C00661/16 |

### Inputs

| **Identifier** Data type | **Information/possible settings** |
|---|---|
| bSelect1 ... bSelect8 BOOL | Binary coded selection of the fixed value to be output<br>• See truth table displayed in the FB above. |

### Outputs

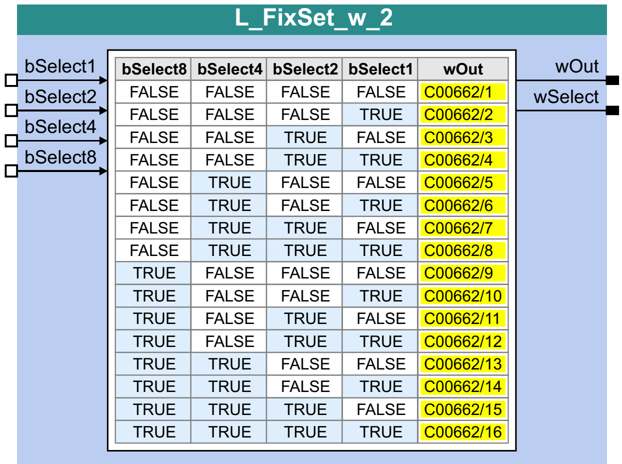
| **Identifier** Data type | **Value/meaning** |
|---|---|
| wOut WORD | Output of the selected fixed value |
| wSelect WORD | Current selection (0 ... 15) |

### Parameter

| **Parameter** | **Possible settings** | | | **Info** |
|---|---|---|---|---|
| C00661/1...16 | 0 | | 65535 | Fixed values 0 ... 15<br>• Lenze setting: 0 |

## 18.1.85 L_FixSet_w_2

This FB outputs one of 16 parameterisable data words. Binary coded selection of the "fixed value" to be output via the four selection inputs.




| bSelect8 | bSelect4 | bSelect2 | bSelect1 | wOut |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | C00662/1 |
| FALSE | FALSE | FALSE | TRUE | C00662/2 |
| FALSE | FALSE | TRUE | FALSE | C00662/3 |
| FALSE | FALSE | TRUE | TRUE | C00662/4 |
| FALSE | TRUE | FALSE | FALSE | C00662/5 |
| FALSE | TRUE | FALSE | TRUE | C00662/6 |
| FALSE | TRUE | TRUE | FALSE | C00662/7 |
| FALSE | TRUE | TRUE | TRUE | C00662/8 |
| TRUE | FALSE | FALSE | FALSE | C00662/9 |
| TRUE | FALSE | FALSE | TRUE | C00662/10 |
| TRUE | FALSE | TRUE | FALSE | C00662/11 |
| TRUE | FALSE | TRUE | TRUE | C00662/12 |
| TRUE | TRUE | FALSE | FALSE | C00662/13 |
| TRUE | TRUE | FALSE | TRUE | C00662/14 |
| TRUE | TRUE | TRUE | FALSE | C00662/15 |
| TRUE | TRUE | TRUE | TRUE | C00662/16 |

### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bSelect1 ... bSelect8<br>BOOL | Binary coded selection of the fixed value to be output<br>• See truth table displayed in the FB above. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output of the selected fixed value |
| wSelect<br>WORD | Current selection (0 ... 15) |

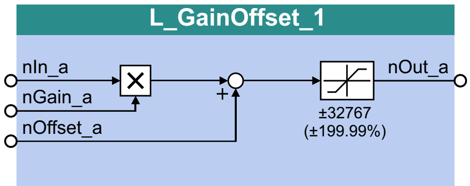
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00662/1...16 | 0 | | 65535 | Fixed values 0 ... 15<br>• Lenze setting: 0 |

## 18.1.86 L_GainOffset_1

This FB can amplify an analog input signal and then add an offset to it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

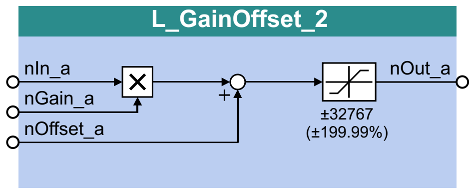
### Function

$$\text{nOut\_a} = (\text{nIn\_a} \cdot \text{Gain factor}) + \text{Offset}$$

## 18.1.87 L_GainOffset_2

This FB can amplify an analog input signal and then add an offset to it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

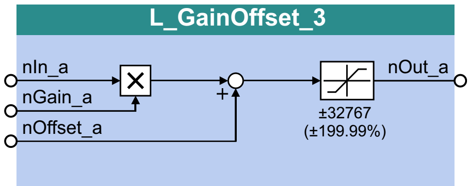
### Function

$$\text{nOut\_a} = (\text{nIn\_a} \cdot \text{Gain factor}) + \text{Offset}$$

## 18.1.88 L_GainOffset_3

This FB can amplify an analog input signal and then add an offset to it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

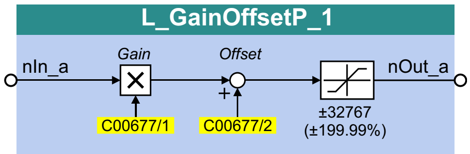
### Function

$$\text{nOut\_a} = (\text{nIn\_a} \cdot \text{Gain factor}) + \text{Offset}$$

## 18.1.89 L_GainOffsetP_1

This FB can amplify an analog input signal and then add an offset to it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Identifier / Data type | | Information/possible settings |
|---|---|---|
| nIn_a | INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Identifier / Data type | | Value/meaning |
|---|---|---|
| nOut_a | INT | Output signal<br>• Internal limitation to ±199.99 % |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00677/1 | -199.99 | % | 199.99 | Gain factor<br>• Lenze setting: 100.00 %<br>• 199.99 % ≈2 |
| C00677/2 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

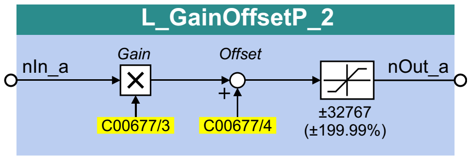
### Function

$$nOut\_a = (nIn\_a \cdot \text{Gain factor}) + \text{Offset}$$

## 18.1.90 L_GainOffsetP_2

This FB can amplify an analog input signal and then add an offset to it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Identifier / Data type | | Information/possible settings |
|---|---|---|
| nIn_a | INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Identifier / Data type | | Value/meaning |
|---|---|---|
| nOut_a | INT | Output signal<br>• Internal limitation to ±199.99 % |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00677/3 | -199.99 | % | 199.99 | Gain factor<br>• Lenze setting: 100.00 %<br>• 199.99 % ≈ 2 |
| C00677/4 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

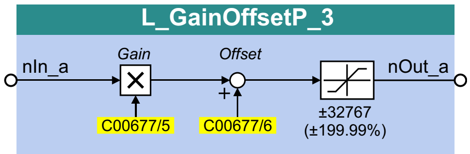
### Function

$$\text{nOut\_a} = (\text{nIn\_a} \cdot \text{Gain factor}) + \text{Offset}$$

### 18.1.91 L_GainOffsetP_3

This FB can amplify an analog input signal and then add an offset to it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00677/5 | -199.99 | % | 199.99 | Gain factor<br>• Lenze setting: 100.00 %<br>• 199.99 % ≈2 |
| C00677/6 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

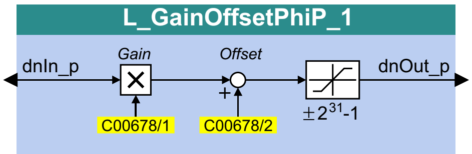
#### Function

$$\text{nOut\_a} = (\text{nIn\_a} \cdot \text{Gain factor}) + \text{Offset}$$

### 18.1.92 L_GainOffsetPhiP_1

This FB can amplify an angle signal and add an offset to it afterwards.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *dnOut_p* output is internally limited to $\pm 2^{31}$-1.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00678/1 | -2147483647 | Incr. | 2147483647 | Offset<br>• Lenze setting: 0 incr. |
| C00678/2 | -2147483647 | | 2147483647 | Gain factor<br>• Lenze setting: 65536 |

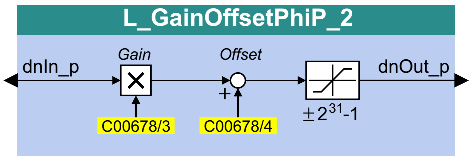
#### Function

$$\text{dnOut\_p} = (\text{dnIn\_p} \cdot \text{Gain factor}) + \text{Offset}$$

### 18.1.93 L_GainOffsetPhiP_2

This FB can amplify an angle signal and add an offset to it afterwards.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Gain and offset are selected via parameters.
- The value provided at the *dnOut_p* output is internally limited to $\pm 2^{31}$-1.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

#### Parameter

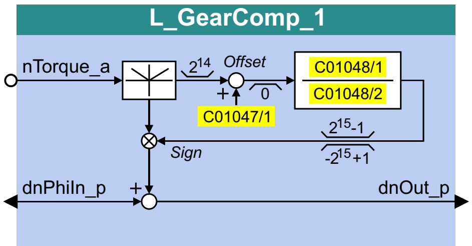
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00678/3 | -2147483647 | Incr. | 2147483647 | Offset<br>• Lenze setting: 0 incr. |
| C00678/4 | -2147483647 | | 2147483647 | Gain factor<br>• Lenze setting: 65536 |

#### Function

$$\text{dnOut\_p} = (\text{dnIn\_p} \cdot \text{Gain factor}) + \text{Offset}$$

## 18.1.94 L_GearComp_1

This FB is used for dynamic compensation of elasticities in the drive train that arise, e.g. by elastic coupling of speed reduction gearboxes or long transmission shafts.




### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| nTorque_a<br>INT | Torque setpoint<br>• 16384 ≡ 100 % |
| dnPhiIn_p<br>DINT | Angle setpoint<br>• 65536 increments ≡ 1 motor revolution |

### Outputs

| Identifier / Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Corrected angle setpoint<br>• 65536 increments ≡ 1 motor revolution |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01047/1 | -16383 | | 16383 | Offset<br>• Static torque offset<br>• Lenze setting: 0 |
| C01048/1 | -32767 | | 32767 | Numerator<br>• Elastic constant<br>• Lenze setting: 1 |
| C01048/2 | 1 | | 32767 | Denominator<br>• Elastic constant<br>• Lenze setting: 1 |

### Function

Elasticity is a measure of how far the load have moved from the ideal setpoint position at motor standstill due to mechanical force effect.

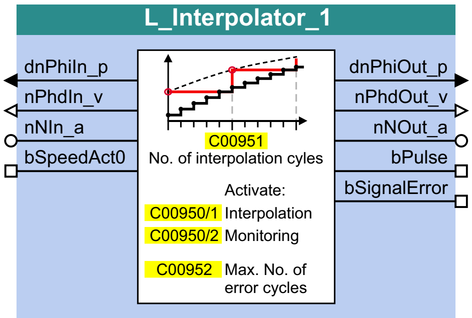
- Example: "Hoist":
  Due to elasticity of the mechanical transmission elements, the real position of the "hook" varies in loaded and unloaded status.
- In order to compensate errors caused by elasticity, the actual torque at the *nTorque_a* input is applied. This torque is a measure for the current load.
- The multiplication by an elasticity factor results in an angle compensation value which is added to the setpoint angle correctly signed depending on the direction of the torque. This serves to correct the false position of the load.
- The elasticity factor is selected in the form of numerator and denominator via C01048/1 and C01048/2.
- C01047/1 serves to select a static correction value (offset).

**Note!**

Please note that the compensation may be reversed in case of a signed numerator/denominator selection of the elasticity factor!

## 18.1.95 L_Interpolator_1

This FB interpolates a position setpoint and/or an analog value e.g. to compensate for larger bus transmission cycles or to continue signal characteristics if data telegrams are missing.




### Inputs

| Identifier / Data type | Information/possible settings | |
|---|---|---|
| dnPhIn_p<br>DINT | Position setpoint<br>• Is interpolated and completed when signal interpolation is activated. | |
| nPhdIn_v<br>INT | Angular velocity<br>• Is only passed through to the *nPhdOut_v* output. | |
| nNIn_a<br>INT | Analog value<br>• Is interpolated when signal interpolation is activated. | |
| bSpeedAct0<br>BOOL | Input for detecting the "Current speed is zero" status<br>• This status signal needs to be transmitted by the setpoint source to ensure trouble-free operation. | |
| | TRUE | Current speed is zero. |

### Outputs

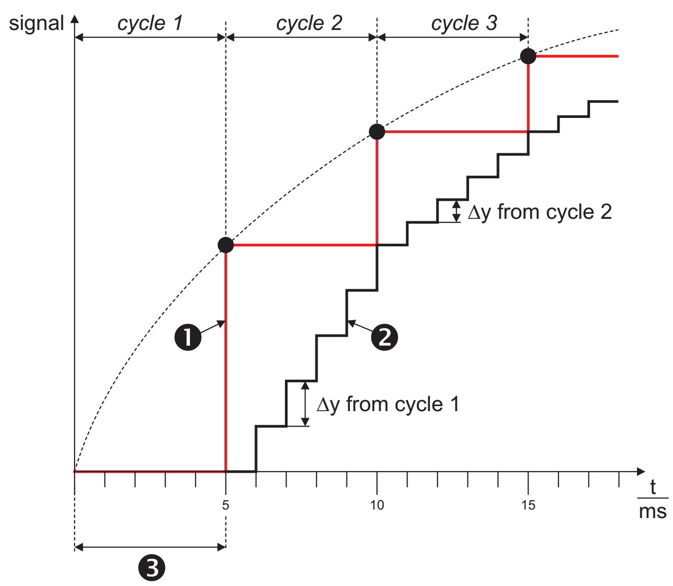
| Identifier / Data type | Value/meaning | |
|---|---|---|
| dnPhOut_p<br>DINT | Output of the *dnPhIn_p* position setpoint which, if applicable, has been interpolated and completed | |
| nPhdOut_v<br>INT | Output of the *nPhdIn_v* angular velocity | |
| nNOut_a<br>INT | Output of the *nNIn_a* analog value which, if applicable, has been interpolated | |
| bPulse<br>BOOL | "Input values have been accepted" status signal | |
| | TRUE | The input values have been accepted during this cycle. |
| bSignalError<br>BOOL | "Signal error" status signal<br>• Only if signal monitoring is active (C00950/2 = "1: On"). | |
| | TRUE | The number of missing data telegrams has exceeded the limit value parameterised in C00952. |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00950/1 | | | | Signal interpolation of the *dnPhIn_p* and *nNIn_a* input signals<br>• Lenze setting: Off<br>▸ Signal interpolation (📖 1260) |
| | 0 | Off | | |
| | 1 | On | | |
| C00950/2 | | | | Signal monitoring of the *dnPhIn_p* input signal<br>• Lenze setting: Off<br>▸ Signal monitoring (📖 1261) |
| | 0 | Off | | |
| | 1 | On | | |
| C00951 | 1 | | 65535 | Number of interpolation steps<br>• Corresponds to the receive cycle of the data telegrams in [ms].<br>• Lenze setting: 1 |
| C00952 | 0 | | 65535 | Limit value for missing data telegrams<br>• Lenze setting: 5<br>▸ Signal monitoring (📖 1261) |
| C00953 | 0 | | 100 | Speed-up<br>In preparation! |

## 18.1.95.1 Signal interpolation

If signal interpolation is active (C00950/1 = 1), the output signal will not reach the level of the corresponding input signal until all interpolation steps parameterised in C00951 have been performed:




❶ Setpoint signal
❷ Interpolated setpoint signal
❸ Receive cycle in [ms] = Number of interpolation steps (in our example: 5)

[18-40] Signal characteristic

**Note!**

Do not change the number of interpolation steps during operation. Otherwise the interpolation becomes inaccurate.

### 18.1.95.2 Signal monitoring

If signal monitoring is active (C00950/2 = 1), the signal characteristic of the *dnPhIn_p* input signal is continued even if the data telegram is missing (setpoint selection via CAN).

Monitoring is performed on the basis of the *dnPhIn_p* position setpoint and the *bSpeedAct0* status signal:

- If the *dnPhIn_p* position setpoint remains the same in the next device cycle, it is either because the speed is zero or because no data telegram has been received.
- The evaluation of the *bSpeedAct0* status signal gives information about which reason applies. This status signal needs to be transmitted by the setpoint source to ensure trouble-free operation:
  - *bSpeedAct0* = FALSE means that the speed is not zero, so an error is assumed: The signal characteristic of the *dnPhIn_p* input signal is completed (the current slope is retained).
  - *bSpeedAct0* = TRUE means that the speed is zero, so the unchanged position setpoint is not treated as an error.
- If the number of missing data telegrams exceeds the limit value parameterised in C00952, the *bSignalError* output is set to TRUE.
  - The *bSignalError* output is automatically reset to FALSE if correct signals are detected at *dnPhIn_p* and *bSpeedAct0* again.

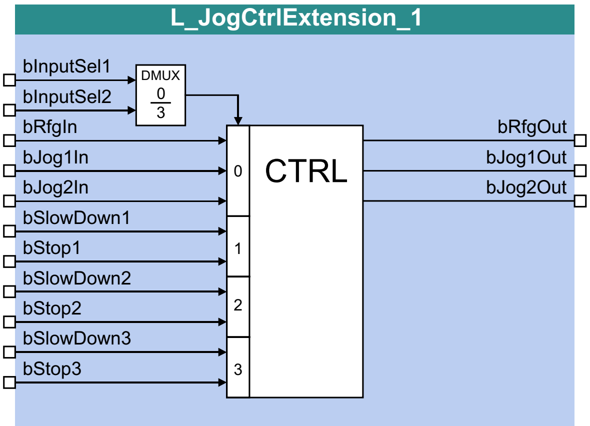
**Note!**

The *nNIn_a* analog value is not monitored!

## 18.1.96 L_JogCtrlExtension_1

This FB can be connected upstream to the L_NSet ramp function generator/setpoint generator to implement a switch-off positioning at limit switch.

▶ Detailed information on this operating mode can be found in the description of the "Switch-off positioning" TA.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bInputSel1<br>bInputSel2<br>BOOL | Activation of the *bSlowDown1/bStop1*, *bSlowDown2/bStop2* and *bSlowDown3/bStop3* signal pairs according to the Truth table |
| bRfgIn<br>BOOL | Ramping down of the setpoint generator in the downstream L_NSet FB according to the Truth table |
| bJog1In<br>bJog2In<br>BOOL | Selection inputs for setting fixed speeds in the setpoint generator<br>• If the pre-switch off is inactive (*bInputSel1* and *bInputSel2* are both set to FALSE), the two control signals are output one-to-one at the *bJog1Out* and *bJog2Out* outputs.<br>• To achieve the desired behaviour (starting at high speed, pre-switch off at low speed), both inputs must be set to TRUE.<br>• Fixed setpoint 2 must be less than fixed setpoint 3! Otherwise, the drive will start at a low speed and accelerate after the pre-switch off.<br>• If, in addition to the *bJog1In* and *bJog2In* inputs, other jog signals are set at the L_NSet FB, new fixed setpoints are reached, and the drive traverses at speeds that differ from the selection via *bJog1In* and *bJog2In*. |
| bSlowDown1<br>bSlowDown2<br>bSlowDown3<br>BOOL | Activation of fixed setpoint 2 in the downstream L_NSet FB<br>• These inputs only fulfil a function if they have been activated via *bInputSel1* and *bInputSel2* previously (see Truth table). |
| bStop1<br>bStop2<br>bStop3<br>BOOL | Ramping down of the ramp function generator in the downstream L_NSet FB<br>• These inputs only fulfil a function if they have been activated via *bInputSel1* and *bInputSel2* previously (see Truth table). |

## Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bRfgOut<br>BOOL | Control signal for ramping down the setpoint generator<br>• Connect this output to the *bRfg0* input of the L_NSet FB. |
| bJog1Out<br>BOOL | Control signal for setting fixed speeds in the setpoint generator<br>• Connect this output to the *bJog1* input of the L_NSet FB. |
| bJog2Out<br>BOOL | Control signal for setting fixed speeds in the setpoint generator<br>• Connect this output to the *bJog2* input of the L_NSet FB. |

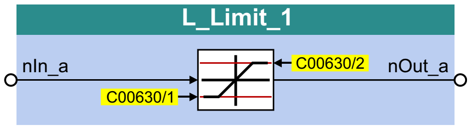
## Truth table

| Input | | Function | Response in the L_NSet FB |
|---|---|---|---|
| **bInputSel1** | **bInputSel2** | | |
| FALSE | FALSE | Pre-switch off inactive | No response<br>• The *bRfgIn* input signal is directly output at the *bRfgOut* output.<br>• The *bJogIn1* and *bJogIn2* input signals are directly output at the *bJog1Out* and *bJog2Out* outputs. |
| TRUE | FALSE | The *bSlowDown1* and *bStop1* inputs are evaluated. | **Pre-switch off can be activated**<br>• If the SlowDown function is activated via the selected *bSlowDown* input, fixed setpoint 2 in the setpoint generator is activated via the *bJog1Out* and *bJog2Out* outputs.<br>• If the Stop function is activated via the selected *bStop* input, the *bRfgOut* output is set to TRUE and hence the setpoint generator is deactivated. |
| FALSE | TRUE | The *bSlowDown2* and *bStop2* inputs are evaluated. | |
| TRUE | TRUE | The *bSlowDown3* and *bStop3* inputs are evaluated. | |

[18-1] Truth table for activating the pre-switch off

## 18.1.97 L_Limit_1

This FB limits an analog input signal to a value range whose upper and lower limit can be set via parameters.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Scaling: 16384 ≡ 100 % |

### Parameter

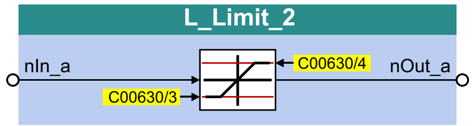
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00630/1 | -199.99 | % | 199.99 | Lower limit<br>• Lenze setting: -100.00 % |
| C00630/2 | -199.99 | % | 199.99 | Upper limit<br>• Lenze setting: 100.00 % |

**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *nOut_a* output.

### 18.1.98 L_Limit_2

This FB limits an analog input signal to a value range whose upper and lower limit can be set via parameters.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Scaling: 16384 ≡ 100 % |

#### Parameter

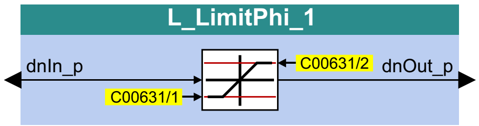
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00630/3 | -199.99 | % | 199.99 | Lower limit<br>• Lenze setting: -100.00 % |
| C00630/4 | -199.99 | % | 199.99 | Upper limit<br>• Lenze setting: 100.00 % |

**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *nOut_a* output.

## 18.1.99 L_LimitPhi_1

This FB limits an angle signal to a value range whose upper and lower value range can be set via parameters.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>INT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>INT | Output signal |

### Parameter

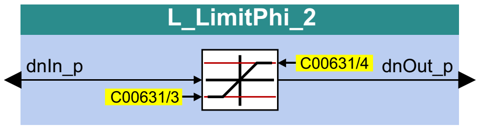
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00631/1 | -2147483647 | Incr. | 2147483647 | Lower limit<br>• Lenze setting: -2147483647 incr. |
| C00631/2 | -2147483647 | Incr. | 2147483647 | Upper limit<br>• Lenze setting: 2147483647 incr. |

**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *dnOut_p* output.

## 18.1.100 L_LimitPhi_2

This FB limits an angle signal to a value range whose upper and lower value range can be set via parameters.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>INT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>INT | Output signal |

### Parameter

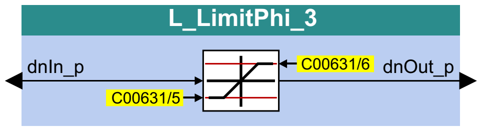
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00631/3 | -2147483647 | Incr. | 2147483647 | Lower limit<br>• Lenze setting: -2147483647 incr. |
| C00631/4 | -2147483647 | Incr. | 2147483647 | Upper limit<br>• Lenze setting: 2147483647 incr. |

**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *dnOut_p* output.

### 18.1.101 L_LimitPhi_3

This FB limits an angle signal to a value range whose upper and lower value range can be set via parameters.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>INT | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>INT | Output signal |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00631/5 | -2147483647 | Incr. | 2147483647 | Lower limit<br>• Lenze setting: -2147483647 incr. |
| C00631/6 | -2147483647 | Incr. | 2147483647 | Upper limit<br>• Lenze setting: 2147483647 incr. |

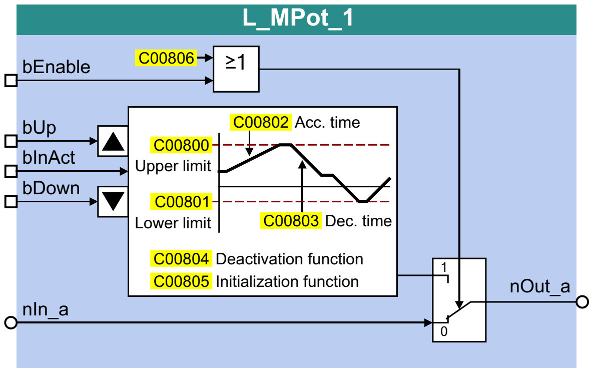
**Tip!**

Always set the lower limit lower than the upper limit, otherwise value "0" is provided at the *dnOut_p* output.

## 18.1.102 L_MPot_1

This FB replaces a hardware motor potentiometer and can be used as an alternative setpoint source which is controlled via two inputs.

- The signal is output via a ramp function generator with linear ramps.
- The acceleration and deceleration times are set via parameters.
- Constant ramping even with speed limit values changed online.
- The motor potentiometer function can be switched on/off online.




### Inputs

| Identifier<br>Data type | | Information/possible settings |
|---|---|---|
| bEnable<br>BOOL | | Switch over motor potentiometer function<br>*bEnable* input and C00806 code are ORed. |
| | TRUE | Motor potentiometer function is active, setpoint can be changed via *bUp* and *bDown*.<br>• With switching to TRUE, the value applied to *nIn_a* is automatically transferred to the motor potentiometer. |
| | FALSE | The value applied to *nIn_a* is output at *nOut_a*. |
| nIn_a<br>INT | | When bEnable = FALSE, the analog input signal nIn_ is switched to the nOut_a output. |
| bUp<br>BOOL | | Approaching of the upper speed limit value set in C00800. |
| | TRUE | The *nOut_a* output signal runs to its upper limit value (*nHighLimit*).<br>• If the *bDown* input is simultaneously set to TRUE, the *nOut_a* output signal is not changed. |
| bDown<br>BOOL | | Approaching of the lower speed limit value set in C00801. |
| | TRUE | The *nOut_a* output signal runs to its lower limit value (*nLowLimit*).<br>• If the *bUp* input is simultaneously set to TRUE, the *nOut_a* output signal is not changed. |
| bInAct<br>BOOL | | Deactivate motor potentiometer function<br>• This input has the highest priority.<br>• When the motor potentiometer is deactivated, the *nOut_a* output signal follows the function set with code C00804. |
| | TRUE | Motor potentiometer function is deactivated. |

## Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

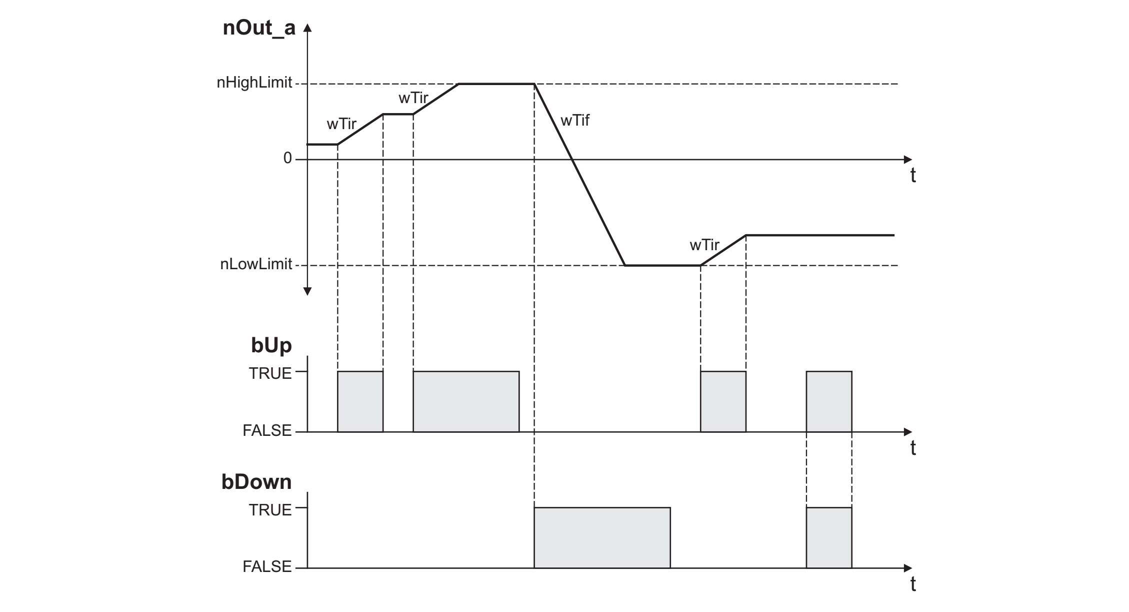
## Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00800 | -199.99 | % | 199.99 | Upper limit<br>• Lenze setting: 100.00 % |
| C00801 | -199.99 | % | 199.99 | Lower limit<br>• Lenze setting: -100.00 % |
| C00802 | 0.1 | s | 6000.0 | Acceleration time<br>• Lenze setting: 10.0 s |
| C00803 | 0.1 | s | 6000.0 | Deceleration time<br>• Lenze setting: 10.0 s |
| C00804 | | | | Inactive function<br>• Selection of response when deactivating the motor potentiometer via the input *bInAct*.<br>• Lenze setting: 0 |
| | 0 | No further action; *nOut_a* retains its value. | | |
| | 1 | The motor potentiometer returns to 0 % within the deceleration time $T_{if}$ | | |
| | 2 | The motor potentiometer runs to the lower limit value (C00801) within the deceleration time $T_{if}$ | | |
| | 3 | The motor potentiometer output immediately changes to 0 % | | **Important for the emergency stop function** |
| | 4 | The motor potentiometer output immediately changes to the lower limit value (C00801) | | |
| | 5 | The motor potentiometer runs to the upper limit value (C00800) within the acceleration time $T_{ir}$ | | |
| C00805 | | | | Init function<br>• Selection of response when switching on the device.<br>• Lenze setting: 0 |
| | 0 | The output value being output during mains power-off is saved non-volatilely in the internal memory of the controller. It will be reloaded during mains power-on. | | |
| | 1 | The lower limit value (C00801) is loaded during mains power-on. | | |
| | 2 | An output value = 0 % is loaded during mains power-on. | | |
| C00806 | | | | Use of the motor potentiometer<br>• When switching to 1: YES, the value applied to *nIn_a* is automatically transferred to the motor potentiometer.<br>• Lenze setting: 0 |
| | 0 | No | | |
| | 1 | Yes | | |

### 18.1.102.1 Activate & control motor potentiometer

When *bInAct* is set to FALSE, the motor potentiometer is activated.

