Inverter

8400

E84DSxxxx...

Inverter Drives 8400 protec StateLine _____

Reference manual EN

13350162

Overview of technical documentation for Inverter Drives 8400

Project planning, selection & ordering
☐ 8400 hardware manual
🗎 Catalogue

Mounting & wiring
🗎 MA 8400 protec StateLine/HighLine
🗎 MA for the accessories

Parameterisation	
🗎 BA keypad	
☐ SW 8400 protec StateLine	← **This documentation**
☐ SW 8400 protec HighLine	
☐ SW for the integrated safety system (optional)	
☐ KHB for the communication medium used	

Drive commissioning
☐ SW 8400 protec StateLine/HighLine
→ 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/reference 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 protec mounting instructions** and in the **8400 protec hardware manual**. Both documents are supplied with the controller.

Target group

This documentation is intended for all persons who want to parameterise, configure and diagnose the 8400 protec StateLine controller with the engineering software L-force »Engineer« and the X400 keypad.

Validity

The information in this documentation are valid for the following standard devices:

Product series	Type designation	from software version
8400 protec StateLine	E84DSxxxx	01.00

Screenshots/application examples

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 → Download

1.1 Document history

Version			Description
4.2	03/2013	TD05	Conversion to new layout, error corrections & supplements, parameter reference V07.00.00
4.1	02/2011	TD05	Error corrections & supplements
4.0	11/2010	TD05	Extended by new functions for 8400 protec StateLine V06.00.00
3.0	02/2010	TD05 TD06	Restructuring of some chapters, error corrections & supplements, parameter reference V02.00.00
2.0	07/2009	TD03	Complete revision
1.0	04/2009	TD03	First edition (only parameter list)

1.2 Conventions used

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

Type of information	Writing	Examples/notes
Spelling of numbers		
Decimal separator	Point	The decimal point is generally used. For example: 1234.56
Text		
Version info	Blue text colour	All information that only applies to a certain controller software version or higher is identified accordingly in this documentation. Example: This function extension is available from software version V3.0!
Program name	» «	The Lenze »Engineer« PC software ...
Window	*italics*	The *Message* window ... / The *Options* dialog box...
Variable identifier		By setting *bEnable* to TRUE...
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: Use **<Shift>+<ESC>** to...
Hyperlink	Underlined	Optically highlighted reference to another topic. In this documentation activated by mouse-click.
Icons		
Page reference	(14)	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.

All information that only applies to a certain controller software version or higher is identified accordingly in this documentation.

1.3 Terminology used

Term	Meaning
Engineering Tools	Software solutions for simple engineering at all stages
	»EASY Navigator« – Ensures easy operator guidance • All practical Lenze engineering tools at a glance • Tools can be selected quickly • Clearly arranged, simplifying the engineering process from the start
	»EASY Starter« – Simple tool for service technicians • Especially developed for the commissioning and maintenance of Lenze devices • Graphical user interface with few buttons • Simple online diagnostics, parameterisation and commissioning • No risk of accidentally changing the application • Ready applications can be loaded to the device
	»Engineer« – Multi-device engineering • For all products from our L-force portfolio • Practice-oriented user interface • Easy handling due to graphical user interfaces • Suitable for all project stages (configuration, commissioning, production) • Parameter setting and configuration
Application block	Block for a technology application (e.g. actuating drive speed) 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. 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 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 from the "HighLine" version onwards can also be reconfigured and extended by individual functions.
Function block	General designation of a function block for free interconnection (from "HighLine" device version). 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. 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. 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. **Note:** Use the diagnosis terminal for the 8400 protec StateLine controller. The diagnosis terminal combines the keypad with a housing and a connecting cable.
LA	Abbreviation for **L**enze **A**pplication block 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 Example: "LP_CanIn1" (CAN1 port block)
LS	Abbreviation for **L**enze **S**ystem block Example: "LS_DigitalInput" (system block for digital input signals)

Term	Meaning
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
Subcode	If a code contains several parameters, these are stored in "subcodes". This Manual uses a slash "/" as a separator between code and subcode (e.g. "C00118/3"). 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. • 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 documentation 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 gives information about how to prevent dangerous situations)

Pictograph	Signal word	Meaning
	Danger!	**Danger of personal injury through dangerous electrical voltage** 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** 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** 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

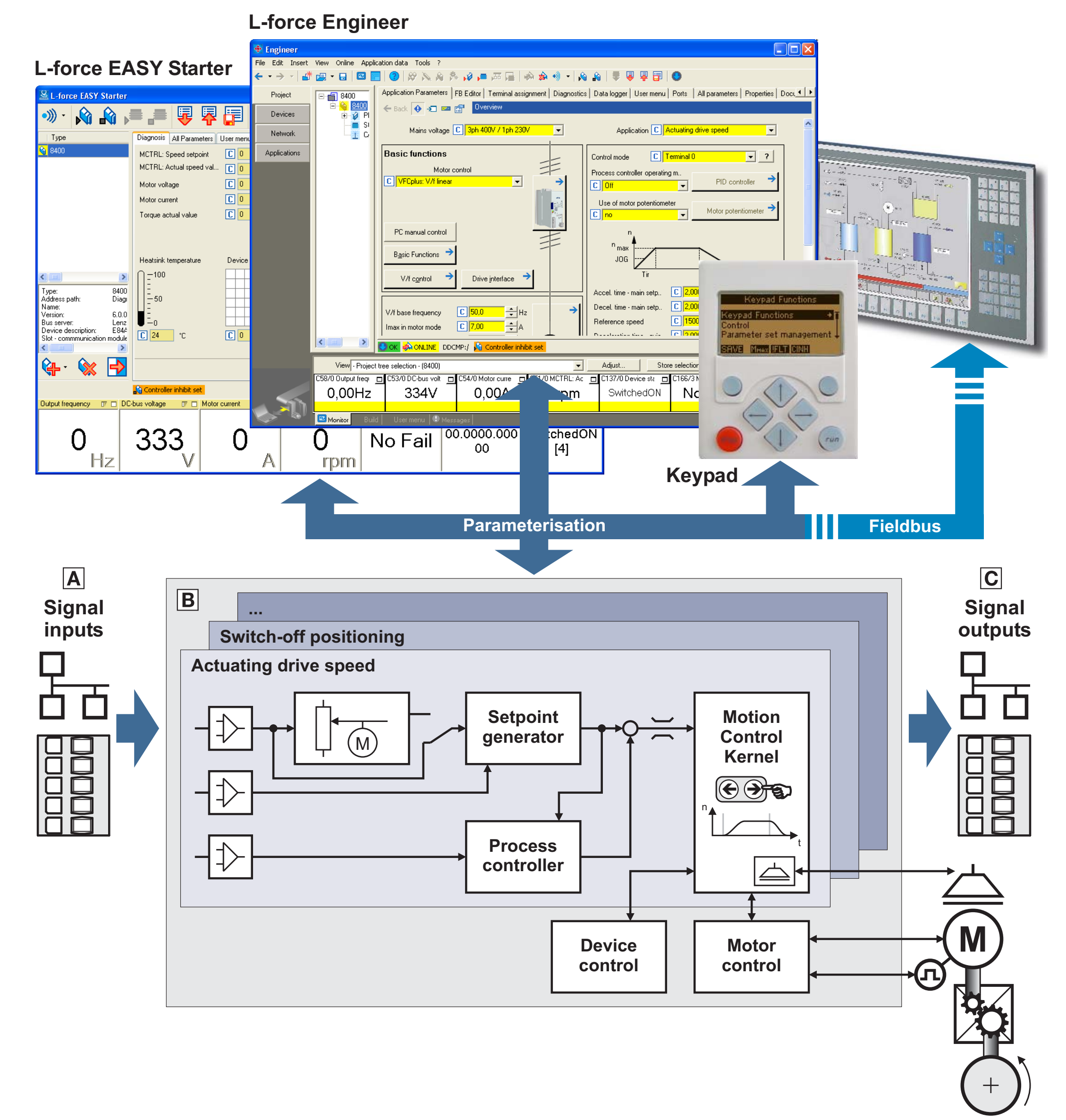
Being a component of a machine which includes a speed-variable drive system, the controller needs to be adjusted to its drive task. The controller is adjusted by changing parameters which are saved in the memory module. The parameters can be accessed by keypad, by the L-force »EASY Starter« or by the L-force »Engineer«. Access is also possible by a master control via fieldbus communication. For this purpose, interfaces X31 (fieldbus input) and X32 (fieldbus output) are available.

Danger!

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

An enabled controller can cause an unwanted behaviour at the motor shaft! For instance, setpoint sources can change over abruptly (e.g. when the signal source is configured 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.



A Signal inputs for control and setpoint signals
B Signal flow of the integrated technology application (see the following subchapter)
C Signal outputs for status and actual value signals

[2-1] Adaptation of the drive solution via parameter setting

2.1 Integrated technology applications

The following technology applications integrated in the drive controller 8400 protec StateLine provide the main signal flow for the implementation of a general or a special drive solution:

Technology application "Actuating drive speed"

This preset 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.

"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.

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.

Detailed information on each technology application can be found in the main chapter "Technology applications". (🕮 234)

2.1.1 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.

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. In practice, some drive tasks are alike so that minor modifications of the corresponding technology applications lead to quick results.

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.

2.1.2 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 solutions of Lenze, it is also possible from the "HighLine" version onwards to implement individual drive solutions by means of the "free interconnection".

Here, a technology application can be used as starting basis, which has to be adapted to the requirement by a change or extension via function block editor (see the following chapter).

2.1.3 Technology application = function block interconnection

In case of the 8400 device version, each technology application is connected to a "function block interconnection" ("FB interconnection"). This FB interconnections serve to implement signal interconnections. Various FBs are available for digital signal processing, signal conversion and logic modules.

For special drive tasks it has proved of value to use the integrated technology applications as a basis for modifications or extensions of the available FB interconnections.

- From "StateLine" version onwards, the preconfigured signal links can be reconfigured in the I/O level by means of parameter setting or using the »Engineer« function block editor.
- Moreover, from the "HighLine" version onwards, the experienced user has the opportunity to implement own drive solutions independent of the predefined technology applications by using the "free interconnection".

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. In the "deeper" application level, the main signal flow is realised in the form of an interconnection of various function and system blocks.

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.

More detailed information:
- A detailed description of the basic functions implemented in the **Motion Control Kernel** can be found in the main chapter "Basic drive functions (MCK)". (🕮 295)
- 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". (🕮 658)
- All available function and system blocks are described in the main chapter "Function library". (🕮 710)

2.2 Selection of the appropriate commissioning tool

There are several possibilities for commissioning the 8400 protec StateLine controller:

Commissioning via keypad X400 (or diagnosis terminal X400)

The keypad is an alternative to the PC for the local operation, parameterisation, and diagnostics in a simple manner. The keypad is especially suited for test and demonstration purposes and for the case that only few parameters have to be adapted.

Note:

- Use the diagnosis terminal for the 8400 protec StateLine controller. The diagnosis terminal combines the keypad with a housing and a connecting cable.
- The description how to make the settings with the keypad also applies to the diagnosis terminal.

Commissioning using PC and »EASY Starter«

The »EASY Starter« is a Lenze tool for simple online diagnostics, parameterisation and commissioning of the controller.

Commissioning using PC and »Engineer«

The »Engineer« is a Lenze engineering software for parameter setting across all devices, configuring and diagnosing individual components (as for instance controllers, industrial PCs, motors, I/O systems) and machine control systems.

Tip!

The engineering tools »EASY Starter« and »Engineer StateLevel« are provided free of charge in the internet:

http://www.Lenze.com → Download → Software downloads

For communication between PC and controller, the USB diagnostic adapter can be used for instance (see the following subchapter).

2.2.1 Overview: Accessories for commissioning

<table>
<tr><th>Version</th><th>Features</th><th>Product key</th></tr>
<tr><td>Diagnosis terminal X400
</td><td>Keypad X400 in a robust housing, also suitable for installation into the control cabinet door.
• Supports hot plugging
• Graphic display with plain texts
• Backlighting
• Easy user guidance
• 4 navigation keys, 2 context-sensitive keys
• Adjustable RUN/STOP function
• Incl. 2.5 m cable
• Enclosure IP20; in case of front installation in control cabinet IP65
• Can be used for L-force Inverter Drives 8400 and Servo Drives 9400</td><td>EZAEBK2001</td></tr>
<tr><td>USB diagnostic adapter
</td><td>For electrical isolation of your PC and the controller.
• Supports hot plugging
• Diagnostic LED for data transfer display
• plug and play
• Input-side voltage supply via USB connection from PC
• Output-side voltage supply via the diagnostic interface of the controller
• Connecting cables can be selected in various lengths:</td><td>E94AZCUS</td></tr>
<tr><td rowspan="3">Connecting cable for USB diagnostic adapter</td><td align="right">2.5 m length</td><td>EWL0070</td></tr>
<tr><td align="right">5 m length</td><td>EWL0071</td></tr>
<tr><td align="right">10 m length</td><td>EWL0072</td></tr>
</table>

2.3 General notes on parameters

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

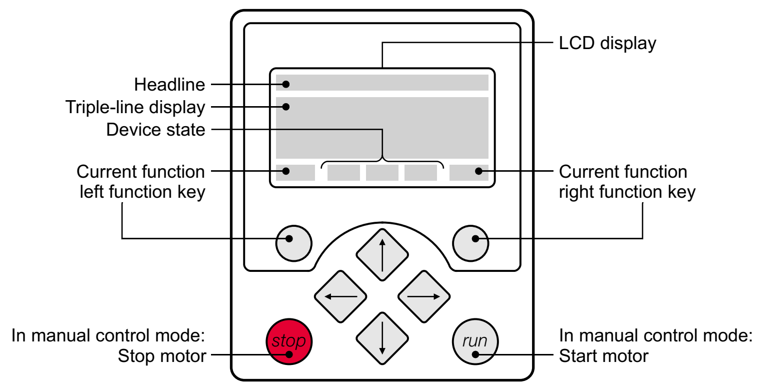
- The codes are numbered and indicated by the prefix "C" before the code, e.g. "C00002".
- Moreover, each code has a name and specific attributes, as for example access type (reading, writing), data type, limit values and default setting ("Lenze setting").
- For the sake of clarity, some codes contain "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:
  - Setting parameters: For specifying setpoints and for setting device / monitoring functions.
  - Configuration parameters: For configuring signal connections and terminal assignments.
  - Diagnostic/display parameters: For displaying device-internal process factors, current actual values and status messages. These are read-only parameters.

2.3.1 Changing the parameterisation with the keypad

Connect the diagnosis terminal with the keypad to the diagnostic interface X70.

- The X70 plug is provided behind the service hatch. To open it, press lightly against the service hatch and push the two plastic bars down. Tools are not required.
- After using the diagnostic interface, remove the connecting cable from X70 and close the service hatch completely.
- The connecting cable can also be connected to the diagnostic interface during operation and removed again.

Keypad display and control elements



LCD display			
Headline			
In the menu level: Menu name In the parameter level: Parameter name			
Three-part display			
In the menu level: List of available menus In the parameter level: Code/subcode and setting or actual value			
Device status			
RDY	Controller is switched on	IMP	Pulse inhibit active
RUN	Controller is enabled	!SFLT	System fault active
CINH	Controller is inhibited	!FLT	"Fault" device status is active
QSP	Quick stop active	!TRB	"Trouble" device status is active
Imax	Current limit exceeded	!TQSP	"TroubleQSP" device status is active
Mmax	Speed controller 1 in the limitation	WRN	A warning is indicated
Function - left function key		**Function - right function key**	
EDIT	Change parameter setting (change to editing mode)	OK	Accept change in the controller (no saving with mains failure protection → SAVE)
	Back to main menu	ESC	Abort (discard change)
CINH!!	Parameter can only be changed when the controller is inhibited.		
EDIT	Change parameter setting (change to editing mode)	OK	Accept change in the controller (no saving with mains failure protection → SAVE)

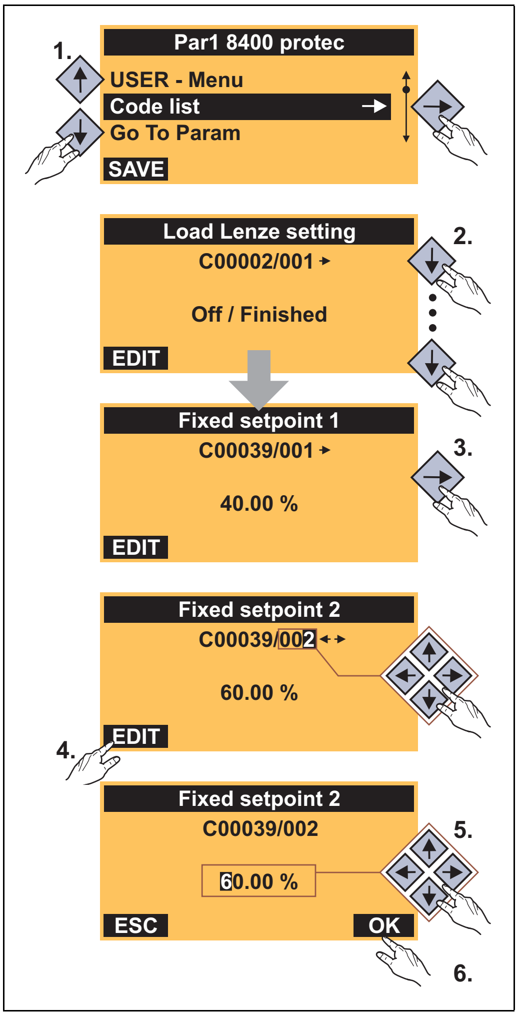
Control elements	
○	Execute the function assigned to the function key (see LCD display)
stop	Execute the stop function set in C00469 (Lenze setting: Inhibit controller)
run	Deactivate stop function again (Lenze setting: Enable controller again)
↑ ↓	In the menu level: Select menu/submenu In the parameter level: Select parameter In the editing mode: Change marked digits or select list entry
CINH	In the menu level: Select submenu/change to parameter level In the editing mode: Cursor to the right
RDY	In the menu level: One menu level higher (if available) In the parameter level: Back to the menu level In the editing mode: Cursor to the left

Menu structure

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

- The **USER menu** includes a selection of frequently used parameters.
- 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.
- The **Diagnostics** menu contains diagnostic/display parameters for displaying device-internal process factors, current actual values and status messages.

General operation



[2-2] Example: Changing parameters with the keypad

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. In order to select another subcode in case of a parameter with subcodes:
   - Press the navigation key ⇨ to change tot he editing mode for the subcode.
   - Use the navigation keys to set the desired subcode.
4. Use the EDIT function key to switch over to the editing mode.
5. Use the navigation keys to set the desired value.
6. 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.

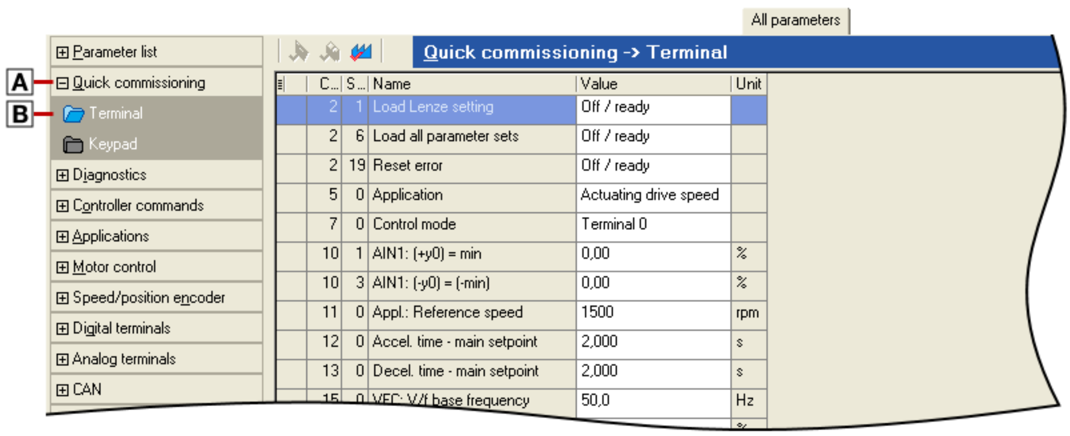
2.3.2 Change parameter settings with PC and Lenze software

The USB diagnostic adapter, for instance, can be used for the communication between the PC (including the L-force »EASY Starter« or L-force »Engineer« software) and the controller (see the following illustration). The USB diagnostic adapter is the connection between the PC (free USB port) and the controller (diagnostic interface X70).

[2-3] Exemplary constellation for parameterising the controller

The **All parameters** tab in the »EASY Starter« and the »Engineer« provides a quick access to all parameters of the controller.

The given categories and subcategories correspond 1:1 to the menus and submenus of the keypad:



A Category
B Subcategories

[2-4] **All parameters** tab in the »Engineer«

Moreover, the »Engineer« provides a commissioning interface on the **Application parameters** tab where you can commission the application in a few steps.

Detailed information on how to handle the »Engineer« can be found in the integrated online help that you can call with the **[F1]** function key.

2.3.3 Save parameter settings in the memory module safe against mains failure

Controller parameter changes via the EASY Starter /»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 integrated memory module.

General information

- In the delivery state, the Lenze setting of the parameters has been saved to the integrated memory module. These parameters are
  - 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 the device or the external 24 V voltage supply is switched on, all parameters are automatically loaded from the memory module into the main memory of the controller.
- 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 X10/24E terminal.
- The memory module can be preconfigured with customised data.
- The memory module is available as a spare part - without any data.

During operation

- Parameter sets can be saved and loaded manually.
- Using the keypad, you can press function key SAVE to save the parameter settings.
- The »EASY Starter«/»Engineer« serves to execute the saving via the icon in the *toolbar* or via the device command "Save all parameter sets" (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.

Note!

In order to prevent data inconsistencies during the saving process:
- Do not switch off the supply voltage!
- Do not remove the memory module from the device!

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

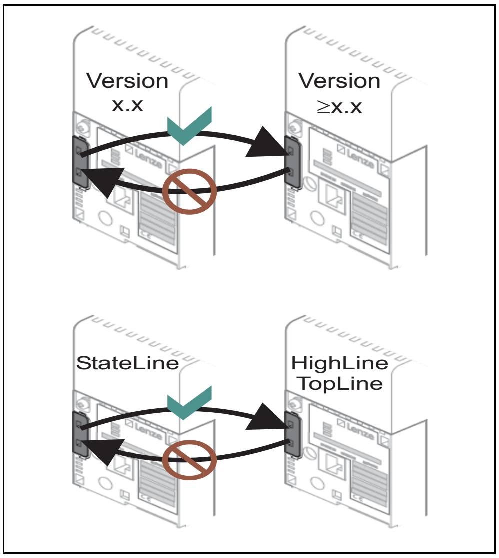
Unplugging the memory module

The memory module is hot-pluggable. A removal during operation causes a warning "PS01: No memory module" and should thus be avoided. The behaviour of the device, however, remains unchanged as all parameters are available in the RAM memory after the device has been started. The device can also be parameterised when the memory module has been unplugged. In this case, the parameter sets cannot be saved in the memory module.

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 can only be processed on new devices with the same or higher device version (downward compatibility).
- 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!
- If the parameter set saved in the memory module is not compatible with the standard device, the "PS03: Par.set device invalid" is output.

2.3.4 User menu for quick access to frequently used parameters

When a system is installed, parameters must be changed time and again until the system runs satisfactorily. The user menu of the controller contains a selection of frequently used parameters to be able to access and change these parameters quickly:

Parameter	Name	Lenze setting
C00051	MCTRL: Actual speed value	-
C00053	DC-bus voltage	-
C00054	Motor current	-
C00061	Heatsink temperature	-
C00137	Device status	-
C00166/3	Mess. - status det. error	-
C00011	Appl.: Reference speed	1500 rpm
C00039/1	Fixed setpoint 1	40.00 %
C00039/2	Fixed setpoint 2	60.00 %
C00012	Acceleration time - main setpoint	2.000 s
C00013	Deceleration time - main setpoint	2.000 s
C00015	VFC: V/f base frequency	50 Hz
C00016	VFC: Vmin boost	1.60 %
C00022	Imax in motor mode	depending on the device power
C00120	Setting of motor overload (I^2xt)	100.00 %
C00087	Rated motor speed	1460 rpm
C00099	Firmware version	-
C00200	Firmware product type	-
C00105	Decel. time - quick stop	2.000 s
C00173	Mains voltage	0: "3ph 400V / 1ph 230V"
Highlighted in grey = display parameter		

Tip!

The user menu can be freely configured in C00517.

In the »Engineer«, you can configure the user menu comfortably via the **User menu** tab (see »Engineer« online help).

The password protection serves to restrict the access to parameters of the user menu. Then, all other parameters cannot be accessed without knowing the password and are thus protected against unwanted changes.

2.4 Device access protection

This function extension is available from version 06.00.00!

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

- Password protection
  - Only authorised persons (with password knowledge) may read/change all parameters of the controller.
  - Non-authorised persons (without password knowledge) can only access the max. 32 parameters of the user menu.
- Device personalisation
  - Only controller personalised with a specific binding ID and memory modules can be used in the system.

Note!

If password protection/device personalisation is used:
- Inform the end customer that Lenze can only provide restricted service for the devices with access protection.
- It is not possible for Lenze to modify a replacement device via special accesses in such a way that it cooperates 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

When the password protection is active, only write/read access to the parameters of the user menu is possible. An option for configuring various protection functions for each individual communication channel is currently being prepared.

- The following describes how to set/check/delete a password by means of the parameters relevant for these functions.
- From »Engineer« V2.14 onwards. these functions can also be executed via dialog (menu command **Online → Set/check/delete password**).

Short overview of the relevant parameters for password protection:

Parameter	Info		Lenze setting
C00505/3	Password • The password may have a maximum length of 16 characters. • The following characters are permitted for the password: Small letters (a - z), capital letters (A - Z), digits (0 - 9) **Note:** After the execution of one of the device commands listed below, this parameter provides the current password status:		
	OFF	No password is set, password protection is not active (Lenze delivery status).	
	ON	Password is set, password protection is active. • This status is also displayed if checking/deleting the password has not been successful due to an invalid entry.	
	ok	Password is set, password protection is not active. • The password protection is temporarily deactivated.	
Device commands Before the following device commands are executed, enter the corresponding password in C00505/3.			
C00002/31	Set password ▸ Activate the password protection		0: Off / ready
C00002/32	Check password ▸ Temporarily deactivate the password protection		0: Off / ready
C00002/33	Delete password ▸ Deactivate password protection/change password		0: Off / ready
Status displays			
C00003	Status of the last device command		-
C00507/1	Password protection - all communication channels • 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			

Activate the password protection

The password protection is activated by setting a password.

How to set a password:

1. Enter the desired password in C00505/3.
   - The password may have a maximum length of 16 characters.
   - The following characters are permitted for the password:
     Small letters (a - z), capital 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.

Temporarily deactivate the password protection

The "Check password" device command serves to temporarily deactivate the password protection in order to execute password-protected functions.

- The password protection remains deactivated until
  - an invalid password will be entered and checked
    - or -
  - the external 24-V supply of the control electronics is switched off (< 19 V).

How to temporarily deactivate the 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.

Deactivate password protection/change password

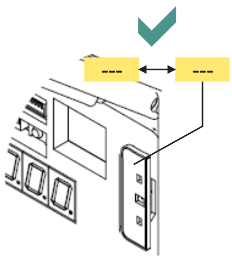
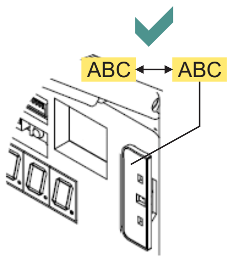
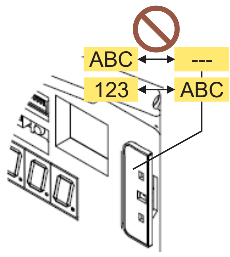
The password protection is simply activated by deleting the set password. If you want to change the set password, first delete the set password as well. Then set the new password.

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 controller and the memory module are married via the device personalisation by means of a binding ID. When the device personalisation is active, all write/read actions between controller and memory module are only executed if both components have the same binding ID.

<table>
<tr><th>Lenze delivery status:</th><th>Procedure carried out by the customer:</th><th>Impermissible replacement by the end user:</th></tr>
<tr><td>
</td><td>
</td><td>
</td></tr>
<tr><td>No binding ID is set.</td><td>Customer sets binding ID for device personalisation.</td><td>When device personalisation is active:
The replacement of the controller or memory module causes an error message if the binding ID is incorrect or not available.</td></tr>
</table>

If, for instance, a parameter set has been loaded from the memory module with an active device personalisation, saving of this parameter set on another memory module with a different or non-existent binding ID is not possible.

- Thus, copying the parameter set from a personalised memory module to a non-personalised memory module is not possible!

Two types of checks are distinguished:

- If a differing binding ID is detected when the controller is switched on (during device initialisation):
  - the "Fault" error response is returned.
  - The "PS10: Invalid memory module binding" error message is entered into the logbook.
- If a differing binding ID is detected while a device command for loading/saving the parameter set is executed:
  - Loading/saving is not executed.
  - 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 • The binding ID may have a maximum length of 16 characters. • The following characters are permitted for the binding ID: Small letters (a - z), capital letters (A - Z), digits (0 - 9) **Note:** After the execution of one of the device commands listed below, this parameter provides the current binding ID status:		
	OFF	No binding ID is set.	
	ON	Binding ID is set.	
Device commands Before the following device commands are executed, enter the corresponding binding ID in C00505/2.			
Highlighted in grey = display parameter			

Parameter	Info		Lenze setting
C00002/29	Set binding ID ▸ Activate device personalisation		0: Off / ready
C00002/30	Delete binding ID ▸ Deactivate device personalisation/change binding ID		0: Off / ready
Status displays			
C00003	Status of the last device command		-
C00507/1	Password protection - all communication channels • 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			

Activate device personalisation

The device personalisation is activated by setting a binding ID.

How to set the binding ID:

1. Enter the desired binding ID in C00505/2.
   - The binding ID may have a maximum length of 16 characters.
   - The following characters are permitted for the binding ID:
     Small letters (a - z), capital 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.

Deactivate device personalisation/change binding ID

The device personalisation is simply deactivated by deleting the set binding ID. If you want to change the set binding ID, first delete the set binding ID as well. Then set the new binding ID.

How to delete the binding ID:

1. Enter the set binding ID in C00505/2.
   - If controller and memory module do not have the same binding ID, enter the binding ID of the memory module to delete the binding ID 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 called "MasterPin". By entering the MasterPin, a controller inhibited by the password mechanisms can be reset to the delivery status.

Stop!

When the MasterPin is used, the parameter set is reset to the Lenze setting both in the controller and in the memory module!

- This results in a permanent loss of the customised parameterisation that must be recreated!
- A reset to the Lenze setting can result in unforeseen level changes at the I/O terminals (e.g. brake control)!

How to restore 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 MasterPin comprises the last 6 digits of the serial number of the memory module.
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.
- Remove the motor plug X21. An active motor temperature monitoring prevents the output of a motor voltage. If monitoring is deactivated, a voltage can be applied to the plug.
- 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. (🕮 323)
- **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 (🕮 72)

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 protec hardware manual**.

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

3.1 Safety instructions with regard to commissioning

General safety instructions

In order to prevent 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
  - whether all connectors are locked correctly in order to ensure trouble-free operation.
  - whether all connectors of the control terminals and interfaces that are not used are closed with the supplied plastic caps in order to maintain the certified product features of the safety system.
- 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.
  - The 8400 protec controllers are provided with a factory set automatic brake operation that operates the holding brake. If no setpoint is set, the motor is locked by the holding brake.

Safety instructions with regard to motor operation

Danger!

- For thermal reasons, continuous operation of self-ventilated motors at a low field frequency and rated motor current is not permissible!
  - In the Lenze setting, the Motor temperature monitoring (PTC) is activated. (📖 183)
  - Activate the Brake resistor monitoring (I2xt) if necessary. (📖 184)
- C00015 must be used to select 87 Hz operation if a delta-connected asynchronous motor (nameplate data: 400 V ⅄ /230 V △) is to be operated in conjunction with a drive controller for a mains voltage of 400 V.

3.2 Commissioning notes for external 24 V supply

For devices with an external 24 V supply at terminal X10/24E, the following sequence is to be observed during commissioning:

Switching on

1. Connection of the external 24 V supply
   - Control electronics and fieldbus communication start to operate.
   - The integrated 7-segment display shows the message "LU" (undervoltage in the DC bus).
2. Connection of the 400 V mains voltage
   - The message "LU" goes off.
   - The integrated 7-segment display changes to the automatic display, i. e. the parameter saved as status value 1 is displayed. (Lenze setting: motor output frequency in [Hz])

Switching off

1. Disconnection of the 400 V mains voltage
2. Disconnection of the external 24 V supply

Note!

When the 24 V supply is disconnected, the control electronics become inoperable. The switch function of Ethernet-based fieldbuses is also inoperable.

The disconnection of the 24 V supply in the case of an available 400 V mains voltage can cause an error status for master controls.

Related topics:

▶ LED status displays (📖 323)
▶ Drive diagnostics via the integrated 7-segment display (📖 326)

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 immediately ready for operation 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". (📖 99)

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) (📖 98)

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 X70 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

When the »Engineer« is used, trouble during commissioning can be detected and eliminated conveniently. Proceed as follows:

- Check whether error messages appear 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 (📖 322)
▶ LED status displays (📖 323)
▶ Drive diagnostics via the integrated 7-segment display (📖 326)
▶ Error messages of the operating system (📖 347)

3.6 Commissioning wizard 8400

This function extension is supported by the »Engineer« from version 2.15 onwards!

The **8400 commissioning wizard** provides for a guided commissioning of the controller based on the Lenze setting of the parameters. Then, the parameter settings made can be stored save against mains failure in the controller.

Note!

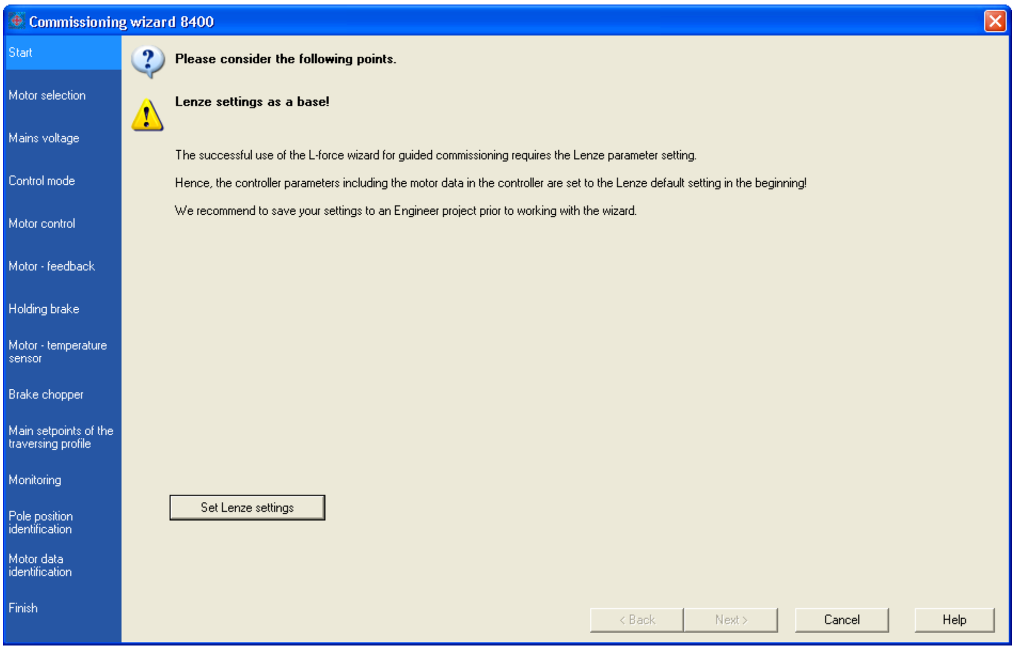
Take all the necessary safety precautions before you carry out the following commissioning steps and switch the device on!

▶ Safety instructions with regard to commissioning (📖 40)

How to carry out a guided commissioning using the »Engineer«:

1. In the *Project view*, select the 8400 protec StateLine controller.
2. Go online.
   - After a connection to the controller has been established, the following status is displayed in the *Status line*:

     ONLINE
3. Click the icon to open the *commissioning wizard 8400* dialog box.
   - Now the commissioning wizard guides you step by step through the setting of the important parameters for a quick commissioning.
   - The **Next** button can only be activated again after all parameter settings in the device have been reset via the **Load Lenze setting** button.



Related topics:

▶ Commissioning of the "Actuating drive speed" technology application (📖 46)

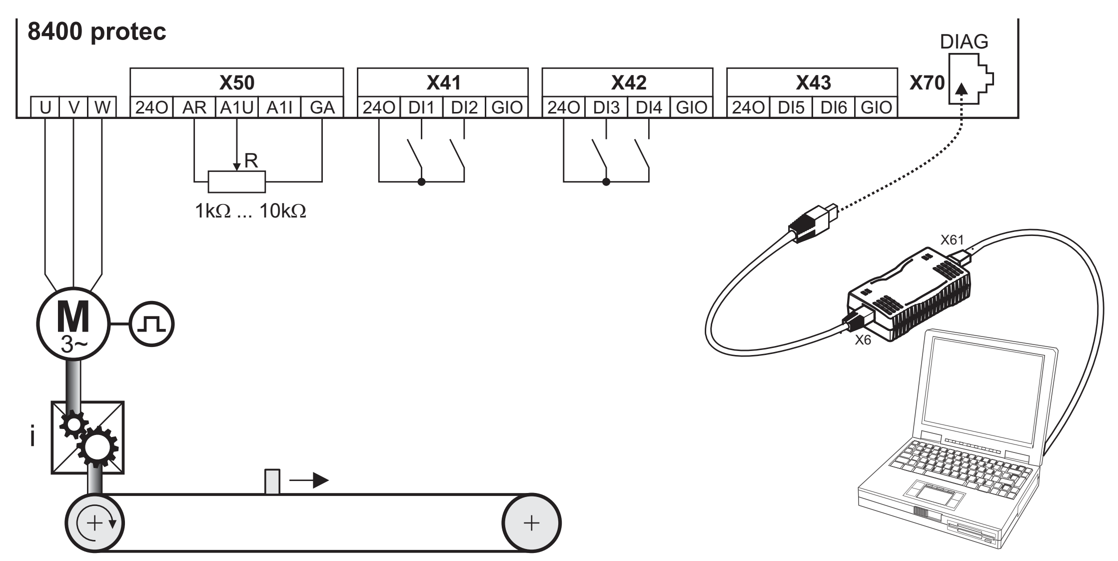
3.7 Commissioning of the "Actuating drive speed" technology application

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 (📖 40)

System constellation



X50/AR	Reference voltage (10 V) for analog signals
X50/A1U	Speed setpoint selection (slider of setpoint potentiometer R) • Scaling: 10 V ≡ 100 % ≡ 1500 rpm[1] (for a 4-pole motor)
X50/GA	Ground potential (GND) for analog signals
X70	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
▶ Creating an »Engineer« project & going online (📖 48)
▶ Parameterising the motor control (📖 49)
▶ Parameterise application (📖 50)
▶ Saving parameter settings safe against mains failure (📖 52)
▶ Enable controller and test application (📖 52)

3.7.1 Prepare controller for commissioning

1. Wire the power connections
   - 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:

	Assignment of the digital terminals			
Control mode	**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. Connect USB diagnostic adapter.

Danger!

For the commissioning with motor operation observe that no setpoint is applied before connecting the mains voltage.

In the Lenze setting, the "Inhibit at power-on" auto-start option has been deactivated in C00142, i.e. the motor can directly start up if the controller is enabled after mains connection!

4. 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:

▶ Commissioning notes for external 24 V supply (📖 41)
▶ Automatic restart after mains connection/trouble... (📖 90)
▶ LED status displays (📖 323)

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.

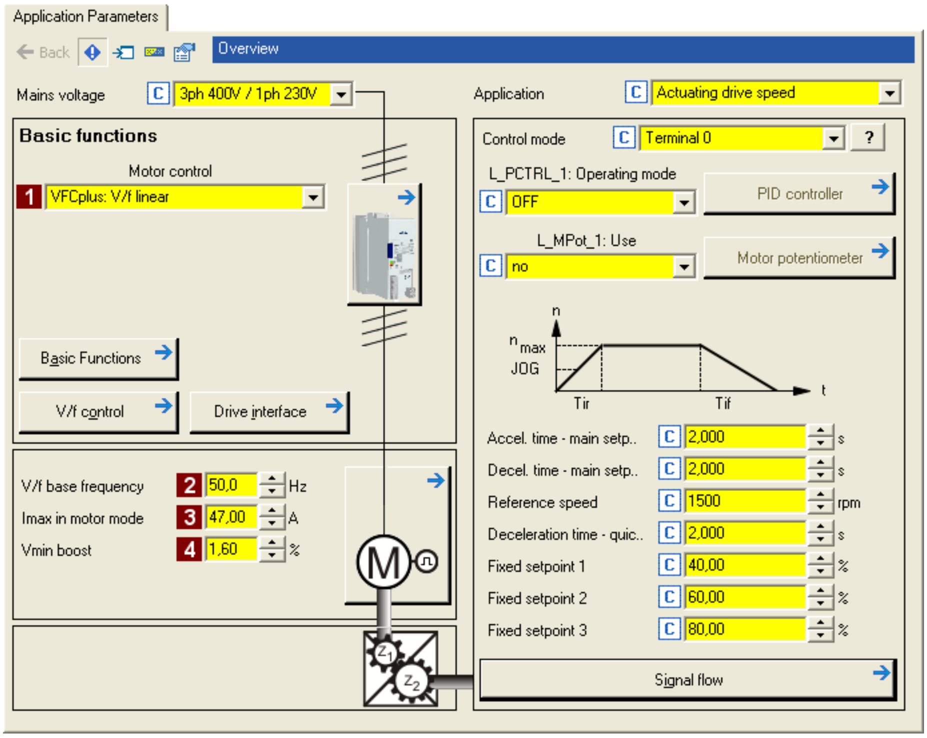
The following steps serve to describe a general method for creating a project with the **Select component from catalogue** option. For this purpose, individual components (controller, motor, etc.) are selected from selection lists.

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 protec StateLine controller.
   - Select the available communication and safety system in the **Device modules** dialog step.
   - 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*:

4. If the controller is enabled: Inhibit controller.
5. Transfer parameter set to the device.
   - This command serves to overwrite the current parameter settings in the controller by parameter settings of the »Engineer« project.

3.7.3 Parameterising the motor control

1. Select the **Application parameters** tab from the *Workspace*.
   - 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:

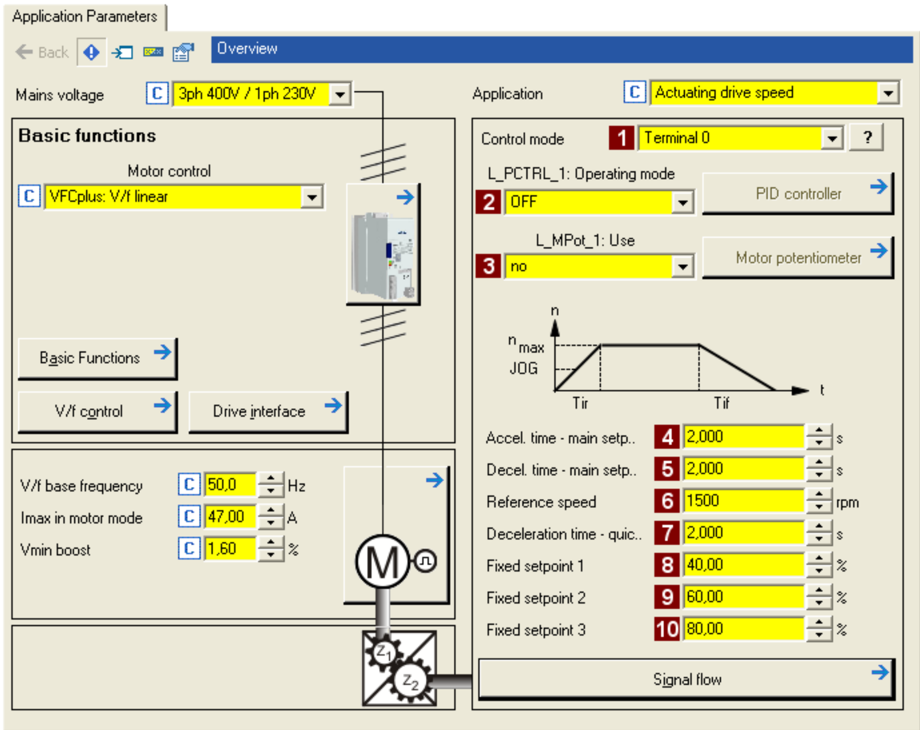
Parameter		Lenze setting Value	Unit	Info
2	**V/f base frequency** (C00015)	50.0	Hz	▶ Adapting the V/f base frequency (📖 120)
3	**Imax in motor mode** (C00022)	47.00	A	▶ Optimising the Imax controller (📖 122)
4	**Vmin boost** (C00016)	1.60	%	▶ Adapting the Vmin boost (📖 121)

Related topics:

▶ Notes on motor control (📖 42)
▶ Motor control (MCTRL) (📖 98)

3.7.4 Parameterise 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". (📖 247)
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. (📖 768)
   - 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. (📖 741)
   - 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 (📖 749)
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 (📖 76)
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 (📖 749)
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". (📖 237)
- The preconfigured I/O connection in the selected control mode can be changed via configuration parameters. See chapter "User-defined terminal assignment". (📖 227)

More detailed information on the technology application:

▸ TA "Actuating drive speed" (📖 236)
▸ Internal interfaces | application block "LA_NCtrl" (📖 239)
▸ Process data assignment for fieldbus communication (📖 256)
▸ Terminal assignment of the control modes (📖 247)
▸ Setting parameters (short overview) (📖 258)
▸ Configuration parameters (📖 260)

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!

1. Enable controller.
   - 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 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 • Scaling: 10 V ≡ 100 % ≡ reference speed (C00011)
HIGH	LOW	Fixed setpoint 1 (C00039/1) is used as setpoint speed. • Lenze setting: 40 % of the reference speed (C00011)
LOW	HIGH	Fixed setpoint 2 (C00039/2) is used as setpoint speed. • Lenze setting: 60 % of the reference speed (C00011)
HIGH	HIGH	Fixed setpoint 3 (C00039/3) is used as setpoint speed. • Lenze setting: 80 % of the reference speed (C00011)

Note!

Observe the actual speed value (display in C00051) as well as the LED status displays. (📖 323)

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 (📖 90)
- Trouble-shooting during commissioning (📖 44)
- Diagnostics & error management (📖 322)

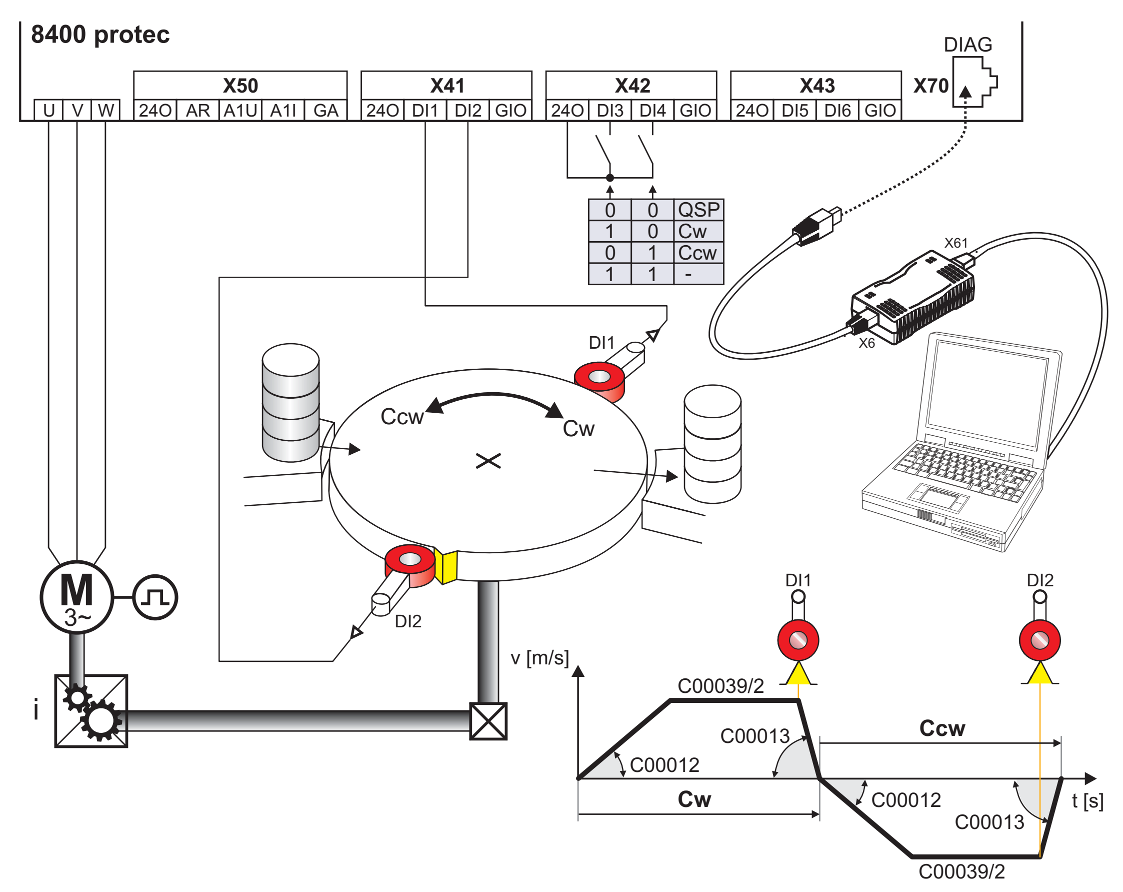
3.8 Commissioning of the "Switch-off positioning" technology application

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 (🕮 40)

System constellation



X41/DI1	Initiator connection for stop function 1
X41/DI2	Initiator connection for stop function 2
X42/DI3	Quick stop for CW rotation / selection of switch-off position 1
X42/DI4	Quick stop for CCW rotation / selection of switch-off position 2
X70	Diagnostic interface for connecting the USB diagnostic adapter

[3-2] 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 the DI2 contact has been reached, the drive is braked to standstill with quick stop (QSP) in the initial position.

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". (🕮 274)

Commissioning steps

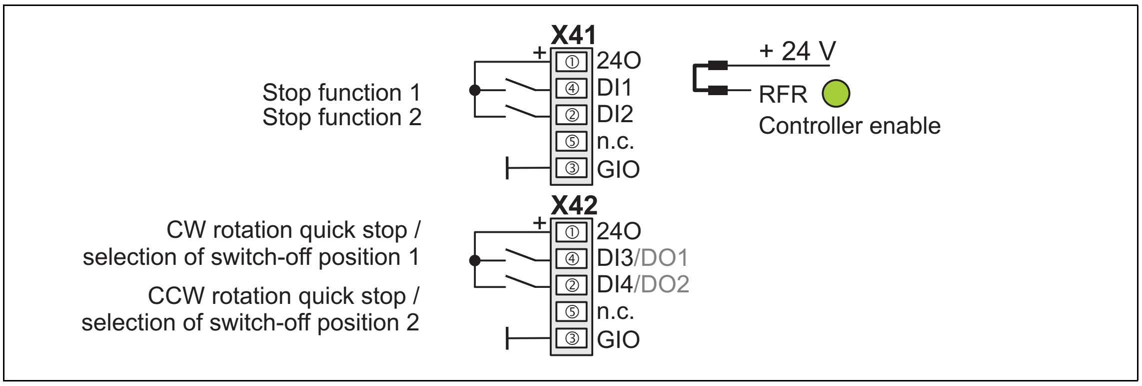
As shown in illustration [3-2], below find a description of the commissioning steps of the "Switch-off positioning" application without pre-switch off.

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:

- Prepare controller for commissioning (🕮 56)
- Creating an »Engineer« project & going online (🕮 57)
- Parameterising the motor control (🕮 58)
- Parameterise application (🕮 59)
- Saving parameter settings safe against mains failure (🕮 61)
- Enable controller and test application (🕮 61)

3.8.1 Prepare controller for commissioning

1. Wire the power connections
   - 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 [3-2], switch-off positioning without pre-switch off, wiring according to the "Terminals 2" control mode makes sense:



3. Connect USB diagnostic adapter.

Danger!

For the commissioning with motor operation observe that no setpoint is applied before connecting the mains voltage.

In the Lenze setting, the "Inhibit at power-on" auto-start option has been deactivated in C00142, i.e. the motor can directly start up if the controller is enabled after mains connection!

4. 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:

- Commissioning notes for external 24 V supply (📖 41)
- Automatic restart after mains connection/trouble... (📖 90)
- LED status displays (📖 323)

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.

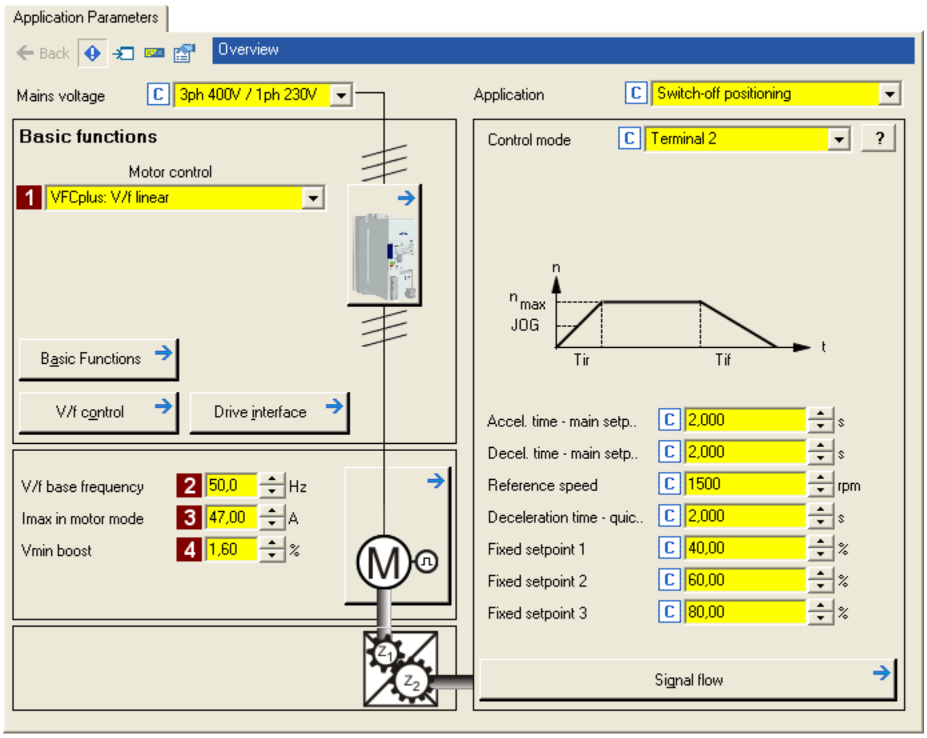
The following steps serve to describe a general method for creating a project with the **Select component from catalogue** option. For this purpose, individual components (controller, motor, etc.) are selected from selection lists.

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 protec StateLine controller.
   - Select the available communication and safety system in the **Device modules** dialog step.
   - 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. If the controller is enabled: Inhibit controller.
5. Transfer parameter set to the device.
   - This command serves to overwrite the current parameter settings in the controller by parameter settings of the »Engineer« project.

3.8.3 Parameterising the motor control

1. Select the **Application parameters** tab from the *Workspace*.
   - 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:

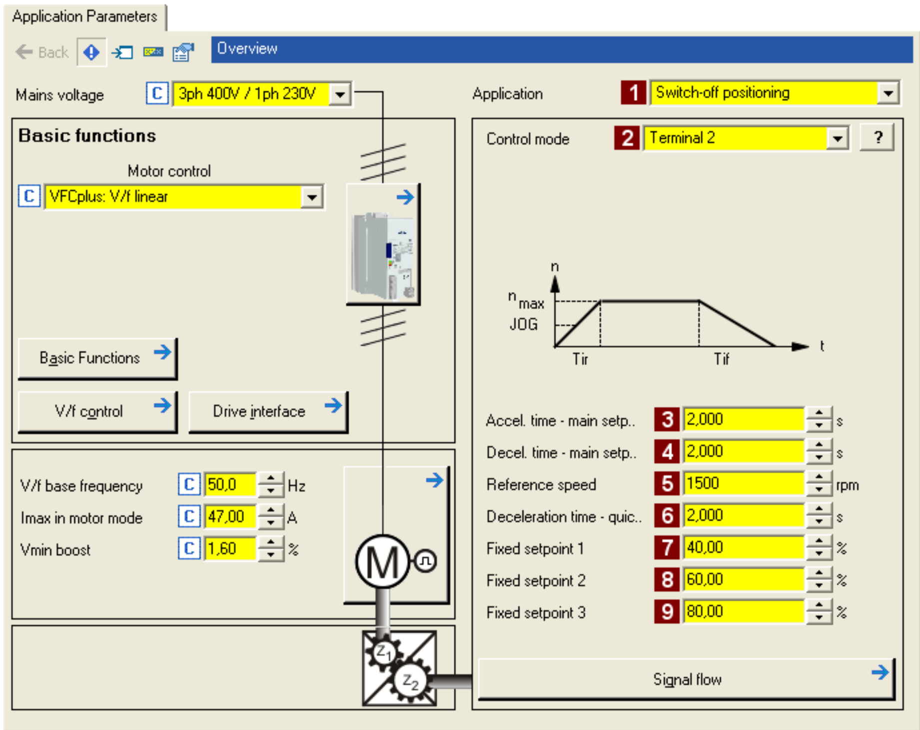
Parameter		Lenze setting		Info
		Value	Unit	
2	**V/f base frequency** (C00015)	50.0	Hz	Adapting the V/f base frequency (📖 120)
3	**Imax in motor mode** (C00022)	47.00	A	Optimising the Imax controller (📖 122)
4	**Vmin boost** (C00016)	1.60	%	Adapting the Vmin boost (📖 121)

Related topics:

- Notes on motor control (📖 42)
- Motor control (MCTRL) (📖 98)

3.8.4 Parameterise 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 [3-2], 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". (📖 247)

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 (📖 76)
7	**Fixed setpoint 1** (C00039/1)	40.00	%	Fixed setpoints are selected in [%] based on the reference speed (C00011). **Fixed setpoint 2 must be smaller than fixed setpoint 3!** Otherwise, the drive will be started with a low speed and accelerated 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". (📖 265)
- The preconfigured I/O connection in the selected control mode can be changed via configuration parameters. See chapter "User-defined terminal assignment". (📖 227)
- 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" (📖 263)
▸ Internal interfaces | application block "LA_SwitchPos" (📖 266)
▸ Process data assignment for fieldbus communication (📖 283)
▸ Terminal assignment of the control modes (📖 274)
▸ Setting parameters (short overview) (📖 285)
▸ Configuration parameters (📖 287)

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!

1. Enable controller
   - 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. (📖 323)

Related topics:

▶ "Inhibit at power-on" auto-start option (📖 90)
▶ Trouble-shooting during commissioning (📖 44)
▶ Diagnostics & error management (📖 322)

3.9 PC manual control

This function extension is available from version 06.00.00 onwards and is supported by the »Engineer« from version 2.13 onwards!

For the purpose of testing and demonstration and when an online connection has been established, the PC manual control enables the manual control of various drive functions from the »Engineer«.

Supported drive functions:

- Speed control (follow speed setpoint)
- Activate/deactivate quick stop

More control functions:

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

Diagnostic functions:

- Display of the actual speed value and motor current (as time 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 Activate PC manual control

STOP **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!

With active PC manual control:

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

- If the online connection is interrupted longer than the timeout time set (Lenze setting: 2 s):
  - The error response "Fault" takes place, i.e. the motor becomes torqueless and coasts, if it has not yet been in standstill.
  - The "Ck16: Time overflow manual control" error message is entered into the logbook.

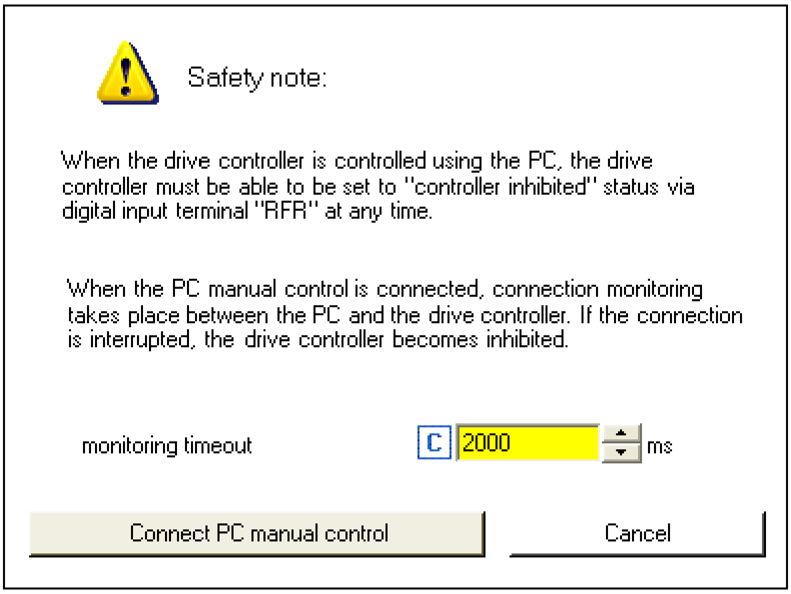
PC manual control provides the **Motion Control Kernel** and the motor interface with all required control signals and setpoint signals.

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

How to activate the PC manual control:

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

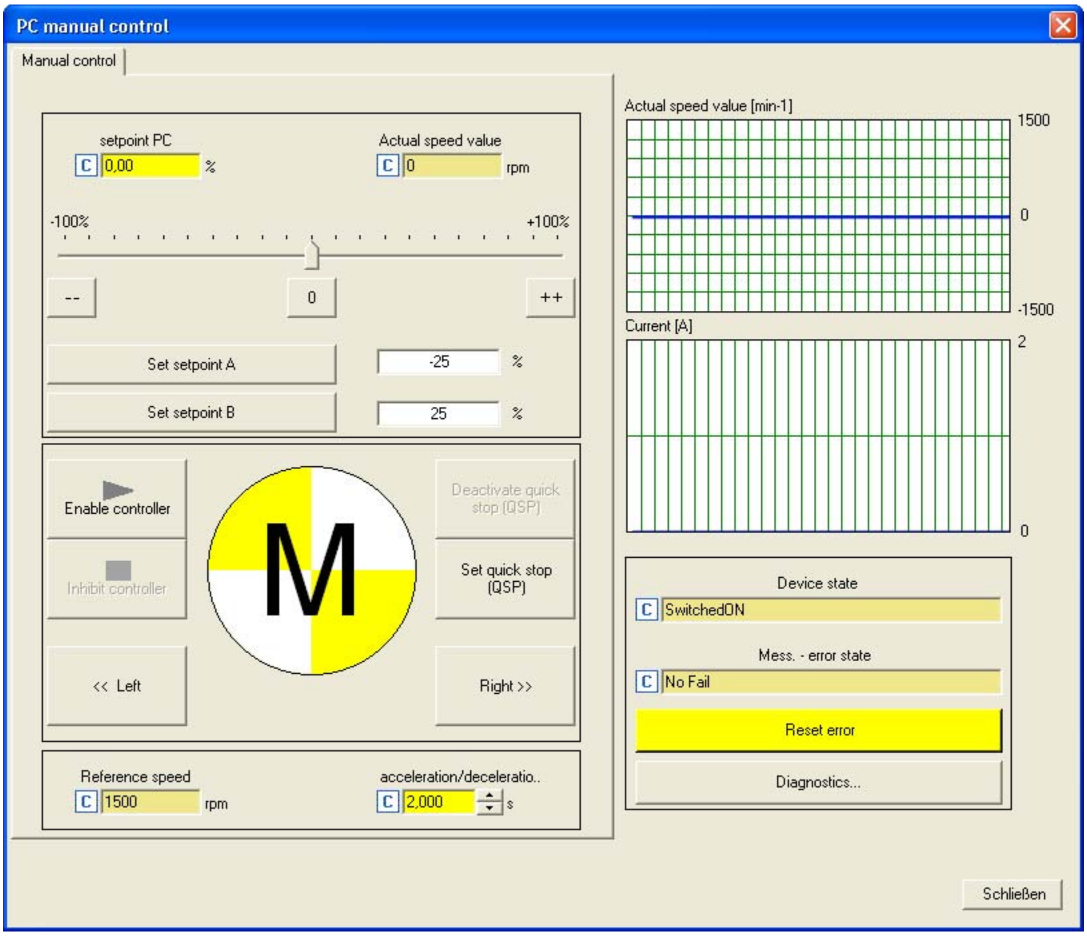
   Go online.
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the **PC manual control** button.
   - First, the following safety note is displayed:



   - 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 **Connect 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 control elements which serve to select various control functions. On the right-hand side, setpoint and status displays are provided for diagnostic purposes:



Note!

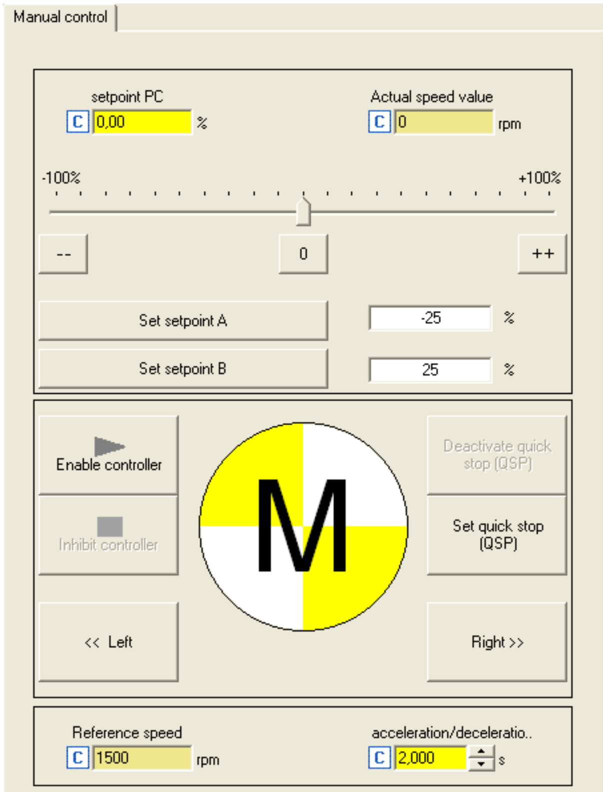
PC manual control can be exited any time by clicking the **Close** button.

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 different versions are 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 easily rotate the motor:

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 note that the controller is only enabled if no other sources of a controller inhibit (e.g. terminal RFR) are active.
   - The enabled drive now follows the defined speed setpoint.
   - In order to prevent shocks or overload at higher setpoint changes, the speed setpoint is lead via 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 **<< Left** and **Right >>** buttons, the direction of rotation can be changed.

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. (📖 325)
  - As text messages in the Logbook. (📖 335)
  - As process signals via the outputs of the LS_DriveInterface system block. (📖 93)
  - Via diagnostic / display parameters which are included in the »Engineer« parameter list as well as in the **Diagnostics** category in the keypad.

Note!

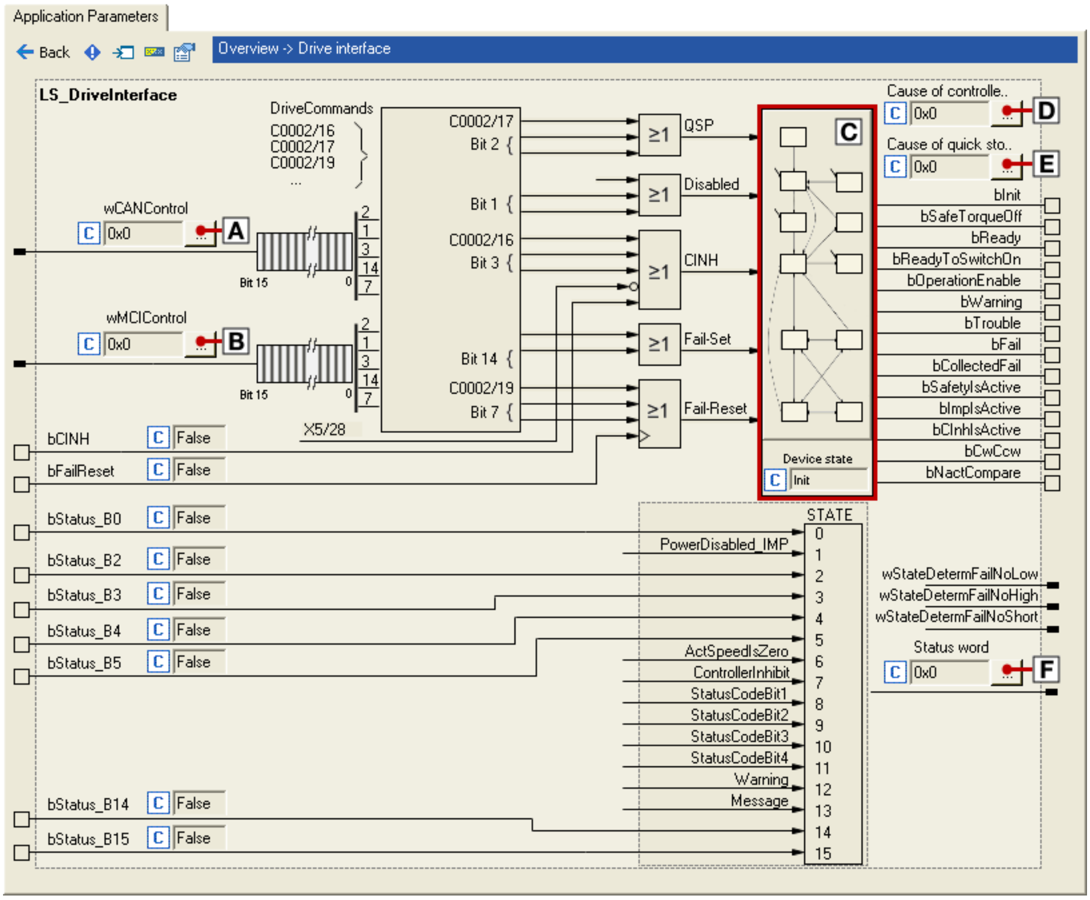
The device statuses of the controller are based on the operating statuses of the CiA402 standard. ▶ Device state machine and device states (📖 79)

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

1. »Engineer« Go to the *Project view* and select the 8400 protec StateLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
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:

Icon	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 a system module is selected in the *Project view*, the corresponding device command will be activated in all lower-level controllers having an online connection with 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 (from version 06.00.00)		static

Device commands described in other chapters:

C00002 Subcode:	Device command	Controller inhibit required	Status information
23	Motor parameter identification ▸ Automatic motor data identification	●	dynamic
26	CAN reset node ▸ Reinitialising the CANopen interface		static
28	Check MasterPin (from version 06.00.00) ▸ Unlocking the controller with a MasterPin	●	static
29	Set binding ID (from version 06.00.00) ▸ Device personalisation		static
30	Delete binding ID (from version 06.00.00) ▸ Device personalisation		static
31	Set password (from version 06.00.00) ▸ Password protection		static
32	Check password (from version 06.00.00) ▸ Password protection		static
33	Delete password (from version 06.00.00) ▸ Password protection		static

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 made since the last saving of the parameter set will get 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 made since the last saving of the parameter set will get 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.

The basic function Parameter change-over provides a change-over between four sets with different parameter values for up to 32 freely selectable parameters. (📖 459)

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 icon in the *Toolbar*.
- 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 (📖 152)
  - 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 is set if the auto-DCB function has been activated via C00019.

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 no longer pending.

- 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. (🕮 322)

4.1.7 Delete logbook

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

Tip!

To display the logbook in the »Engineer«, click the **Logbook** button on the **Diagnostics** tab.

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. (🕮 322)

4.1.8 Device search function

This function extension is available from version 06.00.00!

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, the 7-segment display of the controller indicates "HELLO". Afterwards, the function is turned off again 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)

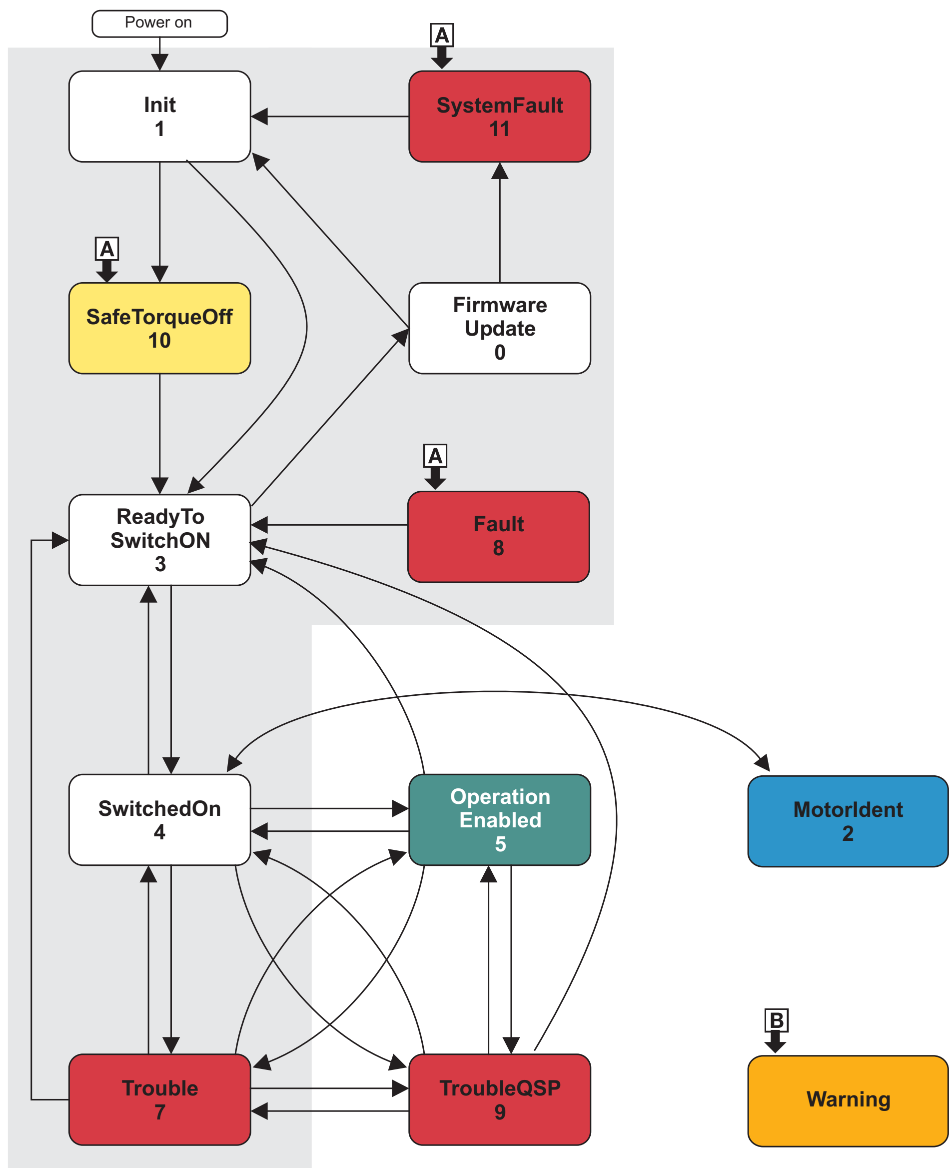
Tip!

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

4.2 Device state machine and device states

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

Device state machine



Grey field: Pulse inhibit

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 states.

- The arrows between the device states mark possible state 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).
- The C00137 displays the current device status.
- C00150 (status word) provides a bit coded representation of the current device status 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 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 active
8	Fault	Prio 7	1	0	0	0	Fault 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 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

"DRV-RDY" LED	"DRV-ERR" LED	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 active yet.
- The controller cannot be parameterised yet and no device commands can be executed.

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

"DRV-RDY" LED	"DRV-ERR" LED	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 state.

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 detected,

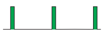
- the controller does not respond to setpoint changes or control processes, (e.g. speed setpoints, quick stop, torque limitations),
- the application remains active,
- all system interfaces (IO, bus systems, etc.) remain active,
- error monitoring remains active,
- the inverter is controlled independently 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. (🕮 104)

4.2.4 SafeTorqueOff

"DRV-RDY" LED	"DRV-ERR" LED	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 state

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

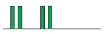
If the safety system deactivates the "Safe torque off (STO)" request, the device changes to the "ReadyToSwitchOn" status.

Detailed information on the integrated safety system can be found in the **8400 protec 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

"DRV-RDY" LED	"DRV-ERR" LED	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 functions are active.
- The controller can be parameterised.
- The application is basically executable.

Note!

- The "ReadyToSwitchOn" state 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/trouble... (📖 90)

4.2.6 SwitchedOn

"DRV-RDY" LED	"DRV-ERR" LED	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 functions 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" **AND** terminal RFR = HIGH level (controller enable)	OperationEnabled
Control bit "SwitchOn" 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 (📖 96)

4.2.7 OperationEnabled

"DRV-RDY" LED	"DRV-ERR" LED	Display in C00137	Display in status word 1 (C00150)			
			Bit 11	Bit 10	Bit 9	Bit 8
	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 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" **OR** terminal RFR = LOW level (controller inhibit).	SwitchedOn
Control bit "SwitchOn" 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 (📖 96)

4.2.8 TroubleQSP

"DRV-RDY" LED	"DRV-ERR" LED	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
Control bit "SwitchOn" of a control channel = "0".	ReadyToSwitchOn
Control bit "EnableOperation" of all control channels = "1" **AND** terminal RFR = HIGH level (controller enable) **AND** error is reset by the control bit "ResetFault" **AND** no more errors are pending.	OperationEnabled
Control bit "EnableOperation" of a control channel = "0" **OR** terminal RFR = LOW level (controller inhibit) **AND** error is reset by the control bit "ResetFault" **AND** no more errors are pending.	SwitchedOn
A message is active in the system.	Trouble

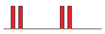
Related topics:

▸ wCANControl/wMCIControl control words (📖 96)

▸ Basics on error handling in the controller (📖 322)

▸ Error messages of the operating system (📖 347)

4.2.9 Trouble

"DRV-RDY" LED	"DRV-ERR" LED	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 in C00142 the "Inhibit at trouble" is activated, 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" **AND** terminal RFR = HIGH level (controller enable) **AND** the message has been cancelled.	OperationEnabled
Control bit "EnableOperation" of a control channel = "0" **OR** terminal RFR = LOW level (controller inhibit) **AND** the message has been cancelled.	SwitchedOn
In the system, there is an error configured on "TroubleQSP". **AND** the message has been cancelled.	TroubleQSP

Related topics:

- wCANControl/wMCIControl control words (📖 96)
- Basics on error handling in the controller (📖 322)
- Error messages of the operating system (📖 347)

4.2.10 Fault

"DRV-RDY" LED	"DRV-ERR" LED	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 (📖 96)
▸ Basics on error handling in the controller (📖 322)
▸ Error messages of the operating system (📖 347)

4.2.11 SystemFault

"DRV-RDY" LED	"DRV-ERR" LED	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/trouble...

.../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	0 ≡ Inhibit is not 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	0 ≡ Inhibit is not active
Bit 4	Auto-start option "Inhibit at Lenze setting" (from version 06.00.00)	1 ≡ Inhibit is active
Bit 5	Reserved	0
Bit 6		
Bit 7		

Note!

In the Lenze setting , only the automatic restart after loading 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.

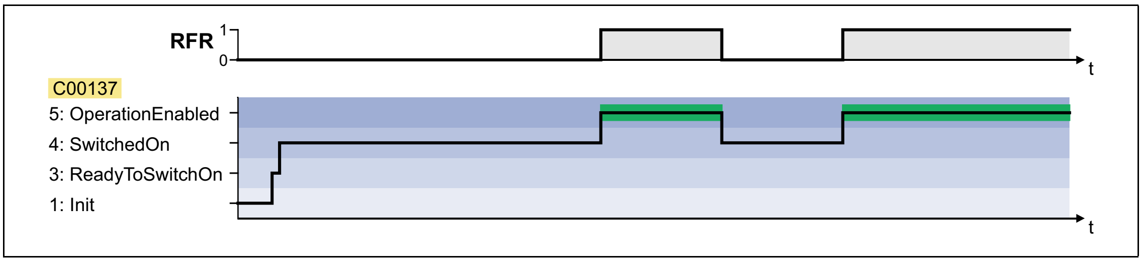
Danger!

In the Lenze setting, the "Inhibit at power-on" auto-start option has been deactivated in C00142 (bit 0 = 0), i.e. 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".

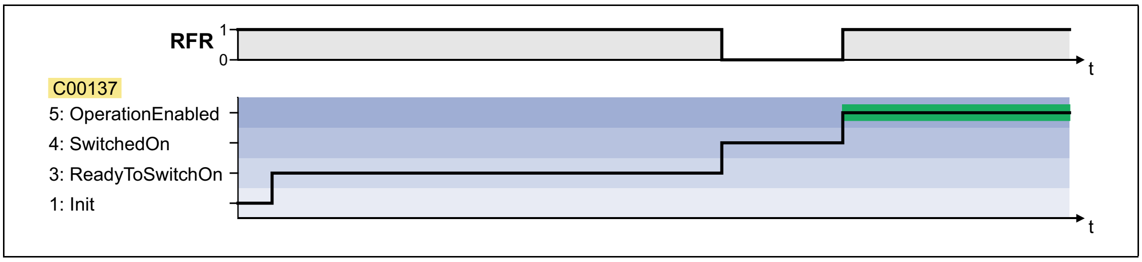
Case 1: No controller enable at mains connection

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



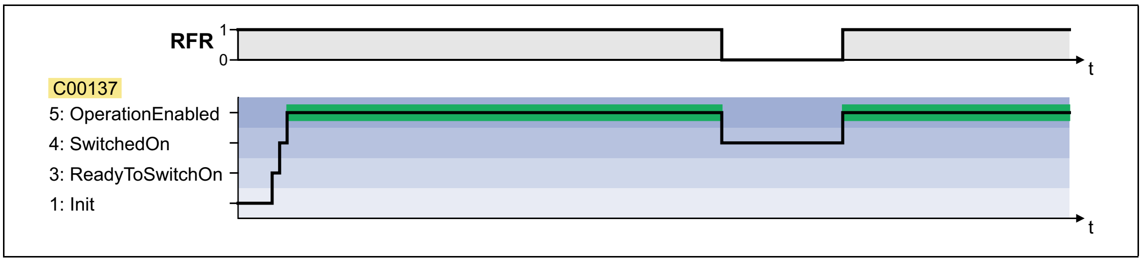
Case 2: Controller enable at mains connection and "Inhibit at power-on" activated

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

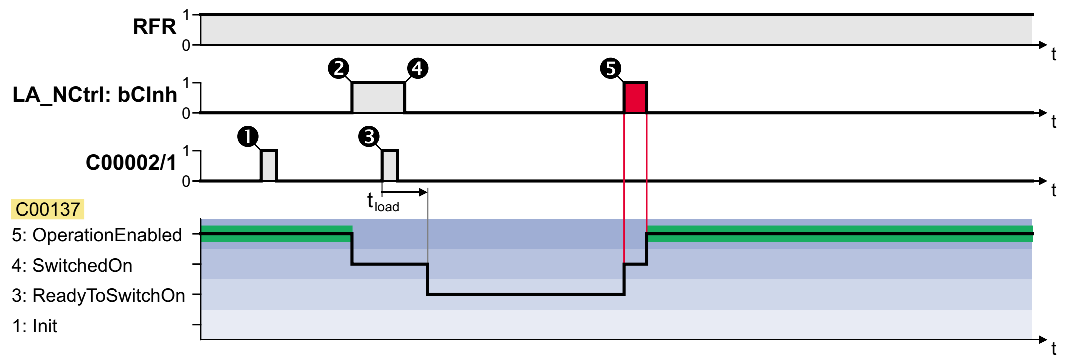


4.3.2 Auto-start option "Inhibit at Lenze setting"

This function extension is available from version 06.00.00!

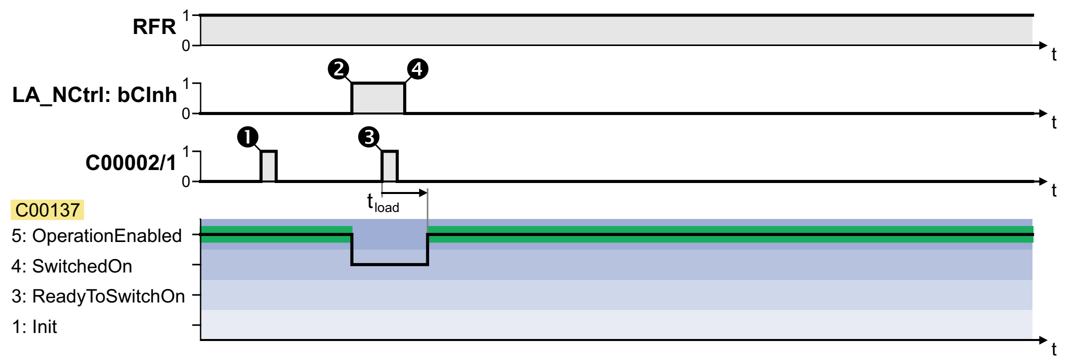
The "Inhibit at Lenze setting" auto-start option configurable via bit 4 of C00142 prevents the change to the "SwitchedOn" status after the Lenze setting has been loaded and the controller is enabled.

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

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

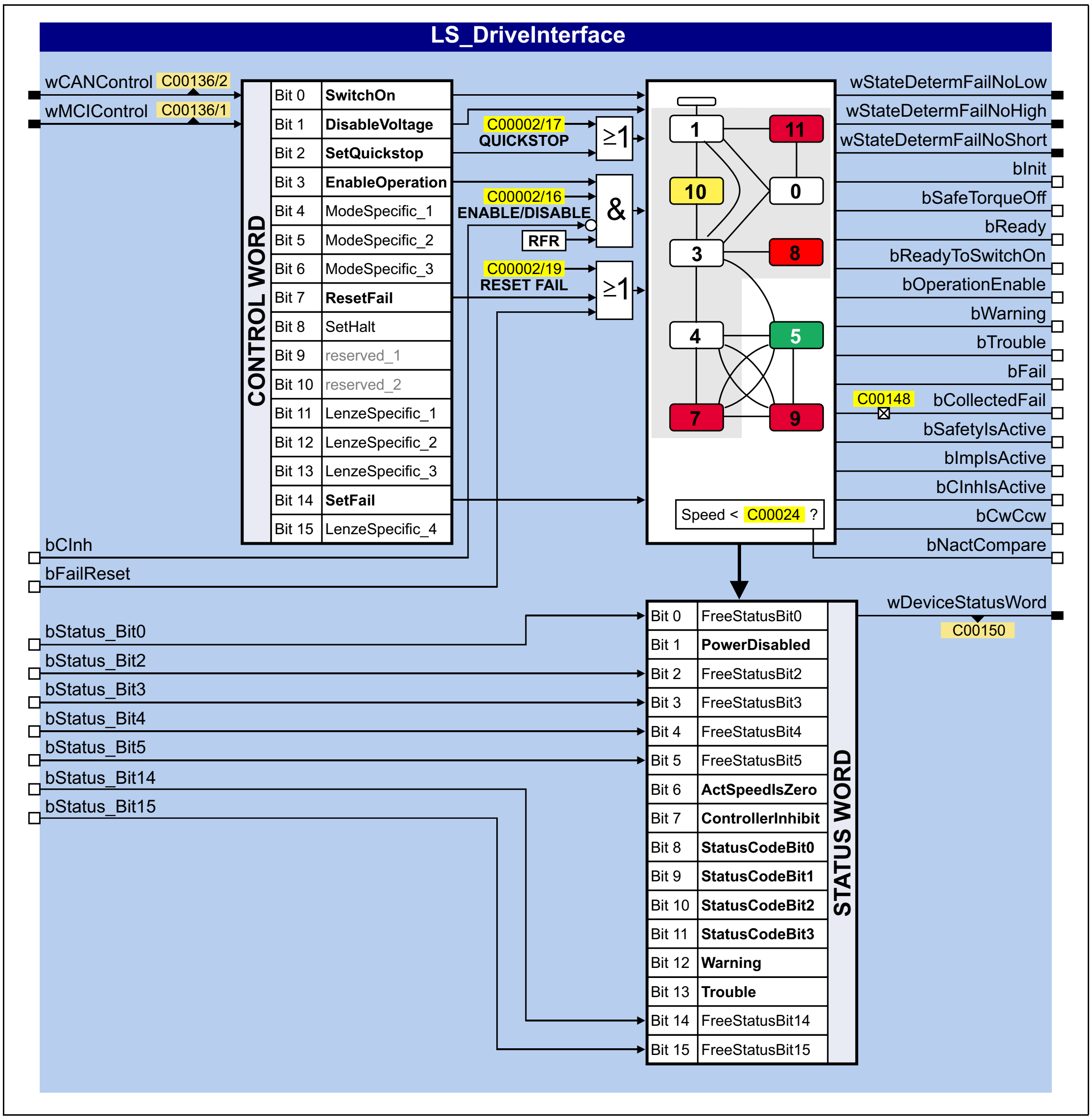


❶ Loading of the Lenze setting not possible since controller inhibit not set.
❷ Controller inhibit via signal change by the user
❸ Loading of the Lenze setting via device command C00002/1 = "1"
❹ Controller enable via signal change by the user
t_{load} = time required for loading the Lenze setting

[4-2] 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 DIS code | data type	Information/possible settings	
wCANControl C00136/2 | WORD	Control word via system bus (CAN) • 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. • For a detailed description of the individual control bits, see chapter "wCANControl/wMCIControl control words". (📖 96)	
wMCIControl C00136/1 | WORD	Control word via communication module (e.g. PROFIBUS) • 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. • For a detailed description of the individual control bits, see chapter "wCANControl/wMCIControl control words". (📖 96)	
bCInh C00833/36 | BOOL	▸ Enable/Inhibit controller (📖 75)	
	FALSE	Enable controller: The controller switches to the "OperationEnabled" device status if no other source for controller inhibit is active. • 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 C00833/37 | BOOL	▸ Reset error message (📖 351)	
	FALSE↗TRUE	The current error is reset.
bStatus_Bit0 bStatus_Bit2 bStatus_Bit3 bStatus_Bit4 bStatus_Bit5 bStatus_Bit14 bStatus_Bit15 C00833/38 ... 44 | BOOL	Freely assignable bits in the status word of the controller. • You can use these bits for returning information to the master control (e.g. IPC).	

Outputs

Identifier DIS code | data type	Value/meaning

wDeviceStatusWord
C00150 | WORD
	Status word of the controller (based on DSP-402)
	• The status word 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:
	  • Port block LP_CanOut1 when the CANopen system bus interface is used or
	  • Port block LP_MciOut when a plugged-in communication module is used (e.g. PROFIBUS).
	• For a detailed description of each status bit see chapter "wDeviceStatusWord status word". (📖 97)

wStateDetermFailNoLow
WORD
	Display of the status determining error (32-bit error number, Low-Word)
	• From version 06.00.00 onwards: If the "Use 16BitFailNo." (Bit 15 = "1") option is activated in C00148, the short16-bit error number (*wStateDetermFailNoShort*) is provided via this output as well.
	  • In this case, the *wStateDetermFailNoHigh* output is "0".
	  • Advantage: The bus transfer of the error numbers is possible via a data word without changing the interconnection of the technology application.

wStateDetermFailNoHigh WORD	Display of the status determining error (32-bit error number, High-Word)
wStateDetermFailNoShort WORD (from version 06.00.00)	Display of the status determining error (16-bit error number)

Identifier DIS code | data type	Value/meaning	
bInit BOOL	TRUE	"Init" device state is active
bSafeTorqueOff BOOL	TRUE	"SafeTorqueOff" device state is active
bReady BOOL	TRUE	"SwitchedOn" device state is active
bReadyToSwitchOn BOOL	TRUE	"ReadyToSwitchOn" device state is active
bOperationEnable BOOL	TRUE	"OperationEnabled" device state is active
bWarning BOOL	TRUE	A warning is indicated
bTrouble BOOL	TRUE	"Trouble" device state is active
bFail BOOL	TRUE	"Fault" device state is active
bCollectedFail 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 BOOL	TRUE	In preparation
bImpIsActive BOOL	TRUE	Pulse inhibit is active
bCInhIsActive BOOL	TRUE	Controller inhibit is active
bCwCcw BOOL	FALSE	Motor rotates in CW direction
	TRUE	Motor rotates in CCW direction
bNactCompare BOOL	TRUE	During open-loop operation: Speed setpoint < Comparison value (C00024)
		During closed-loop operation: Actual speed value < Comparison value (C00024)

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 • This bit has to be set in the CAN **AND** MCI control word in order that the drive changes to the "SwitchedOn" device state. In order to reach the "ReadyToSwitchOn" state, it is sufficient to set the bit to 0 in of the two control words.
Bit 1	DisableVoltage	1 ≡ Inhibit inverter control (pulse inhibit)
Bit 2	SetQuickStop	Activate quick stop (QSP) ▸ Activate/deactivate quick stop (📖 76)
Bit 3	EnableOperation	1 ≡ Enable controller (RFR) • 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) • Acknowledge error message (if the error cause has been eliminated).
Bit 8	SetHalt	1 ≡ Activate stop function • 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 • 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 • 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: 1 ≡ Speed setpoint < Comparison value (C00024)
		During closed-loop operation: 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 ▶ Device state machine and device states (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 • 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.

Topics:

Basic settings:

- Motor selection/Motor data
- Selecting the control mode
- Defining current and speed limits

Description of the motor control types:

- V/f characteristic control (VFCplus)
- V/f control (VFCplus + encoder)
- Sensorless vector control (SLVC)

Parameterisable additional functions:

- Selection of the switching frequency
- Flying restart function
- DC-injection braking
- Slip compensation
- Oscillation damping
- Phase sequence reversal for correcting misconnected UVW motor phases

Further topics:

- Encoder/feedback system
- Braking operation/brake energy management
- Monitoring

Internal interfaces (process signals):

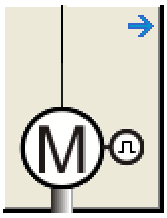
- Internal interfaces | System block "LS_MotorInterface"
- Internal status signals | System block "LS_DeviceMonitor"

5.1 Motor selection/Motor data

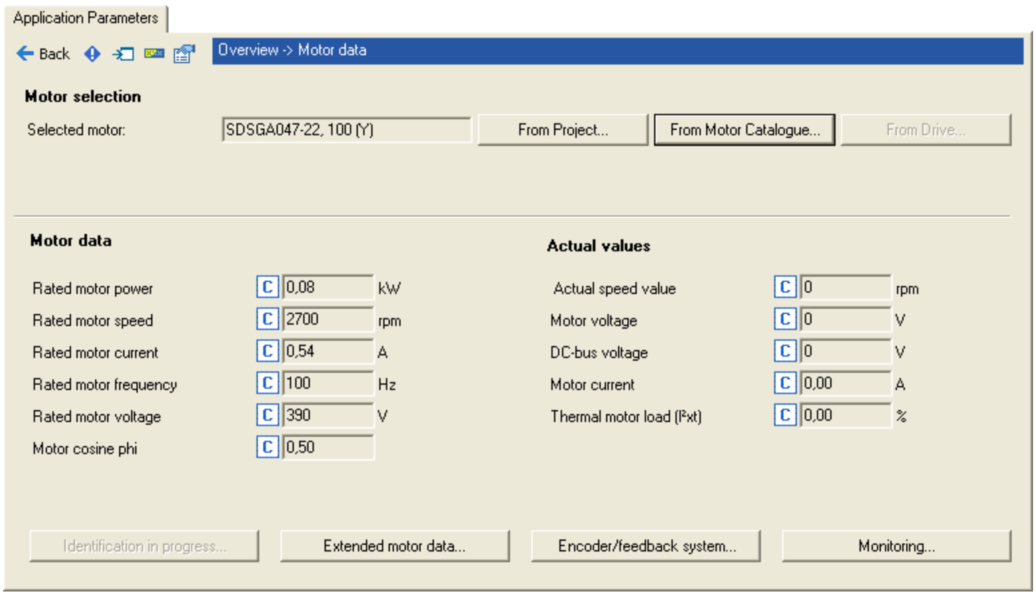
The motor data term comprises all parameters that only depend on the motor and that only characterise the electrical behaviour of the machine. The motor data are independent of the application in which the controller and the motor are used.

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

1. »Engineer« Go to the *Project view* and select the 8400 protec StateLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
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« (📖 102)
- 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 (📖 104)
- The **Encoder/feedback system...** serves to get to the settings for the encoder/feedback system, if available. ▸ Encoder/feedback system (📖 165)

Note!

Sensorless vector control 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.

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

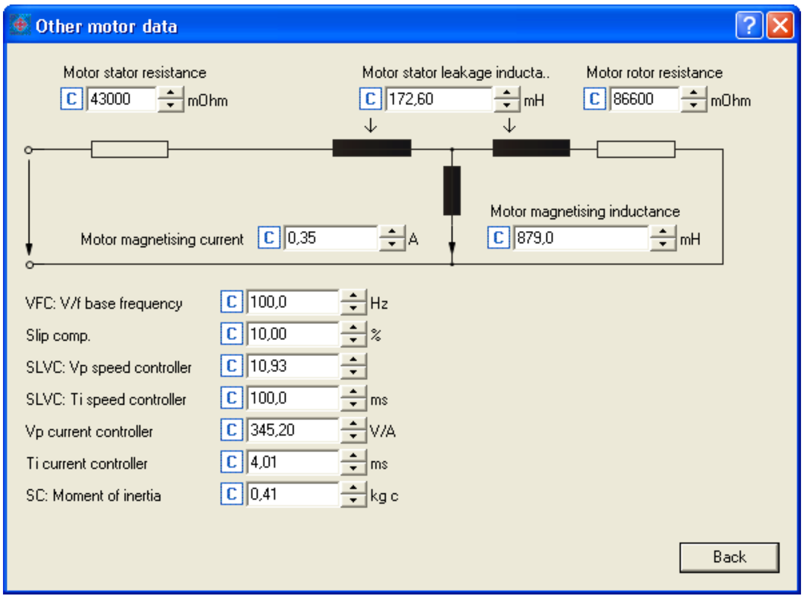
- the adjustment of the inverter error characteristic to the drive system and
- the knowledge of the motor cable resistance.

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

▶ Automatic motor data identification (📖 104)

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):

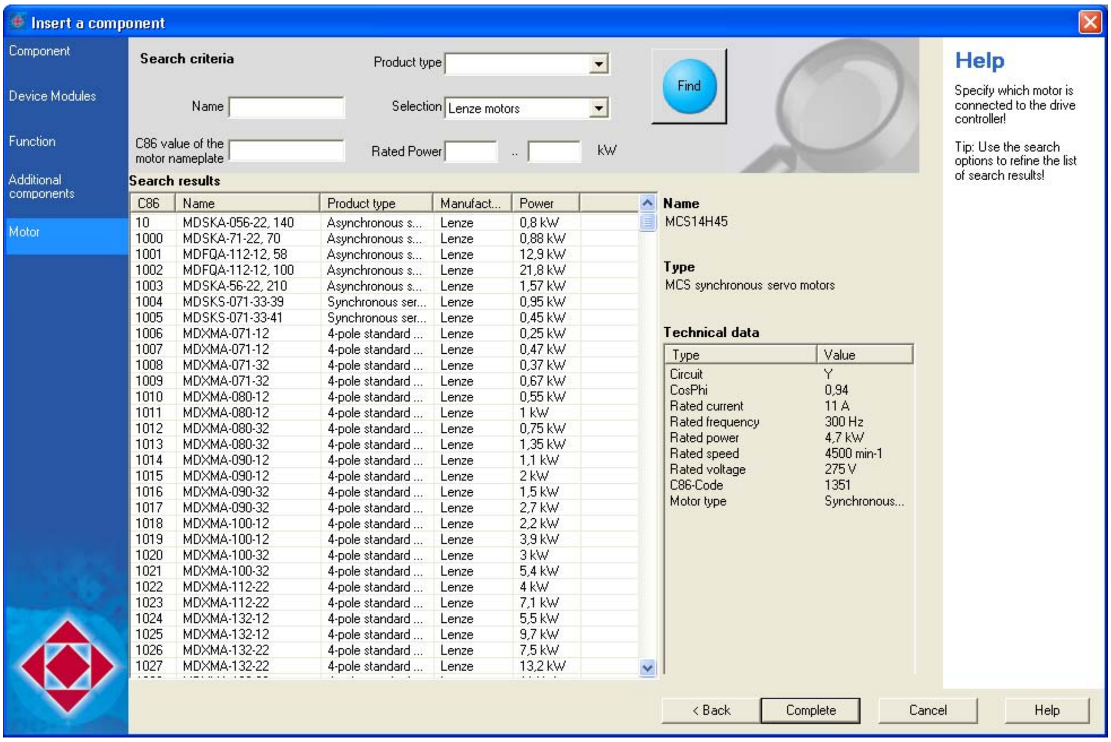


Parameter	Info	ASM	PSM
C00084	Motor stator resistance	●	●
C00085	Motor stator leakage inductance	●	●
C00082	Motor rotor resistance	●	
C00095	Motor magnetising current	●	
C00092	Motor magnetising inductance	●	
C00015	VFCplus: V/f base frequency	●	
C00021	Slip compensation	●	
C00070	SLVC: Vp speed controller	●	
C00071	SLVC: Ti speed controller	●	

Parameter	Info	ASM	PSM
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«

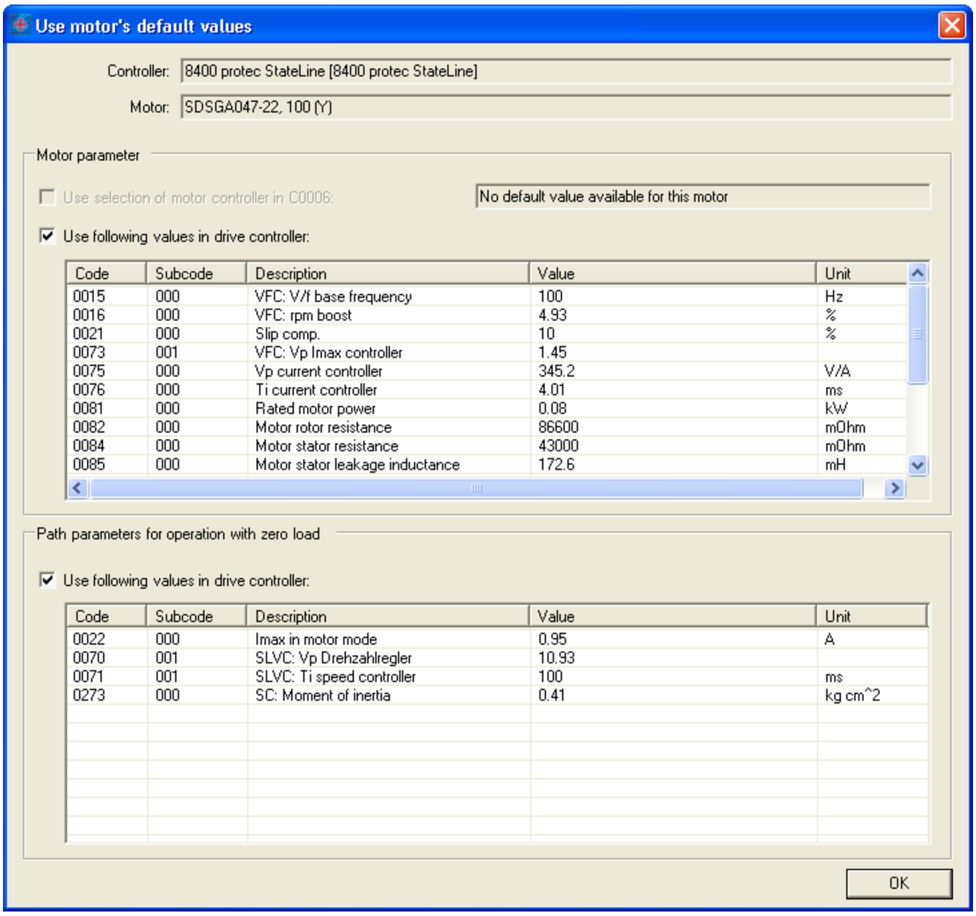
If you, when inserting the controller into the project in the dialog step "Other components", put a checkmark in the control field **Motor**, you can select as a further dialog step the motor for the controller from the motor catalogue:



- As an alternative you can also insert the motor into the project later on via the **Insert 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 "Identify motor parameters" device command (C00002/23), 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	●	

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!

Note!

- We strongly recommend motor parameter identification before the initial commissioning of the sensorless vector control (SLVC).
- 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

Duration & sequence of the motor parameter identification

The duration of the motor parameter identification is approx. 30 s. The following steps are carried out during this time:

1. The motor stator resistance (C00084) is measured.
2. The inverter error characteristic is measured.
3. The motor stator leakage inductance (C00085) is measured.
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. The slip compensation (C00021) is calculated.
8. The 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 a motor with power adaptation to the inverter.

Note!

Motor parameter identification may be aborted if a special motor (e.g. mid-frequency motor) is used or if there is a large deviation between inverter and motor power.

In this case we recommend

- 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.

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.
2. Wait until the drive is at standstill.
3. Transfer the nameplate data to the following codes:
   - C00081: Rated motor power
   - C00087: Rated motor speed
   - C00088: Rated motor current*
   - C00089: Rated motor frequency*
   - C00090: Rated motor voltage*
   - C00091: Motor cos φ

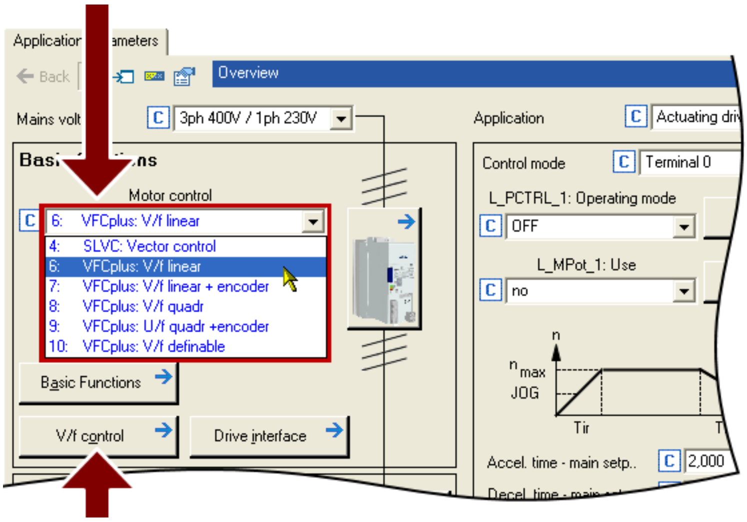
   * According to the star/delta connection method.
4. Defining the motor cable length and motor cable cross-section:
   - C00915: Motor cable length
   - C00916: Motor cable cross-section

   The resulting motor cable resistance is displayed in C00917.
5. Start motor parameter identification:
   - Device command C00002/23 = "1: On / start"
6. Enable the controller again.
   - 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.
7. Inhibit controller again.

5.2 Selecting the control mode

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

- The V/f characteristic control (VFCplus) is preset with a linear characteristic.
- 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". (▯ 110)

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) (📖 114)

V/f control (VFCplus + encoder)

V/f control can be selected if operation with speed feedback is required. This motor control includes an additional parameterisable slip regulator which dynamically adapts the actual speed value to the speed setpoint.

▸ V/f control (VFCplus + encoder) (📖 130)

Sensorless vector control (SLVC)

Sensorless (field-oriented) vector control 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!

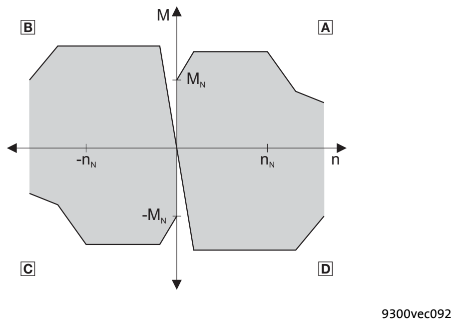
If a high torque without feedback is to be provided at small speeds, we recommend the "Sensorless vector control" motor control mode.

▸ Sensorless vector control (SLVC) (📖 136)

Speed feedback

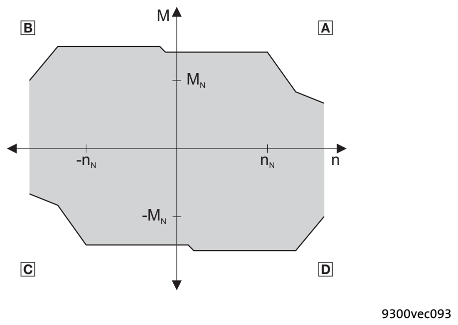
As shown in the following graphics, the drive systems with feedback have, independently of the motor control, more advantages than systems without feedback.

V/f characteristic control (VFCplus)



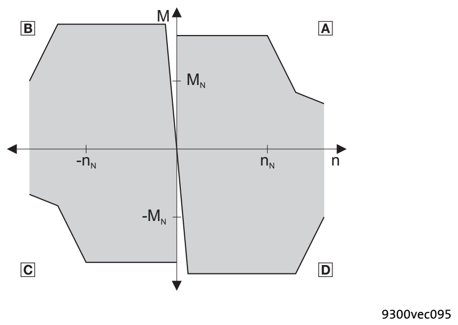
A Operation in motor mode (CW rotation)
B Operation in generator mode (CCW rotation)
C Operation in motor mode (CCW rotation)
D Operation in generator mode (CW rotation)

V/f control (VFCplus + encoder)



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)



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
With extremely alternating loads	VFCplus: V/f linear	SLVC
With high starting duty	SLVC	VFCplus: V/f linear
Torque limitation	SLVC	-
With torque limitation (power control)	VFCplus: V/f linear	SLVC
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	VFCplus: U/f square-law	VFCplus or SLVC
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	VFCplus: V/f linear	SLVC
With extremely alternating loads	VFCplus: V/f linear	SLVC
With high starting duty	VFCplus: V/f linear	SLVC
With speed control (speed feedback)	VFCplus: V/f linear	-
With high dynamic performance e.g. for positioning and infeed drives	VFCplus: V/f linear	-
Torque limitation	VFCplus: V/f linear	SLVC
With torque limitation (power control)	-	-
Winder with dancer position control	VFCplus: V/f linear	-
Unwinder with dancer position control	VFCplus: V/f linear	-
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 • 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) is selected, the 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).

5.4 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 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 your pump drive or fan drive is not suitable for operation with a quadratic V/f characteristic, you must either use the V/f characteristic control function with a linear V/f characteristic or the sensorless vector control (SLVC).
- 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 the quadratic V/f characteristic, we recommend to set a lower V_{min}.
- 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!

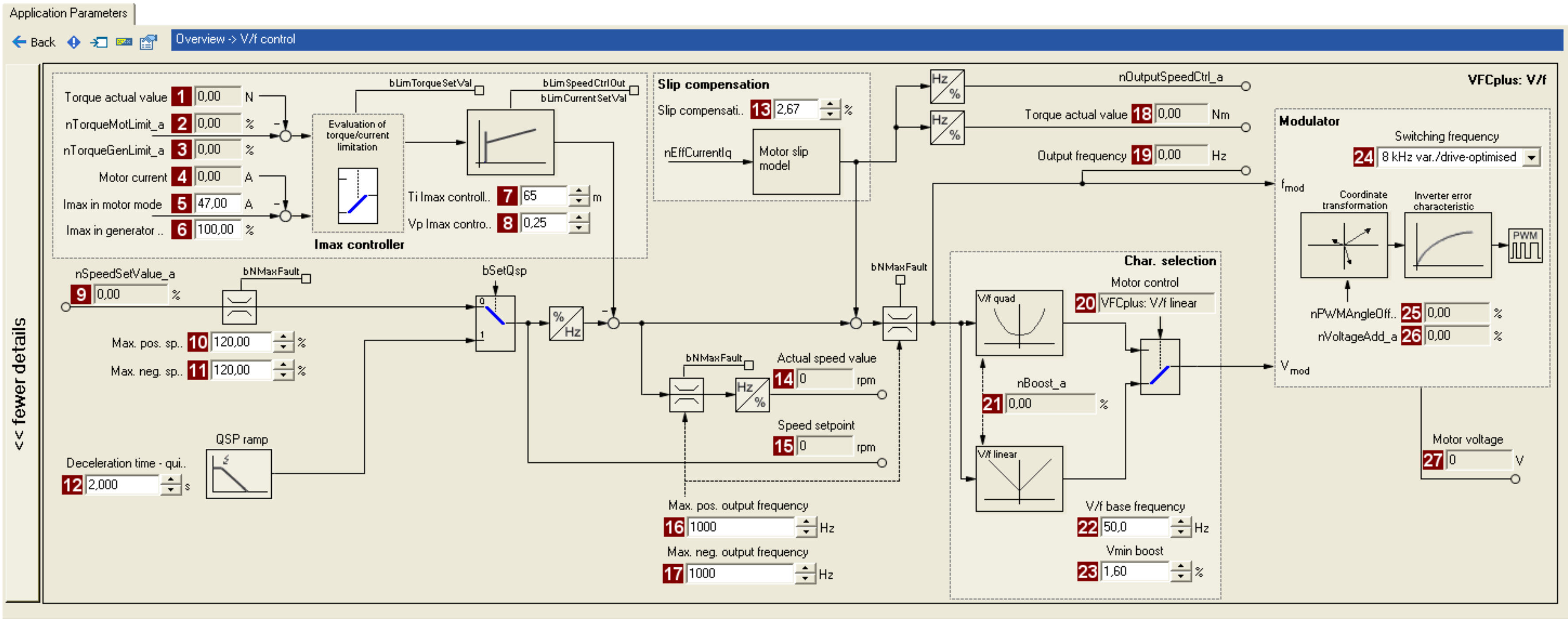
▶ Automatic DC-injection braking (auto DCB) (📖 156)

5.4.1 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 protec StateLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
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

   Alternatively also the motor control "10: VFCplus: V/f definable" can be selected. In the case of this motor control the V/f characteristic can be defined freely. ▶ Defining a user-defined V/f characteristic (126)
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

5.4.2 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. (📖 117)
2.	Defining current limits (Imax controller). (📖 118)

Tip!

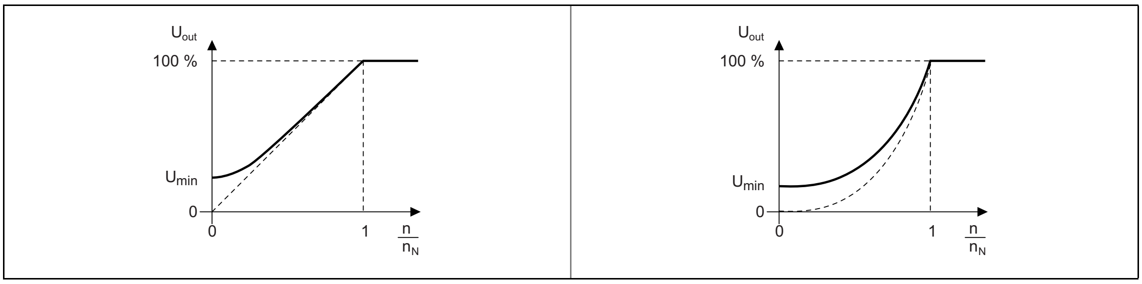
Information on the optimisation of the control mode and the adaptation to the real application is provided in chapter "Optimising the control mode". (📖 119)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 149)

5.4.2.1 Defining the V/f characteristic shape

In principle, three 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)

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

Tip!

You can find detailed information on freely definable V/f characteristics in the subchapter entitled "Defining a user-defined V/f characteristic". (🕮 126)

5.4.2.2 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).

5.4.3 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 (📖 120)
▶ Adapting the Vmin boost (📖 121)

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 (📖 126)

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 (📖 122)
- 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 (📖 133)

5.4.3.1 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:

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

$V_{frequency\ inverter}$: 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 / 690 400	50 50	Δ	50 Hz
230 / 400 280 / 480 400	50 60 87	Δ	87 Hz
220 / 380	50	Δ	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	Δ	50 Hz
220 / 380	50	Δ	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.

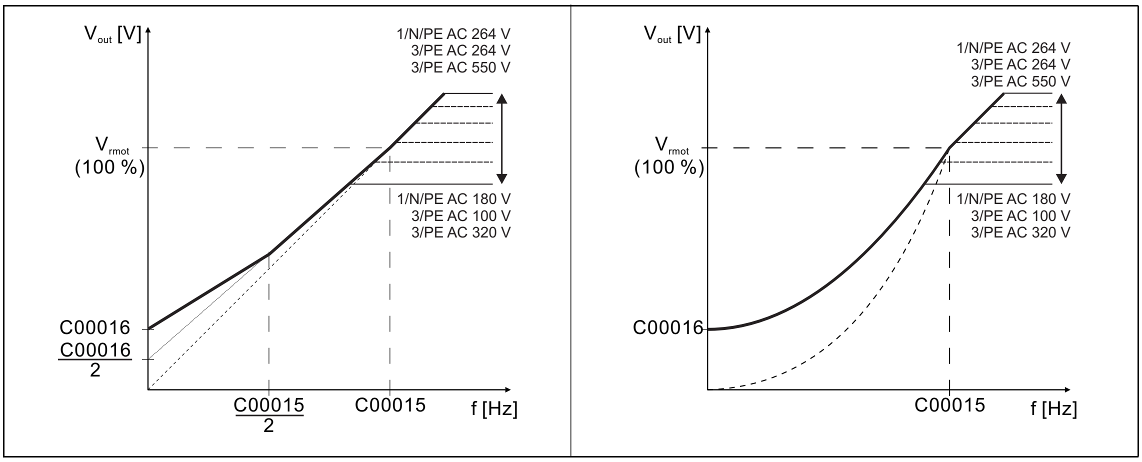
5.4.3.2 Adapting the Vmin boost

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.

5.4.3.3 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.

5.4.3.4 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).

5.4.3.5 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 C00830/29 | INT	Torque limitation in motor mode • Scaling: 16384 ≡ 100 % M_{max} (C00057) • Setting range: 0 ... +199.99 % • If keypad control is performed: Parameterisable via C00728/1.
nTorqueGenLimit_a C00830/28 | INT	Torque limitation in generator mode • Scaling: 16384 ≡ 100 % M_{max} (C00057) • Setting range: -199.99 ... 0 % • If keypad control is performed: Parameterisable via C00728/2.

Note!

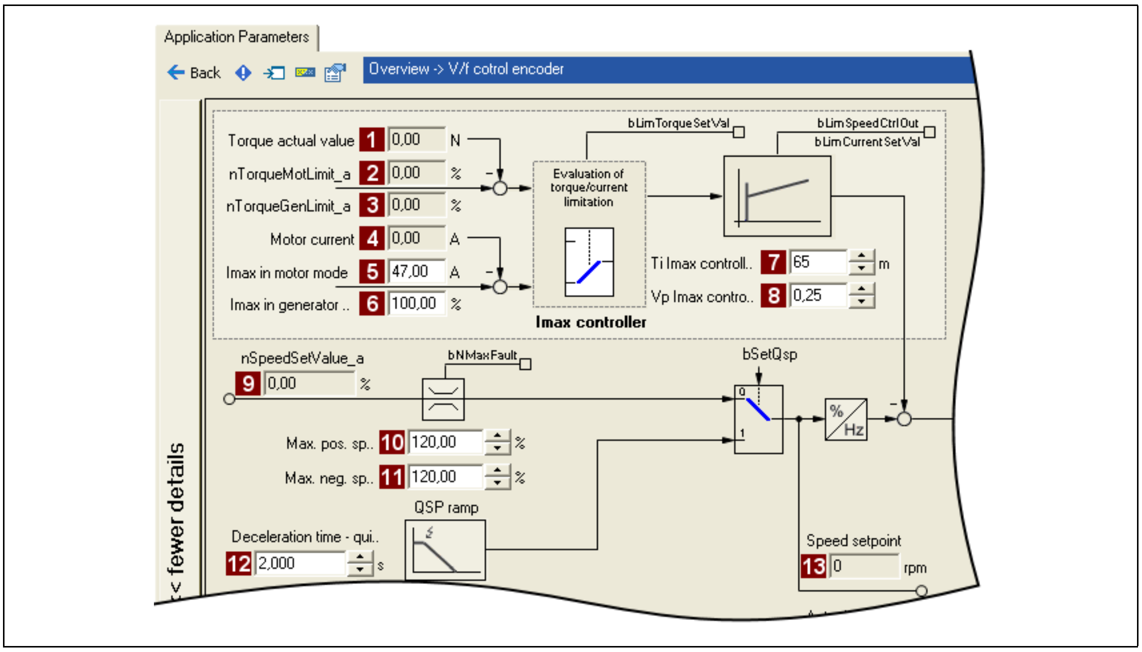
- The actual torque (C00056/2) is directly calculated from the current slip speed of the machine. This requires correct entry of the motor data. ▸ Motor selection/Motor data (📖 99)
- To avoid instabilities during operation with active slip compensation, the torque limit values are internally processed as absolute values.
- If slip compensation is deactivated (C00021 = 0), indirect torque limitation (differential signal between apparent motor current and *nTorqueMotLimit_a* or *nTorqueGenLimit_a*) occurs. 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		Parameter	Info
1	C00056/2	Actual torque value	9	C00830/22	MCTRL: nSpeedSetValue_a
2	C00830/29	Limitation of torque in motor mode	10	C00909/1	Max. pos. speed
3	C00830/28	Limitation of torque in generator mode	11	C00909/2	Max. neg. speed
4	C00054	Motor current	12	C00105	Decel. time - quick stop
5	C00022	Imax in motor mode	13	C00050	Speed setpoint
6	C00023	Imax in generator mode			
7	C00074	Ti Imax controller			
8	C00073	Vp Imax controller			

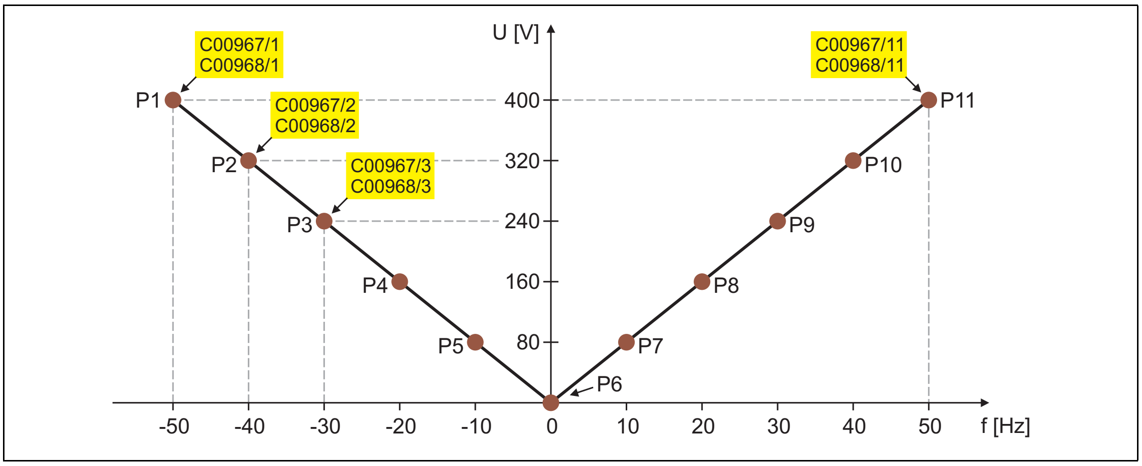
5.4.3.6 Defining a user-defined V/f characteristic

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 selected via 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)

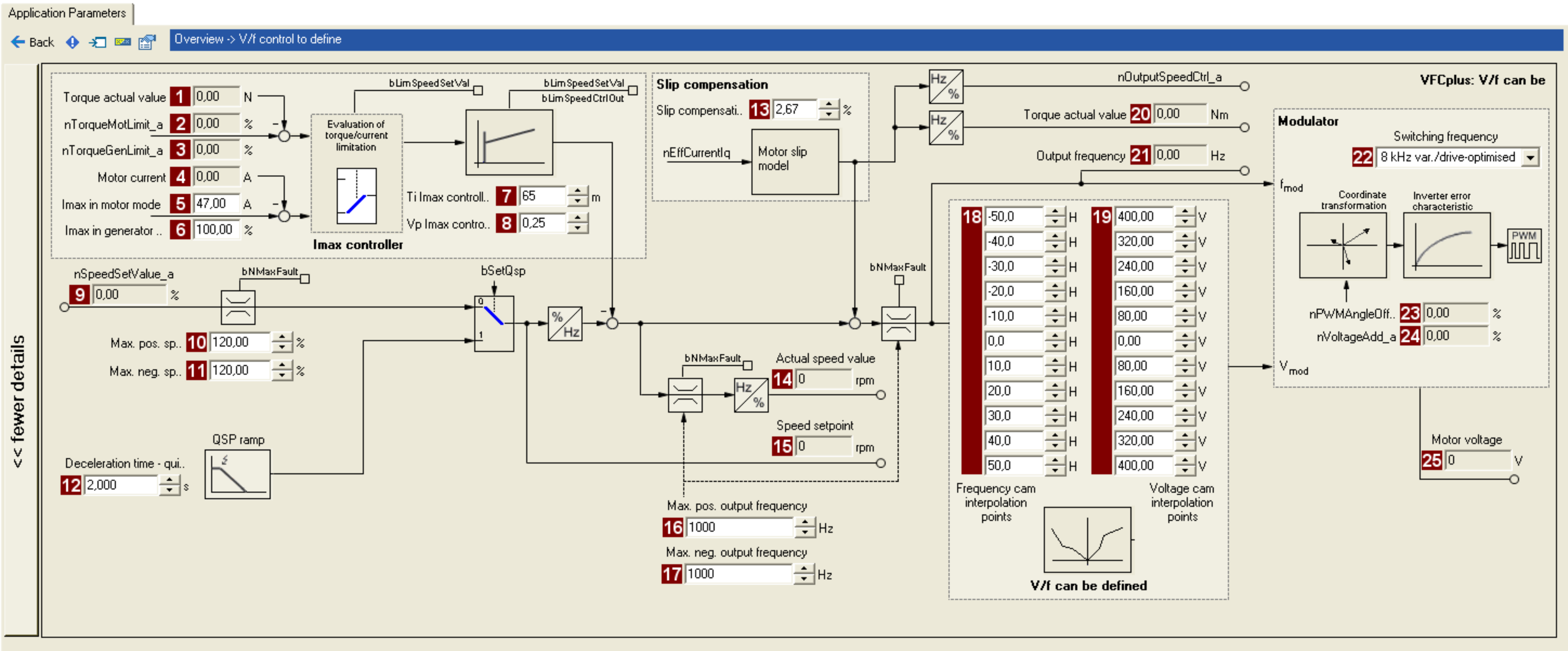
Tip!

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 protec StateLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
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
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
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

5.4.4 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 (📖 104)
Problems in case of high starting duty (great mass inertia)	▸ Adapting the Vmin boost (📖 121)
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: • Prolong acceleration/deceleration times: C00012: Accel. time - main setpoint C00013: Decel. time - main setpoint • Consider a sufficient magnetising time of the motor. Depending on the motor power, the magnetising time amounts to 0.1 ... 0.2 s. • Increase the maximally permissible current: C00022: Imax in motor mode 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). Important: Unstable drive due to overcompensation! • With cyclic load impulses (e. g. centrifugal pump), a smooth motor characteristic is achieved by smaller values in C00021 (possibly negative values). 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) • Reduce the reset time of the I_{max} controller (C00074/1) • Prolong the acceleration time (C00012) • 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) • If motor power < inverter power: Set C00022 to $I_{max} = 2\ I_{rated\ motor}$ • Reduce dynamic performance of setpoint generation

5.5 V/f control (VFCplus + encoder)

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

- Steady-state accuracy of the speed
- Less parameterisation effort compared to the 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 to the V/f control. (📖 114)

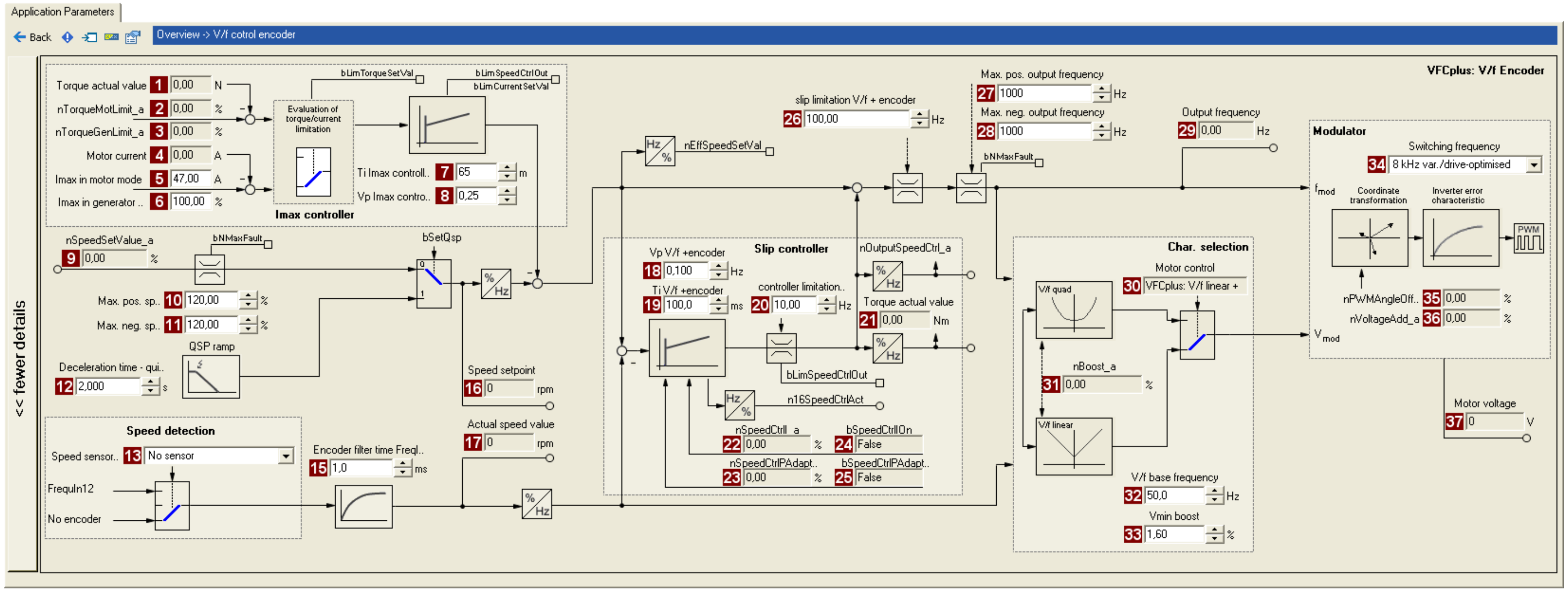
Note!

- Make sure that, when the motor control with speed feedback is in use, the maximum input frequency of 100 kHz is not exceeded. ▸ Using DI1(5) and DI2(6) as frequency inputs (📖 202)
- 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.

5.5.1 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 protec StateLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
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 quadratic 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		Parameter	Info		Parameter	Info
1	C00056/2	Actual torque value	16	C00050	Speed setpoint	29	C00058	Output frequency
2	C00830/29	Limitation of torque in motor mode	17	C00051	Actual speed value	30	C00006	Motor control
3	C00830/28	Limitation of torque in generator mode	18	C00972	Vp Vf+encoder	31	C00830/26	MCTRL: nBoost_a
4	C00054	Motor current	19	C00973	Ti Vf+encoder	32	C00015	V/f base frequency
5	C00022	Imax in motor mode	20	C00971/1	Controller limitation Vf+encoder	33	C00016	Vmin boost
6	C00023	Imax in generator mode	21	C00056/2	Actual torque value	34	C00018	Switching frequency
7	C00074	Ti Imax controller	22	C00830/24	MCTRL: nSpeedCtrlI_a	35	C00830/32	MCTRL: nPWMAngleOffset_a
8	C00073	Vp Imax controller	23	C00830/25	MCTRL: nSpeedCtrlPAdapt_a	36	C00830/31	MCTRL: nVoltageAdd_a
9	C00830/22	Speed setpoint	24	C00833/31	MCTRL: bSpeedCtrlIOn	37	C00052	Motor voltage
10	C00909/1	Max. pos. speed	25	C00833/69	MCTRL: bSpeedCtrlPAdaptOn			
11	C00909/2	Max. neg. speed	26	C00971/2	Slip limitation Vf+encoder	More relevant parameters for Encoder/feedback system:		
12	C00105	Decel. time - quick stop	27	C00910/1	Max. pos. output frequency		C00115	DI1/2 & DI6/7 function
13	C00495	Speed sensor selection	28	C00910/2	Max. neg. output frequency		C00420	Number of encoder increments
14	-	-					C00425	Encoder scanning time
15	C00497/1	Encoder filter time FreqIn12					C00496	Encoder evaluation method

5.5.2 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: • C00006 = 7: Linear characteristic • C00006 = 9: Quadratic characteristic
2.	Defining current limits (Imax controller). (📖 118)
3.	Parameterise encoder/feedback system. ▸ Encoder/feedback system (📖 165)
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. ▸ Motor selection/Motor data (📖 99)
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. • In case of a sign reversal between actual value and setpoint, check the connection of the encoder (e.g. change track A or B of the encoder or invert the actual speed value). • 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.	To protect the drive, reduce the slip regulator limitation in C00971/1. • 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. ▸ Parameterising the slip regulator (📖 133)
10.	In a final step, increase the slip regulator limitation again in C00971/1. • e.g. increase to twice the slip frequency

Tip!

Information on the further 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). (📖 119)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 149)

5.5.2.1 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.

- Unlike traditional speed controllers, the slip regulator only controls the slip.
- In the Lenze setting, the configuration of the slip regulator provides 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 the slip regulator gain (C00972) until the drive is 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.
   - With low encoder resolutions, another reduction of the the slip regulator gain for low speeds may be necessary (speed setpoint ≈ 0).
   - We recommend to finally check the behaviour at setpoint speed = 0 and to further reduce the slip regulator gain if irregular running occurs.
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 the slip regulator time constant (C00973) until the drive is semi-stable.
   - This can be recognised by motor noise, "motor vibrations" or resonance on the actual speed value 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:

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

[5-8] Calculation of the rated slip

Note!

A setting of C00971/1 = 0 Hz deactivates the slip regulator. In this case, the structure of the V/f control corresponds to the structure of a 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).

- If the slip is e.g. limited to twice the rated slip of the motor, a stalling of the motor during very dynamic processes can be avoided.
- Motor stalling is caused by:
  - a 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.

5.6 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 connected motor may be maximally two power classes lower 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 chapter "Adapting the V/f base frequency" relating to V/f characteristic control. (120)

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.

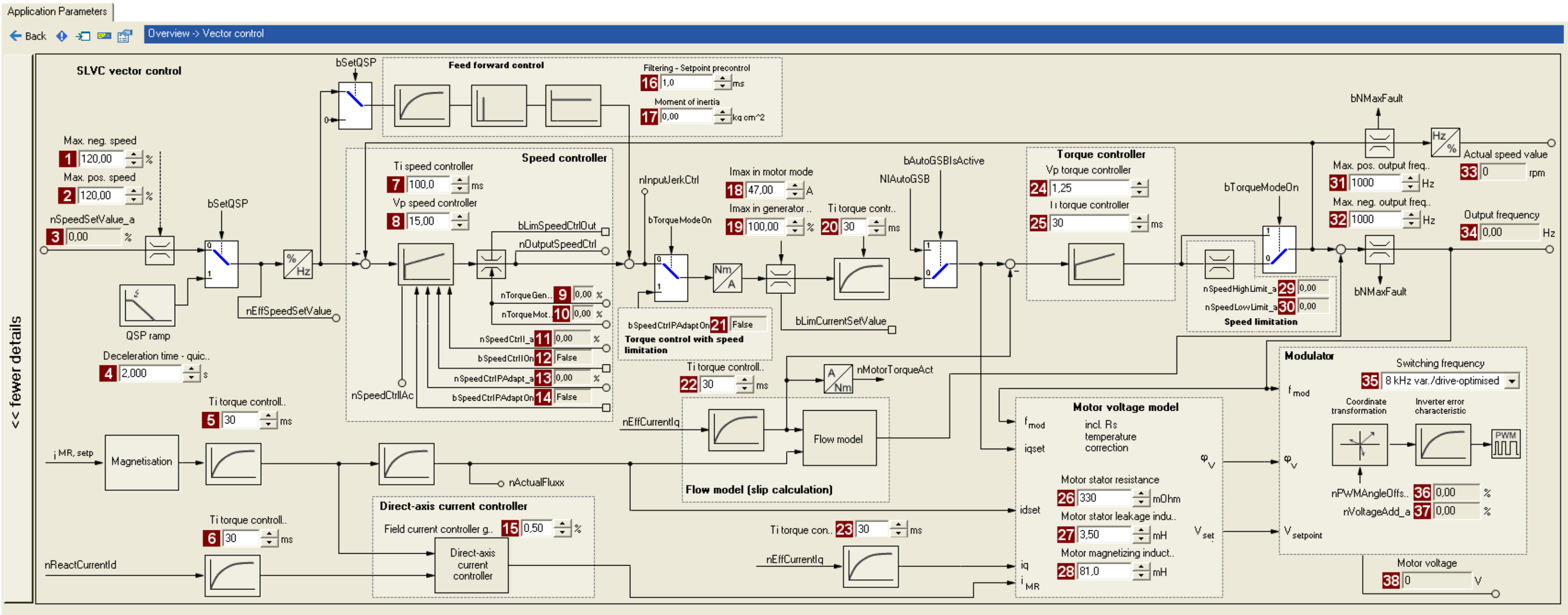
In comparison to the V/f characteristic control without feedback, the following can be achieved by means of sensorless vector control SLVC:

- 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

5.6.1 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 protec StateLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
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
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
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

5.6.2 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)

5.6.2.1 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 DIS code | data type	Information/possible settings
nTorqueMotLimit_a C00830/29 | INT	Torque limitation in motor mode • Scaling: 16384 ≡ 100 % M_{max} (C00057) • Setting range: 0 ... +199.99 % • If keypad control is performed: Parameterisable via C00728/1.
nTorqueGenLimit_a C00830/28 | INT	Torque limitation in generator mode • Scaling: 16384 ≡ 100 % M_{max} (C00057) • Setting range: -199.99 ... 0 % • 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 (📖 160)
  - Using a slip model, the slip of the machine is reconstructed.
  - The slip compensation (C00021) acts as the influencing parameter.

5.6.2.2 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 DIS code | data type	**Information/possible settings**
nTorqueSetValue_a C00830/27 | INT	Torque setpoint / additive torque • Scaling: 16384 ≡ 100 % M_{max} (C00057)
nSpeedHighLimit_a C00830/88 | INT	Upper speed limit for speed limitation (only for torque-controlled operation) • Scaling: 16384 ≡ 100 % rated speed (C00011)
nSpeedLowLimit_a C00830/23 | INT	Lower speed limit for speed limitation (only for torque-controlled operation) • Scaling: 16384 ≡ 100 % rated speed (C00011)

5.6.3 Basic settings

The following "Initial commissioning steps" must be performed to commission the sensorless vector control:

<table>
<tr><th colspan="3">Initial commissioning steps</th></tr>
<tr><td>1.</td><td colspan="2">Determine the motor control:
C00006 = "4: SLVC: Vector control"</td></tr>
<tr><td rowspan="2">2.</td><td colspan="2">Set the motor selection/motor data
• 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. (📖 99)
Depending on the motor manufacturer, proceed as follows:</td></tr>
<tr><td>Lenze motor:
Selecting a motor from the motor catalogue in the »Engineer«
- or -
1. Set the motor nameplate data
2. Automatic motor data identification</td><td>Third party manufacturer's motor:
1. Set the motor nameplate data
2. Automatic motor data identification or set known equivalent circuit diagram data manually:
C00082: Motor rotor resistance
C00084: Motor stator resistance
C00085: Motor stator leakage inductance
C00092: Motor magnetising inductance
C00095: Motor magnetising current</td></tr>
<tr><td>3.</td><td colspan="2">Define the type of control:
bTorquemodeOn = FALSE: Speed control with torque limitation
bTorquemodeOn = TRUE: Torque control with speed limitation</td></tr>
<tr><td>4.</td><td colspan="2">Set the slip compensation (C00021). ▶ Slip compensation (📖 160)</td></tr>
</table>

Tip!

Information on the optimisation of the control mode and the adaptation to the real application is provided in chapter "Optimising the control mode". (📖 142)

We recommend to use the flying restart function for connecting/synchronising the inverter to an already rotating drive system. ▶ Flying restart function (📖 152)

Parameterisable additional functions are described correspondingly in the chapter "Parameterisable additional functions". (📖 149)

5.6.4 Optimising the control mode

5.6.4.1 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!

5.6.4.2 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:

$$V_P \approx 1.5 \dots 3 \cdot \frac{T_M[s]}{0.01[s]}$$

$$T_M[s] = \frac{2 \cdot \pi \cdot n_N[rpm]}{M_N[Nm] \cdot 60} \cdot J_{Drive,\ total}[kgm^2]$$

$$M_N[Nm] = \frac{P_N[W] \cdot 60}{2 \cdot \pi \cdot n_N[rpm]}$$

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

[5-9] 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

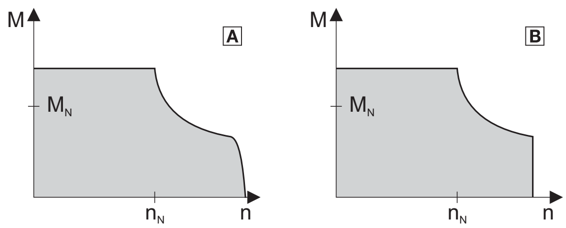
5.6.4.3 Optimising dynamic performance and field weakening behaviour

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

[5-10] 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.

5.6.4.4 Optimising the stalling behaviour

Motor stalling due to a torque overload in the field weakening range is prevented in sensorless vector control 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):

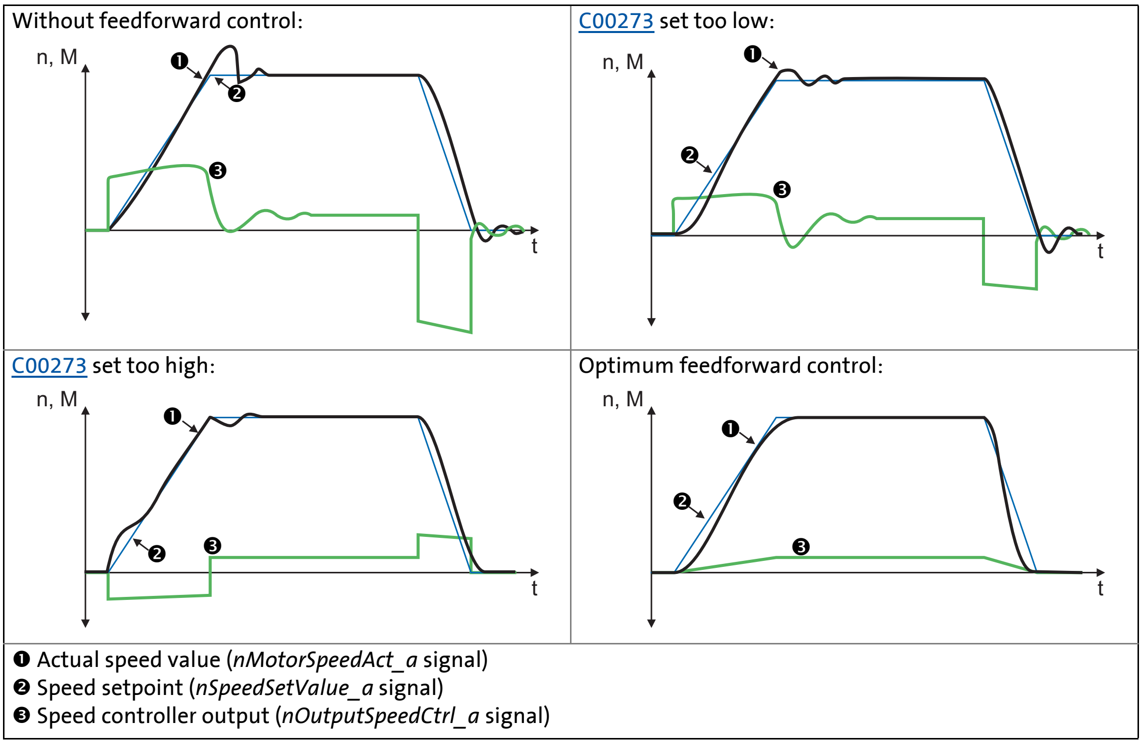
- C00080 > 0 Hz:
  The reduction of the magnetising current is shifted to higher field frequencies. Here, there is a risk of the motor being magnetised too much and having too little voltage reserve for the torque-creating current.
- C00080 < 0 Hz:
  The reduction of the magnetising current is shifted 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.

5.6.4.5 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.



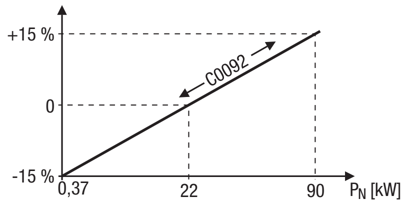
[5-11] Typical signal characteristics for different settings of the load moment of inertia

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. [5-11]).
5. Save the parameter set (device command: C00002/11).

5.6.5 Remedies for undesired drive behaviour

Drive behaviour	Remedy
Deviation between no-load current and magnetising current or bad speed or torque accuracy.	Adapt the motor magnetising inductance (C00092) for no-load operation. • 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. • Otherwise, the magnetising inductance must be increased. Tendency of the correction of C00092:  PN: Rated motor power
Insufficient speed constancy at high load: Setpoint and motor speed are not proportional anymore. **Caution:** Overcompensation of the settings mentioned under " Remedy" may result in unstable behaviour!	Via the slip compensation (C00021), the speed stability under high loads can be affected: • If $n_{act} > n_{slip}$, reduce the value in C00021 • If $n_{act} < n_{slip}$, increase the value in C00021
Unstable control with higher speeds.	• Check the setting of the magnetising inductance (C00092) by comparing the current consumption in no-load operation with the rated magnetising current (C00095). • Optimise oscillation damping (C00234).
"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).	• Increase the gain of the torque controller (C00073/2). • Reduce the reset time of the torque controller (C00074/2). • Increase the acceleration (C00012)/deceleration (C00013) time.
Mechanical resonance at certain speeds.	The L_NSet_1 function block masks out those speed ranges that include resonance.
Speed variations in no-load operation for speeds > 1/3 rated speed.	Minimise speed oscillations with oscillation damping (C00234).
Drive runs unstable.	Check set motor data (nameplate data and equivalent circuit diagram data). ▸ Motor selection/Motor data (📖 99)
Setpoint speed and actual speed differ strongly.	
The torque required is not generated at standstill.	Increase motor magnetising current (C00095).
Current overshoots occur when heavy loads are accelerated from standstill (OC1 or OC11 error).	
The machine runs uneven.	

5.7 Parameterisable additional functions

5.7.1 Selection of the 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 (📖 111)

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: • "var.": Adaptation of the switching frequency depending on the current • "drive-opt.": drive-optimised modulation ("sine/delta modulation") • "fixed": fixed switching frequencies • "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 protec 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

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.

5.7.2 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!

- For a correct functioning of the flying restart circuit, we recommend to perform a parameter identification first. Automatic motor data identification (104)
- 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) (156)

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 ($n \geq 0$ rpm)
   - negative speed range ($n \leq 0$ 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 the flying restart function to accelerate 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)

[5-12] Formulae for the calculation of the gain and reset time of the current controller

5.7.3 DC-injection braking

Danger!

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". (📖 306)

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 • Operating threshold for activating DC-injection braking	3	rpm
C00036	DCB: Current • Braking current in [%] based on rated device current (C00098)	50	%
C00106	Auto-DCB: Hold time	0.500	s
C00107	DCB: Hold time	999.000	s
C00701/4	LA_NCtrl: bSetDCBrake • Selection of the signal source for activating DC-injection braking	Dependent on the selected control mode	

Procedure

DC-injection braking can be carried out in two ways with different types of activation:

- Manual DC-injection braking (DCB) (📖 156)
- Automatic DC-injection braking (auto DCB) (📖 156)

5.7.3.1 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.

5.7.3.2 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. (📖 306)

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 (📖 152)
   - 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.

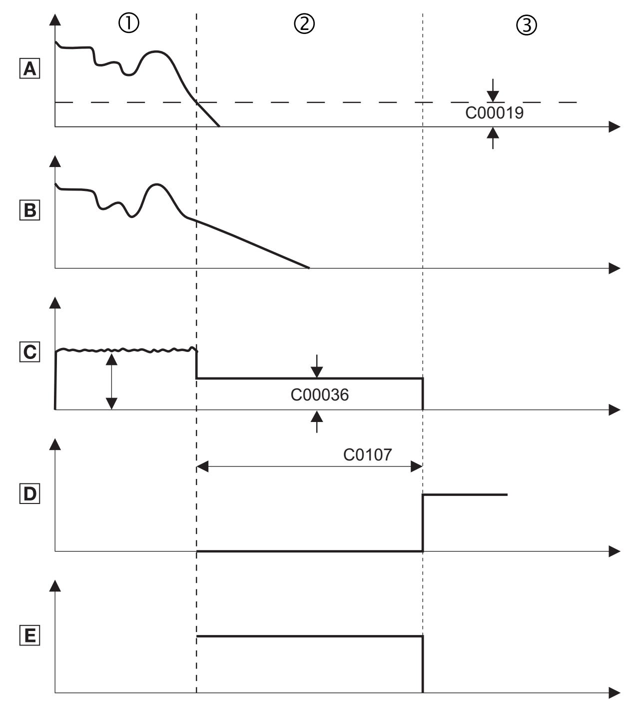
If you want to use the auto DCB function contrary to our recommendation (see above), the auto DCB threshold must not fall below the following values depending on the number of encoder increments (C00420):

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



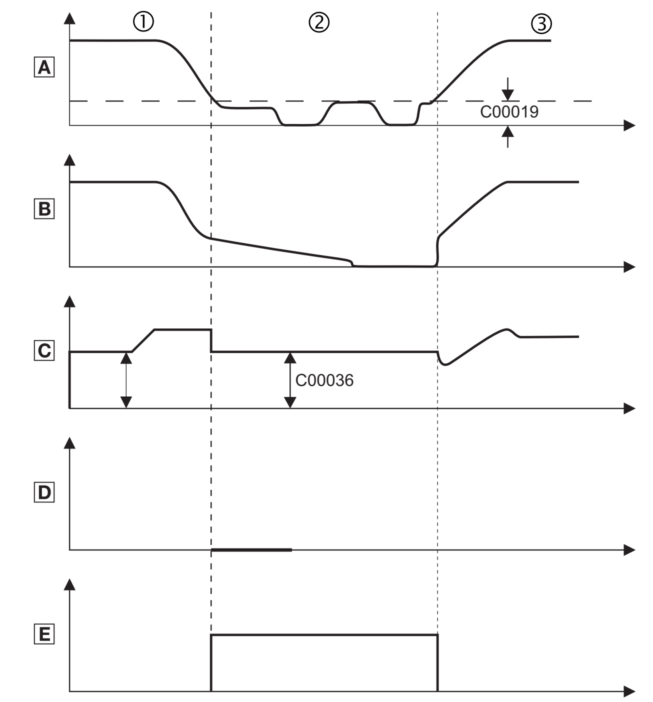
A Speed setpoint
B Actual speed value of the motor
C Output current of the controller
D Pulse inhibit
E DC-injection braking is active

[5-13] Example 1: Signal characteristic for automatic DC-injection braking of a drive without speed feedback

① The motor rotates at a specified speed. The current adjusts itself to the load, see C.

② The DC braking current set in C00036 is injected.

③ After the hold time (C00106) has expired, a pulse inhibit is set.



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.

[5-14] Example 2: Signal characteristic for automatic DC-injection braking of a drive with speed feedback

5.7.4 Slip compensation

Note!

Slip compensation is only active with the following motor control modes:
- V/f characteristic control (VFCplus) (📖 114)
- Sensorless vector control (SLVC) (📖 136)

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 (📖 104)
- 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.

5.7.5 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". (📖 162)
- 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". (📖 163)

5.7.5.1 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 (📖 133)

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

5.7.5.2 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 • 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.

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

5.7.6 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.

5.8 Encoder/feedback system

For motor control with speed feedback, the feedback signal can be fed to the digital terminals (DI1/DI2) via an HTL encoder.

Danger!

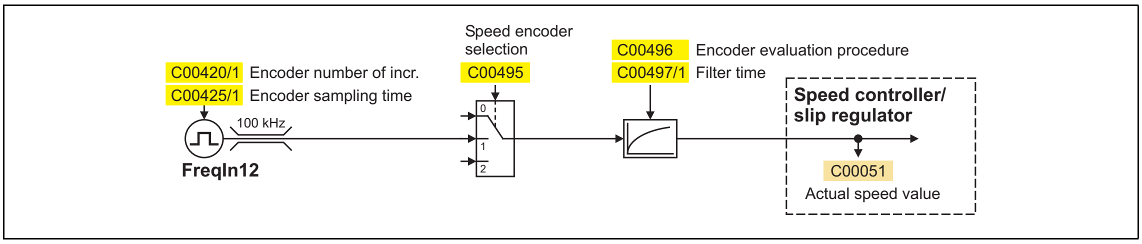
- To avoid interference when using an encoder, only use shielded motor and encoder cables.
- If an HTL encoder is used at the digital input terminals:
  Observe the maximum input frequencies of the digital inputs!
  - DI1/DI2: max. 100 kHz

Note!

In the Lenze setting (e.g. when the device is delivered), the open-circuit monitoring of the encoder is activated. ▶ Encoder open-circuit monitoring (📖 188)

Wiring diagram, assignment and electrical data of the digital input terminals can be found in the **8400 protec hardware manual** in the chapter "Technical data".

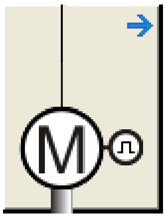
- The hardware manual has been stored in electronic form on the data carrier supplied with the 8400 drive controller.



[5-15] Signal flow - encoder interface

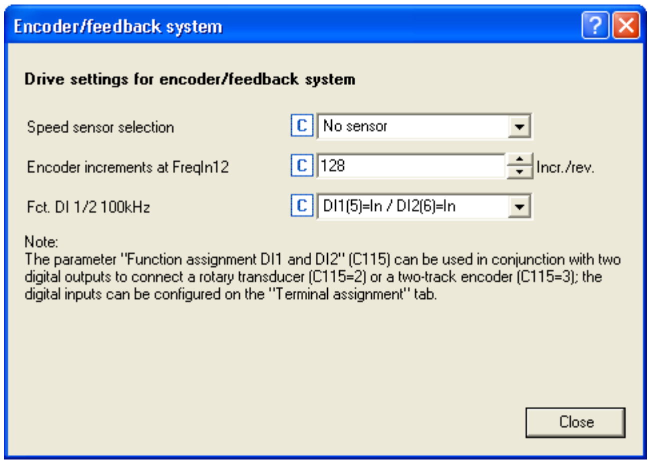
How to get to the parameterisation dialog of the encoder/feedback system:

1. In the »Engineer«, go to *Project view* and select the 8400 controller.
2. Select the **Application parameters** tab from the *Workspace*.
3. Click the following button of the *Overview* dialog level:



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
C00115/1	Fct. DI 1/2 100kHz • Function of the digital inputs DI1 and DI2	DI1=In1 | DI2=In2	
C00420/1	Encoder increments at FreqIn12 • If the digital inputs DI1 and DI2 are used as encoder inputs.	128	Inc/rev.
C00425/1	Encoder sample time FreqIn12 • If the digital inputs DI1 and DI2 are used as encoder inputs.	10	ms
C00495	Speed sensor selection • Source of feedback signal for speed control.	No sensor	
C00496	▶ Encoder evaluation method (📖 168)	Combined encoder method	
C00497/1	Encoder filter time FreqIn12 • If the digital inputs DI1 and DI2 are used as encoder inputs.	1.0	ms
C00586	Resp. to encoder open circuit ▶ Encoder open-circuit monitoring (📖 188)	Fault	

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.

5.8.1 Parameterising digital inputs as encoder inputs

The function of the digital inputs DI1/DI2 is defined in C00115/1.

To be able to use the digital inputs as encoder inputs, select 2, 3 or 4 (Lenze recommendation: 2) in C00115/1 depending on the input terminals used.

Selection in C00115/1	Function
2: (DI1/DI2)=FreqIn12 (2-track)	DI1 and DI2 = 2-track frequency input • Permits a two-track evaluation of the encoder including correct detection of the direction of rotation.
3: (DI1/DI2=+-)=FreqIn12	DI1 = 1-track frequency input DI2 = direction
4: DI1=CountIn1 | DI2=In2	DI1 = counter input DI2 = digital input

Danger!

For single-track evaluation, make sure that the sign is correctly specified. Otherwise, the motor may overspeed.

Note!

If the digital inputs are parameterised as encoder inputs, the corresponding output signals (*bIn1*/*bIn2*) at the LS_DigitalInput system block are automatically set to FALSE.

The wiring diagram and the assignment of the input terminals can be found in the **8400 protec hardware manual**.

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

Related topics:

▸ Digital terminals (📖 198)
▸ Using DI1(5) and DI2(6) as frequency inputs (📖 202)

5.8.2 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
1: Low-resolution encoder (Lenze setting)	High-precision procedure for low-resolution encoders (<=128 increments) • Exact method for speed measurement with automatic scanning time setting (0.5 ... 500 ms) for low-resolution encoders in the range of 4 .... 128 increments. • Evaluation with automatic scanning time minimisation for an optimum dynamic performance. • Method is also suited for encoders with poor signal quality, e.g. for encoders with high error rate in scanning ratio and phase offset. • This method requires an equidistant period length per encoder increment. • 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) • Speed measurement by means of the edges of tracks A and B measured per scanning interval. • Integrated correction algorithm for EMC interference. • Limited suitability for systems with unshielded encoder and/or motor cable. • 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 procedure for low-resolution encoders (C00496 = 1). This procedure can also be used for dynamic applications (e.g. V/f + encoder).