- The currently active function depends on the current output signal *nOut_a*, the limit values set and the control signals at *bUp* and *bDown*.
- When the *nOut_a* output signal is outside the limits set, the output signal runs to the next limit with the Ti times set. This process is independent of the control signals at *bUp* and *bDown*.
- When the *nOut_a* output signal is inside the limits set, the output signal changes according to the control signals at *bUp* and *bDown*.

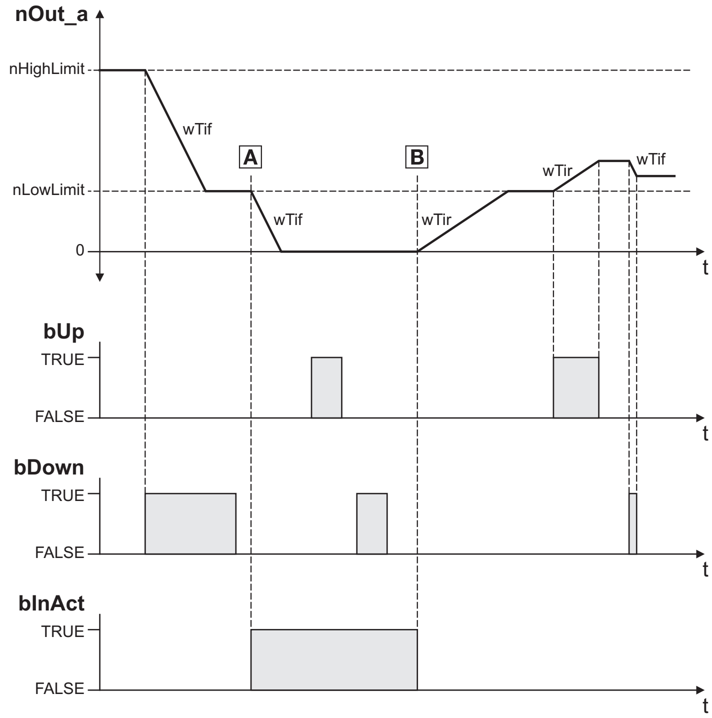



[18-41] Example: Control of the motor potentiometer

| bUp | bDown | bInact | Function |
|---|---|---|---|
| FALSE | FALSE | FALSE | The *nOut_a* output signal remains unchanged. |
| TRUE | FALSE | | The *nOut_a* output signal runs to its upper limit value (*nHighLimit*). |
| FALSE | TRUE | | The *nOut_a* output signal runs to its lower limit value (*nLowLimit*). |
| TRUE | TRUE | | The *nOut_a* output signal remains unchanged. |
| - | - | TRUE | The motor potentiometer function is deactivated. The *nOut_a* output signal responds according to the function selected via *Function*. |

### 18.1.102.2 Deactivate motor potentiometer

When the motor potentiometer is deactivated by setting *bInAct* to TRUE, the *nOut_a* output signal responds according to the function selected via *Function*.



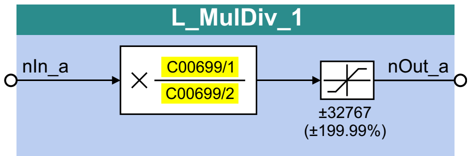

A When the motor potentiometer is deactivated, *nOut_a* is reduced to 0 % with the deceleration time *wTif*.
B If it is activated again, *nOut_a* is increased again to *nLowLimit* with the acceleration time *wTir*.

[18-42] Example: Deactivation of the motor potentiometer when the *Function* = 1 has been selected

## 18.1.103 L_MulDiv_1

This FB multiplies the analog input signal with a parameterisable factor.

- The value of the factor is determined by a quotient consisting of numerator  and denominator .
- The value output at *nOut_a* is limited to ±199.99 %.
- Division is not remainder considered.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1<br>INT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Product value (result of the multiplication)<br>• Internal limitation to ± 32767 |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00699/1 | -32767 | | 32767 | Numerator |
| C00699/2 | -32767 | | 32767 | Denominator |

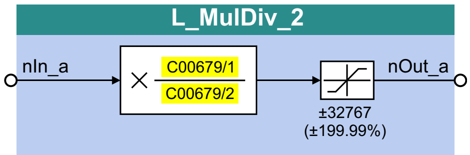
### Function

$$\text{nOut\_a} = \text{nIn\_a} \times \frac{\text{C00699/1}}{\text{C00699/2}}$$

## 18.1.104 L_MulDiv_2

This FB multiplies the analog input signal with a parameterisable factor.

- The value of the factor is determined by a quotient consisting of numerator  and denominator .
- The value output at *nOut_a* is limited to ±199.99 %.
- Division is not remainder considered.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn1<br>INT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Product value (result of the multiplication)<br>• Internal limitation to ± 32767 |

### Parameter

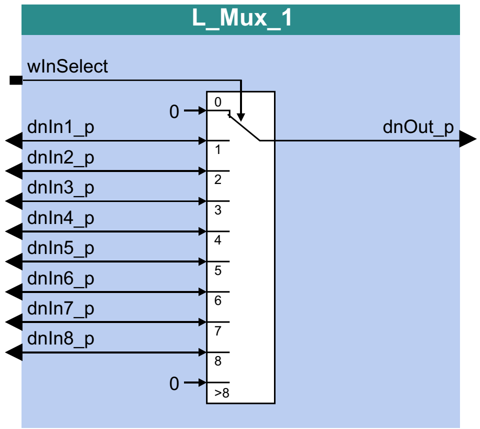
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00679/1 | -32767 | | 32767 | Numerator |
| C00679/2 | -32767 | | 32767 | Denominator |

### Function

$$nOut\_a = nIn\_a \times \frac{C00679/1}{C00679/2}$$

## 18.1.105 L_Mux_1

This FB provides one of the eight input signals *dnIn1_p* ... *dnIn8_p* at the output *dnOut_p*. The selection is made by means of the signal at the input *wInSelect*.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| wInSelect<br>WORD | Input signal | |
| | 1 ... 8 | • The values "1" ... "8" select the input signal to be applied to the output.<br>• Values from"1" ... "8" set the *dnOut_p* output to "0". |
| dnIn1_p<br>DINT<br>...<br>dnIn8_p<br>DINT | Input signal | |

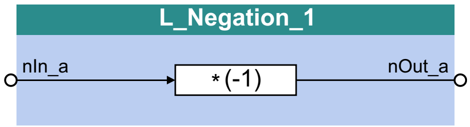
### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

## 18.1.106 L_Negation_1

This FB converts the sign of the input signal, i.e. the input signal is multiplied by the value -1 and is then output.

▶ With the value - 32768 at the *nIn_a* input, the value + 32767 is provided at the *nOut_a* output.




### Inputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nIn_a<br>INT | Input signal |

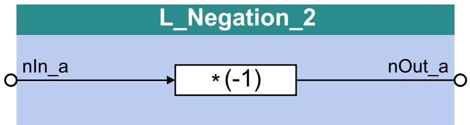
### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 18.1.107 L_Negation_2

This FB converts the sign of the input signal, i.e. the input signal is multiplied by the value -1 and is then output.

- With the value - 32768 at the *nIn_a* input, the value + 32767 is provided at the *nOut_a* output.




#### Inputs

| Identifier / Data type | Value/meaning |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

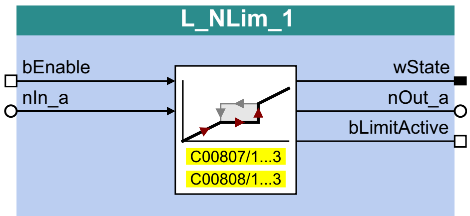
| Identifier / Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal |

### 18.1.108 L_NLim_1

This FB can mask out up to three parameterisable blocking zones within a continuous signal characteristic.

**Note!**

A value of "0" cannot be masked out if the *nIn_a* input signal changes signs.




#### Inputs

| Identifier / Data type | Information/possible settings | |
|---|---|---|
| bEnable<br>BOOL | Activate zone masking | |
| | FALSE | The FB has been deactivated.<br>• The input signal is output one-to-one at the *nOut_a* output. |
| | TRUE | The FB has been activated.<br>• Zone masking of the input signal is carried out according to the parameterised blocking zones. |
| nIn_a<br>INT | Input signal | |

#### Outputs

| Identifier/data type | Value/meaning | |
|---|---|---|
| wState<br>WORD | Bit-coded status word<br>• Bits that are not listed are reserved for future extensions. | |
| | Bit 0 | No blocking zone active |
| | Bit 1 | Blocking zone 1 active |
| | Bit 2 | Blocking zone 2 active |
| | Bit 3 | Blocking zone 3 active |
| nOut_a<br>INT | Output signal<br>• If the FB has been activated, the output signal is outside the blocking zones. | |
| bLimitActive<br>BOOL | "Limitation active" status signal | |
| | TRUE | The input signal is inside a blocking zones and is limited to the respective boundary value of the blocking zone. |

### Parameter

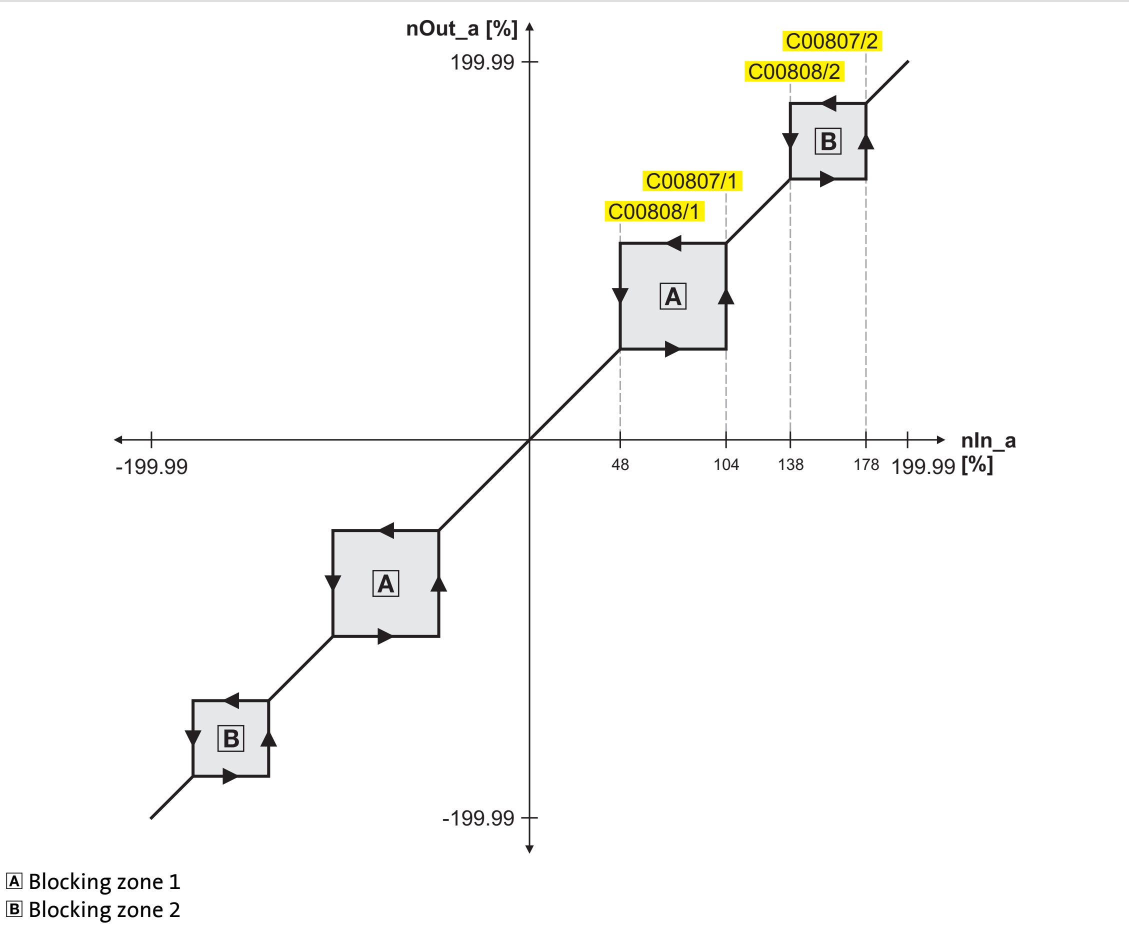
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00807/1...3 | 0.00 | % | 199.99 | Maximum blocking frequency 1 ... 3<br>• Lenze setting: 0.00 % |
| C00808/1...3 | 0.00 | % | 199.99 | Minimum blocking frequency 1 ... 3<br>• Lenze setting: 0.00 % |

### Definition of the blocking zones

Up to three zones can be parameterised which are to be skipped by the *nOut_a* output signal.

The example below shows the parameter setting of two blocking zones for the **L_NLim_1** FB:

| Parameter | Blocking zone 1 | | Blocking zone 2 | | Blocking zone 3 | |
|---|---|---|---|---|---|---|
| Minimum limit value | C00808/1: | 48 % | C00808/2: | 138 % | C00808/3: | 0 % |
| Maximum limit value | C00807/1: | 104 % | C00807/2: | 178 % | C00807/3: | 0 % |




[18-43] Zone masking by means of parameterisable blocking zones (in our example: L_NLim_1)

- The parameterised blocking zones have the same effect on negative input signals.
- A blocking zone is deactivated by entering identical limit values (in our example: Blocking zone 3).

### Overlapping of blocking zones

If blocking zones overlap, the lowest and highest value of the overlapping zones form a new zone.

In this case, the status display (*wState* output) will also just display one zone (the lower one of the two original zones).

### Abutting blocking zones

If two blocking zones abut (e.g. 20 ... 30 % and 30 ... 40 %), the limit value between the two zones (in this example 30 %) is also passed through.

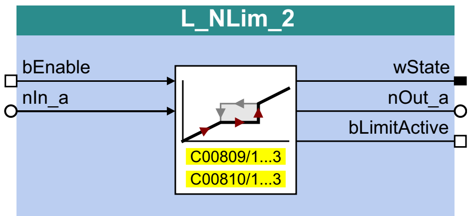
The same applies to a limit range of 0 ... xx %. At the zero crossing of the *nIn_a* input signal, a value of "0" is also output at the *nOut_a* output. Value "0" cannot be excluded. If the *nIn_a* input signal drops back to "0", the *nOut_a* output will retain the upper limit value.

## 18.1.109 L_NLim_2

This FB can mask out up to three parameterisable blocking zones within a continuous signal characteristic.

**Note!**

A value of "0" cannot be masked out if the *nIn_a* input signal changes signs.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bEnable<br>BOOL | Activate zone masking | |
| | FALSE | The FB has been deactivated.<br>• The input signal is output one-to-one at the *nOut_a* output. |
| | TRUE | The FB has been activated.<br>• Zone masking of the input signal is carried out according to the parameterised blocking zones. |
| nIn_a<br>INT | Input signal | |

### Outputs

| Identifier/data type | Value/meaning | |
|---|---|---|
| wState<br>WORD | Bit-coded status word<br>• Bits that are not listed are reserved for future extensions. | |
| | Bit 0 | No blocking zone active |
| | Bit 1 | Blocking zone 1 active |
| | Bit 2 | Blocking zone 2 active |
| | Bit 3 | Blocking zone 3 active |
| nOut_a<br>INT | Output signal<br>• If the FB has been activated, the output signal is outside the blocking zones. | |
| bLimitActive<br>BOOL | "Limitation active" status signal | |
| | TRUE | The input signal is inside a blocking zones and is limited to the respective boundary value of the blocking zone. |

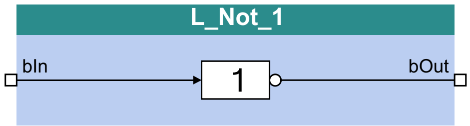
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00809/1...3 | 0.00 | % | 199.99 | Maximum blocking frequency 1 ... 3<br>• Lenze setting: 0.00 % |
| C00810/1...3 | 0.00 | % | 199.99 | Minimum blocking frequency 1 ... 3<br>• Lenze setting: 0.00 % |

For a detailed functional description see L_NLim_1.

## 18.1.110 L_Not_1

This FB negates a signal of BOOL data type.




### Inputs

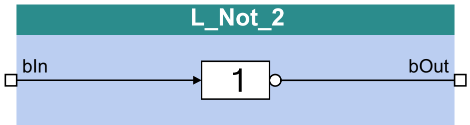
| Identifier<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

## 18.1.111 L_Not_2

This FB negates a signal of BOOL data type.




### Inputs

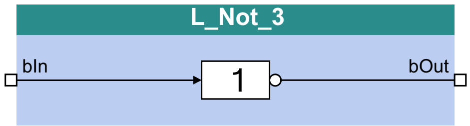
| Identifier<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

## 18.1.112 L_Not_3

This FB negates a signal of BOOL data type.




### Inputs

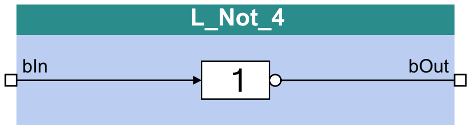
| Identifier<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

## 18.1.113 L_Not_4

This FB negates a signal of BOOL data type.




### Inputs

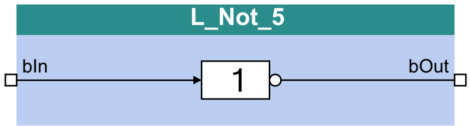
| Identifier<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

## 18.1.114 L_Not_5

This FB negates a signal of BOOL data type.




### Inputs

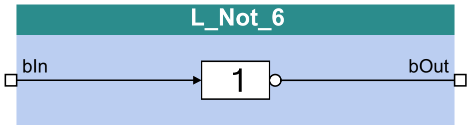
| Identifier<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

## 18.1.115 L_Not_6

This FB negates a signal of BOOL data type.




### Inputs

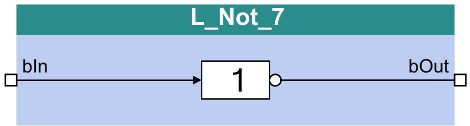
| Identifier<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

### 18.1.116 L_Not_7

This FB negates a signal of BOOL data type.




#### Inputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bIn<br>BOOL | Input signal |

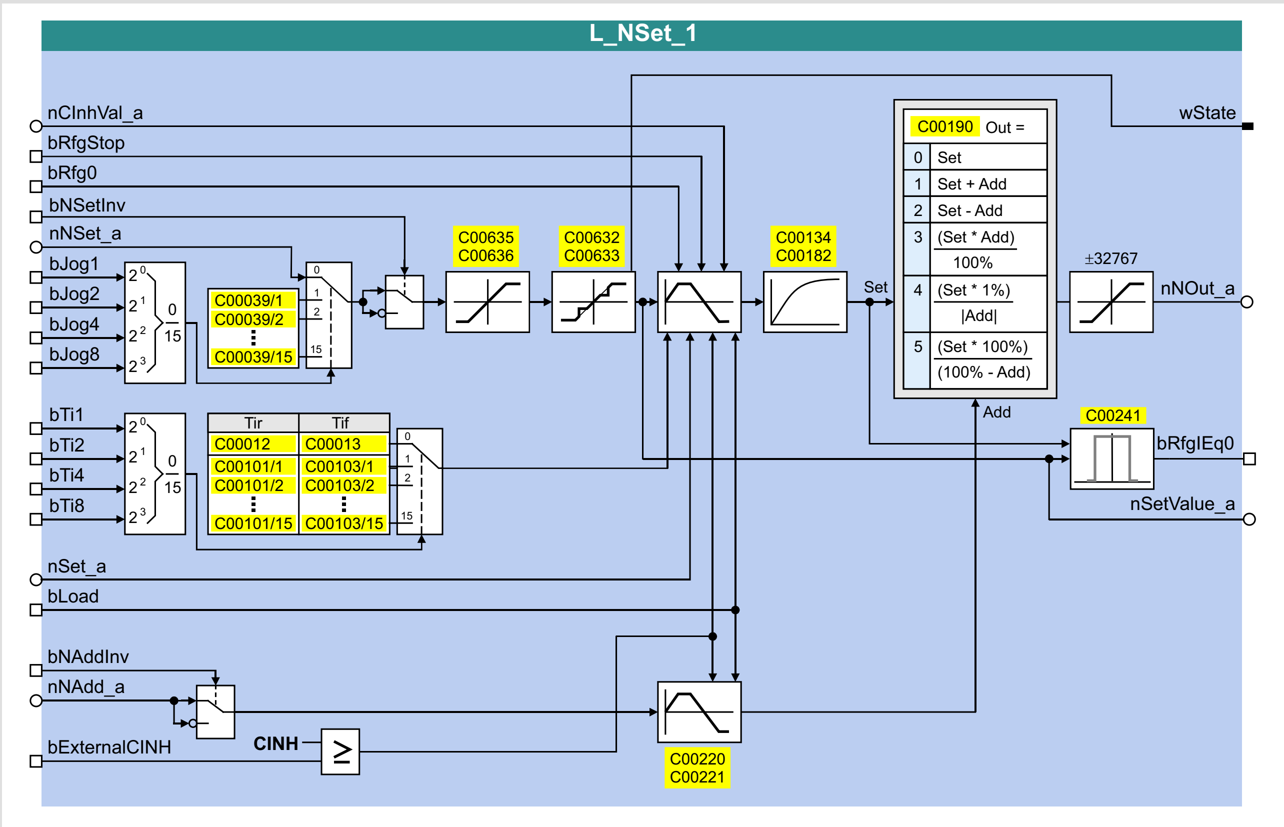
#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Result of the NOT operation (negated input signal) |

### 18.1.117 L_NSet_1

This FB is used for general signal processing of process values and is provided with the following functions:

- Ramp function generator
  - With linear ramps for main and additional setpoint path
  - With S-shaped ramp (PT1 rounding)
  - Setting and holding
- Internal limitation of the input signal
- 3 adjustable blocking zones
- Arithmetic function
- 15 fixed setpoints (JOG setpoints)
- 15 acceleration and deceleration times



## Inputs

| Identifier / Data type | Information/possible settings | |
|---|---|---|
| nCInhVal_a<br>INT | Main setpoint signal which is to be accepted by the main setpoint integrator when the controller is inhibited. | |
| bRfgStop<br>BOOL | Holding (freezing) of the current value of the main setpoint integrator | |
| | TRUE | The current value of the main setpoint integrator is held. |
| bRfg0<br>BOOL | Leading the main setpoint integrator to 0 within the current Ti times | |
| | TRUE | The current value of the main setpoint integrator is led to "0" within the Ti time set. |
| bNSetInv<br>BOOL | Signal inversion for the main setpoint | |
| | TRUE | Main setpoint signal is inverted. |
| nNset_a<br>INT | Main setpoint signal<br>• Other signals are also permitted | |
| bJog1 ... bJog8<br>BOOL | Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint<br>• Selection inputs are binary coded. | |
| bTI1 ... bTI8<br>BOOL | Selection inputs for alternative acceleration/deceleration times for the main setpoint<br>• Selection inputs are binary coded. | |
| nSet_a<br>INT | Starting value which is loaded into the main setpoint integrator by setting *bLoad* to TRUE. | |
| bLoad<br>BOOL | Control of both ramp function generators in special situations, e.g. QSP | |
| | TRUE | The *nSet_a* input signal is loaded into the main setpoint integrator and the additional setpoint integrator is set to "0". |
| bAddInv<br>BOOL | Signal inversion for the additional setpoint | |
| | TRUE | Additional setpoint signal is inverted. |
| nNAdd_a<br>INT | Additional setpoint signal<br>• Other signals are also permitted | |
| bExternalCINH<br>BOOL | Additional load input for the main setpoint integrator and the additional setpoint integrator | |
| | TRUE | The main setpoint integrator is set to the value applied at *nCInhVal_a*.<br>The additional setpoint integrator is set to "0".<br>▸ Application example for the additional load function (📖 1297) |

## Outputs

| Identifier / Data type | Value/meaning |
|---|---|
| nNOut_a<br>INT | Speed setpoint output signal<br>• Scaling: 16384 ≡ 100 % |
| bRfgIEqO<br>BOOL | Status signal "setpoint = 0" |

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| wState<br>WORD | Bit-coded status word<br>• Bits that are not listed are reserved for future extensions. | |
| | Bit 0 | No blocking zone active |
| | Bit 1 | Blocking zone 1 active |
| | Bit 2 | Blocking zone 2 active |
| | Bit 3 | Blocking zone 3 active |
| | Bit 4 | Jog in blocking zone |
| | Bit 5 | MaxLimit active |
| | Bit 6 | MinLimit active |
| nSetValue_a<br>INT | Speed-setpoint input signal of the ramp function generator<br>• Scaling: 16384 ≡ 100 % | |

## Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00012 | 0.000 | s | 999.900 | Acceleration time $T_{ir}$ for the main setpoint<br>• Lenze setting: 0.000 s |
| C00013 | 0.000 | s | 999.900 | Deceleration time $T_{if}$ for the main setpoint<br>• Lenze setting: 0.000 s |
| C00039/1..15 | -199.99 | % | 199.99 | Fixed setpoints (JOG setpoints)<br>• Lenze setting: 0.00 % |
| C00101/1..15 | 0.000 | s | 999.900 | Alternative acceleration times ($T_{ir}$) for the main setpoint<br>• Lenze setting: 0.000 s |
| C00103/1..15 | 0.000 | s | 999.900 | Alternative deceleration times ($T_{if}$) for the main setpoint<br>• Lenze setting: 0.000 s |
| C00134 | | | | Activates ramp rounding with PT1 behaviour for the main setpoint<br>• The corresponding S-ramp time must be set in C00182.<br>• Lenze setting: 0 (deactivated) |
| | 0 | Off | | |
| | 1 | PT1 behaviour | | |
| C00182 | 0.01 | s | 50.00 | S-ramp time PT1<br>• Lenze setting: 20.00 s |
| C00190 | | | | Selection of the arithmetic function for combining main and additional setpoint |
| | 0 | NOut = NSet | | Lenze setting<br>• The additional setpoint is not processed. |
| | 1 | NOut = NSet + NAdd | | |
| | 2 | NOut = NSet - NAdd | | |
| | 3 | NOut = (NSet * NAdd) / 100% | | |
| | 4 | NOut = (NSet * 1%) / \|NAdd\| | | |
| | 5 | NOut = (NSet * 100%) / (100% - NAdd) | | |
| C00220 | 0.000 | s | 999.900 | Acceleration time $T_{ir}$ for the additional setpoint<br>• Lenze setting: 0.000 s |

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00221 | 0.000 | s | 999.900 | Deceleration time $T_{if}$ for the additional setpoint<br>• Lenze setting: 0.000 s |
| C00241 | 0.00 | % | 100.00 | Hysteresis window for zero detection of speed output setpoint (output *bRfgIEqO*)<br>• Lenze setting: 0.50 % |
| C00632/1...3 | 0.00 | % | 199.99 | Maximum limit values for the speed blocking zones<br>• Selection of the maximum limit values for the blocking zones in which the speed must not be constant.<br>• Lenze setting: 0.00 % |
| C00633/1...3 | 0.00 | % | 199.99 | Minimum limit values for the speed blocking zones<br>• Selection of the minimum limit values for the blocking zones in which the speed must not be constant.<br>• Lenze setting: 0.00 % |
| C00634 | | | | Status (bit-coded)<br>• Bits that are not listed are reserved for future extensions. |
| | Bit 0 | No blocking zone active | | |
| | Bit 1 | Blocking zone 1 active | | |
| | Bit 2 | Blocking zone 2 active | | |
| | Bit 3 | Blocking zone 3 active | | |
| | Bit 4 | Jog in blocking zone | | |
| | Bit 5 | MaxLimit active | | |
| | Bit 6 | MinLimit active | | |
| C00635 | -199.99 | % | 199.99 | nMaxLimit<br>• Maximum speed setpoint for speed setpoint limitation<br>• Lenze setting: 199.99 % |
| C00636 | -199.99 | % | 199.99 | nMinLimit<br>• Minimum speed setpoint for speed setpoint limitation<br>• Lenze setting: -199.99 % |

### 18.1.117.1 Main setpoint path

- The signals in the main setpoint path are limited to a value range of ±32767.
- The signal at *nNSet_a* is first led via the JOG selection function.
- A selected JOG value switches the *nNSet_a* input inactive. Then, the subsequent signal conditioning operates with the JOG value.