5.8.3 Encoder with HTL level 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 method (C00496 = 1), 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

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

5.9 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 described in subchapter "Avoiding thermal overload of the brake resistor". (178)

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 (347)

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.

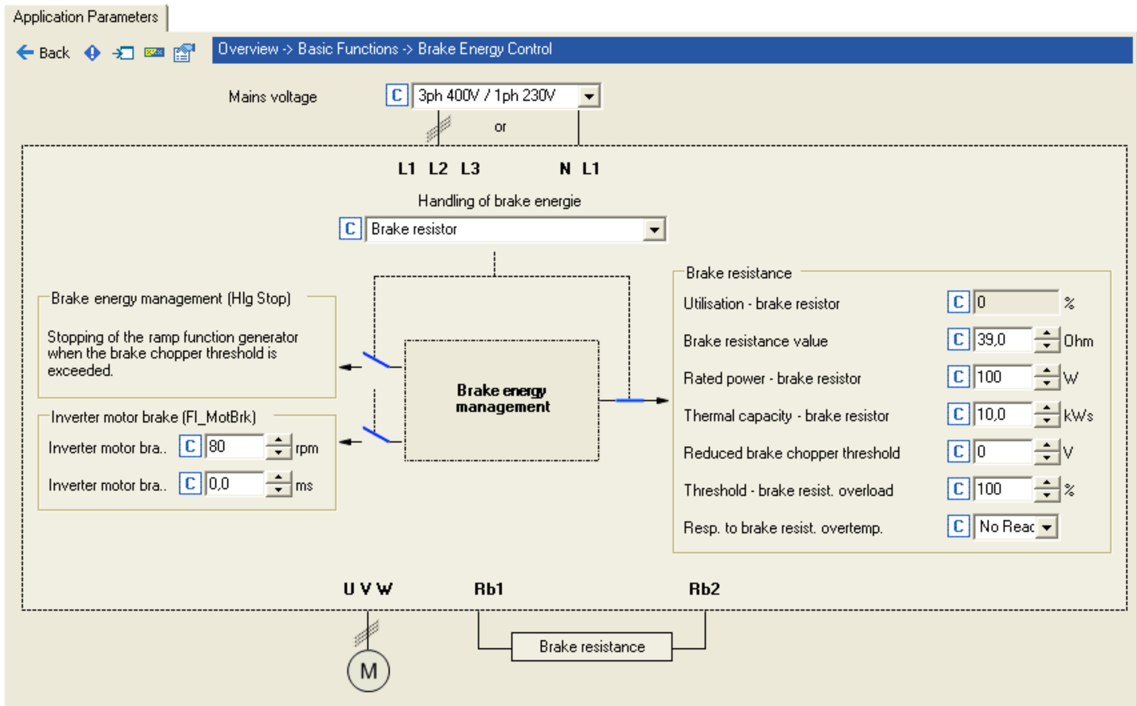
Tip!

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 (173)

Proceed as follows to open the dialog for parameterising the brake energy management:

1. »Engineer« Go to the *Project view* and select the 8400 protec StateLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
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)	
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	Threshold - brake resist. overload	100	%
C00574	Resp. to brake resist. overtemp.	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			

5.9.1 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 **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).

5.9.2 Selecting the response to an increase of the DC-bus voltage

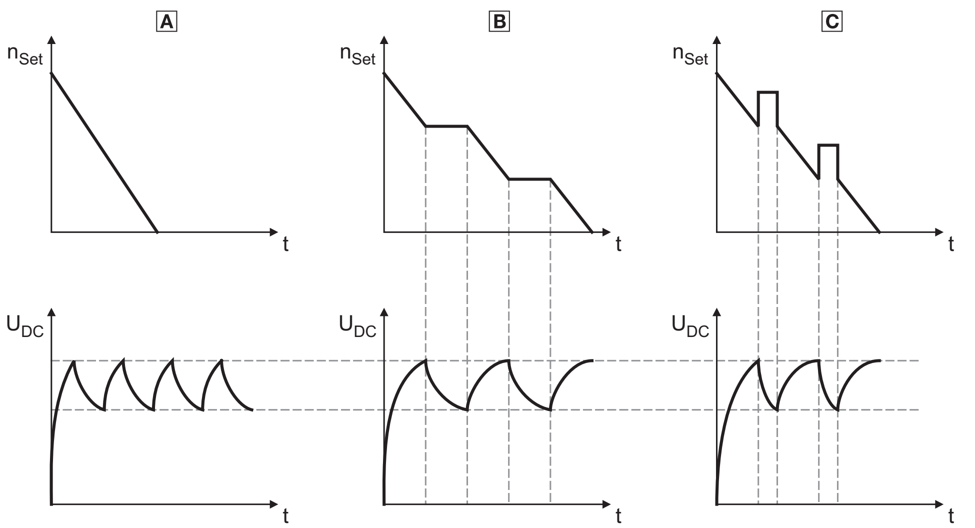
If the brake chopper threshold resulting from C00173 and C00174 is exceeded in the DC bus, the reaction selected in C00175 takes place (use of the brake resistor and/or stop of the ramp function generator and/or inverter-motor brake).

- 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 (📖 175)

STOP **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)

[5-16] 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").

5.9.2.1 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 (📖 155)

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:

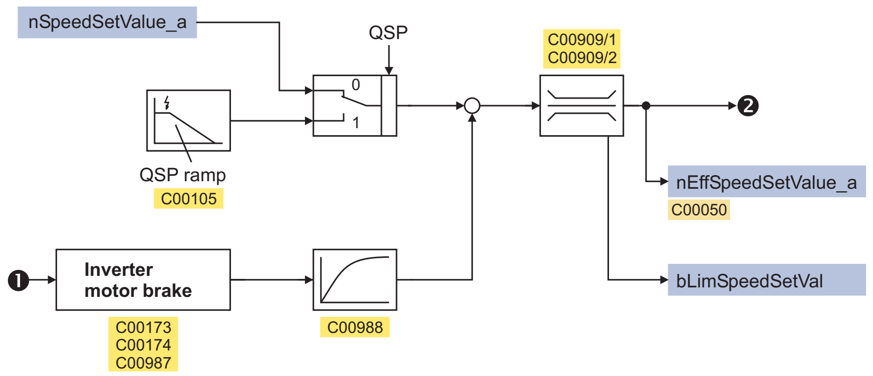
- For all applications that do not require braking to a standstill (e.g.braking to a lower speed setpoint) or the braking process can be interrupted by selecting a new speed setpoint.
- For applications with low mass inertias and a short braking time (< 1 s).
- For all applications where braking should 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

[5-17] 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:

$$C00987\,[\mathrm{rpm}] = 1 \ldots 4 \cdot (n_{Sync}[\mathrm{rpm}] - n_{Rated}[\mathrm{rpm}])$$

$$n_{Sync}[\mathrm{rpm}] = \frac{f_{Rated}\mathrm{Hz} \cdot 60}{p}$$

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

[5-18] 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 • 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 • 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):

- 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):

- 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.

5.9.3 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)". (📖 184)
- External interconnection using the thermal contact on the brake resistor (e.g. supply interruption via the mains contactor and activation of the mechanical brakes).

5.10 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)	Fault	-	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
Mains phase failure monitoring	Warning	C00565	Su02
Maximum current monitoring	No Reaction	C00609	OC7
Maximum torque monitoring	No Reaction	C00608	OT1
Encoder open-circuit monitoring	Fault	C00586	SD3

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 states (📖 79)
▶ Diagnostics & error management (📖 322)
▶ Basics on error handling in the controller (📖 322)
▶ Error messages of the operating system (📖 347)

5.10.1 Device overload monitoring (Ixt)

C00064/1...3 displays the device utilisation (ixt) in [%] in different time intervals:

Parameter	Info
C00064/1	Device utilisation (Ixt) • Maximum value of pulse utilisation (C00064/2) and permanent utilisation (C00064/3).
C00064/2	Device utilisation (Ixt) 15s • Pulse utilisation over the last 15 seconds (only for loads >160 %).
C00064/3	Device utilisation (Ixt) 3 min • 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 "Fault" error response is returned.
  - 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.

5.10.2 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

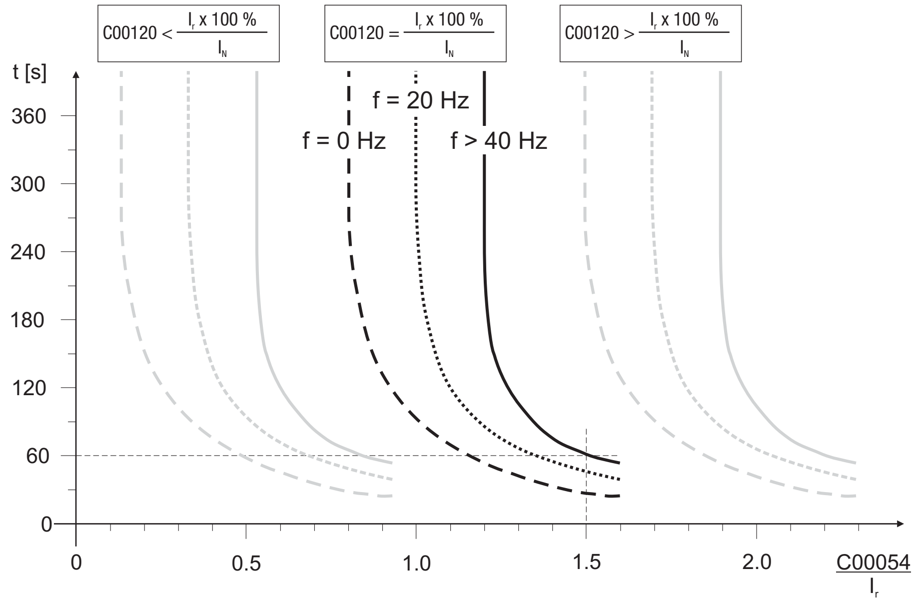
The motor utilisation meter for indicating the motor load in C00066 begins to count when the apparent motor current (C00054) is greater than the motor overload setting (C00120).

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

[5-19] Tripping characteristic of the I^2xt 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^2xt calculation only in an indirect way. However, the operation of the motor at maximum possible load can be averted. ▶ Defining current and speed limits (📖 111)

5.10.3 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) triggered" 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.

5.10.4 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.

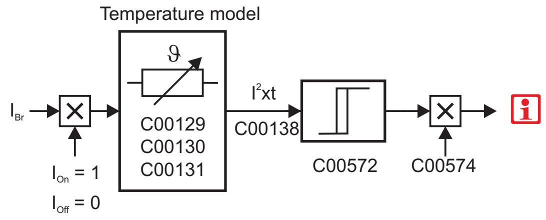
Stop!

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"

[5-20] 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.

- The calculation considers the thermal utilisation of the brake resistor based on the following parameters:
  - 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 (📖 170)

5.10.5 Motor phase failure monitoring

Note!

In the Lenze setting (C00597 = "0: No Reaction"), 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 detection 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.

Note!

If an error response of "1: Fault" is set in C00597, the *bMctrlMotorPhaseFault* status output of the LS_DeviceMonitor SB will be set to TRUE for only 1 second in the event of a motor phase failure because it is no longer possible to detect a motor phase fault via the error response with a pulse inhibit. However, the Logbook and C00561/3...5 still display the cause of the motor phase failure.

- The motor phase failure detection 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.

5.10.6 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: Mains voltage switched-off" error message is entered into the logbook.
  - The *bMctrlMainsFault* status output of the LS_DeviceMonitor system block will be set to TRUE.

5.10.7 Maximum current monitoring

Note!

In the Lenze setting (C00609 = "0: No Reaction"), the maximum current monitoring is not activated!

The maximum currents that can be parameterised in C00022 and C00023 are limit values that serve to protect the motor from destruction and the influence of the rated data.

- If the instantaneous value of the motor current exceeds one of the two limit values, the response set in C00609 will be carried out to protect the motor (Lenze setting: "0: No Reaction"). If the monitoring is active, the "OC10" error message will be entered in the logbook.

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.

5.10.8 Maximum torque monitoring

Note!

In the Lenze setting (C00608 = "0: No Reaction"), 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 Reaction").

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.

5.10.9 Encoder open-circuit monitoring

Note!

In the Lenze setting (C00586 = "1: Fault"), open-circuit monitoring of the encoder is activated!

When does the open-circuit monitoring system respond?

The open-circuit monitoring will trigger if

- an open circuit occurs in the encoder cable.
- an extreme overload (e.g. blocked motor shaft) occurs during the start-up 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 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

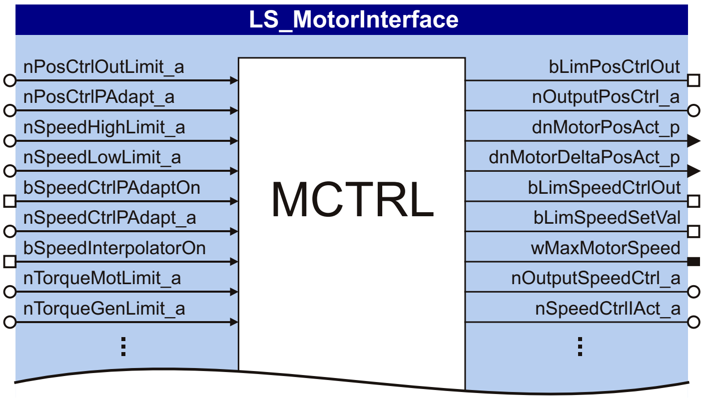
- If the open-circuit monitoring is tripped:
  - The error response set in C00586 is activated (Lenze setting: "Fault").
  - The "SD3: Open circuit - feedback system" is entered into the Logbook.
  - The *bMctrlEncoderComFault* status output of the LS_DeviceMonitor SB is set to TRUE.
- A setting of C00586 = "0: No Reaction" deactivates the monitoring.

Related topics:

▶ Encoder/feedback system (📖 165)

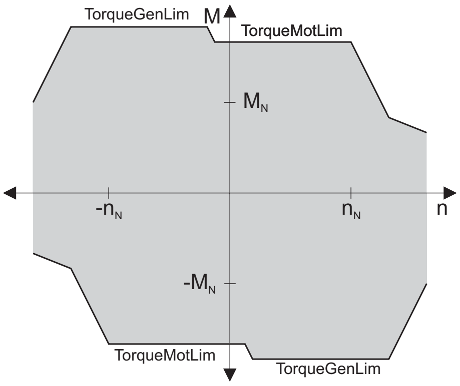
5.11 Internal interfaces | System block "LS_MotorInterface"

The **LS_MotorInterface** system block provides the internal interfaces to the driving machine in the function block editor.



Inputs

Identifier DIS code | data type	Information/possible settings	
nPosCtrlOutLimit_a C00830/21 | INT	Limitation of the position controller output • Scaling: 16384 ≡ 100 % reference speed (C00011)	
nPosCtrlPAdapt_a C00830/20 | INT	Adaptation of the position controller gain • Scaling: 16384 ≡ 100 % Vp position controller	
nSpeedHighLimit_a C00830/88 | INT	Upper speed limit for the speed limitation • During torque-controlled operation only (*bTorquemodeOn* = TRUE) • Scaling: 16384 ≡ 100 % reference speed (C00011)	
nSpeedLowLimit_a C00830/23 | INT	Lower speed limit for speed limitation • During torque-controlled operation only (*bTorquemodeOn* = TRUE) • Scaling: 16384 ≡ 100 % reference speed (C00011)	
bSpeedCtrlPAdaptOn C00833/69 | BOOL	Adaptation of the speed controller gain	
	FALSE	Deactivate adaptive adaptation.
	TRUE	Activate adaptive adaptation.
nSpeedCtrlPAdapt_a C00830/25 | INT	Adaptation of the speed controller gain • Scaling: 16384 ≡ 100 % Vp (C00070)	
bSpeedInterpolatorOn C00833/28 | BOOL	Speed setpoint interpolation	
	FALSE	Deactivate interpolation
	TRUE	Activate interpolation

Identifier DIS code | data type	Information/possible settings	
nTorqueMotLimit_a C00830/29 | INT nTorqueGenLimit_a C00830/28 | INT	Torque limitation in motor mode and in generator mode • The drive cannot output a higher torque in motor/generator mode than set here. • The applied values (any polarity) are internally interpreted as absolute values. • If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called I_{max} controller. • If sensorless vector control (SLVC) is selected, the limitation has a direct effect on the torque-producing current component. • Scaling: 16384 ≡ 100 % M_{max} (C00057)	
	Torque limits in motor and generator mode:  	
bTorqueInterpolatorOn C00833/29 | BOOL	Torque setpoint interpolation	
	FALSE	Deactivate interpolation
	TRUE	Activate interpolation
nVoltageAdd_a C00830/31 | INT	Additive voltage impression • An additional setpoint for the motor voltage can be specified via this process input. • If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time. • If the value is negative, the voltage is reduced. • Scaling: 16384 ≡ 1000 V	
	STOP **Stop!** Values selected too high may cause the motor to heat up due to the resulting current!	
bAutoBoostOn C00833/32 | BOOL	AutoBoost function • Motor voltage boost during the starting torque, controlled by process signals from the function block interconnection.	
	FALSE	Deactivate function
	TRUE	Activate function
nBoost_a C00830/26 | INT	Additional setpoint for the motor voltage at speed = 0 • The entire voltage-frequency characteristic is provided with an offset. • Scaling: 16384 ≡ 1000 V	
	STOP **Stop!** Values selected too high may cause the motor to heat up due to the resulting current!	

<table>
<tr><th>Identifier
DIS code | data type</th><th colspan="2">Information/possible settings</th></tr>
<tr><td>bPosCtrlOn</td><td colspan="2" rowspan="7">Inputs have no function on the 8400 protec StateLine!</td></tr>
<tr><td>bDeltaPosOn</td></tr>
<tr><td>dnDeltaPos_p</td></tr>
<tr><td>dnPosSetValue_p</td></tr>
<tr><td>bPosDerivativeOn</td></tr>
<tr><td>bMotorRefOffsetOn</td></tr>
<tr><td>dnMotorRefOffset_p</td></tr>
<tr><td rowspan="3">bQspOn
C00833/33 | BOOL</td><td colspan="2">Quick stop</td></tr>
<tr><td>FALSE</td><td>Deactivate quick stop</td></tr>
<tr><td>TRUE</td><td>Activate quick stop</td></tr>
<tr><td>nPWMAngleOffset_a
C00830/32 | INT</td><td colspan="2">Angle step change of output voltage phasor
• Scaling: 65535 ≡ 1 revolution</td></tr>
<tr><td rowspan="2">bSpeedCtrlIOn
C00833/31 | BOOL</td><td colspan="2">Directly set the I-component of speed controller
• In order to statically specify a minimum torque, e.g. when a load is being lifted.</td></tr>
<tr><td>TRUE</td><td>Se the I-component of the speed controller to the value nSpeedCtrlI_a.</td></tr>
<tr><td>nSpeedCtrlI_a
C00830/24 | INT</td><td colspan="2">Value of the speed controller integrator
• Scaling depends on the selected motor control:
  • V/f control (VFCplus + encoder):
  16384 ≡ 100 % reference speed (C00011)
  • Vector control (SLVC):
  16384 ≡ 100 % M_{max} (C00057)</td></tr>
<tr><td>nSpeedSetValue_a
C00830/22 | INT</td><td colspan="2">Speed setpoint
• Scaling: 16384 ≡ 100 % reference speed (C00011)</td></tr>
<tr><td rowspan="3">bTorquemodeOn
C00833/30 | BOOL</td><td colspan="2">Selection: Speed/Torque control</td></tr>
<tr><td>FALSE</td><td>Speed control with torque limitation</td></tr>
<tr><td>TRUE</td><td>Torque control with speed limitation</td></tr>
<tr><td>nTorqueSetValue_a
C00830/27 | INT</td><td colspan="2">Torque setpoint / additive torque
• Scaling: 16384 ≡ 100 % M_{max} (C00057)</td></tr>
<tr><td rowspan="3">bDcBrakeOn
C00833/34 | BOOL</td><td colspan="2">Activate DC injection brake</td></tr>
<tr><td>FALSE</td><td>Deactivate DC-injection braking</td></tr>
<tr><td>TRUE</td><td>Activate DC-injection braking</td></tr>
<tr><td rowspan="2">bTorqueLimitAdaptOn
C00833/98 | BOOL</td><td colspan="2">Adaptation of torque limitation</td></tr>
<tr><td>TRUE</td><td>Activate adaptation of torque limitation.</td></tr>
<tr><td>nTorqueLimitAdapt_a
C00830/70 | INT</td><td colspan="2">Value for adaptation of torque limitation
• Scaling: 16384 ≡ 100 % nTorqueMotLimit_a and nTorqueGenLimit_a</td></tr>
</table>

Outputs

<table>
<tr><th>Identifier
DIS code | data type</th><th colspan="2">Value/meaning</th></tr>
<tr><td rowspan="2">bLimPosCtrlOut
BOOL</td><td colspan="2">"Position controller output inside the limitation" status signal</td></tr>
<tr><td>TRUE</td><td>The position controller output is internally limited</td></tr>
<tr><td>nOutputPosCtrl_a
INT</td><td colspan="2">Position controller output
• Scaling: 16384 ≡ 100 % reference speed (C00011)</td></tr>
<tr><td>dnMotorPosAct_p
DINT</td><td colspan="2">Current position of the motor shaft in [increments]</td></tr>
<tr><td>dnMotorDeltaPosAct_p
DINT</td><td colspan="2">Current following error in [increments]
• Following error = Difference between set position and actual position</td></tr>
</table>

Identifier DIS code | data type	Value/meaning	
bLimSpeedCtrlOut BOOL	"Speed controller or manipulating variable of the slip regulator inside the limitation" status signal	
	TRUE	The speed controller output is internally limited
bLimSpeedSetVal 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 C00011 | BOOL	Reference speed (C00011)	
nOutputSpeedCtrl_a INT	Speed or slip controller output • Scaling: 16384 ≡ 100 % reference speed (C00011)	
nSpeedCtrlIAct_a INT	Current value of speed controller integrator • Scaling depends on the selected motor control: • V/f control (VFCplus + encoder): 16384 ≡ 100 % reference speed (C00011) • Vector control (SLVC): 16384 ≡ 100 % M_{max} (C00057)	
nEffSpeedSetValue_a INT	Effective speed setpoint • Scaling: 16384 ≡ 100 % reference speed (C00011)	
nMotorSpeedAct_a C00051 | INT	Actual speed value • Scaling: 16384 ≡ 100 % reference speed (C00011)	
nMotorSpeedAct_v INT	Actual speed value • Scaling: 65535 ≡ 1 revolution	
nMotorFreqAct_a C00058 | INT	Current field frequency • Scaling: 16384 ≡ 327.68 Hz (24000 ≡ 480.00 Hz)	
bLimTorqueSetVal BOOL	"Setpoint torque inside the limitation" status signal	
	TRUE	The setpoint torque is internally limited
wMaxMotorTorque C00057	Maximum motor torque • Scaling: 100 = 0.01 Nm • From version 06.00.00: *wMaxMotorTorque* = 10 * M_{max} (C00057)	
nInputTorqueCtrl_a INT	Input value of the torque control (torque setpoint) • Scaling: 16384 ≡ 100 % M_{max} (C00057)	
nMotorTorqueAct_a C00056/2 | INT	Actual torque • 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.. • Scaling: 16384 ≡ 100 % M_{max} (C00057)	
nInputJerkCtrl_a INT	Input value of the jerk limitation • Scaling: 16384 ≡ 100 % M_{max} (C00057)	
bLimCurrentSetVal BOOL	"Current setpoint inside the limitation" status signal	
	TRUE	The current setpoint is internally limited
nStatorCurrentIS_a INT	Current stator current/effective motor current • Scaling: 16384 ≡ 100 % I_{max_mot} (C00022)	
nEffCurrentIq_a INT	Current torque-producing cross current • Scaling: 16384 ≡ 100 % I_{max_mot} (C00022)	
nReaktCurrentId_a INT	Current field-producing direct-axis current • Scaling: 16384 ≡ 100 % I_{max_mot} (C00022)	
nActualFluxx_a INT	Current magnetising current • Scaling: 16384 ≡ 100 % I_{max_mot} (C00022)	
nDCVoltage_a INT	Actual DC-bus voltage • Scaling: 16384 ≡ 1000 V	

Identifier DIS code | data type	Value/meaning	
nMotorVoltage_a INT	Current motor voltage/inverter output voltage • Scaling: 16384 ≡ 1000 V	
bQspActive BOOL	"Quick stop active" status signal	
	TRUE	Quick stop is active
bAutoDCBActive BOOL	"Automatic DC-injection braking active" status signal ▸ DC-injection braking (📖 155)	
	TRUE	Automatic DC-injection braking is active
bIdentificationActive BOOL	"Motor parameter identification active" status signal ▸ Automatic motor data identification (📖 104)	
	TRUE	Motor parameter identification is active
bFlyingSyncActive BOOL	"Flying restart function active" status signal ▸ Flying restart function (📖 152)	
	TRUE	Flying restart function is active
bHlgLoad BOOL	Control signal for an additional loading function of the ramp function generator • → L_NSet_1.*bExternalCINH* • 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 INT	Setpoint for an additional loading function of the ramp function generator • → L_NSet_1.*nCInhVal_a* • 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. • Scaling: 16384 ≡ 100 % reference speed (C00011)	
bHlgStop BOOL	Control signal for stopping the ramp function generator (L_NSet_1)	
	TRUE	Stop the ramp function generator

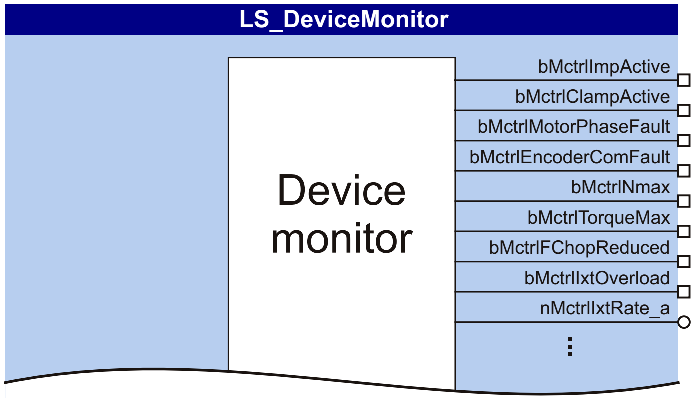
5.12 Internal status signals | System block "LS_DeviceMonitor"

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

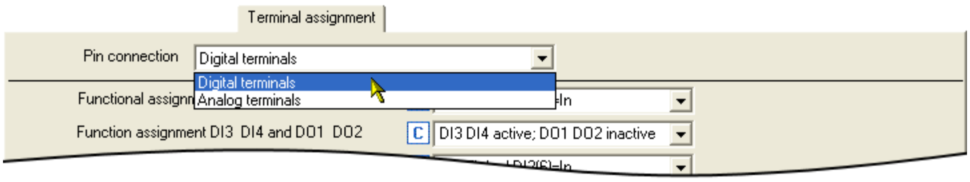
Identifier DIS code | data type	Value/meaning	
bMctrlImpActive BOOL	TRUE	Pulse inhibit is active
bMctrlClampActive BOOL	TRUE	Clamp current limitation is active
bMctrlMotorPhaseFault BOOL	TRUE	Motor phase fault has been detected
bMctrlEncoderComFault BOOL	TRUE	Encoder error has been detected
bMctrlNmax BOOL	TRUE	Max. speed limitation is active
bMctrlTorqueMax BOOL	TRUE	Max. torque limitation is active
bMctrlFChopReduced BOOL	TRUE	PWM frequency reduction is active
bMctrlIxtOverload BOOL	TRUE	Device utilisation (Ixt) ≥ device utilisation threshold (C00123) • Lenze setting: C00123 = 149.99 %
nMctrlIxtRate_a INT	Current device utilisation (Ixt) • Scaling: 16384 ≡ 100 %	
bMctrlI2xtOverload BOOL	TRUE	Thermal motor overload (I^2xt) ≥ motor overload setting (C00120) • Lenze setting: C00120 = 100 %.
nMctrlI2xtRate_a INT	Current thermal motor load (I2xt) • Scaling: 16384 ≡ 100 %	

Identifier DIS code | data type	Value/meaning	
bMctrlMotorPTC BOOL	TRUE	Temperature monitoring: An error has been detected
bMctrlMotorTemp BOOL	TRUE	Thermal motor overload
bMctrlHeatSinkTemp BOOL	TRUE	Thermal inverter overload • The heatsink temperature (display in C00061) has reached the maximally permissible temperature. The "oH1: Overtemperature heatsink" error message is pending and the "Fault" response is activated. • Furthermore, this output is set if the heatsink temperature has exceeded the maximally permissible temperature for the switching frequency set. A response with an "oH4: Heatsink temp.. > switch-off temp. -5°C" error message can be set in C00582. In the Lenze setting, no response will be effected.
bMctrlMainsFault BOOL	TRUE	Mains phase failure/Mains failure
bMctrlFanFault BOOL	TRUE	Fan monitoring: An error has been detected
bMctrlNmaxForFChop BOOL	TRUE	The maximum field frequency for the respective switching frequency has been exceeded.
bMctrlShortCircuit BOOL	TRUE	Motor short circuit has been detected
bMctrlEarthFault BOOL	TRUE	Earth fault has been detected
bMctrlUVDetected BOOL	TRUE	An undervoltage has been detected
bMctrlOVDetected BOOL	TRUE	An overvoltage has been detected
bMctrlBrakeChopper BOOL	TRUE	The Brake resistor monitoring (I2xt) has tripped. • This output is set independent of the set error response of the monitoring function.

6 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 terminals (📖 198)
- Analog terminals (📖 221)

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 (📖 247)
- **TA "Switch-off positioning":** Terminal assignment of the control modes (📖 274)

Wiring diagram, assignment and electrical data of the input and output terminals can be found in the **8400 protec 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". (📖 227)

6.1 Digital terminals

The controller is equipped with six parameterisable inputs (DI1 ... DI6) at the sockets X41 ... X43 for detecting digital signals.

Tip!

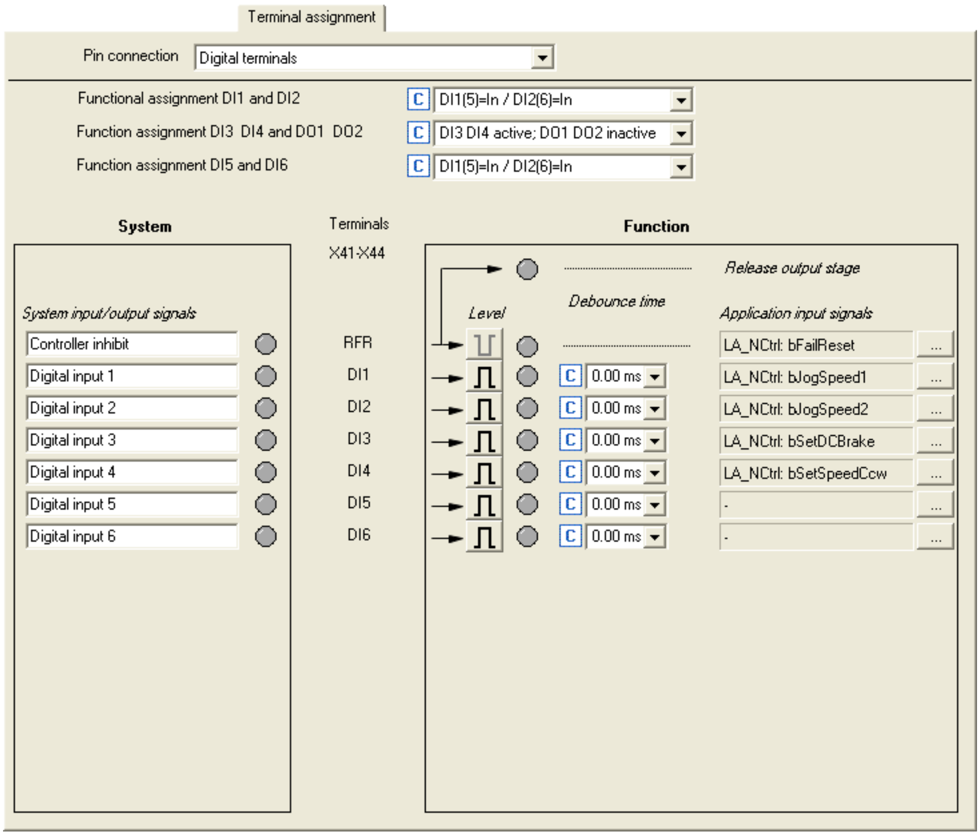
The two inputs DI3 and DI4 (pin 4 and pin 2 of the X42 socket) can be reconfigured alternatively to digital outputs (DO1/DO2) by a corresponding parameter setting.
▶ Reconfiguring DI3(4) to output DO1(2) (📖 211)

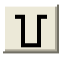
Note!

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.

Parameterisation dialog in the »Engineer«:



Button	Function
	Indicates the polarity of the input is HIGH active. 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. 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. ▶ Changing the terminal assignment with the »Engineer« (📖 231)

Short overview of the parameters for the digital terminals:

Parameter	Info	Lenze setting	
		Value	Unit
C00115/1	Fct. DI 1/2 100kHz ▶ Change function assignment (📖 200)	0: DI1(5)=In / DI2(6)=In	
C00115/2	Fct. DI 5/6 10kHz ▶ Change function assignment (📖 200)	0: DI1(5)=In / DI2(6)=In	
C00116	DI 3/4 DO 1/2: Function ▶ Reconfiguring DI3(4) to output DO1(2) (📖 211)	0: DI3 DI4 active; DO1 DO2 inactive	
Digital inputs DI1 ... DI6			
C00114	DIx: Polarity	Bit coded	
C02830/1...6	DI1...DI6: Debounce time	0: 0.00 ms	
C00443/1	DIx: Terminal level	-	
C00443/2	DIx: Output level	-	
Digital outputs DO1/DO2 ((only available if DI3/DI4 have been reconfigured)			
C00118	DOx: Inversion	Bit coded	
C00423/1	DO1 ON delay	0.000	s
C00423/2	DO1 OFF delay	0.000	s
C00423/3	DO2 ON delay	0.000	s
C00423/4	DO2 OFF delay	0.000	s
C00444/1	DOx: Input level	-	
C00444/2	DOx: Terminal level	-	
Highlighted in grey = display parameter			

Related topics:

▶ Configuring exception handling of the output terminals (📖 226)

▶ User-defined terminal assignment (📖 227)

6.1.1 Change function assignment

The internal processing function of the digital input terminals DI1/DI2 and DI5/DI6 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 DI1 and DI2 C00115/2: Function assignment of DI5 and DI6		Function assignment DI1 / DI5	Function assignment DI2 / DI6
0	**DI1(5)=In / DI2(6)=In**	**Digital input**	**Digital input**
1	DI1(5)=FreqIn / DI2(6)=In	Frequency input	Digital input
2	DI1(5)&DI2(6)=FreqIn (2-track)	Frequency input (2-track)	
3	DI1(5) = FreqIn / DI2(6)=direction	Frequency input (speed)	Frequency input (direction)
4	DI1(5)=CountIn / DI2(6)=In	Count input	Digital input

Note!

- In the Lenze setting of C00115, the digital input terminals DI1/DI2 and DI5/DI6 have been configured as "normal" digital inputs.
- The digital input terminals DI3 ... DI4 are basically designed as "normal" digital inputs or as digital outputs through reconfiguration.
- Very high pulse frequencies can be measured at the DI1/DI2 and DI5/DI6 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:

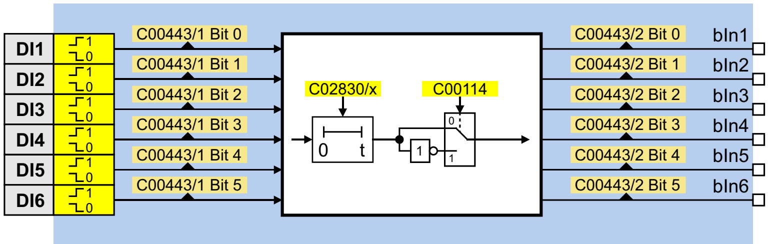
▶ Using DI1(5) and DI2(6) as digital inputs (📖 201)
▶ Using DI1(5) and DI2(6) as frequency inputs (📖 202)
▶ Using DI1(5) as counting input (📖 207)
▶ Reconfiguring DI3(4) to output DO1(2) (📖 211)

6.1.1.1 Using DI1(5) and DI2(6) as digital inputs

Function assignment 0: DI1(5)=In / DI2(6)=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...6) 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 DIS code | data type	Value/meaning
bIn1 ... bIn6 C00443/2 | BOOL	Digital input DI1 ... DI6

Related topics:

▸ Using DI1(5) and DI2(6) as frequency inputs (📖 202)
▸ Using DI1(5) as counting input (📖 207)
▸ Reconfiguring DI3(4) to output DO1(2) (📖 211)
▸ Internal interfaces | System block "LS_DigitalInput" (📖 213)

6.1.1.2 Using DI1(5) and DI2(6) 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
  - DI5/DI6: f_{max} = 10 kHz
- If the encoder signal is used as an actual speed value:
  Number of encoder pulses / revolution ≤ **8192!**

Example for DI5/DI6 (in accordance with the preceding note):

- Number of encoder increments: 512 pulses / motor revolution
- Reference speed (C00011): 1500 rpm
- Speed setpoint: 100 %

$$\text{Input frequency} = \frac{1500\ \text{rpm}}{60\ \text{s}} \times 512\ \text{pulse} = 12800\ \text{pulse/s} = 12.8\ \text{kHz}$$

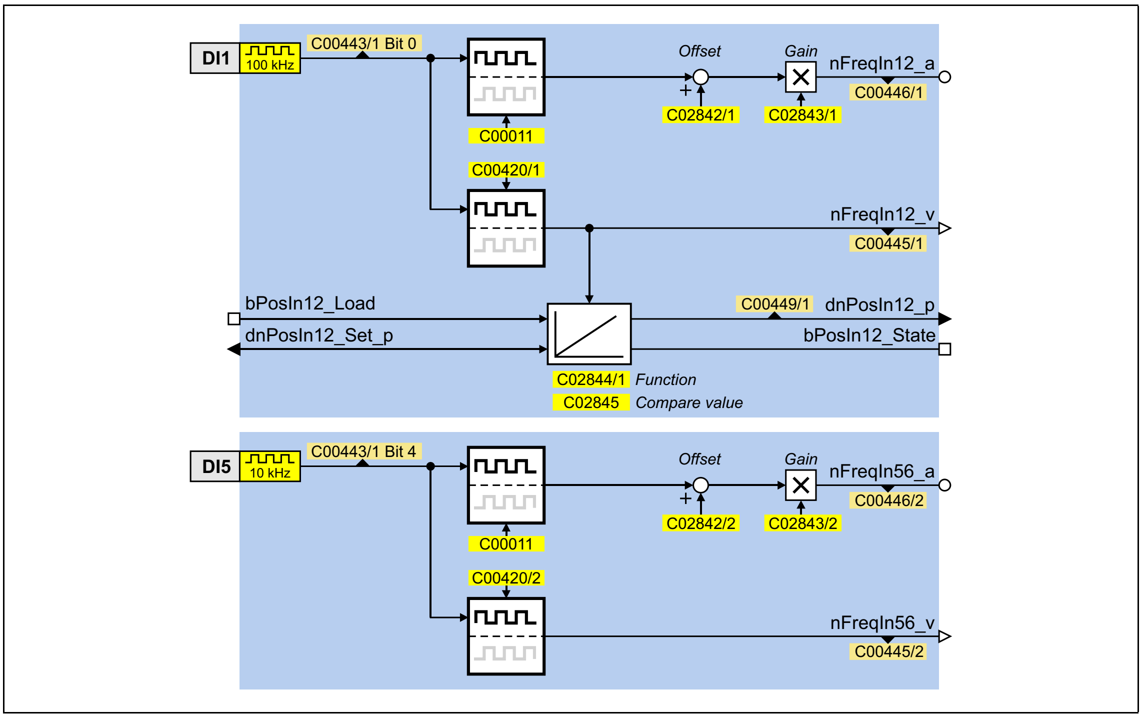
- Result: The speed or the number of increments is too high!

Tip!

From version 06.00.00 onwards, 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". (📖 216)

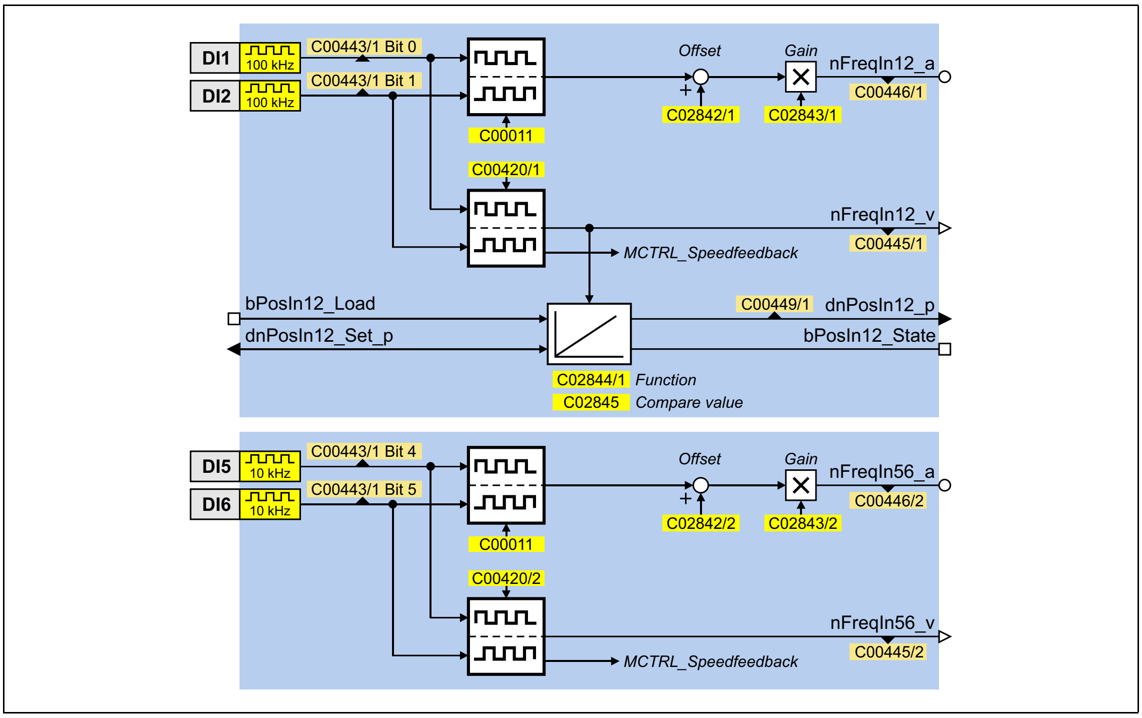
Function assignment 1: DI1(5)=FreqIn / DI2(6)=In

This setting in C00115 configures the input terminal DI1(5) as frequency input. The input terminal DI2 or DI6 remains configured as "normal" digital input.



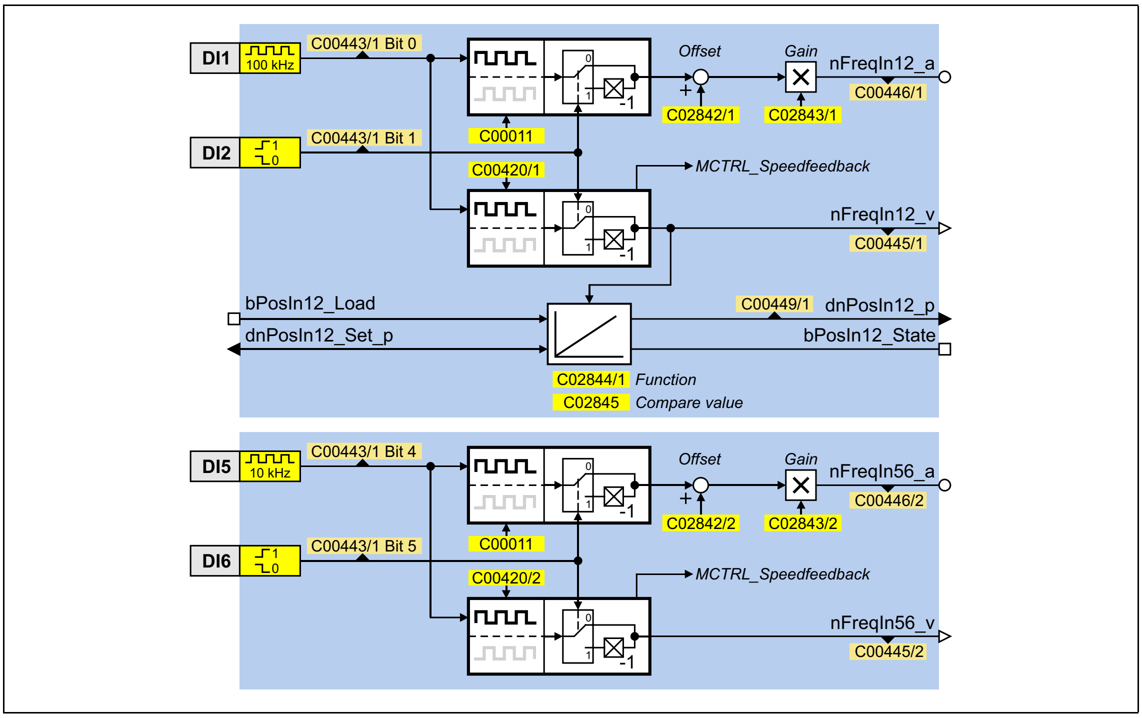
Function assignment 2: DI1(5)&DI2(6)=FreqIn (2-track)

This setting in C00115 can be used to connect a two-track encoder to the DI1/DI2 or DI5/DI6 terminals.



Function assignment 3: DI1(5)=FreqIn / DI2(6)=Direction

This setting in C00115 can be used to connect a single-track encoder to terminals DI1/DI2 or DI5/DI6. For this purpose, the rotation speed is evaluated via terminal DI1(5) and the direction of rotation of the encoder (LOW level ≡ CW direction of rotation) is evaluated via the DI(6) 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 100kHz	0: DI1(5)=In / DI2(6)=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 DI5/DI6			
C00115/2	Fct. DI 5/6 10kHz	0: DI1(5)=In / DI2(6)=In	
C00420/2	Encoder increments at FreqIn56	128	Incr./rev.
C02842/2	FreqIn56: Offset	0.00	%
C02843/2	FreqIn56: Gain	100.00	%
C00443/1	DIx: Terminal level	-	
C00445/2	FreqIn56_nOut_v	-	Incr/ms
C00446/2	FreqIn56_nOut_a	-	%

Highlighted in grey = display parameter

Internal interfaces to the application

- Relevant inputs at the LS_DigitalInput system block:

Input Data type	Information/possible settings	
Frequency input DI1/DI2		
bPosIn12_Load BOOL (from version 06.00.00)	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 DINT (from version 06.00.00)	Starting value for angle integrator	

- Relevant outputs at the LS_DigitalInput system block:

Output Data type	Value/meaning	
Frequency input DI1/DI2		
nFreqIn12_a C00446/1 | INT	Output frequency as scaled analog signal in [%]	
nFreqIn12_v C00445/1 | INT	Output frequency as speed signal in [inc/ms]	
dnPosIn12_p DINT (from version 06.00.00)	Angle output signal • 65536 [incr.] ≡ 1 encoder revolution • Overflow is possible (display via *bPosIn12_State*)	
bPosIn12_State BOOL (from version 06.00.00)	Status signal "Overflow occurred/distance processed" • Status signal can be reset via *bPosIn12_Load*.	
	TRUE	Overflow has occurred or distance is processed.
Frequency input DI5/DI6		
nFreqIn56_a C00446/2 | INT	Output frequency as scaled analog signal in [%]	
nFreqIn56_v C00445/2 | INT	Output frequency as speed signal in [inc/ms]	

Related topics:

▶ Output of the encoder position of the DI1/DI2 frequency input (🕮 216)
▶ Using DI1(5) and DI2(6) as digital inputs (🕮 201)
▶ Using DI1(5) as counting input (🕮 207)
▶ Internal interfaces | System block "LS_DigitalInput" (🕮 213)

6.1.1.3 Using DI1(5) 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.

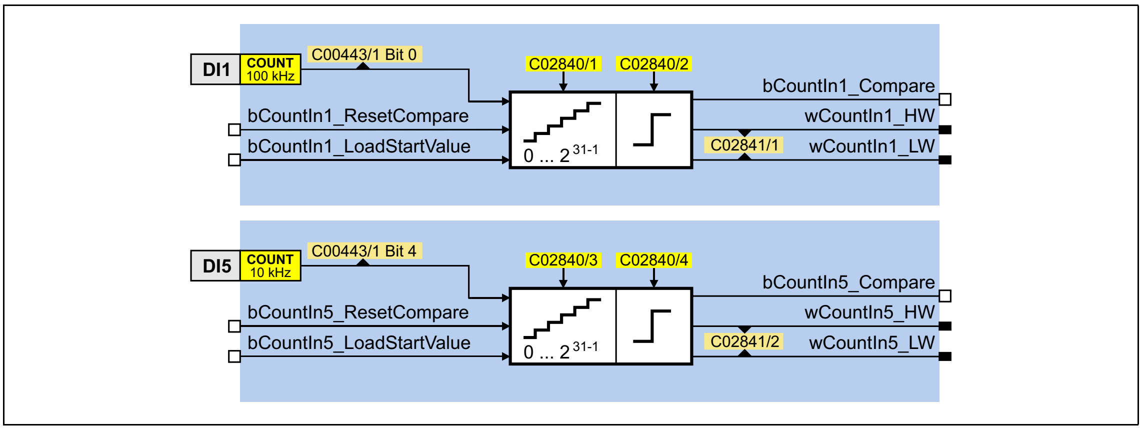
- Possible counting range: 0 ... 2^{31} - 1 (0 ... 2147483647)

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} = 100 kHz
  - DI5: f_{max} = 10 kHz

Function assignment 4: DI1(5)=CountIn / DI2(6)=In

This setting in C00115 configures the input terminal DI1 or DI5 as counting input. The input terminal DI2 or DI6 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 100kHz	0: DI1(5)=In / DI2(6)=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 DI5			
C00115/2	Fct. DI 5/6 10kHz	0: DI1(5)=In / DI2(6)=In	
C00621/97	LS_DigitalInput: bCountIn5_Reset	0: Not connected	
C00621/98	LS_DigitalInput: bCountIn5_LoadStartValue	0: Not connected	
C02840/3	CountIn5: Starting value	0	incr
C02840/4	CountIn5: Comparison value	65535	incr
C02841/2	CountIn5: 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:

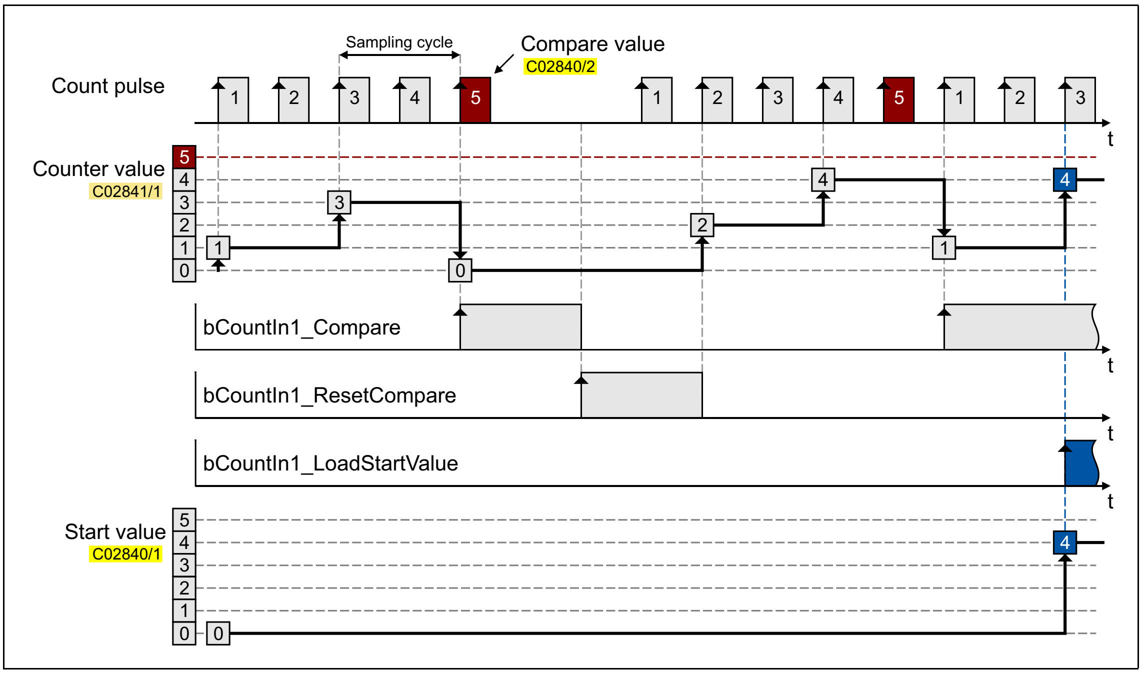
Identifier DIS code | data type	Information/possible settings	
Counting input DI1		
bCountIn1_ResetCompare BOOL	Reset status signal "Comparison value reached"	
	FALSE⇗TRUE	The output *bCountIn1_Compare* is reset to FALSE.
bCountIn1_LoadStartValue BOOL	Load starting value into counter	
	FALSE⇗TRUE	The starting value set in C02840/1 is accepted as the current count value.
Counting input DI5		
bCountIn5_ResetCompare BOOL	Reset status signal "Comparison value reached"	
	FALSE⇗TRUE	The output *bCountIn5_Compare* is reset to FALSE.
bCountIn5_LoadStartValue 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 DIS code | data type	Value/meaning	
Counting input DI1		
bCountIn1_Compare BOOL	Status signal "Comparison value reached"	
	FALSE	Current count value < comparison value (C02840/2)
	TRUE	Current count value ≥ comparison value (C02840/2)
wCountIn1_HW wCountIn1_LW C02841/1 | WORD	Current count value • Output as High and Low word (without sign) • Possible counting range: 0 ... 2^{31} - 1	
Counting input DI5		
bCountIn5_Compare BOOL	Status signal "Comparison value reached"	
	FALSE	Current count value < comparison value (C02840/4)
	TRUE	Current count value ≥ comparison value (C02840/4)
wCountIn5_HW wCountIn5_LW C02841/2 | WORD	Current count value • Output as High and Low word (without sign) • 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:



[6-1] Transient characteristic of a quick counter block, sampling cycle = 1 ms

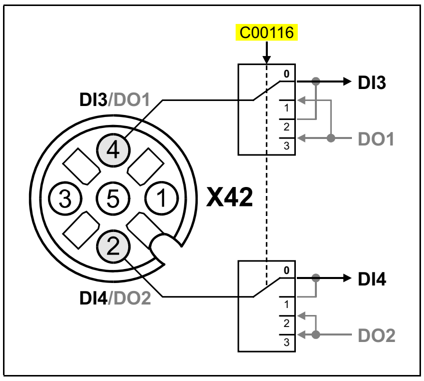
- The counter starts with the parameterised starting value.
- If the comparison value is reached or exceeded:
  - The counter jumps back to its starting value.
  - The output *bCount1(5)_Compare* is set to TRUE.
- If there is a FALSE-TRUE edge is at the input *bCountIn1(5)_ResetCompare*, the output *bCountIn1(5)_Compare* can be reset to FALSE.
- If there is a FALSE-TRUE edge at the input *bCountIn1(5)_LoadStartValue*, the current counter content can be reset to the parameterised starting value.

Related topics:

▶ Using DI1(5) and DI2(6) as digital inputs (📖 201)
▶ Using DI1(5) and DI2(6) as frequency inputs (📖 202)
▶ Internal interfaces | System block "LS_DigitalInput" (📖 213)

6.1.1.4 Reconfiguring DI3(4) to output DO1(2)

Pin 2 and pin 4 of the socket X42 can be used as digital inputs or outputs.

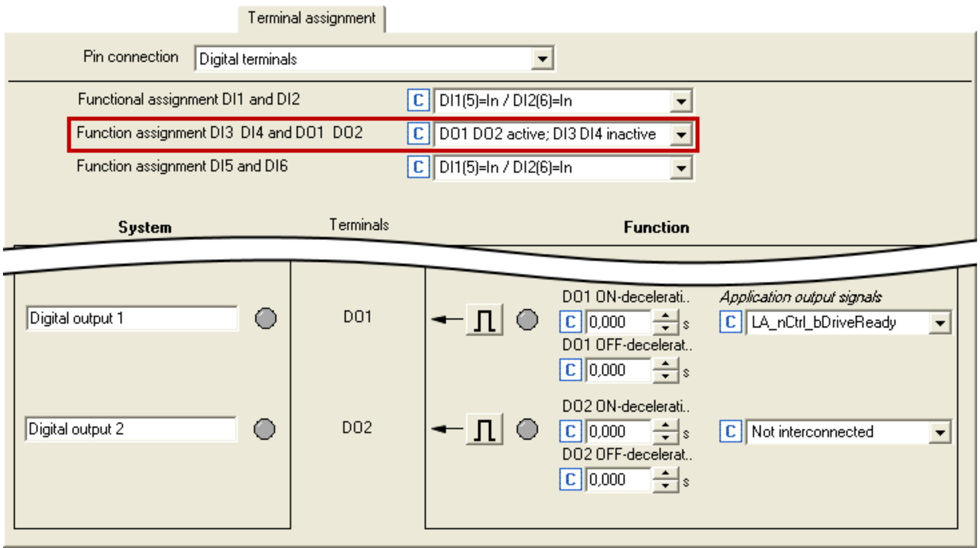


- In the Lenze setting, both pins are configured as digital input (DI3 and DI4).
- In order to reconfigure the function assignment, set the corresponding selection in C00116 (see the following table).

Setting C00116:		Active function assignment			
		DI3	DI4	DO1	DO2
0	**DI3 DI4 active; DO1 DO2 inactive**	●	●		
1	DO1 DI4 active; DI3 DO2 inactive		●	●	
2	DI3 DO2 active; DO1 DI4 inactive	●			●
3	DO1 DO2 active; DI3 DI4 inactive			●	●

Parameterisation dialog in the »Engineer«:

After the reconfiguration, the outputs DO1/DO2 are displayed with the corresponding parameters on the **Terminal assignment** tab instead of the inputs DI3/DI4:



Short overview of the parameters for the digital outputs

Parameter	Info	Lenze setting Value	Lenze setting Unit
Digital outputs DO1/DO2			
C00118	DOx: Inversion	Bit coded	
C00423/1	DO1 ON delay	0.000	s
C00423/2	DO1 OFF delay	0.000	s
C00423/3	DO2 ON delay	0.000	s
C00423/4	DO2 OFF delay	0.000	s
C00444/1	DOx: Input level	-	
C00444/2	DOx: Terminal level	-	
Digital outputs - terminal configuration			
C00621/2	LS_DigitalOutput:bOut1	1000: LA_nCtrl_bDriveReady	
C00621/99	LS_DigitalOutput: bOut2	0: Not connected	
Highlighted in grey = display parameter			

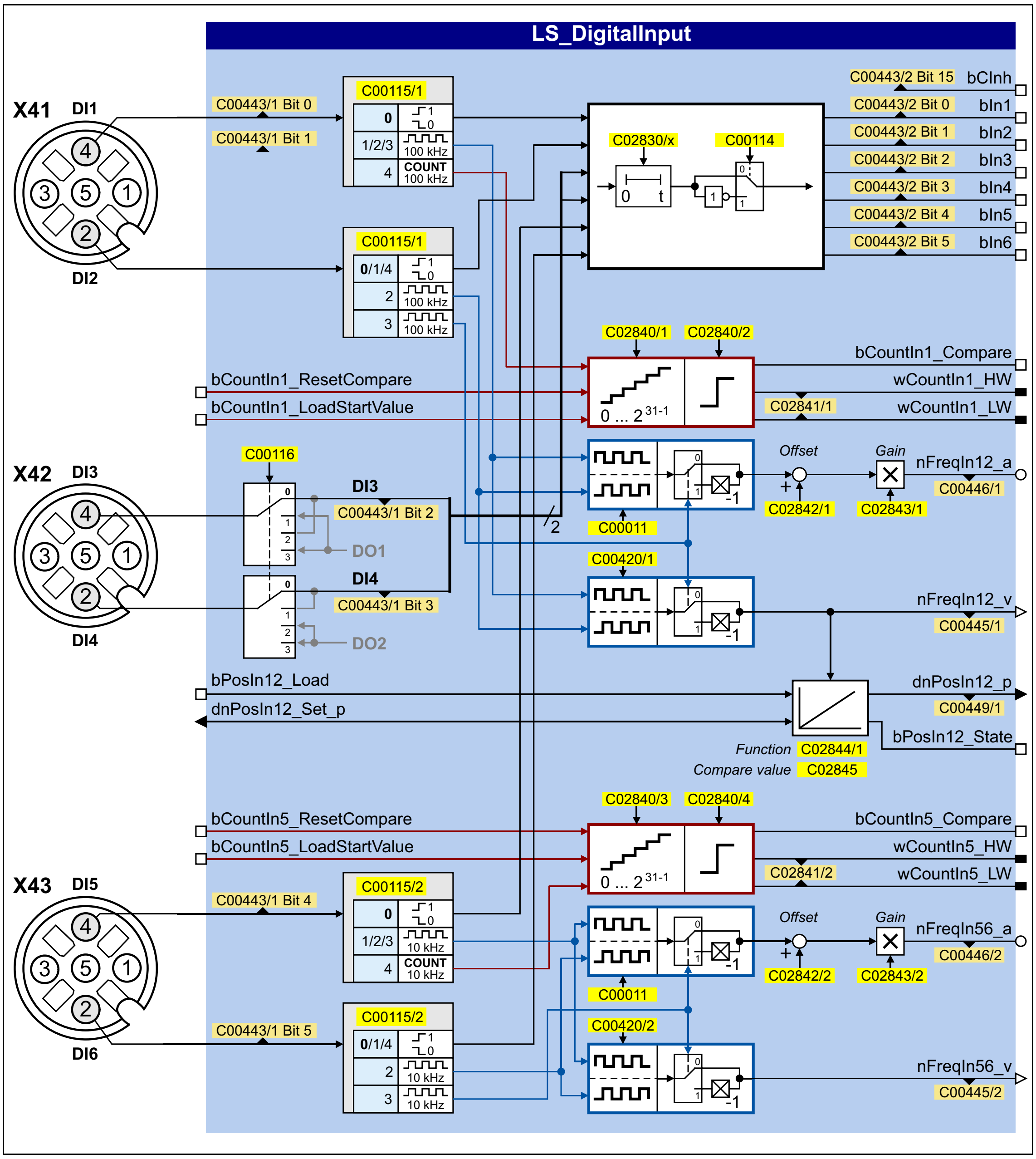
Related topics:

▸ Internal interfaces | System block "LS_DigitalOutput" (📖 220)
▸ Configuring exception handling of the output terminals (📖 226)
▸ User-defined terminal assignment (📖 227)

6.1.2 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/DI2 and DI5/DI6 input terminals can be reconfigured in C00115 if necessary. These input terminals can then be alternatively used as frequency inputs or counting inputs.
- The inputs DI3/DI4 are always designed as "normal" digital inputs or as digital outputs through reconfiguration in C00116.



Inputs

Identifier DIS code | data type	Information/possible settings	
Counting input DI1	▸ Using DI1(5) as counting input	
bCountIn1_ResetCompare BOOL	Reset status signal "Comparison value reached"	
	FALSE↗TRUE	The output *bCountIn1_Compare* is reset to FALSE.
bCountIn1_LoadStartValue 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 BOOL (from version 06.00.00)	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 DINT (from version 06.00.00)	Starting value for angle integrator	
Counting input DI5	▸ Using DI1(5) as counting input	
bCountIn5_ResetCompare BOOL	Reset status signal "Comparison value reached"	
	FALSE↗TRUE	The output *bCountIn5_Compare* is reset to FALSE.
bCountIn5_LoadStartValue BOOL	Load starting value into counter	
	FALSE↗TRUE	The starting value set in C02840/3 is accepted as the current count value.

Outputs

Identifier DIS code | data type	Value/meaning	
bCInh C00443/2 | BOOL	RFR digital input (controller enable)	
Digital inputs DI1 ... DI6	▸ Using DI1(5) and DI2(6) as digital inputs	
bIn1 ... bIn6 C00443/2 | BOOL	Digital input DI1 ... DI6	
Counting input DI1	▸ Using DI1(5) as counting input	
bCountIn1_Compare BOOL	Status signal "Comparison value reached"	
	FALSE	Current count value < comparison value (C02840/2)
	TRUE	Current count value ≥ comparison value (C02840/2)
wCountIn1_HW wCountIn1_LW C02841/1 | WORD	Current count value • Output as High and Low word (without sign) • Possible counting range: 0 ... 2^{31} - 1	
Frequency input DI1/DI2	▸ Using DI1(5) and DI2(6) as frequency inputs	
nFreqIn12_a C00446/1 | INT	Output frequency as scaled analog signal in [%]	
nFreqIn12_v C00445/1 | INT	Output frequency as speed signal in [inc/ms]	
dnPosIn12_p DINT (from version 06.00.00)	Angle output signal • 65536 [incr.] ≡ 1 encoder revolution • Overflow is possible (display via *bPosIn12_State*)	
bPosIn12_State BOOL (from version 06.00.00)	Status signal "Overflow occurred/distance processed" • Status signal can be reset via *bPosIn12_Load*.	
	TRUE	Overflow has occurred or distance is processed.

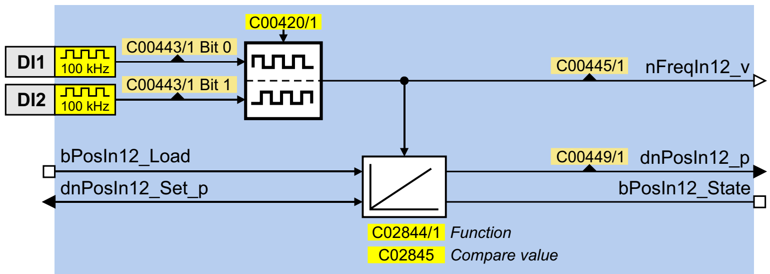
Identifier DIS code | data type	Value/meaning	
Counting input DI5	Using DI1(5) as counting input	
bCountIn1_Compare BOOL	Status signal "Comparison value reached"	
	FALSE	Current count value < comparison value (C02840/4)
	TRUE	Current count value ≥ comparison value (C02840/4)
wCountIn6_HW wCountIn6_LW C02841/2 | WORD	Current count value • Output as High and Low word (without sign) • Possible counting range: 0 ... 2^{31} - 1	
Frequency input DI5/DI6	Using DI1(5) and DI2(6) as frequency inputs	
nFreqIn56_a C00446/2 | INT	Output frequency as scaled analog signal in [%]	
nFreqIn56_v C00445/2 | INT	Output frequency as speed signal in [inc/ms]	

6.1.2.1 Output of the encoder position of the DI1/DI2 frequency input

This function extension is available from version 06.00.00!

The LS_DigitalInput system block has been extended by the integrator function 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 DIS code | data type	Information/possible settings	
bPosIn12_Load 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 DINT	Starting value for angle integrator	

Outputs

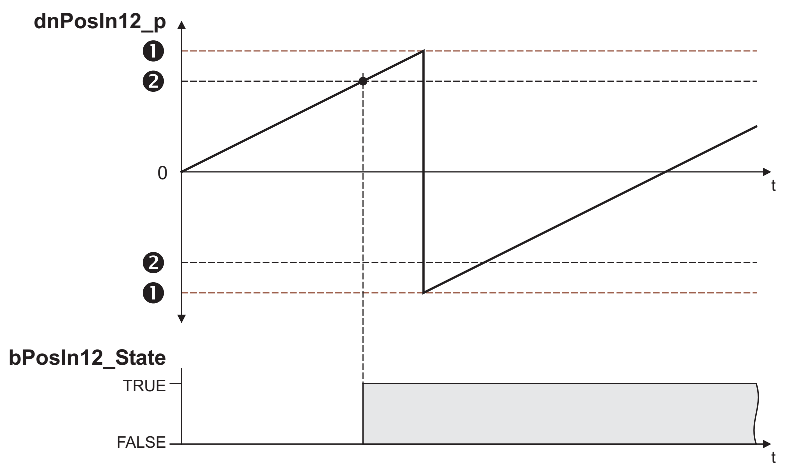
Identifier DIS code | data type	Value/meaning	
dnPosIn12_p DINT	Angle output signal • 65536 [incr.] ≡ 1 encoder revolution • Overflow is possible (display via *bPosIn12_State*)	
bPosIn12_State BOOL	Status signal "Overflow occurred/distance processed" • Status signal can be reset via *bPosIn12_Load*.	
	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 • Is valid for both the positive and the negative value range. • 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)

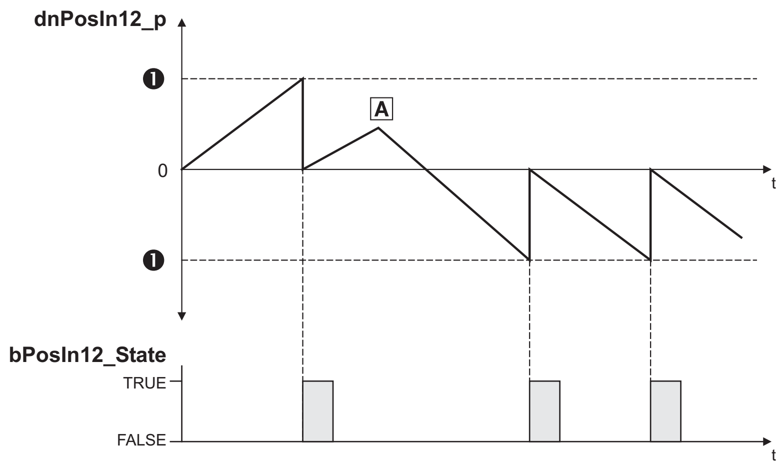
[6-2] Switching performance if the overflow is in the positive direction

- 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"



🄰 Sign reversal of the value at *nFreqIn12_v*
❶ C02845: Comparison value (applies to both positive and negative value range)

[6-3] 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:

$$dnPosIn12\_p\ [incr.] = nFreqIn12\_v\ [rpm] \cdot t\ [s] \cdot 65535\ [incr./rev.]$$

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:

$$1000\ rpm = \frac{1000\ rev.}{60\ s}$$

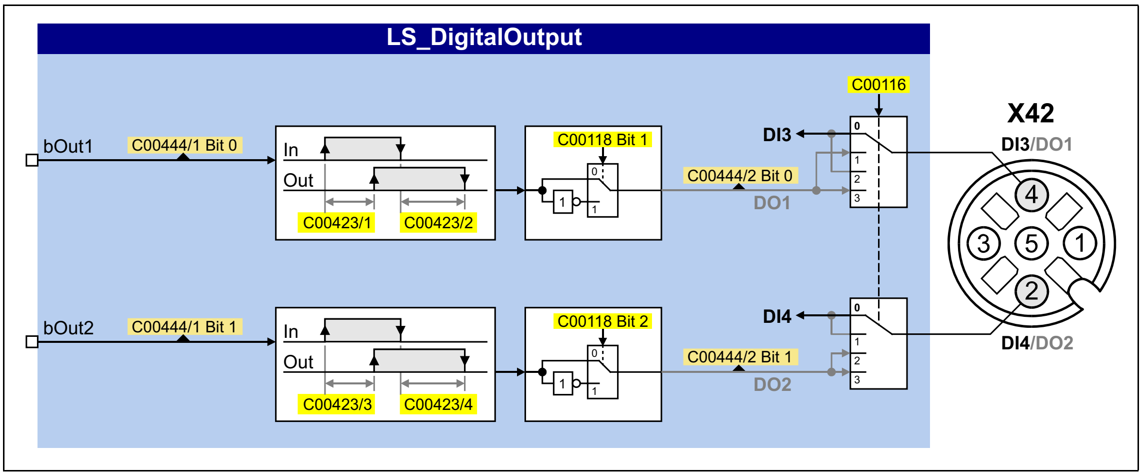
- Calculation of the output value:

$$dnPosIn12\_p = \frac{1000\ rev.}{60\ s} \cdot 10\ s \cdot \frac{65535\ incr.}{Rev.} = 10922666\ incr.$$

6.1.3 Internal interfaces | System block "LS_DigitalOutput"

Both terminals DI3/DO1 and DI4/DO2 of the socket X42 can be optionally used as inputs or outputs.

- In the Lenze setting, both terminals are configured as digital input (DI3 and DI4).
- With a corresponding reconfiguration in C00116, the **LS_DigitalOutput** system block in the FB Editor displays the digital outputs DO1 and DO2.



Input DIS code | data type	Information/possible settings
bOut1 C00444/1 | BOOL	Digital output DO1
bOut2 C00444/2 | BOOL	Digital output DO2

6.2 Analog terminals

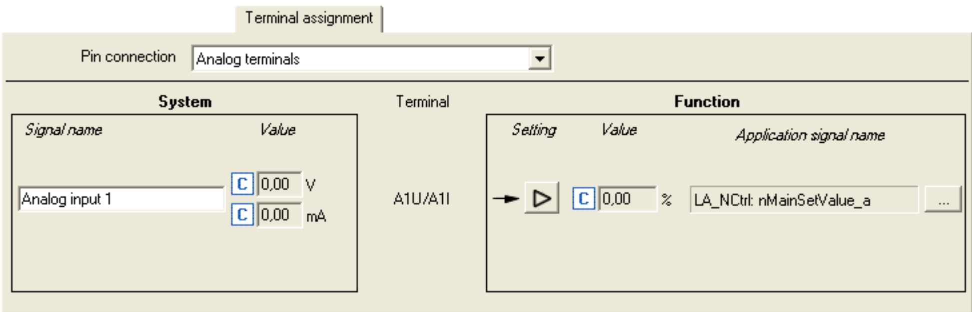
The controller is provided with two analog input terminals (O1U and O1I) at the socket X50 for detecting a current or voltage signal.

- Voltage signal in the 0 ... + 10 V range
  The voltage signal can be e.g. an analog speed setpoint selection or the signal of an external sensor (temperature, pressure, etc.).
- Current signal 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, unassigned inputs of the controller must be assigned as well, e.g. by applying 0 V to the terminal.

Parameterisation dialog in the »Engineer«:



Button	Function
▷	Parameterising analog input (📖 222)
...	Open the parameterising dialog for assigning application inputs to the analog input. ▸ Changing the terminal assignment with the »Engineer« (📖 231)

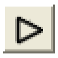
Short overview of the parameters for the analog input:

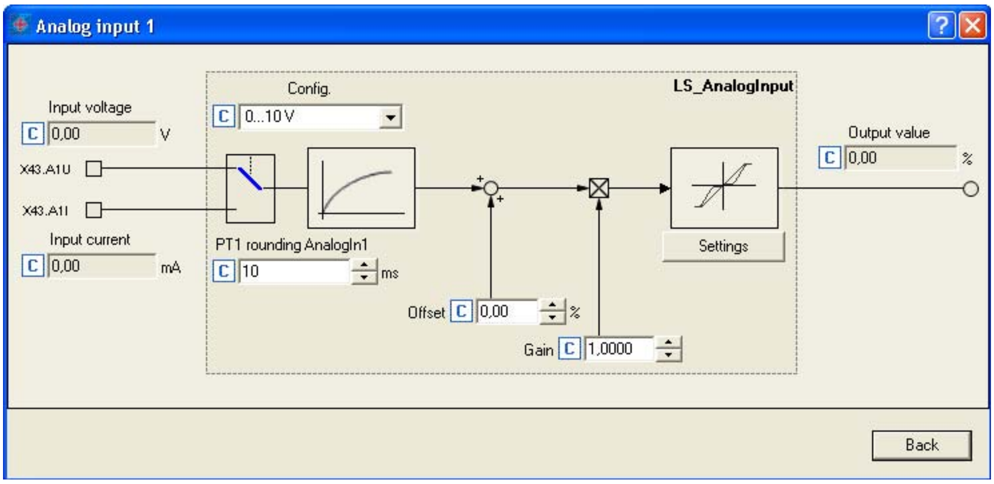
Parameter	Info	Lenze setting Value	Unit
Analog input 1			
C00028/1	AIN1: Input voltage	-	V
C00029/1	AIN1: Input current	-	mA
C00033/1	AIN1: Output value (to application)	-	%
Highlighted in grey = display parameter			

Related topics:

▸ User-defined terminal assignment (📖 227)

6.2.1 Parameterising analog input

By clicking on the  button on the **Terminal assignment** tab, you reach the parameterising dialog for the analog input:



Short overview of the parameters for the analog input:

Parameter	Info	Lenze setting	
		Value	Unit
Analog input 1			
C00034/1	AIN1: Config.	0: 0 V .... + 10 V	
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	
Highlighted in grey = display parameter			

Using current input A1I

In the Lenze setting, voltage signals in the 0 ... +10 V range are evaluated via the analog input A1U. If current signals are to be detected via the input A1I instead, the selection "1: 0...20 mA" or "2: 4...20mA" must 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.

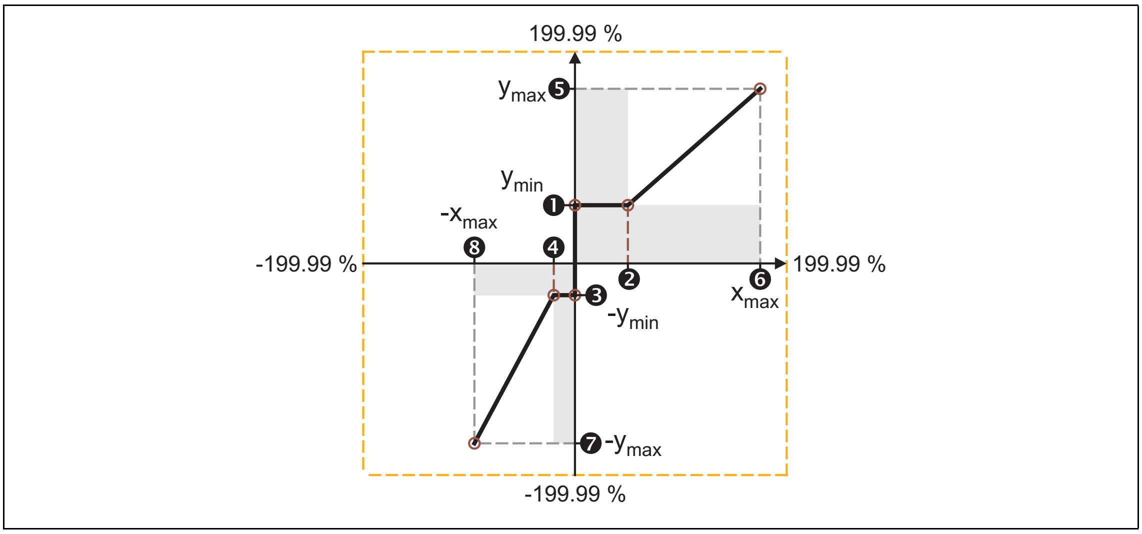
Open-circuit monitoring

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").