### 18.1.117.2 JOG setpoints

In addition to the direct main setpoint selection via the *nNSet_a* input, so-called JOG setpoints can be preset in C00039/1...15.

- The JOG setpoints are binary-coded and can be called using the *bJog1* ... *bJog8* selection inputs so that 15 options are available:

| Selection inputs | | | | Main setpoint<br>Main setpoint |
|---|---|---|---|---|
| **bJog8** | **bJog4** | **bJog2** | **bJog1** | |
| FALSE | FALSE | FALSE | FALSE | *nNset_a* |
| FALSE | FALSE | FALSE | TRUE | C00039/1 |
| FALSE | FALSE | TRUE | FALSE | C00039/2 |
| FALSE | FALSE | TRUE | TRUE | C00039/3 |
| FALSE | TRUE | FALSE | FALSE | C00039/4 |
| FALSE | TRUE | FALSE | TRUE | C00039/5 |
| FALSE | TRUE | TRUE | FALSE | C00039/6 |
| FALSE | TRUE | TRUE | TRUE | C00039/7 |
| TRUE | FALSE | FALSE | FALSE | C00039/8 |
| TRUE | FALSE | FALSE | TRUE | C00039/9 |
| TRUE | FALSE | TRUE | FALSE | C00039/10 |
| TRUE | FALSE | TRUE | TRUE | C00039/11 |
| TRUE | TRUE | FALSE | FALSE | C00039/12 |
| TRUE | TRUE | FALSE | TRUE | C00039/13 |
| TRUE | TRUE | TRUE | FALSE | C00039/14 |
| TRUE | TRUE | TRUE | TRUE | C00039/15 |

- The number of selection inputs to be assigned depends on the number of JOG setpoints required:

| Number of JOG setpoints required | Number of selection inputs to be assigned (bJog1 ... bJog8) |
|---|---|
| 1 | At least 1 |
| 2 ... 3 | at least 2 |
| 4 ... 7 | at least 3 |
| 8 ... 15 | 4 |

### 18.1.117.3 Setpoint inversion

The output signal of the JOG function is led via an inverter.

The sign of the setpoint changes if *bNSetInv* is set to TRUE.

### 18.1.117.4 Value range of the input signal

The value range of the input signal can be limited by using the following parameters:

- C00635: MaxLimit (default setting: +199.99 %)
- C00636: MinLimit (default setting: -199.99 %)

### 18.1.117.5 Skip frequency function

If the speed setpoints in speed-variable drives are linearly increasing, for instance, the frequency/speed range is divided into a number of equal time segments. Therefore, there may be speeds during acceleration time which must be bridged very fast (e.g. natural resonant frequencies).

The skip frequency function offers the opportunity to select a range in which the initial speed is maintained. If the speed setpoint leaves that range, the drive will be accelerated to reach the desired speed.

**Note!**

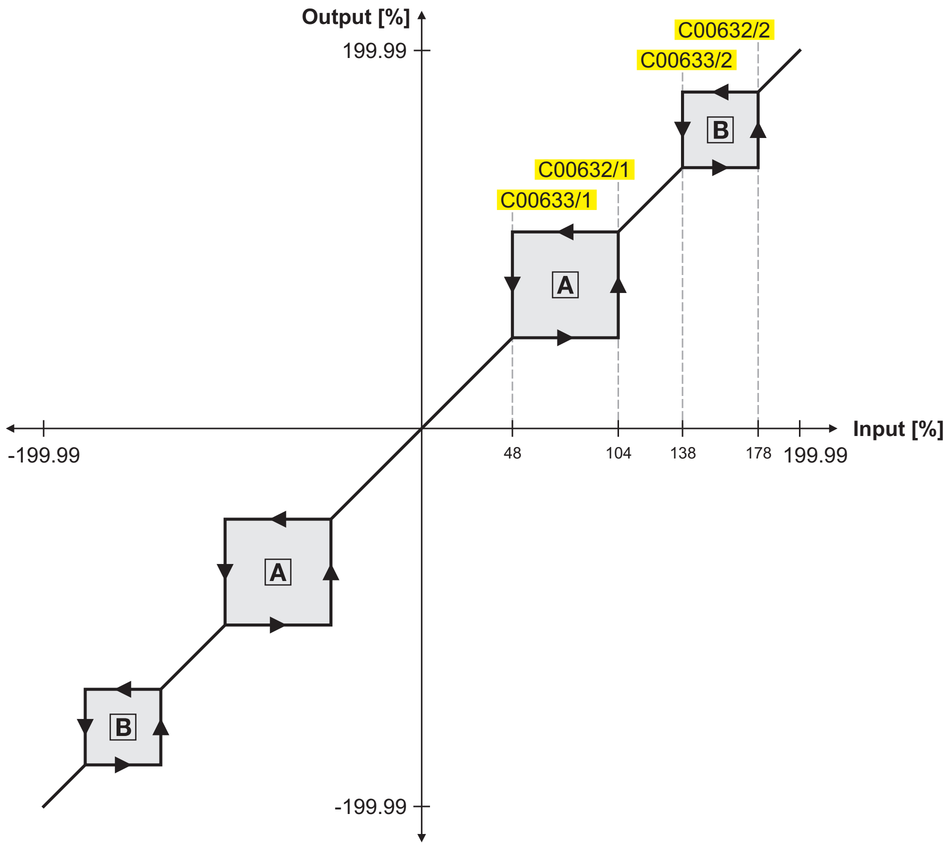
- Skip frequencies only affect main setpoints.
- It is not possible to exclude "0" speed if there is a sign reversal of the speed setpoint.

### Definition of the blocking zones

The subcodes of codes C00632 and C00633 can be used to define three zones which are to be skipped by the output setpoint and which are to be passed as fast as possible by the ramp function generator.

The example below shows the parameter setting of two blocking zones:

| Parameter | Blocking zone 1 | | Blocking zone 2 | | Blocking zone 3 | |
|---|---|---|---|---|---|---|
| Minimum limit value | C00633/1: | 48 % | C00633/2: | 138 % | C00633/3: | 0 % |
| Maximum limit value | C00632/1: | 104 % | C00632/2: | 178 % | C00632/3: | 0 % |




A Blocking zone 1
B Blocking zone 2

[18-44] Zone masking by means of parameterisable blocking zones

- The parameterised blocking zones have the same effect on negative input signals.
- A blocking zone is deactivated by entering identical limit values (in our example: Blocking zone 3).

### Overlapping of blocking zones

If blocking zones overlap, the lowest and highest value of the overlapping zones form a new zone.

In this case, the status display (output *wState* or display parameter C00634) only indicates one zone (the lower of the two original zones).

### Abutting blocking zones

If two blocking zones abut (e.g. 20 ... 30 % and 30 ... 40 %), the limit value between the two zones (in this example 30 %) is also passed through.

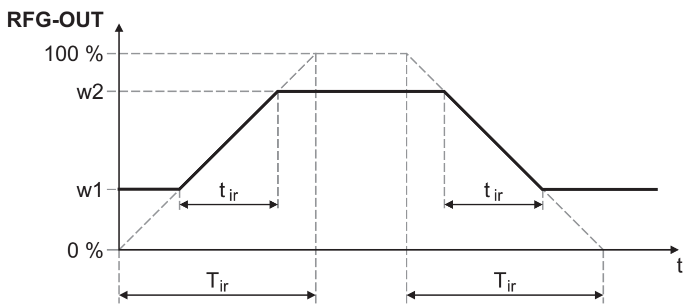
The same applies to a limit range of 0 ... xx %. During zero crossing of the speed setpoint, "0" speed is output as setpoint. It is possible to exclude "0" speed. However, in this case, the output speed will remain on the upper limit value when the input setpoint becomes "0".

**Tip!**

As described above, the acceleration phase starts after the blocking zones have been passed through. The ramp function generator integrated in the **L_Nset** function block limits the progression of the speed. For this reason, the time values set for the integrated ramp function generator should be as low as possible whereas the setpoint for the **L_NSet** function block should be generated by a ramp function generator with higher time values (e.g. L_MPot function block).

### 18.1.117.6 Ramp function generator for the main setpoint

The setpoint is now led via a ramp function generator with linear characteristic. The ramp function generator converts setpoint step-changes at the input into a ramp.




w1, w2 = change of the main setpoint as a function of $t_{ir}$ and $t_{if}$
RFG-OUT = output of the ramp function generator

[18-45] Acceleration and deceleration times

- $t_{ir}$ and $t_{if}$ are the desired times for changing between w1 and w2.
- S-ramps are possible by selecting S-ramp times.
- The $t_{ir}$/$t_{if}$ values are converted into the required Ti times according to the following formula:

$$T_{ir} = t_{ir} \cdot \frac{100\,\%}{w2 - w1} \qquad T_{if} = t_{if} \cdot \frac{100\,\%}{w2 - w1}$$

#### Setting and selection of Ti times

Via parameters, you can select 16 different Tir and Tif times each for the ramp function generator.

- The selection is made via the binary coded selection inputs *bTI1* ... *bTI8*:

| Selection inputs | | | | Used Acceleration time | Used Deceleration time |
|---|---|---|---|---|---|
| bTI8 | bTI4 | bTI2 | bTI1 | | |
| FALSE | FALSE | FALSE | FALSE | C00012 | C00013 |
| FALSE | FALSE | FALSE | TRUE | C00101/1 | C00103/1 |
| FALSE | FALSE | TRUE | FALSE | C00101/2 | C00103/2 |
| FALSE | FALSE | TRUE | TRUE | C00101/3 | C00103/3 |
| FALSE | TRUE | FALSE | FALSE | C00101/4 | C00103/4 |
| FALSE | TRUE | FALSE | TRUE | C00101/5 | C00103/5 |
| FALSE | TRUE | TRUE | FALSE | C00101/6 | C00103/6 |
| FALSE | TRUE | TRUE | TRUE | C00101/7 | C00103/7 |
| TRUE | FALSE | FALSE | FALSE | C00101/8 | C00103/8 |
| TRUE | FALSE | FALSE | TRUE | C00101/9 | C00103/9 |
| TRUE | FALSE | TRUE | FALSE | C00101/10 | C00103/10 |
| TRUE | FALSE | TRUE | TRUE | C00101/11 | C00103/11 |
| TRUE | TRUE | FALSE | FALSE | C00101/12 | C00103/12 |
| TRUE | TRUE | FALSE | TRUE | C00101/13 | C00103/13 |

| Selection inputs | | | | Used Acceleration time | Used Deceleration time |
|---|---|---|---|---|---|
| bTI8 | bTI4 | bTI2 | bTI1 | | |
| TRUE | TRUE | TRUE | FALSE | C00101/14 | C00103/14 |
| TRUE | TRUE | TRUE | TRUE | C00101/15 | C00103/15 |

### Function

- When the controller is inhibited (CINH), the ramp function generator accepts the value applied at *nCInhVal_a* and transfers it to the downstream function. This function has priority over all other functions.
- *bRfgStop* = TRUE
  - The ramp function generator is stopped. Changes at the input of the ramp function generator have no effect on the output signal.
- *bRfg0* = TRUE
  - The ramp function generator runs to 0 along its deceleration ramp.
- Furthermore it is possible to load the ramp function generator online with a defined value. For this purpose, *bLoad* must be set to TRUE. As long as this input is set, the value at *nSet_a* is transferred to the ramp function generator and provided at the output.

### Priorities:

| CINH | bLoad | bRfg0 | bRfgStop | Function |
|---|---|---|---|---|
| FALSE | FALSE | FALSE | FALSE | The ramp function generator follows the input value via the set ramps. |
| FALSE | FALSE | FALSE | **TRUE** | **Stop the ramp function generator:** The value at the output of the ramp function generator is held. |
| FALSE | FALSE | **TRUE** | FALSE | **Ramp down the ramp function generator:** The ramp function generator runs to 0 within the set deceleration time. |
| FALSE | FALSE | **TRUE** | TRUE | |
| FALSE | **TRUE** | FALSE | FALSE | **Load ramp function generator online:** The ramp function generator accepts the value at *nSet_a* and provides it at its output. |
| FALSE | **TRUE** | FALSE | TRUE | |
| FALSE | **TRUE** | TRUE | FALSE | |
| FALSE | **TRUE** | TRUE | TRUE | |
| **TRUE** | FALSE | FALSE | FALSE | **Controller inhibit:** The ramp function generator accepts the value at *nCInhVal_a* and provides it at its output. |
| **TRUE** | FALSE | FALSE | TRUE | |
| **TRUE** | FALSE | TRUE | FALSE | |
| **TRUE** | FALSE | TRUE | TRUE | |
| **TRUE** | TRUE | FALSE | FALSE | |
| **TRUE** | TRUE | FALSE | TRUE | |
| **TRUE** | TRUE | TRUE | FALSE | |
| **TRUE** | TRUE | TRUE | TRUE | |

### 18.1.117.7 S-shaped ramp

A PT1 element is connected downstream of the linear ramp function generator. This arrangement implements an S-shaped ramp for a nearly jerk-free acceleration and deceleration.

- The PT1 element can be switched on/off via the *bSShapeActive* input.
- The corresponding S-ramp time can be set under C00182.

### 18.1.117.8 Additional setpoint

Use the *nNAdd_a* input to define an additional value (e.g. a correcting signal) and combine it arithmetically with the main setpoint *nNSet_a*.

- First, the additional setpoint is led via a ramp function generator with linear characteristic. Its Ti times can be set in C00220 (acceleration time) and C00221 (deceleration time).
- When the input *bNAddInv* is set to TRUE, the additional setpoint can be inverted before having an effect on the ramp function generator.
- When the input *bLoad* is set to TRUE, the ramp function generator is set to zero for the additional setpoint and held there without considering the Ti times. The same applies when the controller is inhibited.
- The following arithmetic combination of main setpoint and additional setpoint can be selected in C00190:

| Value in C00190 | Function | Info |
|---|---|---|
| 0 | *nNOut_a = nNSet_a* | The additional setpoint *nNAdd_a* is not processed. |
| 1 | *nNOut_a = nNSet_a + nNAdd_a* | |
| 2 | *nNOut_a = nNSet_a - nNAdd_a* | |
| 3 | *nNOut_a* = (*nNSet_a* * *nNAdd_a*) / 100 % | Internal scaling:<br>• 100 % ≡ 16384<br>• 1 % ≡ 164 |
| 4 | *nNOut_a* = (*nNSet_a* * 1 %) / \|*nNAdd_a*\| | |
| 5 | *nNOut_a* = (*nNSet_a* * 100 %) / (100 % - *nNAdd_a*) | |

### 18.1.117.9 Application example for the additional load function

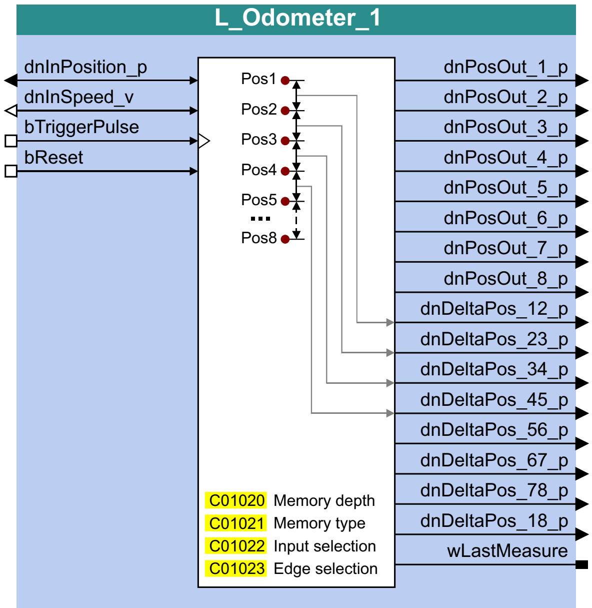
The motor control of the drive is provided with a function for automatically carrying along ramp function generators for "jerk-free" setpoint connection. For speed-controlled drive tasks, the LS_MotorInterface SB outputs the current actual speed value via the *nHlgSetValue_a* output (e.g. in case of a pulse inhibit, flying restart, controller inhibit).

- In case of a pulse inhibit, the main setpoint generator must be carried along with the current actual speed value to ensure jerk-free setpoint transfer.
- The actual speed value is carried along automatically if the following wiring is provided:
  - LS_MotorInterface.*nHlgSetValue_a* → L_NSet_1.*nCInhValue_a*
  - LS_MotorInterface.*bHlgLoad* → L_NSet_1.*bExternalCINH*

### 18.1.118 L_Odometer_1

This FB is used to detect positions and calculate distances.

- Depending on the input selection set in C01022, a position signal can be detected at the *dnInPosition_p* input or a speed can be detected at the *nInSpeed_v* input.
- The position is detected/accepted via the edge selected in C01023 at the *bTriggerPulse* input.
- If "Ring buffer" is selected in C01021, the measurement will start all over again after the number of measurements selected in C01020 has been performed and the old values will be overwritten. Otherwise, the measurement will stop.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnInPosition_p<br>DINT | Position measurement input<br>• Input is only evaluated if C01022 = "0: Pos input" has been set. |
| nInSpeed_v<br>INT | Speed measurement input<br>• The speed is internally integrated into a position.<br>• Input is only evaluated if C01022 = "1: V input" has been set. |
| bTriggerPulse<br>BOOL | Detect position/speed<br>• The tripping edge can be parameterised in C01023. |
| bReset<br>BOOL | Reset measurement results and internal integrator<br>TRUE: Reset measurement results and internal integrator. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnPosOut_1_p<br>...<br>dnPosOut_8_p<br>DINT | Detected positions |
| dnDeltaPos_12_p<br>dnDeltaPos_23_p<br>dnDeltaPos_34_p<br>...<br>dnDeltaPos_18_p<br>DINT | Calculated distances between the detected positions<br>• *dnDeltaPos_12_p* = Distance between *dnPosOut_1_p* and *dnPosOut_1_p*<br>• *dnDeltaPos_23_p* = Distance between *dnPosOut_2_p* and *dnPosOut_3_p*<br>• *dnDeltaPos_34_p* = Distance between *dnPosOut_3_p* and *dnPosOut_4_p*<br>...<br>• *dnDeltaPos_18_p* = Distance between *dnPosOut_1_p* and *dnPosOut_8_p* |
| wLastMeasure<br>WORD | Memory location number (1 ... 8) of the last measurement |

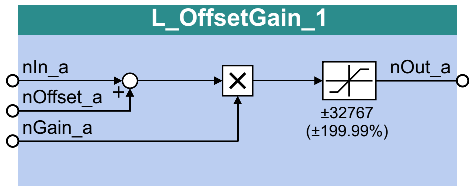
### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01020 | | | Memory size<br>• Lenze setting: 7 measurements |
| | 1 | 1 measurement | |
| | 2 | 2 measurements | |
| | ... | ... | |
| | 7 | 7 measurements | |
| C01021 | | | Memory type<br>• Lenze setting: No ring buffer |
| | 0 | No ring buffer | |
| | 1 | Ring buffer | |
| C01022 | | | Input selection<br>• Lenze setting: Pos input |
| | 0 | Pos input | |
| | 1 | V input | |
| C01023 | | | Edge selection<br>• Lenze setting: HIGH edge |
| | 0 | High edge | |
| | 1 | Low edge | |
| | 2 | High and low edge | |

### 18.1.119 L_OffsetGain_1

This FB can add an offset to an analog input signal and then amplify it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

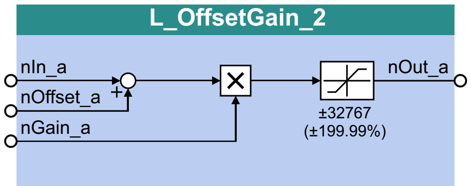
#### Function

$$\text{nOut\_a} = (\text{nIn\_a} + \text{Offset}) \cdot \text{Gain factor}$$

## 18.1.120 L_OffsetGain_2

This FB can add an offset to an analog input signal and then amplify it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

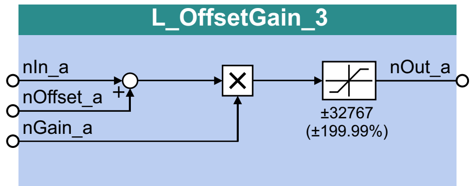
### Function

$$\text{nOut\_a} = (\text{nIn\_a} + \text{Offset}) \cdot \text{Gain factor}$$

## 18.1.121 L_OffsetGain_3

This FB can add an offset to an analog input signal and then amplify it. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via FB inputs.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal<br>• Scaling: 16384 ≡ 100 % |
| nOffset_a<br>INT | Offset<br>• Scaling: 16384 ≡ 100 % |
| nGain_a<br>INT | Gain factor<br>• Scaling: 16384 ≡ 100 %<br>• 199.99 % ≈2 |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

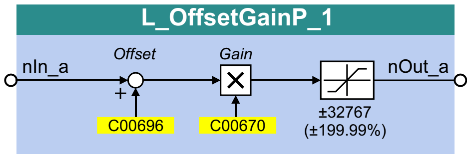
### Function

$$nOut\_a = (nIn\_a + Offset) \cdot Gain\ factor$$

## 18.1.122 L_OffsetGainP_1

This FB can add an offset to an analog input signal and amplify it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00670 | -100.0000 | | 100.0000 | Gain factor<br>• High gain factor for further processing of smallest input signals.<br>• Please observe the difference with regard to the gain factors of other blocks in percent (±199.99 % ≈ 2).<br>• Lenze setting: 1.0000 |
| C00696 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

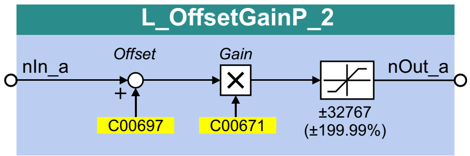
### Function

$$nOut\_a = (nIn\_a + Offset) \cdot Gain\ factor$$

### 18.1.123 L_OffsetGainP_2

This FB can add an offset to an analog input signal and amplify it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Identifier / Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

| Identifier / Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00671 | -100.0000 | | 100.0000 | Gain factor<br>• High gain factor for further processing of smallest input signals.<br>• Please observe the difference with regard to the gain factors of other blocks in percent (±199.99 % ≈ 2).<br>• Lenze setting: 1.0000 |
| C00697 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

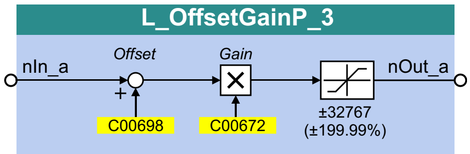
#### Function

$$\text{nOut\_a} = (\text{nIn\_a} + \text{Offset}) \cdot \text{Gain factor}$$

### 18.1.124 L_OffsetGainP_3

This FB can add an offset to an analog input signal and amplify it afterwards. Preferably to be interconnected directly after the analog input terminals.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *nOut_a* output is internally limited to ±199.99 %.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199.99 % |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00672 | -100.0000 | | 100.0000 | Gain factor<br>• High gain factor for further processing of smallest input signals.<br>• Please observe the difference with regard to the gain factors of other blocks in percent (±199.99 % ≈ 2).<br>• Lenze setting: 1.0000 |
| C00698 | -199.99 | % | 199.99 | Offset<br>• Lenze setting: 0.00 % |

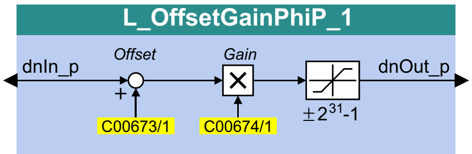
#### Function

$$nOut\_a = (nIn\_a + Offset) \cdot Gain\ factor$$

## 18.1.125 L_OffsetGainPhiP_1

This FB can add an offset to an angle signal and amplify it afterwards.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *dnOut_p* output is internally limited to $\pm 2^{31}$-1.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}$-1 (±2147483647) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00673/1 | -2147483647 | Incr. | 2147483647 | Offset<br>• Lenze setting: 0 incr. |
| C00674/1 | -2147483647 | | 2147483647 | Gain factor<br>• Lenze setting: 65536 |

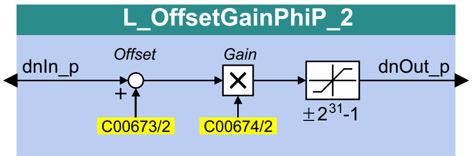
### Function

$$\text{dnOut\_p} = (\text{dnIn\_p} + \text{Offset}) \cdot \text{Gain factor}$$

## 18.1.126 L_OffsetGainPhiP_2

This FB can add an offset to an angle signal and amplify it afterwards.

- The internal calculations (addition and subtraction) are carried out with 32 bits without overflow/underflow. Division is not remainder considered.
- Offset and gain are selected via parameters.
- The value provided at the *dnOut_p* output is internally limited to $\pm 2^{31}-1$.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal<br>• Internal limitation to $\pm 2^{31}-1$ (±2147483647) |

### Parameter

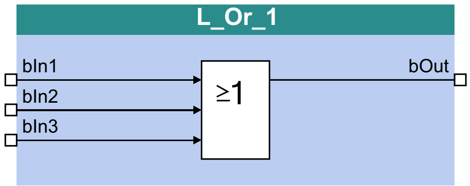
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00673/2 | -2147483647 | Incr. | 2147483647 | Offset<br>• Lenze setting: 0 incr. |
| C00674/2 | -2147483647 | | 2147483647 | Gain factor<br>• Lenze setting: 65536 |

### Function

$$\text{dnOut\_p} = (\text{dnIn\_p} + \text{Offset}) \cdot \text{Gain factor}$$

## 18.1.127 L_Or_1

This FB implements the ORing of the inputs signals.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

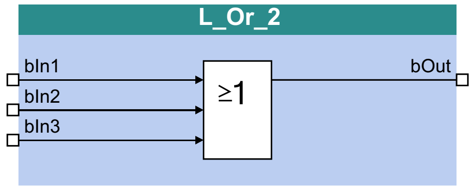
### Function

| Inputs | | | Output |
|---|---|---|---|
| **bIn3** | **bIn2** | **bIn1** | **bOut** |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | |

[18-46] Truth table of the FB L_Or_1

## 18.1.128 L_Or_2

This FB implements the ORing of the inputs signals.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

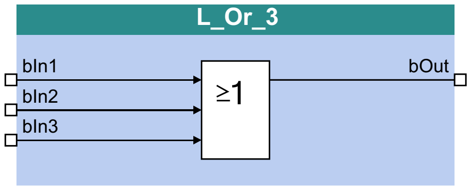
### Function

| Inputs | | | Output |
|---|---|---|---|
| **bIn3** | **bIn2** | **bIn1** | **bOut** |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | |

[18-47] Truth table of the FB L_Or_2

## 18.1.129 L_Or_3

This FB implements the ORing of the inputs signals.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

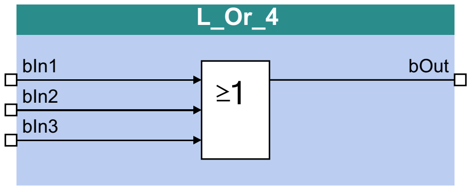
### Function

| Inputs | | | Output |
|---|---|---|---|
| **bIn3** | **bIn2** | **bIn1** | **bOut** |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | |

[18-48] Truth table of the L_Or_3 FB

## 18.1.130 L_Or_4

This FB implements the ORing of the inputs signals.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>bIn2<br>bIn3<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

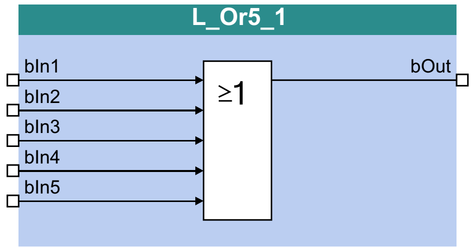
### Function

| Inputs | | | Output |
|---|---|---|---|
| **bIn3** | **bIn2** | **bIn1** | **bOut** |
| FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | TRUE | TRUE |
| FALSE | TRUE | FALSE | |
| FALSE | TRUE | TRUE | |
| TRUE | FALSE | FALSE | |
| TRUE | FALSE | TRUE | |
| TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | |

[18-49] Truth table of the L_Or_4 FB

## 18.1.131 L_Or5_1

This FB implements the ORing of the inputs signals.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

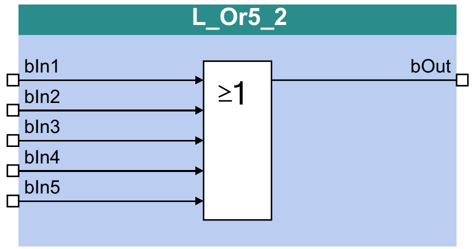
### Function

| Inputs | | | | | Output |
|---|---|---|---|---|---|
| **bIn5** | **bIn4** | **bIn3** | **bIn2** | **bIn1** | **bOut** |
| FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | FALSE | FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE | TRUE | FALSE | |
| FALSE | FALSE | FALSE | TRUE | TRUE | |
| FALSE | FALSE | TRUE | FALSE | FALSE | |
| | | ... | | | |
| TRUE | TRUE | TRUE | FALSE | TRUE | |
| TRUE | TRUE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE | TRUE | |

[18-50] Truth table of the FB L_Or5_1

### 18.1.132 L_Or5_2

This FB implements the ORing of the inputs signals.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn1<br>...<br>bIn5<br>BOOL | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |

#### Function

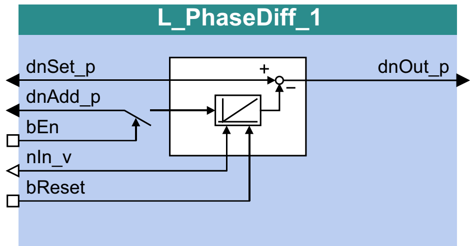
| | | Inputs | | | Output |
|---|---|---|---|---|---|
| bIn5 | bIn4 | bIn3 | bIn2 | bIn1 | bOut |
| FALSE | FALSE | FALSE | FALSE | FALSE | FALSE |
| FALSE | FALSE | FALSE | FALSE | TRUE | TRUE |
| FALSE | FALSE | FALSE | TRUE | FALSE | |
| FALSE | FALSE | FALSE | TRUE | TRUE | |
| FALSE | FALSE | TRUE | FALSE | FALSE | |
| | | ... | | | |
| TRUE | TRUE | TRUE | FALSE | TRUE | |
| TRUE | TRUE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | TRUE | TRUE | TRUE | |

[18-51] Truth table of the FB L_Or5_2

### 18.1.133 L_PhaseDiff_1

This FB generates a position difference for the defined position setpoint from a position value and a speed signal.