6.2.1.1 Signal adaptation by means of characteristic

This function extension is available from version 02.00.00!

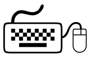
According to the illustration below, an individual characteristic can be parameterised via the subcodes ofC00010 to provide different slopes and a dead band. Here, the input signal corresponds to the X axis and the output signal corresponds to the Y axis:



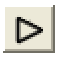
[6-4] Characteristic for analog inputs

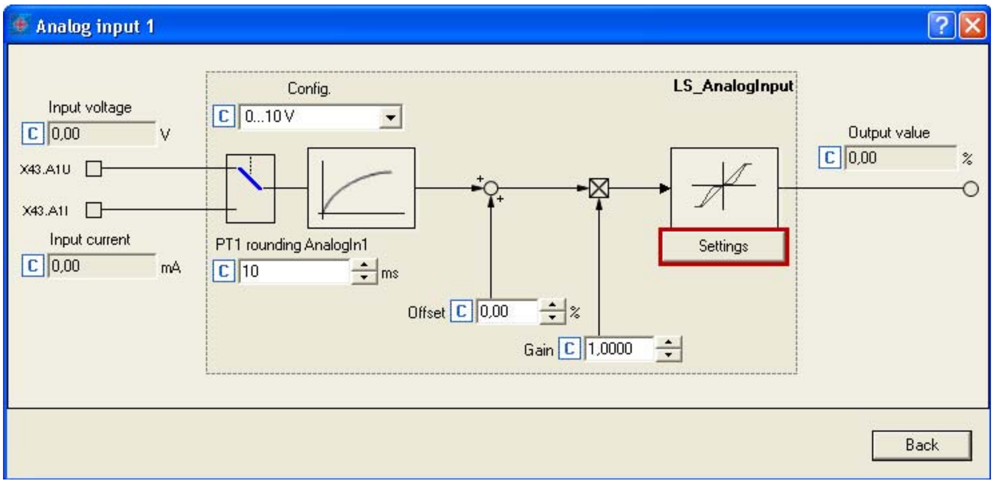
Parameter	Info	Lenze setting	
		Value	Unit
C00010/1	❶ AIN1: (+y0) = min	0.00	%
C00010/2	❷ AIN1: (+x0) = Dead band	1.00	%
C00010/3	❸ AIN1: (-y0) = (-min)	0.00	%
C00010/4	❹ AIN1: (-x0) = (-Dead band)	1.00	%
C00010/5	❺ AIN1: (+ymax)	199.99	%
C00010/6	❻ AIN1: (+xmax)	199.99	%
C00010/7	❼ AIN1: (-ymax)	199.99	%
C00010/8	❽ AIN1: (-xmax)	199.99	%

In the »Engineer«, there is a parameterising dialog for entering the characteristic. This dialog also displays the set characteristic graphically.

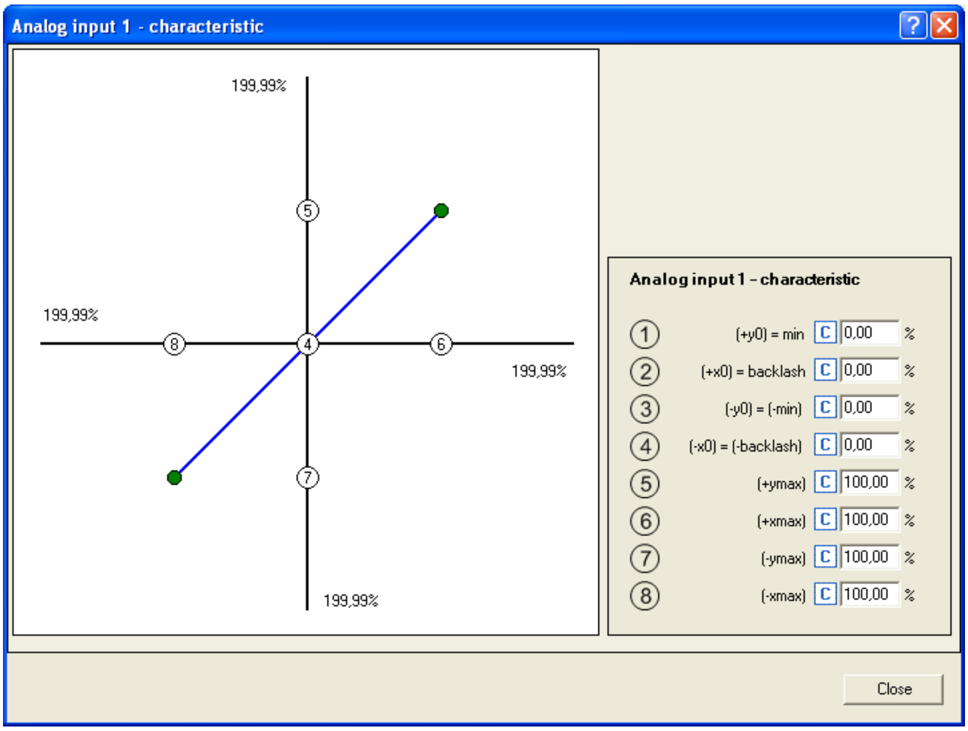


Proceed as follows to open the dialog for parameterising the characteristic:

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.

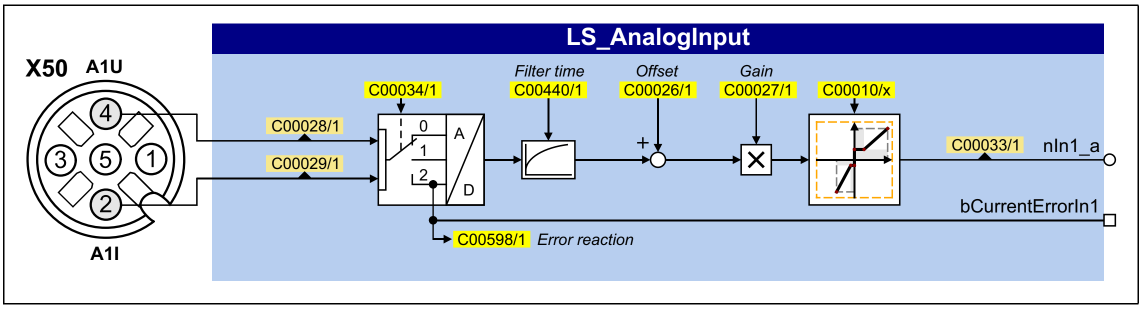


3. Click on the **Settings** button in order to open the *Analog input - Characteristic* dialog box:



6.2.2 Internal interfaces | System block "LS_AnalogInput"

The **LS_AnalogInput** system block maps the analog input in the FB editor.



Output Data type	Value/meaning
nIn1_a C00033/1 | INT	Analog input 1 • Scaling: $\pm 2^{14} \equiv \pm 10$ V for use as voltage input $+2^{14} \equiv +20$ mA for use as current input
bCurrentErrorIn1 BOOL	Status signal "Current input error" • Only when analog input 1 is used as current input. • Application: Cable-breakage monitoring of the 4 ...20 mA circuit. TRUE: $\|I_{AIN1}\| < 4$ mA

6.3 Configuring exception handling of the output terminals

This function extension is available from version 02.00.00!

Exception handling for the 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 C00447, 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 15 ☐	Reserved

- Bit-coded definition is carried out in C00448 specifying the status of the output terminals when being decoupled (bit set = HIGH level):

Bit	Output terminal
Bit 0 ☐	Reserved
Bit 1 ☐	DigOut1_ON
Bit 2 ☐	DigOut2_ON
Bit 3 ☐	Reserved
Bit 4 ☐	Reserved
Bit 5 ☐	BrakeRelease_ON
Bit 6 ☐	Reserved
...	...
Bit 15 ☐	Reserved

Related topics:

▶ Configuring exception handling of the CAN PDOs (📖 398)

6.4 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 inputs/outputs, 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.

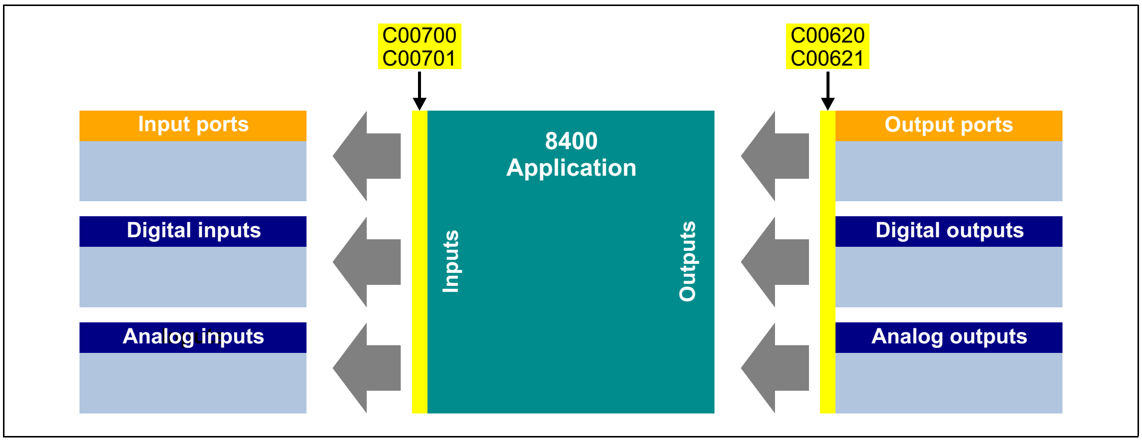
We recommend using the »Engineer« for the implementation of comprehensive user-defined drive solutions.

6.4.1 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:



[6-5] Source-destination principle

Note the following:

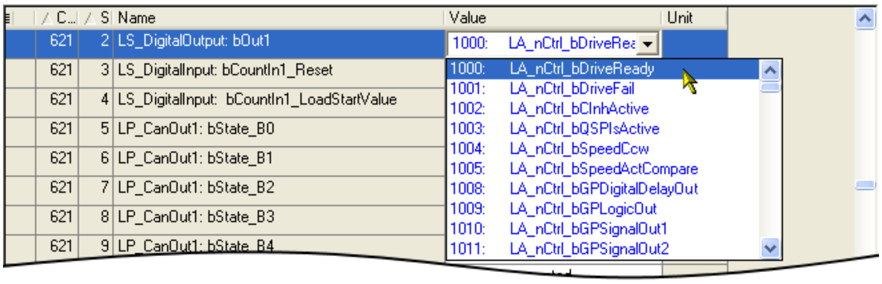
- An equipment input terminal can be logically linked to several inputs of the application block.
- Every input of the application block can only be logically linked to one input signal.
- An output of the application block can be logically linked to several device output terminals.

6.4.2 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 digital output terminals

The subcodes of C00621 serve to change the preconfigured terminal assignment of the digital output terminals:

Parameter	Info	Lenze setting Value | Unit
Digital outputs - terminal assignment		
C00621/2	LS_DigitalOutput:bOut1	1000: LA_nCtrl_bDriveReady
C00621/99	LS_DigitalOutput: bOut2	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 (📖 260)	
C00700/x	Analog connection list
C00701/x	Digital connection list
TA "Switch-off positioning": Configuration parameters (📖 287)	
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 (📖 260)
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.

6.4.3 Changing the terminal assignment with the »Engineer«

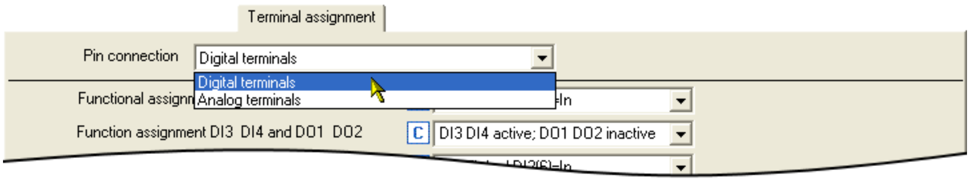
Whereas the configuration parameters referred to have to be parameterised with the keypad, implementation in the »Engineer« is much easier due to the availability of the corresponding dialogs. The following task illustrates the respective procedure.

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.

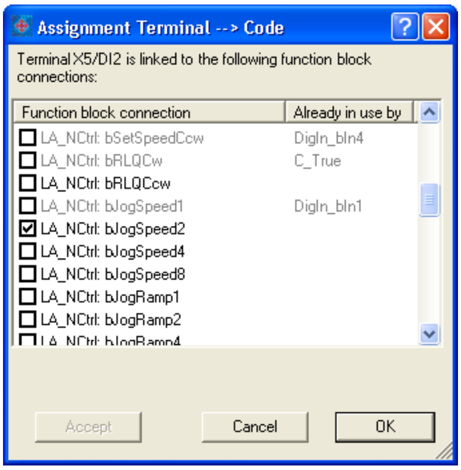
Possibility 1: Change terminal assignment by means of the Terminal Assignment tab

Procedure:

1. Go to the **Terminal Assignment** tab and select "Digital 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:

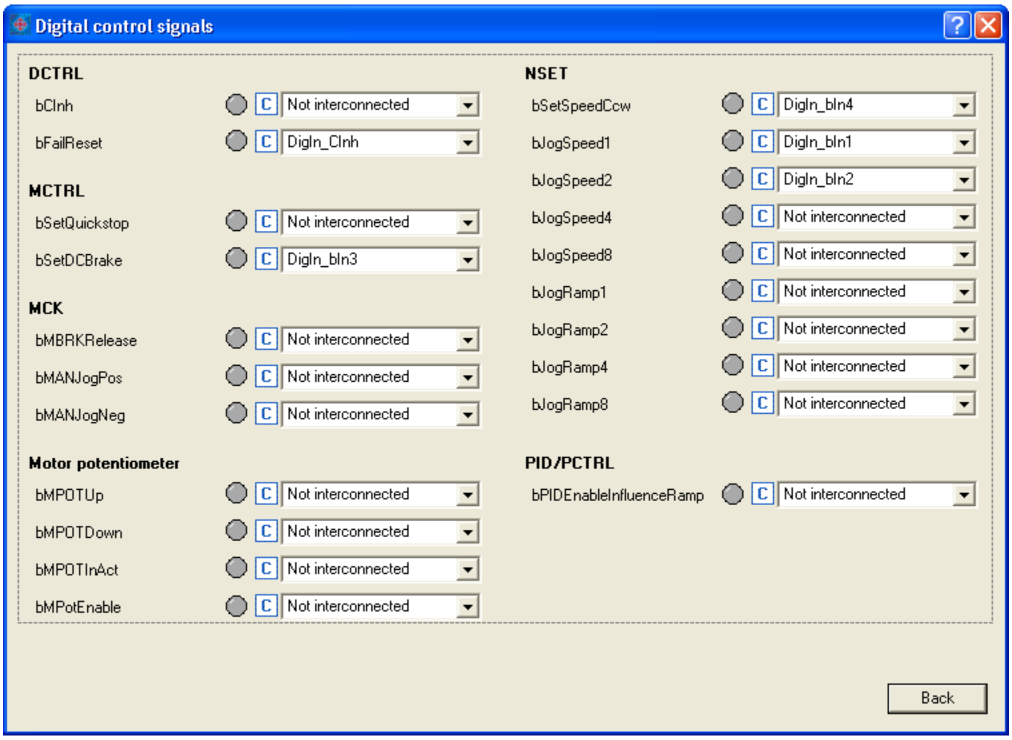


3. Remove checkmark for the connection **LA_NCtrl: bJogSpeed2** in order to cancel the existing logical link.
4. Set checkmark for connection **LA_NCtrl: bJogRamp1** in order to logically link this application input to digital input DI2.

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:

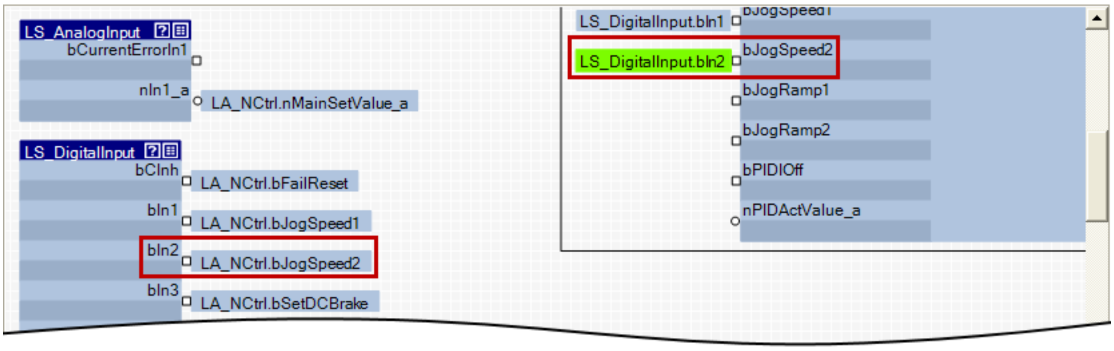


4. In the **bJogSpeed2** list field, set the selection "0: Not interconnected".
5. In the **bJogRamp1** list field, set the selection "16001: DigIn_bIn2".
6. Click on the **Back** button in order to close the dialog box again.

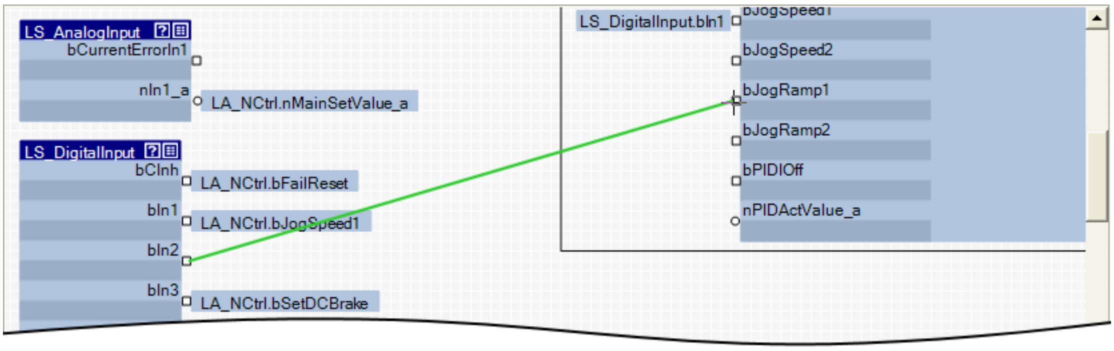
Possibility 3: Change terminal assignment with the FB editor

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". (📖 658)

7 Technology applications

This chapter describes the handling and the functional range of the technology applications available for the 8400 protec StateLine controller.

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" (📖 236)

"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" (📖 263)

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.

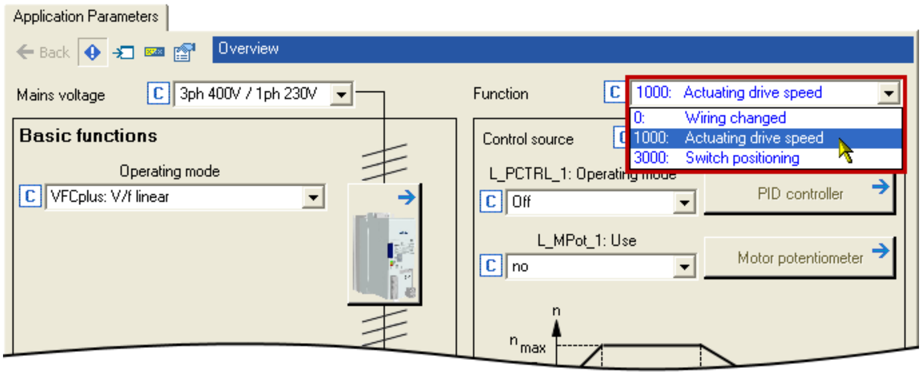
Related topics:

▶ Integrated technology applications (📖 20)
▶ Commissioning of the "Actuating drive speed" technology application (📖 46)
▶ Commissioning of the "Switch-off positioning" technology application (📖 54)

7.1 Selection of the technology application and the control mode

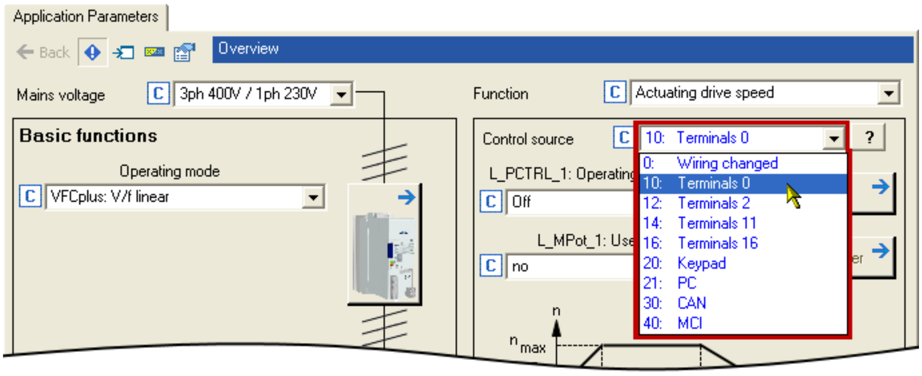
The technology application to be used is selected in C00005.

- 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 (🕮 247)

TA "Switch-off positioning": Terminal assignment of the control modes (🕮 274)

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". (🕮 227)

7.2 TA "Actuating drive speed"

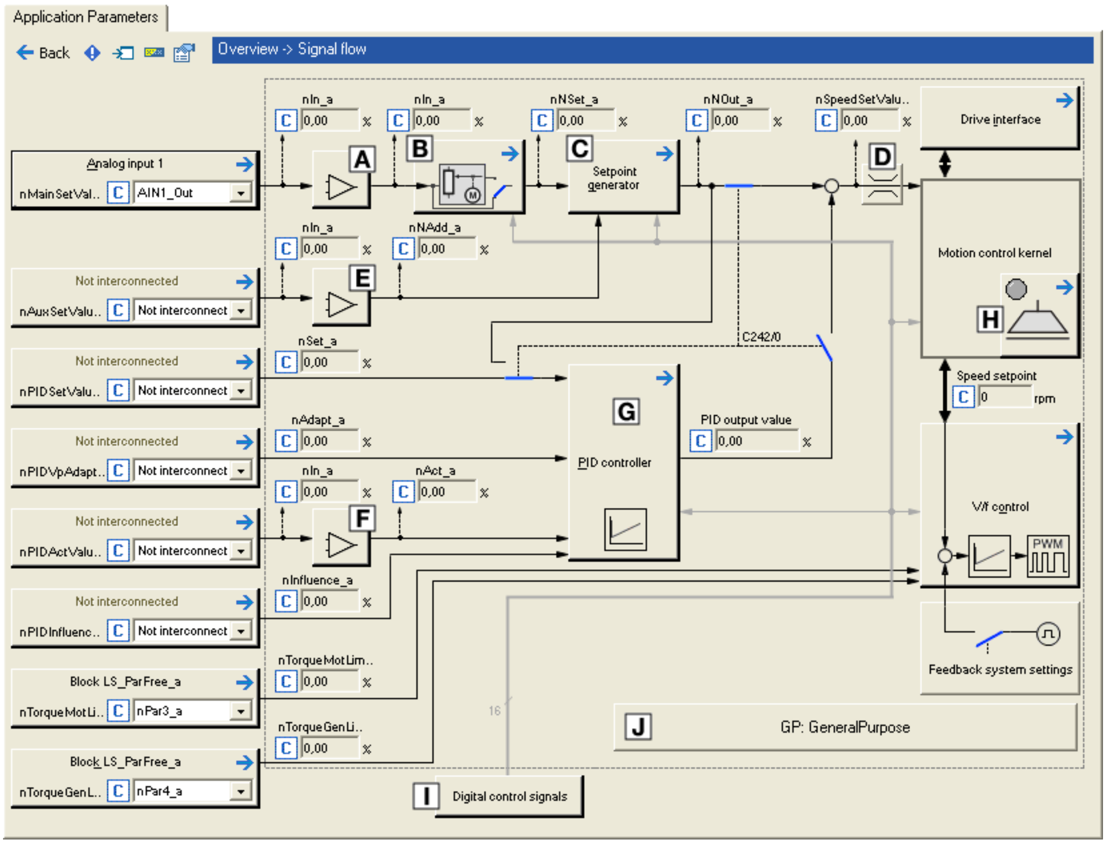
Features

- 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, freely available "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 (📖 46)

7.2.1 Basic signal flow



[7-1] 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 set in C00696 and C00670 for a simple signal 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 the L_NSet FB.

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 the L_MPot FB.

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 the L_NSet FB.

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.

7.2.2 Internal interfaces | application block "LA_NCtrl"

Note!

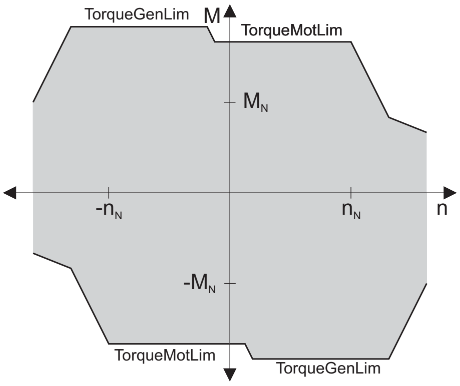
The connectors grayed out in the following table are hidden in the function block editor in the Lenze setting.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

Inputs

Identifier Data type	Information/possible settings	
wCANDriveControl WORD	Control word via system bus (CAN) for device control • 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) for device control • 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. • 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**). • See the subchapter "Interface to safety system" of the chapter on basic drive functions for a detailed description of the individual control bits.	
bCInh BOOL	Enable/Inhibit controller	
	FALSE	Enable controller: The controller switches to the "OperationEnabled" device status if no other source for controller inhibit is active. • 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 message 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. • If the fault still exists, the error status remains unchanged.
bSetQuickstop BOOL	Enable quick stop (QSP) • Also see device command "Activate/deactivate quick stop".	
	TRUE	Activate quick stop • Motor control is decoupled from the setpoint selection and, within the deceleration time parameterised in C00105, the motor is brought to a standstill ($n_{act} = 0$). • The motor is kept at a standstill during closed-loop operation. • A pulse inhibit is set if the auto-DCB function has been activated via C00019.
	FALSE	Deactivate quick stop • The quick stop is deactivated if no other source for the quick stop is active. • C00159 displays a bit code of active sources/causes for the quick stop.

Identifier	Data type	Value	Information/possible settings
bSetDCBrake	BOOL		Manual DC-injection braking (DCB) • Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking".
			Note! Holding braking is not possible when this braking mode is used! 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. • The braking effect stops when the rotor is at standstill. • After the hold time (C00107) has expired, the controller sets the pulse inhibit.
bRFG_Stop	BOOL		Ramp function generator: Maintain the current value of the main setpoint integrator • 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". • 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 • 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 • An additional setpoint for the motor voltage can be specified via this process input. • If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time. • If the value is negative, the voltage is reduced. • Scaling: 16384 ≡ 1000 V
			Stop! 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 • The entire voltage-frequency characteristic is provided with an offset. • Scaling: 16384 ≡ 1000 V
			Stop! 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 • If a torque is connected, e.g. dynamic acceleration processes can be generated. • Scaling: ±32767 ≡ ±180 ° angle of rotation

Identifier (Data type)		Information/possible settings
nTorqueMotLim_a nTorqueGenLim_a INT		Torque limitation in motor mode and in generator mode • These input signals are directly transferred to the motor control to limit the controller's maximum torque in motor and generator mode. • The drive cannot output a higher torque in motor/generator mode than set here. • The applied values (any polarity) are internally interpreted as absolute values. • If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called I_{max} controller. • If sensorless vector control (SLVC) is selected, the limitation has a direct effect on the torque-producing current component. • Scaling: 16384 ≡ 100 % M_{max} (C00057)
		Torque limits in motor and generator mode: 
bSetSpeedCcw BOOL		Change of direction of rotation • 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	Clockwise rotation (Cw)
	TRUE	Direction of rotation to the left (Ccw)
bRLQCw BOOL		Activate clockwise rotation (fail-safe) • For a detailed functional description see the L_RLQ FB.
	FALSE	Quick stop
	TRUE	CW rotation
bRLQCcw BOOL		Activate counter-clockwise rotation (fail-safe) • For a detailed functional description see the L_RLQ FB.
	FALSE	Quick stop
	TRUE	Counter-clockwise rotation
nMainSetValue_a INT		Main speed setpoint • Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal 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 the L_NSet FB.

<table>
<tr><th>Identifier
Data type</th><th colspan="2">Information/possible settings</th></tr>
<tr><td>nAuxSetValue_a
INT</td><td colspan="2">Additional speed setpoint
• 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 the L_NSet FB.</td></tr>
<tr><td>bJogSpeed1
bJogSpeed2
BOOL</td><td colspan="2" rowspan="2">Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint
• A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via these selection inputs.
• The four selection inputs are binary coded, therefore 15 fixed setpoints can be selected.
• In the case of binary coded selection "0" (all inputs = FALSE or not assigned), main setpoint nMainSetValue_a is active.
• The selection of the fixed setpoints is carried out in C00039/1...15 in [%] based on the reference speed (C00011).
• For a detailed functional description see the L_NSet FB.</td></tr>
<tr><td>bJogSpeed4
bJogSpeed8
BOOL</td></tr>
<tr><td>bJogRamp1
bJogRamp2
BOOL</td><td colspan="2" rowspan="2">Selection inputs for alternative acceleration/deceleration times for the main setpoint
• The four selection inputs are binary coded, therefore 15 alternative acceleration/deceleration times can be selected.
• 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).
• Alternative acceleration times are selected in C00101/1...15.
• The selection of the alternative deceleration times is carried out in C00103/1...15.
• For a detailed functional description see the L_NSet FB.</td></tr>
<tr><td>bJogRamp4
bJogRamp8
BOOL</td></tr>
<tr><td colspan="3">Motor potentiometer
Alternatively to the input signal nMainSetValue_a, the main setpoint can also be generated by 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 the L_MPot FB.</td></tr>
<tr><td rowspan="2">bMPotEnable
BOOL</td><td colspan="2">Activating the motor potentiometer function
• This input and C00806 are OR'd.</td></tr>
<tr><td>TRUE</td><td>The motor potentiometer function is active; the speed setpoint can be changed via the bMPotUp and bMPotDown control inputs.</td></tr>
<tr><td rowspan="2">bMPotUp
BOOL</td><td colspan="2">Increasing the speed setpoint</td></tr>
<tr><td>TRUE</td><td>Approach the upper speed limit value set in C00800 with the acceleration time set in C00802.</td></tr>
<tr><td rowspan="2">bMPotInAct
BOOL</td><td colspan="2">Activating the inactive function</td></tr>
<tr><td>TRUE</td><td>The speed setpoint behaves according to the inactive function set in C00804.
• In the Lenze setting, the speed setpoint is maintained.</td></tr>
<tr><td rowspan="2">bMPotDown
BOOL</td><td colspan="2">Decreasing the speed setpoint</td></tr>
<tr><td>TRUE</td><td>Approach the lower speed limit value set In C00801 with the deceleration time set in C00803.</td></tr>
</table>

Identifier	Data type	Information/possible settings
Process controller • In the Lenze setting, the process controller is deactivated. • The activation is executed by selecting the operating mode in C00242. • For a detailed functional description see FB L_PCTRL.		
bPIDEnableInfluenceRamp	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	BOOL	Switch off the I-component of the process controller • In conjunction with the operating mode set in C00242 (Lenze setting: "Off"). TRUE: I-component of the process controller is switched off.
nPIDVpAdapt_a	INT	Adaptation of gain Vp set in C00222 in percent • Scaling: 16384 ≡ 100 % • Internal limitation to ± 199.99 % • Changes can be done online.
nPIDSetValue_a	INT	Sensor and process setpoint for operating modes 2, 4 and 5 • Scaling: 16384 ≡ 100 % • Internal limitation to ± 199.99 %
nPIDActValue_a	INT	Speed or actual sensor value (actual process value) • Offset and gain for this input signal can be set in C00698 and C00672. • Scaling: 16384 ≡ 100 % • Internal limitation to ± 199.99 %
nPIDInfluence_a	INT	Limitation of the influencing factor in percent • The influence factor of the PID controller can be limited to a certain value (-199.99% ... + 199.99%) via *nPIDInfluence_a*. • Scaling: 16384 ≡ 100 % • Internal limitation to ± 199.99 %
MCK basic functions		
bMBRKRelease	BOOL	Holding brake control: Release/apply brake • In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). FALSE: Apply brake. • During automatic operation, the internal brake logic controls the brake. TRUE: Release brake. • During automatic operation, the internal brake logic is deactivated and the brake is released. If the brake control has inhibited the controller, this inhibit is deactivated again.
bManJogPos bManJogNeg	BOOL	reserved (inputs are not interconnected on the application level)
GP: GeneralPurpose The following inputs are interconnected with logic/arithmetic functions on application level for free usage. ▸ "GeneralPurpose" functions		
nGPAnalogSwitchIn1_a nGPAnalogSwitchIn2_a	INT	Analog switch: Input signals • The input signal selected via the selection input *bGPAnalogSwitchSet* is output at output *nGPAnalogSwitchOut_a*.
bGPAnalogSwitchSet	BOOL	Analog switch: Selection input FALSE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn1_a* TRUE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn2_a*
nGPArithmetikIn1_a nGPArithmetikIn2_a	INT	Arithmetic: Input signals • The arithmetic function is selected in C00338. • The result is output at output *nGPArithmetikOut_a*.

Identifier Data type	Information/possible settings
nGPMulDivIn_a INT	Multiplication/Division: Input signal • The factor for the multiplication can be set in C00699/1 (numerator) and C00699/2 (denominator). • The result is output at output *nGPMulDivOut_a*.
bGPDigitalDelayIn BOOL	Binary delay element: Input signal • The on-delay can be set in C00720/1. • The off-delay can be set in C00720/2. • The time-delayed input signal is output at output *bGPDigitalDelayOut*.
bGPLogicIn1 bGPLogicIn2 bGPLogicIn3 BOOL	Binary logic: Input signals • The logic operation is selected in C00820. • The result is output at output *bGPLogicOut*.
nGPCompareIn1_a nGPCompareIn2_a INT	Analog comparison: Input signals • The comparison operation is selected in C00680. • Hysteresis and window size can be set in C00680 and C00682. • If the comparison statement is true, the output *bGPCompareOut* will be set to TRUE.
bGPDFlipFlop_InD bGPDFlipFlop_InClk bGPDFlipFlop_InClr BOOL	D-FlipFlop: Input signals • Data, clock and reset input
Free inputs The following inputs can freely be interconnected on the application level. The signals can be transferred from the I/O level to the application level via these inputs.	
bFreeIn1 ... bFreeIn8 BOOL	Free inputs for digital signals
wFreeIn1 ... wFreeIn4 WORD	Free inputs for 16-bit signals

Outputs

Identifier Data type	Value/meaning	
wDriveControlStatus WORD	Status word of the controller (based on DSP-402) • The status word contains information on the currents status of the drive controller. • See the "wDeviceStatusWord status word" subchapter of the chapter on device control for a detailed description of the bit assignment.	
wStateDetermFailNoLow WORD	Display of the status determining error (LOW word)	
wStateDetermFailNoHigh WORD	Display of the status determining error (HIGH word)	
bDriveFail BOOL	TRUE	Drive controller in error status. • "Fault" device status is active.
bDriveReady BOOL	TRUE	Controller is ready for operation. • "SwitchedOn" device status is active. • 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	Clockwise rotation (Cw)
	TRUE	Direction of rotation to the left (Ccw)

Identifier Data type	Value/meaning	
bSpeedActCompare BOOL	Result of the speed comparison	
	TRUE	During open-loop operation: Speed setpoint < Comparison value (C00024)
		During closed-loop operation: Actual speed value < Comparison value (C00024)
bOverLoadActive BOOL	In preparation (output is not interconnected on the application level)	
bUnderLoadActive BOOL	In preparation (output is not interconnected on the application level)	
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 • Scaling: 16384 ≡ 100 % I_{max_mot} (C00022)	
nMotorSpeedSet_a INT	Speed setpoint • Scaling: 16384 ≡ 100 % reference speed (C00011)	
nMotorSpeedAct_a INT	Actual speed value • Scaling: 16384 ≡ 100 % reference speed (C00011)	
nMotorTorqueAct_a INT	Actual torque • 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. • Scaling: 16384 ≡ 100 % M_{max} (C00057)	
nDCVoltage_a INT	Actual DC-bus voltage • Scaling: 16384 ≡ 1000 V	
nMotorVoltage_a INT	Current motor voltage/inverter output voltage • Scaling: 16384 ≡ 1000 V	
MCK basic functions		
bBrakeReleaseOut BOOL	Holding brake control: Trigger signal for the holding brake control switching element via a digital output • Use bit 0 in C02582 to activate inverted switching element triggering.	
	FALSE	Apply brake.
	TRUE	Release brake.
bBrakeReleased BOOL	Holding brake control: Status signal of the brake control with regard to the release and application times of the brake	
	FALSE	Brake applied (after the brake application time has expired).
	TRUE	Brake released (after the brake release time has expired).

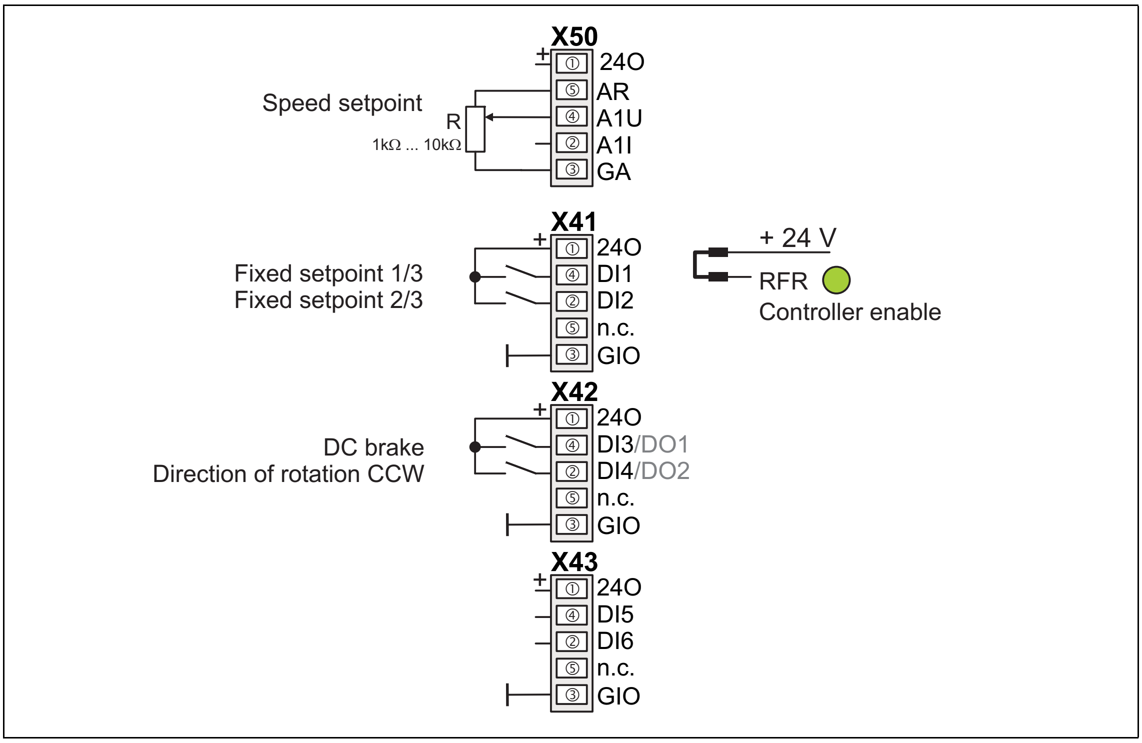
Identifier Data type	Value/meaning
GP: GeneralPurpose The following outputs are interconnected with logic/arithmetic functions on application level for free usage. "GeneralPurpose" functions	
nGPAnalogSwitchInOut_a INT	Analog switch: Output signal
nGPArithmetikOut_a INT	Arithmetic: Output signal
nGPMulDivOut_a INT	Multiplication/Division: Output signal
bGPDigitalDelayOut BOOL	Binary delay element: Output signal
bGPLogicOut BOOL	Binary logic: Output signal
bGPCompareOut BOOL	Analog comparison: Output signal
bGPSignalOut1 ... bGPSignalOut4 BOOL	Binary signal monitor: Output signals • The signal sources to be output are selected in C00411/1...4. • A bit coded inversion of the output signals can be parameterised in C00412.
nGPSignalOut1_a ... nGPSignalOut4_a BOOL	Analog signal monitor: Output signals • The signal sources to be output are selected in C00410/1...4. • Gain and offset for each output signal can be parameterised in C00413/1...8.
bGPDFlipFlop_Out BOOL	D-FlipFlop: Output signal
bGPDFlipFlop_NegOut BOOL	D-FlipFlop: Negated output signal
Free outputs The following outputs can freely be interconnected on the application level. The signals from the application level can be transferred to the I/O level via these outputs.	
bFreeOut1 ... bFreeOut8 BOOL	Free outputs for digital signals
wFreeOut1 ... wFreeOut4 WORD	Free outputs for 16-bit signals

7.2.3 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.

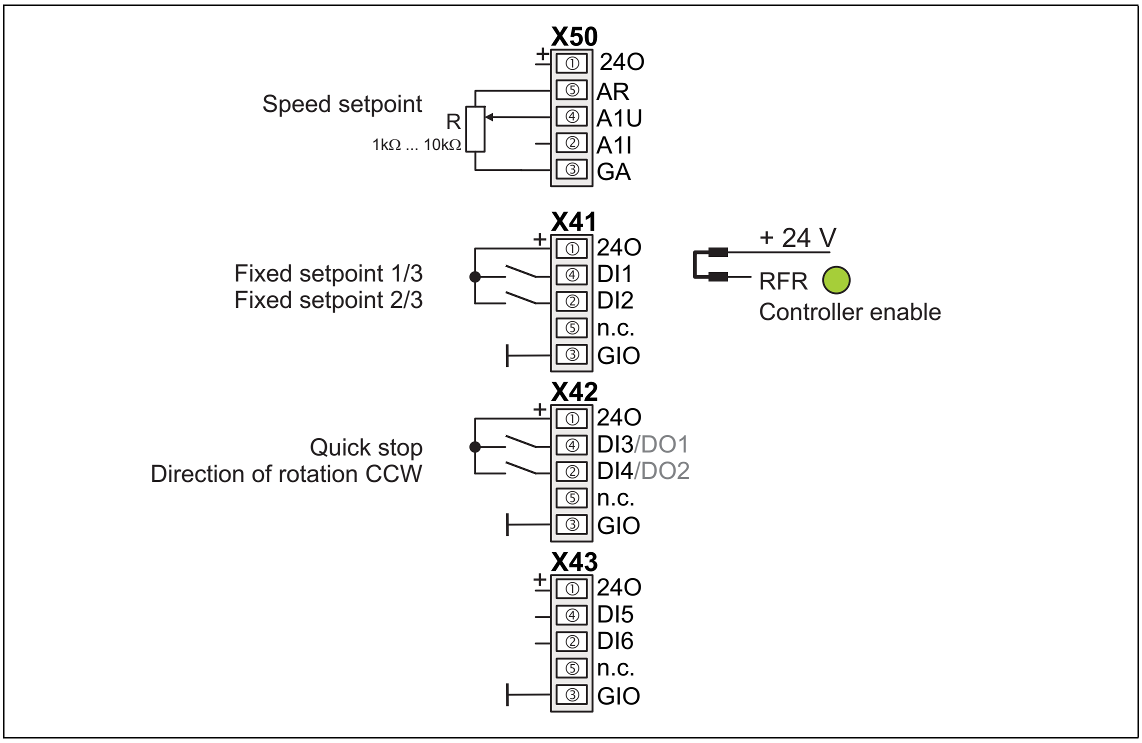
	Control mode (C00007)							
	10: Terminals 0	12: Terminals 2	14: Terminals 11	16: Terminal 16	20: Keypad	21: PC	30: CAN	40: MCI
Digital input terminals								
RFR	**Controller enable / Reset of error message** bFailReset (In case of 8400 protec, the RFR terminal is internally set to HIGH level via wire jumper.)							
X41/DI1	**Fixed setpoint 1/3** bJogSpeed1		**Change of direction of rotation** bSetSpeedCcw	**Fixed setpoint 1/3** bJogSpeed1	-	-	**Quick stop** bSetQuickstop	
X41/DI2	**Fixed setpoint 2/3** bJogSpeed2		**Activate manual DC-injection braking (DCB)** bSetDCBrake	**Fixed setpoint 2/3** bJogSpeed2	-	-	-	-
X42/DI3	**Activate manual DC-injection braking (DCB)** bSetDCBrake	**Quick stop** bSetQuickstop	**Motor potentiometer: Increase speed** bMPotUp	**CW rotation quick stop** bRLQCw	-	-	-	-
X42/DI4	**Change of direction of rotation** bSetSpeedCcw		**Motor potentiometer: Decrease speed** bMPotDown	**CCW rotation quick stop** bRLQCcw	-	-	-	-
X43/DI5	-	-	-	-	-	-	-	-
X43/DI5	-	-	-	-	-	-	-	-
Analog input terminals								
X50/A1U	**Main speed setpoint** nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)				-	-	**Additional speed setpoint** nAuxSetValue_a 10 V ≡ 100 % reference speed (C00011)	
X50/A1I	-	-	-	-	-	-	-	-
Digital output terminals								
DO1/DO2 is only available after respective change of the function assignment in C00116.								
X42/DO1	**Status "Drive is ready"** bDriveReady							
X42/DO2	-							

7.2.3.1 Terminals 0



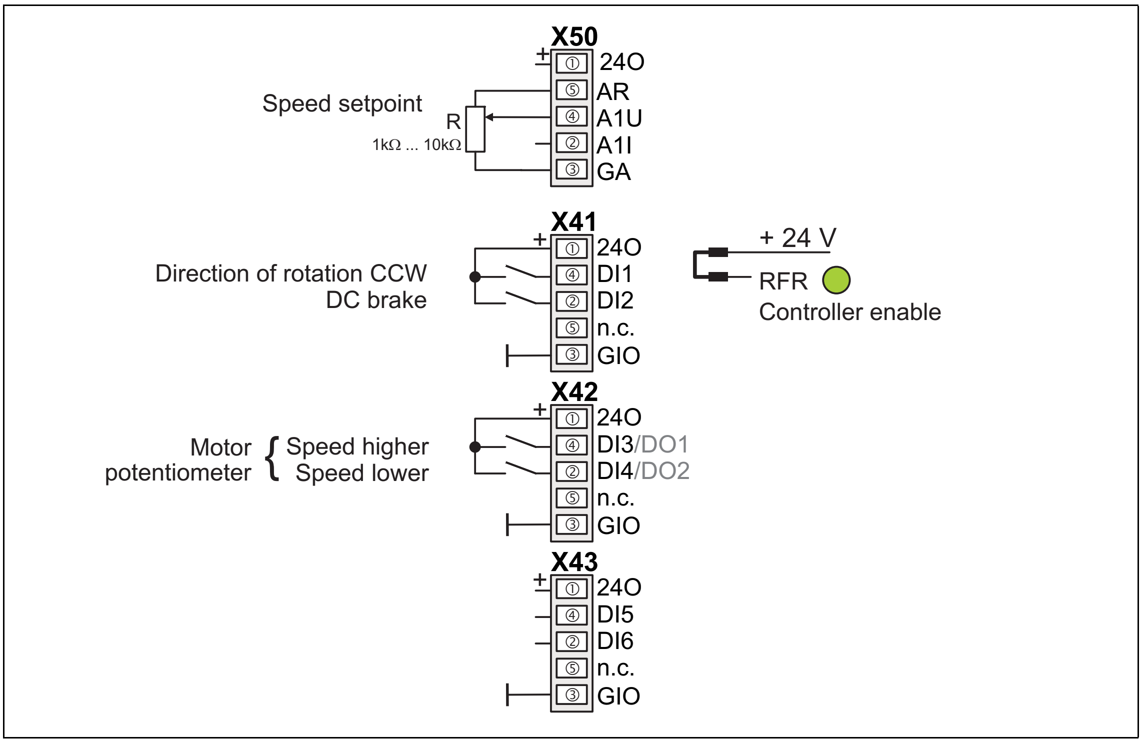
Connection	Assignment		Connection	Assignment
X41/DI1	LA_NCtrl.bJogSpeed1		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_NCtrl.bJogSpeed2			
X42/DI3	LA_NCtrl.bSetDCBrake		X42/DO1	-
X42/DI4	LA_NCtrl.bSetSpeedCcw		X42/DO2	LA_NCtrl.bDriveReady
X43/DI5	-		**X50/A1U**	LA_NCtrl.nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	-			

7.2.3.2 Terminals 2



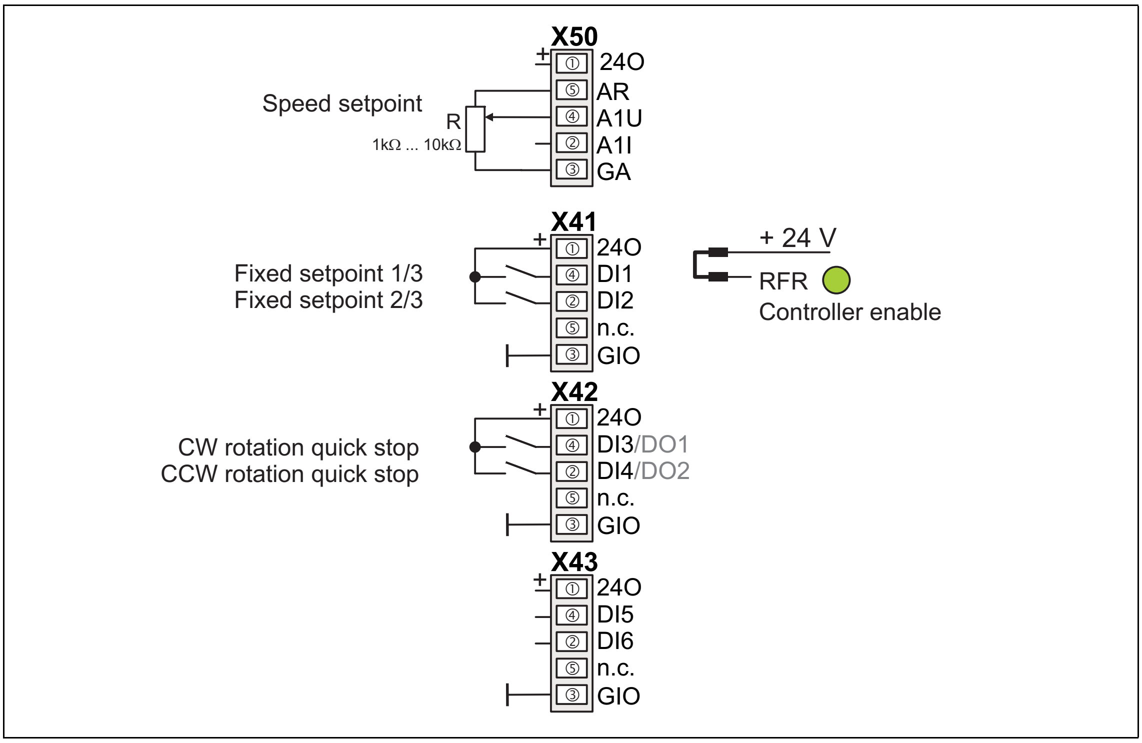
Connection	Assignment		Connection	Assignment
X41/DI1	LA_NCtrl.bJogSpeed1		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_NCtrl.bJogSpeed2			
X42/DI3	LA_NCtrl.bSetQuickstop		X42/DO1	-
X42/DI4	LA_NCtrl.bSetSpeedCcw		X42/DO2	LA_NCtrl.bDriveReady
X43/DI5	-		**X50/A1U**	LA_NCtrl.nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	-			

7.2.3.3 Terminals 11



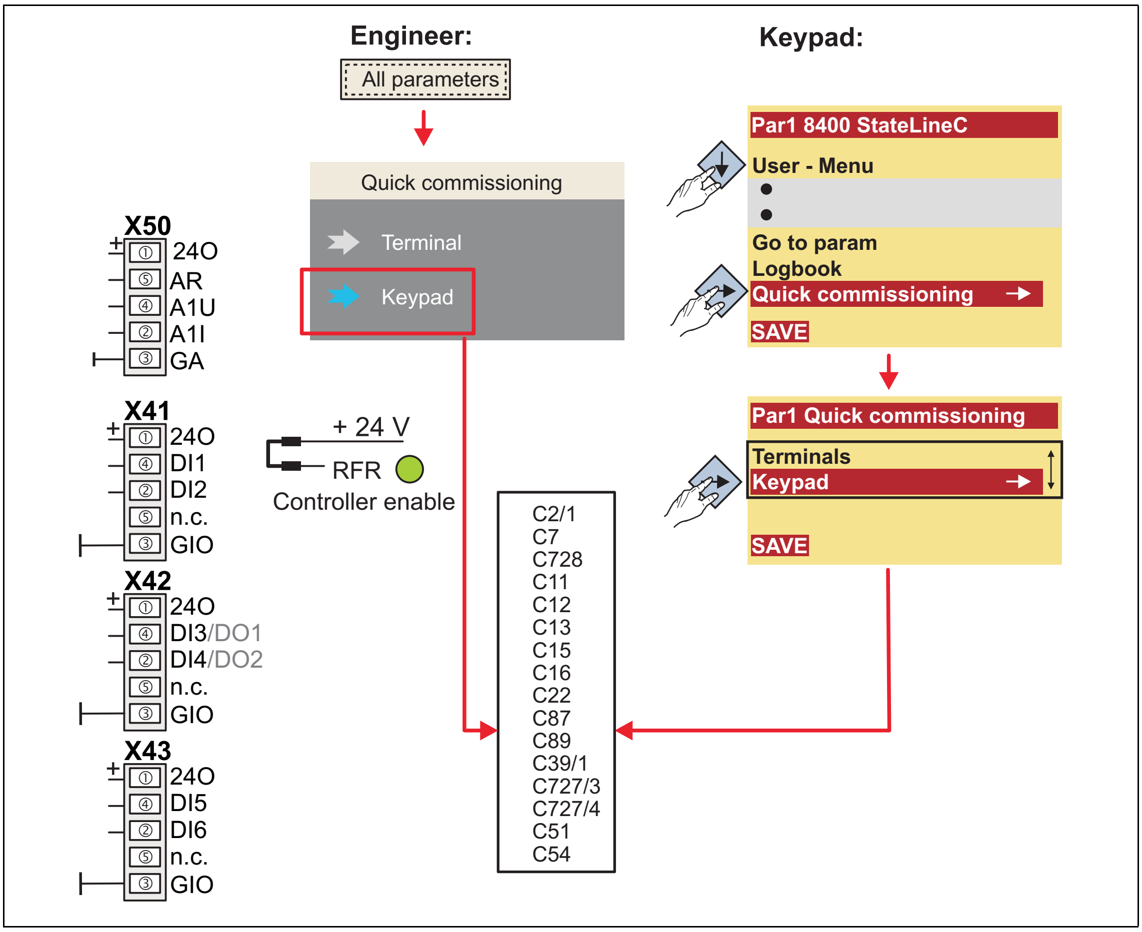
Connection	Assignment		Connection	Assignment
X41/DI1	LA_NCtrl.bSetSpeedCcw		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_NCtrl.bSetDCBrake			
X42/DI3	LA_NCtrl.bMPotUp		X42/DO1	-
X42/DI4	LA_NCtrl.bMPotDown		X42/DO2	LA_NCtrl.bDriveReady
X43/DI5	-		**X50/A1U**	LA_NCtrl.nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	-			

7.2.3.4 Terminal 16



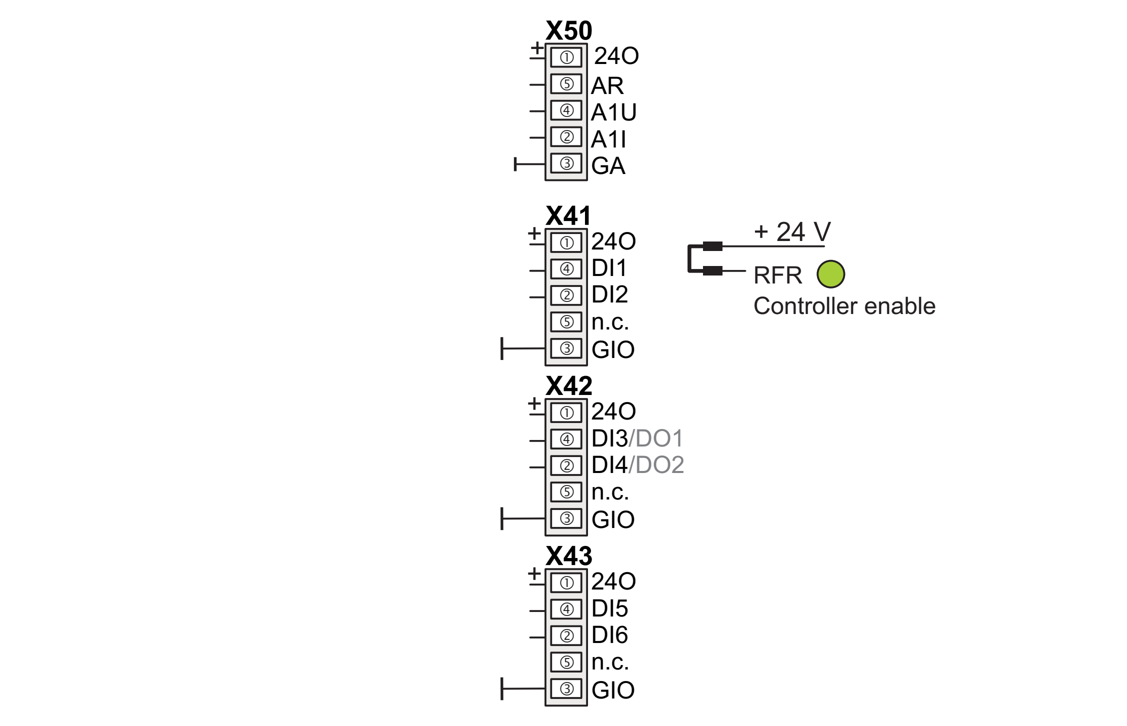
Connection	Assignment		Connection	Assignment
X41/DI1	LA_NCtrl.bJogSpeed1		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_NCtrl.bJogSpeed2			
X42/DI3	LA_NCtrl. bRLQCw		X42/DO1	-
X42/DI4	LA_NCtrl. bRLQCcw		X42/DO2	LA_NCtrl.bDriveReady
X43/DI5	-		**X50/A1U**	LA_NCtrl.nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	-			

7.2.3.5 Keypad



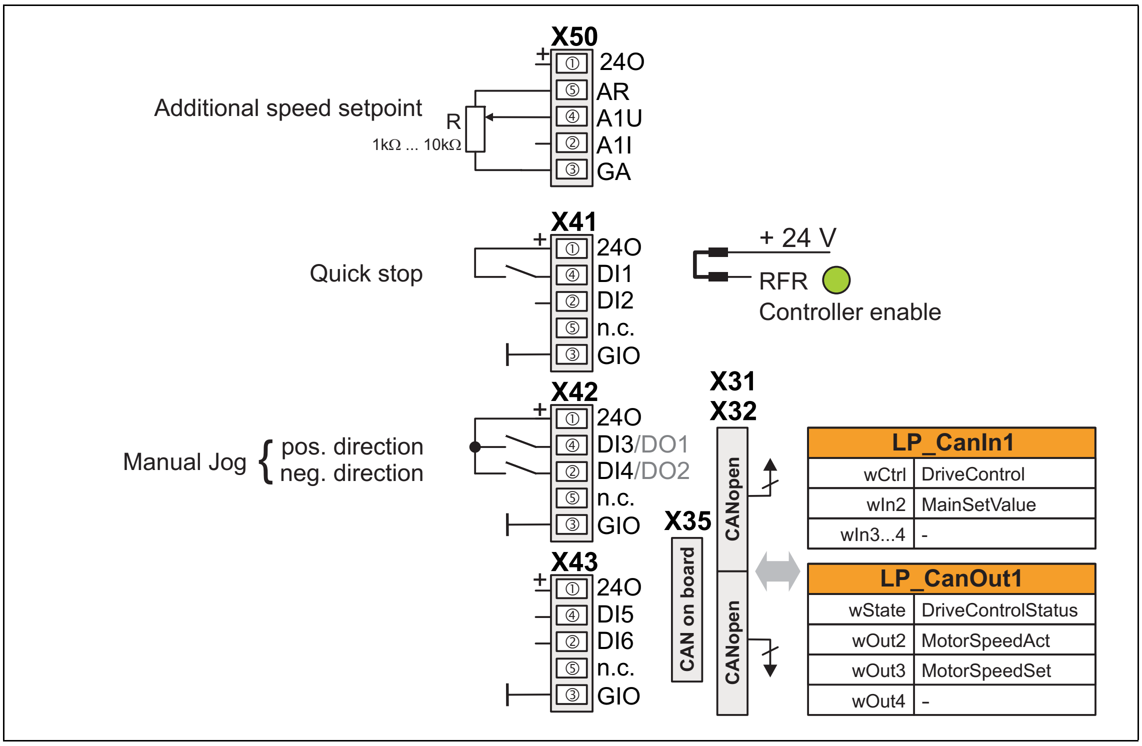
Connection	Assignment		Connection	Assignment
X41/DI1	-		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	-			
X42/DI3	-		X42/DO1	-
X42/DI4	-		X42/DO2	LA_NCtrl.bDriveReady
X43/DI5	-		**X50/A1U**	-
X43/DI6	-			

7.2.3.6 PC



Connection	Assignment		Connection	Assignment
X41/DI1	-		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	-			
X42/DI3	-		X42/DO1	-
X42/DI4	-		X42/DO2	LA_NCtrl.bDriveReady
X43/DI5	-		**X50/A1U**	-
X43/DI6	-			

7.2.3.7 CAN



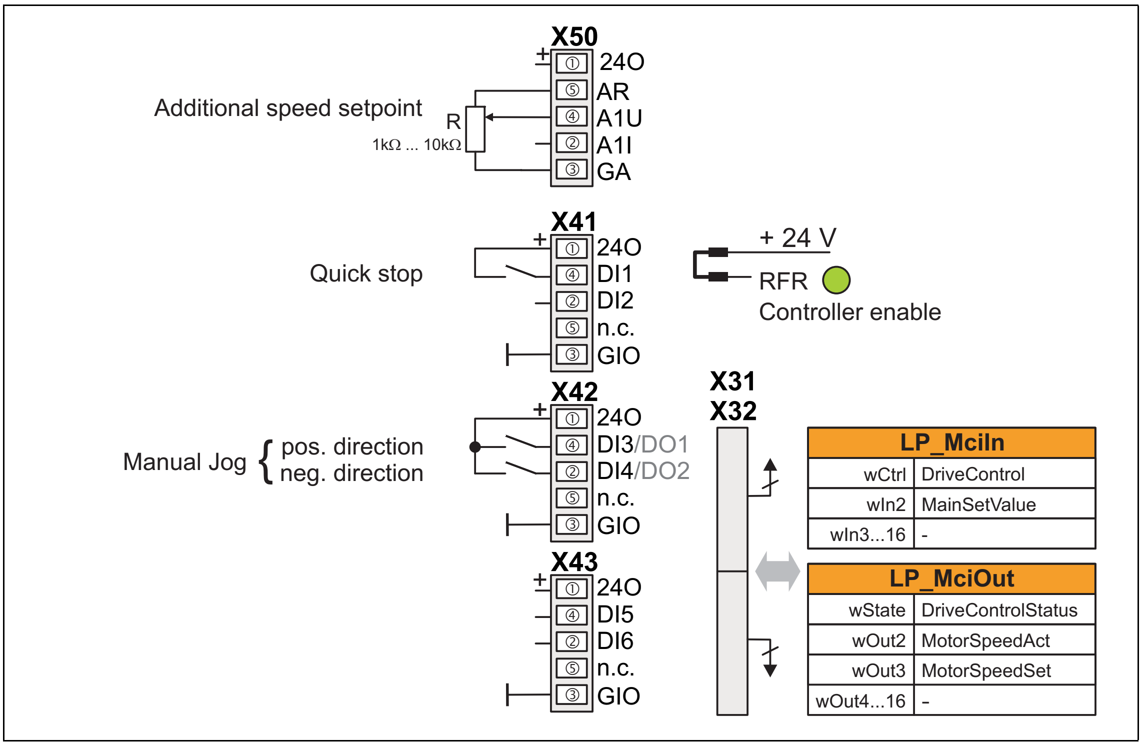
Connection	Assignment		Connection	Assignment
X41/DI1	LA_NCtrl.bSetQuickStop		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	-			
X42/DI3	LA_NCtrl.bMANJogPos (currently has no function)		X42/DO1	-
X42/DI4	LA_NCtrl.bMANJogNeg (currently has no function)		X42/DO2	LA_NCtrl.bDriveReady
X43/DI5	-		**X50/A1U**	LA_NCtrl.nAuxSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	-			

▶ Process data assignment for fieldbus communication (📖 256)

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!

7.2.3.8 MCI



Connection	Assignment		Connection	Assignment
X41/DI1	LA_NCtrl.bSetQuickStop		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	-		X42/DO1	-
X42/DI3	LA_NCtrl.bMANJogPos (currently has no function)		X42/DO2	LA_NCtrl.bDriveReady
X42/DI4	LA_NCtrl.bMANJogNeg (currently has no function)		**X50/A1U**	LA_NCtrl.nAuxSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI5	-			
X43/DI6	-			

▸ Process data assignment for fieldbus communication (📖 256)

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!

7.2.4 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 depends only on the application, not on the bus system used:

Input words	Name	Assignment
Word 1	DriveControl	Control word • For bit assignment see the table below.
Word 2	MainSetValue	Speed setpoint • Scaling: 16384 ≡ 100 % reference speed (C00011)
Word 3	-	Not preconfigured
Word 4	-	Not preconfigured
Words 5 ... 16	-	Not preconfigured • Only available in control mode "40: MCI".

Control word	Name	Function
Bit 0	SwitchOn	1 ≡ Change to the "SwitchedOn" device status • 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. • 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 (pulse inhibit)
Bit 2	SetQuickStop	1 ≡ Activate quick stop (QSP). ▶ Activate/deactivate quick stop (🕮 76)
Bit 3	EnableOperation	1 ≡ Enable controller (RFR) • 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. ▶ Enable/Inhibit controller (🕮 75)
Bit 4	ModeSpecific_1	Reserved (currently not assigned)
Bit 5	ModeSpecific_2	
Bit 6	ModeSpecific_3	
Bit 7	ResetFault	1 ≡ Reset fault (trip reset) • Acknowledge error message (if the error cause has been eliminated). ▶ Reset error (🕮 77)
Bit 8	SetHalt	1 ≡ Activate stop function • 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 ▶ Manual DC-injection braking (DCB) (🕮 156)
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) 1 ≡ Direction of rotation to the left (Ccw)

Output words	Name	Assignment
Word 1	DriveControlStatus	Status word • For bit assignment see the table below.
Word 2	MotorSpeedAct	Actual speed value • Scaling: 16384 ≡ 100 % reference speed (C00011)
Word 3	MotorSpeedSet	Resulting overall setpoint • Scaling: 16384 ≡ 100 % reference speed (C00011)
Word 4	-	Not preconfigured
Words 5 ... 16	-	Not preconfigured • Only available in 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: 1 ≡ Speed setpoint < Comparison value (C00024)
		During closed-loop operation: 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 ▸ Device state machine and device states (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 • 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)

7.2.5 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 smoothing 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: Acceleration time	0.010	s
C00228	L_PCTRL_1: Deceleration 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: Deceleration 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	

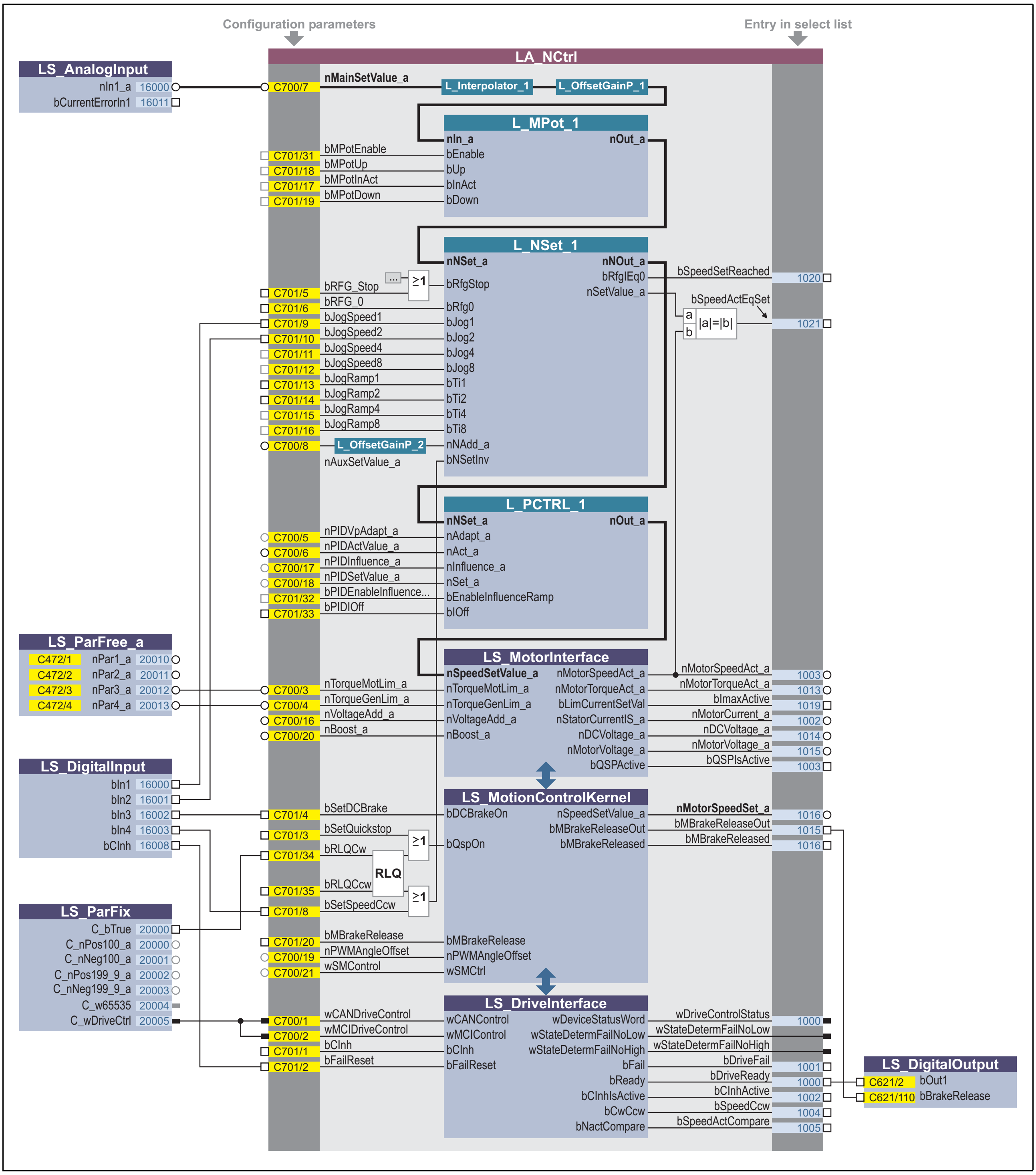
Parameter	Info	Lenze setting	
		Value	Unit
C00696	L_OffsetGainP_1: Offset	0.00	%
C00697	L_OffsetGainP_2: Offset	0.00	%
C00698	L_OffsetGainP_3: Offset	0.00	%
C00800	L_MPot_1: Upper limit	100.00	%
C00801	L_MPot_1: Lower limit	-100.00	%
C00802	L_MPot_1: Acceleration time	10.0	s
C00803	L_MPot_1: Deceleration 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 (📖 290)

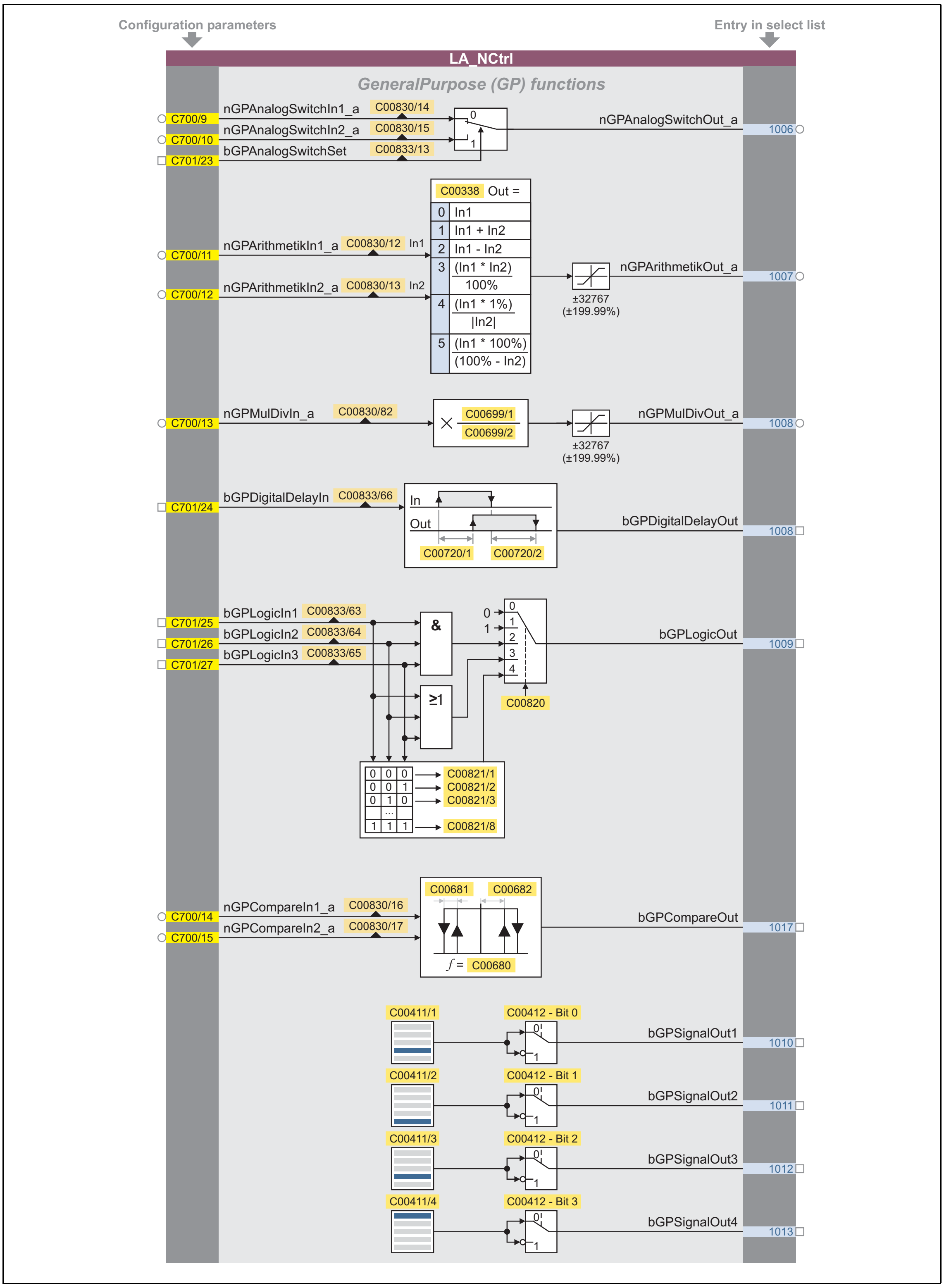
7.2.6 Configuration parameters

If required, the subcodes of C00700 and C00701 serve to change the pre-configured assignment of the application inputs:

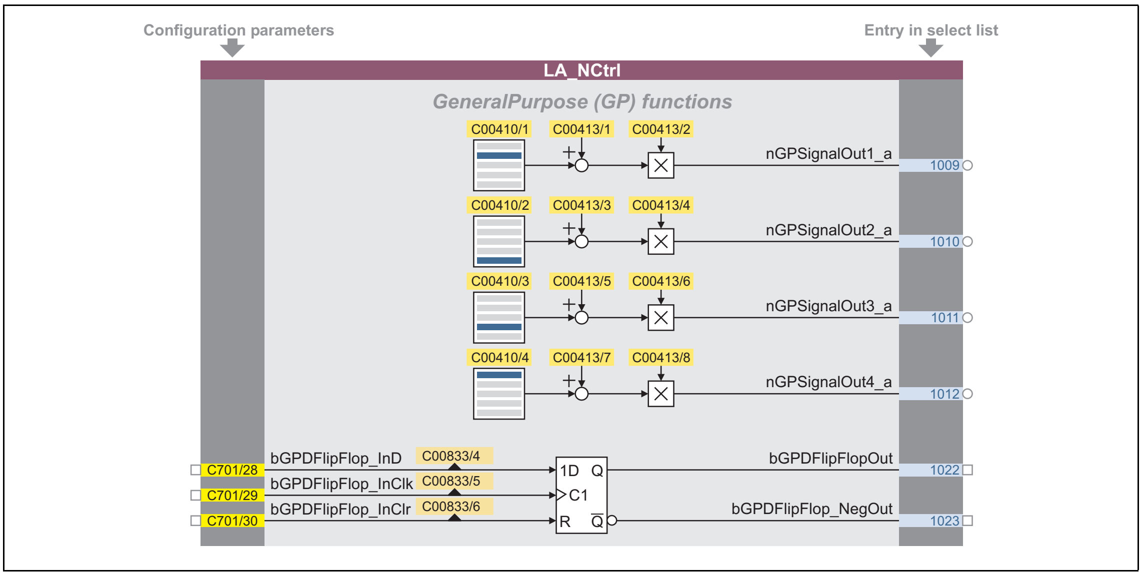


[7-2] Pre-assignment of the "Actuating drive speed" application in the "Terminals 0" control mode

Configuration parameters for "GeneralPurpose" functions



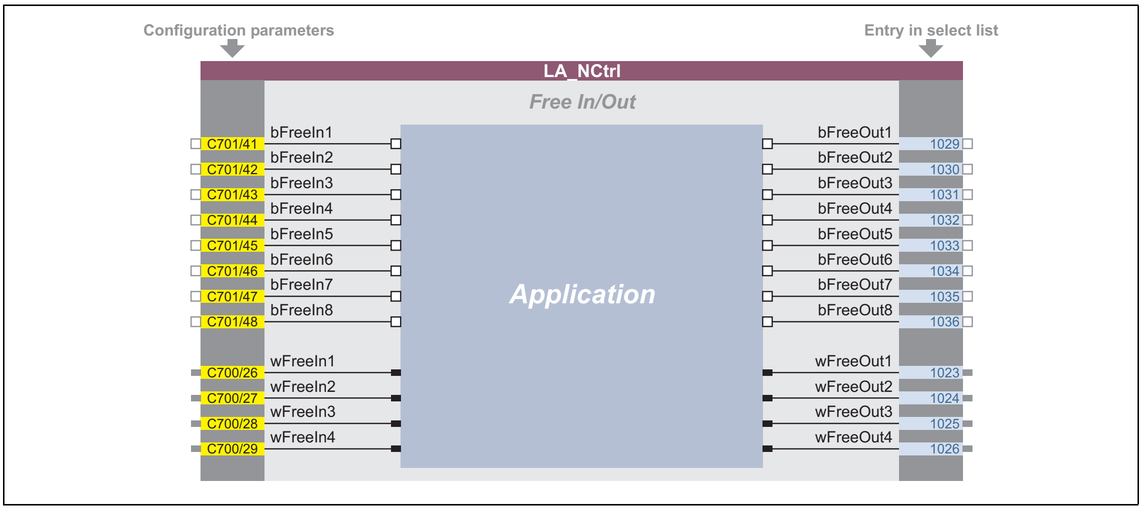
[7-3] "GeneralPurpose" functions



[7-4] "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.



[7-5] Free inputs/outputs

Related topics:

- User-defined terminal assignment (📖 227)
- "GeneralPurpose" functions (📖 290)

7.3 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, freely available "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. 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 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. Consider the influence of the motor temperature. 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 HighLine version does not meet safety-related functions except STO (Safe Torque Off), you must observe that all safety-related aspects are realised by the plant constructor.

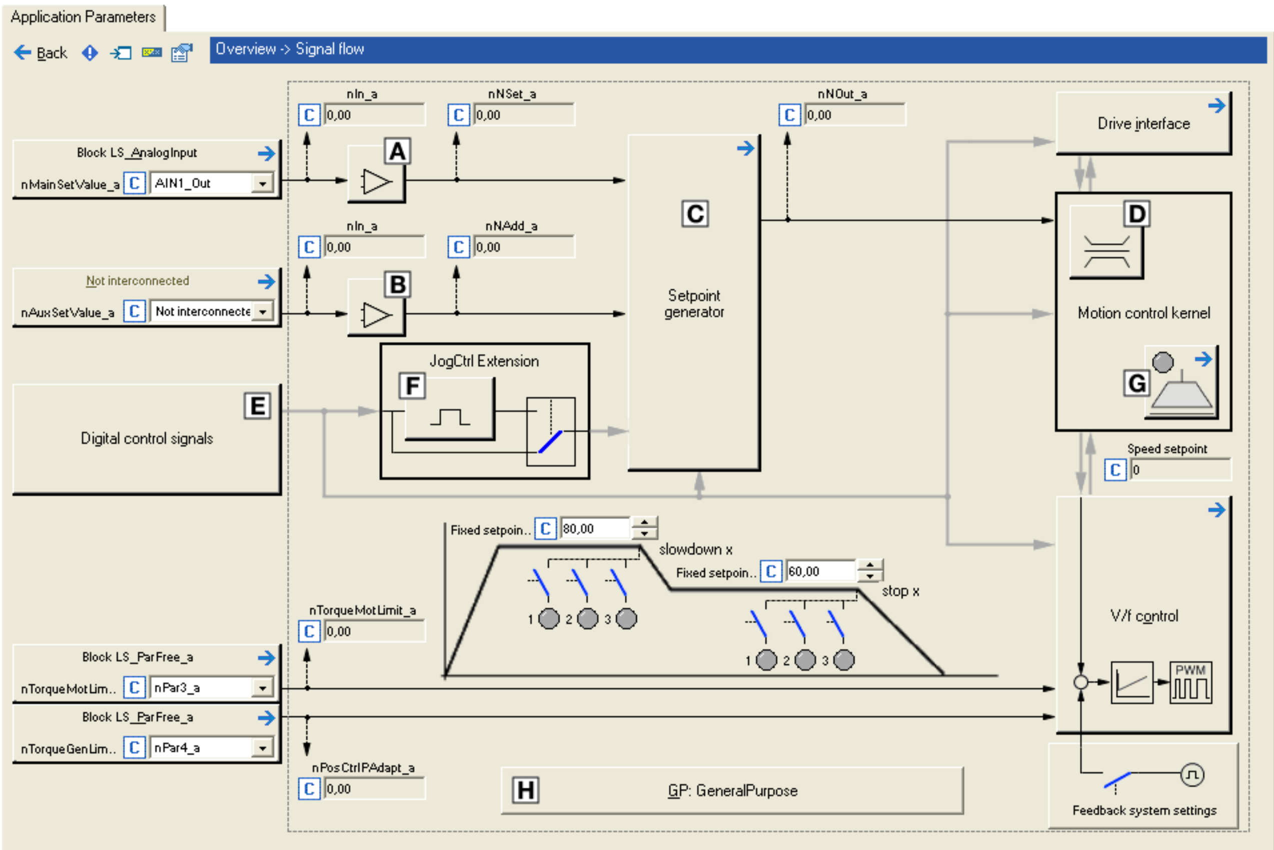
Especially in the case of an outdoor use or in wet areas, you must consider the corresponding discharge currents when operated with a fault current circuit breaker.

Highly dynamic applications and jerk-free traversing profiles require table positioning or sequential positioning control (with HighLine).

Related topics:

- Commissioning of the "Switch-off positioning" technology application (📖 54)

7.3.1 Basic signal flow



[7-6] 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

7.3.2 Internal interfaces | application block "LA_SwitchPos"

Note!

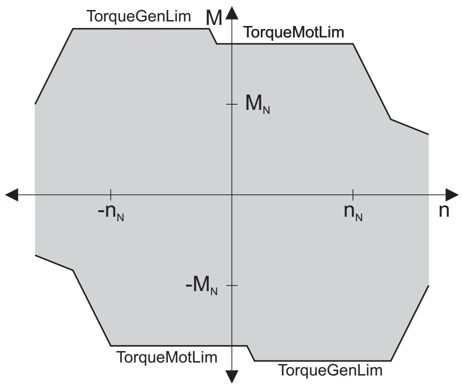
The connectors grayed out in the following table are hidden in the function block editor in the Lenze setting.

- These connections can be shown via the **Connector visibilities** command in the *Context menu* of the application block.

Inputs

Identifier	Data type	Information/possible settings	
wCANDriveControl	WORD	Control word via system bus (CAN) for device control • 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) for device control • 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. • 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**). • See the subchapter "Interface to safety system" of the chapter on basic drive functions for a detailed description of the individual control bits.	
bCInh	BOOL	Enable/Inhibit controller	
		FALSE	Enable controller: The controller switches to the "OperationEnabled" device status if no other source for controller inhibit is active. • 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 message 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. • If the fault still exists, the error status remains unchanged.
bSetQuickstop	BOOL	Enable quick stop (QSP) • Also see device command "Activate/deactivate quick stop".	
		TRUE	Activate quick stop • 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). • The motor is kept at a standstill during closed-loop operation. • A pulse inhibit is set if the auto-DCB function has been activated via C00019.
		FALSE	Deactivate quick stop • The quick stop is deactivated if no other source for the quick stop is active. • C00159 displays a bit code of active sources/causes for the quick stop.

<table>
<tr><th>Identifier
Data type</th><th colspan="2">Information/possible settings</th></tr>
<tr><td rowspan="4">bSetDCBrake
BOOL</td><td colspan="2">Manual DC-injection braking (DCB)
• Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking".</td></tr>
<tr><td colspan="2">Note!
Holding braking is not possible when this braking mode is used!
Use the basic "Holding brake control" function for controlling the holding brake with a low rate of wear.</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.
• The braking effect stops when the rotor is at standstill.
• After the hold time (C00107) has expired, the controller sets the pulse inhibit.</td></tr>
<tr><td rowspan="2">bRFG_Stop
BOOL</td><td colspan="2">Ramp function generator: Maintain the current value of the main setpoint integrator
• 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".
• 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
INT</td><td colspan="2">Additive voltage impression
• An additional setpoint for the motor voltage can be specified via this process input.
• If there are, for instance, different loads at the motor output end, it is possible to apply a voltage boost at the starting time.
• If the value is negative, the voltage is reduced.
• Scaling: 16384 ≡ 1000 V</td></tr>
<tr><td colspan="2">Stop!
Values selected too high may cause the motor to heat up due to the resulting current!</td></tr>
<tr><td rowspan="2">nBoost_a
INT</td><td colspan="2">Additional setpoint for the motor voltage at speed = 0
• The entire voltage-frequency characteristic is provided with an offset.
• Scaling: 16384 ≡ 1000 V</td></tr>
<tr><td colspan="2">Stop!
Values selected too high may cause the motor to heat up due to the resulting current!</td></tr>
<tr><td>nPWMAngleOffset
INT</td><td colspan="2">Additional offset for the electrical angle of rotation
• If a torque is connected, e.g. dynamic acceleration processes can be generated.
• Scaling: ±32767 ≡ ±180 ° angle of rotation</td></tr>
</table>

<table>
<tr><th>Identifier
Data type</th><th colspan="2">Information/possible settings</th></tr>
<tr><td rowspan="2">nTorqueMotLim_a
nTorqueGenLim_a
INT</td><td colspan="2">Torque limitation in motor mode and in generator mode
• These input signals are directly transferred to the motor control to limit the controller's maximum torque in motor and generator mode.
• The drive cannot output a higher torque in motor/generator mode than set here.
• The applied values (any polarity) are internally interpreted as absolute values.
• If V/f characteristic control (VFCplus) is selected, limitation is indirectly performed via a so-called I_{max} controller.
• If sensorless vector control (SLVC) is selected, the limitation has a direct effect on the torque-producing current component.
• Scaling: 16384 ≡ 100 % M_{max} (C00057)</td></tr>
<tr><td colspan="2">Torque limits in motor and generator mode:

</td></tr>
<tr><td rowspan="3">bSetSpeedCcw
BOOL</td><td colspan="2">Change of direction of rotation
• 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.</td></tr>
<tr><td>FALSE</td><td>Clockwise rotation (Cw)</td></tr>
<tr><td>TRUE</td><td>Direction of rotation to the left (Ccw)</td></tr>
<tr><td rowspan="3">bRLQCw
BOOL</td><td colspan="2">Activate clockwise rotation (fail-safe)
• For a detailed functional description see the L_RLQ FB.</td></tr>
<tr><td>FALSE</td><td>Quick stop</td></tr>
<tr><td>TRUE</td><td>CW rotation</td></tr>
<tr><td rowspan="3">bRLQCcw
BOOL</td><td colspan="2">Activate counter-clockwise rotation (fail-safe)
• For a detailed functional description see the L_RLQ FB.</td></tr>
<tr><td>FALSE</td><td>Quick stop</td></tr>
<tr><td>TRUE</td><td>Counter-clockwise rotation</td></tr>
<tr><td>nMainSetValue_a
INT</td><td colspan="2">Main speed setpoint
• Offset and gain of this input signal can be set in C00696 and C00670 for a simple signal 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 the L_NSet FB.</td></tr>
<tr><td>nAuxSetValue_a
INT</td><td colspan="2">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 additional speed setpoint can be linked arithmetically with the main speed setpoint behind the ramp function generator.
• The additional speed setpoint can be shown via ramp times of a second ramp function generator.
• For a detailed functional description see the L_NSet FB.</td></tr>
</table>

Identifier Data type	Information/possible settings
Switch-off positioning	
bJogCtrlInputSel1 bJogCtrlInputSel2 BOOL	Selection inputs for a binary coded selection of the switch-off position 1 ... 3 • Activation of the signal pairs *bJogCtrlSlowDown1/bJogCtrlStop1*, *bJogCtrlSlowDown2/bJogCtrlStop2* or *bJogCtrlSlowDown3/bJogCtrlStop3* according to the Truth table for activating the pre-switch off.
bJogCtrlRfgIn 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 bJogCtrlJog2 BOOL	Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint • 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. • To achieve the desired behaviour (starting at high speed, pre-switch off at low speed), both inputs must be set to TRUE. • 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. • 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 bJogCtrlSlowDown2 bJogCtrlSlowDown3 BOOL	Activation of fixed setpoint 2 in the downstream L_NSet FB • These inputs only have a function if they have been previously activated via *bJogCtrlInputSel1* and *bJogCtrlInputSel2* (see Truth table for activating the pre-switch off).
bJogCtrlStop1 bJogCtrlStop2 bJogCtrlStop3 BOOL	Ramping down of the ramp function generator in the downstream L_NSet FB • These inputs only have a function if they have been previously activated via *bJogCtrlInputSel1* and *bJogCtrlInputSel2* (see Truth table for activating the pre-switch off).
bJogSpeed4 bJogSpeed8 BOOL	Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint • A fixed setpoint for the setpoint generator can be activated instead of the main setpoint via these selection inputs. • The selection inputs are binary coded. • For a detailed functional description see the L_NSet FB.
bJogRamp1 ... bJogRamp8 BOOL	Selection inputs for alternative acceleration/deceleration times for the main setpoint • The four selection inputs are binary coded, therefore 15 alternative acceleration/deceleration times can be selected. • 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). • Alternative acceleration times are selected in C00101/1...15. • The selection of the alternative deceleration times is carried out in C00103/1...15. • For a detailed functional description see the L_NSet FB.
MCK basic functions	
bMBRKRelease BOOL	Holding brake control: Release/apply brake • In conjunction with the operating mode selected in C02580 (Lenze setting: "Brake control off"). FALSE: Apply brake. • During automatic operation, the internal brake logic controls the brake. TRUE: Release brake. • During automatic operation, the internal brake logic is deactivated and the brake is released. If the brake control has inhibited the controller, this inhibit is deactivated again.

Identifier Data type	Information/possible settings
GP: GeneralPurpose The following inputs are interconnected with logic/arithmetic functions on application level for free usage. ▸ "GeneralPurpose" functions	
nGPAnalogSwitchIn1_a nGPAnalogSwitchIn2_a INT	Analog switch: Input signals • The input signal selected via the selection input *bGPAnalogSwitchSet* is output at output *nGPAnalogSwitchOut_a*.
bGPAnalogSwitchSet BOOL	Analog switch: Selection input FALSE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn1_a* TRUE: *nGPAnalogSwitchOut_a = nGPAnalogSwitchIn2_a*
nGPArithmetikIn1_a nGPArithmetikIn2_a INT	Arithmetic: Input signals • The arithmetic function is selected in C00338. • The result is output at output *nGPArithmetikOut_a*.
nGPMulDivIn_a INT	Multiplication/Division: Input signal • The factor for the multiplication can be set in C00699/1 (numerator) and C00699/2 (denominator). • The result is output at output *nGPMulDivOut_a*.
bGPDigitalDelayIn BOOL	Binary delay element: Input signal • The on-delay can be set in C00720/1. • The off-delay can be set in C00720/2. • The time-delayed input signal is output at output *bGPDigitalDelayOut*.
bGPLogicIn1 bGPLogicIn2 bGPLogicIn3 BOOL	Binary logic: Input signals • The logic operation is selected in C00820. • The result is output at output *bGPLogicOut*.
nGPCompareIn1_a nGPCompareIn2_a INT	Analog comparison: Input signals • The comparison operation is selected in C00680. • Hysteresis and window size can be set in C00680 and C00682. • If the comparison statement is true, the output *bGPCompareOut* will be set to TRUE.
bGPDFlipFlop_InD bGPDFlipFlop_InClk bGPDFlipFlop_InClr BOOL	D-FlipFlop: Input signals • Data, clock and reset input
Free inputs The following inputs can freely be interconnected on the application level. The signals can be transferred from the I/O level to the application level via these inputs.	
bFreeIn1 ... bFreeIn8 BOOL	Free inputs for digital signals
wFreeIn1 ... wFreeIn4 WORD	Free inputs for 16-bit signals
dnFreeIn1_p ... dnFreeIn2_p DINT	Free inputs for 32-bit signals

Outputs

Identifier Data type	Value/meaning
wDriveControlStatus WORD	Status word of the controller (based on DSP-402) • The status word contains information on the currents status of the drive controller. • See the "wDeviceStatusWord status word" subchapter of the chapter on device control for a detailed description of the bit assignment.
wStateDetermFailNoLow 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. • "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. • "SafeTorqueOff" device status is active.
bDriveReady BOOL	TRUE	Controller is ready for operation. • "SwitchedOn" device status is active. • 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	Clockwise rotation (Cw)
	TRUE	Direction of rotation to the left (Ccw)
bSpeedActCompare BOOL	Result of the speed comparison	
	TRUE	During open-loop operation: Speed setpoint < Comparison value (C00024)
		During closed-loop operation: 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 • Scaling: 16384 ≡ 100 % I_{max_mot} (C00022)	
nMotorSpeedSet_a INT	Speed setpoint • Scaling: 16384 ≡ 100 % reference speed (C00011)	
nMotorSpeedAct_a INT	Actual speed value • Scaling: 16384 ≡ 100 % reference speed (C00011)	
nMotorTorqueAct_a INT	Actual torque • 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. • Scaling: 16384 ≡ 100 % M_{max} (C00057)	
nDCVoltage_a INT	Actual DC-bus voltage • Scaling: 16384 ≡ 1000 V	
nMotorVoltage_a INT	Current motor voltage/inverter output voltage • Scaling: 16384 ≡ 1000 V	

Identifier Data type	Value/meaning	
MCK basic functions		
bBrakeReleaseOut BOOL	Holding brake control: Trigger signal for the holding brake control switching element via a digital output • Use bit 0 in C02582 to activate inverted switching element triggering.	
	FALSE	Apply brake.
	TRUE	Release brake.
bBrakeReleased BOOL	Holding brake control: Status signal of the brake control with regard to the release and application times of the brake	
	FALSE	Brake applied (after the brake application time has expired).
	TRUE	Brake released (after the brake release time has expired).
GP: GeneralPurpose The following outputs are interconnected with logic/arithmetic functions on application level for free usage. "GeneralPurpose" functions		
nGPAnalogSwitchInOut_a INT	Analog switch: Output signal	
nGPArithmetikOut_a INT	Arithmetic: Output signal	
nGPMulDivOut_a INT	Multiplication/Division: Output signal	
bGPDigitalDelayOut BOOL	Binary delay element: Output signal	
bGPLogicOut BOOL	Binary logic: Output signal	
bGPCompareOut BOOL	Analog comparison: Output signal	
bGPSignalOut1 ... bGPSignalOut4 BOOL	Binary signal monitor: Output signals • The signal sources to be output are selected in C00411/1...4. • A bit coded inversion of the output signals can be parameterised in C00412.	
nGPSignalOut1_a ... nGPSignalOut4_a BOOL	Analog signal monitor: Output signals • The signal sources to be output are selected in C00410/1...4. • Gain and offset for each output signal can be parameterised in C00413/1...8.	
bGPDFlipFlop_Out BOOL	D-FlipFlop: Output signal	
bGPDFlipFlop_NegOut BOOL	D-FlipFlop: Negated output signal	
Free outputs The following outputs can freely be interconnected on the application level. The signals from the application level can be transferred to the I/O level via these outputs.		
bFreeOut1 ... bFreeOut8 BOOL	Free outputs for digital signals	
wFreeOut1 ... wFreeOut4 WORD	Free outputs for 16-bit signals	
dnFreeOut1_p dnFreeOut2_p WORD	Free outputs for 32-bit signals	

7.3.2.1 Truth table for activating the pre-switch off

<table>
<tr><th colspan="2">Input</th><th rowspan="2">Function</th><th rowspan="2">Response in the setpoint generator (FB L_NSet)</th></tr>
<tr><th>bJogCtrl InputSel1</th><th>bJogCtrl InputSel2</th></tr>
<tr><td>FALSE</td><td>FALSE</td><td>Pre-switch off inactive</td><td>No response
• The input signal bJogCtrlRfgIn is output directly at output bRfgOut.
• The input signals bJogCtrlJog1 and bJogCtrlJog2 are passed through 1:1 to the downstream FB L_NSet for the selection of fixed setpoints.</td></tr>
<tr><td>TRUE</td><td>FALSE</td><td>The bJogCtrlSlowDown1 and bJogCtrlStop1 inputs are evaluated.</td><td rowspan="3">Pre-switch off can be activated
• If the slowdown function is activated via the selected bJogCtrlSlowDown input, fixed setpoint 2 is activated in the setpoint generator.
• If the stop function is activated via the selected bJogCtrlStop input, the setpoint generator is deactivated.</td></tr>
<tr><td>FALSE</td><td>TRUE</td><td>The bJogCtrlSlowDown2 and bJogCtrlStop2 inputs are evaluated.</td></tr>
<tr><td>TRUE</td><td>TRUE</td><td>The inputs bSJogCtrllowDown3 and bJogCtrlStop3 are evaluated.</td></tr>
</table>

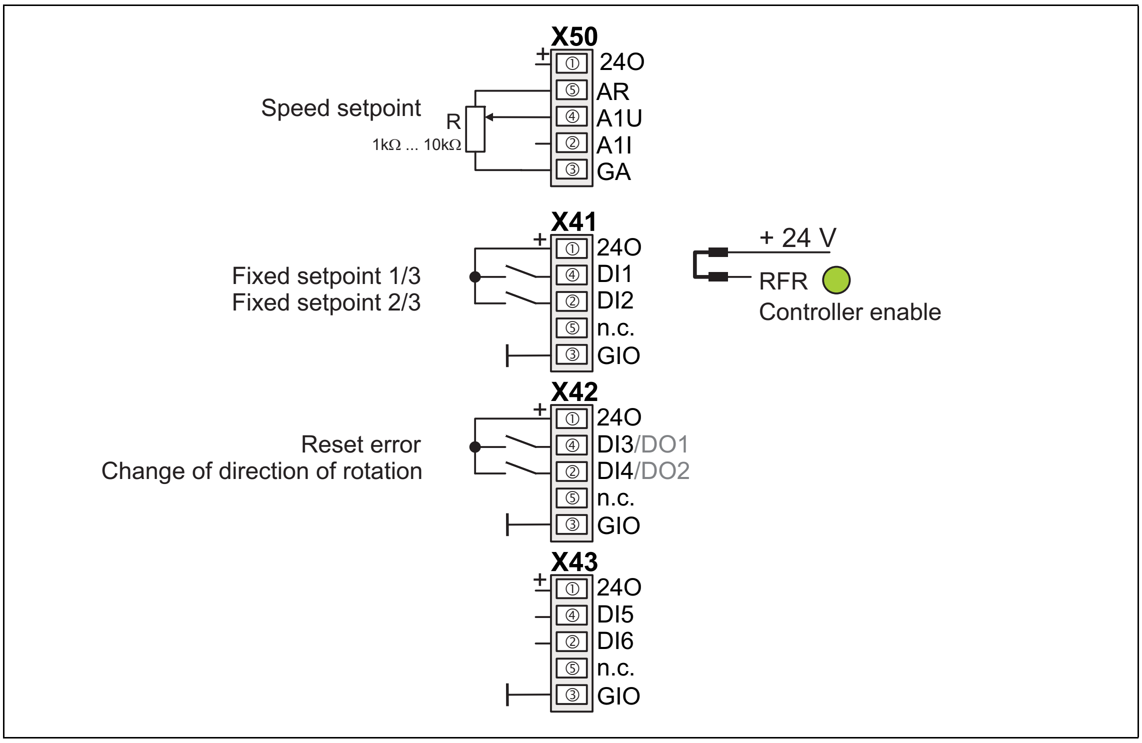
[7-1] Truth table for activating the pre-switch off

7.3.3 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.

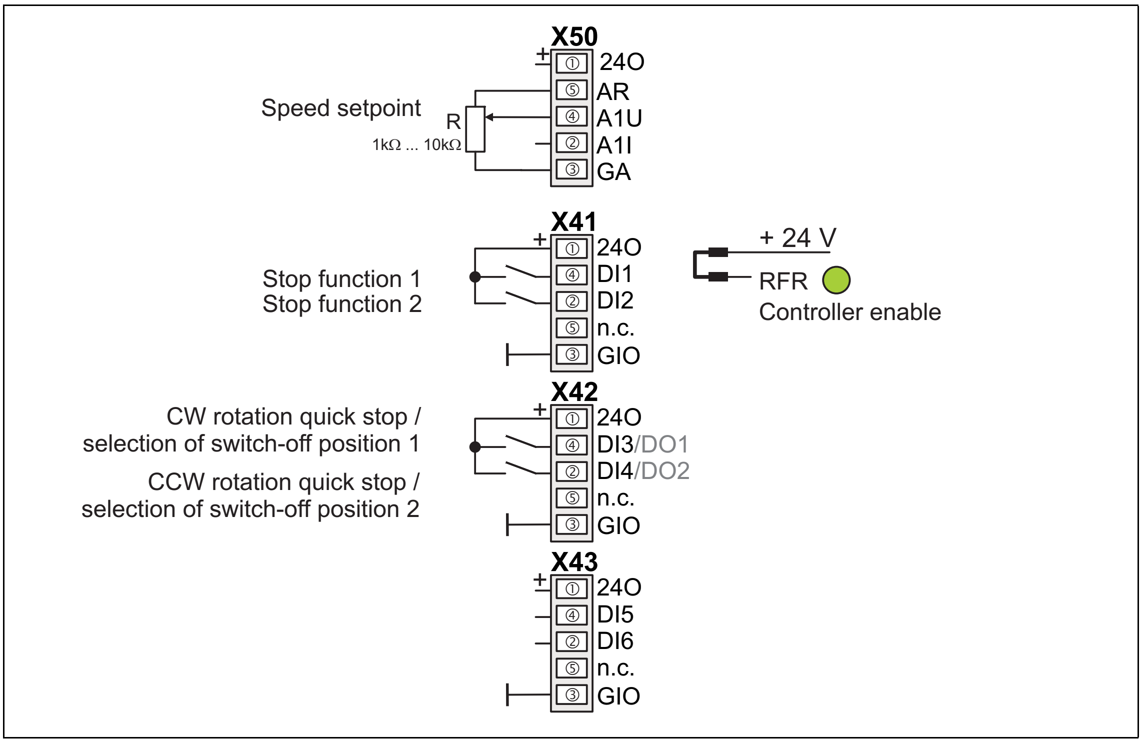
	Control mode (C00007)							
	10: Terminals 0	12: Terminals 2	14: Terminals 11	16: Terminal 16	20: Keypad	21: PC	30: CAN	40: MCI
Digital input terminals								
RFR	**Controller enable**	**Controller enable / Reset of error message** bFailReset (In case of 8400 protec, the RFR terminal is internally set to HIGH level via wire jumper.)						
X41/DI1	**Fixed setpoint 1/3** bJogCtrlJog1	**Stop function 1** bJogCtrlStop1		**Fixed setpoint 1/3** bJogCtrlJog1	-	-	**Stop function 1** bJogCtrlStop1	
X41/DI2	**Fixed setpoint 2/3** bJogCtrlJog2	**Stop function 2** bJogCtrlStop2	**Selection: Pre-switch off 1** bJogCtrlSlowDown1	**Fixed setpoint 2/3** bJogCtrlJog2	-	-	**Selection: Pre-switch off 1** bJogCtrlSlowDown1	
X42/DI3	**Reset error** bFailReset	**CW rotation quick stop** bRLQCw Selection: Switch-off position 1 bJogCtrlInputSel1		**CW rotation quick stop** bRLQCw	-	-	**Stop function 2** bJogCtrlStop2	
X42/DI4	**Change of direction of rotation** bSetSpeedCcw	**CCW rotation quick stop** bRLQCcw Selection: Switch-off position 2 bJogCtrlInputSel2		**CCW rotation quick stop** bRLQCcw	-	-	**Selection: Pre-switch off 2** bJogCtrlSlowDown2	
X43/DI5	-	-	**Stop function 2** bJogCtrlStop2	-	-	-	**Stop function 3** bJogCtrlStop3	
X43/DI6	-	-	**Selection: Pre-switch off 2** bJogCtrlSlowDown2	-	-	-	**Selection: Pre-switch off 3** bJogCtrlSlowDown3	
Analog input terminals								
X50/A1U	**Main speed setpoint** nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)				-	-	**Additional speed setpoint** nAuxSetValue_a 10 V ≡ 100 % reference speed (C00011)	
X50/A1I	-	-	-	-	-	-	-	-
Digital output terminals								
DDO1/DO2 is only available after respective change of the function assignment in C00116.								
X42/DO1	**Status "Drive is ready"** bDriveReady							
X42/DO2	-							

7.3.3.1 Terminals 0



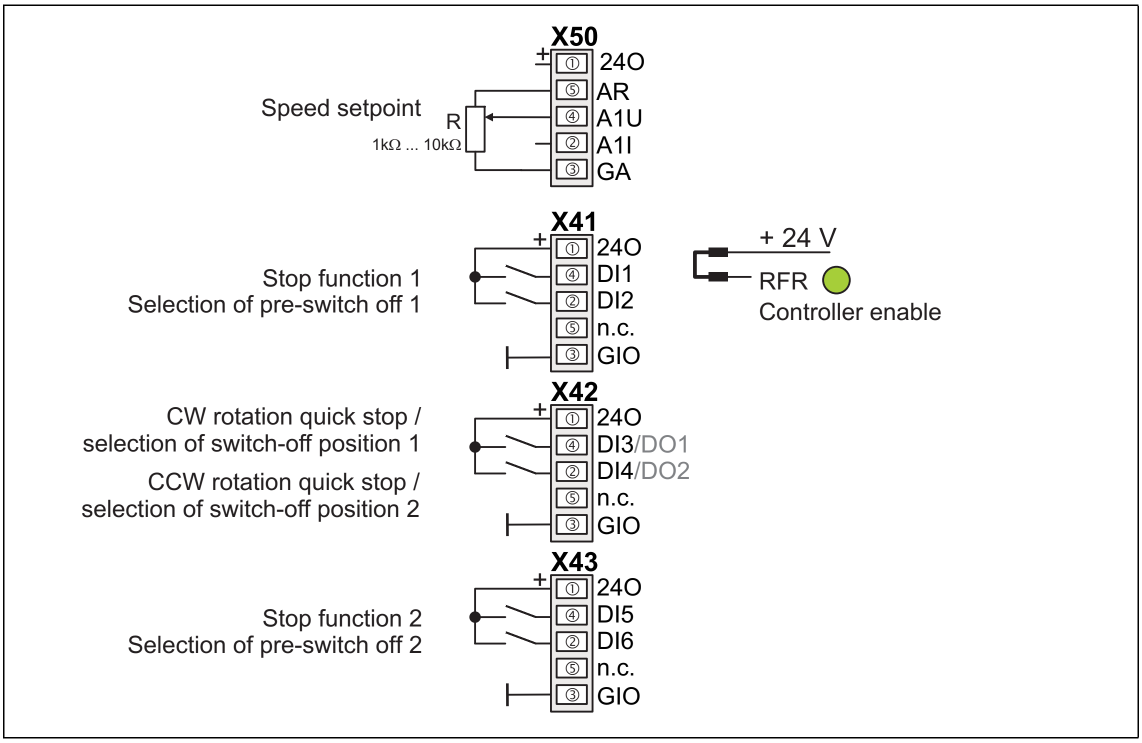
Connection	Assignment		Connection	Assignment
X41/DI1	LA_SwitchPos.bJogCtrlJog1		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_SwitchPos.bJogCtrlJog2		X42/DO1	-
X42/DI3	LA_SwitchPos.bFailReset		X42/DO2	LA_SwitchPos.bDriveReady
X42/DI4	LA_SwitchPos.bSetSpeedCcw		**X50/A1U**	LA_SwitchPos.nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI5	-			
X43/DI6	-			

7.3.3.2 Terminals 2



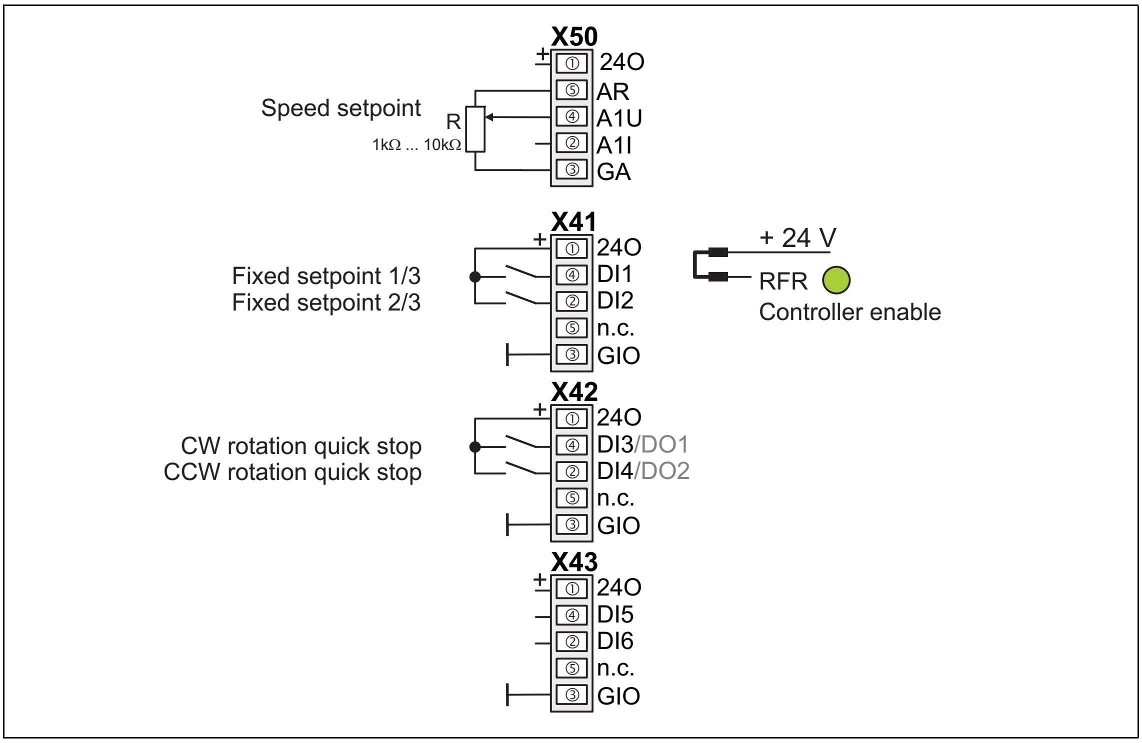
Connection	Assignment		Connection	Assignment
X41/DI1	LA_SwitchPos: bJogCtrlStop1		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_SwitchPos: bJogCtrlStop2			
X42/DI3	LA_SwitchPos: bRLQCw LA_SwitchPos: bJogCtrlInputSel1		X42/DO1	-
X42/DI4	LA_SwitchPos: bRLQCcw LA_SwitchPos: bJogCtrlInputSel2		X42/DO2	LA_SwitchPos.bDriveReady
X43/DI5	-		**X50/A1U**	LA_SwitchPos.nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	-			

7.3.3.3 Terminals 11



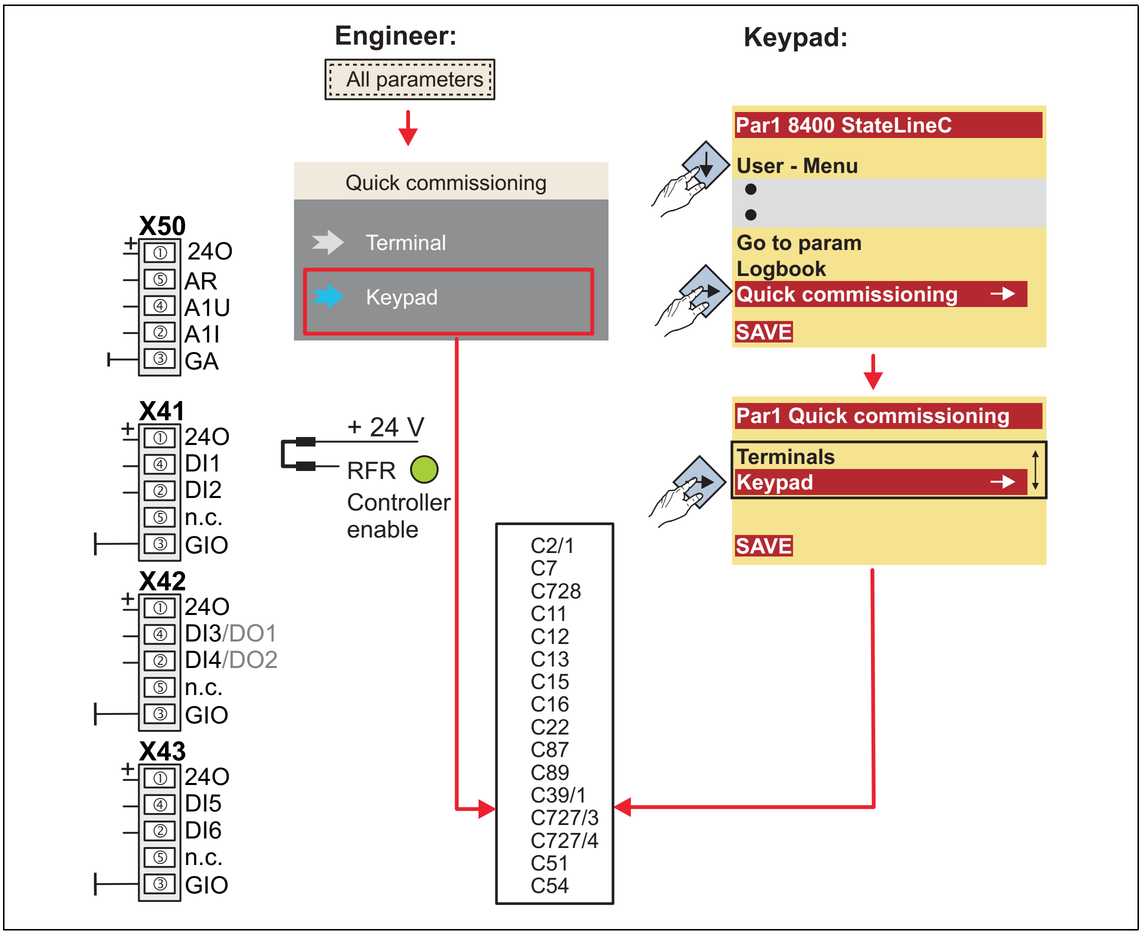
Connection	Assignment		Connection	Assignment
X41/DI1	LA_SwitchPos: bJogCtrlStop1		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_SwitchPos: bJogCtrlSlowDown1			
X42/DI3	LA_SwitchPos: bRLQCw LA_SwitchPos: bJogCtrlInputSel1		X42/DO1	-
X42/DI4	LA_SwitchPos: bRLQCcw LA_SwitchPos: bJogCtrlInputSel2		X42/DO2	LA_SwitchPos.bDriveReady
X43/DI5	LA_SwitchPos: bJogCtrlStop2		**X50/A1U**	LA_SwitchPos.nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	LA_SwitchPos: bJogCtrlSlowDown2			

7.3.3.4 Terminal 16



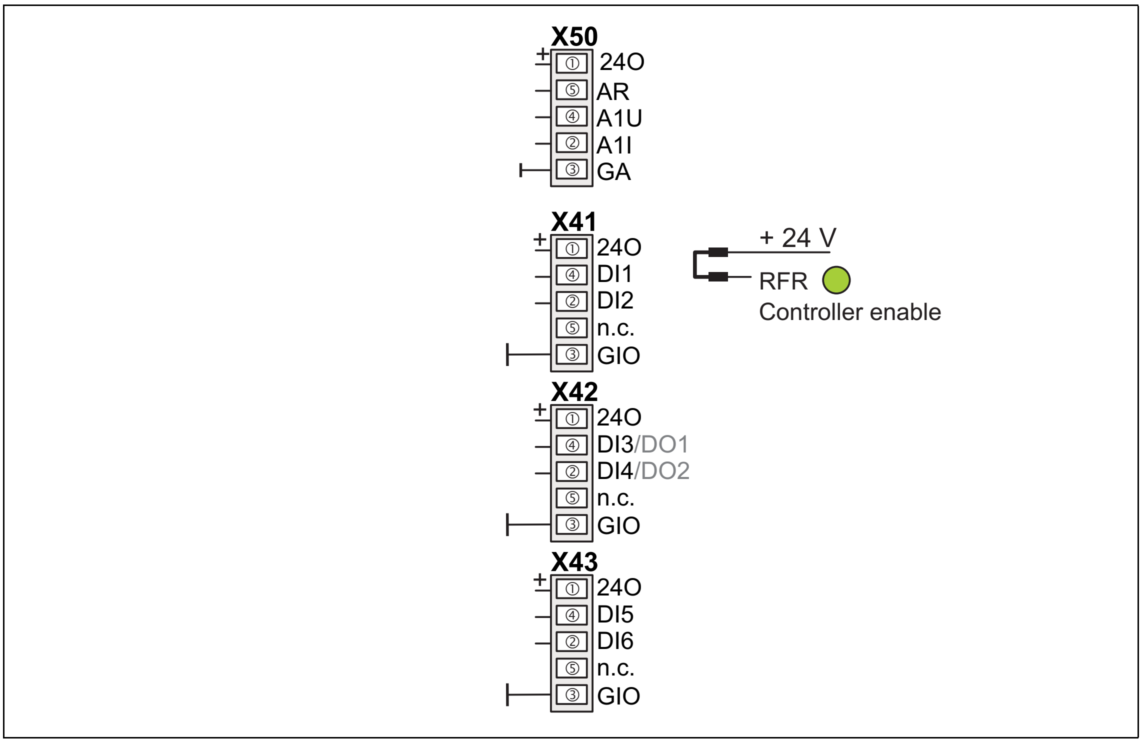
Connection	Assignment		Connection	Assignment
X41/DI1	LA_SwitchPos: bJogCtrlJog1		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_SwitchPos: bJogCtrlJog2			
X42/DI3	LA_SwitchPos: bRLQCw		X42/DO1	-
X42/DI4	LA_SwitchPos: bRLQCcw		X42/DO2	LA_SwitchPos.bDriveReady
X43/DI5	-		**X50/A1U**	LA_SwitchPos.nMainSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	-			

7.3.3.5 Keypad



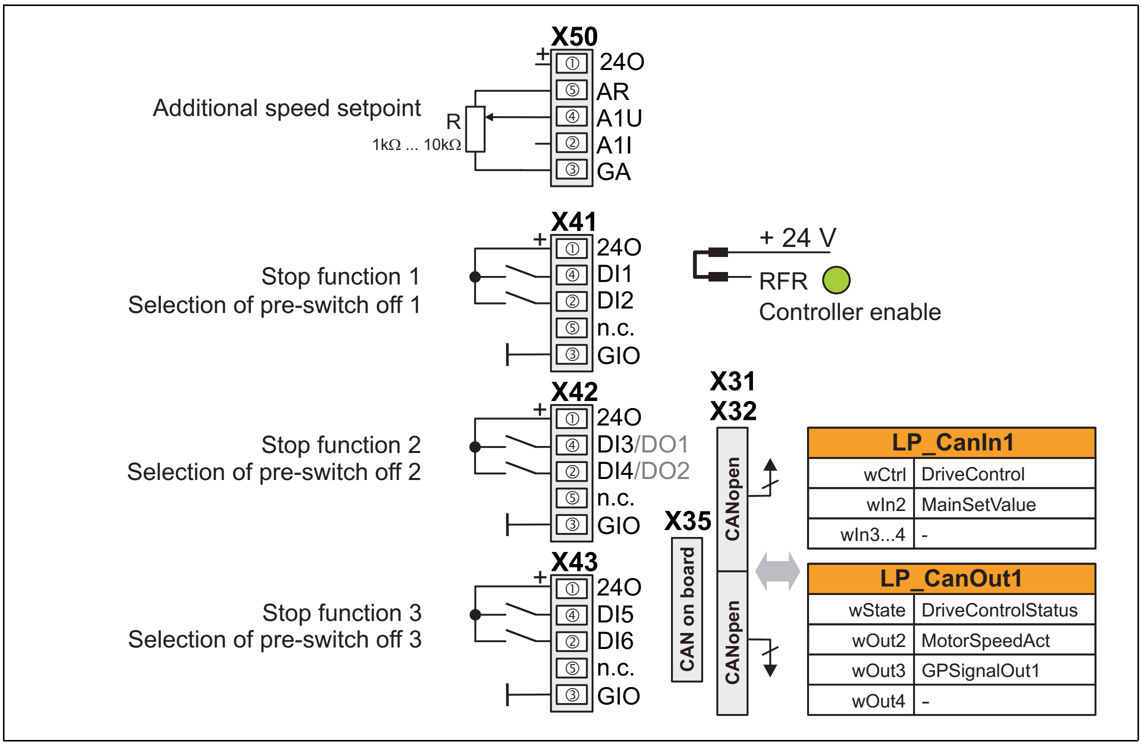
Connection	Assignment		Connection	Assignment
X41/DI1	-		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	-			
X42/DI3	-		X42/DO1	-
X42/DI4	-		X42/DO2	-
X43/DI5	-		**X50/A1U**	-
X43/DI6	-			

7.3.3.6 PC



Connection	Assignment		Connection	Assignment
X41/DI1	-		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	-			
X42/DI3	-		X42/DO1	-
X42/DI4	-		X42/DO2	-
X43/DI5	-		**X50/A1U**	-
X43/DI6	-			

7.3.3.7 CAN



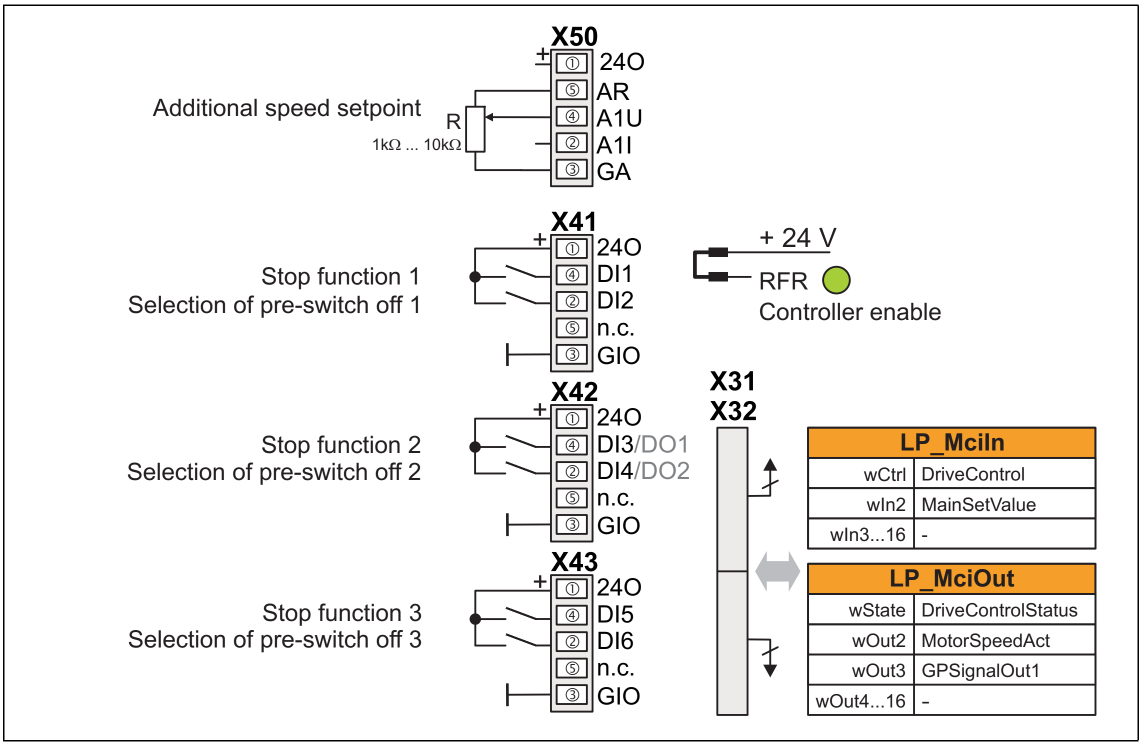
Connection	Assignment		Connection	Assignment
X41/DI1	LA_SwitchPos: bJogCtrlStop1		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_SwitchPos: bJogCtrlSlowDown1			
X42/DI3	LA_SwitchPos: bJogCtrlStop2		X42/DO1	-
X42/DI4	LA_SwitchPos: bJogCtrlSlowDown2		X42/DO2	-
X43/DI5	LA_SwitchPos: bJogCtrlStop3		**X50/A1U**	LA_SwitchPos.nAuxSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	LA_SwitchPos: bJogCtrlSlowDown3			

▶ Process data assignment for fieldbus communication (📖 283)

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.

7.3.3.8 MCI



Connection	Assignment		Connection	Assignment
X41/DI1	LA_SwitchPos: bJogCtrlStop1		DO1/DO2 is only available after respective change of the function assignment in C00116.	
X41/DI2	LA_SwitchPos: bJogCtrlSlowDown1			
X42/DI3	LA_SwitchPos: bJogCtrlStop2		X42/DO1	-
X42/DI4	LA_SwitchPos: bJogCtrlSlowDown2		X42/DO2	-
X43/DI5	LA_SwitchPos: bJogCtrlStop3		**X50/A1U**	LA_SwitchPos.nAuxSetValue_a 10 V ≡ 100 % reference speed (C00011)
X43/DI6	LA_SwitchPos: bJogCtrlSlowDown3			

▶ Process data assignment for fieldbus communication (📖 283)

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.

7.3.4 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 depends only on the application, not on the bus system used:

Input words	Name	Assignment
Word 1	DriveControl	Control word • For bit assignment see the table below.
Word 2	MainSetValue	Speed setpoint • Scaling: 16384 ≡ 100 % reference speed (C00011)
Word 3	-	Not preconfigured
Word 4	-	Not preconfigured
Words 5 ... 16	-	Not preconfigured • Only available in control mode "40: MCI".

Control word	Name	Function
Bit 0	SwitchOn	1 ≡ Change to the "SwitchedOn" device status • 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. • 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 (pulse inhibit)
Bit 2	SetQuickStop	1 ≡ Activate quick stop (QSP). ▸ Activate/deactivate quick stop (📖 76)
Bit 3	EnableOperation	1 ≡ Enable controller (RFR) • 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. ▸ Enable/Inhibit controller (📖 75)
Bit 4	ModeSpecific_1	Reserved (currently not assigned)
Bit 5	JogCtrlInputSel1	Binary coded selection of the switch-off position 1 ... 3 • 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) • Acknowledge error message (if the error cause has been eliminated). ▸ Reset error (📖 77)
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: 0 ≡ Apply brake 1 ≡ Release brake • 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) 1 ≡ Direction of rotation to the left (Ccw)

Output words	Name	Assignment
Word 1	DriveControlStatus	Status word • For bit assignment see the table below.
Word 2	MotorSpeedAct	Actual speed value • Scaling: 16384 ≡ 100 % reference speed (C00011)
Word 3	GPSignalOut1	Analog signal monitor: Output signal 1 • The selection of the signal source to output is executed in C00410/1. • Gain and offset for the output signal can be parameterised in C00413/1 and C00413/2. • For a detailed functional description see the L_SignalMonitor_a FB.
Word 4	-	Not preconfigured
Words 5 ... 16	-	Not preconfigured • Only available in control mode "40: MCI".

Status word	Name	Status
Bit 0	DriveFail	1 ≡ Drive controller in error status • "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 • "SwitchedOn" device status is active. • 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) 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: 1 ≡ Speed setpoint < Comparison value (C00024)
		During closed-loop operation: 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 ▸ Device state machine and device states (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 • E.g. if an overvoltage has occurred.
Bit 14	JogCtrlInputSel1	Binary coded selection of the switch-off position 1 ... 3 • Bit 5 and bit 6 of the control word.
Bit 15	JogCtrlInputSel2	

7.3.5 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 smoothing 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	

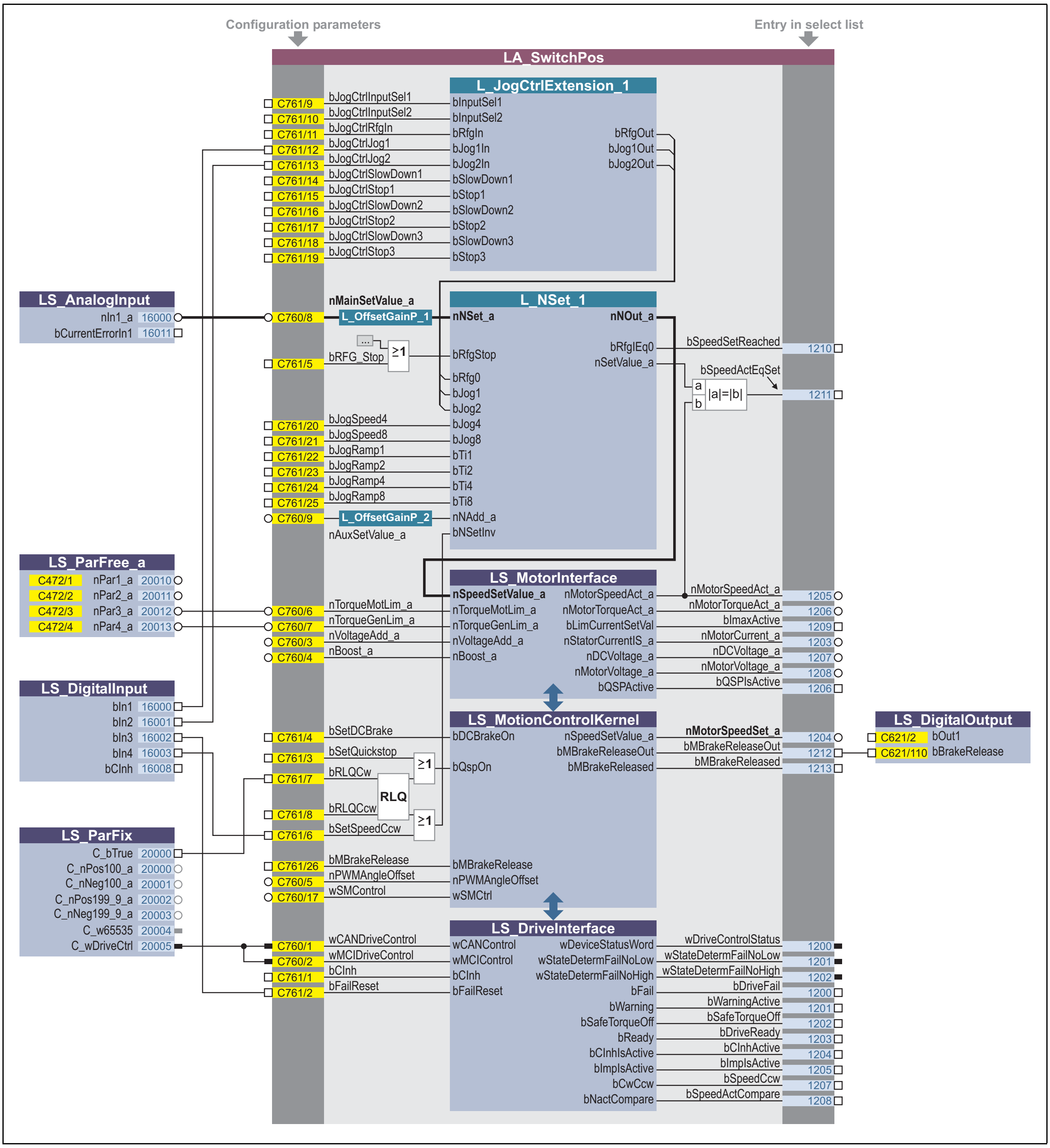
Parameter	Info	Lenze setting	
		Value	Unit
C00672	L_OffsetGainP_3: Gain	1.0000	
C00696	L_OffsetGainP_1: Offset	0.00	%
C00697	L_OffsetGainP_2: Offset	0.00	%
C00698	L_OffsetGainP_3: Offset	0.00	%
C00800	L_MPot_1: Upper limit	100.00	%
C00801	L_MPot_1: Lower limit	-100.00	%
C00802	L_MPot_1: Acceleration time	10.0	s
C00803	L_MPot_1: Deceleration 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 (📖 290)

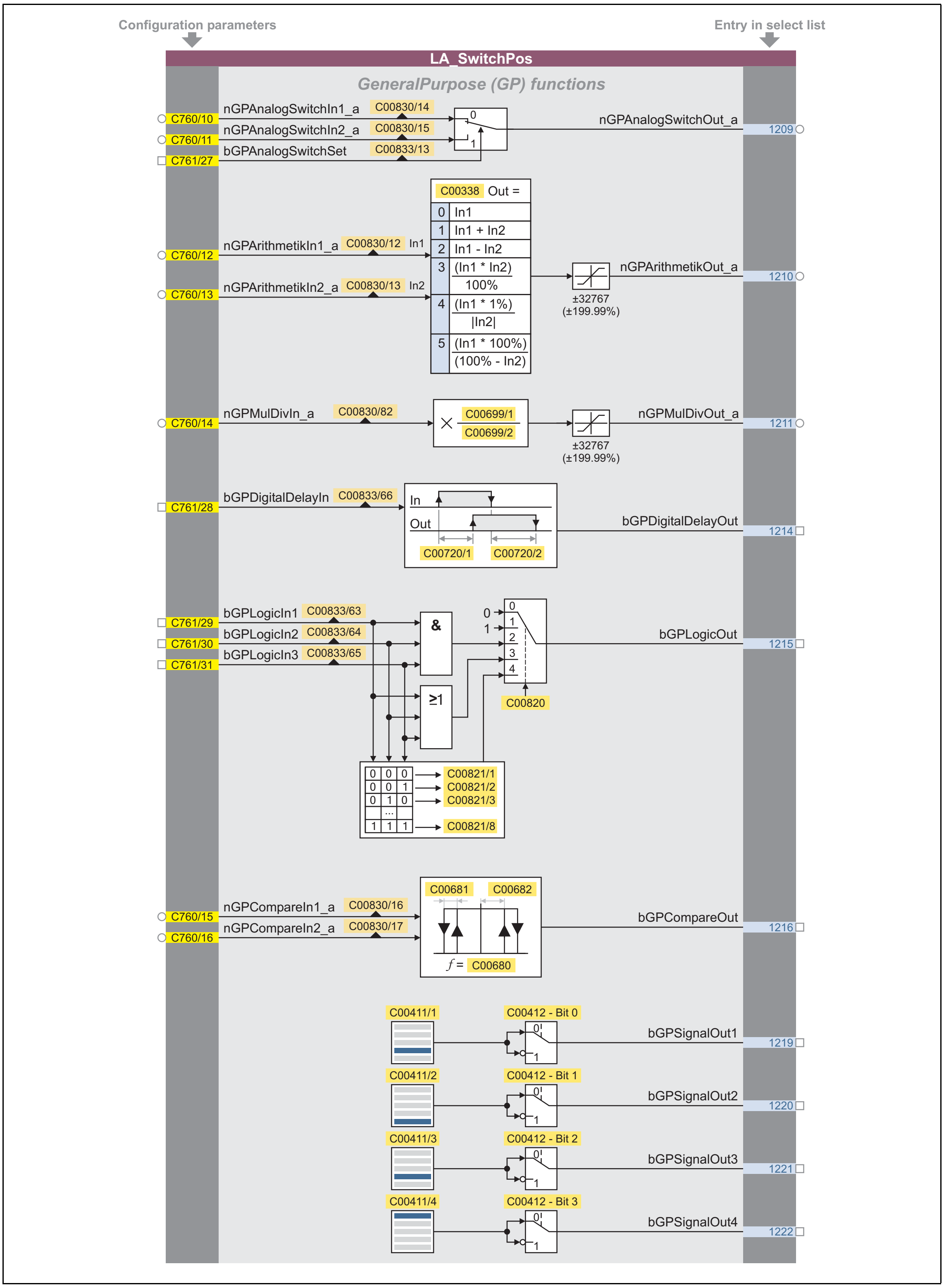
7.3.6 Configuration parameters

If required, the subcodes of C00760 and C00761 serve to change the pre-configured assignment of the application inputs:

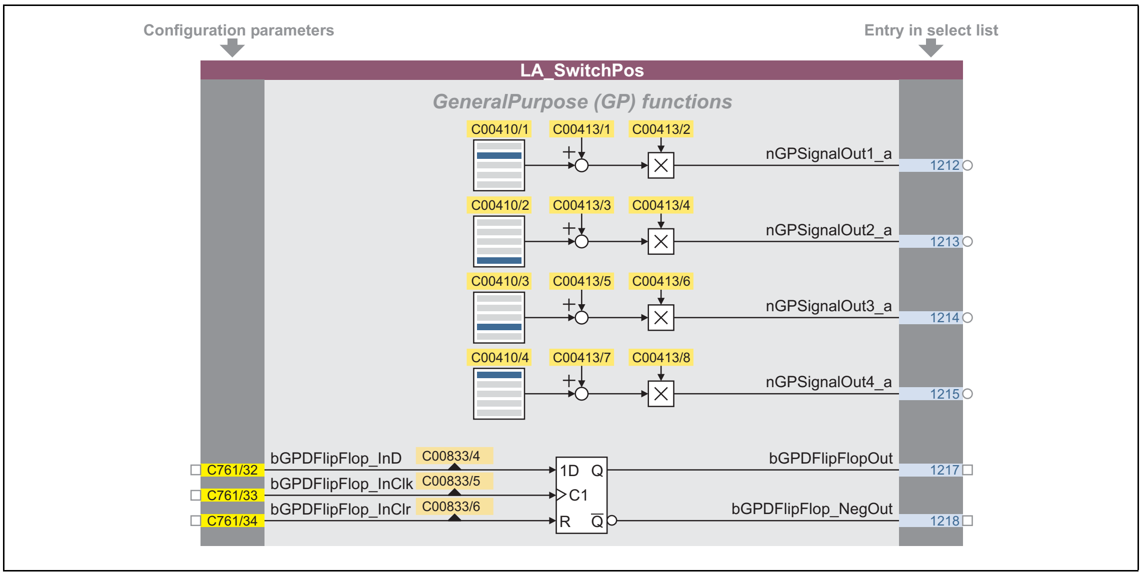


[7-7] Pre-assignment of the "Switch-off positioning" application in the "Terminals 0" control mode

Configuration parameters for "GeneralPurpose" functions



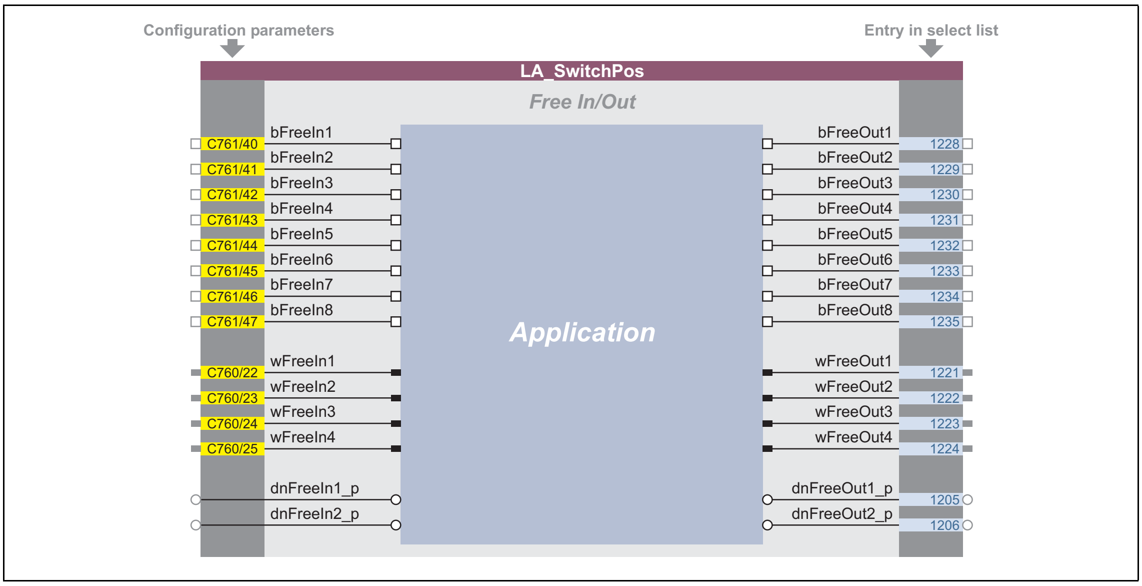
[7-8] "GeneralPurpose" functions



[7-9] "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.



[7-10] Free inputs/outputs

Related topics:

▸ User-defined terminal assignment (📖 227)
▸ "GeneralPurpose" functions (📖 290)

7.4 "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.

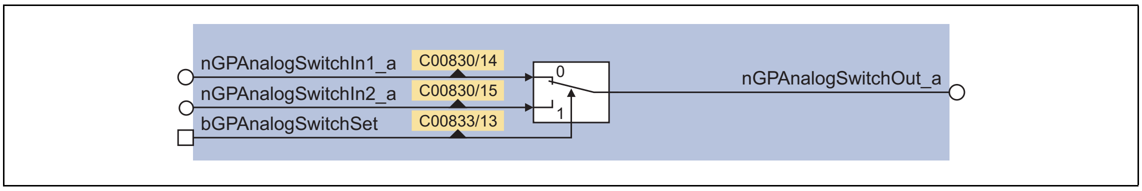
On the other hand, the outputs of the "GeneralPurpose" functions can be selected in the configuration parameters of other inputs.

Related topics:

▶ User-defined terminal assignment (📖 227)
▶ TA "Actuating drive speed": Configuration parameters (📖 260)
▶ TA "Switch-off positioning": Configuration parameters (📖 287)

7.4.1 Analog switch

This function switches between two analog input signals. The switch-over is controlled by a boolean input signal.

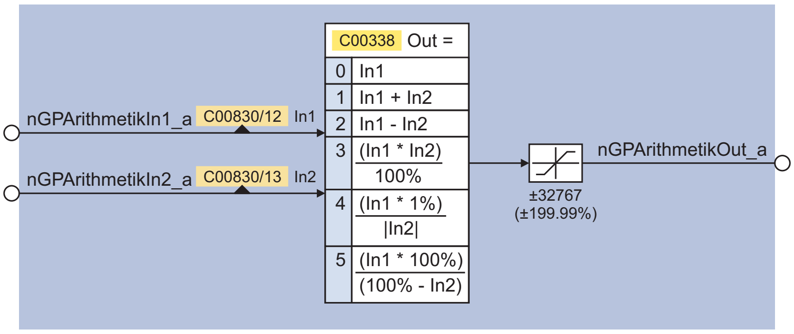


[7-11] GeneralPurpose function "Analog switch"

- For a detailed functional description see FB L_AnalogSwitch.

7.4.2 Arithmetic

This function links two analog signals arithmetically. The arithmetic function can be parameterised.



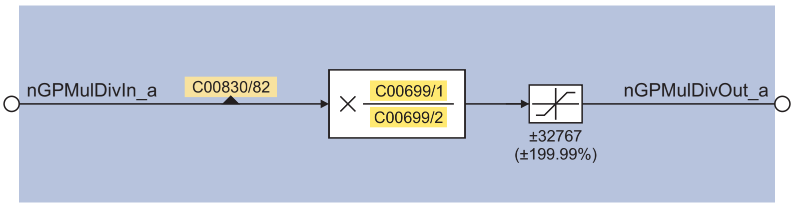
[7-12] GeneralPurpose function "Arithmetic"

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.

7.4.3 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).



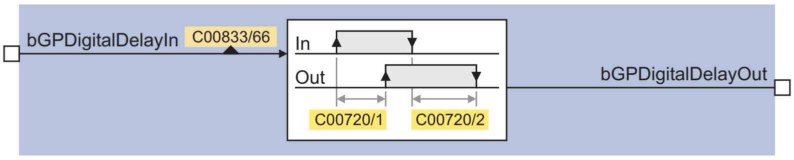
[7-13] GeneralPurpose function "Multiplication/division"

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.

7.4.4 Binary delay element

This function timely delays binary signals. On-delay and off-delay can be parameterised separately.



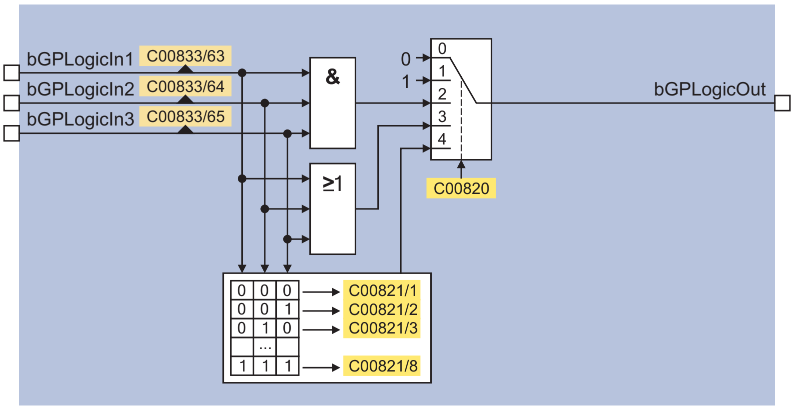
[7-14] GeneralPurpose function "Binary delay element"

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.

7.4.5 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.



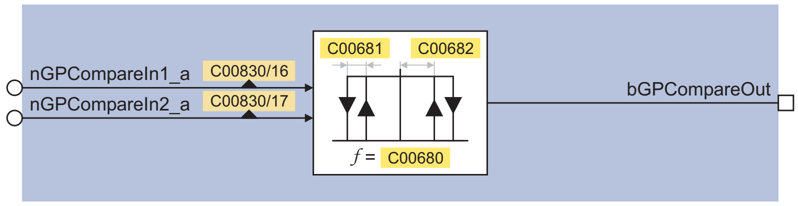
[7-15] GeneralPurpose function "Binary logic"

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.

7.4.6 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.



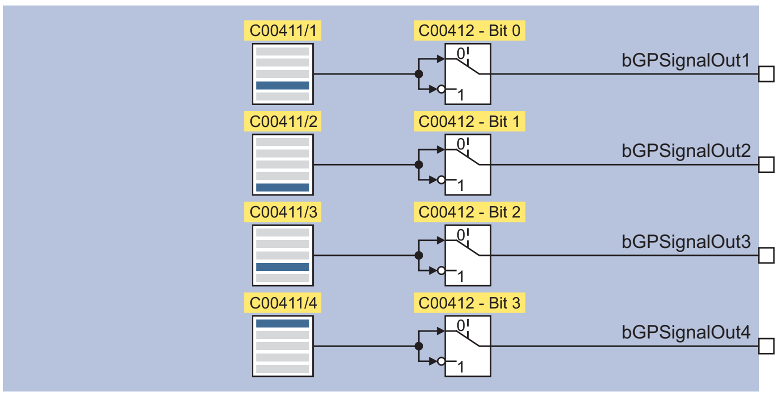
[7-16] GeneralPurpose function "Analog comparison"

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.

7.4.7 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.



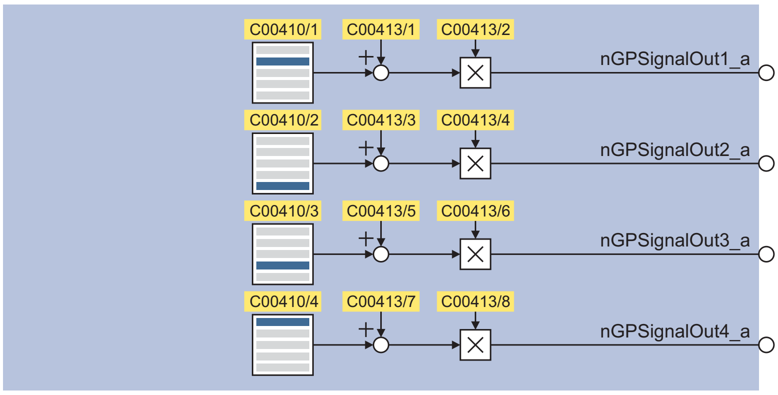
[7-17] GeneralPurpose function "Binary signal monitor"

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.

7.4.8 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.



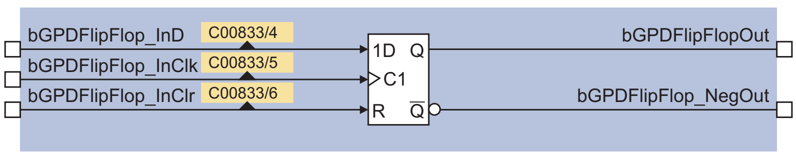
[7-18] GeneralPurpose function "Analog signal monitor"

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.

7.4.9 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.



[7-19] GeneralPurpose function "D-FlipFlop" (clock-edge controlled)

- For a detailed functional description see FB L_DFlipFlop.

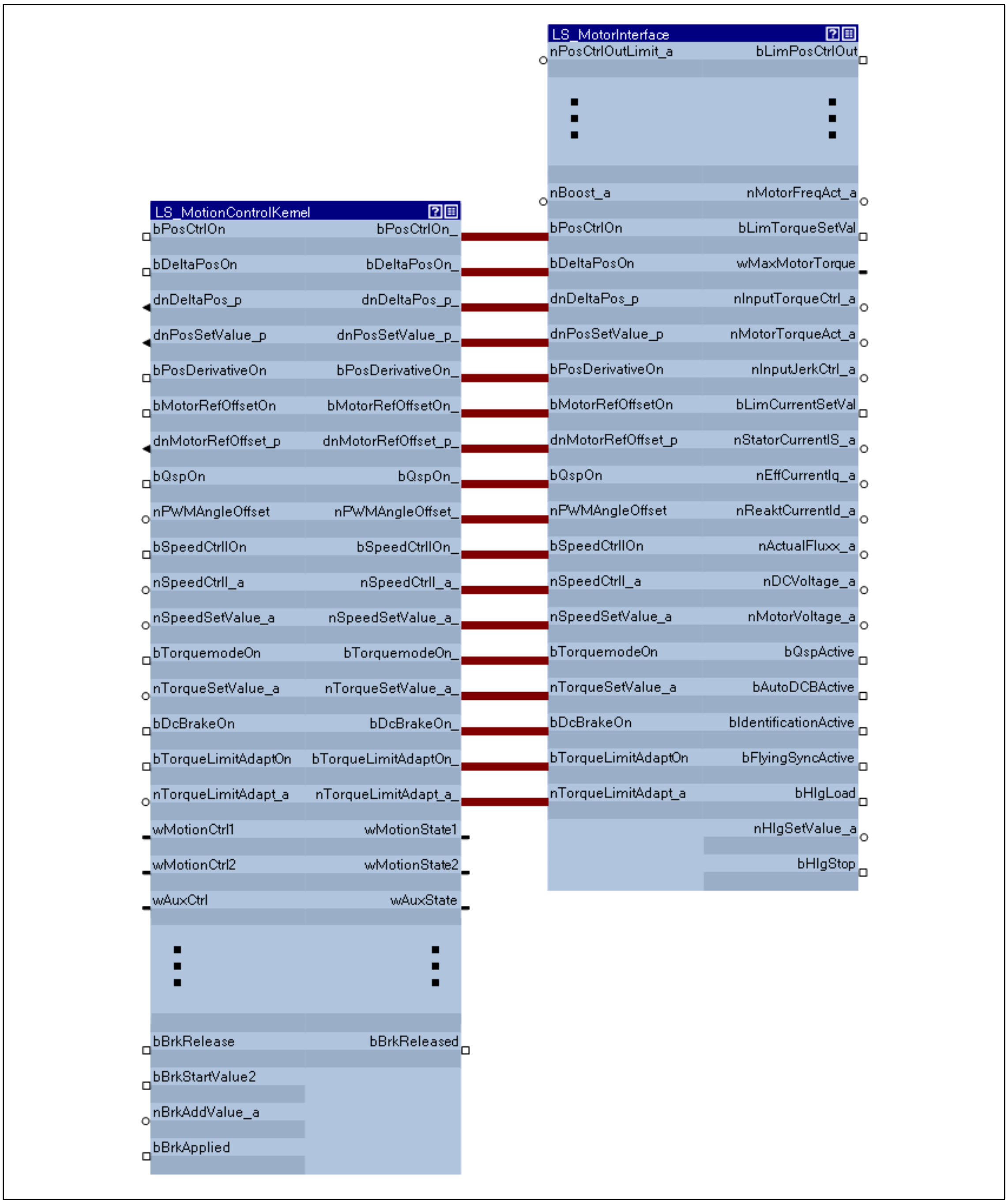
8 Basic drive functions (MCK)

In this chapter, the standard and basic drive functions integrated in the **Motion Control Kernel** (MCK) of the 8400 protec StateLine 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.

8.1 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:



[8-1] 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).