- In an integrator, the *nIn_v* speed signal is integrated into a position value and subtracted from the *dnSet_p* position setpoint.
- In addition, the adaptive *dnAdd_p* position value can be added to the integrator content by setting *bEn* to TRUE.




#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| dnSet_p<br>DINT | Selection of a position setpoint | |
| dnAdd_p<br>DINT | Adaptive position value for the actual position | |
| bEn<br>BOOL | Activate addition of the adaptive position value | |
| | FALSE | 1. The speed signal at *nIn_v* is integrated by the angle integrator.<br>2. The result of the angle integrator is subtracted from the angle signal at *dnSet_p* and output at *dnOut_p* afterwards. |
| | TRUE | The adaptive position value selected via *dnAdd_p* is added:<br>1. The speed signal at *nIn_v* is integrated by the angle integrator.<br>2. The angle signal at *dnAdd_p* is added to the integrated speed signal in every task cycle.<br>3. The result of the angle integrator is subtracted from the angle signal at *dnSet_p* and output at *dnOut_p* afterwards. |
| nIn_v<br>INT | Selection of the actual speed to be converted into the position value | |
| bReset<br>BOOL | Reset actual angle integrator | |
| | TRUE | Actual angle integrator is set to "0". |

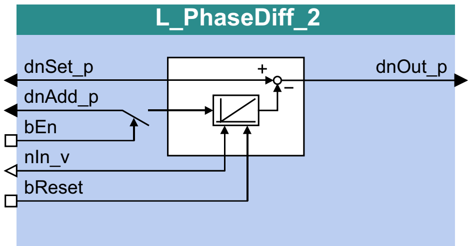
#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output angle signal (position difference)<br>• Without limitation |

### 18.1.134 L_PhaseDiff_2

This FB generates a position difference for the defined position setpoint from a position value and a speed signal.

- In an integrator, the *nIn_v* speed signal is integrated into a position value and subtracted from the *dnSet_p* position setpoint.
- In addition, the adaptive *dnAdd_p* position value can be added to the integrator content by setting *bEn* to TRUE.




#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| dnSet_p<br>DINT | Selection of a position setpoint | |
| dnAdd_p<br>DINT | Adaptive position value for the actual position | |
| bEn<br>BOOL | Activate addition of the adaptive position value | |
| | FALSE | 1. The speed signal at *nIn_v* is integrated by the angle integrator.<br>2. The result of the angle integrator is subtracted from the angle signal at *dnSet_p* and output at *dnOut_p* afterwards. |
| | TRUE | The adaptive position value selected via *dnAdd_p* is added:<br>1. The speed signal at *nIn_v* is integrated by the angle integrator.<br>2. The angle signal at *dnAdd_p* is added to the integrated speed signal in every task cycle.<br>3. The result of the angle integrator is subtracted from the angle signal at *dnSet_p* and output at *dnOut_p* afterwards. |
| nIn_v<br>INT | Selection of the actual speed to be converted into the position value | |
| bReset<br>BOOL | Reset actual angle integrator | |
| | TRUE | Actual angle integrator is set to "0". |

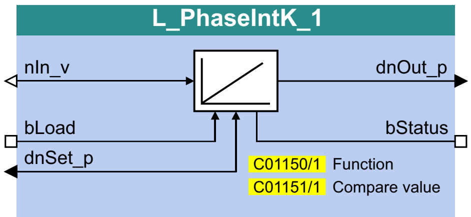
#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output angle signal (position difference)<br>• Without limitation |

## 18.1.135 L_PhaseIntK_1

The FB can integrate a speed or velocity to an angle (path). In addition, the FB can recognise a relatively covered distance.

- The integrator can take max. ±32000 encoder revolutions.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_v<br>INT | Selection of the actual speed<br>• 16384 ≡ 15000 rpm | |
| bLoad<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnSet_p* and *bStatus* is reset to FALSE. |
| dnSet_p<br>DINT | Starting value for angle integrator | |

### Outputs

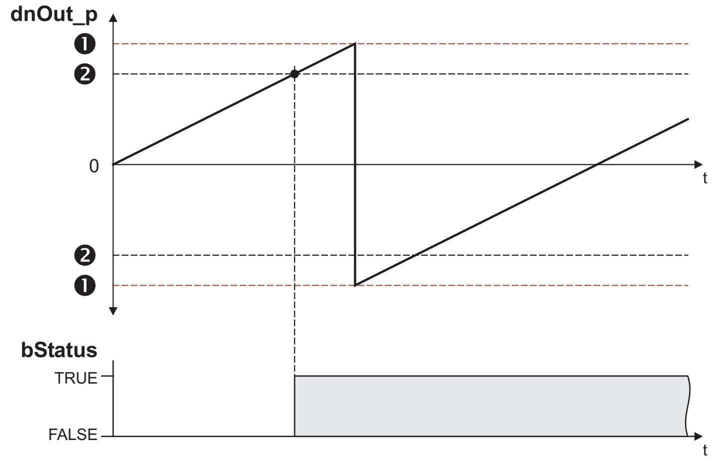
| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| dnOut_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bStatus*) | |
| bStatus<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bLoad*. | |
| | TRUE | Overflow has occurred or distance is processed. |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01150/1 | | | | Function |
| | 0 | Loading with level | | Load integrator with TRUE level at the input *bLoad*(Lenze setting) |
| | 1 | Loading with edge | | Load integrator with FALSE/TRUE edge at the input *bLoad*. |
| | 2 | Loading with level + reset | | Load integrator when reaching the comparison value or with TRUE level at the input *bLoad*. |
| C01151/1 | 0 | | 2000000000 | Comparison value<br>• Is valid for both the positive and the negative value range.<br>• Lenze setting: 0 |

### 18.1.135.1 Function at constant input value

Selection: C01150 = "0: Loading with level" or "1: Loading with edge"



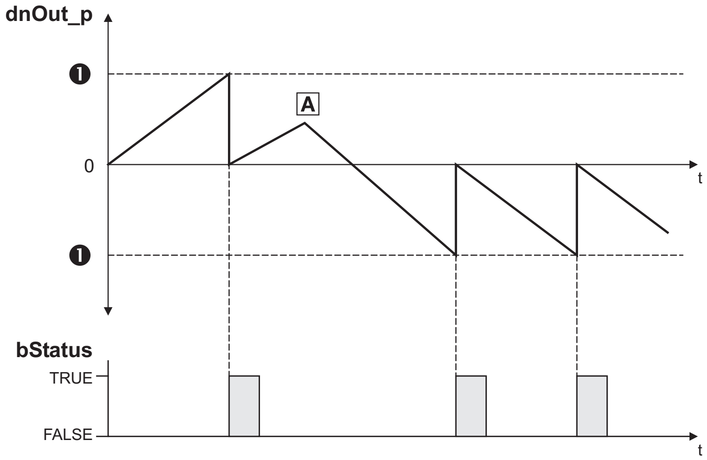

❶ Overflow at ±2147483648 incr. (±32767 encoder revolutions)
❷ C01151: Comparison value (applies to both positive and negative value range)

[18-52] Switching performance if the overflow is in the positive direction

- If "0: Loading with level" is selected in C01150, the *bLoad* input is status-controlled: In case of a TRUE signal, the integrator is loaded with the value at *dnSet_p* and the *bStatus* output is set to FALSE.
- If "1: Loading with edge" is selected in C01150, the *bLoad* input is edge-controlled: In case of a FALSE/TRUE edge, the integrator is loaded with the value at *dnSet_p* and then immediately continues to integrate, the *bStatus* output is set to FALSE.
- A positive signal at *nIn_v* is incremented (the counter content is increased with every function call).
- A negative signal at *nIn_v* is decremented (the counter content is reduced with every function call).
- *dnOut_p* outputs the counter content of the bipolar integrator.
  - If the counter content exceeds a value of +32767 encoder revolutions (corresponds to +2147483647 incr.), an overflow occurs and the counting process continues at a value of -32768 encoder revolutions.
  - If the counter content falls below a value of -32768 encoder revolutions (corresponds to -2147483648 incr.), an overflow occurs and the counting process starts at a value of +32767 encoder revolutions.
- *bStatus* will be set to TRUE if the comparison value set in C01151/x is reached.

### 18.1.135.2 Function at input value with sign reversal

Selection: C01150 = "2: Loading with level + reset"




A Sign reversal of the value at *nIn_v*
❶ C01151/x: Comparison value (is valid for both the positive and the negative value range)

[18-53] Switching performance if the input signal changes signs

- If "2: Loading with level + reset" is selected in C01150, the *bLoad* input is status-controlled: In case of a TRUE signal, the integrator is loaded with the value at *dnSet_p* and the *bStatus* output is set to FALSE.
- A positive signal at *nIn_v* is incremented (the counter content is increased with every function call).
- A negative signal at *nIn_v* is decremented (the counter content is reduced with every function call).
- *dnOut_p* outputs the counter content of the bipolar integrator.
  - If the positive counter content is higher than the comparison value set in C01151/x, the comparison value will be subtracted from the counter content, and *bStatus* will be set to TRUE for one task cycle.
  - If the negative counter content is lower than the comparison value set in C01151/x, the comparison value will be added to the counter content, and *bStatus* will be set to TRUE for one task cycle.

### 18.1.135.3 Calculation of the output signal

The output value at *dnOut_p* can be detected according to the following formula:

$$\text{dnOut\_p [incr.]} = \text{nIn\_v [rpm]} \cdot \text{t [s]} \cdot \text{65535 [incr./rev.]}$$

t = integration time
16384 ≡ 15000 rpm
1 ≡ 1 incr.

#### Example

You want to determine the counter content of the integrator at a certain speed at the input and a certain integration time **t**.

Given values:

- *nIn_v* = 1000 rpm ≈ integer value 1092
- Integration time t = 10 s
- Starting value of the integrator = 0

Solution:

- Conversion of the input signal at *nIn_v*:

$$1000\ \text{rpm} = \frac{1000\ \text{rev.}}{60\ \text{s}}$$

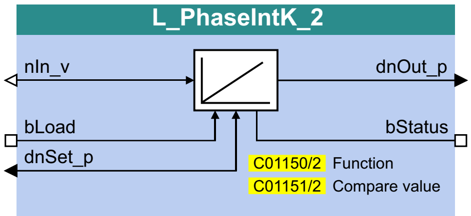
- Calculation of the output value:

$$\text{dnOut\_p} = \frac{1000\ \text{rev.}}{60\ \text{s}} \cdot 10\ \text{s} \cdot \frac{65535\ \text{incr.}}{\text{Rev.}} = 10922666\ \text{incr.}$$

## 18.1.136 L_PhaseIntK_2

The FB can integrate a speed or velocity to an angle (path). In addition, the FB can recognise a relatively covered distance.

▶ The integrator can take max. ±32000 encoder revolutions.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_v<br>INT | Selection of the actual speed<br>• 16384 ≡ 15000 rpm | |
| bLoad<br>BOOL | Load angle integrator with starting value and reset status signal | |
| | TRUE | Angle integrator is loaded with the value at *dnSet_p* and *bStatus* is reset to FALSE. |
| dnSet_p<br>DINT | Starting value for angle integrator | |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| dnOut_p<br>DINT | Angle output signal<br>• 65536 [incr.] ≡ 1 encoder revolution<br>• Overflow is possible (display via *bStatus*) | |
| bStatus<br>BOOL | Status signal "Overflow occurred/distance processed"<br>• Status signal can be reset via *bLoad*. | |
| | TRUE | Overflow has occurred or distance is processed. |

### Parameter

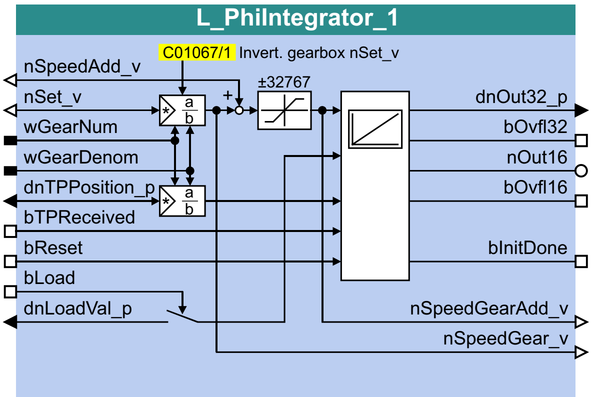
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01150/2 | | | | Function |
| | 0 | Loading with level | | Load integrator with TRUE level at the input *bLoad*(Lenze setting) |
| | 1 | Loading with edge | | Load integrator with FALSE/TRUE edge at the input *bLoad*. |
| | 2 | Loading with level + reset | | Load integrator when reaching the comparison value or with TRUE level at the input *bLoad*. |
| C01151/2 | 0 | | 2000000000 | Comparison value<br>• Is valid for both the positive and the negative value range.<br>• Lenze setting: 0 |

For a detailed functional description see L_PhaseIntK_1.

## 18.1.137 L_PhiIntegrator_1

This FB evaluates a speed with a gearbox factor and adds them in an integrator.

- The integrator value is output as 16-bit value and 32-bit value.
- An overflow of the prevailing output is signalled.
- The integrator can be loaded and initialised (e.g. with a touch probe signal).




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nSpeedAdd_v<br>INT | Additive speed | |
| nSet_v<br>INT | Speed input (angular difference) | |
| wGearNum<br>WORD | Gearbox factor (numerator) | |
| wGearDenom<br>WORD | Gearbox factor (denominator)<br>• When *wGearDenom* = 0, it is internally calculated with 1. | |
| dnTPPosition_p<br>DINT | Initialisation value | |
| bTPReceived<br>BOOL | Initialise integrator | |
| | FALSE⇗TRUE | Initialise integrator |
| bReset<br>BOOL | Reset integrator | |
| | TRUE | Reset integrator<br>• The integrator is not integrated. |
| bLoad<br>BOOL | Load integrator | |
| | TRUE | Load integrator with *dnLoadVal_p*<br>• The integrator is not integrated.<br>• The *bInitDone* output is reset to FALSE with a delay of one call cycle after *bLoad* is reset to FALSE and a subsequent FALSE/TRUE edge to *bTPReceived*. |
| dnLoadVal_p<br>DINT | Value the integrator is to be loaded with. | |

## Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| dnOut32_p<br>DINT | Integrator value (32 bits) | |
| bOvfl32<br>BOOL | Status signal "overflow (32 bits)" | |
| | TRUE | Overflow |
| nOut16<br>INT | Integrator value (16 bits) | |
| bOvfl16<br>BOOL | Status signal "overflow (16 bits)" | |
| | TRUE | Überlauf |
| bInitDone<br>BOOL | Status signal "Initialisation completed" | |
| | TRUE | Integrator is initialised |
| nSpeedGearAdd_v<br>INT | Resulting speed with additive speed<br>(*nSet_v* * gearbox factor + *nSpeedAdd_v*) | |
| nSpeedGear_v<br>INT | Resulting speed without additive speed<br>(*nSet_v* * gearbox factor) | |

## Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01067/1 | | | Invert. gearbox nSet_v |
| | 0 | not inverted | |
| | 1 | inverted | |
| | 2 | Automatically from MCK | |

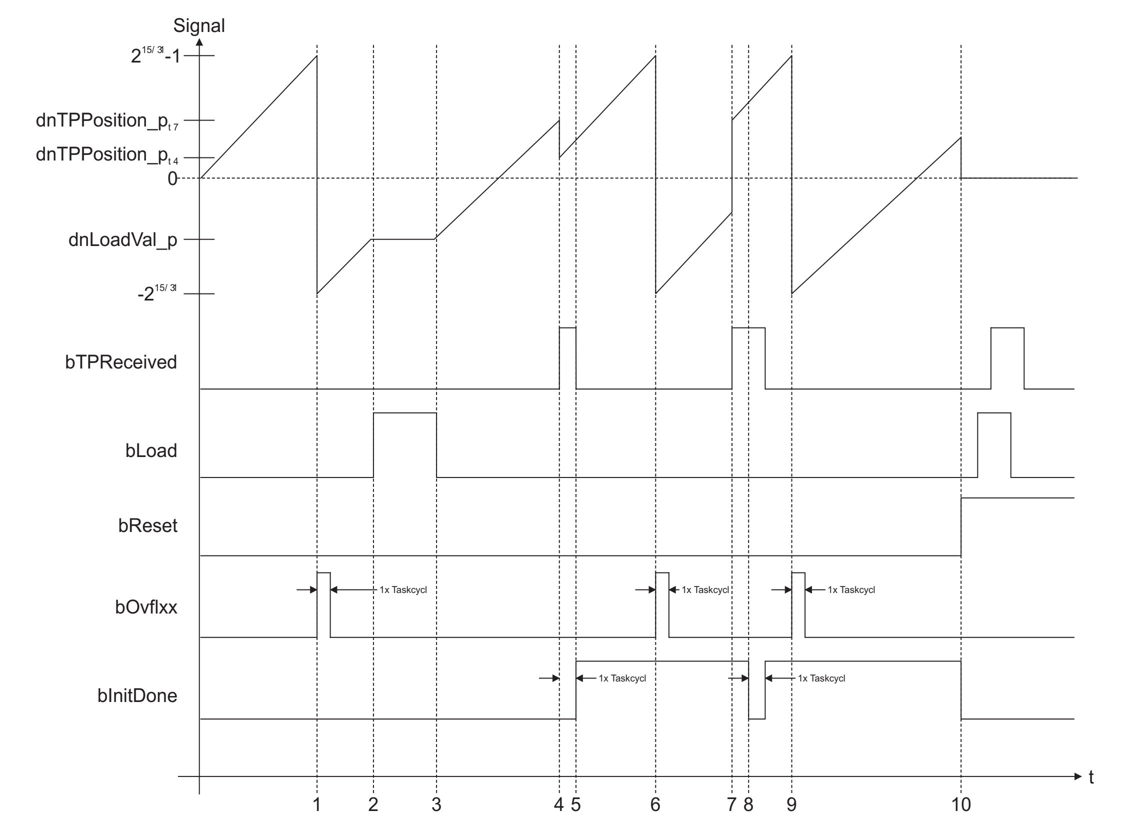
### 18.1.137.1 Function

The *nSet_v* input value is evaluated, integrated and output as 16-bit value ( *nOut16*) and 32-bit value *dnOut32_p*) with the gearbox factors *wGearNum* and *wGearDenom*.

$$\text{nOut16} = \text{nOut16} + \text{nSet\_v} \cdot \frac{\text{wGearNum}}{\text{wGearDenom}}$$

$$\text{dnOut32\_p} = \text{dnOut32\_p} + \text{nSet\_v} \cdot \frac{\text{wGearNum}}{\text{wGearDenom}}$$

- Evaluation of the gearbox factor is remainder considered.
- A positive or negative overflow of the integrator is signalised by a TRUE signal (for a task cycle) at the *bOvfl16* output for the *nOut16* output or at the *bOvfl32* output for the *dnOut32_p* output.




[18-54] Signal characteristic

### 18.1.137.2 Example

The current speed at *nSet_v* and the gearbox factors serve to create a specific machine measuring system from the motor measuring system.

- 216 increments in the motor measuring system are to correspond to 1 motor revolution.
- 216 increments in the machine measuring system (*nOut16*) are to correspond to 1 machine shaft revolution.

**Initialise integrator**

A FALSE/TRUE edge at *bTPReceived* initialises the integrator, i.e. a calculation with *dnTPPosition_p* and *dnLoadVal_p* is made:

$$\begin{aligned} \text{nOut16} &= (\text{INT})(\text{dnLoadVal\_p} + \text{dnTPPosition\_p} \cdot \text{Gearbox factor}) \\ \text{dnOut32\_p} &= \text{dnLoadVal\_p} + \text{dnTPPosition\_p} \cdot \text{Gearbox factor} \end{aligned}$$

- After initialisation is completed, the *bInitDone* output is set to TRUE in the next call cycle.
- In case of a renewed initialisation without a previous reset (*bReset* = TRUE), the *bInitDone* output is set to FALSE for a task cycle.

**Reset integrator**

When *bReset* is set to TRUE, the integrator is set to 0. No calculations are made.

- The Boolean outputs are set to FALSE.
- The *bReset* input has the highest priority.

**Load integrator**

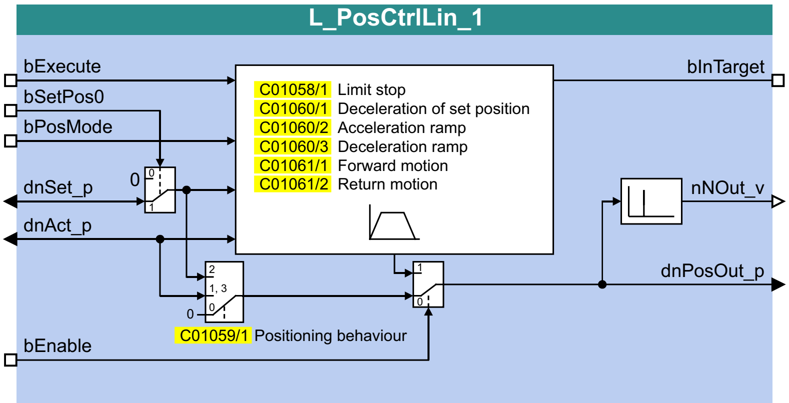
When *bLoad* is set to TRUE, the integrator is loaded with the value at *dnLoadVal_p* and the outputs are set.

- When *bLoad* is set to TRUE, the input is not integrated and no check for overflow is executed.
- When *bLoad* is reset to FALSE, the integrator continues from the loaded value. The *bInitDone* output is reset to FALSE after a FALSE/TRUE edge at *bTPReceived* with a delay of one call cycle.

### 18.1.138 L_PosCtrlLin_1

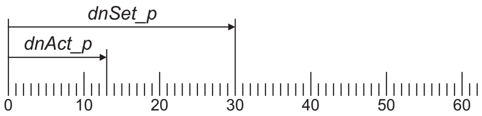
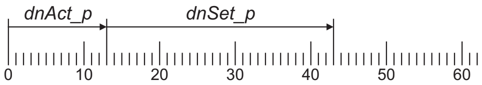
This FB serves to implement the following positioning functions:

- Bring drive to curve position (e.g. after mains connection, manual jog, homing)
- Release drive from curve and bring it to safe position (position override function)
- Positioning the X axis via X offset (higher-level positioning)
- Positioning the Y axis via Y offset (higher-level positioning)




#### Inputs

| Identifier / Data type | | Information/possible settings |
|---|---|---|
| bExecute<br>BOOL | | Execute positioning profile<br>• Only possible if the *bEnable* input is set to TRUE. |
| | FALSE | No positioning/positioning is aborted.<br>• The speed at the *nNOut_v* output is braked to standstill via the deceleration ramp set in C01060/1 (*nNOut_v* = 0).<br>• At the same time, the *dnPosOut_p* output is stopped. |
| | FALSE↗TRUE | Positioning is executed/continued.<br>• In case of an activated limit stop (C01058/1 = TRUE), a renewed positive edge for a following positioning procedure is required. |
| bSetPos0<br>BOOL | | Travel to zero position<br>• Only possible if the *bEnable* input is set to TRUE. |
| | FALSE | The value at the *dnSet_p* input is used as setpoint position. |
| | TRUE | The value at the *dnSet_p* input is ignored. The setpoint position is internally set to "0" and the *dnPosOut_p* output can be travelled to zero position. |

| Identifier (Data type) | | Information/possible settings |
|---|---|---|
| bPosMode (BOOL) | | Positioning mode |
| | FALSE | Absolute positioning<br>• The setpoint at *dnSet_p* is the absolute setpoint position (with regard to zero position):<br> |
| | TRUE | Relative positioning<br>• With a FALSE/TRUE edge at the *bExecute* input, the outputs of the current position (*dnAct_p*) are traversed by the *dnSet_p* value:<br><br>**Note:**<br>Only to be used with activated limit stop (C01058/1 = TRUE)! |
| dnSet_p (DINT) | | Setpoint for positioning in [increments]<br>• Scaling: a revolution is displayed with 65536 increments or steps. |
| dnAct_p (DINT) | | Actual position in [increments]<br>• Connect this input with e.g. *dnPosOut_p* of this FB or with another FB which outputs the actual position of the drive. |
| bEnable (BOOL) | | Activate/deactivate positioning function<br>• This input has the highest priority. |
| | FALSE | Positioning function deactivated.<br>• For behaviour see C01059/1. |
| | TRUE | Positioning function activated. |

## Outputs

| Identifier (Data type) | | Value/meaning |
|---|---|---|
| bInTarget (BOOL) | | Status signal "Target position reached" |
| | TRUE | Target position reached. |
| nNOut_v (INT) | | Speed output of the profile generator<br>• Scaling: 16384 ≡ 15000 rpm |
| dnPosOut_p (DINT) | | Output position of the profile generator in [increments]<br>• Scaling: a revolution is displayed with 65536 increments or steps. |

## Parameter

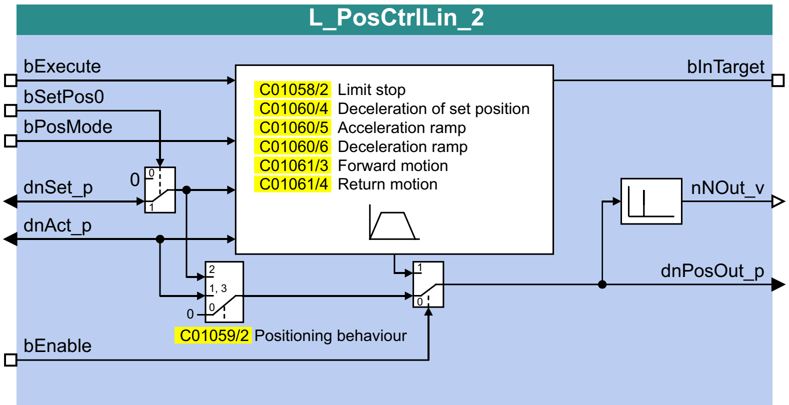
| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01058/1 | | | Limit stop |
| | 0 | deactivated<br>(Lenze setting) | The outputs continuously follow the setpoint at *dnSet_p*. |
| | 1 | activated | The outputs follow the setpoint at *dnSet_p* once.<br>• If the setpoint position changes, a new FALSE/TRUE edge is required at the *bExecute* input for another positioning process. |

| Parameter | Possible settings | | | | Info |
|---|---|---|---|---|---|
| C01059/1 | | | | | Positioning behaviour<br>• Behaviour with deactivated positioning function (*bEnable* = FALSE). |
| | 0 | *dnOut_p* = 0<br>(Lenze setting) | | | The *dnPosOut_p* output jumps to "0".<br>• The speed signal at the *nNOut_v* output maps this position step change as angle signal.<br>**Note:**<br>If the *nNOut_v* output has an impact on the setpoint generation for the drive, undefined motions (shocks, following errors) may occur. |
| | 1 | *dnOut_p/nNOut_v* follow *dnAct_p* | | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnAct_p* input (actual position). |
| | 2 | *dnOut_p/nNOut_v* follow *dnSet_p* | | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnSet_p* input (setpoint). |
| | 3 | *dnOut_p/nNOut_v* follow *dnAct_p*<br>(without limitation) | | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnAct_p* input (actual position), but without limitation. |
| C01060/1 | 0.010 | | s | 130.000 | Deceleration of set position<br>• Deceleration ramp for stop before reaching the setpoint position (*bExecute* = FALSE).<br>• Lenze setting: 1.000 s |
| C01060/2 | 0.010 | | s | 130.000 | Acceleration ramp<br>• Lenze setting: 1.000 s |
| C01060/3 | 0.010 | | s | 130.000 | Deceleration ramp<br>• Lenze setting: 1.000 s |
| C01061/1 | -15000 | | rpm | 15000 | Forward motion<br>• Positive speed<br>• Lenze setting: 200 rpm |
| C01061/2 | -15000 | | rpm | 15000 | Return motion<br>• Negative speed<br>• Lenze setting: 200 rpm |

## 18.1.139 L_PosCtrlLin_2

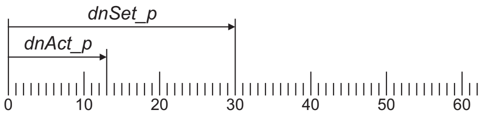
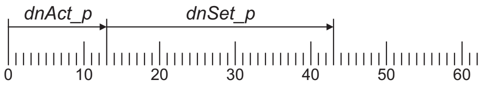
This FB serves to implement the following positioning functions:

- Bring drive to curve position (e.g. after mains connection, manual jog, homing)
- Release drive from curve and bring it to safe position (position override function)
- Positioning the X axis via X offset (higher-level positioning)
- Positioning the Y axis via Y offset (higher-level positioning)




### Inputs

| Identifier (Data type) | | Information/possible settings |
|---|---|---|
| bExecute (BOOL) | | Execute positioning profile<br>• Only possible if the *bEnable* input is set to TRUE. |
| | FALSE | No positioning/positioning is aborted.<br>• The speed at the *nNOut_v* output is braked to standstill via the deceleration ramp set in C01060/4 (*nNOut_v* = 0).<br>• At the same time, the *dnPosOut_p* output is stopped. |
| | FALSE↗TRUE | Positioning is executed/continued.<br>• In case of an activated limit stop (C01058/2 = TRUE), a renewed positive edge for a following positioning procedure is required. |
| bSetPos0 (BOOL) | | Travel to zero position<br>• Only possible if the *bEnable* input is set to TRUE. |
| | FALSE | The value at the *dnSet_p* input is used as setpoint position. |
| | TRUE | The value at the *dnSet_p* input is ignored. The setpoint position is internally set to "0" and the *dnPosOut_p* output can be travelled to zero position. |

| Identifier | Data type | | Information/possible settings |
|---|---|---|---|
| bPosMode | BOOL | | Positioning mode |
| | | FALSE | Absolute positioning<br>• The setpoint at *dnSet_p* is the absolute setpoint position (with regard to zero position):<br><br> |
| | | TRUE | Relative positioning<br>• With a FALSE/TRUE edge at the *bExecute* input, the outputs of the current position (*dnAct_p*) are traversed by the *dnSet_p* value:<br><br><br>**Note:**<br>Only to be used with activated limit stop (C01058/2 = TRUE)! |
| dnSet_p | DINT | | Setpoint for positioning in [increments]<br>• Scaling: a revolution is displayed with 65536 increments or steps. |
| dnAct_p | DINT | | Actual position in [increments]<br>• Connect this input with e.g. *dnPosOut_p* of this FB or with another FB which outputs the actual position of the drive. |
| bEnable | BOOL | | Activate/deactivate positioning function<br>• This input has the highest priority. |
| | | FALSE | Positioning function deactivated.<br>• For behaviour see C01059/2. |
| | | TRUE | Positioning function activated. |

## Outputs

| Identifier | Data type | | Value/meaning |
|---|---|---|---|
| bInTarget | BOOL | | Status signal "Target position reached" |
| | | TRUE | Target position reached. |
| nNOut_v | INT | | Speed output of the profile generator<br>• Scaling: 16384 ≡ 15000 rpm |
| dnPosOut_p | DINT | | Output position of the profile generator in [increments]<br>• Scaling: a revolution is displayed with 65536 increments or steps. |

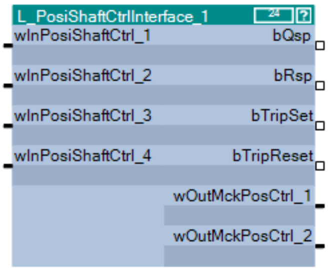
## Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C01058/2 | | | Limit stop |
| | 0 | deactivated<br>(Lenze setting) | The outputs continuously follow the setpoint at *dnSet_p*. |
| | 1 | activated | The outputs follow the setpoint at *dnSet_p* once.<br>• If the setpoint position changes, a new FALSE/TRUE edge is required at the *bExecute* input for another positioning process. |

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01059/2 | | | | Positioning behaviour<br>• Behaviour with deactivated positioning function (*bEnable* = FALSE). |
| | 0 | *dnOut_p* = 0<br>(Lenze setting) | | The *dnPosOut_p* output jumps to "0".<br>• The speed signal at the *nNOut_v* output maps this position step change as angle signal.<br>**Note:**<br>If the *nNOut_v* output has an impact on the setpoint generation for the drive, undefined motions (shocks, following errors) may occur. |
| | 1 | *dnOut_p/nNOut_v* follow *dnAct_p* | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnAct_p* input (actual position). |
| | 2 | *dnOut_p/nNOut_v* follow *dnSet_p* | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnSet_p* input (setpoint). |
| | 3 | *dnOut_p/nNOut_v* follow *dnAct_p* (without limitation) | | The outputs *dnPosOut_p* and *nNOut_v* follow the value at the *dnAct_p* input (actual position), but without limitation. |
| C01060/4 | 0.010 | s | 130.000 | Deceleration of set position<br>• Deceleration ramp for stop before reaching the setpoint position (*bExecute* = FALSE).<br>• Lenze setting: 1.000 s |
| C01060/5 | 0.010 | s | 130.000 | Acceleration ramp<br>• Lenze setting: 1.000 s |
| C01060/6 | 0.010 | s | 130.000 | Deceleration ramp<br>• Lenze setting: 1.000 s |
| C01061/3 | -15000 | rpm | 15000 | Forward motion<br>• Positive speed<br>• Lenze setting: 200 rpm |
| C01061/4 | -15000 | rpm | 15000 | Return motion<br>• Negative speed<br>• Lenze setting: 200 rpm |

### 18.1.140 L_PosiShaftCtrlInterface_1

FB in preparation!

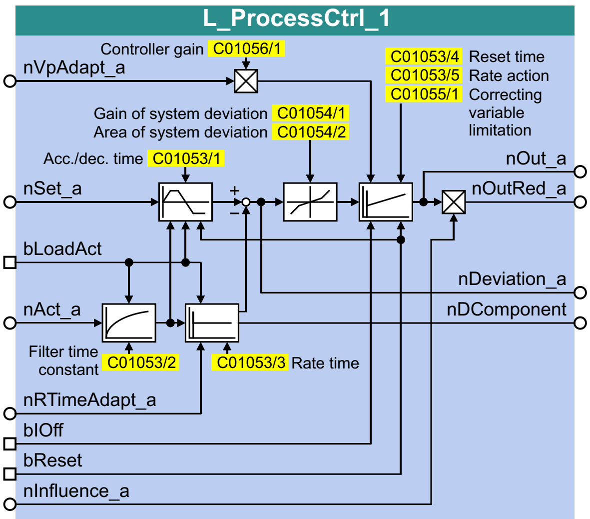


## 18.1.141 L_ProcessCtrl_1

This FB serves to implement a dancer position or tension control.

The FB is provides with the following functions:

- Adjustable control algorithm (P, PI, PID) with adaptable gain
- Reduced controller dynamics at low system deviation
- Setpoint ramp generator for preventing setpoint step-changes at the input
- Setpoint ramp generator can be loaded with actual value
- Low-pass filter and rate action in the actual value feedback
- Integral action component can be switched off
- Interruptible control




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nVpAdapt_a<br>INT | Proportional evaluation of the controller gain (Vp)<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to 0 ... 16384 (0 ... 100 %) | |
| nSet_a<br>INT | Controller setpoint | |
| bLoadAct<br>BOOL | Accept actual controller value | |
| | TRUE | The actual controller value *nAct_a* is taken over into the ramp generator, the low pass and the rate action. |
| nAct_a<br>INT | Actual controller value | |
| nRTimeAdapt_a<br>INT | Proportional evaluation of the rate time in the actual value path<br>• Scaling: 16384 ≡ 100 % rate time (C01053)<br>• Internal limitation to 0 ... 16384 (0 ... 100 %) | |
| bIOff<br>BOOL | Reset controller I component | |
| | TRUE | The controller I component is reset. |

| Identifier / Data type | | Information/possible settings |
|---|---|---|
| bReset / BOOL | Reset entire control | |
| | TRUE | All outputs are reset to 0. |
| nInfluence_a / INT | Proportional evaluation of the controller correcting variable *nOutRed_a*<br>• Scaling: 16384 ≡ 100 % controller correcting variable (*nOut_a*) | |

## Outputs

| Identifier / Data type | Value/meaning |
|---|---|
| nOut_a / INT | Controller correcting variable<br>• Internal limitation to ±16384 (±100 %) |
| nOutRed_a / INT | Controller correcting variable (evaluated by *nInfluence_a*) |
| nDeviation_a / INT | System deviation<br>• Internal limitation to ±32767 (±199.99 %) |
| nDComponent / INT | D component of the rate action<br>• Internal limitation to ±16384 (±100 %) |

## Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01053/1 | 0.000 | s | 30.000 | Acceleration/deceleration time of the setpoint ramp generator<br>• Lenze setting: 0.000 s |
| C01053/2 | 0.000 | s | 30.000 | Filter time constant for actual controller value<br>• Lenze setting: 0.000 s |
| C01053/3 | 0.000 | s | 30.000 | Rate time for actual controller value<br>• Lenze setting: 0.000 s |
| C01053/4 | 0.000 | s | 30.000 | Controller reset time<br>• Lenze setting: 1.000 s |
| C01053/5 | 0.000 | s | 30.000 | Controller rate action<br>• Lenze setting: 0.000 s |
| C01054/1 | 0.00 | % | 199.99 | Gain of the system deviation in the range of reduced sensitivity<br>• Lenze setting: 100.00 % |
| C01054/2 | 0.00 | % | 199.99 | Area of system deviation with reduced gain/sensitivity<br>• Lenze setting: 0.00 % |
| C01055/1 | | | | Limitation of the controller correcting variable to the positive area |
| | False | Limitation not active. (Lenze setting) | | |
| | True | Limitation active. | | |
| C01056/1 | 0.00 | | 100.00 | Controller gain<br>• Lenze setting: 0.10 |

### 18.1.141.1 Control characteristic

The dancer position or tension controller can be optionally operated as P, PI or PID controller. In the Lenze setting, the PI algorithm is active.

#### Gain (P component)

The input value is controlled by a linear characteristic. The slope of the characteristic is determined by the controller gain Vp.

The controller gain Vp is set in C01056/1.

- The controller gain can be adapted via the *nVpAdapt_a* input (also possible in online mode).
- The *nVpAdapt_a* input value has a direct effect on the controller gain:

$$P = \text{nVpAdapt\_a} \cdot \text{C01056/1}$$

Example: With the parameterised controller gain Vp = 2.0 and *nVpAdapt_a* = 75 %, the resulting gain factor is as follows:

$$P = \frac{75\ [\%]}{100\ [\%]} \cdot 2.0 = 1.5$$

#### Reset time Tn (I component)

The adjustment time Tn is set under C01053/4.

- The I component of the controller can be deactivated by setting the input *bIOff* to TRUE.
- The I component can be switched on and off online.

#### Differential component Kd (D component)

The differential component Kd is set under C01053/5.

- The setting "0.0 s" deactivates the D component (Lenze setting). In this way, the PID controller becomes a PI controller or P controller, if the I component has been deactivated as well.

#### Evaluation of the output signal (controller influence)

If the motor speed or motor torque is precontrolled, a low influence is sufficient for the controller to comply with the setpoint.

Use the *nInfluence_a* input to select the influencing factor the controller correcting variable (*nOut_a*) is to be evaluated with. The evaluated controller correcting variable is output at *nOutRed_a*.

**Note!**

The controller influence evaluates the output signal multiplicatively. A change of the *nInfluence_a* influencing factor also changes the dynamics of the controller!

### Switching on/off the controller

By setting the *bReset* input to TRUE, the process controller can be switched off.

### Loading the setpoint ramp generator with the actual value

If the actual value is loaded into the setpoint ramp generator in switched-off state, it has the advantage that in the moment of the renewed controller enable, the system deviation is zero first. Thus, compensation processes can be mostly prevented.

When the *bLoadAct* input is set to TRUE, the setpoint ramp generator can be loaded with the actual value. This keeps the system deviation equal to zero and I component of the controller.

### Low pass and rate action in the actual value path

In order to filter signal interferences more effectively, you can activate a low pass in the actual value path.

- The filter time constant for the low pass is set in C01053/2.
- The rate time constant in the feedback path can be set in C01053/3. This serves to compensate interfering decelerations.

### Reduced controller dynamics at low system deviation

Due to a reduced controller dynamics with low system deviations, the dampening behaviour of the control loop is mostly influenced favourably.

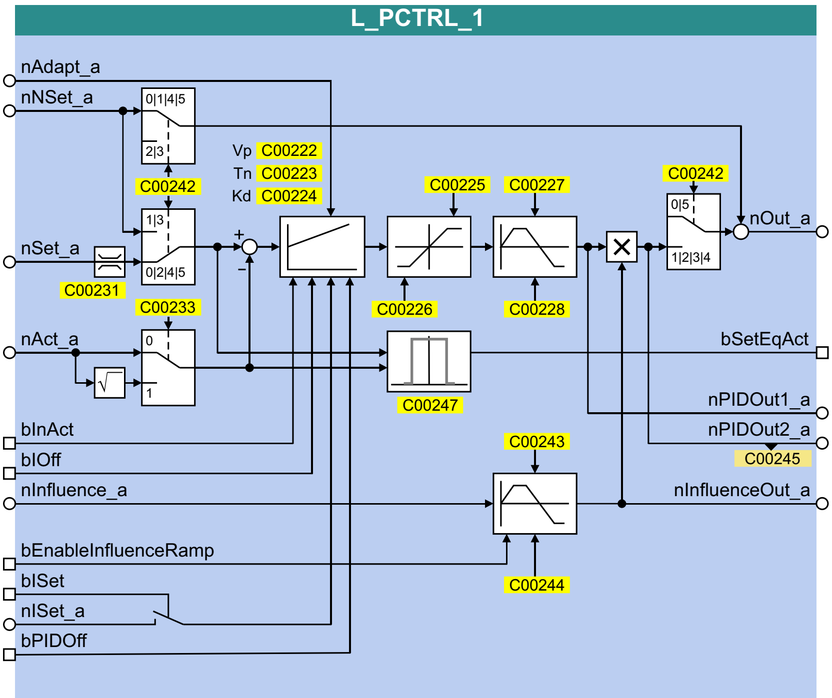
- C01054/2 serves to determine the tolerance zone in which the system deviation is transmitted to the controller with a slight gain.
- C01054/1 serves to determine the amount of percent the gain in the defined tolerance zone is to be reduced to.

## 18.1.142 L_PCTRL_1

This FB is a PID controller and can be used for various control tasks (e.g. as dancer position controller, tension controller, or pressure controller).

The FB is provides with the following functions:

- Adjustable control algorithm (P, PI, PID)
- Ramp function generator for preventing setpoint step-changes at the input
- Limitation of the controller output
- Factorisation of the output signal
- Vp adaptation
- Integral action component can be switched off
- Comparison function "Actual value = setpoint"




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nAdapt_a<br>INT | Adaptation of gain Vp set in C00222 in percent<br>• Internal limitation to ± 199.99 %<br>• Changes can be done online.<br>• Display parameter: C00830/62 |
| nNset_a<br>INT | Speed setpoint<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Display parameter: C00830/89 |

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nSet_a<br>INT | Sensor and process setpoint for operating modes 2, 4 and 5<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Display parameter: C00830/63 | |
| nAct_a<br>INT | Speed or actual sensor value (actual process value)<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Display parameter: C00830/61 | |
| bInAct<br>BOOL | Deactivate process controller temporarily (stop)<br>• Changes can be done online.<br>• Display parameter: C00833/76<br>Note: This input is not interconnected in the LA_NCtrl application block. | |
| | TRUE | • The current output value is frozen.<br>• The internal control algorithm is stopped.<br>• However, a setpoint selected via input *nNSet_a* is still provided in operating modes 0/1/4/5. |
| bIOff<br>BOOL | Switch off I-component of process controller<br>• Changes can be done online.<br>• Display parameter: C00833/77 | |
| | TRUE | The I component of the process controller is set to zero. |
| nInfluence_a<br>INT | Limitation of the influencing factor in percent<br>• *nInfluence_a* serves to limit the influencing factor of the PID controller contained in the FB to a required value (- 199.99 % ... + 199.99 %).<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 %<br>• Display parameter: C00830/64 | |
| bEnableInfluenceRamp<br>BOOL | Activate ramp for influencing factor<br>• Display parameter: C00833/106 | |
| | TRUE | Influencing factor of the PID controller is ramped up to the *nInfluence_a* value. |
| | FALSE | Influencing factor of the PID controller is ramped down to "0". |
| bISet<br>BOOL | Accept I component *nISet_a* in PID controller | |
| | TRUE | The value at the input *nISet_a* is accepted in the PID controller. |
| nISet_a<br>INT | Selection of I component of PID controller<br>• With a TRUE signal at *bISet*, the assigned value is accepted in the PID controller.<br>• Scaling: 16384 ≡ 100 %<br>• Internal limitation to ± 199.99 % | |
| bPIDOff<br>BOOL | Reset the entire PID controller | |
| | TRUE | • The I component of the controller is set to zero.<br>• The controller output is set to zero.<br>• The internal control algorithm is stopped. |

## Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±32767 (±199.99 %)<br>• Scaling: 16384 ≡ 100 % | |
| bSetEqAct<br>INT | Status output "Setpoint and actual value are identical" | |
| | TRUE | Setpoint and actual value are identical, i.e. no system deviation available. |

| Identifier<br>Data type | Value/meaning |
|---|---|
| nPIDOut1_a<br>INT | PID controller output without influencing factor *nInfluence_a*<br>• Inputs *bEnableInfluenceRamp* and *nInfluence_a* do not have any effect here, the limited PID output value influenced by the internal ramp times is output.<br>• There is no connection with the additive input *nNSet_a*.<br>• Scaling: 16384 ≡ 100 % |
| nPIDOut2_a<br>INT | PID controller output with influencing factor *nInfluence_a*.<br>• There is no connection with the additive input *nNSet_a*.<br>• Scaling: 16384 ≡ 100 %<br>• Display parameter: C00245 |
| nInfluenceOut_a<br>INT | Current influencing factor ("ramp status") on the PID output value<br>• Scaling: 16384 ≡ 100 % |

## Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00222 | 0.1 | 0.1 | 500.0 | Gain Vp<br>• Lenze setting: 1.0 |
| C00223 | 20 | ms | 6000 | Reset time Tn<br>• Lenze setting: 400 ms |
| C00224 | 0.0 | 0.1 | 5.0 | Differential component Kd<br>• Lenze setting: 0.0 |
| C00225 | -199.99 | % | +199.99 | MaxLimit<br>• Maximum value of the PID operating range<br>• Lenze setting: 199.99 % |
| C00226 | -199.99 | % | +199.99 | MinLimit<br>• Minimum value of the PID operating range<br>• Lenze setting: -199.99 % |
| C00227 | 0.000 | s | 999.999 | Acceleration time for the ramp at the PID output (should be set as steep as possible)<br>• Lenze setting: 0.010 s |
| C00228 | 0.000 | s | 999.999 | Deceleration time for the ramp at the PID output<br>• Lenze setting: 0.010 s |
| C00231/1 (Pos. Maximum)<br>C00231/2 (Pos. Minimum)<br>C00231/3 (Neg. Minimum)<br>C00231/4 (Neg. Maximum) | 0.00 | % | 199.99 | Operating range<br>• Determination of the operating range for the PID process controller by limiting the input signal *nSet_a*.<br>• Lenze setting: No limitation (-199.99 % ... +199.99 %) |
| C00233 | | | | Root function<br>• Lenze setting: "0: Off" |
| | 0 | Off | | The actual value at *nAct_a* is not changed for further processing. |
| | 1 | On | | The square root of the actual value at *nAct_a* is taken for further processing. |

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00242 | | | | Operating mode<br>• Lenze setting: "0: Off" |
| | 0 | Off | | The input setpoint *nNSet_a* is output without any changes at the output *nOut_a*. |
| | 1 | nNSet + nNSet_PID | | *nNSet_a* and *nAct_a* are used as PID input values. The arriving *nNSet_a* is additively linked to the value output by the PID element. |
| | 2 | nSet_PID | | *nSet_a* and *nAct_a* are used as PID input values. The input *nNSet_a* is not considered. |
| | 3 | nNSet_PID | | *nNSet_a* and *nAct_a* are used as PID input values. The input *nSet_a* is not considered. |
| | 4 | nNSet + nSet_PID | | *nSet_a* and *nAct_a* are used as PID input values. The arriving *nNSet_a* setpoint is additively linked to the value output by the PID element. |
| | 5 | nNSet \|\| nSet_PID | | *nSet_a* and *nAct_a* are used as PID input values. The setpoint *nNSet_a* is output at the output *nOut_a* . The PID output value is output at the output *nPIDOut_a*. |
| C00243 | 0.000 | s | 999.999 | Influence acceleration time<br>• Acceleration time $T_{ir}$ for the influencing factor.<br>• Lenze setting: 5.000 s |
| C00244 | 0.000 | s | 999.999 | Influence deceleration time<br>• Deceleration time $T_{if}$ for the influencing factor.<br>• Lenze setting: 5.000 s |
| C00245 | -199.99 | % | +199.99 | Display of PID output value *nPIDOut_a* |
| C00247 | 0 | % | 100 | Window for comparison function "Actual value = setpoint"<br>• Lenze setting: 2 %<br>• Hysteresis: 1 % (fixed) |

### 18.1.142.1 Control characteristic

The PI algorithm is active in the Lenze setting.

#### Gain (P component)

The input value is controlled by a linear characteristic. The slope of the characteristic is determined by the controller gain Vp.

The controller gain Vp is set under C00222.

- The controller gain can be adapted via the input *nAdapt_a* (also possible in online mode).
- The input value *nAdapt_a* has a direct effect on the controller gain:

$$P = \text{nAdapt\_a} \cdot \text{C00222}$$

Example: With the parameterised controller gain Vp = 2.0 and *nAdapt_a* = 75 %, the resulting gain factor is as follows:

$$P = \frac{75\,[\%]}{100\,[\%]} \cdot 2.0 = 1.5$$

#### Integral action component (I component)

The I component can be selected via the input *nISet_a*. With a TRUE signal at *bISet*, the assigned value is accepted in the PID controller.

- Setting the adjustment time Tn to the maximum value of "6000 ms" deactivates the I component.
- The I component of the controller can also be deactivated by setting the input *bIOff* to TRUE.
- The I component can be switched on and off online.

#### Adjustment time

The adjustment time Tn is set under C00223.

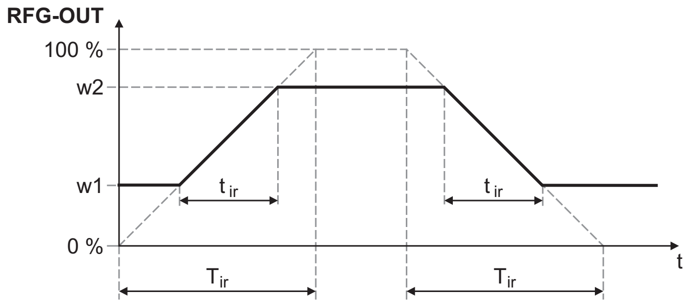
#### Differential component Kd (D component)

The differential component Kd is set under C00224.

- The setting "0.0 s" deactivates the D component (Lenze setting). In this way, the PID controller becomes a PI controller or P controller, if the I component has been deactivated as well.

### 18.1.142.2 Ramp function generator

The PID output is led via a ramp function generator with linear characteristic. This serves to transfer setpoint step-changes at the PID output into a ramp which should be as steep as possible.




w1, w2 = change of the main setpoint as a function of $t_{ir}$ and $t_{if}$
RFG-OUT = output of the ramp function generator

[18-55] Acceleration and deceleration times

- $t_{ir}$ and $t_{if}$ are the desired times for changing between w1 and w2.
- The ramps for acceleration and deceleration can be set individually.
  - C00227: Acceleration time $t_{ir}$
  - C00228: Deceleration time $t_{if}$
- The $t_{ir}$/$t_{if}$ values are converted into the required Ti times according to the following formula:

$$T_{ir} = t_{ir} \cdot \frac{100\ \%}{w2 - w1} \qquad T_{if} = t_{if} \cdot \frac{100\ \%}{w2 - w1}$$

- The ramp function generator is immediately set to "0" by setting *bInAct* to TRUE.

### 18.1.142.3 Operating range of the PID process controller

The value range of the input signal *nSet_a* and thus the operating range of the PID process controller can be limited with the following parameters:

- C00231/1: Pos. maximum (default setting: 199.99 %)
- C00231/2: Pos. minimum (default setting: 0.00 %)
- C00231/3: Neg. minimum (default setting: 0.00 %)
- C00231/4: Neg. maximum (default setting: 199.99 %)

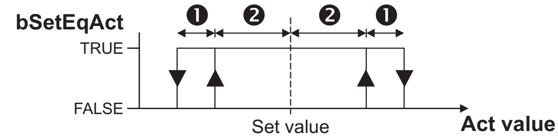
### 18.1.142.4 Evaluation of the output signal

After the limitation, the output signal is evaluated with the influencing factor *nInfluence_a*. The evaluation is activated/suppressed along a ramp when the *bEnableInfluenceRamp* input is set to TRUE. The ramp times are set with the parameters "Influence acceleration time" (C00243) and "Influence deceleration time" (C00244).

### 18.1.142.5 Comparison function "Actual value = setpoint"

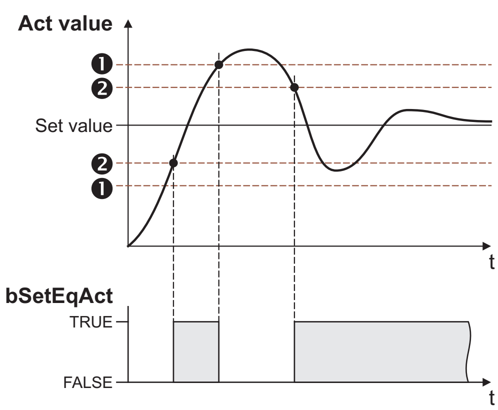
If setpoint and actual value are identical and there is no system deviation, the *bSetEqAct* status output is set to TRUE.

- The hysteresis of the comparison function has a fixed value of 1 %.
- From V10.00.00 the symmetrical window around the setpoint for the comparison function can be set in C00247 (Lenze setting: 2 %).




❶ Hysteresis (1 %, fixed value)
❷ C00247: Window

[18-56] Comparison function: Switching performance




❶ Hysteresis (1 %, fixed value)
❷ C00247: Window

[18-57] Comparison function: Example

### 18.1.142.6 Control functions

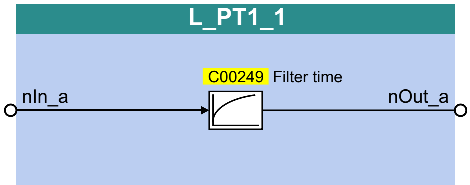
The process controller has various digital inputs for controlling the FB:

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bInAct<br>BOOL | Deactivate process controller temporarily (stop)<br>• Changes can be done online.<br>• Display parameter: C00833/76<br>Note: This input is not interconnected in the LA_NCtrl application block. | |
| | TRUE | • The current output value is frozen.<br>• The internal control algorithm is stopped.<br>• However, a setpoint selected via input *nNSet_a* is still provided in operating modes 0/1/4/5. |
| bIOff<br>BOOL | Switch off I-component of process controller<br>• Changes can be done online.<br>• Display parameter: C00833/77 | |
| | TRUE | The I component of the process controller is set to zero. |
| bPIDOff<br>BOOL | Reset the entire PID controller | |
| | TRUE | • The I component of the controller is set to zero.<br>• The controller output is set to zero.<br>• The internal control algorithm is stopped. |

### 18.1.143 L_PT1_1

This FB filters and delays analog signals.

- The filter time constant T can be set under C00249.
- The gain is defined with $V_p$ = 1.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

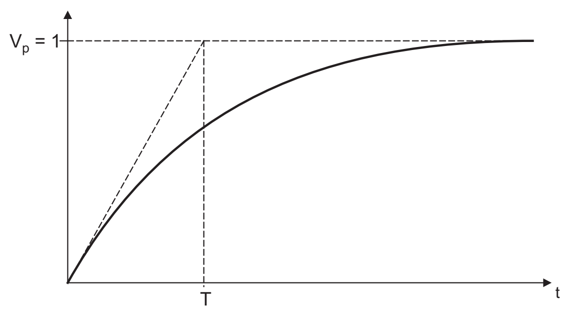
#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00249 | 0 | ms | 5000 | Filter time constant<br>• The filter is not active with a setting of "0 ms". The input signal is passed through one-to-one to the output.<br>• Lenze setting: 2000 ms |

#### Function

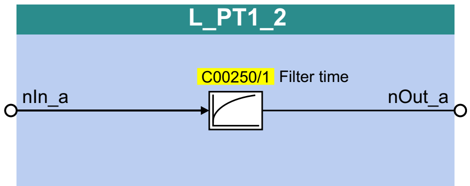



[18-58] Filter time constant T of the first-order delay element

### 18.1.144 L_PT1_2

This FB filters and delays analog signals.