8.2 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

<table>
<tr><th>Identifier
Data type</th><th colspan="2">Information/possible settings</th></tr>
<tr><td colspan="3">Control and setpoint signals for motor control
The purpose of the following inputs is to transfer control signals and setpoints to the internal motor control function (LS_MotorInterface).</td></tr>
<tr><td>bPosCtrlOn</td><td colspan="2" rowspan="7">Inputs have no function on the 8400 protec StateLine!</td></tr>
<tr><td>bDeltaPosOn</td></tr>
<tr><td>dnDeltaPos_p</td></tr>
<tr><td>dnPosSetValue_p</td></tr>
<tr><td>bPosDerivativeOn</td></tr>
<tr><td>bMotorRefOffsetOn</td></tr>
<tr><td>dnMotorRefOffset_p</td></tr>
<tr><td rowspan="3">bQspOn
BOOL</td><td colspan="2">Trigger quick stop (QSP) via the MCK
• Also see device command "Activate/deactivate quick stop".</td></tr>
<tr><td>TRUE</td><td>Activate quick stop
• 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).
• A pulse inhibit is set if the auto-DCB function has been activated via C00019.
• The motor is kept at a standstill during closed-loop operation (function in preparation).</td></tr>
<tr><td>FALSE</td><td>Deactivate quick stop
• The quick stop is deactivated if no other source for the quick stop is active.
• C00159 displays a bit code of active sources/causes for the quick stop.</td></tr>
<tr><td>nPWMAngleOffset
INT</td><td colspan="2">Angular offset input
• Scaling: 16384 ≡ 100 %
• Setting range: 0 ... 199.99 %</td></tr>
<tr><td rowspan="2">bSpeedCtrlIOn
BOOL</td><td colspan="2">Directly set the I-component of speed controller
• In order to statically specify a minimum torque, e.g. when a load is being lifted.</td></tr>
<tr><td>TRUE</td><td>Se the I-component of the speed controller to the value nSpeedCtrlI_a.</td></tr>
<tr><td>nSpeedCtrlI_a
INT</td><td colspan="2">I-component of the speed controller
• Value is adopted in the case of a FALSE-TRUE edge at the input bSpeedCtrlIOn.</td></tr>
<tr><td>nSpeedSetValue_a
INT</td><td colspan="2">Rotation speed/velocity setpoint</td></tr>
<tr><td>bTorquemodeOn
BOOL</td><td>TRUE</td><td>Switch on torque-controlled operation</td></tr>
<tr><td>nTorqueSetValue_a
INT</td><td colspan="2">Torque setpoint</td></tr>
</table>

Identifier Data type		Information/possible settings
bDcBrakeOn BOOL		Manual DC-injection braking (DCB) • For this function, the *bDcBrakeOn_* output signal must be connected to the *bDcBrakeOn* input of the same name of the LS_MotorInterface system block. • Detailed information on DC-injection braking is provided in the motor control chapter, subchapter "DC-injection braking".
		Note! Holding braking is not possible when this braking mode is used! 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. • The braking effect stops when the rotor is at standstill. • After the hold time (C00107) has expired, the controller sets the pulse inhibit.
bTorqueLimitAdaptOn		Inputs have no function on the 8400 protec StateLine!
nTorqueLimitAdapt_a		
Control words		
MCK: wMotionCtrl1 wMotionCtrl2		Inputs have no function on the 8400 protec StateLine!
wAuxCtrl		
wSMCtrl WORD		Interface to the optional safety system. • 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**). • 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		Inputs have no function on the 8400 protec StateLine!
nSpeedAddValue_v		
nSpeedOverride_a		
nAccOverride_a		
nSRampOverride_a		
bLimitSwitchPos		
bLimitSwitchNeg		
bHomingMark		
bBrkRelease BOOL		Holding brake control: Releasing/applying the brake in connection with the selected operating mode
	FALSE	Apply brake. • During automatic operation, the internal brake logic controls the brake.
	TRUE	Release brake manually (forced release). • During automatic operation, the internal brake logic is deactivated and the brake is released. If the brake control has inhibited the controller, this inhibit is deactivated again.
bBrkStartValue2		Inputs have no function on the 8400 protec StateLine!
nBrkAddValue_a		
bBrkApplied		

Outputs

Identifier Data type	Value/meaning	
Control and setpoint signals for motor control The following outputs are used to transfer control signals and setpoints to the internal motor control function (LS_MotorInterface).		
bPosCtrlOn_	Outputs have no function on the 8400 protec StateLine!	
bDeltaPosOn_		
dnDeltaPos_p_		
dnPosSetValue_p_		
bPosDerivativeOn_		
bMotorRefOffsetOn_		
dnMotorRefOffset_p_		
bQspOn_ BOOL	TRUE	Activate quick stop
nPWMAngleOffset_a_ INT	Angular offset input	
bSpeedCtrlIOn_ BOOL	TRUE	Set I-component of speed controller.
nSpeedCtrlI_a_ INT	I-component of the speed controller	
nSpeedSetValue_a_ INT	Main setpoint of speed	
bTorqueModeOn_ BOOL	TRUE	Switch on torque-guided operation.
nTorqueSetValue_a_ INT	Torque setpoint	
bDcBrakeOn_ BOOL	TRUE	Activate DC-injection braking.
bTorqueLimitAdaptOn_	Outputs have no function on the 8400 StateLine!	
nTorqueLimitAdapt_a_		
Status words		
wMotionState1 wMotionState2 WORD	MCK status word 1 & 2 • For a detailed description of the individual status bits, see subchapter entitled "MCK status word.	
wAuxState	Output has no function on the 8400 protec StateLine!	
Status signal and actual-value signals from Motion Control Kernel functions		
nSpeedSet_v	Outputs have no function on the 8400 protec StateLine!	
dnPosTarget_p		
dnPosSet_p		
wActProfileNo		
wFollowProfileNo		
bPosBusy		
bPosDone		
bHomingDone		
bHomePosAvailable		

Identifier Data type	Value/meaning	
bBrkReleaseOut BOOL	Trigger signal for switching element holding brake control via a digital output • Use bit 0 in C02582 to activate inverted switching element triggering. ▶ Holding brake control	
	FALSE	Apply brake.
	TRUE	Release brake.
bBrkReleased BOOL	"Brake released" status signal considering the brake release time • When the holding brake is triggered to close, *bBrkReleased* is immediately reset to FALSE even if the brake closing time has not yet elapsed! ▶ Holding brake control	
	TRUE	Brake released (after the brake release time has expired).

8.2.1 MCK status word

MCK status word 1 (wMotionState1)

Bit	Designation	Description				
0 ... 3	ActOpModeBit00 ... ActOpModeBit03	**Active operating mode**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
		Speed follower	0	0	0	0
		Safe stop 1 (SS1)	0	1	0	0
		StandBy (internal operating mode in the event of quick stop, pulse inhibit, and DC-injection braking)	1	1	1	1
4	Busy	Has no function on the 8400 protec StateLine (always "0")!				
5	Done					
6	AcceleratingActive					
7	ConstSpeedDuty					
8	DeceleratingActive					
9	S_ShapingActive					
10	Pos. HW-Limit Detected					
11	Neg. HW-Limit Detected					
12	HomPosDone					
13	HomPosAvailable					
14	Reserved	-				
15	Reserved	-				

Note!

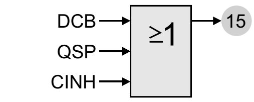
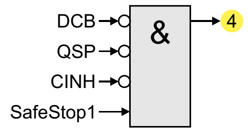
The internal "StandBy" operating mode is assumed if pulse inhibit, quick stop and/or DC-injection braking are activated.

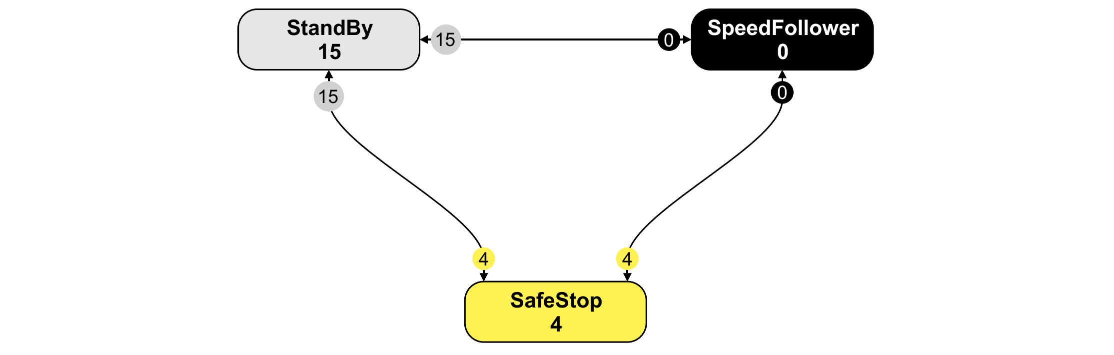
- No setpoint generation via the **Motion Control Kernel** in this operating mode.
- If the holding brake control sets a controller inhibit when the holding brake is closed, the internal "StandBy" operating mode is not assumed.

MCK status word 2 (wMotionState2)

Bit	Designation	Description
16	DwellTime	Has no function on the 8400 protec StateLine (always "0")!
17	InTarget	
18	PosDone	
19	Reserved	
20 ... 23	ActPosMode_Bit00 ... ActPosMode_Bit03	
24 ... 31	ActProfileNo_Bit00 ... ActProfileNo_Bit07	

8.2.2 MCK state machine

Prio	Condition		
1	15	Setpoint generation through Motor control (MCTRL): • DCB = DC-injection braking • QSP = quick stop • CINH = controller inhibit	
2	4	"Safe stop 1" (SS1) requested ▶ Interface to safety system	
3	0	Speed follower requested	



[8-2] MCK state machine

8.2.3 Interface to safety system

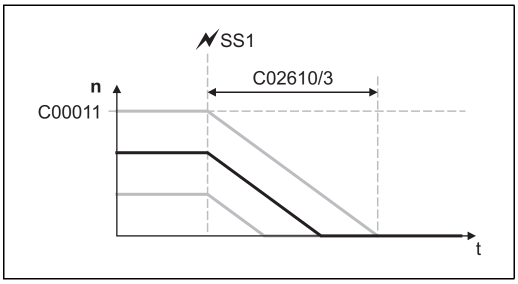
For operation with optional safety system, the LS_MotionControlKernel system block has the *wSMCtrl* input, This interface is used to transfer a control word by means of which the **Motion Control Kernel** can be supplied with information on requested or active safety functions. The **Motion Control 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 ... 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.



[8-3] 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.

8.3 Speed follower

The 8400 protec StateLine controller only supports the "Speed follower" operating mode, in which the drive follows a speed setpoint.

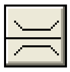
8.3.1 Parameter setting

Short overview of parameters for the "speed follower" operating mode:

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*:



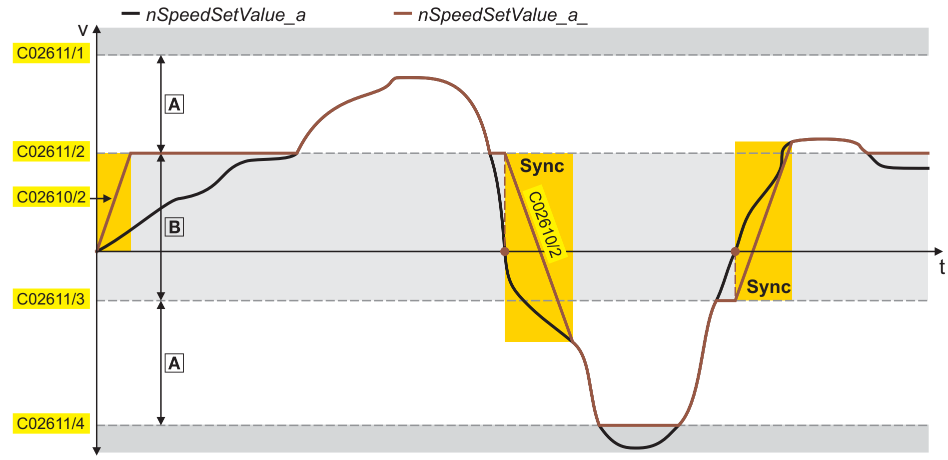
8.3.2 Setpoint selection

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.

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:



A Setpoint range
B Synchronisation range

[8-4] Example: Speed setpoint generation in the "Speed follower" operating mode (with nSpeedAddValue_v = 0)

8.4 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 (📖 311)
- 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.

8.4.1 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 Data type	Information/possible settings	
bBrkRelease BOOL	Releasing/applying the brake in connection with the selected operating mode	
	FALSE	Apply brake. • During automatic operation, the internal brake logic controls the brake.
	TRUE	Release brake manually (forced release). • During automatic operation, the internal brake logic is deactivated and the brake is released. If the brake control has inhibited the controller, this inhibit is deactivated again.
bBrkStartValue2	Inputs have no function on the 8400 protec StateLine!	
nBrkAddValue_a		
bBrkApplied		

Outputs

Identifier Data type	Value/meaning	
bBrkReleaseOut BOOL	Trigger signal for switching element holding brake control via a digital output • Use bit 0 under C02582 to activate inverted switching element triggering. ▸ Functional settings	
	FALSE	Apply brake.
	TRUE	Release brake.
bBrkReleased BOOL	"Brake released" status signal considering the brake release time • When the holding brake is triggered to close, *bBrkReleased* is immediately reset to FALSE even if the brake closing time has not yet elapsed!	
	TRUE	Brake released (after the brake release time has expired).

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

8.4.2 Parameter setting

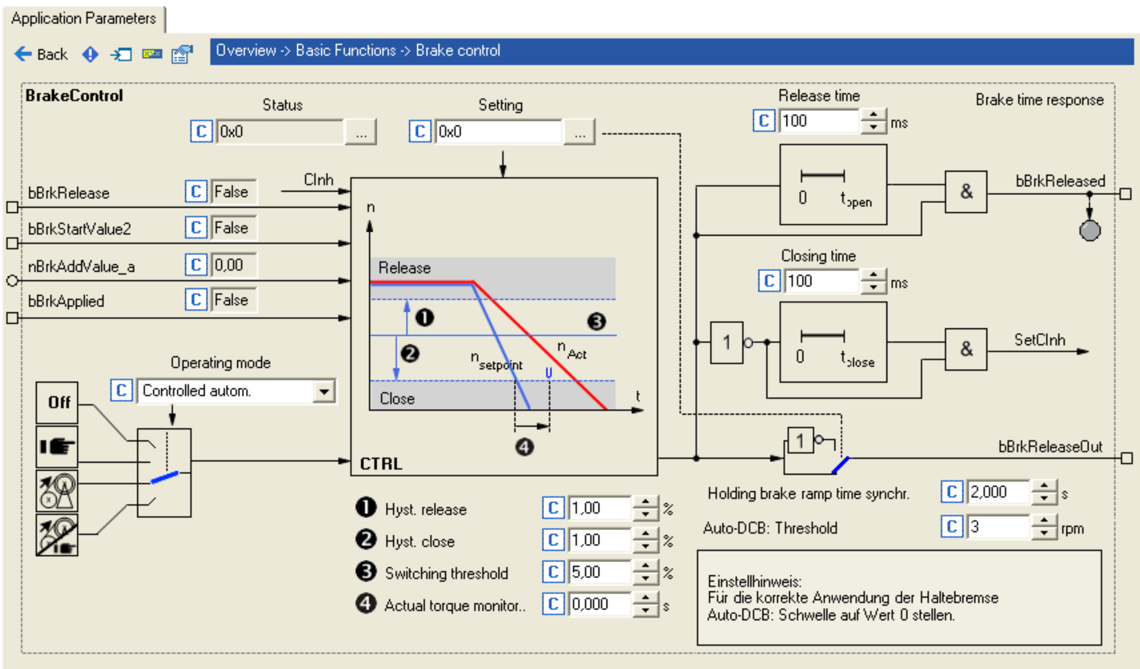
Danger!

A faultless 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 protec StateLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
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 holding brake control:

Parameter	Info	Lenze setting	
		Value	Unit
C02580	Holding brake: Operating mode	12: Autom. controlled	
C02581/1	Holding brake: Switching threshold	5.00	%
C02581/2	Holding brake: Hyst. release	1.00	%
C02581/3	Holding brake: Hyst. close	1.00	%
C02582	Holding brake: Setting	0	
C02589/1	Holding brake: Closing time	100	ms
C02589/2	Holding brake: Release time	100	ms
C02589/3	In preparation - without function at the moment! Holding brake: Waiting time status	100	ms
C02593/1	Holding brake: Actual value monitoring	0.000	ms
C02610/1	MCK: Holding brake ramp time synchr.	2.000	s
C02607	Holding brake: Status	-	
C00830/68	MCK: nBrkAddValue_a	-	%
C00833/80	MCK: bBrkRelease	-	
C00833/81	MCK: bBrkStartValue2	-	
C00833/82	MCK: bBrkApplied	-	

Highlighted in grey = display parameter

8.4.2.1 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 *bBrkReleaseOut* trigger signal for the holding brake control switching element is set to FALSE.
- The *bBrkReleased* status signal is set to FALSE.

Mode 11: Manual control

In this mode, brake release and brake application can be directly controlled via the *bBrkReleaseBrake* input without special logic or automatic.

- Setting pulse inhibit or controller inhibit has no influence on the *bBrkReleaseOut* trigger signal 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.

- 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.
- 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 expiration of the brake application time, the controller is inhibited automatically by the basic "Holding brake control" function.

Tip!

The 2/12 mode is the usual mode to control the brake.

In this mode, the *bBrkReleaseBrake* input should be permanently set to FALSE unless manual release is required.

When *bBrkReleaseBrake* = TRUE, the brake is permanently released and the automatic control cannot apply the brake.

Related topics:

- TroubleQSP (📖 87)
- Behaviour in case of pulse inhibit (📖 320)

8.4.2.2 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 • "1" ≡ Inverted logic of the trigger signal for the holding brake control switching element
Bit 1	[--]: nAct < nMin at CInh	Brake response in case of pulse inhibit • "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. **Note:** • 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. • 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	[||]: Feedforward control inverted	Direction of feedforward control with vertical/hoist technology: • "0" ≡ Positive direction • "1" ≡ Negative direction **Note:** Reversal (Ccw) is then considered.
Bit 3	[--]: Horizontal	Direction of movement of the axis • "0" ≡ The axis performs vertical movements. Gravitational acceleration causes movements. • "1" ≡ The direction of the axis is horizontal or rotary. The gravitational acceleration does not cause any movement.
Bit 4	Reserved	
Bit 5	Reserved	
Bit 6	Sync ramp L_NSet_1 (from version 11.00.00)	Selection of the ramp time for the synchronisation process to setpoint speed after the brake opening time has elapsed **Revised behaviour from version 11.00.00:** • "1" ≡ The ramp time of the effective acceleration of the ramp function generator (L_NSet_1) is used (Lenze setting). • "0" ≡ As before, the ramp time set in C02610/1 is used. **Note:** 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 (📖 320)
▸ Feedforward control of the motor before release (📖 321)

8.4.2.3 Switching thresholds

Stop!

Do not set the lower speed threshold for closing the brake too high to prevent excessive wear of the brake!

Note!

When comparing speeds, only the absolute value of the motor speed and not the direction of rotation is considered.

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:

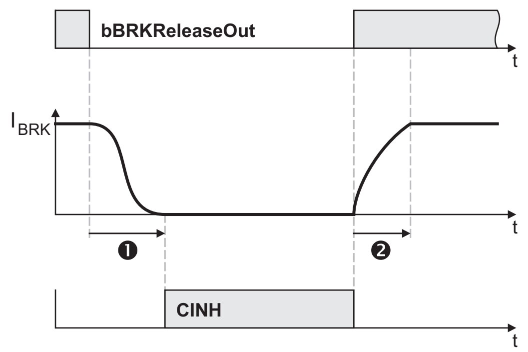
- Process when brake is released (317)
- Process when brake is closed (318)

8.4.2.4 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.

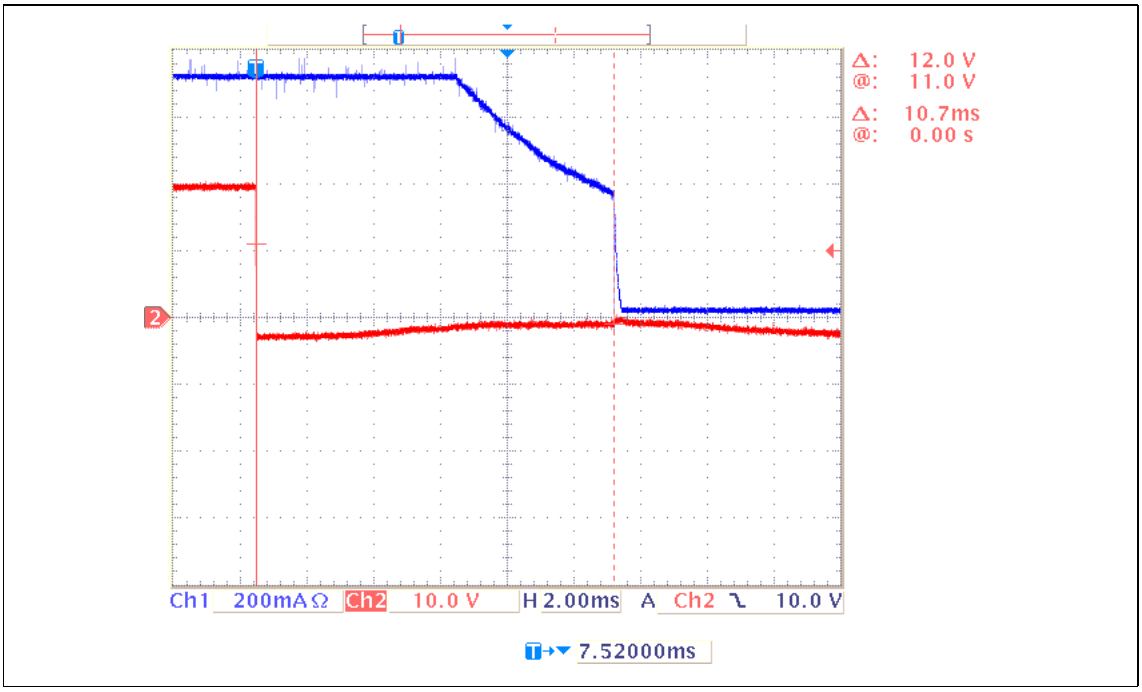
[8-5] Definition of the application and release time with the example of the PM brake

Tip!

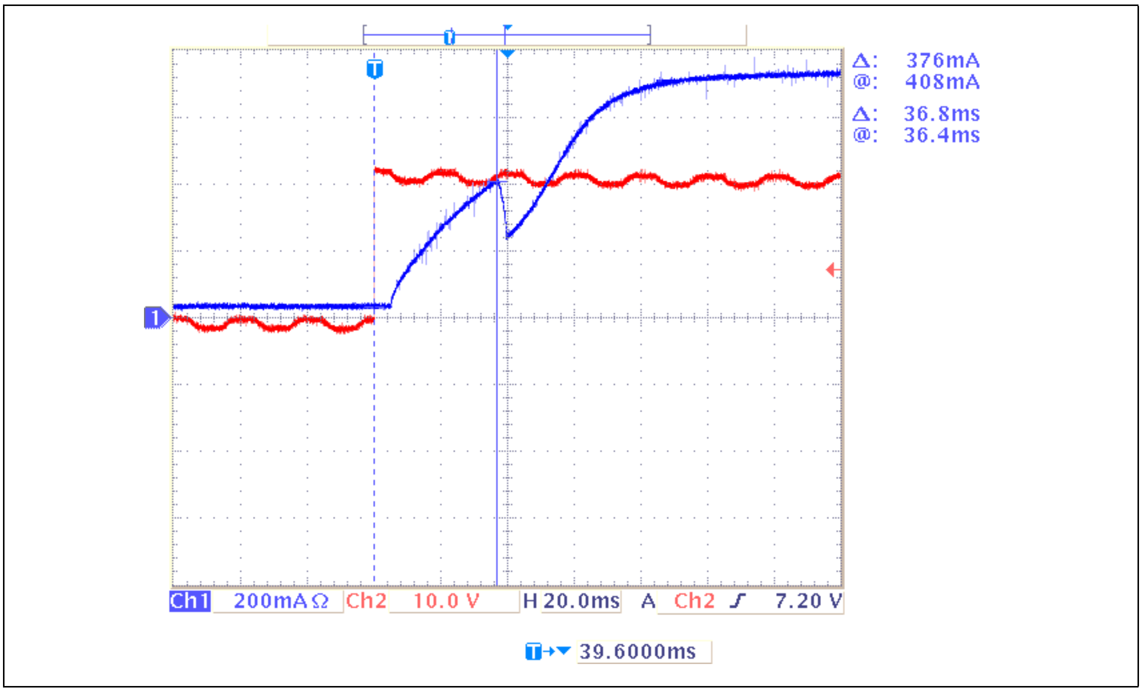
The application and release times do not only vary between the brake types but also depend on the basic conditions in the plant:

- 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.



[8-6] Oscillogram 1: Current characteristic for the application of a mechanical holding brake (application time: 10.7 ms)



[8-7] Oscillogram 2: Current characteristic for the release of a mechanical holding brake (release time: 36.8 ms)

Related topics:

▶ Process when brake is released (📖 317)
▶ Process when brake is closed (📖 318)

8.4.2.5 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.

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).

This example is based on the V/f characteristic control mode (VFCplus). The ramp process to the "elapsed" setpoint, however, is valid for all control modes of the motor control as there is always a mechanical and/or electrical delay when the holding brake is activated.

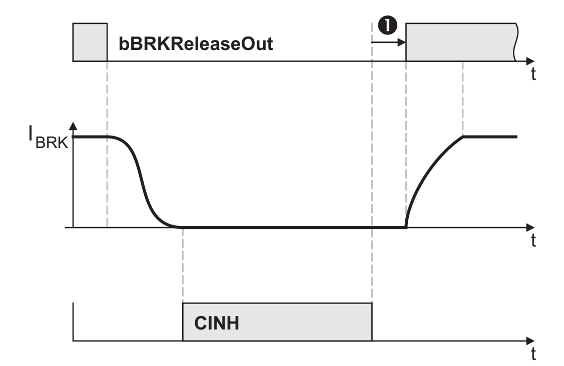
This delay is due to:

- 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 (📖 317)

8.4.2.6 Motor magnetising time (only with asynchronous motor)



❶ Motor magnetising time
CINH = controller inhibit

[8-8] 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.

8.4.2.7 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 [8-11] 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!

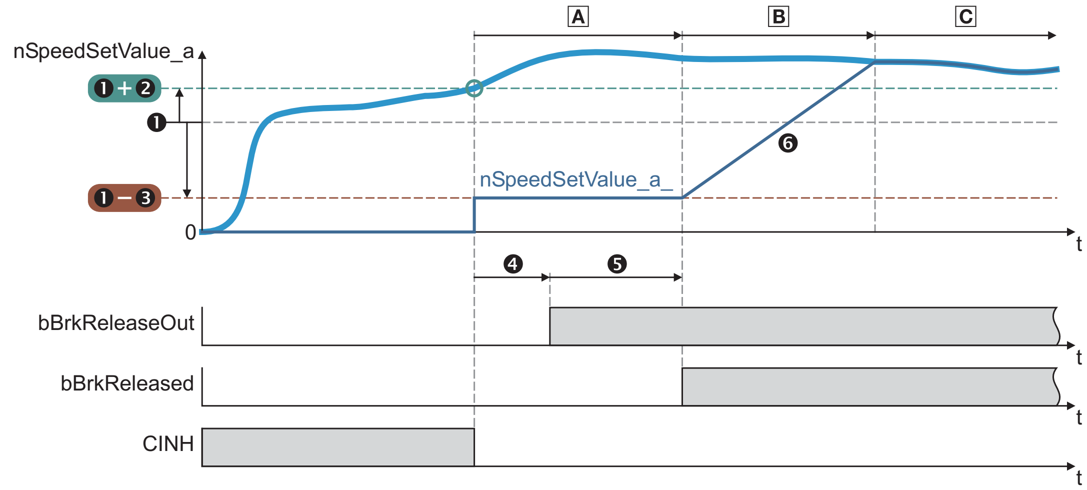
In the Lenze setting, the actual speed time monitoring is deactivated (C02593/1 = "0 s"), i.e. the brake will only be applied when the actual speed has reached the lower switching threshold.

8.4.3 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.
   - The drive synchronises to the already accelerated speed setpoint.

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

[8-9] Release holding brake in automatic mode via speed threshold

Related topics:

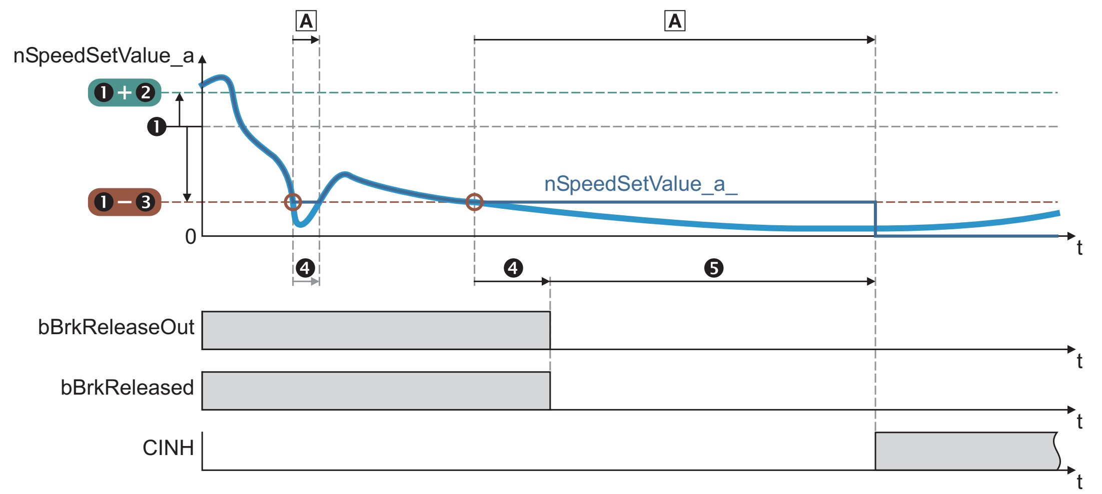
▶ Feedforward control of the motor before release (📖 321)

8.4.4 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.
   - The brake application time starts to expire.
3. After the brake application time has expired, the controller is inhibited.

Time diagrams

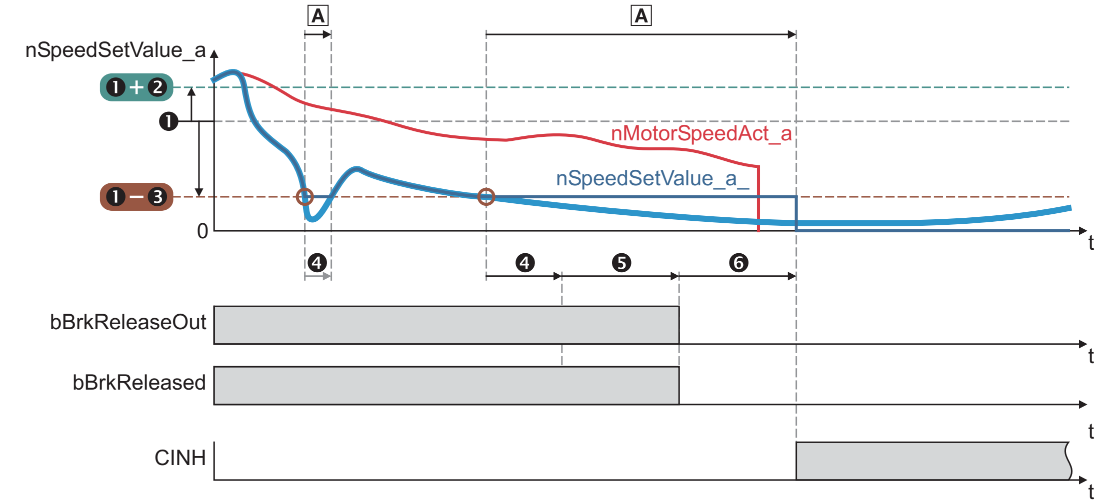
A Feedforward control



❶ C02581/1: Switching threshold
❷ C02581/2: Hysteresis for release
❸ C02581/3: Hysteresis for application
❹ Application delay
❺ C02589/1: Application time

[8-10] 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 Application delay
❺ C02593/1: Actual value monitoring time
❻ C02589/1: Application time

[8-11] Close holding brake in automatic mode with actual value monitoring time (C02593/1 > 0 s)

8.4.5 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

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!

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.
- The brake will not be applied until the motor speed has reached the threshold for brake activation. Hence, 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 (📖 311)
▸ Switching thresholds (📖 312)

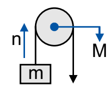
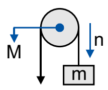
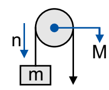
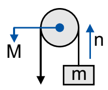
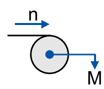
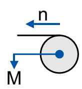
8.4.6 Feedforward control of the motor before release

The motor is precontrolled by selecting the lower speed threshold for applying the brake. When the upper speed threshold for brake release is reached, the motor is precontrolled for 200 ms with the lower threshold before the brake switches to the release mode.

Here, the direction of the feedforward control depends on two conditions:

1. On the settings selected under C02582:
   - Bit 2 = inverted feedforward control
   - Bit 3 = direction 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				−	−

Related topics:

▸ Functional settings (📖 311)
▸ Switching thresholds (📖 312)

9 Diagnostics & error management

This chapter provides information on error handling, drive diagnostics, and fault analysis.

9.1 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) (📖 347)

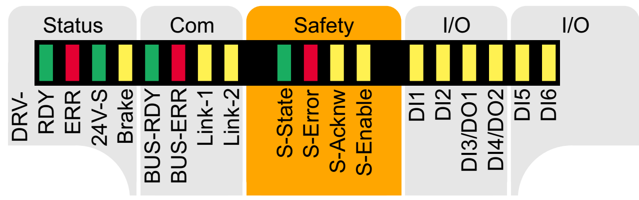
In addition to the control of the device status by the error handler, a logbook function records the errors and their histories. ▶ Logbook (📖 335)

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 (📖 342)

9.2 LED status displays

Information on some operating statuses can quickly be obtained via LED displays:



[9-1] LED display on the controller front panel

Labelling	Colour	Status	Description	
DRV-RDY	green	On	Standard device ready for operation	▶ LED status displays of the device status (📖 325)
DRV-ERR	red	On	Warning/trouble/fault	
24V-S	green	On	24-V voltage available	
Brake	yellow	Off	Motor holding brake not triggered (applied)	
		On	Motor holding brake triggered (released)	
LED status displays for communication				
BUS-RDY	green	Off	Communication not initialised	
		Blinking	Communication has been established	
		On	Communication has stopped	
BUS-ERR	red	Blinking	With PROFINET: Node detection	
		Fast blinking	With PROFINET: Nodes not detected	
		On	Communication error	
Link-1 Link-2	yellow	Off	Communication is not active	
		Blinking	Initialisation	
		Fast blinking	With PROFINET: Communication is active, telegrams are sent.	
LED status displays for integrated safety **Note**: The status of the safety option 10 is only shown via the "S_Enable" display.				
S-State	green	Off	Communication between Standard device and safety engineering is not possible	
		Blinking	Integrated safety is in service state	
		On	Communication between standard device and safety system has been established	
S-Error	red	Off	Error-free operation	
		Blinking	Integrated safety system is not supported by the standard device	
		On	Warning/trouble/fault	
S-Acknw	yellow	On	Acknowledgement of a parameter set acceptance is required	
S-Enable	yellow	Blinking	Safety function system active (unsafe display)	
		On	Controller enabled	

Labelling	Colour	Status	Description
LED status displays for I/O terminals			
DI1	yellow	On	DI1 = HIGH
DI2	yellow	On	DI2 = HIGH
DI3/DO1	yellow	On	DI3/DO1 = HIGH
DI4/DO2	yellow	On	DI4/DO2 = HIGH
DI5	yellow	On	DI5 = HIGH
DI6	yellow	On	DI6 = HIGH

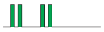
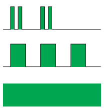
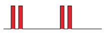
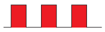
Related topics:

- Device control (DCTRL) (📖 67)
- Device state machine and device states (📖 79)
- CANopen option (📖 371)
- PROFIBUS/PROFINET option (📖 442)
- Drive-based safety (📖 450)
- I/O terminals (📖 197)

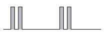
9.2.1 LED status displays of the device status

The control of the two LEDs "DRV-RDY" and "DRV-ERR" on the front of the controller depends on the device status.

The meaning can be seen from the table below:

DRV-RDY	DRV-ERR	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 active	Trouble
OFF		Fault active	Fault
OFF		System fault active	SystemFault

Legend
Meaning of the symbols used to describe the LED states:

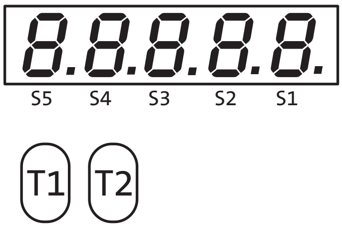
	LED is flashing once approx. every 3 seconds (*slow flash*)
	LED is flashing once approx. every 1.25 seconds (*flash*)
	LED is flashing twice approx. every 1.25 seconds (*double flash*)
	LED is blinking every second
	LED is permanently on

Related topics:

▸ Device state machine and device states (📖 79)

9.3 Drive diagnostics via the integrated 7-segment display

The controller comes with an integrated 7-segment display which can show 4 or 5 characters.



Key	Function
T1	Operator button for status display of subordinated devices (in preparation)
T2	Operator button for browsing the status values S1 ... S5 or if several messages are present at the same time.

[9-2] 7-segment display, 5 characters (S1 .... S5)

The display has three different modes:

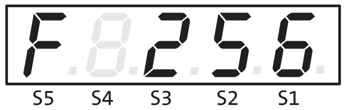
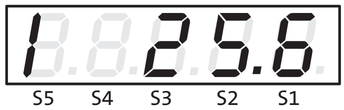
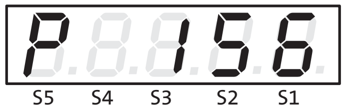
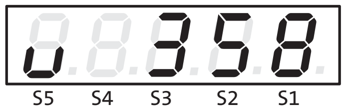
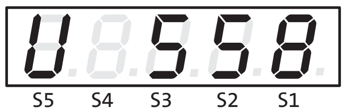
1. Automatic display
   - When the controller is switched on, the parameter saved as status value 1 is displayed.
2. Manual operation display
   - For triggering the controller in manual operation.
3. Message display
   - For errors or warnings.
   - This display has the highest priority and overrides the two other modes.

Detailed information on the different modes can be found in the following sections.

Automatic display

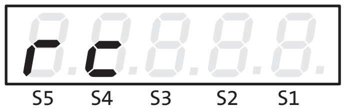
In this mode, five parameters can be displayed in a row.

- The selection key T1 serves to browse the parameters in ascending order. After the status value S5, it starts anew with status value S1.
- The parameters to be displayed as status values S1 ... S5 in the display can be configured using the »Engineer« (in preparation).
- In the Lenze setting, the following status values are displayed:

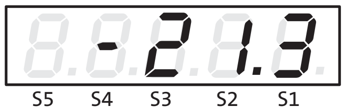
Status value	Example
S1: Motor output frequency in [Hz] • The output frequency is displayed as a function of the direction of rotation, i.e. in case of CCW rotation, a minus sign is displayed. • Display area: ±999 Hz	 "F 256" = output frequency 256 Hz
S2: Actual current value in [0.1 A]	 "I 25.6" = actual current value 25.6 A
S3: Device utilisation in [%]	 "P 156" = Device utilisation 156 %
S4: Motor voltage in [V]	 "u 358" = Motor voltage 358 V
S5: DC-bus voltage in [V]	 "U 558" = DC-bus voltage 558 V

Manual operation display (in preparation)

If the controller is changed over to the manual operation mode using the operator button, "rc" is displayed (remote control):



- When it is changed over between CCW and CW rotation using the operator button, the output frequency of the motor in CW rotation is displayed without sign and in CCW rotation with minus sign.
  - Example: "-21.3" = motor rotates in CCW rotation with an output frequency of 21.3 Hz.



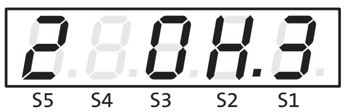
Message display

If warnings or errors exist, these are shown in blinking mode.

- If several errors exist, the active status determining error is displayed and all further errors will only be entered into the logbook.

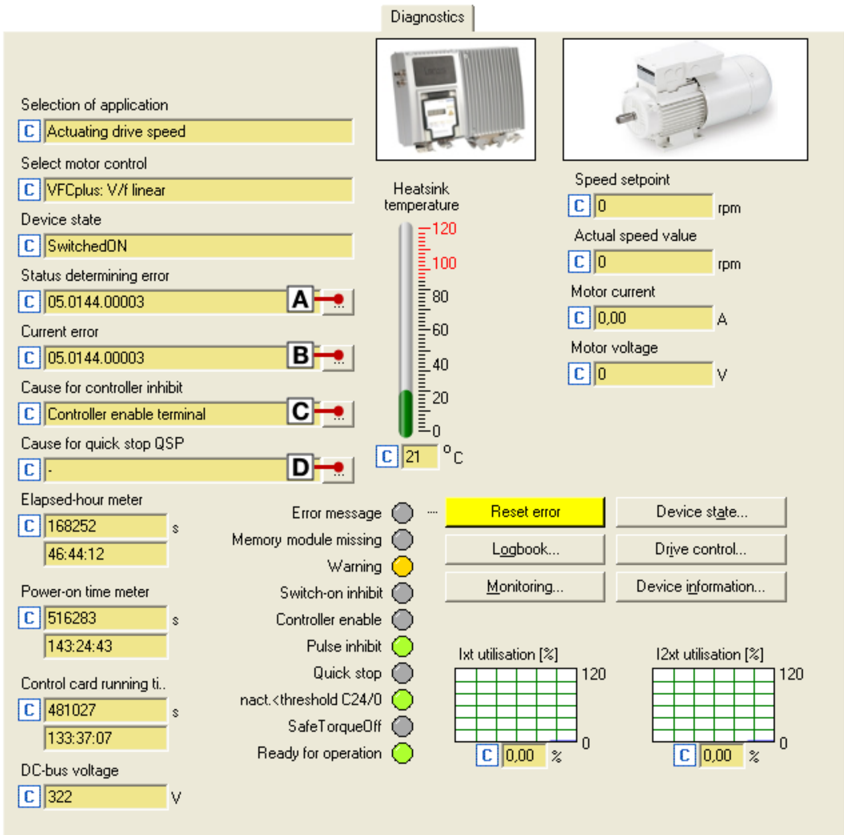
The messages are structured as follows:

- The character in position S5 indicates the device number if several subordinated device are connected to the controller via the fieldbus interface, for example:
  - "1" = 8400 protec controller (master)
  - "2" = 8400 protec controller (slave 1 to node address 2)
  - "3" = 8400 protec controller (slave 2 to node address 3), etc.
- The characters in positions S1 to S4 indicate the abbreviated warning or error. ▶ Short overview (A-Z) (🕮 353)
- Example: "2 OH3" = In case of the device on node address 2, the warning "OH3: Motor temperature (X21) triggered" is active.



9.4 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.
Reset error		Acknowledge error message (if the error cause has been eliminated).
Logbook...		Display the Logbook of the controller. (📖 335)
Monitoring...		Configure the Monitoring. (📖 340)
Device status...		Display the internal state machine including the current device status.
Drive control...		Display the bit assignment of the following control-related words: • MCI control word (C00136/1) • CAN control word (C00136/2) • Cause of controller inhibit (C00158) • Cause of quick stop (C00159) • Status word (C00150) • 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. In the *Project view*, select the 8400 protec StateLine controller.
2. Click the icon or select the **Online→Go online** command to build up an online connection with the controller.
3. Select the **Diagnostics** tab.
   - With an online connection, the **Diagnostics** tab displays current status information about the controller.

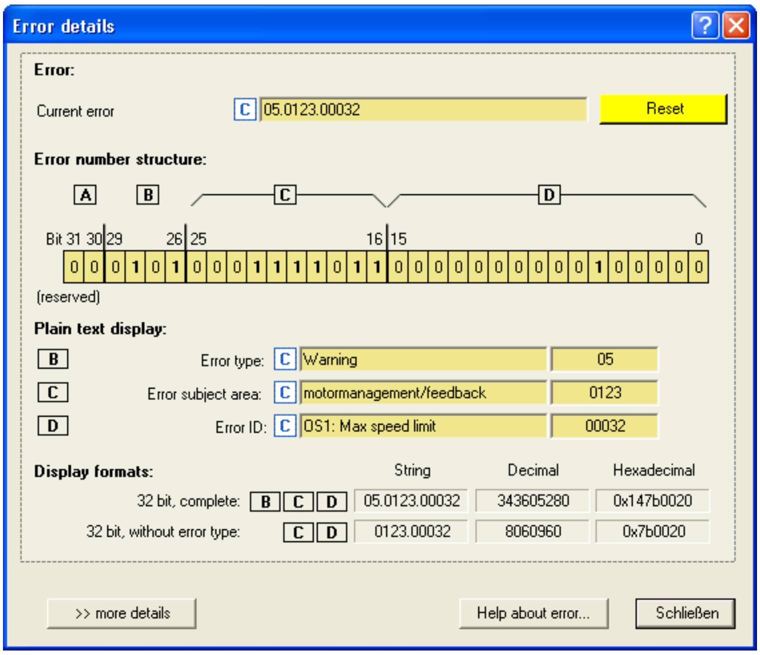
Tip!

The online connection to the controller can be established via the following device interfaces:

- Diagnostic interface X70
- CANopen option (371)
- PROFIBUS/PROFINET option (442)

9.4.1 Display details of the error

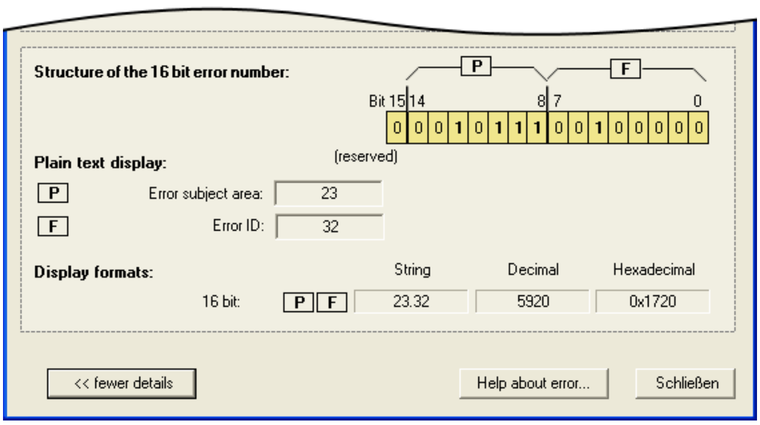
If you go to the **Diagnostics** tab and click the ... button for the status determining or current error, the *Error details* dialog box displays further information on the error:



- Click the **Help about error...** button to open the online help with information on the error cause and possible remedies.

From version 06.00.00 / »Engineer V2.13« onwards:

- The **>> more details** button serves to provide more information about the structure of the 16-bit error number:

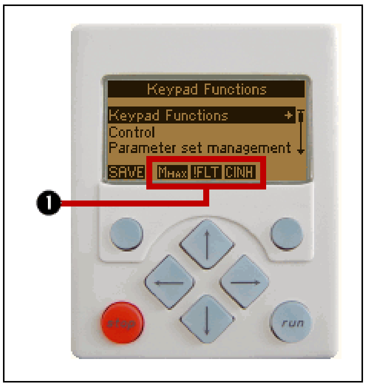


Related topics:

▸ Structure of the 32-bit error number (bit coding) (📖 347)
▸ Structure of the 16 bit error number (bit coding) (📖 350)

9.5 Drive diagnostics via keypad/bus system

Keypad display of the controller status



- If the diagnosis terminal is connected to the diagnostic interface X70, the status of the controller is shown via different icons on the LCD display in the area ❶.

Icon	Meaning	Note
RDY	Controller is switched on.	▶ SwitchedOn (🕮 85)
RUN	Controller is enabled.	
STP	Application in the controller is stopped.	
QSP	Quick stop active	
CINH	Controller is inhibited.	The power outputs are inhibited.
OFF	Controller is ready to start.	▶ ReadyToSwitchOn (🕮 84)
Mmax	Speed controller 1 at the limit.	The drive is torque-controlled.
Imax	Set current limit has been exceeded in motor or generator mode.	
IMP	Pulse inhibit active	The power outputs are inhibited.
!SFLT	System fault active	
!FLT	Fault	▶ Fault (🕮 89)
!TRB	Trouble	▶ Trouble (🕮 88)
!TQSP	TroubleQSP	▶ TroubleQSP (🕮 87)
WRN	Warning 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" (465).

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
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

Parameter	Display
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, which in the »Engineer« parameter list and in the keypad are classified in the category **Identification → Controller**, 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

9.6 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 (📖 347)
- Error messages generated by the application (via LS_SetError)
- Switching the controller on/off

Information saved

For each event, the following information is saved in the logbook:

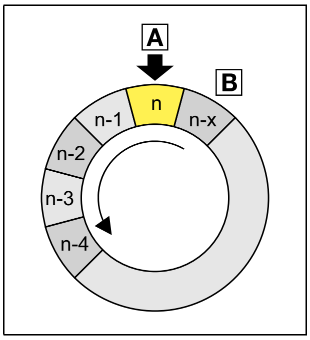
- Type of response to the event (e.g. fault, warning or information)
- Subject area that activated the event (e.g. CAN or USER).
- Event
- Value of power-on time meter
- Selected process values (analog % signals, binary signals)

Memory depth

Maximum number of logbook entries: 14 *(extension is being prepared)*

9.6.1 Functional description

The structure of the logbook corresponds to a ring buffer:



- As long as free logbook memory is available, the entry is placed in the next free position within the memory (A).
- If all memory units are assigned, the oldest entry (B) is deleted for 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.

9.6.2 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.

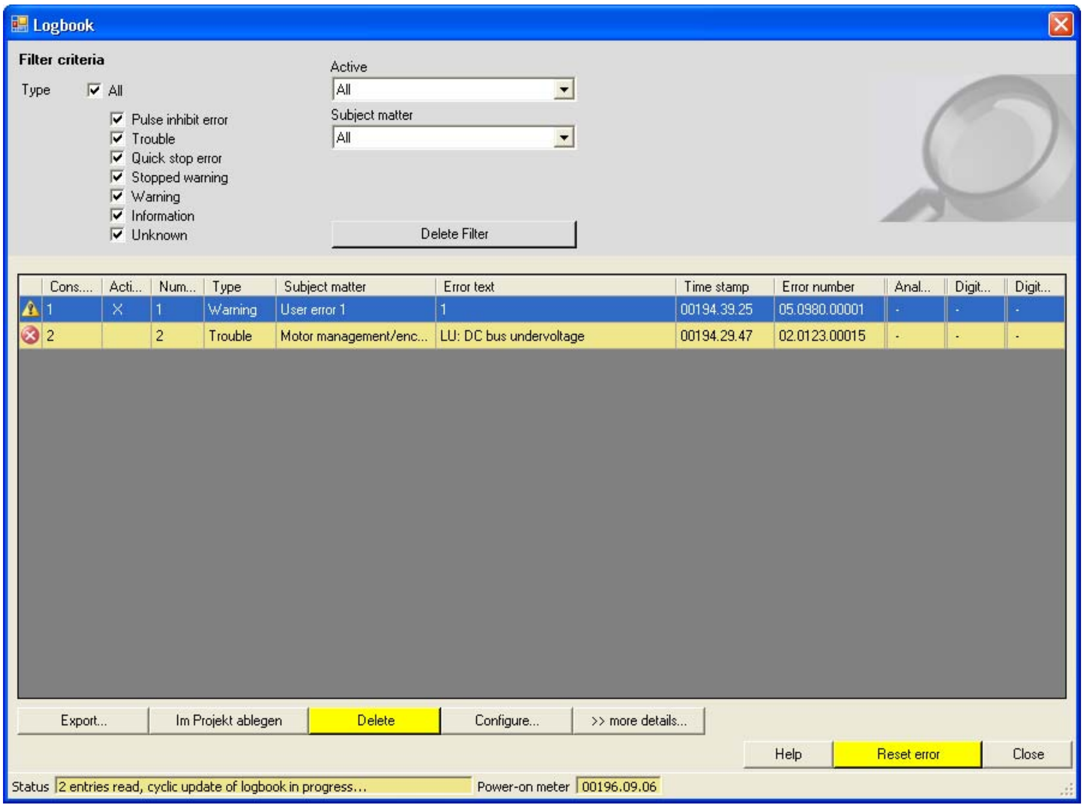
9.6.3 Reading out logbook entries

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. In the *Project view*, select the 8400 protec StateLine controller.
2. Click the icon or select the **Online→Go online** command to build up an online connection with 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. ▸ Exporting logbook entries to a file (📖 338)
File in project	File the current logbook in the Engineer project to be able to access it offline, too. ▸ Storing the logbook in the project (📖 339)
Delete	Delete all entries available in the logbook.
Configure...	Open parameterisation dialog for configuring the logbook.

Button	Function
>> more details	Show more details: • Analog value 1, digital values 1 & 2 • More output types of the error numbers (32-bit, internal 32-bit and internal 16-bit). • 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.
Reset error	Acknowledge 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.
Close	Close the *Logbook* dialog box again.

9.6.4 Exporting logbook entries to a file

How to export the logbook entries to a file:

1. Click **Export...** in the *Logbook* dialog box.
   - 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 into the given 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. Counter
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

9.6.5 Storing the logbook in the 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 store the current logbook in the project.

How to store 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 stored in the Engineer project independent of the set filter criteria.
- A logbook of the same device already stored before will be overwritten without querying the user.
- The filter settings are not stored 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.

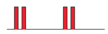
Note!

Storing 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 stored in the project remain stored.

9.7 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 "DRV- ERR" LED on the front of the controller goes on:

Response	Entry in the logbook	Display in C00168	Pulse inhibit	Controller inhibit	Acknowledgement required	"DRV-ERR" LED
None						OFF
Fault	☑	☑	☑	☑	☑	
Trouble	☑	☑	☑	☑		
TroubleQSP	☑	☑			☑	
WarningLocked	☑	☑			☑	
Warning	☑	☑				
Information	☑	☑				OFF
System fault	☑	☑	☑	☑	Mains switching is required!	

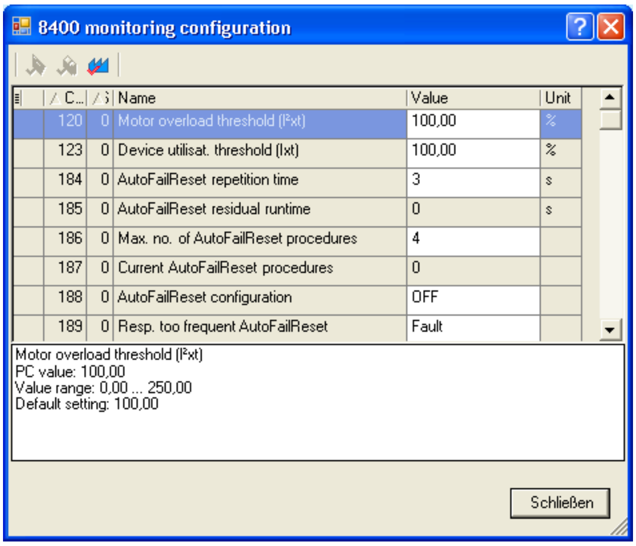
Related topics:

- LED status displays of the device status (📖 325)
- Device state machine and device states (📖 79)
- Device overload monitoring (Ixt) (📖 180)
- Motor load monitoring (I2xt) (📖 181)
- Motor temperature monitoring (PTC) (📖 183)
- Brake resistor monitoring (I2xt) (📖 184)
- Motor phase failure monitoring (📖 186)
- Mains phase failure monitoring (📖 187)
- Maximum current monitoring (📖 187)
- Maximum torque monitoring (📖 188)
- Encoder open-circuit monitoring (📖 188)

9.7.1 Monitoring configuration

How to configure the monitoring functions using the »Engineer«:

1. In the *Project view*, select the 8400 protec StateLine controller.
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:



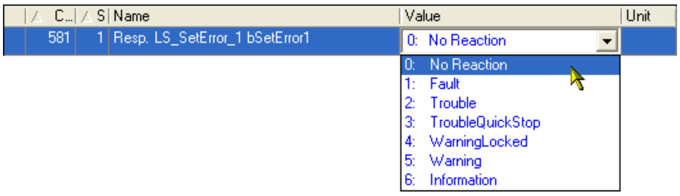
Related topics:

▶ Setting the error response (📖 342)

9.7.2 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 the chapter "Short overview (A-Z)" contains the error messages for which the response can be set. (📖 353)

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 (I^2xt)
C00123	Device utilisat. threshold (Ixt)
C00572	Brake resistor overload threshold
C00599	Motor phase failure threshold

9.7.3 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 (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 permissible unsuccessful error reset processes • When the number set is reached, the response parameterised in C00189 is executed.
C00187	Current number of unsuccessful error reset processes carried out
C00188	Configuration of the AutoFailReset function • 0: Off • 1: Fault + TroubleQSP • 2: WarningLocked • 3: All locking
C00189	Response after max. number of unsuccessful error reset processes has been reached
Highlighted in grey = display parameter	

9.8 Maloperation of the drive

Maloperation	Cause	Remedy
Motor does not rotate	DC-bus voltage is too low • Red LED is blinking every 1 s • Display in the keypad: *LU*	Check mains voltage
	Controller is inhibited • Green LED is blinking • Display in the keypad: CINH	Deactivate controller inhibit • Note: Controller inhibit can be set via several sources ! • C00158 displays all active sources for controller inhibit.
	Automatic start is inhibited (Bit 0 in C00142 = 1)	LOW/HIGH edge at RFR 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 • Display in the keypad: IMP	Deactivate quick stop • Note: Quick stop can be set via several sources! • C00159 displays all active sources for quick stop.
	Setpoint = 0	Select setpoint
	JOG frequency = 0 at activated JOG setpoint	Set JOG setpoint in C00039/1...15
	Trouble 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 C00023: Imax in generator mode
	Motor is underexcited or overexcited	Check parameterisation: C00006: Motor control C00015: VFC: V/f base frequency 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 - or - Adjust motor parameters manually: C00084: Motor stator resistance C00087: Rated motor speed C00088: Rated motor current C00089: Rated motor frequency C00090: Rated motor voltage C00091: Motor cosine phi 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 - or - Adjust motor parameters manually: C00084: Motor stator resistance C00087: Rated motor speed C00088: Rated motor current C00089: Rated motor frequency C00090: Rated motor voltage C00091: Motor cosine phi 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 or 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 too low	Increase mains voltage

An asynchronous motor with feedback rotates without control and with too low speed	Motor phases have been interchanged
	• Thus the rotating field of the motor is not identical anymore with the rotating field of the feedback system.
	• Therefore, the drive shows the following behaviour if V/f characteristic control (C00006 = 7) is performed:
	  • The motor rotates faster than the speed setpoint by the value set in C00074.
	  • After the controller has been enabled, the controller will not stop if the speed setpoint = 0 or a quick stop (QSP) occurs.
	  • 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
		If possible: Actuate 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)

9.9 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

9.10 Error messages of the operating system

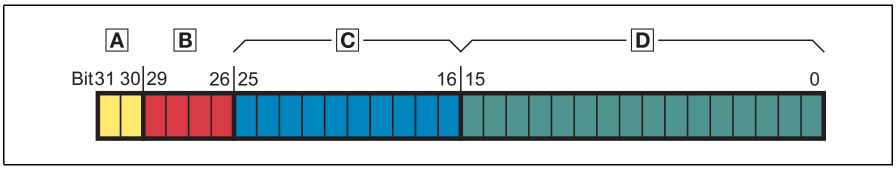
This chapter describes all error messages of the controller operating system and possible causes & remedies.

Tip!

Each error message is also saved to the logbook in chronological order. ▶ Logbook (📖 335)

9.10.1 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:



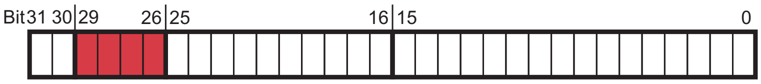
A Reserved
B Error type
C Error subject area
D Error ID

[9-3] Structure of the error number

- Display parameter: C00168
- The LS_DriveInterface system block shows the 32-bit error number at the outputs *wStateDetermFailNoLow* (Low Word) and *wStateDetermFailNoHigh* (High Word).
  - From version 06.00.00 onwards: If the "Use 16BitFailNo." (Bit 15 = "1") option is activated in C00148, the short 16-bit error number is provided by the SB LS_DriveInterface at the *wStateDetermFailNoLow* output and the value "0" is provided at the *wStateDetermFailNoHigh* output (see the following chapter).
- 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**]

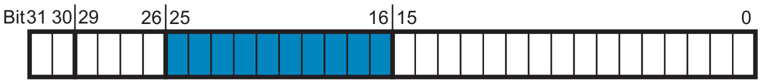
Error type



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

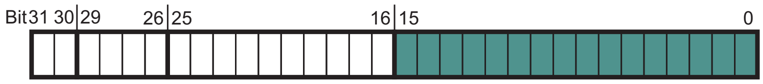
Error subject area



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
303	Safety option	Errors that occur within the integrated safety system. See the documentation for integrated safety.	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 ... 983	User error 1 ... User error 4	Errors generated by the user (by the application) via the LS_SetError_1 system block.	Yes

Error ID

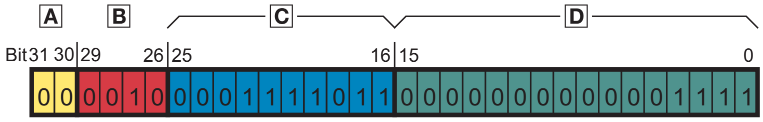


16-bit value (0 ... 65535) for error identification within the error subject area.

Example for bit coding of the error number

C00168 displays the internal error number "142278671".

- This decimal value corresponds to the following bit sequence:



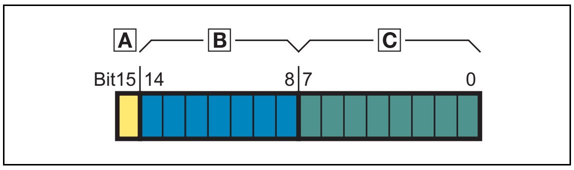
Assignment	Information	Meaning in the example
0 0	Reserved	-
0 0 1 0	Error type	2: Trouble
0 0 0 1 1 1 1 0 1 1	Error subject area	123: Motor management / encoder
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	Error ID	15: "LU: DC bus undervoltage"

- 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.

9.10.2 Structure of the 16 bit error number (bit coding)

This function extension is available from version 06.00.00!

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:

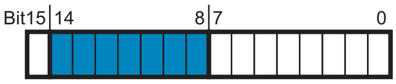


A Reserved
B Error subject area
C Error ID

[9-4] Structure of the error number

- Display parameter: C00160
- The LS_DriveInterface system block shows the 16-bit error number at the output *wStateDetermFailNoShort*.
- If the "Use 16BitFailNo." option is activated in C00148 (bit 15 = "1"), the LS_DriveInterface system block also shows the short 16-bit error number at the output *wStateDetermFailNoLow* (Low Word of the 32-bit error number).
  - In this case, the *wStateDetermFailNoHigh* output (high word of the 32-bit error number) is "0".
  - Advantage: The bus transfer of the error numbers is possible via a data word without changing the interconnection of the technology application.
- 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.

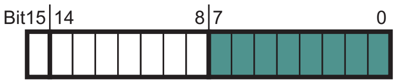
Note!

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

Error subject area		Assigned errors	Remedy possible by user?
No.	Name		
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
98	Safety option	Errors that occur within the integrated safety system. See the documentation for integrated safety.	Yes
100 ... 103	User error 1 ... User error 4	Errors generated by the user (by the application) via the LS_SetError_1 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. (353)

9.10.3 Reset error message

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 device command C00002/19 = "1".

Tip!

When an online connection to the controller has been established, use the **Diagnostics** tab of »Engineer« and click **Reset error** to reset a pending error message.

9.10.4 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             No error                   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 protec StateLine 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.

9.10.5 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 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". (📖 355)

Error number			Error message	Response (Lenze setting)	can be set in	CAN Emergency Error Code
32 bits	16 bits$_{hex}$	16 bits$_{dec}$				
xx.0125.00001	0x1901	6401	An01: AIN1_I < 4 mA	TroubleQuickStop	C00598/1	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	C00594/2	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.0184.00064	0x5440	21568	Ck16: Time overflow manual control	Fault	-	
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.0145.00024	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.0145.00050	0x2d32	11570	dF50: Retain 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.0123.00099	0x1763	5987	FC1: Field controller limitation	No Reaction	C00570/4	0xF000
xx.0123.00094	0x175e	5982	FCH1: Switching frequency reduction	No Reaction	C00590	0x2000
xx.0123.00095	0x175f	5983	FCH2: Max. speed for Fchop	No Reaction	C00588	0xF000
xx.0123.00057	0x1739	5945	ID1: Motor data identification error	WarningLocked	-	0xF000
xx.0123.00058	0x173a	5946	ID3: CINH motor data identification	WarningLocked	-	0xF000
xx.0123.00059	0x173b	5947	ID4: Resistor identification error	Warning	-	0xF000

Error number 32 bits	16 bits$_{hex}$	16 bits$_{dec}$	Error message	Response (Lenze setting)	can be set in	CAN Emergency Error Code
xx.0125.00011	0x190b	6411	Io11: DigOut level	Warning	C00598/3	0xF000
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.0140.00013	0x280d	10253	MCI1: Module missing / incompatible	No Reaction	C01501/2	0x7000
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	Fault	-	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: Max. current reached	Fault	-	0x2000
xx.0123.00071	0x1747	5959	OC11: Active clamp operation	Warning	-	0xF000
xx.0123.00065	0x1741	5953	OC12: I2xt overload - brake resistor	Fault	-	0xF000
xx.0123.00090	0x175a	5978	OC13: Max. 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 (X21) triggered	Fault	C00585	0x4000
xx.0119.00000	0x1300	4864	OH4: Heatsink temp. > shutdown temp. -5°C	No Reaction	C00582	0x4000
xx.0123.00032	0x1720	5920	OS1: Max. speed limit reached	No Reaction	C00579	0x8400
xx.0123.00001	0x1701	5889	OT1: Max. torque reached	No Reaction	C00608	0x8300
xx.0123.00093	0x175d	5981	OT2: Speed controller output limited	No Reaction	C00567	0xF000
xx.0123.00014	0x170e	5902	OU: Überspannung Zwischenkreis	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: Invalid memory module par. set	Fault	-	0x6300
xx.0144.00008	0x2c08	11272	PS08: Invalid device par.	Fault	-	0x6300
xx.0144.00009	0x2c09	11273	PS09: Invalid par. format	Fault	-	0x6300
xx.0144.00010	0x2c0a	11274	PS10: Memory module binding invalid	Fault	-	
xx.0123.00205	0x17cd	6093	SD3: Open circuit - feedback system	Fault	C00586	0x7300
xx.0123.00200	0x17c8	6088	SD10: Speed limit - feedback system 12	Fault	C00607	0x7300
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	Warning	-	0x3000
xx.0111.00005	0x0b05	2821	Su05: IO supply overload	Warning	C00598/4	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

9.10.6 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 Logbook and 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)" (📖 353).

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 (plug X10).

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. • The device recognises if the power section is switched on and off too frequently. • 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. The minimum waiting time between two mains switching processes is approximately three seconds.

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 (100 ms after switch-on U is < 19V).	With internal supply voltage via the power electronics, the controller must be replaced. With external supply voltage, check the correct connection and/or the stability of the supply voltage.

Su05: IO supply overload [xx.0111.00005]

Response (Lenze setting printed in bold)	**Setting:** C00598/4 (☑ Adjustable response)
☑ 0: No Reaction ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information	

Cause	Remedy
The supply of the I/Os is overloaded. The total current at pin 1 of X41 ... X43, X50 is higher than 1 A.	Check connected input sensors. Do not exceed maximally available total current of 1 A.

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: Heatsink overtemperature [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). Maybe the ambient temperature of the controller is too high or the fan or its ventilation slots are dirty.	• Check control cabinet temperature. • Clean filter. • Clean controller. • If required, clean or replace the fan. • Provide for sufficient cooling of the device.

OH3: OH3: Motor temperature (X21) 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 at plug X21 has been tripped. Possible causes: • The motor is overheated so that the thermal contact integrated into the motor has been switched. • An open circuit or a loose contact at the connections mentioned above has occurred.	• Check motor temperature monitoring. • Provide for sufficient cooling of the motor. • Check terminals for open circuit or loose contact.

OC5: Ixt overload [xx.0119.00050]

Response (Lenze setting printed in bold)
☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information

Cause	Remedy
The Ixt overload check has tripped. Possible causes: • Wrong dimensioning of the device with regard to its motor load. • Load cycles are not complied with.	• Check and, if required, correct dimensioning of the device and the motor load with regard to technical data. • 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. • 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**
Maximum current monitoring has tripped.	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. • Depending on the configuration of the auto-start lock function, set C00142 so that, when this error is tripped, the controller only restarts after the controller inhibit has been switched. • If this error message remains active longer than the time set in C00601, a "Fault" is tripped.	• Reduce load in generator mode. • Use a brake resistor. • Use a regenerative power supply unit. • 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. • Depending on the configuration of the auto-start lock function, set C00142 so that, when this error is tripped, the controller only restarts after the controller inhibit has been switched.	• Switch on mains supply or ensure sufficient supply via DC bus. • 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. • Mostly, incorrectly executed motor connections are the cause. • 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. ▶ Motor control: Defining current limits	• Check motor connections and the corresponding plug connector on the device. • Only use permissible combinations of device power and motor power. • 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.
• Mostly, incorrectly executed motor connections are the cause.
• 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.
• 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.
• 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.
	• Use motor filters, cable lengths, and cable types recommended by Lenze.
	• If motor filters with additional terminals for +UG and –UG and devices greater or equal 3 kW are used:
	  • Up to version 01.xx.xx: Set resp. to earth fault (C00602) to "0: No Reaction".
	  • From version 02.00.00: Deactivate earth fault detection during operation by setting the filter time (C01770) to 250 ms.
	• If motor cables longer than 50 m are used:
	  • From version 02.00.00: Increase filter time for earth fault detection during operation (C01770).

OC10: Maximum current reached [xx.0123.00030]

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 displays that the maximum current has been reached.	• Check and, if required, correct dimensioning of the load with regard to the installed device power. • 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. • 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. • Adjust set speed limitation (C00909) and frequency limitation (C00910).

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 motor parameters, an error has occurred. Possible causes: • Interrupted motor cable. • Switched-off power section during the identification. • Implausible start parameter settings.	• Check the motor connections and the corresponding plug connector on the device and, if necessary, the motor terminal box. • Correct start parameters for the motor parameter identification (motor nameplate data). • 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. • This cancels the identification process. The Lenze setting of the motor data is used.	• Do not set controller inhibit during the motor data identification. • Do not execute any device function which may activate controller inhibit.

ID4: Resistance 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. • The parameters for cable cross-section and cable length are implausible.	Enter sensible values for cable cross-section and motor cable length.

OC12: I2xt overload - brake resistor [xx.0123.00065]

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 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. • A permanent clamp operation causes an overload disconnection.	Reduce setpoint generation dynamics or motor load.

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. • 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. • 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 status, the speed controller is not able anymore to correct the system deviation. • Only during "Closed loop" operation or with vector control (SLVC).	• Observe load requirements. • Correct dimensioning or reduce setpoint generation dynamics if necessary. ▶ Motor control

FCH1: 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. • Correct dimensioning or reduce setpoint generation dynamics if necessary. ▶ Motor control

FCH2: 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. • The maximum speed has been exceeded depending on the switching frequency.	Select the correct maximum speed as a function of the switching frequency. ▶ Motor control: Determine 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. • Correct dimensioning or reduce setpoint generation dynamics if necessary. ▶ 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. • Correct dimensioning or reduce setpoint generation dynamics if necessary. • Check parameter setting of the current controller with regard to the motor controllers (e.g. reduce Vp). ▶ 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. • Correct dimensioning or reduce setpoint generation dynamics if necessary. ▶ Motor control

FC1: 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. • Correct dimensioning or reduce setpoint from the field weakening range if necessary. ▶ 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. • Correct dimensioning if necessary. • In case of VFCplus operation: Check Vmin boost (C00016). 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 • 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. • Check the trigger threshold (C00599).

SD10: Speed limit for 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. $n_{encoder}$ <= (f_{max} x 60) / encoder increment (for f_{max} = 10 kHz)

SD3: Feedback system open circuit [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**
The device has detected open circuit in the feedback system.	Check wiring of the feedback system and the corresponding terminals.

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. • Only if the analog input has been configured as a current loop of 4 ... 20 mA (C00034/1 = 2).	• Check wiring of the analog X3/A1I input terminal for open circuit. • Check minimum current values of the signal sources.

Io11: DigOut level [xx.0125.00011]

Response (Lenze setting printed in bold)	**Setting:** C00598/3 (☑ Adjustable response)
☑ 0: No Reaction ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☐ 4: WarningLocked ☒ **5: Warning** ☐ 6: Information	

Cause	**Remedy**
The actual level of a digital output does not correspond to the setpoint level.	• Check wiring of the digital output terminal for short circuit to earth. • Check whether the digital output terminal is by mistake triggered externally with a HIGH signal although a LOW signal should be output.

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. • Switch off controller, correctly plug in the module, switch on the controller again. • Mains switching or restart of the controller, respectively. • Replace module/controller. • Please contact Lenze if the problem occurs again.

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 • Received too many faulty telegrams. • Damaged cable (e.g. loose contact). • Two nodes with the same ID.	• Check wiring and bus terminating resistor. • Set identical baud rate for each bus node. • Assign different IDs to nodes. • 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. • 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. • The current number of incorrectly transmitted CAN telegrams is displayed in C00372/1. • The current number of incorrectly received CAN telegrams is displayed in C00372/2. • The current CAN error status is displayed in C00345.	• Check wiring and bus terminating resistor. • Set identical baud rate for each bus node. • Assign different IDs to nodes. • 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 • Being a Heartbeat consumer, the device has not received a Heartbeat telegram from Heartbeat producer 1 ... 7 within the defined time. • The current statuses of the Heartbeat producers are displayed in C00347/1...7.	• Reactivate Heartbeat producers by mains switching, restarting the controller, or a CAN Reset Node. • Reparameterise CAN Heartbeat producer time or switch off consumer monitoring and reset error status if latched. ▸ "CAN on board" system bus: Heartbeat protocol

CA0F: CAN control word [xx.0131.00015]

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") 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 been triggered. • RPDO1 has not been received within the monitoring time set in C00357/1 or was faulty.	• Set correct telegram length for CAN master (transmitter). • Eliminate electrical interference (e.g. EMC). • Adjust monitoring time 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 been triggered. • RPDO2 has not been received within the monitoring time set in C00357/2 or was faulty.	• Set correct telegram length for CAN master (transmitter). • Eliminate electrical interference (e.g. EMC). • Adjust monitoring time 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 been triggered. • RPDO3 has not been received within the monitoring time set in C00357/3 or was faulty.	• Set correct telegram length for CAN master (transmitter). • Eliminate electrical interference (e.g. EMC). • Adjust monitoring time 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 been triggered. • RPDO4 has not been received within the monitoring time set in C00357/4 or was faulty.	• Set correct telegram length for CAN master (transmitter). • Eliminate electrical interference (e.g. EMC). • Adjust monitoring time C00357/4 or switch off time monitoring.

MCI1: 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. • Check if the module is plugged in correctly. • 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. • 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. • 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. • 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: • OK: StateLine V2.0 to StateLine V3.0 • OK: StateLine V2.0 to HighLine V2.0 • Not OK: HighLine Vx.x to StateLine Vx.x • 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. • 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: • OK: StateLine V2.0 to StateLine V3.0 • OK: StateLine V2.0 to HighLine V2.0 • Not OK: HighLine Vx.x to StateLine Vx.x • 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. • The error occurs while loading the parameter set. • 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. • The error occurs while loading the parameter set. • 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. • 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
Device personalisation is active: The binding ID of the memory module does not comply with the binding ID of the controller.	• Use memory module/controller with matching binding IDs. • Contact machine manufacturer. **Note:** It is not possible for Lenze to modify a replacement device via special accesses in such a way that it cooperates 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.

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. • Cyclic mains switching every 3 min is permissible.	• After switching the mains 3 times in one minute, there must be a switching pause of 9 min. • Please contact Lenze if the problem occurs again.

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. • 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. • Either caused by an internal hardware error or by lack of mains switching after a firmware download.	Mains switching • Please contact Lenze if the problem occurs again.

Ck16: Time overflow manual operation [xx.0184.00064]

Response (Lenze setting printed in bold)
☐ 0: No Reaction ☒ **1: Fault** ☐ 2: Trouble ☐ 3: TroubleQuickStop ☐ 4: WarningLocked ☐ 5: Warning ☐ 6: Information

Cause	**Remedy**
PC manual control: The connection monitoring has tripped. • 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 communication link between PC and controller. • Check voltage supply/function of the controller. • Adjust the timeout (C00464/1).

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. Possible causes: • The short-circuit check of the fan connection has tripped. • The speed monitoring of the fan has tripped.	• Check the fan for short-circuit. • 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]

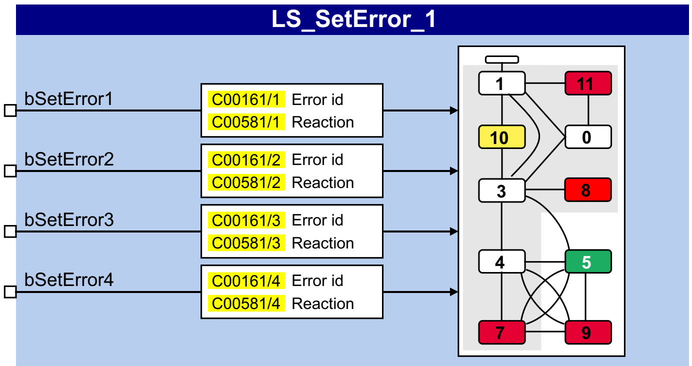
Response (Lenze setting printed in bold)	**Setting:** C00581/4 (☑ Adjustable response)
☒ **0: No Reaction** ☑ 1: Fault ☑ 2: Trouble ☑ 3: TroubleQuickStop ☑ 4: WarningLocked ☑ 5: Warning ☑ 6: Information	

Cause	**Remedy**
User error 4 has been tripped via the *bSetError4* input of the LS_SetError_1 system block.	User-defined.

9.11 "LS_SetError_1" system block

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 Data type	Information/possible settings
bSetError1 BOOL	Input for tripping "US01: User error 1" • Error subject number: 980 • Error number: (C00581/1 x 0x0400000) + (980 x 0x10000) + (C00161/1)
bSetError2 BOOL	Input for tripping "US02: User error 2" • Error subject number: 981 • Error number: (C00581/2 x 0x0400000) + (981 x 0x10000) + (C00161/2)
bSetError3 BOOL	Input for tripping "US03: User error 3" • Error subject number: 982 • Error number: (C00581/3 x 0x0400000) + (982 x 0x10000) + (C00161/3)
bSetError4 BOOL	Input for tripping "US04: User error 4" • Error subject number: 983 • 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		
	2	Trouble		
	3	TroubleQuickStop		
	4	WarningLocked		
	5	Warning		
	6	Information		

10 CANopen option

The 8400 protec StateLine controller has a CANopen interface which is used to exchange e.g. 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 CANopen (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 interface classified in different subcategories.
- Information on CAN communication modules and CANopen interfaces of other Lenze devices is provided in the "CAN" communication manual in the Lenze library.

Short overview of CAN interfaces for 8400 protec series:

Connection	Interface	Device version	
		StateLine HighLine	EMS
X31/X32	**CANopen option** CANopen is an isolated fieldbus und can be used for networks with multiple nodes. In case of fieldbuses, node address and baud rate can be set using the DIP switch under the service cover or parameterised with the »Engineer«.	optional	optional
X34	**CANopen master PLC** The control (PLC) integrated in the 8400 protec EMS controller has an own CANopen interface (X34). This interface is configured via he CANopen manager implemented in the »PLC Designer« programming software.	-	●
X35	**CAN on board** "CAN on board" is exclusively suitable for short point-to-point connections between two controllers, e.g. for synchronisation. Node address and baud rate must be parameterised with the »Engineer«. • Without CANopen option, but only with PROFIBUS or PROFINET. • Without safety option 30 (SO30). • Pay attention to notes in the hardware manual for EMC-compliant wiring and short cable lengths as there is no isolation towards the control electronics of the controller.	optional	-

10.1 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:
  - 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)

10.1.1 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 (max. number of nodes)	Depending on the number of SDO channels set in C00366: • 1 SDO: Node address 1 ... 127 (max. 127 nodes) • 2 SDO: Node address 1 ... 63 (max. 63 nodes) • adjustable via DIP switches or via code C00350.
Adjustable baud rates	20*, 50, 125, 250, 500, 800*, 1000* kbps (* from HighLine onwards) • adjustable via DIP switches or via code C00351.
Process data	• Max. 3 transmit PDOs (TPDOs) with 1 ... 8 bytes (adjustable) • 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) • Time-controlled, 1 to x ms • After the reception of 1 to 240 sync telegrams

10.1.2 Supported protocols

Protocols	
Standard PDO protocols	PDO write PDO read
SDO protocols	SDO download SDO download initiate SDO download segment
	SDO upload SDO upload initiate SDO upload segment
	SDO abort transfer
	SDO block download SDO block download initiate SDO block download end
	SDO block upload SDO block upload initiate 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) • Up to 7 Heartbeat Producers can be monitored.
	Emergency telegram (to master)
More protocols	Transmitting and receiving a sync telegram • Synchronisation of the internal time base to the reception of the CAN sync telegram is possible. ▶ Synchronisation of the internal time base

10.1.3 Communication time

The communication time is the time between the start of a request and the arrival of the corresponding response.