- Filter time constant T can be set in C00250/1.
- The gain is defined with $V_p = 1$.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

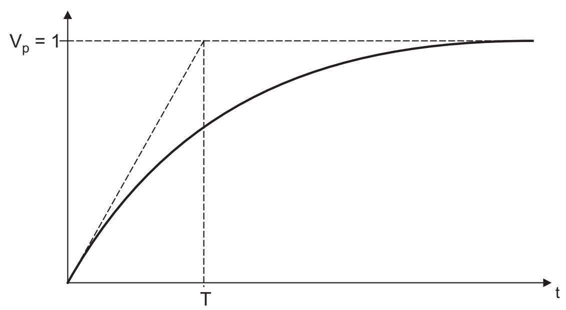
#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00250/1 | 0 | ms | 5000 | Filter time constant<br>• The filter is not active with a setting of "0 ms". The input signal is passed through one-to-one to the output.<br>• Lenze setting: 2000 ms |

#### Function

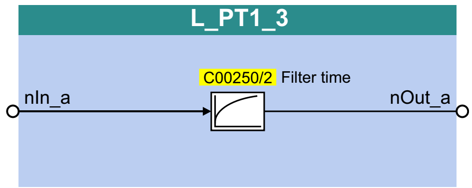



[18-59] Filter time constant T of the first-order delay element

## 18.1.145 L_PT1_3

This FB filters and delays analog signals.

- Filter time constant T can be set in C00250/2.
- The gain is defined with $V_p = 1$.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| nIn_a<br>INT | Input signal |

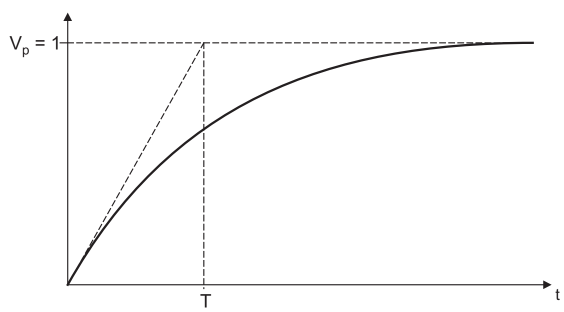
### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00250/2 | 0 | ms | 5000 | Filter time constant<br>• The filter is not active with a setting of "0 ms". The input signal is passed through one-to-one to the output.<br>• Lenze setting: 2000 ms |

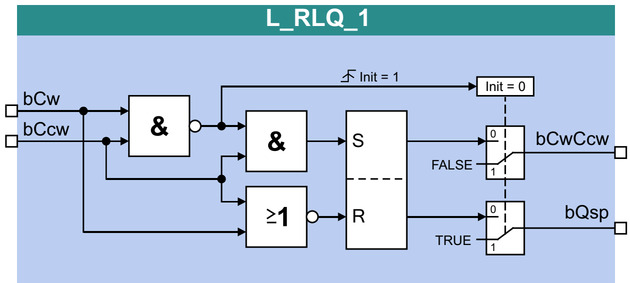
### Function




[18-60] Filter time constant T of the first-order delay element

## 18.1.146 L_RLQ_1

This FB links a selected direction of rotation to the quick stop function with wire-break protection.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bCw<br>BOOL | Input<br>• TRUE = CW rotation |
| bCCw<br>BOOL | Input<br>• TRUE = CCW rotation |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bQSP<br>BOOL | Output signal for quick stop (QSP) |
| bCwCcw<br>BOOL | Output signal for CW/CCW rotation<br>• TRUE = CCW rotation |

### Function

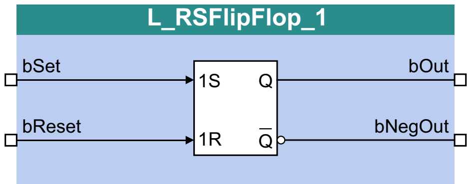
| Inputs | | Outputs | | Notes |
|---|---|---|---|---|
| **bCw** | **bCCw** | **bCwCcw** | **bQSP** | |
| TRUE | TRUE | FALSE | TRUE | The inputs have this status only if a TRUE signal is being applied to both inputs at the moment of switch-on!<br>See also FB illustration above, "Init" = 1. |
| If *one* of the inputs has the TRUE status, the following truth table applies: | | | | |
| FALSE | FALSE | FALSE | TRUE | See also FB illustration above, "Init" = 0. |
| TRUE | FALSE | FALSE | FALSE | |
| FALSE | TRUE | TRUE | FALSE | |
| TRUE | TRUE | X (save) | | |

[18-61] Truth table of the FB L_RLQ, 0 = FALSE, 1 = TRUE

## 18.1.147 L_RSFlipFlop_1

The functionality of this FB corresponds to that of an RS flipflop:

- An input signal at *bSet* serves to set the *bOut* output permanently.
- An input signal at *bReset* serves to reset the output.
- Additional, inverted output *bNegOut*




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bSet<br>BOOL | Setting input | |
| | TRUE | • The *bOut* output is set to TRUE.<br>• The *bNegOut* output is set to FALSE. |
| bReset<br>BOOL | Reset input | |
| | TRUE | • The *bOut* output is set to FALSE.<br>• The *bNegOut* output is set to TRUE. |

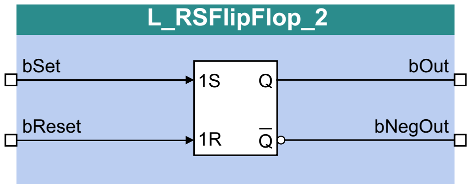
### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |
| bNegOut<br>BOOL | Output signal, inverted |

### 18.1.148 L_RSFlipFlop_2

The functionality of this FB corresponds to that of an RS flipflop:

- An input signal at *bSet* serves to set the *bOut* output permanently.
- An input signal at *bReset* serves to reset the output.
- Additional, inverted output *bNegOut*




#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bSet<br>BOOL | Setting input | |
| | TRUE | • The *bOut* output is set to TRUE.<br>• The *bNegOut* output is set to FALSE. |
| bReset<br>BOOL | Reset input | |
| | TRUE | • The *bOut* output is set to FALSE.<br>• The *bNegOut* output is set to TRUE. |

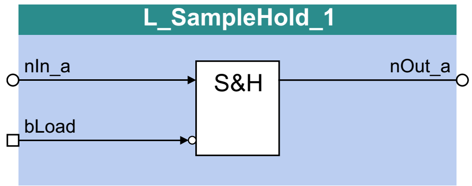
#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output signal |
| bNegOut<br>BOOL | Output signal, inverted |

## 18.1.149 L_SampleHold_1

This FB can store a signal.

- The saved value is also available after mains switching.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_a<br>INT | Input signal | |
| bLoad<br>BOOL | Save input signal | |
| | FALSE | The last-valid value at *nIn* is saved and output to *nOut*. A signal change at *nIn* does not cause a change at *nOut*. |
| | TRUE | The *nOut* output provides *dnIn*. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

### Function

- When *bLoad* = TRUE, the *nIn_a* signal is switched to *nOut_a*.
- When *bLoad* = FALSE, the last-valid value is saved and output at *nOut_a*. A signal change at *nIn_a* does not cause a change at *nOut_a*.

### Behaviour after mains switching

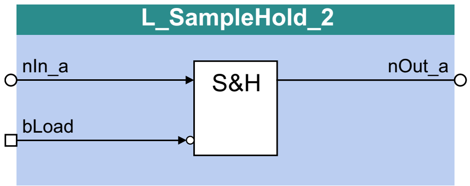
The last-loaded value is permanently stored after switching of the supply voltage and reloaded after restart.

- In order that the saved value is not immediately overwritten with the current input signal at *nIn* after restart, *bLoad* must be set to FALSE at restart.

## 18.1.150 L_SampleHold_2

This FB can store a signal.

- The saved value is also available after mains switching.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_a<br>INT | Input signal | |
| bLoad<br>BOOL | Save input signal | |
| | FALSE | The last-valid value at *nIn* is saved and output to *nOut*. A signal change at *nIn* does not cause a change at *nOut*. |
| | TRUE | The *nOut* output provides *dnIn*. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut<br>INT | Output signal |

### Function

- When *bLoad* = TRUE, the *nIn_a* signal is switched to *nOut_a*.
- When *bLoad* = FALSE, the last-valid value is saved and output at *nOut_a*. A signal change at *nIn_a* does not cause a change at *nOut_a*.

### Behaviour after mains switching

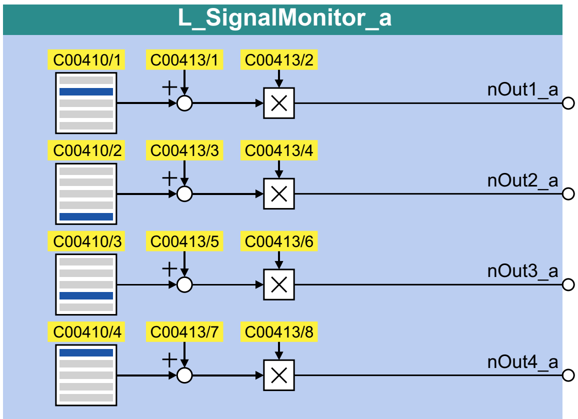
The last-loaded value is permanently stored after switching of the supply voltage and reloaded after restart.

- In order that the saved value is not immediately overwritten with the current input signal at *nIn* after restart, *bLoad* must be set to FALSE at restart.

## 18.1.151 L_SignalMonitor_a

This FB outputs four analog signals which can be selected from a list of analog output signals of all function blocks provided in the device.

- Offset and gain of the source signals are adjustable.




### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut1_a<br>INT | Output signal<br>• Internal limitation to ±32767 |
| nOut2_a<br>INT | Output signal<br>• Internal limitation to ±32767 |
| nOut3_a<br>INT | Output signal<br>• Internal limitation to ±32767 |
| nOut4_a<br>INT | Output signal<br>• Internal limitation to ±32767 |

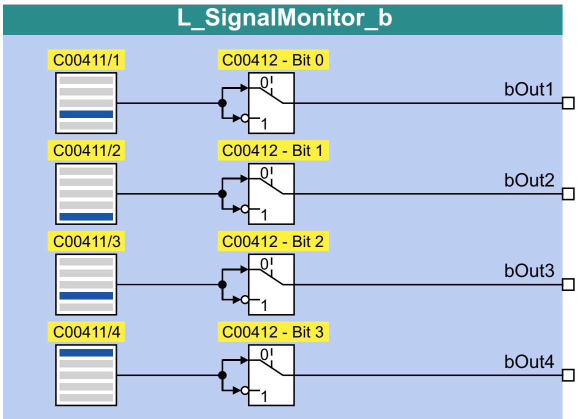
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00410/1<br>...<br>C00410/4 | | | | Selection of the signal sources for *nOut1_a ... nOut4_a* |
| | 0 | Not connected | | |
| | 1000 | LA_nCtrl_wDriveControlStatus | | |
| | 1001 | LA_nCtrl_wFailNumber | | |
| | 1002 | LA_nCtrl_nMotorCurrent_a | | |
| | ... | ... | | |
| | 42017 | LA_nCtrl_In_nPIDSerValue_a | | |
| C00413/1<br>C00413/3<br>C00413/5<br>C00413/7 | -199.99 | % | +199.99 | Offset |
| C00413/2<br>C00413/4<br>C00413/6<br>C00413/8 | -199.99 | % | +199.99 | Gain |

## 18.1.152 L_SignalMonitor_b

This FB outputs four binary signals which can be selected from a list of binary output signals of all function blocks provided in the device.

- Inversion of the output signals can be set.




### Outputs

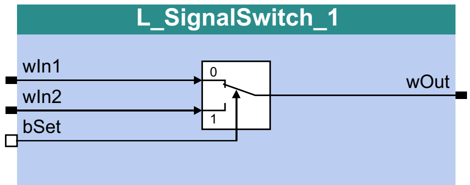
| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut1<br>...<br>bOut4<br>BOOL | Output signal<br>FALSE / TRUE |

### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00411/1<br>...<br>C00411/4 | | | Selection of the signal sources for *bOut1 ... bOut4* |
| | 0 | Not connected | |
| | 1000 | LA_nCtrl_bDriveReady | |
| | 1001 | LA_nCtrl_bDriveFail | |
| | 1002 | LA_nCtrl_bCInhActive | |
| | .. | ... | |
| | 42033 | LA_NCtrl_bPIDIOff | |
| C00412 | | | Inversion<br>• Bit set = inversion active |
| | Bit 0 | bOut1 inverted | |
| | Bit 1 | bOut2 inverted | |
| | Bit 2 | bOut3 inverted | |
| | Bit 3 | bOut4 inverted | |
| | Bit 4 | Reserved | |
| | Bit 5 | Reserved | |
| | Bit 6 | Reserved | |
| | Bit 7 | Reserved | |

## 18.1.153 L_SignalSwitch_1

This FB switches between two input signals of the "WORD" data type. The switch-over is controlled by means of a boolean input signal.




### Inputs

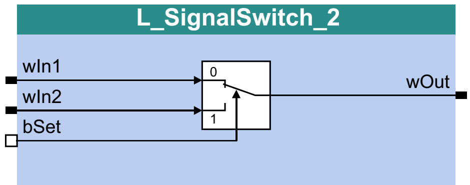
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| wIn1<br>WORD | Input signal 1 | |
| wIn2<br>WORD | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *wOut* | |
| | FALSE | *wIn1* |
| | TRUE | *wIn2* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

## 18.1.154 L_SignalSwitch_2

This FB switches between two input signals of the "WORD" data type. The switch-over is controlled by means of a boolean input signal.




### Inputs

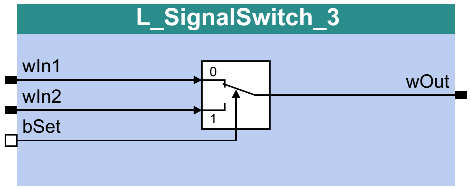
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| wIn1<br>WORD | Input signal 1 | |
| wIn2<br>WORD | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *wOut* | |
| | FALSE | *wIn1* |
| | TRUE | *wIn2* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

## 18.1.155 L_SignalSwitch_3

This FB switches between two input signals of the "WORD" data type. The switch-over is controlled by means of a boolean input signal.




### Inputs

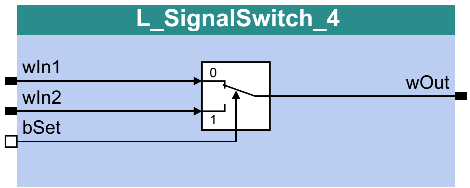
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| wIn1<br>WORD | Input signal 1 | |
| wIn2<br>WORD | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *wOut* | |
| | FALSE | *wIn1* |
| | TRUE | *wIn2* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

## 18.1.156 L_SignalSwitch_4

This FB switches between two input signals of the "WORD" data type. The switch-over is controlled by means of a boolean input signal.




### Inputs

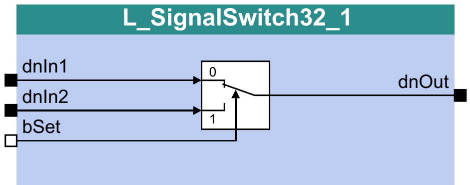
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| wIn1<br>WORD | Input signal 1 | |
| wIn2<br>WORD | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *wOut* | |
| | FALSE | *wIn1* |
| | TRUE | *wIn2* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut<br>WORD | Output signal |

## 18.1.157 L_SignalSwitch32_1

This FB switches between two input signals of the "DINT" data type. The switch-over is controlled by means of a boolean input signal.




### Inputs

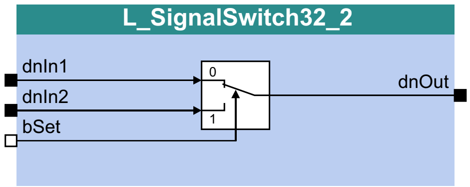
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn1<br>DINT | Input signal 1 | |
| dnIn2<br>DINT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *dnOut* | |
| | FALSE | *dnIn1* |
| | TRUE | *dnIn2* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut<br>DINT | Output signal |

### 18.1.158 L_SignalSwitch32_2

This FB switches between two input signals of the "DINT" data type. The switch-over is controlled by means of a boolean input signal.




#### Inputs

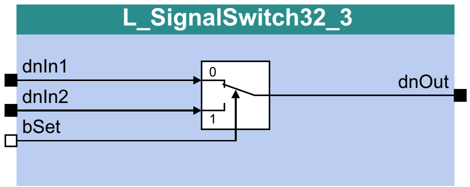
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn1<br>DINT | Input signal 1 | |
| dnIn2<br>DINT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *dnOut* | |
| | FALSE | *dnIn1* |
| | TRUE | *dnIn2* |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut<br>DINT | Output signal |

## 18.1.159 L_SignalSwitch32_3

This FB switches between two input signals of the "DINT" data type. The switch-over is controlled by means of a boolean input signal.




### Inputs

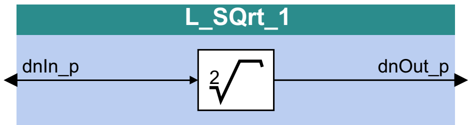
| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| dnIn1<br>DINT | Input signal 1 | |
| dnIn2<br>DINT | Input signal 2 | |
| bSet<br>BOOL | Selection of the input signal for the output at *dnOut* | |
| | FALSE | *dnIn1* |
| | TRUE | *dnIn2* |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut<br>DINT | Output signal |

### 18.1.160 L_SQrt_1

This FB outputs the square root for the DINT input value.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| dnIn_p<br>DINT | Input signal |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| dnOut_p<br>DINT | Output signal |

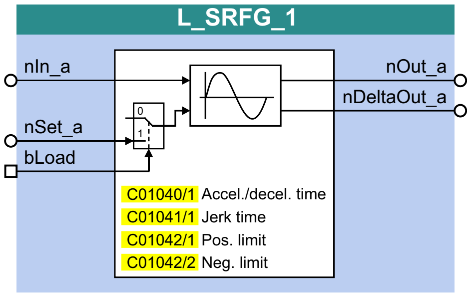
#### Function

$$dnOut\_p = \sqrt{dnIn\_p}$$

### 18.1.161 L_SRFG_1

This FB is a ramp function generator with S-shaped ramps for limiting the temporal rise of analog signals. The ramps are S-shaped due to trapezoidal acceleration.

- The ramp function generator is provided with a setting function so that a value can directly be loaded into the internal ramp generator.
- The balanced acceleration/deceleration time is set in C01040.
- An S-ramp time can be set in C01041 for jerk-free acceleration until maximum acceleration is reached.
- At the *nDeltaOut_a* output, the dy/dt slope of the *nOut_a* output signal is output.




#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_a<br>INT | Input signal | |
| nSet_a<br>INT | Starting value for the ramp function generator<br>• Will be accepted if *bLoad* = TRUE | |
| bLoad<br>BOOL | Initialise ramp function generator | |
| | FALSE | With the set acceleration/deceleration time, the ramp function generator switches over from the value loaded via *nSet_a* to the value at *nIn_a*. |
| | TRUE | At the *nOut_a* output, *nSet_a* is output.<br>• *nDeltaOut_a* remains at 0 %. |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |
| nDeltaOut_a<br>INT | Acceleration of the ramp function generator<br>• Internal limitation to ±100 % (100 % ≡ 16384) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01040/1 | 0.001 | s | 999.999 | Acceleration/Deceleration time<br>• Initialisation: 100.000 s |
| C01041/1 | 0.001 | s | 50.000 | S-ramp time<br>• Initialisation: 0.200 s |
| C01042/1 | -199.99 | s | 199.99 | Pos. limit<br>• Initialisation: 100.00 % |
| C01042/2 | -199.99 | s | 199.99 | Neg. limit<br>• Initialisation: -100.00 % |

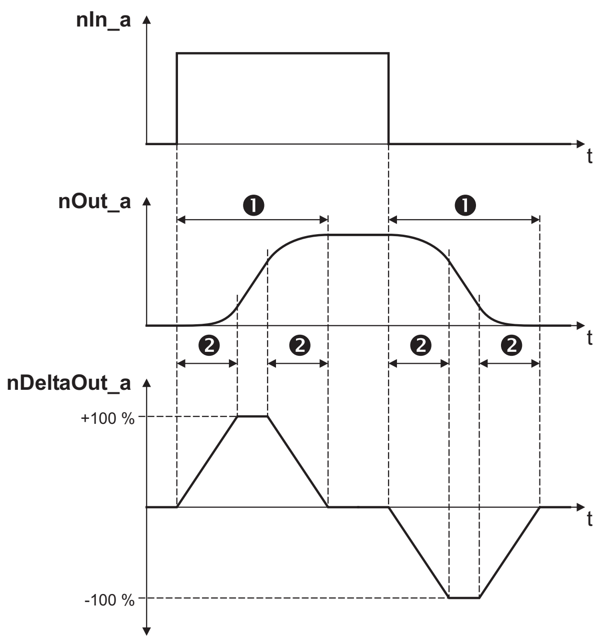
### Loading the ramp function generator

When *bLoad* is set to TRUE, the ramp function generator is loaded with the signal at *nSet_a*. This value is accepted immediately and output at *nOut_a*. There is no acceleration or deceleration via an S shape. As long as *bLoad* = TRUE, the ramp function generator remains inhibited.

### Acceleration/Deceleration time and S-ramp time

The acceleration/deceleration time and the S-ramp time for jerk-free acceleration can be set separately.

- Acceleration/Deceleration time = Time until the *nOut_a* output value has reached the *nIn_a* input value.
- S-ramp time = Time until the ramp function generator operates at maximum acceleration.



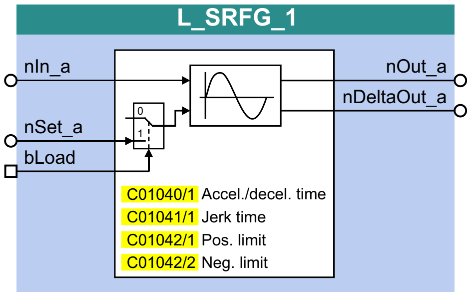

❶ C01040/x: Acceleration/Deceleration time
❷ C01041/x: S-ramp time

[18-62] Signal flow

## 18.1.162 L_SRFG_2

This FB is a ramp function generator with S-shaped ramps for limiting the temporal rise of analog signals. The ramps are S-shaped due to trapezoidal acceleration.

- The ramp function generator is provided with a setting function so that a value can directly be loaded into the internal ramp generator.
- The balanced acceleration/deceleration time is set in C01040.
- An S-ramp time can be set in C01041 for jerk-free acceleration until maximum acceleration is reached.
- At the *nDeltaOut_a* output, the dy/dt slope of the *nOut_a* output signal is output.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| nIn_a<br>INT | Input signal | |
| nSet_a<br>INT | Starting value for the ramp function generator<br>• Will be accepted if *bLoad* = TRUE | |
| bLoad<br>BOOL | Initialise ramp function generator | |
| | FALSE | With the set acceleration/deceleration time, the ramp function generator switches over from the value loaded via *nSet_a* to the value at *nIn_a*. |
| | TRUE | At the *nOut_a* output, *nSet_a* is output.<br>• *nDeltaOut_a* remains at 0 %. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| nOut_a<br>INT | Output signal<br>• Internal limitation to ±199 % (100 % ≡ 16384) |
| nDeltaOut_a<br>INT | Acceleration of the ramp function generator<br>• Internal limitation to ±100 % (100 % ≡ 16384) |

### Parameter

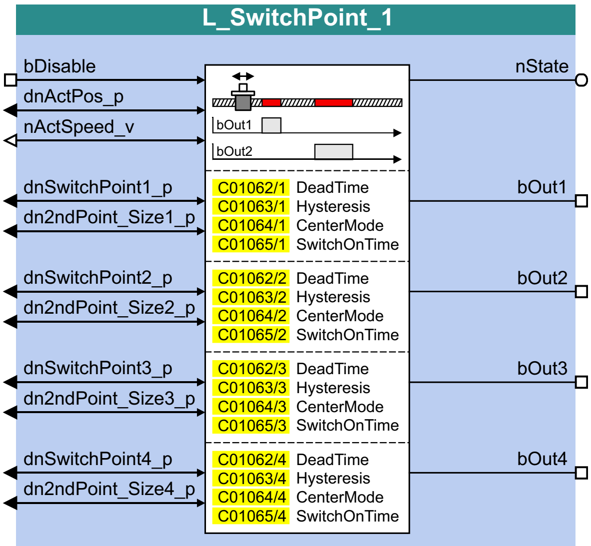
| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01040/2 | 0.001 | s | 999.999 | Acceleration/Deceleration time<br>• Initialisation: 100.000 s |
| C01041/2 | 0.001 | s | 50.000 | S-ramp time<br>• Initialisation: 0.200 s |
| C01042/3 | -199.99 | s | 199.99 | Pos. limit<br>• Initialisation: 100.00 % |
| C01042/4 | -199.99 | s | 199.99 | Neg. limit<br>• Initialisation: -100.00 % |

For a detailed functional description see L_SRFG_1.

## 18.1.163 L_SwitchPoint_1

This FB provides four position switch points, i.e. digital switches the binary statuses (FALSE/TRUE) of which depend on the actual position.

- A position switch point serves to start peripherals as paint nozzles or knifes depending on the tool position.
- Moreover the FB supports the compensation of delay times of external switching elements (dead time compensation).
- By setting a running time, position/time-based cams can be realised as well.




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bDisable<br>BOOL | Deactivate position switch points<br>• This input has the highest priority. | |
| | TRUE | Position switch points are deactivated.<br>• Outputs *bOut1* ... *bOut4* = FALSE |
| dnActPos_p<br>DINT | Actual position in [increments] | |
| nActSpeed_v<br>INT | Actual speed in [increments/ms]<br>• Scaling: 16384 ≡ 15000 rpm | |
| dnSwitchPoint**x**_p<br>DINT | Position switch point **x**: Position of the first switching point in [increments] | |
| dn2ndPoint_Size**x**_p<br>DINT | Position switch point **x**: Position of the second switching point or size of the switching window depending on the CenterMode set in C01064/x: | |
| | When CenterMode = FALSE:<br>Position of the second switching point in [increments]<br>• *dn2ndPoint_Size_p* must be higher than *dnSwitchPoint_p*. | |
| | When CenterMode = TRUE:<br>Size of the switching window in [increments]<br>• Due to the symmetrical arrangement of the window around the first switching point, uneven values are rounded to even values.<br>• Only positive values are permissible. | |

## Outputs

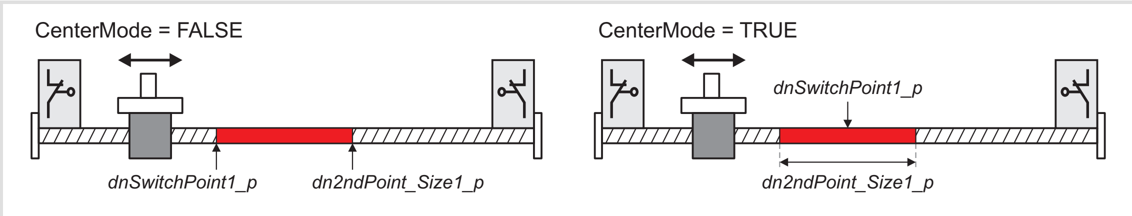
| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bOut1 ... bOut4<br>BOOL | Switching output 1 ... 4 | |
| | TRUE | The actual position is inside the defined switching window. |
| nState<br>INT | Status | |
| | 1 | FB is not active |
| | 2 | OK |
| | 3 | The data form resulting from the switching points are not plausible (*dnSwitchPoint_p*, *dn2ndPoint_Size_p*, parameterised hysteresis). |

## Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01062/1...4 | 0.00 | ms | 655.00 | Dead time for dead time compensation<br>• The resulting switching position is not subject to a plausibility check.<br>• Lenze setting: 0.00 ms |
| C01063/1...4 | 0 | Incr. | 65535 | Switching hysteresis<br>• Lenze setting: 0 incr. |
| C01064/1...4 | | | | CenterMode<br>• Definition how the *dn2ndPoint_Size_p* selection is interpreted. |
| | False | *dn2ndPoint_Size_p* defines the second switching point.<br>(Lenze setting) | | |
| | True | *dn2ndPoint_Size_p* defines the size of the switching window. | | |
| C01065/1...4 | 0 | ms | 60000 | Running time for position/time-based cams<br>• Lenze setting: 0 ms |
| | 0 ms = position-based cam | | | |

### 18.1.163.1 Definition of the switching range

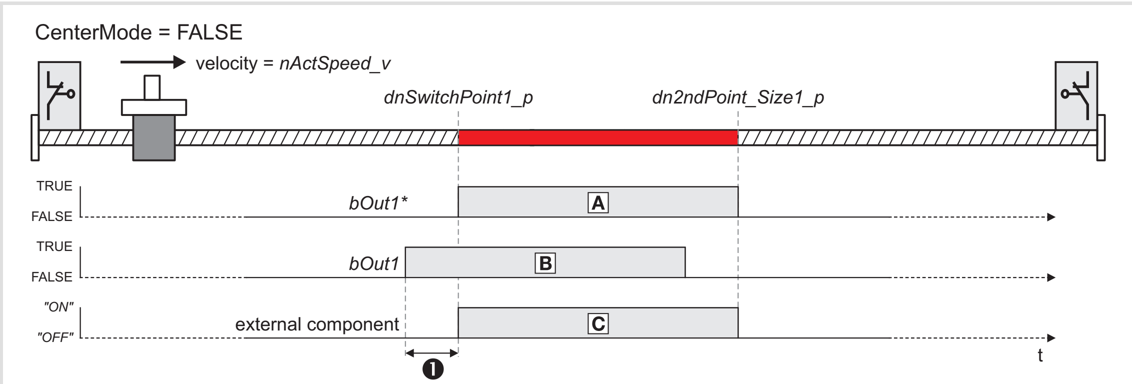
Depending on the setting of the CenterMode (C01064/x), the switching range can be defined via a start and end position or via the data of the center point/size of switching range:




[18-63] Definition of the switching range

### 18.1.163.2 Dead time compensation

The delay time of external switching elements can be compensated by setting the corresponding delay time in C01062/x. From this input and the current speed ,the FB calculates a preliminary stop, i.e. the *bOut* switching output is set correspondingly earlier.