Tip!

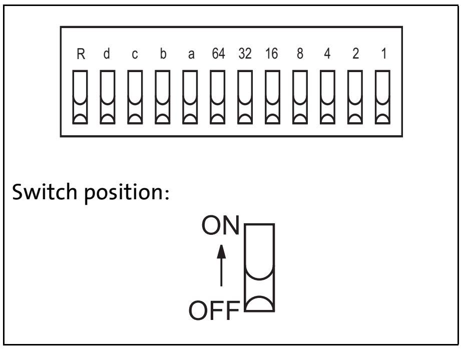
The communication times in the CAN network depend on:
- the processing time in the device
- the telegram runtime (baud rate/telegram length)
- the bus load (especially if the bus is charged with PDOs and SDOs at a low baud rate)

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

10.2 Possible settings via DIP switch



[10-1] DIP switch

The following can be set via the front panel DIP switches:

- Bus terminating resistor
  Switch: "R"
- Baud rate
  Switch: "a" ... "d"
- Node address
  Switch: "1" ... "64"

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.

10.2.1 Activating the bus terminating resistor

The system 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 "R" DIP switch.

DIP switch position ("R"):

- OFF = bus terminating resistor is inactive
- ON = bus terminating resistor is active

10.2.2 Setting the baud rate

The baud rate can be set via code C00351 or with the DIP switches a to d:

Note!
- All DIP switches (a ... d, 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
d	c	b	a	
OFF	OFF	**ON**	**ON**	50 kbps
OFF	OFF	**ON**	OFF	125 kbps
OFF	OFF	OFF	**ON**	250 kbps
OFF	OFF	OFF	OFF	**500 kbps**
ON	**ON**	**ON**	OFF	800 kbps (only effective from "HighLine" onwards)
OFF	**ON**	OFF	OFF	1000 kbps (only effective from "HighLine" onwards)

10.2.3 Setting the node address

The node address can be set via code C00350 or with the DIP switches 1 to 64.

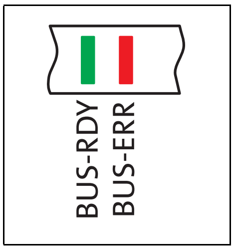
- The labelling on the housing corresponds to the values of the individual DIP switches for determining the node address.
- The valid address range depends on the number of SDO channels set in C00366:
  - 1 SDO (Lenze setting): 1 ... 127
  - 2 SDO: 1 ... 63

Note!
- The addresses of the nodes must differ from each other.
- All DIP switches (a ... d, 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**						

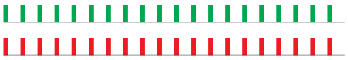
10.3 LED status displays for the system bus



Information about the status of the system bus can be obtained quickly via LED displays "BUS-RDY" and "BUS-ERR" on the front of the controller.

- The meaning can be seen from the tables below.

Controller is not (yet) active on the system bus

LED display	Meaning
 (BUS-ERR is permanently lit)	Controller is not active on the system bus / Bus Off
 (BUS-RDY and BUS-ERR jitter)	Automatic detection of baud rate is active

Controller is active on the system bus

- The LED "BUS-RDY" LED signals the CANopen status:

LED display	CANopen state
 (BUS-RDY is blinking every 0.2 seconds)	Pre-Operational
 (BUS-RDY is permanently lit)	Operational
 (BUS-RDY is blinking every 1 seconds)	Stopped

- The "BUS-ERR" LED signals a CANopen error:

LED display	CANopen error
 (BUS-ERR is blinking once, then off for 1 second)	Warning Limit reached
 (BUS-ERR is blinking twice, then off for 1 second)	Node Guard Event
 (BUS-ERR is blinking three times, then off for 1 second)	Sync Message Error (only possible in the "Operational" state)

10.4 Going online via the system bus

The system bus (CANopen) 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 incl. connection cable and voltage supply adapter • for DIN keyboard connection (EMF2173IB) • for PS/2 keyboard connection (EMF2173IBV002) • for PS/2 keyboard connection with electrical isolation (EMF2173IBV003)	Parallel interface (LPT port)
PC system bus adapter 2177 incl. connection cable (EMF2177IB)	USB (Universal Serial Bus)

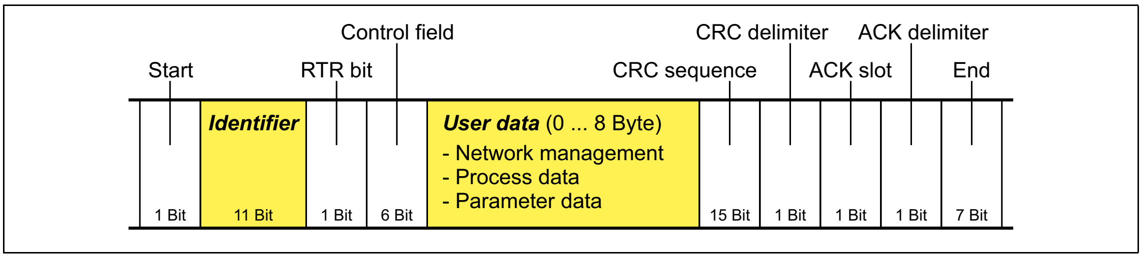
Note!

- 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.

10.5 Reinitialising the CANopen interface

The C00002/26 = "1: On / start" device command reinitialises the CANopen interface of the controller ("Reset node"), which is required after e.g. changing the data transfer rate, the node address or the identifiers, respectively.

10.6 Structure of the CAN data telegram



[10-2] 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

10.6.1 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: For process data, heartbeat and emergency objects as well as network management and syn telegrams, the identifier is assigned freely by the user (either manually or automatically by the network configurator) or is firmly allocated.

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 (📖 376)

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 (📖 394)

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.

10.6.2 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.

10.7 Communication phases/network management

Regarding communication via the system bus, the controller distinguishes between the following statuses:

Status	Explanation
"Initialisation" (Initialisation)	After switch-on, an initialisation run is carried out. • During this phase, the controller is not involved in the data exchange via the bus. • The standard values are re-written to all CAN-relevant parameters. • After initialisation is completed, the controller is automatically set to the "Pre-Operational" status.
"Pre-Operational" (before being ready for operation)	Parameter data can be received, process data are ignored.
"Operational" (ready for operation)	Parameter data and process data can be received!
"Stopped" (stopped)	Only network management telegrams can be received.

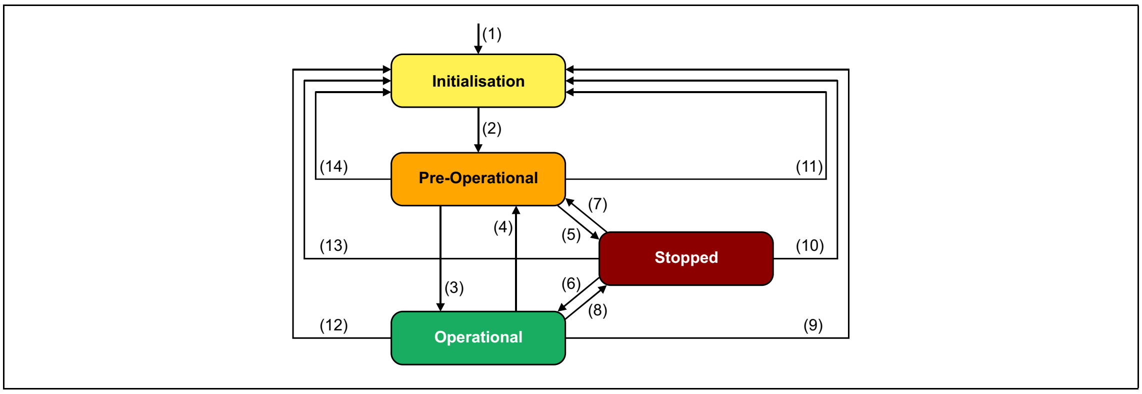
Communication object	Initialisation	Pre-Operational	Operational	Stopped
PDO			●	
SDO		●	●	
Sync		●	●	
Emergency		●	●	
Boot-up	●			
Network management (NMT)		●	●	●

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.

10.7.1 Status transitions

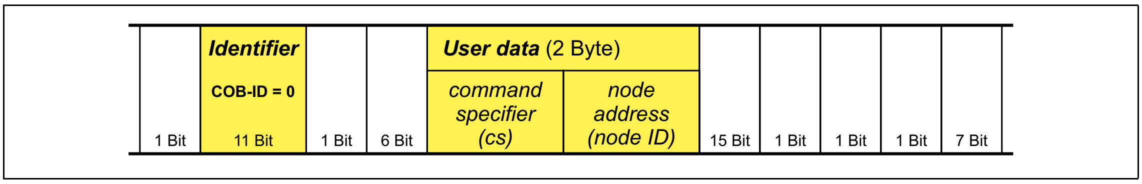


[10-3] NMT status transitions in the CAN network

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. • During initialisation, the controller is not involved in the data exchange. • 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.
[i]	From here, the master changes the statuses for the entire network. • A target address included in the NMT command defines the receiver(s). • 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. • Data can only be exchanged via process data objects if the status is "Operational"!		
(3), (6)	0x01 xx Start remote node	Operational	Network management/sync/emergency telegrams as well as process data (PDO) and parameter data (SDO) are active. Optional: When the status is changed, event and time-controlled process data (PDOs) are transmitted once.
(4), (7)	0x80 xx Enter Pre-Operational	Pre-Operational	Network management/sync/emergency telegrams and parameter data (SDO) are active.
(5), (8)	0x02 xx Stop remote node	Stopped	Only network management telegrams can be received.
(9), (10), (11)	0x81 xx Reset node	Initialisation	All CAN-relevant parameters (CiA DS 301) are initialised with the saved values.
(12), (13), (14)	0x82 xx Reset communication		All CAN-relevant parameters (CiA DS 301) are initialised with the saved values.
[i]	Meaning of the node address in the NMT command: • 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. • xx = Node ID: If a node address is specified, only the status of the node with the corresponding address changes.		

10.7.2 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:



[10-4] 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 (📖 385)

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)

10.7.3 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

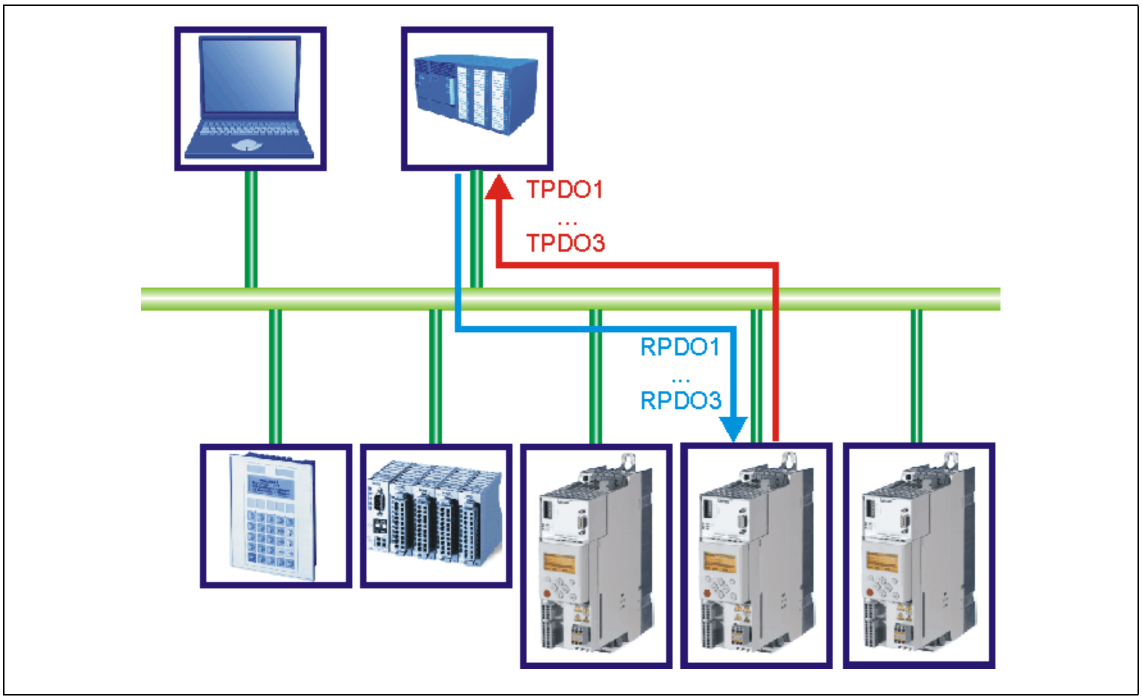
- the "Reset node" or "Reset communication" NMT telegram has been transmitted to the controller.

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.

10.8 Process data transfer



[10-5] 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 (📖 382)

10.8.1 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 | "LP_CanIn2" port block	●	●
RPDO3 | "LP_CanIn3" port block		●
TPDO1 | "LP_CanOut1" port block	●	●
TPDO2 | "LP_CanOut2" port block	●	●
TPDO3 | "LP_CanOut3" port block		●

Receive PDOs (RPDOs)

The process data objects transmitted from the system 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 (📖 256)
  - **TA "Abschaltpositionierung":** Process data assignment for fieldbus communication (📖 283)

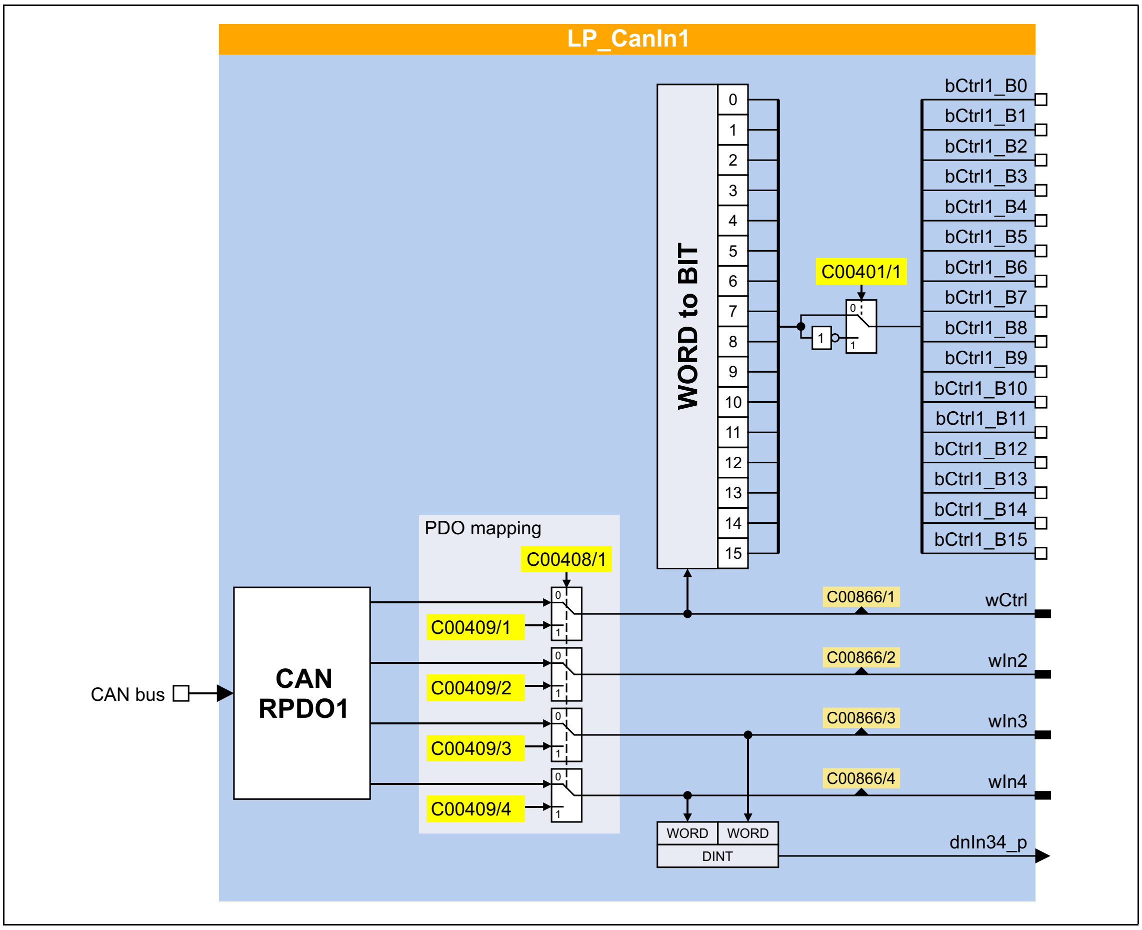
Transmit PDOs (TPDOs)

The process data transmitted from the drive to the system bus 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.
  - Vordefinierte Belegung siehe wDeviceStatusWord status word der Antriebsschnittstelle.

10.8.1.1 RPDO1 | Port block "LP_CanIn1"

The **LP_CanIn1** port block maps process data object RPDO1 in the FB Editor.

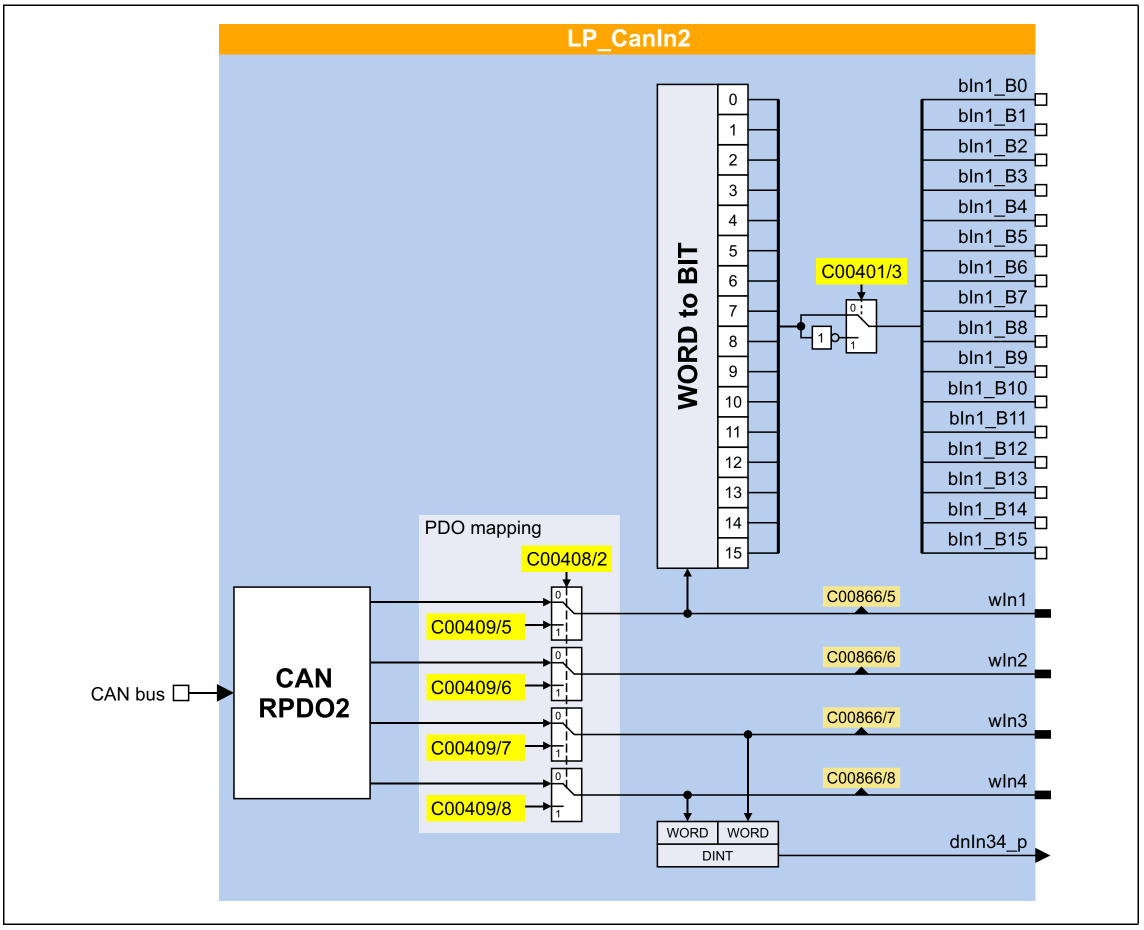


Short overview of the parameters for LP_CanIn1:

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			

10.8.1.2 RPDO2 | "LP_CanIn2" port block

The **LP_CanIn2** port block maps process data object RPDO2 in the FB Editor.

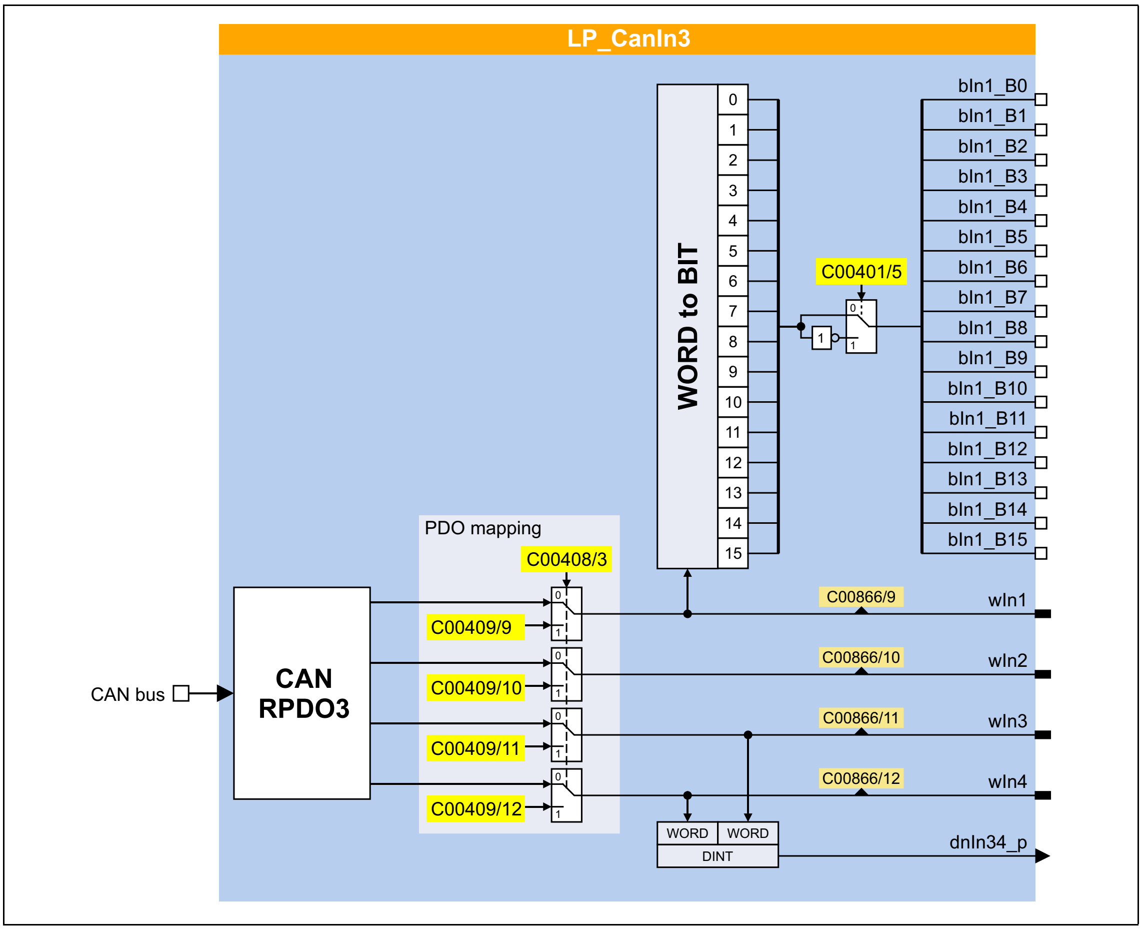


Short overview of the parameters for LP_CanIn2:

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		

10.8.1.3 RPDO3 | "LP_CanIn3" port block

The **LP_CanIn3** port block maps process data object RPDO3 in the FB Editor.

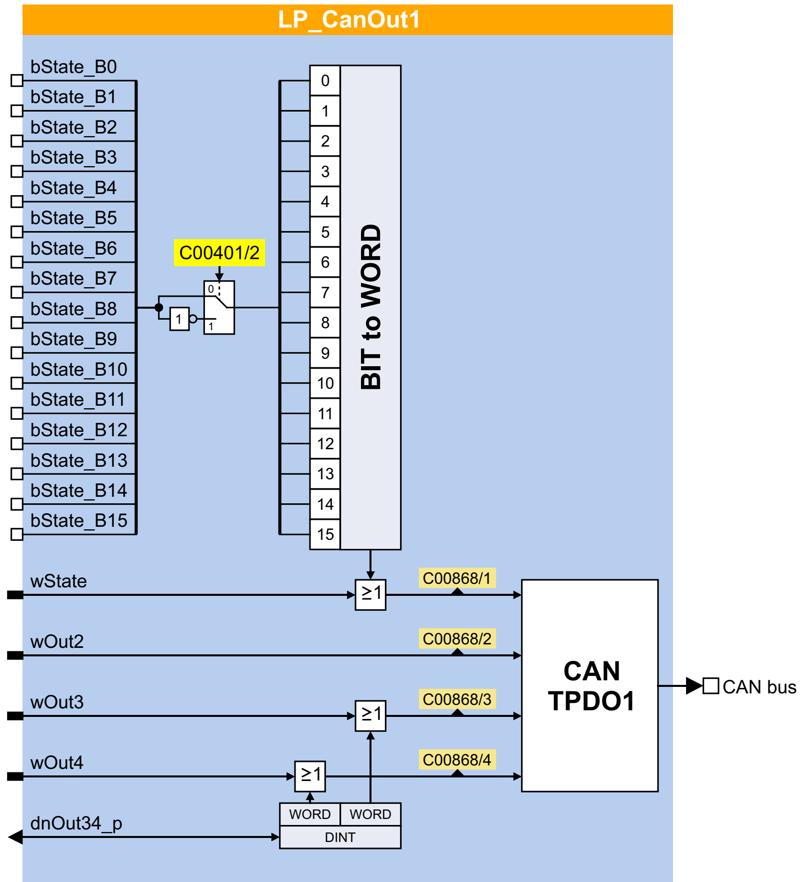


Short overview of the parameters for LP_CanIn3:

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		

10.8.1.4 TPDO1 | "LP_CanOut1" port block

The **LP_CanOut1** port block maps process data object TPDO1 in the FB Editor.

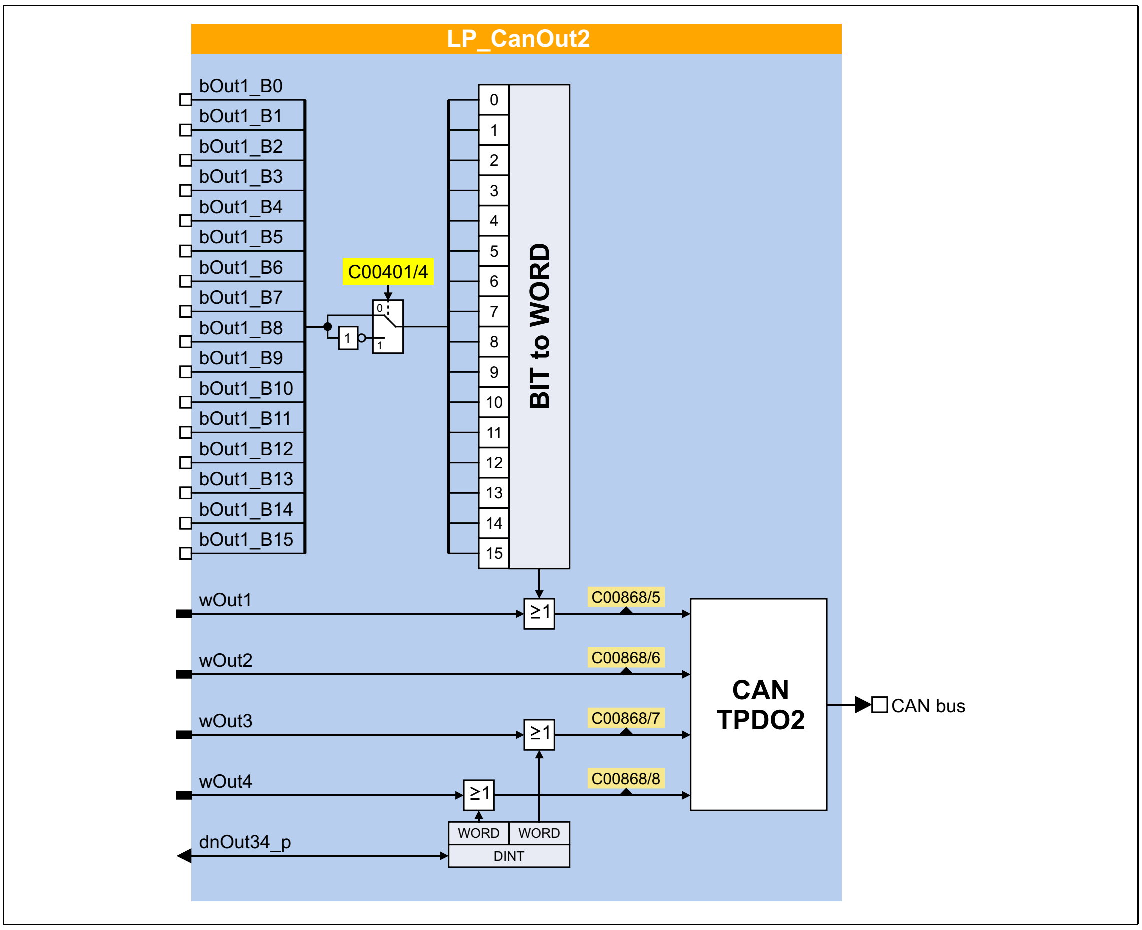


Short overview of the parameters for LP_CanOut1:

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		

10.8.1.5 TPDO2 | "LP_CanOut2" port block

The **LP_CanOut2** port block maps process data object TPDO2 in the FB Editor.

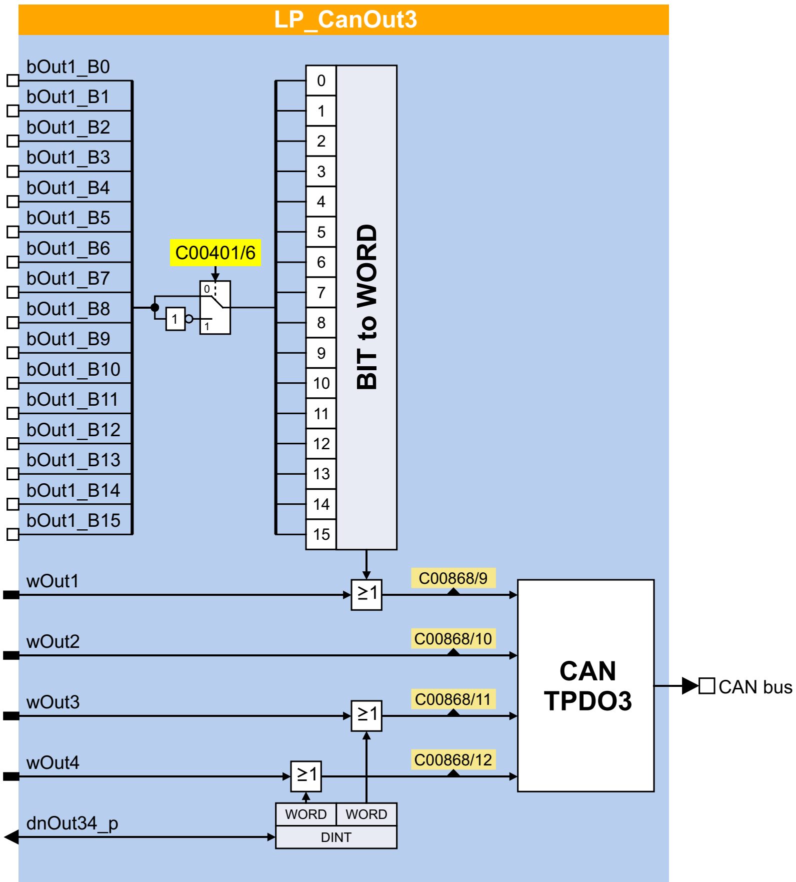


Short overview of the parameters for LP_CanOut2:

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		

10.8.1.6 TPDO3 | "LP_CanOut3" port block

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		

10.8.2 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	

10.8.3 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: 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): • Selection n = 1: The PDO is transmitted with every sync. • 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: If this value is entered, the PDO transmission is event-controlled or cyclic. (Note: The values "254" and "255" have the same meaning). 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	Bytes
CAN2_OUT			
C00322/2	Transmission mode	254	
C00324/3	Blocking time	0	ms
C00356/2	Cycle time	0	ms
C00358/2	Data length	8	Bytes
CAN3_OUT			
C00322/3	Transmission mode	254	
C00324/4	Blocking time	0	ms
C00356/3	Cycle time	0	ms
C00358/3	Data length	8	Bytes
CAN1_IN ... CAN3_IN			
C00323/1...3	Transmission mode CAN1_IN ... CAN3_IN • In the case of the RPDO serves as monitoring setting in the case of sync-controlled PDOs.	254	

Blocking time

From version 06.00.00 on, a so-called "blocking time" can be set in C00324/x, which defines the shortest transmission cycle in the case of the "Asynchronous - manufacturer-specific/device profile-specific" transmission type.

Example: Cycle time = 500 ms, blocking time = 100 ms, sporadic data change:

- With a sporadic data change < 500 ms, quickest transmission takes place every 100 ms due to the set blocking time (event-controlled transmission). The transmission cycle timer is reset to 0 if the transmission has been activated in an event-controlled way.
- 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:

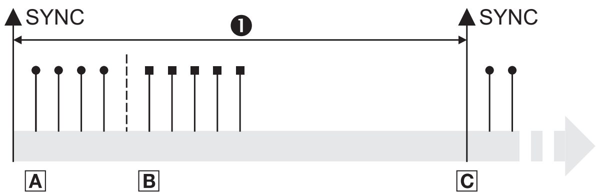
- I-1400 ... I-1402: Communication parameters for RPDO1 ... RPDO3
- I-1800 ... I-1802: Communication parameters for TPDO1 ... TPDO3

10.8.4 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 (📖 395)

Basic workflow



❶ Sync cycle time (C01121)

[10-6] 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 [10-6] 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 (📖 448)

10.8.5 Monitoring of the RPDOs for data reception

For RPDO1 ... RPDO4 each, a monitoring time can be parameterised within which the RPDO must arrive. If the RPDO is not received within the monitoring time or not with the configured sync, the response parameterised for each RPDO takes place.

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	

10.8.6 Configuring exception handling of the CAN PDOs

This function extension is available from version 02.00.00!

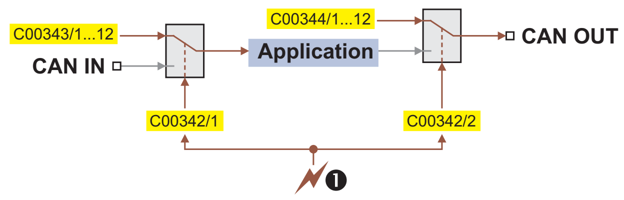
Exception handling for the CAN PDOs in the event of an error can be set via decoupling configuration and decoupling values.

- 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	



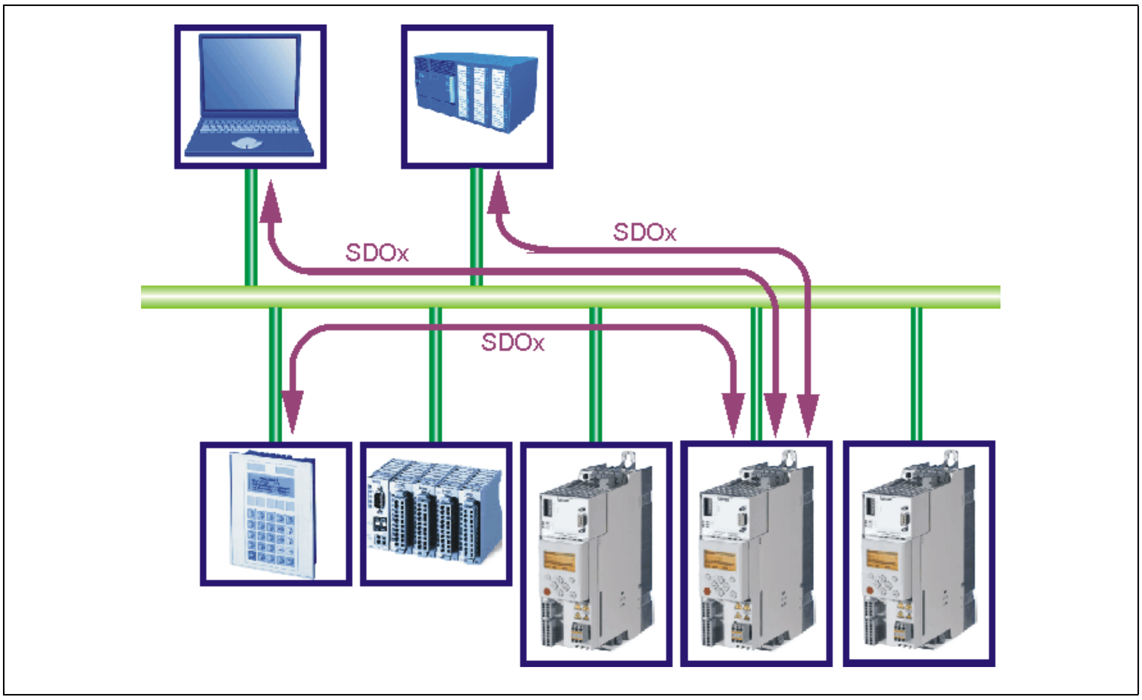
❶ Exception event (an error has occurred)

[10-7] General signal flow in the event of a configured exception

Related topics:

▶ Configuring exception handling of the output terminals (📖 226)

10.9 Parameter data transfer



[10-8] 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!

Up to and including version 02.00.00, parameter data channels 1 and 2 are activated in the Lenze setting.

From version 02.02.00 onwards, only the parameter data channel 1 is activated in the Lenze setting according to CANopen.

- In order to activate both parameter data channels according to the previous behaviour, set "2 SDO Lenze" in C00366.

10.9.1 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

10.9.2 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 (407)

The following subchapters provide detailed information on user data.

10.9.2.1 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 (📖 405)

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).

10.9.2.2 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**				

10.9.2.3 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 $_{(1 \dots 4)}$ = 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**

10.9.2.4 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 objects to be transferred longer than PDO 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).

10.9.3 Parameter data telegram examples

10.9.3.1 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 (1536 = SDO1 basic identifier to the controller)
Command	= 0x40 = "Read request" (read request of 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 (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]

10.9.3.2 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 (1536 = SDO1 basic identifier to the controller)
Command	= 0x23 = "Write request" (write request of 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 (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 (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	

10.9.3.3 Read block parameters

Task: The firmware version (code C00099) is to be read from the parameter set of the controller with the node address "12". The firmware version has a length of 11 ASCII characters which are transmitted as block parameters. Within the user data, the data width from the 2nd to the 8th byte is assigned per block.

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 (1536 = SDO1 basic identifier to the controller)
Command	= 0x40 = "Read request" (read request of 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 (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) • Bit 4 = 0 (toggle bit)
	Influence of the toggle bit on the request command The blocks are toggled one after another, i.e. the request ist made with the "0x60" (= 011**0***0000$_{bin}$) command, then with the "0x70" (= 011**1***0000$_{bin}$) command, and then again with the "0x60" command, etc. * 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$_{bin}$ • Bit 4 = 0 (toggle bit)
	Influence of the toggle bit on the transmission command • The 1st response of the controller in the command byte is "0x000**0***0000$_{bin}$" if bytes 2 ... 8 are completely filled with data and other telegrams are following. • The 2nd response of the controller in the command byte is "0x000**1***0000$_{bin}$" if bytes 2 ... 8 are completely filled with data and other telegrams are following, etc. * 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) • 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}: • Bit 0 = 1 (end of transmission) • Bit 1 ... bit 3 = 011_{bin} (3 bytes do not contain any data) • Bit 4 = 1 (toggle bit)
	Influence of the final bit and the residual data length on the transmission command • The end of transmission is signalled via the set final bit 0. • Bits 1 ... 3 reveal the number of bytes that do not contain any data anymore. * Toggle bit
Data 1 ... 7	= "0.00" (ASCII representation) The result of the data block transmission is: "01.00.00.00"

10.10 Monitoring

10.10.1 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:

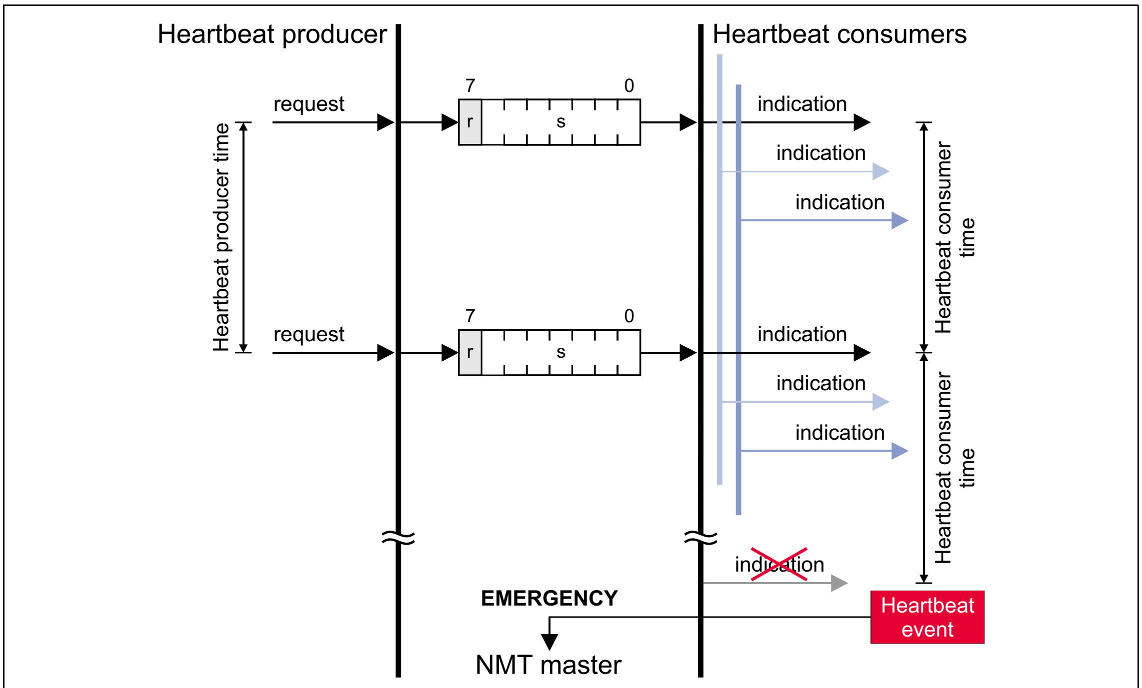
- Several nodes with identical node address
- Wrong baud rate of one or several nodes
- Too high cable length
- Too many or no terminating resistors
- Too high bus load/too many data telegrams
  (e.g. since a node permanently transmits event-controlled due to data changes of an analog signal/actual value.)
- EMC interferences on the system bus
  (e.g. since the CAN bus cable next to the motor cable is unshielded.)

C00364 displays whether such an error is active.

10.10.2 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)

[10-9] 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.

10.10.2.1 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

10.10.2.2 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 of the heartbeat producer 1 ... n	-		●	
C00381	Heartbeat producer time	0	ms		●
C00385/1...n	CAN node address of the heartbeat producer 1 ... n	0		●	
C00386/1...n	Heartbeat consumer time 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.

10.10.2.3 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

10.10.3 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:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Emergency error codes		**Error register**	**Manufacturer-specific error message**				
LOW byte	HIGH byte	I-1001	0x00 (Reserved)	LOW word		HIGH word	
				LOW byte	HIGH byte	LOW byte	HIGH byte
See table below			• For emergency error code 0xF000: Lenze error number (value displayed in C00168) • All other emergency error codes have a value of "0" here.				

Emergency error codes	Error register	Cause
0x0000	0xXX	One of several errors eliminated
	0x00	One error has been eliminated (error-free status afterwards)
0x3100	0x01	Supply voltage of standard device faulty or failed
0x8100	0x11	Communication error (warning)
0x8130	0x11	Life guarding error or heartbeat error
0x8150	0x11	Collision of identifiers (COB-IDs): An identifier parameterised for reception is also used for transmission.
0x8210	0x11	PDO length shorter than expected
0x8220	0x11	PDO length greater than expected
0x8700	0x11	Monitoring of the sync telegram
0xF000	0x01	Generic error • An error with a "Fault", "Trouble", "TroubleQSP", "Warning", or "SystemFault" error response occurred in the standard device. • Error message is the Lenze error number (C00168).

The Short overview (A-Z) of error messages of the operating system includes a list of more emergency error codes. (📖 353)

Example

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Emergency error codes		**Error register**	**Manufacturer-specific error message**				
0x00	0xF0	0x01	0x00 (Reserved)	Lenze error number			
Generic error				▶ Error messages of the operating system Corresponding error-free message: Value "0x00000000"			

Tip!
A detalied description can be found in CAN specification DS301, V4.02.

10.11 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 • "BaseLine C" version: n = 1 • "StateLine" version: n = 7 • "HighLine" version: n = 15	C00385/1...n 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 Index	 Subindex	 Name	Relationship to Lenze code
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 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 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 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 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 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 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 **I-1000**	Name: **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**	

[10-1] 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: **I-1001**	Name: **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 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 (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 or 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".				

[10-2] 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".				

[10-3] 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

<table>
<tr><td>Index:
I-1016</td><td colspan="6">Name:
Consumer heartbeat time</td></tr>
<tr><td>Subindex</td><td>Default setting</td><td colspan="3">Setting range (min. value | unit | max. value)</td><td>Access</td><td>Data type</td></tr>
<tr><td>0: Highest subindex supported</td><td>1 (for BaseLine)
7 (for StateLine)
15 (for HighLine)</td><td colspan="3">- (read access only)</td><td>ro</td><td>U16</td></tr>
<tr><td>1 ... n: Consumer heartbeat time</td><td>0</td><td>0</td><td></td><td>65535</td><td>rw</td><td>U16</td></tr>
</table>

Monitoring time for the nodes to be monitored via heartbeat. ▶ Heartbeat protocol (📖 414)

- 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: C00385/x Heartbeat time: 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 (Reserved)	**Node ID**	**Heartbeat time** in [ms]	

[10-4] Data telegram assignment

I-1017

Index: **I-1017**	Name: **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 (414)

- 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: **I-1018**	Name: **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 • 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 • For SDO server channel 1: node address (C00350) + 0x600
2	Specification of send identifier • For SDO server channel 1: node address (C00350) + 0x580

<table>
<tr><th colspan="3">8th byte</th><th>7th byte</th><th colspan="2">6th byte</th><th>5th byte</th></tr>
<tr><th colspan="3">Data 4</th><th>Data 3</th><th colspan="2">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 colspan="2">Bit 10 ... bit 0</td></tr>
<tr><td>0</td><td>0</td><td colspan="3">Extended identifier*</td><td colspan="2">11-bit identifier</td></tr>
<tr><td colspan="7">* The extended identifier is not supported. Bit 11 ... bit 29 must be set to "0".</td></tr>
</table>

[10-5] Data telegram assignment

I-1201

Index: **I-1201**	Name: **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

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".				

[10-6] 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)

<table>
<tr><th colspan="3">8th byte</th><th>7th byte</th><th colspan="2">6th byte</th><th>5th byte</th></tr>
<tr><th colspan="3">Data 4</th><th>Data 3</th><th colspan="2">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 colspan="2">Bit 10 ... bit 0</td></tr>
<tr><td>0</td><td>0</td><td colspan="3">Extended identifier = 0</td><td colspan="2">11-bit identifier = 0x644</td></tr>
<tr><td colspan="3">**0x00**</td><td>**0x00**</td><td colspan="2">**0x06**</td><td>**0x44**</td></tr>
</table>

[10-7] Data telegram assignment for subindex 1

<table>
<tr><th colspan="3">8th byte</th><th>7th byte</th><th colspan="2">6th byte</th><th>5th byte</th></tr>
<tr><th colspan="3">Data 4</th><th>Data 3</th><th colspan="2">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 colspan="2">Bit 10 ... bit 0</td></tr>
<tr><td>0</td><td>0</td><td colspan="3">Extended identifier = 0</td><td colspan="2">11-bit identifier = 0x5C4</td></tr>
<tr><td colspan="3">**0x00**</td><td>**0x00**</td><td colspan="2">**0x05**</td><td>**0xC4**</td></tr>
</table>

[10-8] 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**

[10-9] 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**

[10-10] 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. • Max. 5 subindices are supported.	-
1	RPDO1 identifier • 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 ▸ Transmission type (📖 395)	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".

[10-11] 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

[10-12] 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.

[10-13] 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. • Max. 5 subindices are supported.	-
1	RPDO2 identifier • 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 ▸ Transmission type (📖 395)	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. • Max. 5 subindices are supported.	-
1	RPDO3 identifier • 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 ▸ Transmission type (📖 395)	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: **I-1600**	Name: **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 • 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**

[10-14] 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" (388)

I-1601

Index: **I-1601**	Name: **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 • 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 | "LP_CanIn2" port block (📖 389)

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 • 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 | "LP_CanIn3" port block (📖 390)

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. • Max. 5 subindices are supported.	-
1	TPDO1 identifier • 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 ▶ Transmission type (📖 395)	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 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".				

[10-15] 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 system 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. • Max. 5 subindices are supported.	-
1	TPDO2 identifier • 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 ▸ Transmission type (📖 395)	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 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. • Max. 5 subindices are supported.	-
1	TPDO3 identifier • 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 ▸ Transmission type (📖 395)	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 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 • 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

[10-16] Data telegram assignment

IEC 61131 process data words are mapped. Only whole bytes can be mapped (1-byte/mapping entry).

Related topics:

▶ TPDO1 | "LP_CanOut1" port block (📖 391)

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 • 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 | "LP_CanOut2" port block (📖 392)

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 • The 4th mapping entry is used for the statistic mapping. For this, there is no value available.

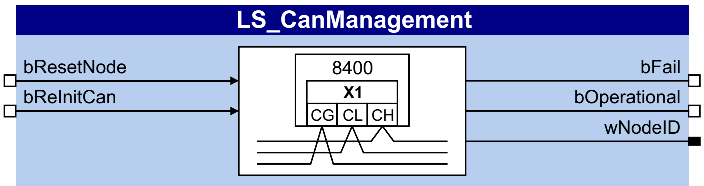
- For data telegram assignment, see object I-1A00.

Related topics:

▸ TPDO3 | "LP_CanOut3" port block (📖 393)

10.12 Internal interfaces | System block "LS_CANManagement"

The **LS_CANManagement** system block serves to control internal functions of the CAN driver (Reset node and ReInit) and to display the "Operational" status as well as the node address (analogous to the 9300 ServoPLC and ECS devices).



Inputs

Identifier Data type	Information/possible settings	
bResetNode BOOL	Reset node	
	TRUE	Carry out reset node • 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 BOOL	Reinitialisation	
	TRUE	Reinitialise "CAN on board" interface.

Outputs

Identifier DIS code | data type	Value/meaning	
bFail BOOL	Fault	
	TRUE	An event according to the error configuration in C00341 has occurred
bOperational BOOL	"Operational" status signal	
	TRUE	The system bus is in the "Operational" status
wNodeID 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 system bus ("Auto bus off recovery").

11 PROFIBUS/PROFINET option

The 8400 protec StateLine controller is also available with a PROFIBUS or PROFINET interface in order to take part in the data transfer of a fieldbus system.

When using one of these options, the major advantage for the user is the possibility of parameterising, controlling, and diagnosing the drive system via the available fieldbus.

11.1 Selection of the communication option

If you insert the 8400 protec StateLine controller via the *Insert a component* dialog into the *Project view* of the »Engineer«, you will be queried in the second dialog step **Device modules** about the communication and safety system in the device.

Select the available option in the **Communication** list field in order that the related communication parameters & parameterisation dialogs are available in the »Engineer«.

Tip!

The available communication option can also be assigned subsequently to the device in the »Engineer« any time:

1. In the *Project view*, select the 8400 protec StateLine controller.
2. Click the icon.
3. Select the available communication option in the *Insert device module* dialog box.
4. Press **Complete** to confirm your selection.

Detailed information on PROFIBUS/PROFINET is provided in the communication manual (KHB) for the respective fieldbus and in the »Engineer« online help.

11.2 Parameter setting

All codes which must be parameterised for establishing the fieldbus communication are saved in the memory module of the controller.

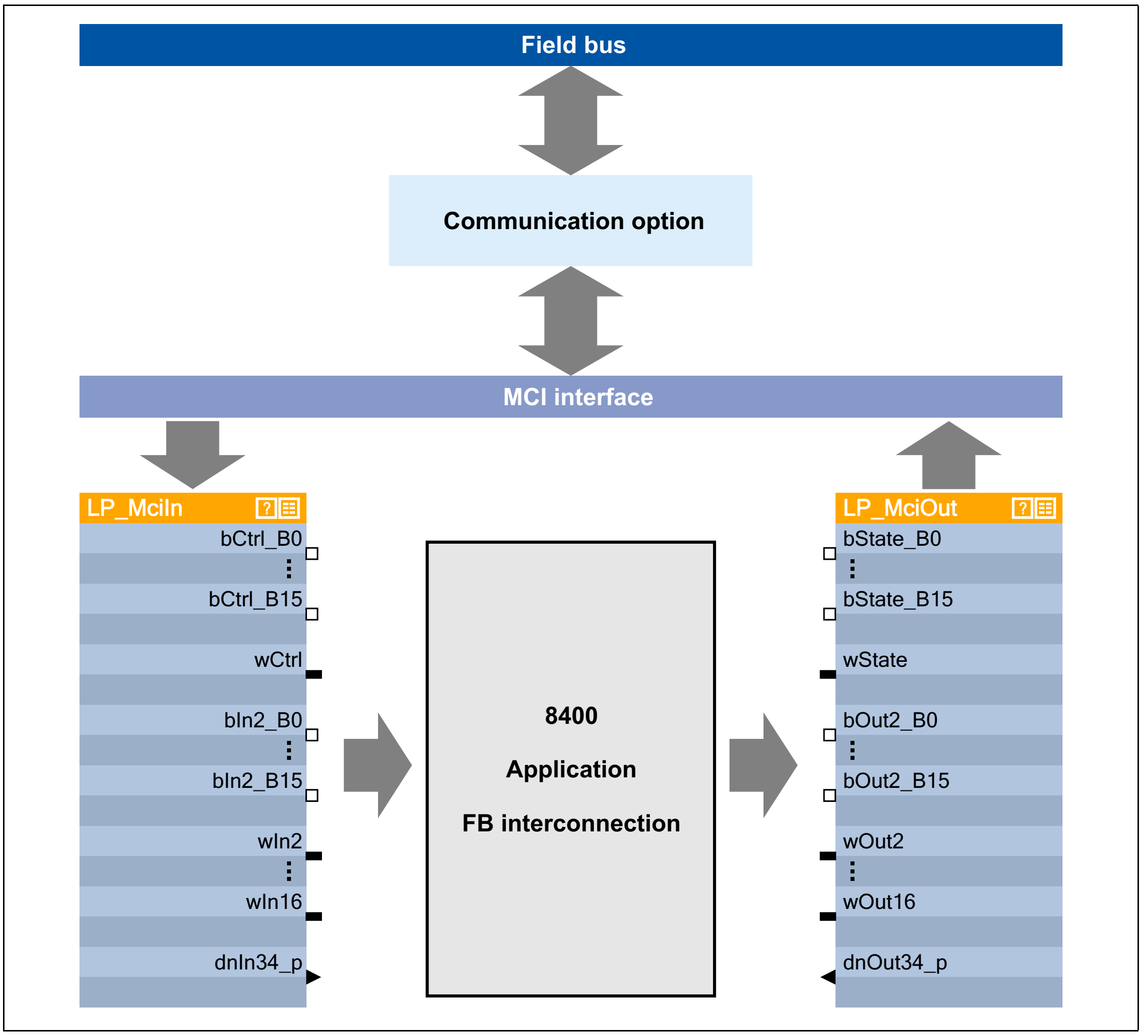
The archived data can be addressed by all bus systems supported by the controller.

The codes of the corresponding communication option are described in the respective online help and in the communication manual (KHB).

11.3 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 1ms, irrespective of the respective communication option and the type of 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 protec StateLine 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.



[11-1] External and internal data transfer between bus system, drive controller and function block interconnection

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 option 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 option 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.

11.4 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:

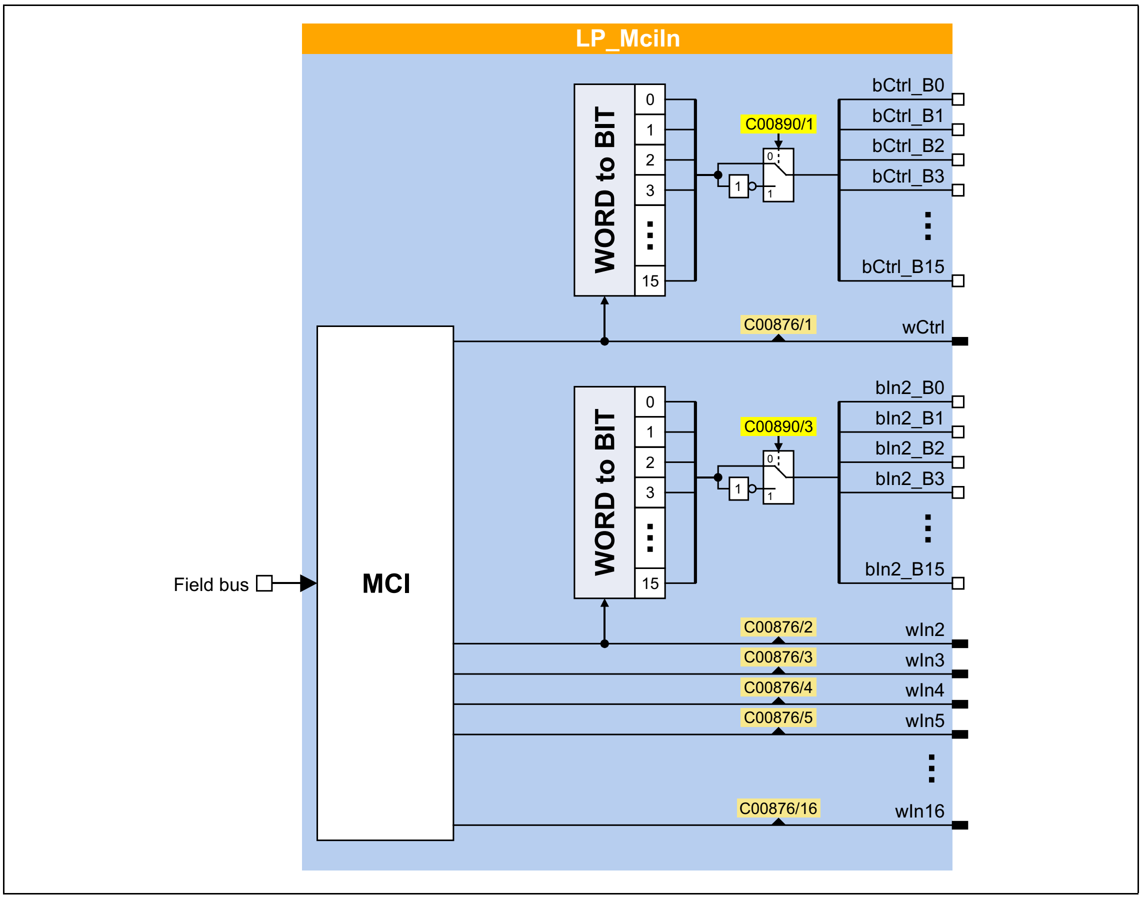
- **TA "Actuating drive speed":**
  Process data assignment for fieldbus communication (📖 256)
- **TA "Abschaltpositionierung":**
  Process data assignment for fieldbus communication (📖 283)

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).

11.4.1 Port block "LP_MciIn"

The **LP_MciIn** port block maps the received MCI-PDOs in the FB Editor.

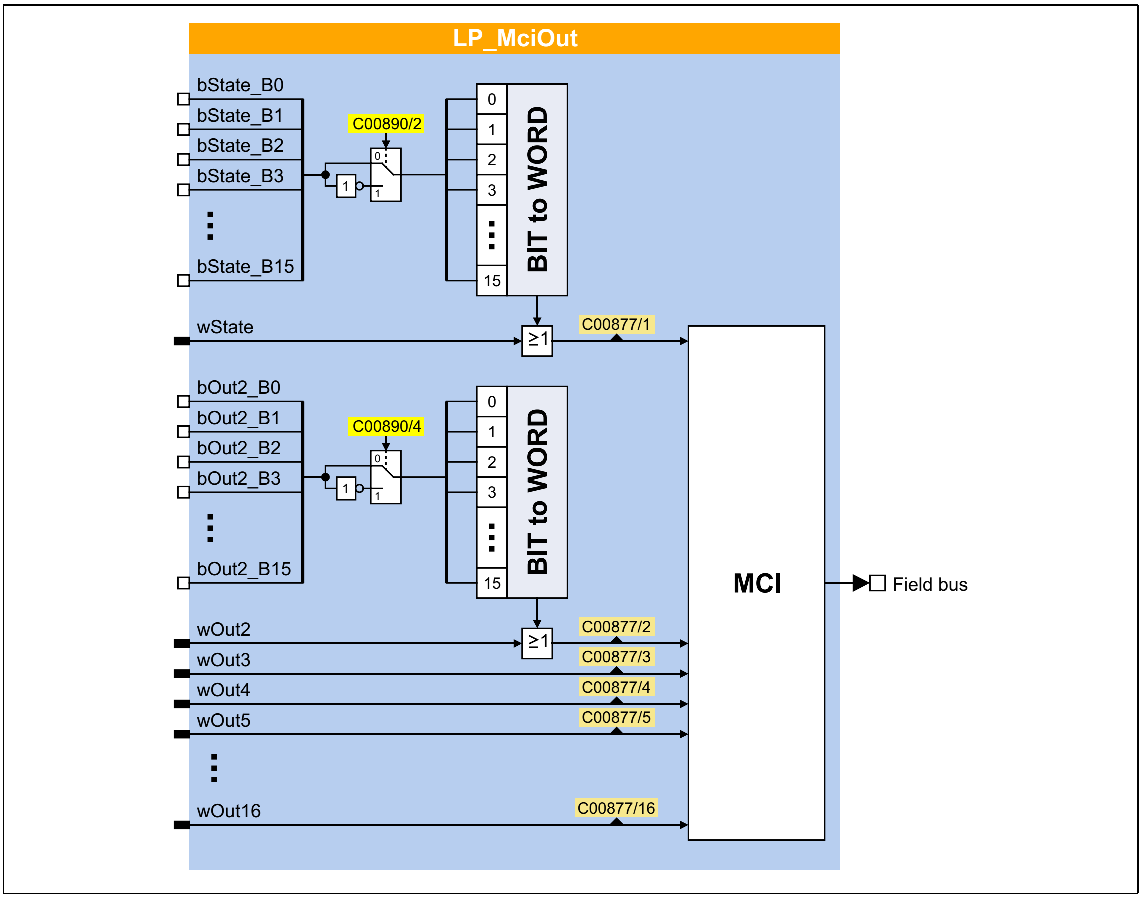


Short overview of the parameters for LP_MciIn:

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		

11.4.2 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		

12 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 ➜ sync telegram
  - Digital input

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	300	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.

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.

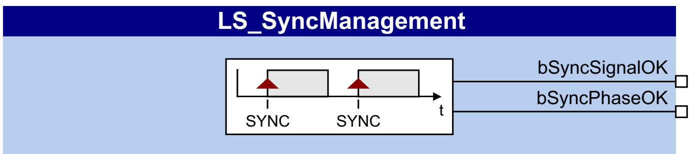
Sync correction width

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 300 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.

12.1 Internal interfaces | System block "LS_SyncManagement"

The SB **LS_SyncManagement** provides status information for synchronising the internal time base:



Outputs

Identifier Data type	Value/meaning	
bSyncSignalOK BOOL	TRUE	Sync signal OK
bSyncPhaseOK BOOL	TRUE	Sync phase position OK

13 Drive-based safety

The safety concept of the 8400 protec StateLine controller provides three safety options per design.

Safety option 10 (SO10):

- The drive-based safety implemented in the controller permits to connect external safety components, e.g. passive sensors. Active sensors with self-testing signals can be directly connected without using further components.

Safety option 20 (SO20):

- The drive is switched off safely by a higher-level safety PLC via PROFIsafe/PROFINET.

Safety option 30 (SO30):

- The safe disconnection can both be carried out by a higher-level safety PLC via PROFIsafe/PROFINET and through the connection of active or passive sensors.

Note!

Safety options 20 and 30 can be parameterised via the »Engineer«.

The motion functions are continued to be executed by the controller. The drive-based safety monitors the safe compliance with the limit values. When the limit values are exceeded, the drive-based safety starts the control functions according to EN 60204-1 directly in the controller.

The safety functions are suitable for applications according to IEC 61508 to SIL 3 and achieve the performance level (PL) e according to EN ISO 13849-1.

The requirements of the EN 954-1 standard which was valid until 30 November 2009 are fulfilled for safety option 10 to control category 4 and for safety option 20 and 30 to control category 3.

Detailed information on the integrated safety system can be found in the corresponding software manual and in the »Engineer« online help.

Detailed information on technical data and electrical installation of the integrated safety system can be found in the mounting instructions of 8400 protec StateLine.

13.1 Selection of the safety option

If you insert the 8400 protec StateLine controller via the *Insert a component* dialog into the *Project view* of the »Engineer«, you will be queried in the second dialog step **Device modules** about the communication and safety system in the device.

Select the available safety option in the **Safety engineering** list field in order that the tabs of the integrated safety system are available in the »Engineer«.

Tip!

The available safety option can also be assigned subsequently to the device in the »Engineer« any time:

1. In the *Project view*, select the 8400 protec StateLine controller.
2. Click the icon.
3. Select the available safety option in the *Insert device module* dialog box.
4. Press **Complete** to confirm your selection.

13.2 Parameter setting

Note!

Safety-relevant parameters can only be transmitted to the integrated safety system via safe parameter setting with the »Engineer«.

The parameter set is stored in the memory module and in the integrated safety system with a unique module ID, which must correspond to the effective safety address in the integrated safety system.

Detailed information on safe parameter setting of the integrated safety system can be found in the online documentation "Integrated safety".

13.3 Integration into the application

When a safety function is requested, the safety technology activates the corresponding safe monitoring function. The only standstill function executed directly is the "safe torque off" (STO) function. All other safety functions require a controller action which is safely monitored.

Note!

The execution of the corresponding action (e.g. braking, braking to standstill, holding the standstill position) requires an appropriate application interconnection which must be provided by the operator!

"LS_SMInterface" system block

The LS_SMInterface system block in the function block editor of the »Engineer« serves to transmit the control and status information from the safety system to the application. (🕮 453)

Basic workflow

1. Activation of the safety function (e.g. SS1 - safe stop 1).
   - Monitoring starts.
2. Via a control word, the safety system transmits the information to the controller that the safety function has been activated.
3. The application evaluates the control word and starts the required motion sequence (e.g. braking).

Internal communication

The drive-based safety system and the standard device communicate via an internal interface.

Note!

If the communication to the controller is interrupted, e.g. by switching off the controller, the safety system responds with the following actions:
- Fault stop with STO is activated.
- "Warning" error message is transmitted.
- The "S_Error" LED on the front of the controller is on.

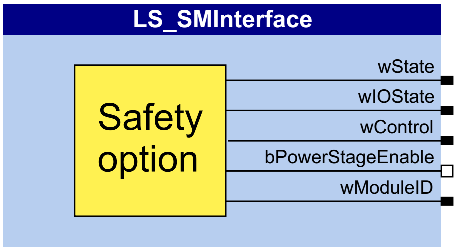
The required error acknowledgement (AIE) is possible via the safety bus and with SO30 via the error acknowledgement input (plug X62).

Related topics:

▸ Transferring the control information to the application (🕮 455)

13.4 Internal interfaces | system block "LS_SMInterface"

The system block **LS_SMInterface** is the interface to the integrated safety system in the function block editor of the »Engineer«.



Output / Data type	Value/meaning	
wState / WORD	Bit coded status information of the drive-based safety ▸ Status information (📖 453)	
wIOState / WORD	Bit coded I/O information of the drive-based safety ▸ I/O status information (📖 454)	
wControl / WORD	Bit coded control information of the drive-based safety ▸ Control information (📖 454)	
bPowerStageEnable / BOOL	Status signal "Inverter enable"	
	TRUE	Inverter is enabled by the safety system.
wModuleID / WORD	ID of the safety system in the controller	
	0	No safety system available
	1	Safety option 10 (SO10):
	2	Reserved
	3	Safety option 20 (SO20):
	4	Safety option 30 (SO30):

13.4.1 Status information

The drive-based safety system transmits information about the status of the requested or active safety functions with the bit coded status signal *wState*.

Bit	Name	Meaning
0	STO	"Safe torque off (STO)" function is active. • The drive is safely switched to torqueless operation.
3	EC_STO	Error stop category 0: "Safe torque off (STO)" function is active.
4	EC_SS1	Error stop category 1: "Safe stop 1 (SS1)" function is active.
14	Fault active	Integrated safety system in error status (trouble or warning).
Bits not listed are reserved for future extensions!		

[13-1] Bit coding of the status signal *wState*

13.4.2 I/O status information

The bit-coded *wIOState* status signal serves to transfer the status the safe inputs and the safe output:

Bit	Name	Meaning
0	SD-In1	Sensor input 1 in ON state.
1	SD-In2	Sensor input 2 in ON state.
5	AIS	Restart acknowledgement via terminal effected (negative edge: 1↘0).
6	AIE	Error acknowledgement via terminal effected (negative edge: 1↘0).
8	PS_AIS	Restart acknowledgement via safety bus effected (positive edge: 0↗1)
9	PS_AIE	Error acknowledgement via safety bus effected (positive edge: 0↗1)
Bits not listed are reserved for future extensions!		

[13-2] Bit coding of the *wIOState* status signal

13.4.3 Control information

The bit coded *wControl* control signal serves to transfer information about requested or active safety functions. The application in the controller must evaluate the control signal and carry out the corresponding action.

- It is possible to request/activate several safety functions at the same time.

Bit	Name	Meaning
0	SS1 active	Safe stop 1 (SS1) is active.
2	ES active	Enable switch (ES) for motion functions in special operations is active.
3	OMS	Operation mode selector (OMS) for special operations is requested.
4	SSE active	Emergency stop function (SSE) is active. • At the end of the function, bit 1 (SS1 active) or bit 0 of the status signal *SMI_wState* (STO active) is set according to the emergency stop function parameterised.
5	OMS active	Special operation is active.
Bits not listed are reserved for future extensions!		

[13-3] Bit coding of the *wControl* control signal

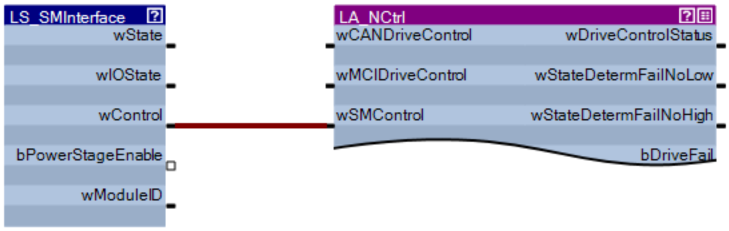
Note!

The application in the controller has to evaluate the *wControl* control signal and execute the corresponding action. This action (e.g.braking to a standstill) has to be implemented by a corresponding application interconnection, which has to be carried out by an application engineer!

See the following subchapter "Transferring the control information to the application". (📖 455)

13.4.4 Transferring the control information to the application

In the simplest case, you only have to go to the I/O level in the FB editor and connect the *wControl* output of the **LS_SMInterface** system block with the *wSMControl* input of the application block:



On the application level, the *wSMControl* input is connected with the **motion control kernel**. The **motion control kernel** evaluates the transmitted control information and activates the required motion sequence (e.g. braking).

Note!

At present, the **Motion Control Kernel** only evaluates bit 0 (SS1). When this safety function is requested, the drive will be decelerated to standstill along the stop ramp set in C02610/3.

Additional functions are in preparation.

Related topics:

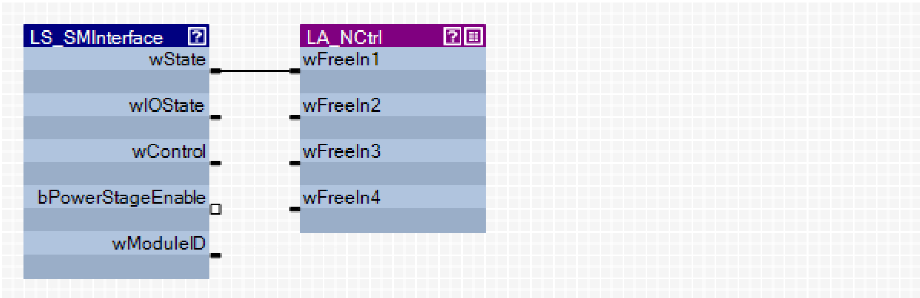
▸ Interface to safety system (📖 303)

13.4.5 Interconnection examples

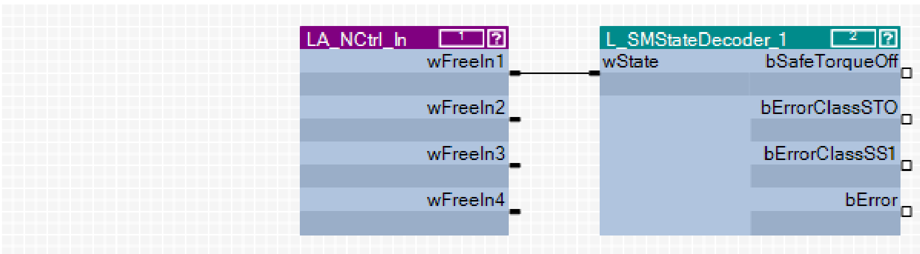
... for decoding the status and control information into single boolean signals.

How to decode the status information into single boolean status signals:

1. Go to the I/O level in the FB editor and connect the *wState* output of the **LS_SMInterface** system block with one of the free inputs *wFreeIn1* ... *wFreeIn4* of the application block.
   - In the following example, the *wState* output is connected with the free *wFreeIn1* input of the **LA_NCtrl** application block on the I/O level.
   - For a better overview, all other connections of the **LA_NCtrl** application block are not shown here.



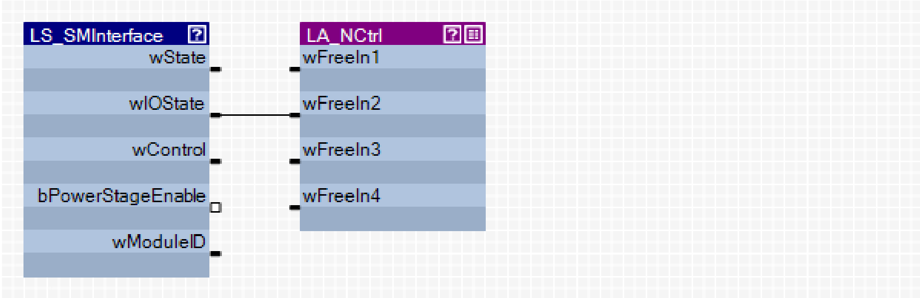
2. Go to the application level and connect the selected free input *wFreeIn* with the *wState* input of the **L_SMStateDecoder_1** function block.
   - The free inputs *wFreeIn1* ... *wFreeIn4* are outputs on the application level.



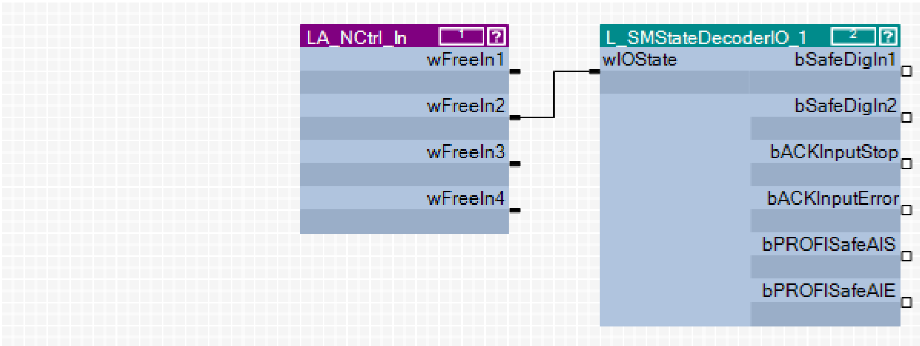
The **L_SMStateDecoder_1** function block decodes the status signal assigned to the *wState* input into single boolean status signals for further use in the FB interconnection.

How to decode the I/O status information into single boolean status signals:

1. Go to the I/O level in the FB editor and connect the *wIOState* output of the **LS_SMInterface** system block with one of the free inputs *wFreeIn1* ... *wFreeIn4* of the application block.
   - In the following example, the *wIOState* output is connected with the free *wFreeIn2* input of the **LA_NCtrl** application block on the I/O level.
   - For a better overview, all other connections of the **LA_NCtrl** application block are not shown here.



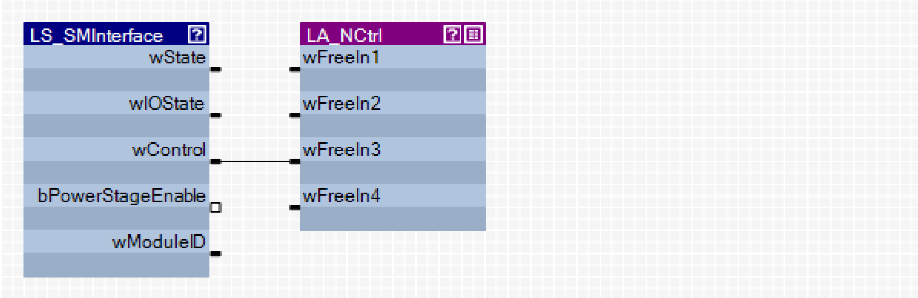
2. Go to the application level and connect the selected free input *wFreeIn* with the *wIOState* input of the **L_SMStateDecoderIO_1** function block.
   - The free inputs *wFreeIn1* ... *wFreeIn4* are outputs on the application level.