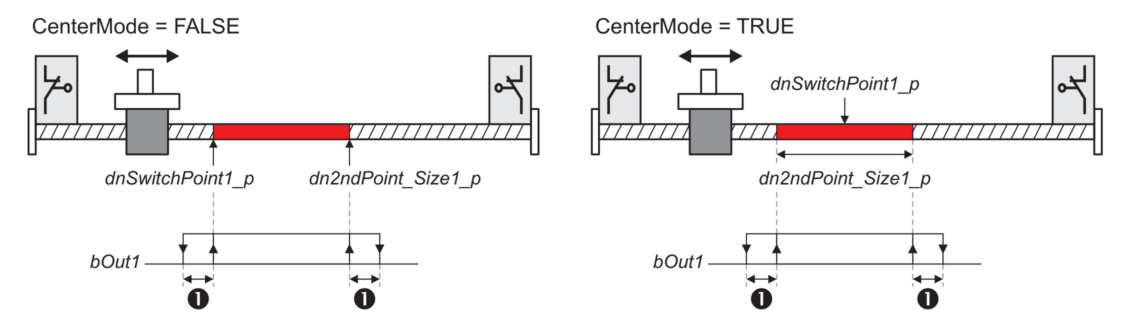



A Switching performance without dead time compensation
B Switching performance with dead time compensation
C Switching performance of the external switching element (with dead time)
❶ C01062/1: Dead time

[18-64] Dead time compensation

### 18.1.163.3 Switching hysteresis

Setting a hysteresis in C01063/x serves to prevent a permanent state change of the *bOut* switching output that may be caused by actual position value changes due to a mechanical irregularity at the axis.




❶ C01063/1: Hysteresis

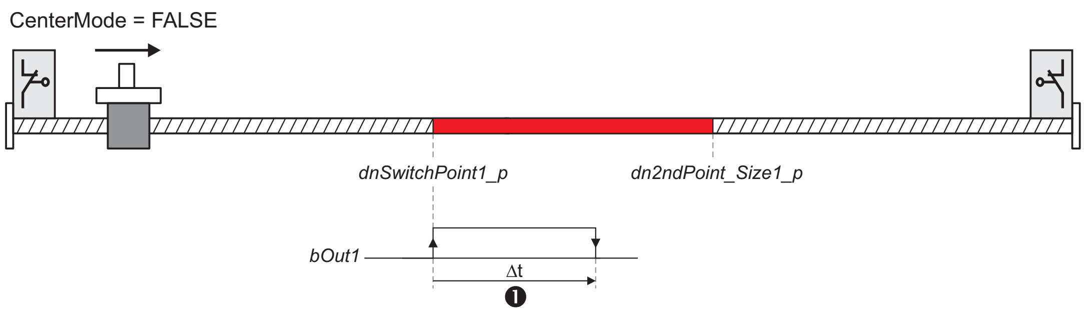
[18-65] Switching hysteresis

**Tip!**

Please note that a delay time set in C01062/x shifts the switching points including the hysteresis.

### 18.1.163.4 Position/time-based cams

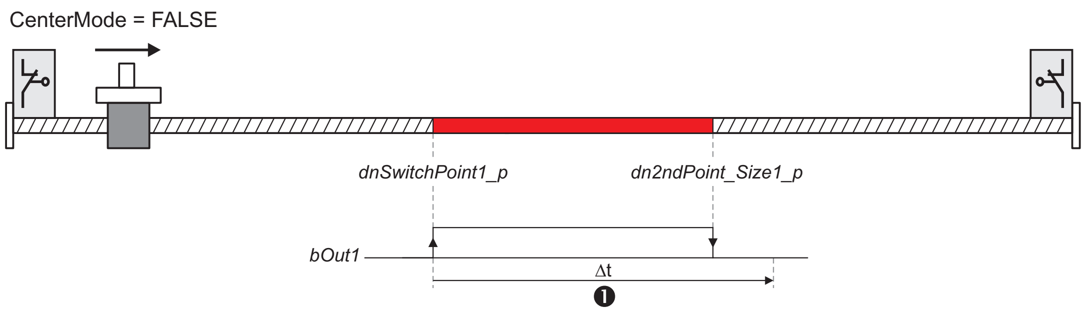
The setting of a running time in C01065/x higher 0 ms serves to realise "position/time-based cams": If the drive reaches the switching range, the *bOut* switching output is set to TRUE and reset to FALSE after the set switching time has elapsed.




❶ C01065/1: Running time

[18-66] Position/time-based cams (here without switching hysteresis and dead time compensation)

If the switching range is left before the running time has expired, the *bOut* switching output is reset to FALSE when the switching range is left:




❶ C01065/1: Running time

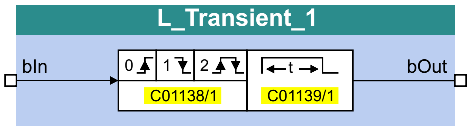
[18-67] Position/time-based cams: Behaviour when the switching range is left before the running time has expired

**Tip!**

Please note that for a resetting of the switching output the switching range must be left first.

## 18.1.164 L_Transient_1

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01138/1. |

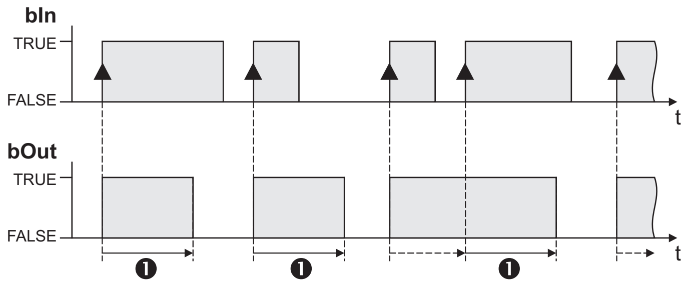
### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01138/1 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01139/1 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

### 18.1.164.1 Function 0: Evaluate rising signal edges



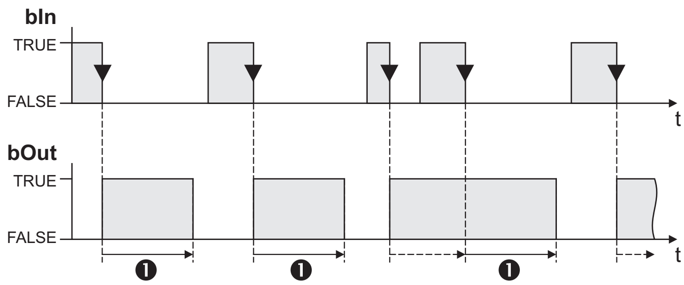

❶ Pulse duration

[18-68] Switching performance for function selection "0: High edge"

#### Functional sequence

1. A FALSE-TRUE edge at the *bIn* input sets the *bOut* output to TRUE.
2. After the parameterised pulse duration has elapsed, the *bOut* output is reset to FALSE unless another FALSE/TRUE edge has been set at the *bIn* input.
   – If an additional FALSE-TRUE edge occurs at the *bIn* input, the pulse duration starts again from the beginning, i.e. the *bOut* output can be retriggered.

### 18.1.164.2 Function 1: Evaluate falling signal edges



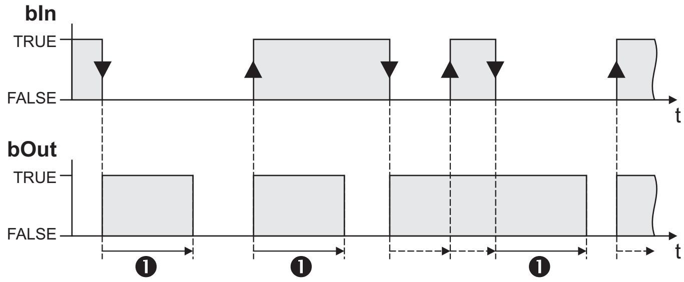

❶ Pulse duration

[18-69] Switching performance for function selection "1: Low edge"

#### Functional sequence

1. A TRUE-FALSE edge at the *bIn* inputs sets the *bOut* output to TRUE.
2. After the parameterised pulse duration has elapsed, the *bOut* output is reset to FALSE unless another TRUE/FALSE edge has been set at the *bIn* input.
   – If an additional TRUE-FALSE edge occurs at the *bIn* input, the pulse duration starts again from the beginning, i.e. the *bOut* output can be retriggered.

### 18.1.164.3 Function 2: Evaluate rising and falling signal edges




➊ Pulse duration

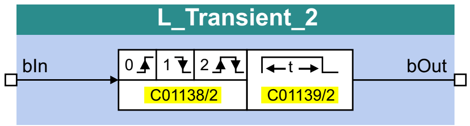
[18-70] Switching performance for function selection "2: High and low edge"

#### Functional sequence

1. A signal change (FALSE/TRUE edge or TRUE/FALSE edge) at the *bIn* input sets the *bOut* output to TRUE.
2. After the parameterised pulse duration has elapsed, the *bOut* output is reset to FALSE unless another signal change has taken place at the *bIn* input.
   – In case of another signal change at the input *bIn*, the pulse time restarts to elapse, i.e. the output *bOut* can be retriggered.

## 18.1.165 L_Transient_2

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01138/2. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

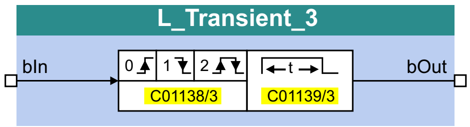
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01138/2 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01139/2 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

## 18.1.166 L_Transient_3

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01138/3. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

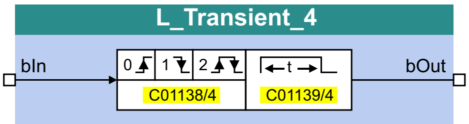
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01138/3 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01139/3 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

## 18.1.167 L_Transient_4

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01138/4. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

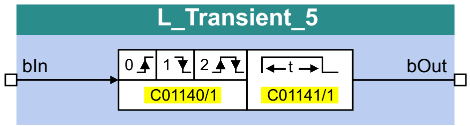
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01138/4 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01139/4 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

## 18.1.168 L_Transient_5

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01140/1. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

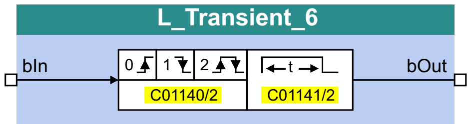
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01140/1 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01141/1 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

## 18.1.169 L_Transient_6

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01140/2. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

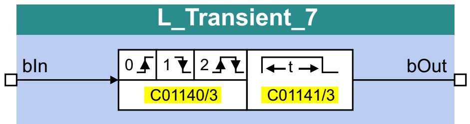
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01140/2 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01141/2 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

## 18.1.170 L_Transient_7

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01140/3. |

### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

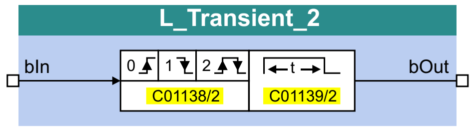
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01140/3 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01141/3 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

### 18.1.171 L_Transient_8

This FB serves to evaluate digital signal edges and convert them into timed, retriggerable pulses. Rising signal edges, falling signal edges or both signal edges can be evaluated.




#### Inputs

| Identifier<br>Data type | Information/possible settings |
|---|---|
| bIn<br>BOOL | Input for edge evaluation<br>• The function depends on the selection of edge evaluation in C01140/4. |

#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bOut<br>BOOL | Output (retriggerable) |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01140/4 | | | | Function<br>• Selection of edge evaluation |
| | 0 | High edge | | Lenze setting |
| | 1 | Low edge | | |
| | 2 | High and low edge | | |
| C01141/4 | 0.001 | s | 60.000 | Pulse duration<br>• Lenze setting: 0.001 s |

For a detailed functional description see L_Transient_1.

## 18.2 System blocks

This chapter describes the system blocks which are available for the controller in the FB Editor.

The function blocks are described in the chapter "Function blocks". (📖 1110)

### Overview of system blocks available

| System block | Function | can be inserted into level: I/O | can be inserted into level: Appl. |
|---|---|---|---|
| LS_AnalogInput | Interface to the analog input terminals<br>▸ Analog terminals (📖 352) | ● | |
| LS_AnalogOutput | Interface to the analog output terminals<br>▸ Analog terminals (📖 352) | ● | |
| LS_AxisBusAux | Interface to the axis bus<br>▸ Axis bus (📖 717) | ● | ● |
| LS_AxisBusIn | | ● | ● |
| LS_AxisBusIO | | ● | ● |
| LS_AxisBusOut | | ● | ● |
| LS_CANManagement | Control of internal functions of the CAN driver and display of the "Operational" status as well as the node address<br>▸ System bus "CAN on board" (📖 638) | ● | ● |
| LS_DataAccess | *Lenze internal only* | | ● |
| LS_DeviceMonitor | Motor control status signals<br>▸ Motor control (MCTRL) (📖 120) | | ● |
| LS_DigitalInput | Interface to the digital input terminals<br>▸ Digital input terminals (📖 329) | ● | |
| LS_DigitalOutput | Interface to the digital output terminals<br>▸ Digital output terminals (📖 349) | ● | |
| LS_DisFree | Display of 8 arbitrary 16-bit signals of the application on display codes | ● | ● |
| LS_DisFree_a | Display of 8 arbitrary analog signals of the application on display codes | ● | ● |
| LS_DisFree_b | Display of 16 arbitrary digital signals of the application on a bit coded display code | ● | ● |
| LS_DisFree_p | Display of 8 arbitrary position signals of the application on display codes | ● | ● |
| LS_DriveInterface | Interface to drive control (DCTRL)<br>▸ Device control (DCTRL) (📖 89) | | ● |
| LS_Keypad | Control via keypad | ● | |
| LS_MotionControlKernel | Interface to the basic drive function implemented in the Motion Control Kernel (MCK)<br>▸ Basic drive functions (MCK) (📖 473) | | ● |
| LS_MotorInterface | Interface to motor control (MCTRL)<br>▸ Motor control (MCTRL) (📖 120) | | ● |
| LS_MultiEncoder | Interface to the Multi-Encoder<br>▸ Multi-Encoder at X8 (📖 302) | ● | ● |
| LS_ParFix<br>LS_ParFix_2 | Output of frequently used constants (TRUE, FALSE, 100 %, etc.) to be used in the interconnection<br>• LS_ParFix_2 is available from version 02.00.00. | ● | ● |

| System block | Function | can be inserted into level: I/O | can be inserted into level: Appl. |
|---|---|---|---|
| LS_ParFree<br>LS_ParFree_2 | Output of 32 parameterisable 16-bit signals<br>• LS_ParFree_2 is available from version 02.00.00. | ● | ● |
| LS_ParFree_a<br>LS_ParFree_a_2 | Output of 16 parameterisable analog signals<br>• LS_ParFree_a_2 is available from version 02.00.00. | ● | ● |
| LS_ParFree_b | Output of 32 parameterisable digital signals | ● | ● |
| LS_ParFree_p | Output of 8 parameterisable position signals | ● | ● |
| LS_ParFree_v<br>LS_ParFree_v_2 | Output of 8 parameterisable speed signals<br>• LS_ParFree_v_2 is available from version 02.00.00. | ● | ● |
| LS_ParFree32 | Output of 8 parameterisable 32-bit signals<br>• This SB is available from version 02.00.00. | ● | ● |
| LS_ParFreeUnit<br>LS_ParFreeUnit_2 | Output of 16 parameterisable position signals with internal conversion of [unit] in [increments]<br>• These SBs are available from version 02.00.00. | ● | ● |
| LS_ParReadWrite_1<br>...<br>LS_ParReadWrite_6 | Reading/Writing of local parameters | ● | ● |
| LS_PulseGenerator | Output of 9 fixed frequencies and 1 parameterisable frequency | ● | ● |
| LS_Resolver | Interface to the resolver<br>▶ Resolver at X7 (📖 298) | ● | ● |
| LS_RetainData | Selection and saving of retain data<br>• This SB is available from version 02.00.00. | ● | ● |
| LS_SetError_1 | Parameterisable responses to user-defined events are tripped<br>▶ Diagnostics & error management (📖 581) | ● | ● |
| LS_SetError_2 | | ● | ● |
| LS_SyncManagement | Output of status information for synchronising the internal time base<br>▶ Synchronisation of the internal time base (📖 731) | ● | ● |
| LS_TouchProbe | Interface for touch probe detection<br>▶ Touch probe detection (📖 362) | ● | ● |
| LS_WriteParamList | Interface to the basic "Parameter change-over" function<br>▶ Parameter change-over (📖 734) | ● | |

**Related topics:**

▶ Overview of function blocks available (📖 1110)

▶ Working with the FB Editor (📖 1058)

### 18.2.1 LS_AnalogInput

Interface to the analog input terminals.

For a detailed description see the main chapter "I/O terminals":

▶ Internal interfaces | System block "LS_AnalogInput" (📖 360)

### 18.2.2 LS_AnalogOutput

Interface to the analog output terminals.

For a detailed description see the main chapter "I/O terminals":

▶ Internal interfaces | System block "LS_AnalogInput" (📖 360)

### 18.2.3 LS_AxisBusAux

Interface to the axis bus.

For a detailed description see main chapter "Axis bus":

▶ Internal interfaces | System block "LS_AxisBusAux" (📖 724)

### 18.2.4 LS_AxisBusIn

Interface to the axis bus.

For a detailed description see main chapter "Axis bus":

▶ Internal interfaces | System block "LS_AxisBusIn" (📖 723)

### 18.2.5 LS_AxisBusIO

Interface to the axis bus.

For a detailed description see main chapter "Axis bus":

▶ Internal interfaces | System block "LS_AxisBusIO" (📖 730)

### 18.2.6 LS_AxisBusOut

Interface to the axis bus.

For a detailed description see main chapter "Axis bus":
- Internal interfaces | System block "LS_AxisBusOut" (722)

### 18.2.7 LS_CANManagement

Control of internal functions of the CAN driver and display of the "Operational" status as well as the node address.

For a detailed description see the main chapter "System bus CAN on board":
- Internal interfaces | System block "LS_CANManagement" (710)

### 18.2.8 LS_DataAccess

Only for Lenze-internal use.

### 18.2.9 LS_DeviceMonitor

Motor control status signals.

For a detailed description see the main chapter "Motor control (MCTRL)":
- Internal status signals | system block "LS_DeviceMonitor" (291)

### 18.2.10 LS_DigitalInput

Interface to the digital input terminals.

For a detailed description see the main chapter "I/O terminals":
- Internal interfaces | System block "LS_DigitalInput" (342)

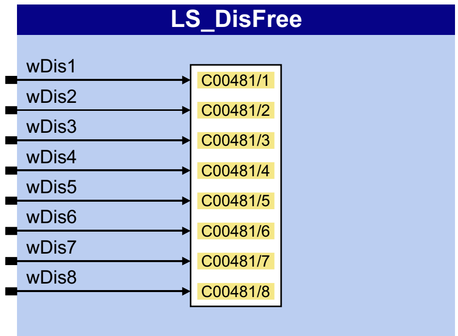
### 18.2.11 LS_DigitalOutput

Interface to the digital output terminals.

For a detailed description see the main chapter "I/O terminals":
- Internal interfaces | System block "LS_DigitalOutput" (351)

## 18.2.12 LS_DisFree

This system block displays 8 arbitrary 16-bit signals of the application on display codes.




### Inputs

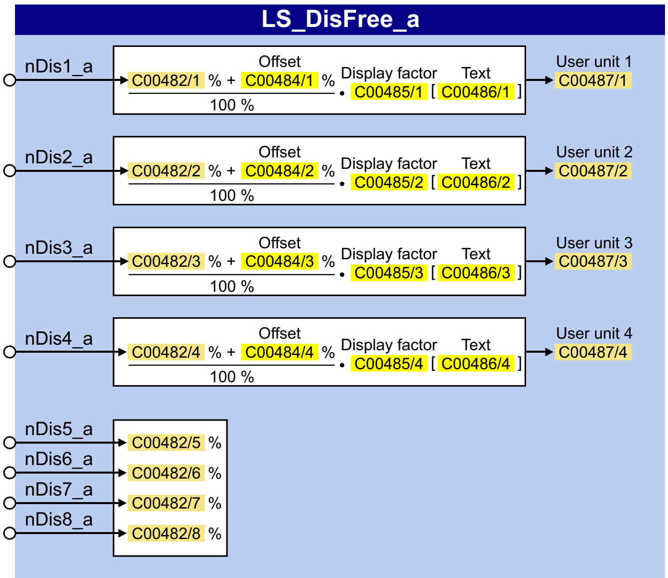
| Identifier<br>Data type | Information/possible settings |
|---|---|
| wDis1 ... wDis8<br>WORD | Inputs for any 16-bit signal of the application |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00481/1...8 | 0x0000 | | 0xFFFF | Display of the 16-bit signals which are applied at the *wDis1 ... wDis8* inputs |

## 18.2.13 LS_DisFree_a

This system block displays 8 arbitrary analog signals of the application on display codes.




### Inputs

| **Identifier** Data type | **Information/possible settings** |
|---|---|
| nDis1_a ... nDis8_a INT<br>From version 02.00.00:<br>nC482_1_a ... nC482_8_a | Inputs for any analog signal of the application<br>**Note:** From version 02.00.00 the inputs are named according to the display parameter for an easier allocation. |

### Parameter

| **Parameter** | **Possible settings** | | | **Info** |
|---|---|---|---|---|
| C00482/1...8 | -199.99 | % | 199.99 | Display of the analog signals which are applied at the *nDis1_a ... nDis8_a* inputs |
| C00484/1...4<br>...<br>C00487/1...4 | ▸ Display of internal process variables in application units | | | |

### 18.2.13.1 Display of internal process variables in application units

In addition to the display in percent in C00482/1...8, for the first four analog signals *nDis1_a* ... *nDis4_a* the configurable display parameters C00487/1...4 are provided. Via these display parameters, internal process variables can be displayed, e.g. on the keypad, with an individual scaling and an individual unit.

**Configuration of the display parameters (C00487/1...4):**

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00484/1...4 | -199.99 | % | 199.99 | Offset 1 ... 4<br>• See formula [18-71].<br>• Lenze setting: 0.00 % |
| C00485/1...4 | -65536.0000 | | 65536.0000 | Display factor 1 ... 4<br>• Scaling of the input variable for the display.<br>• See formula [18-71].<br>• Lenze setting: 1.0000 |
| C00486/1...4 | String of digits | | | Text 1 ... 4<br>• For each display value, an individual unit (e.g. "bottles") can be set. |

$$\text{Application unit 1} = \frac{\text{nDis1\_a [\%]} + \text{Offset 1 [\%]}}{\text{100 [\%]}} \cdot \text{Display factor 1 [text 1]}$$

[18-71] Formula for scaling the display

**Example 1:**

- Input variable *nDis1_a* = 100 %
- Offset 1 (C00484/1) = 0 %
- Display factor 1 (C00485/1) = 123.45
- Text 1 (C00486/1) = "bottles"

$$\text{Application unit 1} = \frac{100\text{ [\%]} + 0\text{ [\%]}}{100\text{ [\%]}} \cdot 123.45\text{ [bottles]} = 123.45\text{ bottles}$$

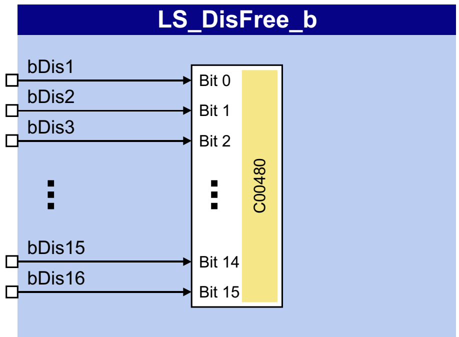
**Example 2:**

- Input variable *nDis2_a* = 40 %
- Offset 2 (C00484/2) = 35 %
- Display factor 2 (C00485/2) = 20
- Text 2 (C00486/2) = "kg"

$$\text{Application unit 2} = \frac{40\text{ [\%]} + 35\text{ [\%]}}{100\text{ [\%]}} \cdot 20\text{ [kg]} = 15.00\text{ kg}$$

## 18.2.14 LS_DisFree_b

This system block displays 16 arbitrary digital signals of the application on a bit coded display code.




### Inputs

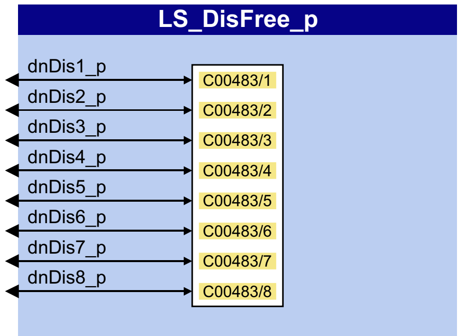
| Identifier<br>Data type | Information/possible settings |
|---|---|
| bDis1 ... bDis16<br>BOOL | Inputs for any digital signal of the application |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00480 | 0x0000 | | 0xFFFF | Display of the digital signals as hexadecimal values which are applied at the *bDis1* ... *bDis16* inputs |
| | Bit 0 | Signal level at the *bDis1* input | | |
| | Bit 1 | Signal level at the *bDis2* input | | |
| | Bit 2 | Signal level at the *bDis3* input | | |
| | ... | ... | | |
| | Bit 15 | Signal level at the *bDis16* input | | |

## 18.2.15 LS_DisFree_p

This system block displays 8 arbitrary position signals of the application on display codes.




### Inputs

| **Identifier** Data type | **Information/possible settings** |
|---|---|
| dnDis1_p ... dnDis8_p DINT | Inputs for arbitrary position signals of the application |

### Parameter

| **Parameter** | **Possible settings** | | | **Info** |
|---|---|---|---|---|
| C00483/1...8 | -2147483647 | Incr. | 2147483647 | Display of the position signals which are applied at the *dnDis1_p* ... *dnDis8_p* inputs |

## 18.2.16 LS_DriveInterface

Interface to internal device control.

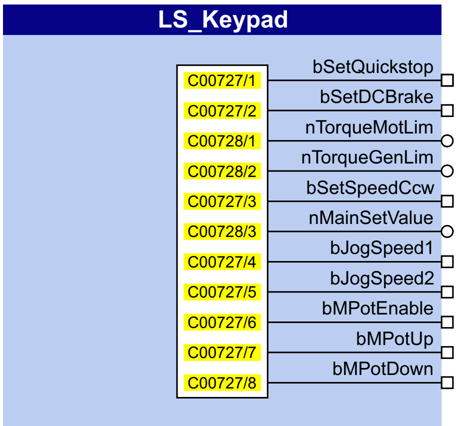
For a detailed description see main chapter "Device control (DCTRL)":

- Internal interfaces | "LS_DriveInterface" system block (115)

### 18.2.17 LS_Keypad

This system block is used on I/O interconnection level if the "Keypad" control mode has been selected in C00007.

In the "Keypad" control mode, the **LS_Keypad** system block passes on various setpoints and control commands to the technology application which can be selected/activated via codes using the keypad.




#### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bSetQuickstop<br>BOOL | C00727/1 = "1" ≡ Request quick stop |
| bSetDCBrake<br>BOOL | C00727/2 = "1" ≡ Request DC-injection braking |
| nTorqueMotLim<br>INT | Torque limit in motor mode set in C00728/1<br>• Lenze setting: 100.00 % |
| nTorqueGenLim<br>INT | Torque limit in generator mode set in C00728/2<br>• Lenze setting: 100.00 % |
| bSetSpeedCcw<br>BOOL | C00727/3 = "1" ≡ Request reversal |
| nMainSetValue<br>INT | Setpoint speed set in C00728/3<br>• Lenze setting: 0.00 % |
| bJogSpeed1<br>BOOL | C00727/4 = "1" ≡ Request fixed speed setpoint 1 |
| bJogSpeed2<br>BOOL | C00727/5 = "1" ≡ Request fixed speed setpoint 2 |
| bMPotEnable<br>BOOL | C00727/6 = "1" ≡ Motor potentiometer: Request activation |
| bMPotUp<br>BOOL | C00727/7 = "1" ≡ Motor potentiometer: Request positive acceleration |
| bMPotDown<br>BOOL | C00727/8 = "1" ≡ Motor potentiometer: Request negative acceleration |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00727/1...8 | 0 | | 1 | Keypad digital values<br>• Execution of control commands for keypad operation<br>• See the "Outputs" table for the meaning of the individual subcodes |
| C00728/1...3 | -199.99 | % | 199.99 | Analog values - keypad<br>• Specification of various setpoints for keypad operation<br>• See the "Outputs" table for the meaning of the individual subcodes |

## 18.2.18 LS_MotionControlKernel

Interface to the basic drive functions implemented in **Motion Control Kernel** (MCK).

For a detailed description see the main chapter "Basic drive functions":

▸ Internal interfaces | System block "LS_MotionControlKernel" (475)

## 18.2.19 LS_MotorInterface

Interface to internal motor control.

For a detailed description see the main chapterl "Motor control (MCTRL)":

▸ Internal interfaces | system block "LS_MotorInterface" (285)

## 18.2.20 LS_MultiEncoder

Interface to the Multi-Encoder.

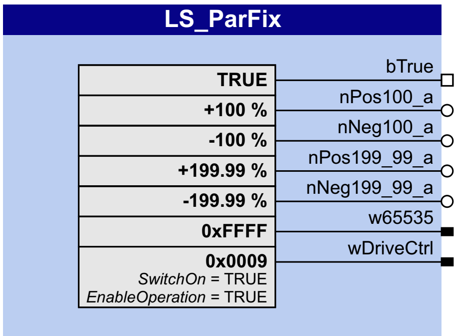
For detailed information please see the main chapter "Encoder/Feedback system":

▸ Internal interfaces | "LS_MultiEncoder" system block (312)

## 18.2.21 LS_ParFix

This system block outputs various fixed values (constants) to be used in the interconnection.

▶ From version 02.00.00, another **LS_ParFix_2** system block is available which can be used e.g. in the application level.

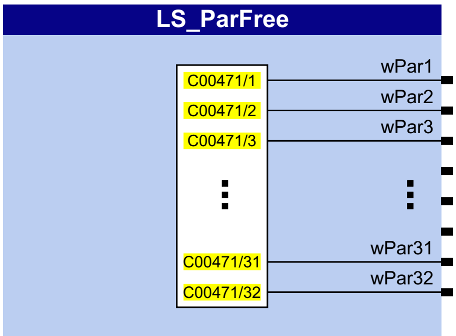



### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| bTrue<br>BOOL | 1 ≡ TRUE |
| nPos100_a<br>INT | 16384 ≡ + 100 % |
| nNeg100_a<br>INT | -16384 ≡ - 100 % |
| nPos199_99_a<br>INT | 32767 ≡ + 199.99 % |
| nNeg199_99_a<br>INT | -32767 ≡ - 199.99 % |
| w65535<br>WORD | 65535 ≡ 0xFFFF |
| wDriveCtrl<br>WORD | 9 ≡ 0x0009<br>• Bit 0, SwitchOn = TRUE<br>• Bit 3, EnableOperation = TRUE<br>• All others: FALSE<br>See also:<br>▸ wCANControl/wMCIControl control words (📖 118) |

### 18.2.22 LS_ParFree

This system block outputs 32 parameterisable 16-bit signals.




#### Outputs

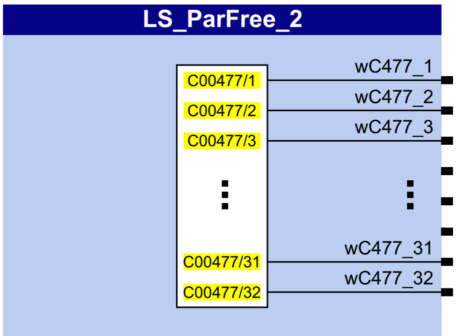
| Identifier<br>Data type | Value/meaning |
|---|---|
| wPar1 ... wPar32<br>WORD<br>From version 02.00.00:<br>wC471_1 ... wC471_32 | Output of the 16-bit signals parameterised in C00471/1...32<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00471/1...32 | 0x0000 | | 0xFFFF | Setting of the 16-bit signals to be output |

## 18.2.23 LS_ParFree_2

This system block outputs 32 parameterisable 16-bit signals.




### Outputs

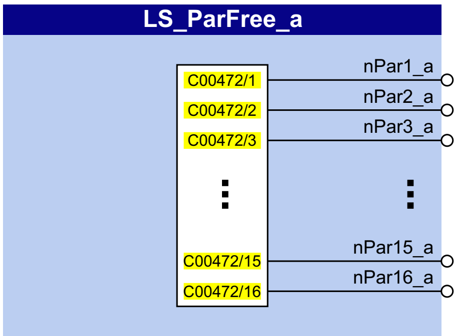
| Identifier<br>Data type | Value/meaning |
|---|---|
| wC477_1 ... wC477_32<br>WORD | Output of the 16-bit signals parameterised in C00477/1...32 |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00477/1...32 | 0x0000 | | 0xFFFF | Setting of the 16-bit signals to be output |

### 18.2.24 LS_ParFree_a

This system block outputs 16 parameterisable analog signals.




#### Outputs

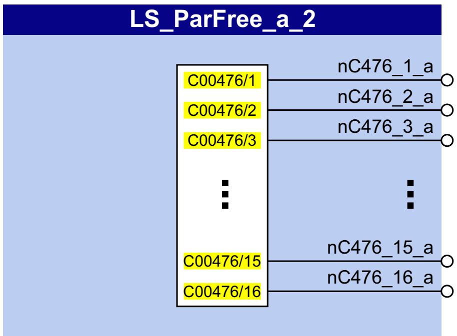
| Identifier<br>Data type | Value/meaning |
|---|---|
| nPar1_a ... nPar16_a<br>INT<br>From version 02.00.00:<br>nC472_1_a ... nC472_16_a | Output of the analog signals parameterised in C00472/1...16<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00472/1...16 | -199.99 | % | +199.99 | Selection of analog signals to be output |

## 18.2.25 LS_ParFree_a_2

This system block outputs 16 parameterisable analog signals.




### Outputs

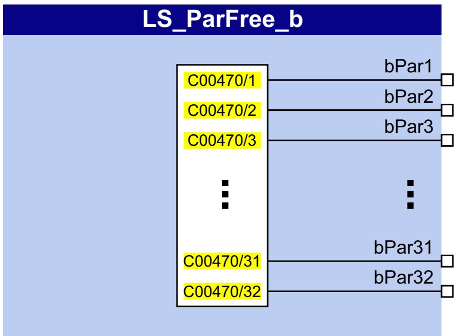
| Identifier<br>Data type | Value/meaning |
|---|---|
| nC476_1_a ... nC476_16_a<br>INT | Output of the analog signals parameterised in C00476/1...16 |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00476/1...16 | -199.99 | % | +199.99 | Selection of analog signals to be output |

### 18.2.26 LS_ParFree_b

This system block outputs 32 parameterisable digital signals.




#### Outputs

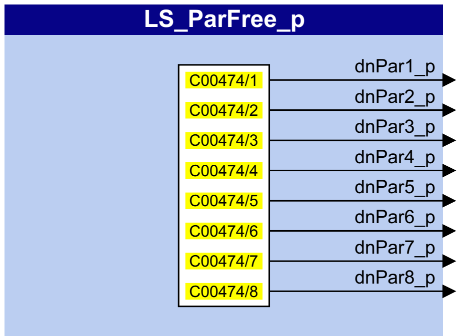
| Identifier<br>Data type | Value/meaning |
|---|---|
| bPar1 ... bPar32<br>BOOL<br>From version 02.00.00:<br>bC470_1 ... bC470_32 | Output of the signal levels (FALSE/TRUE) parameterised in C00470/1...32<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

#### Parameter

| Parameter | Possible settings | | Info |
|---|---|---|---|
| C00470/1...32 | | | Selection of signal levels to be output<br>• Bit 0 ... 31 = *bPar1* ... *bPar32* |
| | 0 | "FALSE" signal is output | |
| | 1 | "TRUE" signal is output | |

## 18.2.27 LS_ParFree_p

This system block outputs 8 parameterisable position signals.




### Outputs

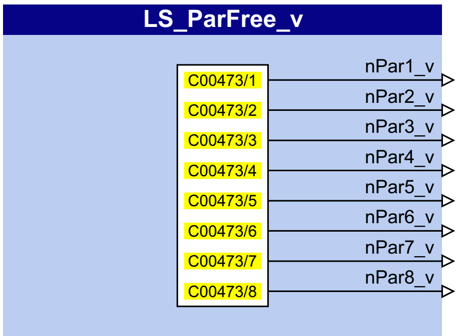
| Identifier<br>Data type | Value/meaning |
|---|---|
| dnPar1_p ... dnPar8_p<br>DINT<br>From version 02.00.00:<br>dnC474_1_p ... dnC474_8_p | Output of the position signals parameterised in C00474/1...8<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00474/1...8 | -2147483647 | Incr. | 2147483647 | Setting of the position signals to be output |

### 18.2.28 LS_ParFree_v

This system block outputs 8 parameterisable speed signals.




#### Outputs

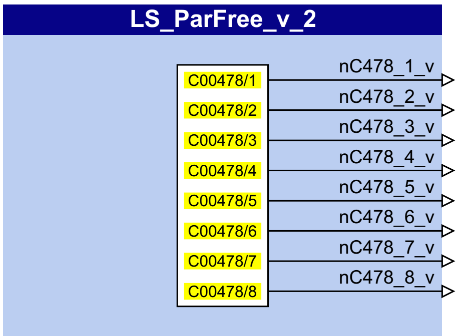
| Identifier<br>Data type | Information/possible settings |
|---|---|
| nPar1_v ... nPar8_v<br>INT<br>From version 02.00.00:<br>nC473_1_v ... nC473_8_v | Output of the speed signals parameterised in C00473/1...8<br><br>**Note:** From version 02.00.00 the outputs are named according to the respective setting parameter for an easier allocation. |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00473/1...8 | -32767 | Incr/ms | +32767 | Selection of speed signals to be output |

## 18.2.29 LS_ParFree_v_2

This system block outputs 8 parameterisable speed signals.




### Outputs

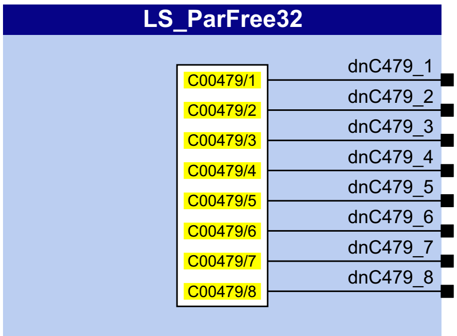
| Identifier<br>Data type | Information/possible settings |
|---|---|
| nC478_1_v ... nC478_8_v<br>INT | Output of the speed signals parameterised in C00478/1...8 |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00478/1...8 | -32767 | Incr/ms | +32767 | Selection of speed signals to be output |

### 18.2.30 LS_ParFree32

This system block outputs 8 parameterisable 32-bit signals.




#### Outputs

| **Identifier** Data type | **Value/meaning** |
|---|---|
| dnC479_1 ... dnC479_8 DINT | Output of the 32-bit signals parameterised in C00479/1...8 |

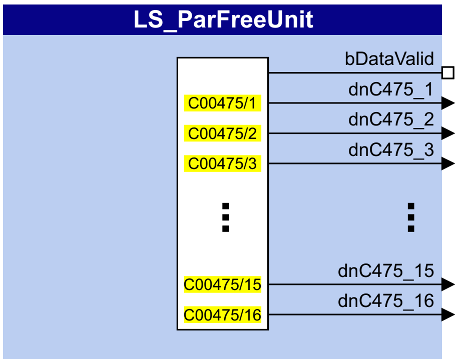
#### Parameter

| **Parameter** | **Possible settings** | | | **Info** |
|---|---|---|---|---|
| C00479/1...8 | -2147483647 | | 2147483647 | Setting of the 32-bit signals to be output |

## 18.2.31 LS_ParFreeUnit

This system block outputs 16 parameterisable position signals.

- In contrast to the SB LS_ParFree_p, the positions are set in the application unit [unit] instead of in [increments].
- The *bDataValid* status signal indicates that the internal position conversion is completed and the output data are consistent.




### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bDataValid<br>BOOL | Status signal "Position conversion completed, data consistent" | |
| | TRUE | The conversion of the positions from [unit] to [increments] is completed. |
| dnC475_1 ... dnC475_16<br>DINT | Output of the position signals parameterised in C00475/1...16 in [increments]<br>• Observe *bDataValid* status signal! | |

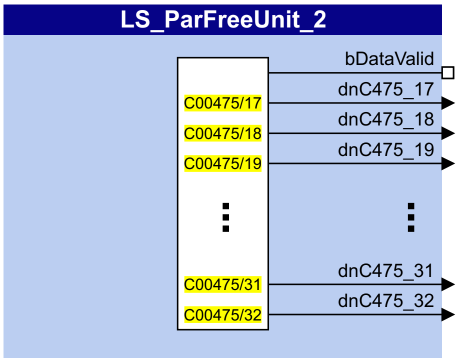
### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00475/1...16 | -214748.3647 | unit | 214748.3647 | Setting of the position signals to be output |

### 18.2.32 LS_ParFreeUnit_2

This system block outputs 16 parameterisable position signals.

- In contrast to the SB LS_ParFree_p, the positions are set in the application unit [unit] instead of in [increments].
- The *bDataValid* status signal indicates that the internal position conversion is completed and the output data are consistent.




#### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bDataValid<br>BOOL | Status signal "Position conversion completed, data consistent" | |
| | TRUE | The conversion of the positions from [unit] to [increments] is completed. |
| dnC475_17 ... dnC475_32<br>DINT | Output of the position signals parameterised in C00475/17...32 in [increments]<br>• Observe *bDataValid* status signal! | |

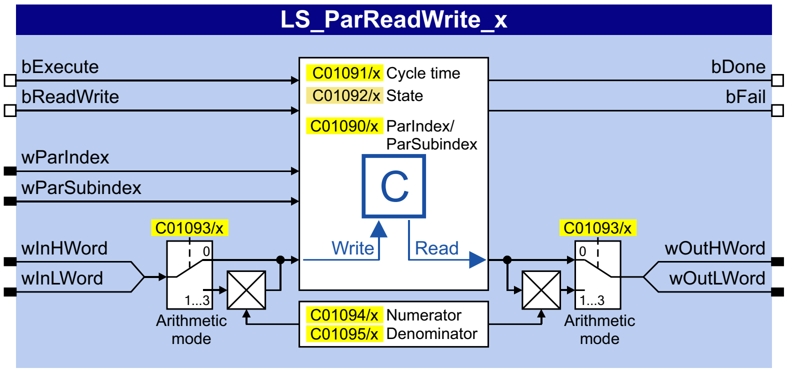
#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00475/17...32 | -214748.3647 | unit | 214748.3647 | Setting of the position signals to be output |

### 18.2.33 LS_ParReadWrite_1-6

The **LS_ParReadWrite_1** ... **LS_ParReadWrite_6** system blocks are used for reading and writing local parameters.

- If several system blocks are activated at the same time, the blocks will be processed one at a time every time the main program is executed.
- The SB supports one-time and cyclic reading/writing in an adjustable time interval.




#### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bExecute<br>BOOL | Trip read/write request | |
| | FALSE↗TRUE | If cycle time (C01091) = "0 ms":<br><u>One-time</u> reading/writing of the parameter value which has been addressed via the*wParIndex* and *wParSubindex* inputs. |
| | | If cycle time (C01091) > "0 ms":<br><u>Cyclic</u> reading/writing of the parameter value which has been addressed via the *wParIndex* and *wParSubindex* inputs. |
| | TRUE↘FALSE | Deactivate cyclic reading/writing again. |
| bReadWrite<br>BOOL | Selection: Read or write request | |
| | FALSE | Read request |
| | TRUE | Write request |
| wParIndex<br>WORD | Code to be read or written.<br>• As an alternative, this selection can be carried out via C01090. | |
| wParSubindex<br>WORD | Subcode to be read or written.<br>• As an alternative, this selection can be carried out via C01090. | |
| wInHWord<br>wInLWord<br>WORD | Value to be written (DataHigh/DataLow portion) | |

### Outputs

| Identifier<br>Data type | Value/meaning | |
|---|---|---|
| bDone<br>BOOL | "Read/Write request successfully completed" status signal<br>• The output is automatically reset to FALSE if a new request is activated via *bExecute* or the cycle time (C01091) expires. | |
| | TRUE | Read/Write request successfully completed. |
| | FALSE | The FALSE status can have the following meanings:<br>1. There is no active read/write request.<br>2. The read/write request has not been completed yet.<br>3. An error has occurred (if *bFail* = TRUE). |
| bFail<br>BOOL | "Error" status | |
| | TRUE | An error has occurred (group signal).<br>• See display parameter (C01092) for details. |
| wOutHWord<br>wOutLWord<br>WORD | Value which was read (DataHigh/DataLow portion) after read request | |

### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01090/1...6 | 0,000 | | 16000,000 | Parameter to be read or written.<br>• For a setting of "0,000", inputs *wParIndex* and *wParSubindex* are effective for addressing purposes instead.<br>• Lenze setting: 0,000 |
| | Format:<br><code number>,<subcode number> | | | |
| C01091/1...6 | | | | Cycle time<br>• Subcode 1 = **LS_ParReadWrite_1**<br>Subcode 2 = **LS_ParReadWrite_2**<br>...<br>Subcode 6 = **LS_ParReadWrite_6**<br>• Lenze setting: 0 |
| | 0 | One-time reading/writing at *bExecute* in case of a FALSE/TRUE edge | | |
| | Cyclic reading/writing: | | | |
| | 20 | 20 ms | | |
| | 50 | 50 ms | | |
| | 100 | 100 ms | | |
| | 200 | 200 ms | | |
| | 500 | 500 ms | | |
| | 1000 | 1 s | | |
| | 2000 | 2 s | | |
| | 5000 | 5 s | | |
| | 10000 | 10 s | | |

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01092/1...6 | | | | Error status<br>• If *bFail* = TRUE:<br>Error status is displayed.<br>• Subcode 1 = **LS_ParReadWrite_1**<br>Subcode 2 = **LS_ParReadWrite_2**<br>...<br>Subcode 6 = **LS_ParReadWrite_6** |
| | 0 | No error | | |
| | 33803 | Invalid data type (e.g. STRING) | | |
| | 33804 | Limit violation | | |
| | 33806 | Invalid code | | |
| | 33813 | No element in the selection list | | |
| | 33815 | Writing of the parameter not permitted | | |
| | 33816 | Writing of the parameter only permitted if controller is inhibited | | |
| | 33829 | Invalid subcode | | |
| | 33865 | No parameter with subcodes | | |
| C01093/1...6 | | | | Arithmetic mode<br>• Lenze setting: "0: No arithmetic"<br>▸ Arithmetic function |
| | 0 | No arithmetic | | |
| | 1 | In16Bit: LW=+/-32767 | | |
| | 2 | In16Bit: HW=+/-; LW=0..65535 | | |
| | 3 | In32Bit: HW_LW=+/-2147483647 | | |
| C01094/1...6 | -32767 | | 32767 | Numerator<br>• For internal conversion into arithmetic modes 1 ... 3.<br>• Lenze setting: 1 |
| C01095/1...6 | 1 | | 32767 | Denominator<br>• For internal conversion into arithmetic modes 1 ... 3.<br>• Lenze setting: 1 |

### 18.2.33.1 Arithmetic function

The integrated arithmetic function allows for easy arithmetic conversion of the process value to be written or which was read into the format of the target parameter via parameterisable factors and without the need for an additional arithmetic FB.

▶ In C01093, the interpretation of the *wInHWord* and *wInLWord* inputs can be set to be able to write to parameters:

| Arithmetic mode | | wInHWord | wInLWord | Internal conversion |
|---|---|---|---|---|
| 0 | No arithmetic<br>(Lenze setting) | **INTEGER_32**<br>(4 bytes with sign) | | No<br>(behaviour as before) |
| | | DataHigh component | DataLow component | |
| 1 | In16Bit:<br>LW=+/-32767 | - | **INTEGER_16**<br>(2 bytes with sign) | Yes<br>(see section below) |
| 2 | In16Bit: HW=+/-;<br>LW=0..65535 | **Sign**<br>(0 ≡ positive value) | **UNSIGNED_16**<br>(2 bytes without sign) | |
| 3 | In32Bit: HW_LW=<br>+/-2147483647 | **INTEGER_32**<br>(4 bytes with sign) | | |
| | | DataHigh component | DataLow component | |

### Internal conversion

If arithmetic modes 1 ... 3 are selected in C01093, the input value / read parameter value is internally converted via parameterisable factors.

- Division is not remainder considered.

$$\text{Parameter value to be written} = \text{Input value}_{[32]} \cdot \frac{\text{Numerator}_{[16]}}{\text{Denominator}_{[16]}}$$

C01094: Numerator
C01095: Denominator

[18-72] Internal conversion for write access

$$\text{Output value}_{[32]} = \text{Read parameter value} \cdot \frac{\text{Numerator}_{[16]}}{\text{Denominator}_{[16]}}$$

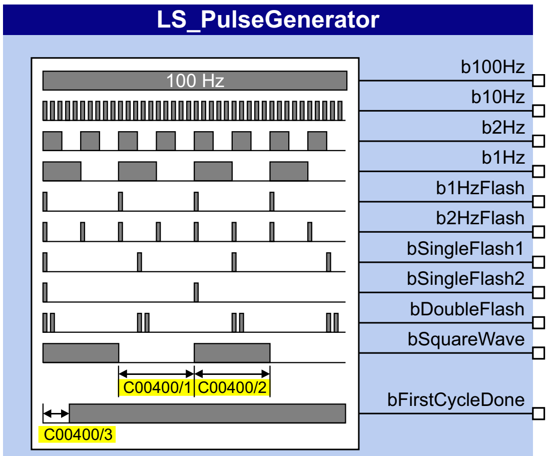
C01094: Numerator
C01095: Denominator

[18-73] Internal conversion for read access

## 18.2.34 LS_PulseGenerator

This system block outputs 9 different fixed frequencies and 1 frequency with parameterisable pulse/dead time.

The SB provides a TRUE signal at the *bFirstCycleDone* output when the first 1-ms processing cycle is completed and the time set in 1C00400/3 has expired. This status signal can e.g. be used for the delayed enable of peripheral devices or motor control setpoints so that all required initial values are calculated first after the controller switch-on.




### Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| b100Hz<br>BOOL | Rectangular signal 100 Hz |
| b10Hz<br>BOOL | Rectangular signal 10 Hz |
| b2Hz<br>BOOL | Rectangular signal 2 Hz |
| b1Hz<br>BOOL | Rectangular signal 1 Hz |
| b1HzFlash<br>BOOL | 80 ms-pulse, repetition rate every second |
| b2HzFlash<br>BOOL | 80 ms-pulse, repetition rate every 0.5 seconds |
| bSingleFlash1<br>BOOL | 80 ms pulse, repetition rate every 1.25 seconds |
| bSingleFlash2<br>BOOL | 80 ms pulse, repetition rate every 2 seconds |
| bDoubleFlash<br>BOOL | 80 ms-double pulse, repetition rate every 1.25 seconds |
| bSquareWave<br>BOOL | Output frequency with pulse/dead time set in C00400/1...2 |
| bFirstCycleDone<br>BOOL | Status signal "First processing cycle completed"<br>TRUE: The first 1-ms processing cycle has been completed and the time set in C00400/3 has expired (i.e. all FBs have been called at least once). |

#### Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C00400/1 | 0 | ms | 60000 | Length of LOW level (break)<br>• For output *bSquareWave*<br>• Lenze setting: 1000 ms |
| C00400/2 | 0 | ms | 60000 | Length of HIGH level<br>• For output *bSquareWave*<br>• Lenze setting: 1000 ms |
| C00400/3 | 0 | ms | 60000 | Delay of the *bFirstCycleDone* status<br>• Lenze setting: 100 ms |

### 18.2.35 LS_Resolver

Interface to the resolver.

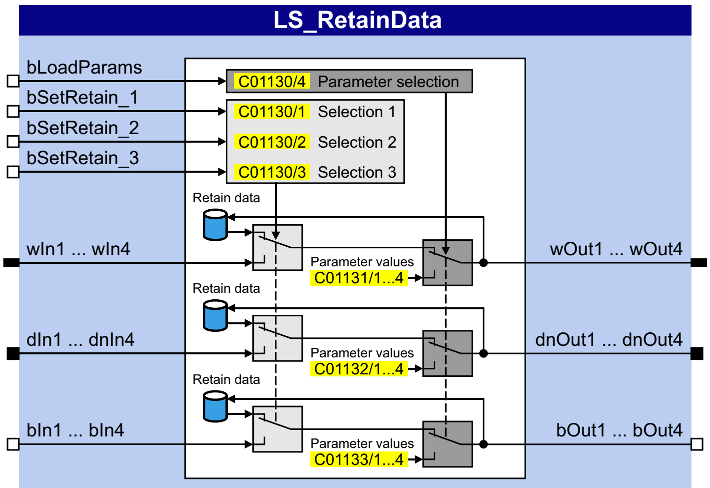
For detailed information please see the main chapter "Encoder/Feedback system":

▸ Internal interfaces | "LS_Resolver" system block (📖 301)

## 18.2.36 LS_RetainData

This system block serves to select and save retain data similar to the FBs L_SampleHold_1 and L_SampleHold_2.

- Retain data are values that are saved automatically at mains disconnection or disconnection of the 24-V supply in the controller and are restored when the 24-V supply is switched on again.
- A total of 12 retain data can be stored in the retain memory:
  - 4 WORD signals (16 bits, 0 ... 65535)
  - 4 DINT signals (32 bits, -2147483647 ... 2147483647)
  - 4 BOOL signals (FALSE or TRUE)




### Inputs

| Identifier<br>Data type | Information/possible settings | |
|---|---|---|
| bLoadParams<br>BOOL | Set selected outputs & retain values to parameter values.<br>• This input has priority over the *bSetRetain* inputs.<br>• Typical application: Initialisation of the outputs & retain values. | |
| | FALSE | The input or retain values are output, depending on the *bSetRetain* input. |
| | TRUE | The outputs selected in C01130/4 and the respective retain values are set to the parameter values and held as long as *bLoadParams* is TRUE.<br>If the parameter values change during this time, the respective outputs and retain values change as well. |
| bSetRetain_1...3<br>BOOL | Transfer selected input values to the retain memory ("latch") | |
| | FALSE | No transfer. Retain values or parameter values (when *bLoadParams* = TRUE) are provided at the outputs. |
| | TRUE | The input values selected in C01130/1...3 are stored in the retain memory and provided at the outputs. |

| Identifier<br>Data type | Information/possible settings |
|---|---|
| wIn1 ... wIn4<br>WORD | Input values |
| dnIn1 ... dnIn4<br>DINT | |
| bIn1 ... bIn4<br>BOOL | |

## Outputs

| Identifier<br>Data type | Value/meaning |
|---|---|
| wOut1 ... wOut4<br>WORD | Output of the input or retain values, depending on the *bSetRetain* input |
| dnOut1 ... dnOut4<br>DINT | |
| bOut1 ... bOut4<br>BOOL | |

## Parameter

| Parameter | Possible settings | | | Info |
|---|---|---|---|---|
| C01130/1...4 | Setting is bit coded: | | | Subcodes 1 ... 3: Selection 1 ... 3<br>• Select which input values are to be stored in the retain memory when setting the respective *bSetRetain*-input to TRUE.<br>• Lenze setting: 0x0000<br><br>Subcode 4: Parameter selection<br>• Select which parameter values are to be stored in the retain memory when setting *bLoadParams* to TRUE.<br>• Lenze setting: 0x0000 |
| | | Subcodes 1 ... 3: | Subcode 4: | |
| | Bit 0 | wIn1 | C01131/1 | |
| | Bit 1 | wIn2 | C01131/2 | |
| | Bit 2 | wIn3 | C01131/3 | |
| | Bit 3 | wIn4 | C01131/4 | |
| | Bit 4 | dnIn1 | C01132/1 | |
| | Bit 5 | dnIn2 | C01132/2 | |
| | Bit 6 | dnIn3 | C01132/3 | |
| | Bit 7 | dnIn4 | C01132/4 | |
| | Bit 8 | bIn1 | C01133/1 | |
| | Bit 9 | bIn2 | C01133/2 | |
| | Bit 10 | bIn3 | C01133/3 | |
| | Bit 11 | bIn4 | C01133/4 | |
| | Bit 12 | Reserved | | |
| | Bit 13 | | | |
| | Bit 14 | | | |
| | Bit 15 | | | |
| C01131/1...4 | 0 | | 65536 | Parameter value for retain memory (outputs *wOut1 ... wOut4*)<br>• Lenze setting: 0 |
| C01132/1...4 | -2147483647 | | 2147483647 | Parameter values for retain memory (outputs *dnOut1 ... dnOut4*)<br>• Lenze setting: 0 |
| C01133/1...4 | | | | Parameter values for retain memory (outputs *bOut1 ... bOut4*)<br>• Lenze setting: 0: FALSE |
| | 0 | FALSE | | |
| | 1 | TRUE | | |

### 18.2.37 LS_SetError_1

Parameterisable responses to user-defined events are tripped.

For a detailed description see the main chapter "Diagnostics & error management":

▶ System block "LS_SetError_1" (📖 634)

### 18.2.38 LS_SetError_2

Parameterisable responses to user-defined events are tripped.

For a detailed description see the main chapter "Diagnostics & error management":

▶ System block "LS_SetError_2" (📖 636)

### 18.2.39 LS_SyncManagement

Output of status information for synchronising the internal time base.

For a detailed description see main chapter "Synchronisation of the internal time base of the controller":

▶ Internal interfaces | System block "LS_SyncManagement" (📖 733)

### 18.2.40 LS_TouchProbe

Interface for touch probe detection.

For a detailed description see the main chapter "I/O terminals":

▶ Touch probe detection (📖 362)

### 18.2.41 LS_WriteParamList

Writing to a configurable list which contains up to 32 local parameters.

For a detailed description see the main chapter "Parameter change-over". (📖 734)

# 19 Index

## M

## P

## R



## S

FEEDBACK

## Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

If you have suggestions for improvement, please e-mail us to:

feedback-docu@Lenze.de

Thank you for your support.

*Your Lenze documentation team*

Lenze Drives GmbH
Breslauer Straße 3
D-32699 Extertal
Germany

+49 (0) 51 54 / 82-0

Lenze@Lenze.de

www.Lenze.com

Service Lenze Service GmbH
Breslauer Straße 3
D-32699 Extertal
Germany

00 80 00 / 24 4 68 77 (24 h helpline)

+49 (0) 51 54 / 82-11 12

Service@Lenze.de

EDS84AVTCxx ▪ 13387462 ▪ EN 2.0 ▪ TD05

10 9 8 7 6 5 4 3 2 1