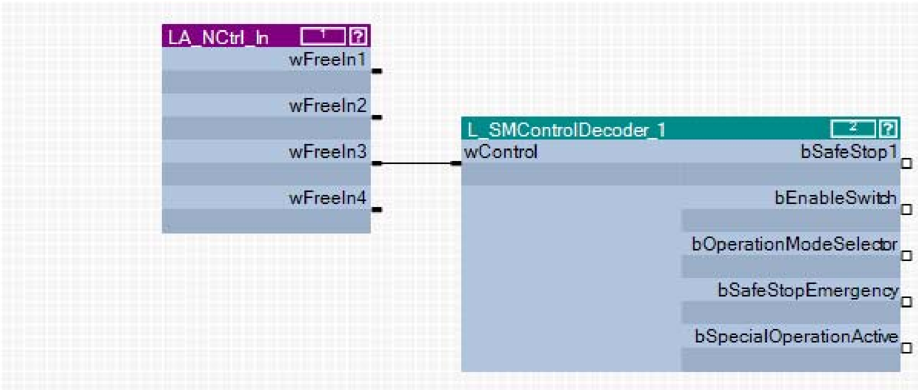
The **L_SMStateDecoderIO_1** function block decodes the status signal assigned to the *wIOState* input into single boolean status signals for further use in the FB interconnection.

How to decode the control information into single boolean control signals:

1. Go to the I/O level in the FB editor and connect the *wControl* output of the **LS_SMInterface** system block with one of the free inputs *wFreeIn1* ... *wFreeIn4* of the application block.
   - In the following example, the *wControl* output is connected with the free *wFreeIn3* input of the **LA_NCtrl** application block on the I/O level.
   - For a better overview, all other connections of the **LA_NCtrl** application block are not shown here.



2. Go to the application level and connect the selected free input *wFreeIn*, which is an output on this level, with the *wControl* input of the **L_SMControlDecoder_1** function block.
   - The free inputs *wFreeIn1* ... *wFreeIn4* are outputs on the application level.



The **L_SMControlDecoder_1** function block decodes the control signal assigned to the *wControl* input into single boolean control signals for further use in the FB interconnection.

14 Parameter change-over

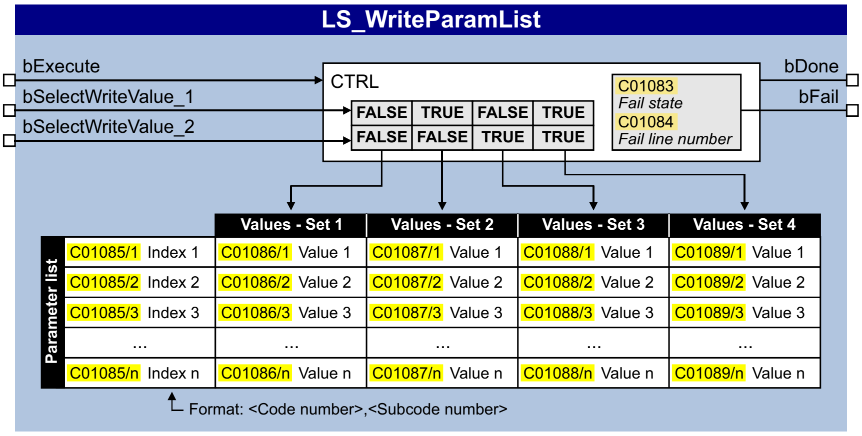
This function extension is available from version 02.00.00!

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.

14.1 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": Activate writing of the parameter list	
bSelectWriteValue_1 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)

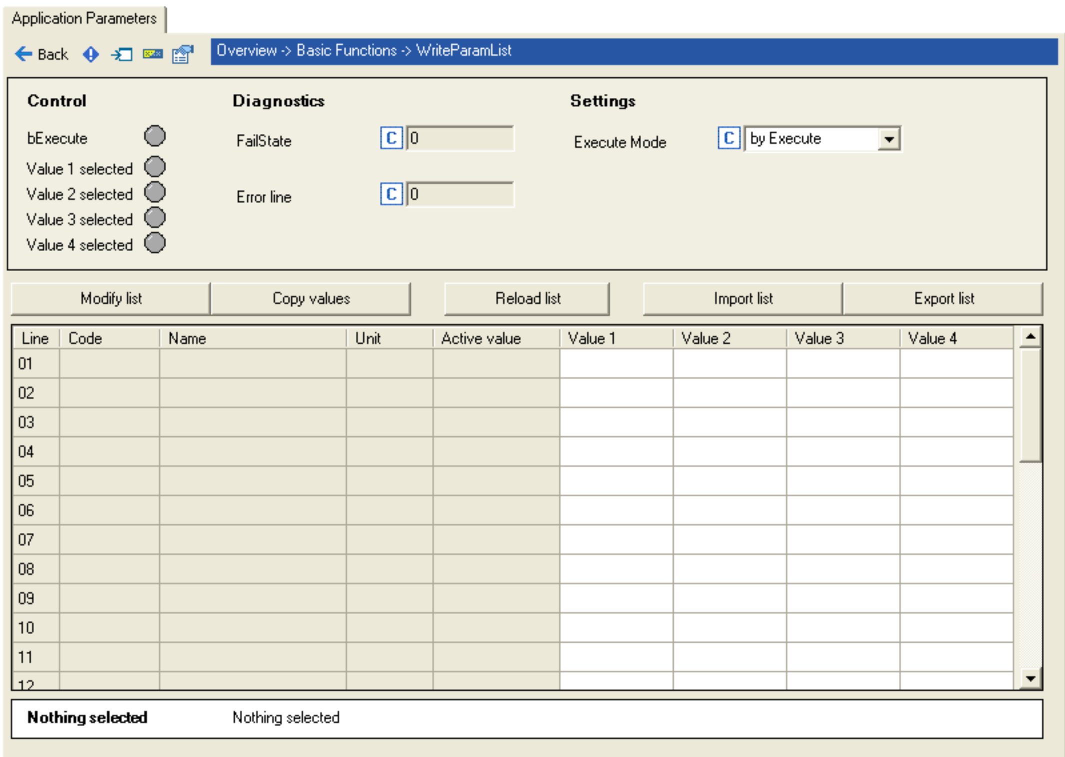
Outputs

Identifier Data type	Value/meaning	
bDone BOOL	"Writing of the parameter list completed" status signal • 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: 1. There is no active writing of the parameter list. 2. Writing of the parameter list has not been completed yet. 3. An error has occurred (if *bFail* = TRUE).
bFail BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • For details see display parameter C01083.

14.2 Configuring the list using the »Engineer« parameterisation dialog

Proceed as follows to open the dialog for parameterising the parameter change-over:

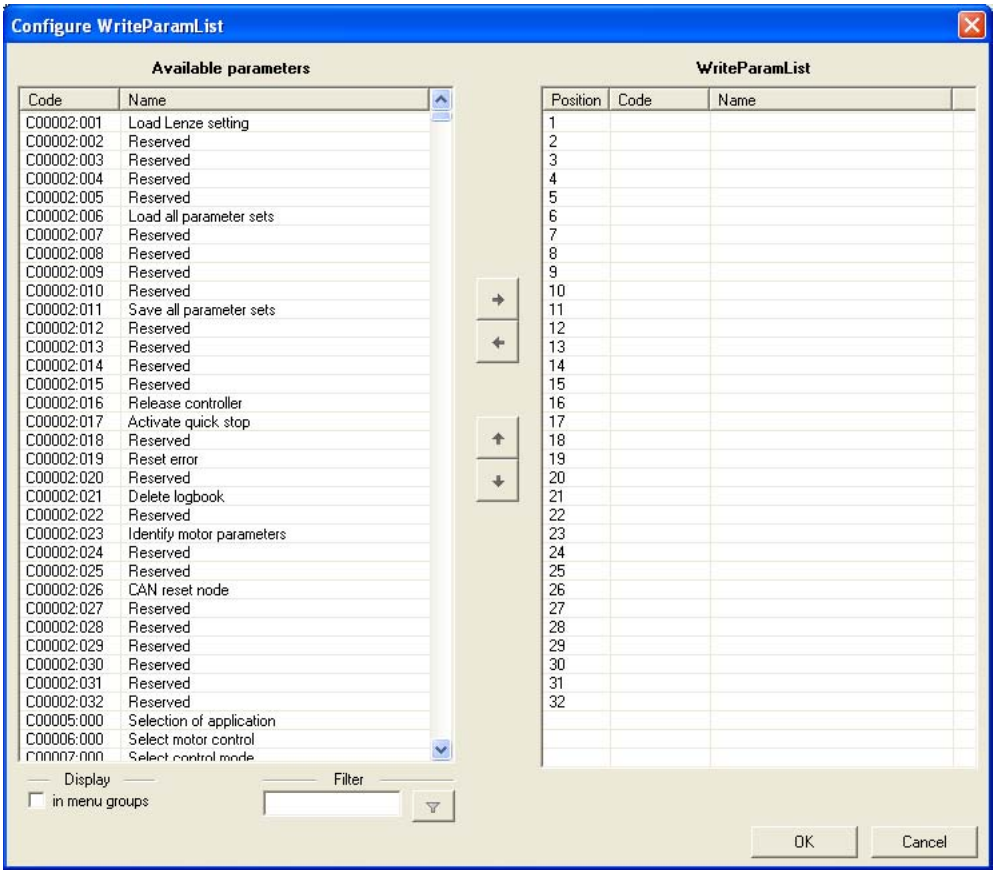
1. »Engineer« Go to the *Project view* and select the 8400 protec StateLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
3. Go to the *Overview* dialog level and click the "basic functions" button.
4. Go to the *Overview* → *Basic functions* dialog box and click the **Parameter change-over** button.



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**).

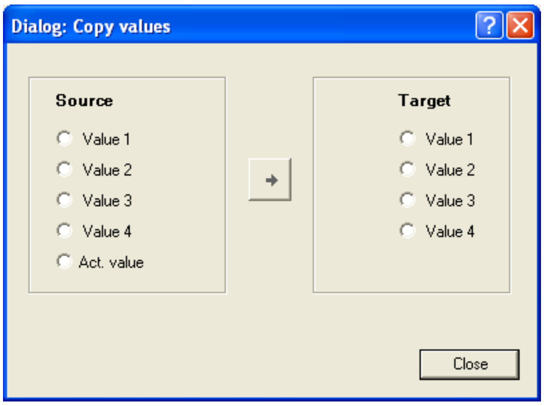
If you place the cursor in an input field, the permitted value range for the corresponding parameter is shown under the table.

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.

14.3 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 when being entered.

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:

14.4 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)

14.5 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 case of an error, corresponding error messages are output.

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 a fault

... 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.

15 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 V07.00.00

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 behind the name.

General information on parameter setting can be found in the chapter "Introduction: Parameterising the controller". (📖 18)

For general information on how to read and change parameters, please see the online documentation for the »Engineer«.

15.1 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.

15.1.1 Data type

The following data types are available for parameters:

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 of printable characters

15.1.2 Parameters with read-only access

Parameters for which the "write access" attribute has not been set can only be read and not be changed by the user.

Description structure

Parameter | Name: **Cxxxxx | ________**				Data type: ______ Index: ______
Description				
Display range **(min. value | unit | max. value)**				
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1				

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	

15.1.3 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«.

15.1.3.1 Parameters with setting range

Description structure

Parameter | Name: **Cxxxxx | ______**				Data type: ______ 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:

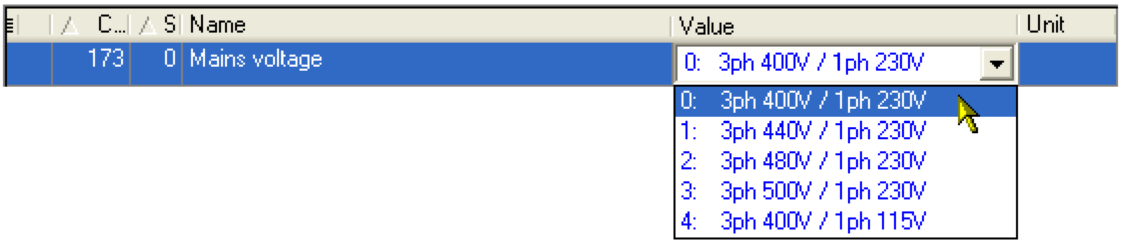
15.1.3.2 Parameters with selection list

Description structure

Parameter | Name: **Cxxxxx | ______**		Data type: ______ 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:



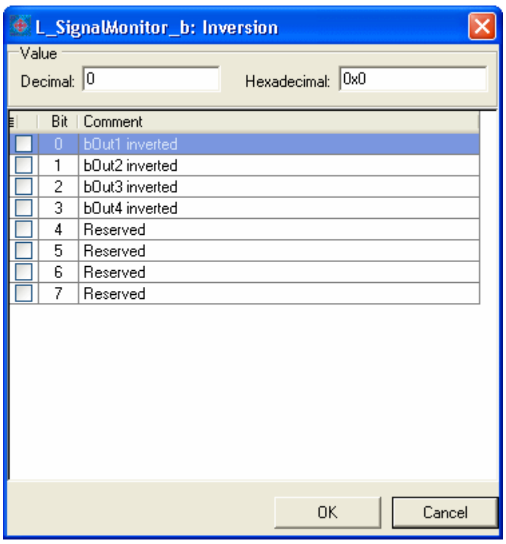
15.1.3.3 Parameters with bit-coded setting

Description structure

Parameter | Name: **Cxxxxx | ______________**		Data type: _______ 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:



15.1.3.4 Parameters with subcodes

Description structure

Parameter | Name: **Cxxxxx | ___________**			Data type: _______ 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	%

15.1.4 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. • 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 • 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".

$$\frac{\text{Read value (via bus system)}}{\text{Scaling factor}} = \text{Indicated value (Engineer)}$$

[15-1] 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".

$$\text{Value to be written (via bus system)} = \text{Value to be set} \cdot \text{Scaling factor}$$

[15-2] 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.

15.2 Parameter list

This chapter lists all parameters of the operating system in numerically ascending order.

Note!

The parameter descriptions are based on software version V07.00.00.

C00002

Parameter | Name: **C00002 | Device command**		Data type: UNSIGNED_8 Index: 24573_d = $5FFD_h$
Note: • Before switching off the supply voltage after a device command has been executed, check the successful execution of the device command via the status display in 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 • All parameters are reset to the Lenze setting. • 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 • All parameter sets are loaded by the memory module. • 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 • 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 "1" ≡ Enable controller "0" ≡ Inhibit controller
C00002/17	0: Off / ready	Activate quick stop "1" ≡ Activate quick stop "0" ≡ Deactivate quick stop
C00002/18	0: Off / ready	Reserved
C00002/19	0: Off / ready	Reset error • 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.
C00002/20	0: Off / ready	Reserved
C00002/21	0: Off / ready	Delete logbook • All entries in the logbook of the controller are deleted. • In the logbook, information on the error history is saved.
C00002/22	0: Off / ready	Reserved
C00002/23	0: Off / ready	Identify motor parameter (ASM) • This device command serves to carry out automatic motor parameter identification. • The device command is only executed when the drive controller is in the "SwitchedOn" status. • In order to identify the motor parameters, the controller must be enabled after this device command. ▶ Automatic motor parameter identification
C00002/24	0: Off / ready	Reserved
C00002/25	0: Off / ready	Reserved
C00002/26	0: Off / ready	CAN reset node • Reinitialise "CAN on board" interface. • Required when changing the baud rate, node address, or identifiers. ▶ "CAN on board" system bus
C00002/27	0: Off / ready	Device search function • From version 06.00.00 • This device command serves to optically locate a controller connected online (e.g. for maintenance work). ▶ Device search function
C00002/28	0: Off / ready	Check MasterPin • From version 06.00.00 ▶ Unlocking the controller with a MasterPin
C00002/29	0: Off / ready	Set binding ID • From version 06.00.00 ▶ Device personalisation
C00002/30	0: Off / ready	Delete binding ID • From version 06.00.00 ▶ Device personalisation
C00002/31	0: Off / ready	Set password • From version 06.00.00 ▶ Password protection

Parameter | Name: C00002 | Device command		Data type: UNSIGNED_8 Index: 24573_d = $5FFD_h$
C00002/32	0: Off / ready	Check password • From version 06.00.00 ▸ Password protection
C00002/33	0: Off / ready	Delete password • From version 06.00.00 ▸ Password protection
☑ 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.
3	Time-out	Device command could not be processed in the defined time (timeout).
4	System fault	
5	Command server assigned	
6	Controller inhibit required	
10	Memory module binding error	
11	Password too short	
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.
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**
4	SLVC: Vector control	This control type is used for sensorless vector control of an asynchronous motor. • The control type requires motor parameters to be set as exactly as possible! ▸ 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. • For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered. ▸ 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. • The control type requires a speed feedback via an encoder mounted to the motor! • For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered. ▸ V/f control
8	VFCplus: V/f quadr	This control type is used for speed control of an asynchronous motor via a square-law V/f characteristic. • For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered. ▸ 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. • The control type requires a speed feedback via an encoder mounted to the motor! • For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered. ▸ 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. • For setting the V/f characteristic, only the rated frequency (C00089) and the rated voltage (C00090) of the motor have to be entered. ▸ V/f characteristic control
☑ 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
  • "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". ▶ "CAN on board" system bus	
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

	"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
DI5	-	-	-	-
DI6	-	-	-	-
A1U	Main speed setpoint (10 V ≡ 100 % reference speed)			

"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 message	CW rotation quick stop Selection: Switch-off position 1		CW rotation quick stop
DI4	Change of direction of rotation	CCW rotation quick stop 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)			

C00008

Parameter | Name: **C00008 | Original application|control source**		Data type: UNSIGNED_16 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. I/O level: Interconnection has been changed.
10	Free|Terminal0	Application: Interconnection has been changed. I/O level: "Terminals 0" control mode
12	Free|Terminal2	Application: Interconnection has been changed. I/O level: "Terminal 2" control mode
14	Free|Terminal11	Application: Interconnection has been changed. I/O level: "Terminal 11" control mode
16	Free|Terminal 16	Application: Interconnection has been changed. I/O level: "Terminal 16" control mode
20	Free|Keypad	Application: Interconnection has been changed. I/O level: "Keypad" control mode
21	Free|PC	Application: Interconnection has been changed. I/O level: "PC" control mode
30	Free|CAN	Application: Interconnection has been changed. I/O level: "CAN" control mode
40	Free|MCI	Application: Interconnection has been changed. I/O level: "MCI" control mode
1000	Speed|Free	Application: Actuating drive speed I/O level: Interconnection has been changed.
1010	Speed|Terminal0	Application: Actuating drive speed I/O level: "Terminals 0" control mode
1012	Speed|Terminal2	Application: Actuating drive speed I/O level: "Terminal 2" control mode
1014	Speed|Terminal11	Application: Actuating drive speed I/O level: "Terminal 11" control mode
1016	Speed|Terminal16	Application: Actuating drive speed I/O level: "Terminal 16" control mode

Parameter | Name: **C00008 | Original application|control source**		Data type: UNSIGNED_16 Index: 24567_d = $5FF7_h$
1020	Speed|Keypad	Application: Actuating drive speed I/O level: "Keypad" control mode
1021	Speed|PC	Application: Actuating drive speed I/O level: "PC" control mode
1030	Speed|CAN	Application: Actuating drive speed I/O level: "CAN" control mode
1040	Speed|MCI	Application: Actuating drive speed I/O level: "MCI" control mode
3000	SwitchPos|Free	Application: Switch-off positioning I/O level: Interconnection has been changed.
3010	SwitchPos|Terminal0	Application: Switch-off positioning I/O level: "Terminals 0" control mode
3012	SwitchPos|Terminal 2	Application: Switch-off positioning I/O level: "Terminal 2" control mode
3014	SwitchPos|Terminal 11	Application: Switch-off positioning I/O level: "Terminal 11" control mode
3016	SwitchPos|Terminal 16	Application: Switch-off positioning I/O level: "Terminal 16" control mode
3020	SwitchPos|Keypad	Application: Switch-off positioning I/O level: "Keypad" control mode
3021	SwitchPos|PC	Application: Switch-off positioning I/O level: "PC" control mode
3030	SwitchPos|CAN	Application: Switch-off positioning I/O level: "CAN" control mode
3040	SwitchPos|MCI	Application: Switch-off positioning 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 = $5FF5_h$
From version 02.00.00 ▸ 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 • 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 • 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) • 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. • 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) • This may increase the starting torque. • 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): 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			

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 | Comparison value N_Act**			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 input ▶ Analog terminals			
Setting range (min. value | unit | max. value)			
-199.99	%	199.99	
Subcodes	**Lenze setting**		**Info**
C00026/1	0.00 %		AIN1: 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 input ▶ Analog terminals			
Setting range (min. value | unit | max. value)			
-100.0000		100.0000	
Subcodes	**Lenze setting**		**Info**
C00027/1	1.0000		AIN1: 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 input ▶ Analog terminals			
Display range (min. value | unit | max. value)			
-10.00	V	10.00	
Subcodes			**Info**
C00028/1			AIN1: 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 input • When the analog input has been configured for current measurement (C00034/1 = 1 or 2). • When C00034/1 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
☑ 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
☑ 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 input for current or voltage measurement ▶ Analog terminals		
Selection list		**Info**
0	0...10 V	Input signal is the voltage signal -10 V ... +10 V • -10 V ... +10 V ≡ -100 % ... +100 %
1	0...20 mA	Input signal is the current signal 0 mA ... 20 mA • 0 mA ... 20 mA ≡ 0 % ... +100 %
2	4...20 mA	Input signal is the current signal 4 mA ... 20 mA • 4 mA ... 20 mA ≡ 0 % ... +100 % • The current loop is monitored for open circuit (I < 4 mA) by the device.
Subcodes	**Lenze setting**	**Info**
C00034/1	0: 0...10 V	AIN1: Config.
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C00036

Parameter | Name: **C00036 | DCB: 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: **C00039 | Fixed setpoint x (L_NSet_1 n-Fix)**			Data type: INTEGER_16 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: **C00050 | MCTRL: Speed setpoint**			Data type: INTEGER_32 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: **C00051 | MCTRL: Actual speed value**			Data type: INTEGER_32 Index: 24524_d = $5FCC_h$
Display of the actual speed value of the motor shaft **Note:** 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). 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			

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 • Only in case of sensorless vector control (SLVC).
C00056/2			Actual torque value • 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 • 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			

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			

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 • If the value displayed here exceeds the threshold set in C00123, the error message "OC5: Device overload (Ixt)" is output and the "Fault" error response is returned.			
Display range (min. value | unit | max. value)			
0.00	%	250.00	
Subcodes			**Info**
C00064/1			Device utilisation (Ixt) • Maximum value of the pulse utilisation (C00064/2) and permanent utilisation (C00064/3).
C00064/2			Device utilisation (Ixt) 15s • Pulse utilisation over the last 15 seconds (only for loads >160 %).
C00064/3			Device utilisation (Ixt) 3 min • Permanent utilisation over the last 3 minutes.
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100			

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 • If the value displayed here exceeds the threshold set in 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
☑ 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
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 10			

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 • 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 certain inverter functions (parameter identification, flying restart circuit) • 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 certain inverter functions (parameter identification, flying restart circuit) • 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			

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: C00081 | Rated motor power			Data type: UNSIGNED_16 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. **Note:** It is mandatory to give the rated motor power for the sensorless vector control (SLVC).			
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: C00082 | Motor rotor resistance			Data type: UNSIGNED_32 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: C00083 | Motor rotor time constant			Data type: UNSIGNED_16 Index: 24492_d = $5FAC_h$
Display of the rotor time constant of the motor • 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: C00084 | Motor stator resistance			Data type: UNSIGNED_32 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: **C00085 | Motor stator leakage inductance**			Data type: UNSIGNED_16 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).			
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).			
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	1000	**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 • 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	

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 • 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 • 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			

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 • 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 | DCB: Hold time**			Data type: UNSIGNED_32 Index: 24468_d = $5F94_h$
Maximum hold time of the manual DC injection brake • 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 | DIx: Polarity**			Data type: UNSIGNED_16 Index: 24461_d = $5F8D_h$
Polarity of the digital inputs • The polarity of each digital input of the device can be inverted via this bit field. ▶ Digital 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 ☐	DI1 inverted		Inversion of digital input 1
Bit 1 ☐	DI2 inverted		Inversion of digital input 2
Bit 2 ☐	DI3 inverted		Inversion of digital input 3
Bit 3 ☐	DI4 inverted		Inversion of digital input 4
Bit 4 ☐	DI5 inverted		Inversion of digital input 5
Bit 5 ☐	DI6 inverted		Inversion of digital input 6
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 ☐	RFR inverted		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 & 5/6: Fct.**		Data type: UNSIGNED_8 Index: 24460_d = $5F8C_h$
Function assignment of the digital terminals DI1/2 and DI5/6 ▶ Digital terminals: Function assignment		
Selection list		**Info**
0	DI1(5)=In / DI2(6)=In	DI1/5 = digital input DI2/6 = digital input
1	DI1(5)=FreqIn / DI2(6)=In	DI1/5 = 1-track frequency input DI2/6 = digital input
2	DI1(5)&DI2(6)=FreqIn (2-track)	DI1/5 and DI2/6 = 2-track frequency input
3	DI1(5)=FreqIn / DI2(6)=direction	DI1/5 = 1-track frequency input DI2/6 = specification of direction
4	DI1(5)=CountIn / DI2(6)=In	DI1/5 = counter input DI2/6 = digital input
Subcodes	**Lenze setting**	**Info**
C00115/1	0: DI1(5)=In / DI2(6)=In	Function assignment of DI1 and DI2
C00115/2	0: DI1(5)=In / DI2(6)=In	Function assignment of DI5 and DI6
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C00116

Parameter | Name: **C00116 | DI 3/4 DO 1/2: Fct.**		Data type: UNSIGNED_8 Index: 24459_d = $5F8B_h$
Function assignment of the digital terminals DI3/DO1 and DI4/DO2 • Pin 4 of the socket X42 can either be used as digital input DI3 or as digital output DO1. • Pin 2 of the socket X42 can either be used as digital input DI4 or as digital output DO2. ▶ Digital terminals: Reconfiguring DI3(4) to digital output DO1(2)		
Selection list (Lenze setting printed in bold)		
0	**DI3 DI4 active; DO1 DO2 inactive**	
1	DO1 DI4 active; DI3 DO2 inactive	
2	DI3 DO2 active; DO1 DI4 inactive	
3	DO1 DO2 active; DI3 DI4 inactive	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C00118

Parameter | Name: **C00118 | DOx inversion**			Data type: UNSIGNED_8 Index: 24457_d = $5F89_h$
Polarity of the digital outputs • The polarity of each digital output of the device can be inverted via this bit field. ▸ Digital 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 ☐	Reserved		
Bit 1 ☐	DO1 inverted		Inversion of digital output 1
Bit 2 ☐	DO2 inverted		Inversion of digital output 2
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			

C00120

Parameter | Name: **C00120 | Motor overload threshold (I²xt)**			Data type: INTEGER_16 Index: 24455_d = $5F87_h$
Operating threshold for the "OC6: Motor overload (I²xt)" error message • The response for reaching the threshold can be selected in C00606. • 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			

C00122

Parameter | Name: **C00122 | Initial value motor overload (I²xt)**			Data type: UNSIGNED_16 Index: 24453_d = $5F85_h$
From version 07.00.00 The thermal motor load displayed in C00066 is pre-initialised with the value set here when the device is connected to the mains. • If "100.00 %" is set, the last value at switching off the device is used for the initialisation. • Recommended setting for operation according to UL: 50.00 %			
Setting range (min. value | unit | max. value)			
0.00	%	100.00	
Subcodes	**Lenze setting**		**Info**
C00122/1	0.00 %		Initial value motor overload (I²xt)
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☑ MOT Scaling factor: 100			

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 • The current device utilisation is displayed in C00064.				
Setting range (min. value | unit | max. value)			**Lenze setting**	
0.00	%	150.00	**149.99 %**	
☑ 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 • 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 • 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 • 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 | Ramp rounding main setpoint**		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 • 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 • Control word of the MCI communication interface (communication module)
C00136/2			CAN control word • 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: **C00138 | Internal control signals**			Data type: UNSIGNED_16 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: **C00142 | Auto-start option**				Data type: UNSIGNED_8 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. ⚠ **Danger!** In the Lenze setting, the autostart option "Inhibit at power-on" is deactivated, i. e. the motor can directly start up if the controller is enabled after mains connection! ▶ Automatic restart after mains connection/fault...				
Setting range (min. hex value | max. hex value)			**Lenze setting**	
0x00		0xFF	**0x10** (decimal: 16)	
Value is bit-coded: (☑ = bit set)			**Info**	
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		From version 06.00.00	
Bit 5 ☐	Reserved			
Bit 6 ☐	Reserved			
Bit 7 ☐	Reserved			
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT				

C00144

Parameter | Name: **C00144 | Switching freq. reduct. (Temp.)**			Data type: UNSIGNED_8 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: **C00148 | Config. group error**			Data type: UNSIGNED_16 Index: 24427_d = $5F6B_h$
From version 02.00.00 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		From version 06.00.00
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 ☐	Reserved		
Bit 15 ☐	Use 16BitFailNo.		From version 06.00.00 If this bit is set, the short 16-bit error number (*wStateDetermFailNoShort*) is also provided at the *wStateDetermFailNoLow* output of the SB LS_DriveInterface. • In this case, the *wStateDetermFailNoHigh* output is "0". • 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		Fault
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 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: **C00158 | Cause of controller inhibit**		Data type: UNSIGNED_16 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: C00159 | Cause of quick stop QSP			Data type: UNSIGNED_16 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: C00160 | Status determining error (16bit)			Data type: UNSIGNED_16 Index: 24415_d = $5F5F_h$
From version 06.00.00 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
☑ 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	
26	CAN2 input bit 3	
27	CAN2 input bit 4	

Parameter | Name: **C00163 | Logbook - binary elements**		Data type: UNSIGNED_16 Index: 24412_d = $5F5C_h$
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	
70	MCI word1 input bit15	
71	MCI word2 input bit0	

Parameter | Name: **C00163 | Logbook - binary elements**		Data type: UNSIGNED_16 Index: 24412_d = $5F5C_h$
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 active	
94	Pulse inhibit active	
95	Controller inhibit 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	
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	
35	Current motor current	
36	Current motor voltage	
37	Current motor frequency	
38	Effective speed setpoint	
39	Device utilisation	

Parameter | Name: **C00164 | Logbook - analog elements**		Data type: UNSIGNED_16 Index: 24411_d = $5F5B_h$
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	

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 • Response to the status determining error
C00166/2	Subj. - status det. error • Subject area of the status determining error
C00166/3	Mess. - status det. error • Textual message of the status determining error
C00166/4	Resp. to curr. error • Response of the currently pending error
C00166/5	Subj. - curr. error • Subject area of the currently pending error
C00166/6	Mess. - curr. error • Textual message of the currently pending error
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT	

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 of the logbook 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		

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 = $5F4A_h$
From version 06.00.00 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 | 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 • 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 | 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 = $5F38_h$
From version 06.00.00 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		

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	

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.
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT	

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	Type: Display module
C00203/10	Type: Wiring module
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT	

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	Serial no.: Display module
C00204/10	Serial no.: Wiring module
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT	

C00205

Parameter | Name: **C00205 | Info**	Data type: VISIBLE_STRING 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: **C00206 | Production date**	Data type: VISIBLE_STRING 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: **C00210 | HW version**	Data type: VISIBLE_STRING 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: **C00220 | Accel. time - add. setpoint**			Data type: UNSIGNED_32 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: **C00221 | Decel. time - add. setpoint**			Data type: UNSIGNED_32 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: **C00222 | L_PCTRL_1: Vp**			Data type: INTEGER_16 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: **C00223 | L_PCTRL_1: Tn**			Data type: UNSIGNED_16 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: **C00224 | L_PCTRL_1: Kd**			Data type: UNSIGNED_16 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 • The actual value *nAct_a* remains unchanged for further processing	
1	On		Root function active • 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 • 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: ActEqSet window**			Data type: INTEGER_16 Index: 24328_d = $5F08_h$
From version 06.00.00 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			

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!	

C00273

Parameter | Name: **C00273 | Moment of inertia**			Data type: UNSIGNED_32 Index: 24302_d = $5EEE_h$
Moment of inertia for setpoint feedforward control at 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			

C00275

Parameter | Name: **C00275 | Filter setpoint feedforward control**			Data type: UNSIGNED_16 Index: 24300_d = $5EEC_h$
Filter time of the setpoint feedforward control at sensorless vector control (SLVC) • 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			

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!	

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).
▶ "CAN on board" system bus			

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).
▸ "CAN on board" system bus		

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 = $5EBB_h$
From version 06.00.00 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). ▸ "CAN on board" system bus		
Setting range (min. value | unit | max. value)		
0	ms	6500
Subcodes	**Lenze setting**	**Info**
C00324/1	0 ms	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		

C00341

Parameter | Name: **C00341 | CAN management - error configuration**			Data type: UNSIGNED_16 Index: 24234_d = $5EAA_h$
From version 02.00.00 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 = $5EA9_h$
From version 02.00.00 Configuration defining the events that lead to a decoupling of the process data words. "CAN on board" system bus: Configuring the 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 = $5EA8_h$
From version 02.00.00 Definition of the value the process data words are to have in the decoupled state. "CAN on board" system bus: Configuring the 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 = $5EA7_h$
From version 02.00.00 Definition of the value the process data words are to have in the decoupled state. "CAN on board" system bus: Configuring the 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 = $5EA6_h$
		"CAN on board" system 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		

C00347

Parameter | Name: **C00347 | CAN status HeartBeat producer**		Data type: UNSIGNED_8 Index: 24228_d = $5EA4_h$
		"CAN on board" system bus: Heartbeat protocol
Selection list		
0	Boot-up	
4	Stopped	
5	Operational	
127	Pre-Operat.	
250	Failed	
255	NoResponse	
Subcodes		**Info**
C00347/1		Status of node 1 ... 7
C00347/...		
C00347/7		
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C00349

Parameter | Name: **C00349 | Setting - DIP switch SW1**			Data type: UNSIGNED_16 Index: 24226_d = $5EA2_h$
DIP switch setting during last mains power-on "CAN on board" system bus			
Display area (min. hex value | max. hex value)			
0x0000		0xFFFF	
Value is bit-coded:			
Bit 0	Address 1		
Bit 1	Address 2		
Bit 2	Address 4		
Bit 3	Address 8		
Bit 4	Address 16		
Bit 5	Address 32		
Bit 6	Address 64		
Bit 7	CAN baud rate 1		
Bit 8	CAN baud rate 2		
Bit 9	CAN baud rate 4		
Bit 10	Reserved		
Bit 11	Reserved		
Bit 12	Reserved		
Bit 13	Reserved		
Bit 14	Reserved		
Bit 15	CAN 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 • The node address can only be parameterised if the node address "0" is set via the DIP switches. • A change in the node address will not be effective until a CAN Reset Node is performed. "CAN on board" system bus				
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. "CAN on board" system bus		
Selection list (Lenze setting printed in bold)		
0	**500 kbps**	
1	250 kbps	
2	125 kbps	
3	50 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. "CAN on board" system bus		
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 "CAN on board" system bus		
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 • A change in the COBID will not be effective until a CAN reset node is performed. "CAN on board" system bus			
Setting range (min. hex value | max. hex value)			
0x00000000		0xFFFFFFFF	
Value is bit-coded:			**Info**
Bit 0	COBID Bit0		• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • 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 "CAN on board" system bus			
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 ▸ "CAN on board" system bus			
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" to "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) • If a value unequal to "0" is entered, the RPDO is not expected before the set time has expired. • If the RPDO is not received within the expected time, the response set in C00593/1...3 will be triggered. ▸ "CAN on board" system bus			
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 ▸ "CAN on board" system bus			
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 = $5E98_h$
	"CAN on board" system bus

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 = $5E97_h$
	"CAN on board" system bus

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 = $5E93_h$
	▶ "CAN on board" system 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

C00366

Parameter | Name: **C00366 | Number of CAN SDO channels**	Data type: UNSIGNED_8 Index: 24209_d = $5E91_h$

Change in function from version 02.02.00
Selection of the number of active parameter data channels
- Up to and including version 02.01.00, parameter data channels 1 and 2 are activated in the Lenze setting.
- From version 02.02.00, only parameter data channel 1 is activated according to CANopen in the Lenze setting. To activate both parameter data channels according to the previous behaviour, select "2 SDO Lenze".

▶ "CAN on board" system bus

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. • Mapping of the CANopen object I-1005 (see DS301 V4.02). "CAN on board" system bus			
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). ▶ "CAN on board" system bus			
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). ▶ "CAN on board" system bus			
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 = $5E8D_h$
From version 02.00.00 ▸ "CAN on board" system bus			
Display range (min. value | unit | max. value)			
-1638	µs	1638	
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 = $5E8B_h$
▸ "CAN on board" system bus			
Display range (min. value | unit | max. value)			
0		255	
Subcodes			**Info**
C00372/1			Tx_Error
C00372/2			Rx_Error
C00372/3			Tx_Overflow
C00372/4			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). • The heartbeat telegram is sent automatically as soon as a time > 0 ms is set. • Mapping of the CANopen object I-1017 (see DS301 V4.02). ▸ "CAN on board" system bus			
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. ▶ "CAN on board" system bus: Heartbeat protocol			
Setting range (min. value | unit | max. value)			
0		127	
Subcodes	**Lenze setting**		**Info**
C00385/1	0		CAN node address 1 ... 7
C00385/...			
C00385/7			
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C00386

Parameter | Name: **C00386 | HeartBeat-ConsumerTime**			Data type: UNSIGNED_16 Index: 24189_d = $5E7D_h$
The subcodes represent the nodes to be monitored by heartbeat. ▶ "CAN on board" system bus: Heartbeat protocol			
Setting range (min. value | unit | max. value)			
0	ms	60000	
Subcodes	**Lenze setting**		**Info**
C00386/1	0 ms		ConsumerTime HeartBeat Producer 1 ... 7
C00386/...			
C00386/7			
☑ 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		From version 06.00.00 Delay time for status signal *bFirstCycleDone* • 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. ▶ System bus "CAN on board": Port blocks			
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 ▶ System bus "CAN on board": Port blocks		
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 ▸ System bus "CAN on board": Port blocks			
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			

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			
Setting range (min. value | unit | max. value)			
1	Incr./rev.	32768	
Subcodes	**Lenze setting**	**Info**	
C00420/1	128 incr./rev.	Encoder increments at FreqIn12	
C00420/2	128 incr./rev.	Number of encoder increments at FreqIn56	
☑ 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 terminals			
Setting range (min. value | unit | max. value)			
0.000	s	65.000	
Subcodes	**Lenze setting**	**Info**	
C00423/1	0.000 s	DO1 ON delay	
C00423/2	0.000 s	DO1 OFF delay	
C00423/3	0.000 s	DO2 ON delay	
C00423/4	0.000 s	DO2 OFF delay	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000			

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(5) and DI2(6) 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	Encoder sample time FreqIn12	
C00425/2	3: 10 ms	Encoder sample time FreqIn56	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C00440

Parameter | Name: **C00440 | LS_AnalogIn1: PT1 time constant**			Data type: UNSIGNED_16 Index: 24135_d = $5E47_h$
PT1 time constant (S-ramp time) for the analog input Analog terminals			
Setting range (min. value | unit | max. value)			
0	ms	1000	
Subcodes	**Lenze setting**		**Info**
C00440/1	10 ms		PT1 rounding AnalogIn1
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C00443

Parameter | Name: **C00443 | DIx: Level**			Data type: UNSIGNED_16 Index: 24132_d = $5E44_h$
Bit coded display of the level of the digital inputs Digital 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	Reserved		
Bit 7	Reserved		
Bit 8	Reserved		
Bit 9	Reserved		
Bit 10	Reserved		
Bit 11	Manually		
Bit 12	Right		
Bit 13	Left		
Bit 14	Mains		
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 terminals
Display area (min. hex value | max. hex value)			
0x0000		0xFFFF	
Value is bit-coded:			**Info**
Bit 0	DO1		Bit set = HIGH level
Bit 1	DO2		
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			**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(5) and DI2(6) as frequency inputs
Display range (min. value | unit | max. value)			
-32767	Incr/ms	32767	
Subcodes			**Info**
C00445/1			FreqIn12_nOut_v
C00445/2			FreqIn56_nOut_v
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C00446

Parameter | Name: **C00446 | FreqInxx_nOut_a**			Data type: INTEGER_16 Index: 24129_d = $5E41_h$
Display of the frequency input signals which are fed into the application. ▸ Using DI1(5) and DI2(6) as frequency inputs			
Display range (min. value | unit | max. value)			
-199.99	%	199.99	
Subcodes			**Info**
C00446/1			FreqIn12_nOut_a
C00446/2			FreqIn56_nOut_a
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100			

C00447

Parameter | Name: **C00447 | Decoupling DigOut**			Data type: UNSIGNED_16 Index: 24128_d = $5E40_h$
From version 02.00.00 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 = $5E3F_h$
From version 02.00.00 Definition of the value the digital output terminals are to have in the decoupled state. • 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 ☐	Reserved		
Bit 1 ☐	DigOut1_ON		
Bit 2 ☐	DigOut2_ON		
Bit 3 ☐	Reserved		
Bit 4 ☐	Reserved		
Bit 5 ☐	BrakeRelease_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 = $5E3E_h$
From version 06.00.00 ▸ 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!	

C00461

Parameter | Name: C00461 | Remote: Acceleration/deceleration time			Data type: UNSIGNED_32 Index: 24114_d = $5E32_h$
From version 06.00.00			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 = $5E31_h$
From version 06.00.00			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 = $5E30_h$
From version 06.00.00 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 = $5E2F_h$
From version 06.00.00			▸ 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 show 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: Fct. STOP key**		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 response	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 ... bPar16*
C00470/...		
C00470/16		
☑ 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	Active		
...	...		
Bit 15	Active		
Subcodes	**Lenze setting**		**Info**
C00471/1	0		Value for output *wPar1 ... wPar4*
C00471/...			
C00471/4			
☑ 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*
☑ 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 ... nPar4_v*
C00473/...			
C00473/4			
☑ 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 = $5E25_h$
Setting range (min. value | unit | max. value)			
-2147483647	Incr.	2147483647	
Subcodes	**Lenze setting**		**Info**
C00474/1	0 incr.		LS_ParFree: dnPar1_p
C00474/2	0 incr.		LS_ParFree: dnPar2_p
C00474/3	0 incr.		LS_ParFree: dnPar3_p
C00474/4	0 incr.		LS_ParFree: dnPar4_p
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C00480

Parameter | Name: **C00480 | LS_DisFree_b**			Data type: UNSIGNED_8 Index: 24095_d = $5E1F_h$
The LS_DisFree_b SB: Display of the input values			
Display area (min. hex value | max. hex value)			
0x00		0xFF	
Value is bit-coded:			**Info**
Bit 0	bDis1		Signal level input *bDis1* ... *bDis8*
...	...		
Bit 7	bDis8		
☑ 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* ... *wDis4*
C00481/...			
C00481/4			
☑ 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* ... *nDis4_a*
C00482/...			
C00482/4			
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 100			

_ _

C00484

Parameter | Name: **C00484 | Application units: Offset**			Data type: INTEGER_16 Index: 24091_d = $5E1B_h$
From version 06.00.00 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 = $5E1A_h$
From version 06.00.00 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: Text**		Data type: VISIBLE_STRING Index: 24089_d = $5E19_h$
From version 06.00.00 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		

C00487

Parameter | Name: C00487 - Application units			Data type: INTEGER_32 Index: 24088$_d$ = 5E18$_h$
From version 06.00.00 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 • 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 • Selection of edge or level for starting slow-down function 1
C00488/2	0: Level	InputSens.Stop1 • Selection of edge or level for stop function 1
C00488/3	0: Level	InputSens.SlowDown2 • Selection of edge or level for starting slow-down function 2
C00488/4	0: Level	InputSens.Stop2 • Selection of edge or level for stop function 2
C00488/5	0: Level	InputSens.SlowDown3 • Selection of edge or level for starting slow-down function 3
C00488/6	0: Level	InputSens.Stop3 • Selection of edge or level for stop function 3
☑ 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 FreqIn56	Speed sensor signal is fed via the digital DI5 and DI6 inputs
☑ 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	EdgeCounting analysis	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		Encoder filter time FreqIn12
C00497/2	1.0 ms		Encoder filter time FreqIn56
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 10			

C00505

Parameter | Name: **C00505 | Password data**		Data type: VISIBLE_STRING Index: 24070_d = $5E06_h$
From version 06.00.00		
		▶ 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		

C00507

Parameter | Name: **C00507 | Current password protection**			Data type: UNSIGNED_16 Index: 24068_d = $5E04_h$
From version 06.00.00 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. • Format: <code number>,<subcode number> • 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: Festsollwert 1
C00517/10	39.002		C00039/2: Festsollwert 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: Motor overload threshold (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 WU
C00560/2		Switching status - internal fan DU
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C00561

Parameter | Name: **C00561 | Fan failure**		Data type: UNSIGNED_8 Index: 24014_d = $5DCE_h$
Error status display of device fan		
Selection list		
0	No error	
1	Fault	
Subcodes		**Info**
C00561/1		Failure - internal fan WU
C00561/2		Failure - internal fan DU
☑ 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
C00570/2	0: No Reaction	Response if cross current controller is limited
C00570/3	0: No Reaction	Response if torque setpoint is limited • Limitation of the speed controller output, the differential setpoint feedforward control, and the additive torque at sensorless vector control (SLVC).
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 • 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		

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		

C00581

Parameter | Name: **C00581 | Resp. to LS_SetError_x**		Data type: UNSIGNED_8 Index: 23994_d = $5DBA_h$
Selection of the error responses for application error messages • 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
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C00582

Parameter | Name: **C00582 | Resp. to heatsink temp. > shutdown temp. -5°C**		Data type: UNSIGNED_8 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		

C00585

Parameter | Name: **C00585 | Resp. to motor overtemp. PTC**		Data type: UNSIGNED_8 Index: 23990_d = $5DB6_h$
Response to motor overtemperature • The motor temperature is measured by means of a PTC thermistor at connector X21.		
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**		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: **C00588 | Resp. to max. speed at switching freq.**		Data type: UNSIGNED_8 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: **C00590 | Resp. to switching frequency red.**		Data type: UNSIGNED_8 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: **C00592 | Resp. to CAN bus connection**		Data type: UNSIGNED_8 Index: 23983_d = $5DAF_h$
Configuration of monitoring of the CAN interface (group 1)		
		"CAN on board" system bus
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)		
		"CAN on board" system bus
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		

C00597

Parameter | Name: **C00597 | Resp. to motor phase failure**		Data type: UNSIGNED_8 Index: 23978_d = $5DAA_h$
Response to motor phase failure • 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 I/O monitoring**		Data type: UNSIGNED_8 Index: 23977_d = $5DA9_h$
Configuration of analog input monitoring ▸ 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	0: No Reaction	Reserved
C00598/3	5: Warning	Resp. to digital output level
C00598/4	5: Warning	Resp. to IO supply overload
☑ 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 • 100 % ≡ rated inverter current (C00098) • 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 • 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		

_ _

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. • 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. DIG12/56**	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!	

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	0: Not connected	Reserved
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

Parameter | Name: **C00620 | System connection list: 16-bit**		Data type: UNSIGNED_16 Index: 23955_d = $5D93_h$
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
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	Reserved
C00620/39	0: Not connected	Reserved
C00620/40	0: Not connected	Reserved
C00620/41	0: Not connected	Reserved
C00620/42	0: Not connected	Reserved
C00620/43	0: Not connected	Reserved
C00620/44	0: Not connected	Reserved
C00620/45	0: Not connected	Reserved
C00620/46	0: Not connected	Reserved
C00620/47	0: Not connected	Reserved
C00620/48	0: Not connected	Reserved
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	Reserved
C00620/62	0: Not connected	Reserved
C00620/63	0: Not connected	Reserved
C00620/64	0: Not connected	Reserved
C00620/65	0: Not connected	Reserved
C00620/66	0: Not connected	Reserved
C00620/67	0: Not connected	Reserved
C00620/68	0: Not connected	Reserved
C00620/69	0: Not connected	Reserved
C00620/70	0: Not connected	Reserved

Parameter | Name: **C00620 | System connection list: 16-bit**		Data type: UNSIGNED_16 Index: 23955_d = $5D93_h$
C00620/71	0: Not connected	Reserved
C00620/72	0: Not connected	Reserved
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
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
☑ 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 • Selection of the binary output signals to be connected to the binary input signals • 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	0: Not connected	Reserved
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
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

Parameter | Name: **C00621 | System connection list: Bool**		Data type: UNSIGNED_16 Index: 23954_d = $5D92_h$
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
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

Parameter | Name: **C00621 | System connection list: Bool**		Data type: UNSIGNED_16 Index: 23954_d = $5D92_h$
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
C00621/97	0: Not connected	LS_DigitalInput: bCountIn5_Reset
C00621/98	0: Not connected	LS_DigitalInput: bCountIn5_LoadStartValue
C00621/99	0: Not connected	LS_DigitalOutput: bOut2
C00621/100	0: Not connected	Reserved
C00621/101	0: Not connected	Reserved
C00621/102	0: Not connected	Reserved
C00621/103	0: Not connected	Reserved

Parameter | Name: **C00621 | System connection list: Bool**		Data type: UNSIGNED_16 Index: 23954_d = $5D92_h$
C00621/104	0: Not connected	Reserved
C00621/105	0: Not connected	Reserved
C00621/106	0: Not connected	Reserved
C00621/107	0: Not connected	Reserved
C00621/108	0: Not connected	Reserved
C00621/109	0: Not connected	Reserved
C00621/110	1015: LA_nCtrl_bBrakeReleaseOut	LS_DigitalOutput: bBrakeRelease
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	Reserved
C00621/118	0: Not connected	Reserved
C00621/119	0: Not connected	Reserved
C00621/120	0: Not connected	Reserved
C00621/121	0: Not connected	Reserved
C00621/122	0: Not connected	Reserved
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
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	Reserved

Parameter | Name: **C00621 | System connection list: Bool**		Data type: UNSIGNED_16 Index: 23954_d = $5D92_h$
C00621/148	0: Not connected	Reserved
C00621/149	0: Not connected	Reserved
C00621/150	0: Not connected	Reserved
C00621/151	0: Not connected	Reserved
C00621/152	0: Not connected	Reserved
C00621/153	0: Not connected	Reserved
C00621/154	0: Not connected	Reserved
C00621/155	0: Not connected	Reserved
C00621/156	0: Not connected	Reserved
C00621/157	0: Not connected	Reserved
C00621/158	0: Not connected	Reserved
C00621/159	0: Not connected	Reserved
C00621/160	0: Not connected	Reserved
C00621/161	0: Not connected	Reserved
C00621/162	0: Not connected	Reserved
☑ 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 • Selection of the 32-bit output signals for connection with the 32-bit input signals. • The selection list contains all 32-bit output signals which can be assigned to the 32-bit inputs mapped by the subcodes.		
Selection list		
0	Not connected	
1005	LA_NCtrl_dnFreeOut1_p	
1006	LA_NCtrl_dnFreeOut1_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	
17020	DigIn_dnPosIn12_p	
20000	dnPar1_p	
20001	dnPar2_p	
20002	dnPar3_p	
20003	dnPar4_p	
32000	MCTRL_dnMotorPosAct_p	
32001	MCTRL_dnMotorDeltaPosAct_p	
32200	MCK_dnPosTarget_p	
32201	MCK_dnPosSetValue_p	
32202	MCK_dnDeltaPos_p	
32203	MCK_dnMotorRefOffset_p	
32204	MCK_dnPosSet_p	
36080	L_Interpolator_1_dnPhiOut_p	
42005	LA_NCtrl_dnFreeIn1_p	
42006	LA_NCtrl_dnFreeIn2_p	
42205	LA_SwitchPos_dnFreeIn1_p	
42206	LA_SwitchPos_dnFreeIn2_p	
Subcodes	**Lenze setting**	**Info**
C00622/1	0: Not connected	Reserved
C00622/2	0: Not connected	Reserved
C00622/3	0: Not connected	Reserved
C00622/4	0: Not connected	Reserved
C00622/5	0: Not connected	Reserved
C00622/6	0: Not connected	Reserved
C00622/7	0: Not connected	Reserved
C00622/8	0: Not connected	Reserved
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

Parameter | Name: **C00622 | System connection list: Angle**		Data type: UNSIGNED_16 Index: 23953_d = $5D91_h$
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
☑ 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 • 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: Blocking speed1 max
C00632/2	0.00 %		L_NSet_1: Blocking speed2 max
C00632/3	0.00 %		L_NSet_1: Blocking speed3 max
☑ 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 • 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: Blocking speed1 min
C00633/2	0.00 %		L_NSet_1: Blocking speed2 min
C00633/3	0.00 %		L_NSet_1: Blocking speed3 min
☑ 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			

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			

C00680

Parameter | Name: C00680 | L_Compare_1: Fct.		Data type: UNSIGNED_8 Index: 23895_d = $5D57_h$
The L_Compare_1 FB: Comparison operation • 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 • 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			

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 • 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**
C00700/1	20005: LS_ParFix_wDriveCtrl	LA_NCtrl: wCANDriveControl (input for control word from CAN to device control)
C00700/2	20005: LS_ParFix_wDriveCtrl	LA_NCtrl: wMCIDriveControl (input for control word from communication interface to device control)
C00700/3	20012: LS_ParFree_nPar3_a	LA_NCtrl: nTorqueMotLim_a (input for maximum torque in motor mode)
C00700/4	20013: LS_ParFree_nPar4_a	LA_NCtrl: nTorqueGenLim_a (input for maximum torque in generator mode)
C00700/5	0: Not connected	LA_NCtrl: nPIDVpAdapt_a (input for adapting the PID controller gain)
C00700/6	0: Not connected	LA_NCtrl: nPIDActValue_a (input for actual PID controller value)
C00700/7	16000: AIN1_Out	LA_NCtrl: nMainSetValue_a (input for main speed setpoint)
C00700/8	0: Not connected	LA_NCtrl: nAuxSetValue_a (input for additional speed setpoint)
C00700/9	0: Not connected	LA_NCtrl: nGPAnalogSwitchIn1_a (input for analog-value selector analog signal 1)
C00700/10	0: Not connected	LA_NCtrl: nGPAnalogSwitchIn2_a (input for analog-value selector analog signal 2)

Parameter | Name: **C00700 | LA_NCtrl: Analog connection list**		Data type: UNSIGNED_16 Index: 23875_d = $5D43_h$
C00700/11	0: Not connected	LA_NCtrl: nGPArithmetikIn1_a (input for arithmetic function analog signal 1)
C00700/12	0: Not connected	LA_NCtrl: nGPArithmetikIn2_a (input for arithmetic function analog signal 2)
C00700/13	0: Not connected	LA_NCtrl: nGPMulDivIn_a (input for analog signal for multiplication/division)
C00700/14	0: Not connected	LA_NCtrl: nGPCompareIn1_a (input for comparison operation analog signal 1)
C00700/15	0: Not connected	LA_NCtrl: nGPCompareIn2_a (input for comparison operation analog signal 2)
C00700/16	0: Not connected	LA_NCtrl: nVoltageAdd_a (input for additive voltage boost)
C00700/17	0: Not connected	LA_NCtrl: nPIDInfluence_a (input for influence signal of PID controller correcting variable)
C00700/18	0: Not connected	LA_NCtrl: nPIDSetValue_a (input for PID controller setpoint)
C00700/19	0: Not connected	LA_NCtrl: nPWMAngleOffset (input for pulse width modulation phase offset)
C00700/20	0: Not connected	LA_NCtrl: nBoost_a (input for additional setpoint for the motor voltage at speed = 0)
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 • Selection of the binary output signals to be connected to the binary input signals • 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)
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)

Parameter | Name: **C00701 | LA_NCtrl: Digital connection list**		Data type: UNSIGNED_16 Index: 23874_d = $5D42_h$
C00701/20	0: Not connected	LA_NCtrl: bMBRKRelease (control input for manual holding brake release request)
C00701/21	0: Not connected	LA_NCtrl: bMANJogPos (control input for manual jog in positive direction request)
C00701/22	0: Not connected	LA_NCtrl: bMANJogNeg (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)
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)

Parameter | Name: **C00701 | LA_NCtrl: Digital connection list**		Data type: UNSIGNED_16 Index: 23874_d = $5D42_h$
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: 23870_d = $5D3E_h$
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: 23869_d = $5D3D_h$
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			

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 | 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			

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!	

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)
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

Parameter | Name: **C00760 | LA_SwitchPos: Analog connection list**		Data type: UNSIGNED_16 Index: 23815_d = $5D07_h$
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 • Selection of the binary output signals to be connected to the binary input signals • 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)
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)

Parameter | Name: **C00761 | LA_SwitchPos: Digital connection list**		Data type: UNSIGNED_16 Index: 23814_d = $5D06_h$
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)
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

Parameter | Name: **C00761 | LA_SwitchPos: Digital connection list**		Data type: UNSIGNED_16 Index: 23814_d = $5D06_h$
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**	Keep 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. • 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. • 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			

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	bIn3...bIn1=0 0 0	
C00821/2	0: FALSE	bIn3...bIn1=0 0 1	
C00821/3	0: FALSE	bIn3...bIn1=0 1 0	
C00821/4	0: FALSE	bIn3...bIn1=0 1 1	
C00821/5	0: FALSE	bIn3...bIn1=1 0 0	
C00821/6	0: FALSE	bIn3...bIn1=1 0 1	
C00821/7	0: FALSE	bIn3...bIn1=1 1 0	
C00821/8	0: FALSE	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

Parameter | Name: **C00830 | 16-bit inputs [%]**	Data type: INTEGER_16 Index: 23745_d = $5CC1_h$
C00830/39	Reserved
C00830/40	Reserved
C00830/41	L_Compare_2: nIn1_a
C00830/42	L_Compare_2: nIn2_a
C00830/43	Reserved
C00830/44	Reserved
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	Reserved
C00830/50	Reserved
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

Parameter | Name: **C00830 | 16-bit inputs [%]**	Data type: INTEGER_16 Index: 23745d = 5CC1h
C00830/83	LS_DataAccess: wIn1 (Lenze-internal)
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	L_SMStateDecoder_1: wState
C00830/94	L_SMStateDecoderIO_1: wIOState
C00830/95	L_SMControlDecoder_1: wControl
☑ 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: 23744d = 5CC0h
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

Parameter | Name: **C00831 | 16-bit inputs**	Data type: UNSIGNED_16 Index: 23744_d = $5CC0_h$
C00831/16	L_Compare_1: nIn1_a
C00831/17	L_Compare_1: nIn2_a
C00831/18	MCTRL: nTorqueLimitAdapt_a
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	Reserved
C00831/40	Reserved
C00831/41	L_Compare_2: nIn1_a
C00831/42	L_Compare_2: nIn2_a
C00831/43	Reserved
C00831/44	Reserved
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	Reserved
C00831/50	Reserved
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

Parameter | Name: **C00831 | 16-bit inputs**	Data type: UNSIGNED_16 Index: 23744_d = $5CC0_h$
C00831/60	L_OffsetGain_2: nGain_a
C00831/61	L_PCTRL_1: nAct_a
C00831/62	L_PCTRL_1: nAdapt_a
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	L_SMStateDecoder_1: wState
C00831/94	L_SMStateDecoderIO_1: wIOState
C00831/95	L_SMControlDecoder_1: wControl
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT	

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
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

Parameter | Name: **C00833 | Binary inputs**	Data type: UNSIGNED_8 Index: 23742_d = $5CBE_h$
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
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: bBrkStartValue2

Parameter | Name: **C00833 | Binary inputs**	Data type: UNSIGNED_8 Index: 23742_d = $5CBE_h$
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	Reserved
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
C00833/107	Reserved
C00833/108	Reserved
C00833/109	Reserved
C00833/110	Reserved
C00833/111	L_JogCtrlExtension_1: bInputSel1
C00833/112	L_JogCtrlExtension_1: bInputSel2
C00833/113	L_JogCtrlExtension_1: bRfgIn
C00833/114	L_JogCtrlExtension_1: bJog1In
C00833/115	L_JogCtrlExtension_1: bJog2In
C00833/116	L_JogCtrlExtension_1: bSlowDown1
C00833/117	L_JogCtrlExtension_1: bStop1
C00833/118	L_JogCtrlExtension_1: bbSlowDown2
C00833/119	L_JogCtrlExtension_1: bStop2
C00833/120	L_JogCtrlExtension_1: bSlowDown3
C00833/121	L_JogCtrlExtension_1: bStop3

Parameter | Name: C00833 | Binary inputs	Data type: UNSIGNED_8 Index: 23742_d = $5CBE_h$
C00833/122	L_PCTRL_1: bISet
C00833/123	L_Interpolator_1: bSpeedAct0
☑ 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
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C00840

Parameter | Name: C00840 | 16-bit inputs I/O level [%]			Data type: INTEGER_16 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			Reserved
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

Parameter | Name: **C00840 | 16-bit inputs I/O level [%]**	Data type: INTEGER_16 Index: 23735_d = $5CB7_h$
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	Reserved
C00840/39	Reserved
C00840/40	Reserved
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	Reserved

Parameter | Name: **C00840 | 16-bit inputs I/O level [%]**	Data type: INTEGER_16 Index: 23735_d = $5CB7_h$
C00840/62	Reserved
C00840/63	Reserved
C00840/64	Reserved
C00840/65	Reserved
C00840/66	Reserved
C00840/67	Reserved
C00840/68	Reserved
C00840/69	Reserved
C00840/70	Reserved
C00840/71	Reserved
C00840/72	Reserved
C00840/73	LS_SMStateDecoder_1: wState
☑ 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		
Subcodes			**Info**
C00841/1			Reserved
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

Parameter | Name: **C00841 | 16-bit inputs I/O level**	Data type: UNSIGNED_16 Index: 23734_d = $5CB6_h$
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	Reserved
C00841/39	Reserved
C00841/40	Reserved
C00841/41	LS_DisFree_a: nDis5_a
C00841/42	LS_DisFree_a: nDis6_a
C00841/43	LS_DisFree_a: nDis7_a
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	Reserved
C00841/62	Reserved
C00841/63	Reserved

Parameter | Name: C00841 | 16-bit inputs I/O level	Data type: UNSIGNED_16 Index: 23734_d = $5CB6_h$
C00841/64	Reserved
C00841/65	Reserved
C00841/66	Reserved
C00841/67	Reserved
C00841/68	Reserved
C00841/69	Reserved
C00841/70	Reserved
C00841/71	Reserved
C00841/72	Reserved
C00841/73	LS_SMStateDecoder_1: wState
☑ 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		Reserved
C00843/2		LS_DigitalOutput: bOut1
C00843/3		LS_DigitalInput: bCountIn1_Reset
C00843/4		LS_DigitalInput: bCountIn1_LoadStartValue
C00843/5		LP_CanOut1: bState_B0
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

Parameter | Name: C00843 | Binary inputs I/O level	Data type: UNSIGNED_8 Index: 23732_d = $5CB4_h$
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
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

Parameter | Name: **C00843 | Binary inputs I/O level**	Data type: UNSIGNED_8 Index: 23732_d = $5CB4_h$
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
C00843/94	LS_SetError_1: bSetError2
C00843/95	LS_SetError_1: bSetError3
C00843/96	LS_SetError_1: bSetError4
C00843/97	LS_DigitalInput: bCountIn5_Reset
C00843/98	LS_DigitalInput: bCountIn5_LoadStartValue
C00843/99	LS_DigitalOutput: bOut2
C00843/100	Reserved
C00843/101	Reserved
C00843/102	Reserved
C00843/103	Reserved
C00843/104	Reserved
C00843/105	Reserved
C00843/106	Reserved
C00843/107	Reserved
C00843/108	Reserved
C00843/109	Reserved
C00843/110	LS_DigitalOutput: bBrakeRelease
C00843/111	LS_ParReadWrite_1: bExecute
C00843/112	LS_ParReadWrite_1: bReadWrite

Parameter | Name: **C00843 | Binary inputs I/O level**	Data type: UNSIGNED_8 Index: 23732_d = $5CB4_h$
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	Reserved
C00843/118	Reserved
C00843/119	Reserved
C00843/120	Reserved
C00843/121	Reserved
C00843/122	Reserved
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
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	Reserved
C00843/148	Reserved
C00843/149	Reserved
C00843/150	Reserved
C00843/151	Reserved
C00843/152	Reserved
C00843/153	Reserved
C00843/154	Reserved
C00843/155	Reserved
C00843/156	Reserved

Parameter | Name: **C00843 | Binary inputs I/O level**	Data type: UNSIGNED_8 Index: 23732_d = $5CB4_h$
C00843/157	Reserved
C00843/158	Reserved
C00843/159	Reserved
C00843/160	Reserved
C00843/161	Reserved
C00843/162	Reserved
☑ 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 range (min. value | unit | max. value)			
-2147483647	Incr.	2147483647	
Subcodes			**Info**
C00844/1			Reserved
C00844/2			Reserved
C00844/3			Reserved
C00844/4			Reserved
C00844/5			Reserved
C00844/6			Reserved
C00844/7			Reserved
C00844/8			Reserved
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 ▶ "CAN on board" system bus		
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 "CAN on board" system bus			
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 • 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 • 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 • 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				

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!	

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	Activation of signal interpolation
C00950/2	0: Off	Activation of signal 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: Filter**			Data type: UNSIGNED_8 Index: 23622_d = $5C46_h$
From version 02.00.00 FB L_Interpolator_1: Filter			
Setting range (min. value | unit | max. value)			**Lenze setting**
0		5	**0**
☑ 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) • The time constant of slip compensation serves to specify the dynamics of slip compensation for V/f characteristic control without feedback. • 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 • The slip regulator output is limited to the value set here in motor/generator mode. • 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. • 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) • The gain must be selected depending on the drive system and the sensor resolution (range: 0.005 ... 5). • 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) • 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				

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) • 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. ▶ 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) ▶ 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 ▶ 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 (C00081). • 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			

C01082

Parameter | Name: **C01082 | LS_WriteParamList: Execute Mode**			Data type: UNSIGNED_8 Index: 23493_d = $5BC5_h$
From version 02.00.00 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: FailState**			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 • Format: <code number>.<subcode number> • 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 • 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 • 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 • 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 • 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 1-3: Index**			Data type: INTEGER_32 Index: 23485_d = $5BBD_h$
From version 06.00.00 Parameter to be read or written. • Format: <code number>,<subcode number> • 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
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1000			

C01091

Parameter | Name: **C01091 | LS_ParReadWrite 1-3: CycleTime**		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	1000 ms	
2000	2000 ms	
5000	5000 ms	
10000	10000 ms	
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
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C01092

Parameter | Name: C01092 | LS_ParReadWrite 1-3: FailState			Data type: UNSIGNED_16 Index: 23483_d = $5BBB_h$
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	
Subcodes			**Info**
C01092/1			LS_ParReadWrite_1: FailState
C01092/2			LS_ParReadWrite_2: FailState
C01092/3			LS_ParReadWrite_3: FailState
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☑ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C01093

Parameter | Name: C01093 - LS_ParReadWrite 1-3: Arithmetic mode		Data type: UNSIGNED_8 Index: 23482_d = $5BBA_h$
From version 06.00.00 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
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C01094

Parameter | Name: **C01094 - LS_ParReadWrite 1-3: Numerator**			Data type: INTEGER_16 Index: 23481_d = $5BB9_h$
From version 06.00.00 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
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C01095

Parameter | Name: **C01095 - LS_ParReadWrite 1-3: Denominator**			Data type: INTEGER_16 Index: 23480_d = $5BB8_h$
From version 06.00.00 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
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C01120

Parameter | Name: **C01120 | Sync signal source**		Data type: UNSIGNED_8 Index: 23455_d = $5B9F_h$
Selection of the signal source for device synchronisation • The drive can only be synchronised by one source.		
Selection list (Lenze setting printed in bold)		**Info**
0	**Off**	Synchronisation off
1	CAN on board	Synchronisation via "CAN on board" system bus ▶ Sync telegram
☑ 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 interval at which the phase control loop (PLL) in the controller expects the synchronisation signals. • 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 on board" system bus, a distance of 2 ms has been set between two synchronisation signals. If the system bus is to be used as synchronisation source, a synchronisation cycle of 2000 µs must be set in C01121. ▶ "CAN on board" system bus: Sync telegram			
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: **C01122 | Sync phase position**			Data type: UNSIGNED_16 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.			
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: **C01123 | Sync window**			Data type: UNSIGNED_16 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.			
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: **C01124 | Sync correction width**			Data type: UNSIGNED_8 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.			
Selection list (Lenze setting printed in bold)			
1	100ns		
2	200ns		
3	**300ns**		
4	400ns		
☑ 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			

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 • Response to a communication error of the attached communication module.	
C01501/2	0: No Reaction	Resp. to invalid MCI module • Response to an unplugged or incompatible communication module	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C01751

Parameter | Name: **C01751 | Service code inverter charact.**	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. TRC 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. TRC 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 X70: 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 X70	
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: **C01903 | Diagnostics X70: Change baud rate**	Data type: UNSIGNED_8 Index: 22672_d = 5890_h
New baud rate determination at the diagnostic interface X70	
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: **C01905 | Diagnostics X70: Current baud rate**			Data type: UNSIGNED_32 Index: 22670_d = $588E_h$
Current baud rate at the diagnostic interface X70			
Display range (min. value | unit | max. value)			
0	Bd	3000000	
☑ 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.	
☑ 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)			
0.00	%	199.99	
Subcodes	**Lenze setting**		**Info**
C02581/1	5.00 %		Holding brake: Switching threshold • Switching threshold of the speed setpoint from which on the holding brake is released/applied automatically.
C02581/2	1.00 %		Holding brake: Hyst.release • Hysteresis for holding brake release. • Release threshold = switching threshold + release hysteresis
C02581/3	1.00 %		Holding brake: Hyst. close • Hysteresis for holding brake application. • Application threshold = switching threshold - application hysteresis
☑ 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	**0x00** (decimal: 0)
Value is bit-coded: (☑ = bit set)			**Info**
Bit 0 ☐	bBrkReleaseOut invert.		Activation of inverted control • "1" ≡ Inverted logic of the control signal for the holding brake control switching element.

Parameter | Name: **C02582 | Holding brake: Setting**			Data type: UNSIGNED_8 Index: 21993_d = $55E9_h$
Bit 1 ☐	[--]: nAct < nMin at CInh	Brake response in case of pulse inhibit • "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. **Note:** • 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. • 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 ☐	[||]: Feedforward control inverted	Direction of feedforward control with vertical/hoist technology: • "0" ≡ Positive direction • "1" ≡ Negative direction **Note:** Reversal (Ccw) is then considered.	
Bit 3 ☐	[--]: Horizontal	Direction of movement of the axis • "0" ≡ The axis performs vertical movements. Gravitational acceleration causes movements. • "1" ≡ The direction of the axis is horizontal or rotary. The gravitational acceleration does not cause any movement.	
Bit 4 ☐	Reserved		
Bit 5 ☐	Reserved		
Bit 6 ☑	Sync ramp L_NSet_1	From version 11.00.00 Selection of the ramp time for the synchronisation process to setpoint speed after the brake opening time has elapsed **Revised behaviour from version 11.00.00:** • "1" ≡ The ramp time of the effective acceleration of the ramp function generator (L_NSet_1) is used (Lenze setting). • "0" ≡ As before, the ramp time set in C02610/1 is used. **Note:** 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: **C02589 | Holding brake: Time system**			Data type: UNSIGNED_16 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 • 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 • Time in which the holding brake is completely released from the beginning of control.
C02589/3	100 ms		Holding brake: Waiting time status • 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.
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1			

C02593

Parameter | Name: **C02593 | Holding brake: Activation time**			Data type: UNSIGNED_32 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 • Time in which the actual value is supposed to reach the threshold for closing the brake if the setpoint has already reached the threshold. • 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. • 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
☑ 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	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			

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		MCK: Holding brake ramp time synchr. • Ramp time for approaching the speed setpoint at the MCK after holding brake release. ▸ Holding brake control
C02610/2	2.000 s		MCK: Ramp time synchr. setpoint • 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. ▸ Speed Min/Max			
Setting range (min. value | unit | max. value)			
0.00	%	199.99	
Subcodes	**Lenze setting**		**Info**
C02611/1	199.99 %		MCK: Pos. max. speed • Upper limit of the speed setpoint limitation in positive direction of rotation.
C02611/2	0.00 %		MCK: Pos. min. speed • Lower limit of the speed setpoint limitation in positive direction of rotation.
C02611/3	0.00 %		MCK: Neg. min. speed • Lower limit of the speed setpoint limitation in negative direction of rotation.
C02611/4	199.99 %		MCK: Neg. max. speed • 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			

C02830

Parameter | Name: **C02830 | DIx: Debounce time**		Data type: UNSIGNED_8 Index: 21745_d = $54F1_h$
Debounce times for the digital inputs		▶ Digital terminals
Selection list		
0	0.00 ms	
4	1.00 ms	
8	2.00 ms	
12	3.00 ms	
16	4.00 ms	
20	5.00 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	0: 0.00 ms	DI1: Debounce time
C02830/2	0: 0.00 ms	DI2: Debounce time
C02830/3	0: 0.00 ms	DI3: Debounce time
C02830/4	0: 0.00 ms	DI4: Debounce time
C02830/5	0: 0.00 ms	DI5: Debounce time
C02830/6	0: 0.00 ms	DI6: Debounce time
☑ 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(5) 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.		CountIn5: Starting value
C02840/4	65535 incr.		CountIn5: 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(5) as counting input
Display range (min. value | unit | max. value)			
0	Incr.	2147483647	
Subcodes			**Info**
C02841/1			CountIn1: Counter content
C02841/2			CountIn5: 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(5) and DI2(6) 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 %		FreqIn56: 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(5) and DI2(6) 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 %		FreqIn56: Gain
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 100			

C02844

Parameter | Name: **C02844 | PosIn12: Function**		Data type: UNSIGNED_8 Index: 21731_d = $54E3_h$
From version 06.00.00 ▸ 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	PosIn12: Function
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C02845

Parameter | Name: **C02845 | PosIn12: Comparison**			Data type: INTEGER_32 Index: 21730_d = $54E2_h$
From version 06.00.00 ▸ Output of the encoder position of the DI1/DI2 frequency input			
Setting range (min. value | unit | max. value)			**Lenze setting**
0		2147418112	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1			

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!	

15.2.1 Selection lists for connection parameters

15.2.1.1 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
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
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
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
16015	DIGIN_wCountIn1_HW
16016	DIGIN_nFreqIn12_a
16017	DIGIN_nFreqIn12_v
16018	DIGIN_wCountIn5_LW
16019	DIGIN_wCountIn5_HW
16020	DIGIN_nFreqIn56_a
16021	DIGIN_nFreqIn56_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

Selection list - analog signals	
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
16200	LS_SafetyModuleInterface_SMI_wState
16201	LS_SafetyModuleInterface_SMI_wIOState
16202	LS_SafetyModuleInterface_SMI_wControl
16203	LS_SafetyModuleInterface_SMI_wModuleID
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_nPar1_a
20011	LS_ParFree_nPar2_a
20012	LS_ParFree_nPar3_a
20013	LS_ParFree_nPar4_a
20018	LS_ParFree_nPar1_v
20019	LS_ParFree_nPar2_v
20020	LS_ParFree_nPar3_v
20021	LS_ParFree_nPar4_v
20026	LS_ParFree_wPar1
20027	LS_ParFree_wPar2
20028	LS_ParFree_wPar3
20029	LS_ParFree_wPar4
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
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

Selection list - analog signals	
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
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
36018	L_NSet_NOut_1
36019	L_MotorPoti_1_Out
36020	L_PCTRL_1_Out
36021	L_SigMonitor_a_Out1
36022	L_SigMonitor_a_Out2
36023	L_NLim_1_nOut_a
36025	L_OffsetGainP_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
36091	L_PCTRL_1_nPIDOut1_a
36092	L_PCTRL_1_nPIDOut2_a
36093	L_PCTRL_1_nInfluenceOut_a

Selection list - analog signals	
36095	L_Interpolator_1_nPhdOut_v
36096	L_Interpolator_1_nNOut_a
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
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

15.2.1.2 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
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
1200	LA_SwitchPos_bDriveFail
1201	LA_SwitchPos_bWarningActive
1202	LA_SwitchPos_bSafeTorqueOff
1203	LA_SwitchPos_bDriveReady
1204	LA_SwitchPos_bCInhActive
1205	LA_SwitchPos_bImpIsActive
1206	LA_SwitchPos_bQSPIsActive
1207	LA_SwitchPos_bSpeedCcw
1208	LA_SwitchPos_bSpeedActCompare
1209	LA_SwitchPos_bImaxActive

Selection list - digital signals	
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
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
16008	DigIn_CInh
16009	DigIn_bCountIn1_Compare
16010	DigIn_bCountIn5_Compare
16011	Ain_bCurrentErrorIn1
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

Selection list - digital signals	
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
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	McIn_bCtrl_B0
16062	McIn_bCtrl_B1
16063	McIn_bCtrl_B2
16064	McIn_bCtrl_B3
16065	McIn_bCtrl_B4
16066	McIn_bCtrl_B5
16067	McIn_bCtrl_B6
16068	McIn_bCtrl_B7
16069	McIn_bCtrl_B8
16070	McIn_bCtrl_B9
16071	McIn_bCtrl_B10
16072	McIn_bCtrl_B11
16073	McIn_bCtrl_B12
16074	McIn_bCtrl_B13
16075	McIn_bCtrl_B14
16076	McIn_bCtrl_B15
16077	McIn_bIn2_B0
16078	McIn_bIn2_B1
16079	McIn_bIn2_B2
16080	McIn_bIn2_B3
16081	McIn_bIn2_B4
16082	McIn_bIn2_B5
16083	McIn_bIn2_B6

Selection list - digital signals	
16084	McIn_bIn2_B7
16085	McIn_bIn2_B8
16086	McIn_bIn2_B9
16087	McIn_bIn2_B10
16088	McIn_bIn2_B11
16089	McIn_bIn2_B12
16090	McIn_bIn2_B13
16091	McIn_bIn2_B14
16092	McIn_bIn2_B15
16093	LS_Keypad_bSetQuickstop
16094	LS_Keypad_bSetDCBrake
16095	LS_Keypad_bSetSpeedCcw
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
16122	LS_WriteParamList_bDone
16123	LS_WriteParamList_bFail
16124	LS_SafetyModuleInterface_bPowerStageEnable
16125	L_SMStateDecoder_1_bSafeTorqueOff
16126	L_SMStateDecoder_1_bErrorClassSTO
16127	L_SMStateDecoder_1_bErrorClassSS1
16128	L_SMStateDecoder_1_bError
16129	L_SMStateDecoderIO_1_bSafeDigIn1
16130	L_SMStateDecoderIO_1_bSafeDigIn2
16131	L_SMStateDecoderIO_1_bACKInputStop
16132	L_SMStateDecoderIO_1_bACKInputError
16133	L_SMStateDecoderIO_1_bPROFIsafeAIS
16134	L_SMStateDecoderIO_1_bPROFIsafeAIE
16135	L_SMControlDecoder_1_bSafeStop1
16136	L_SMControlDecoder_1_bEnableSwitch
16137	L_SMControlDecoder_1_bOperationModeSelector
16138	L_SMControlDecoder_1_bSafeStopEmergency
16139	L_SMControlDecoder_1_bSpecialOperationActive
16140	LS_IRInterface_bIRSignal_F1
16141	LS_IRInterface_bIRSignal_F2
16142	LS_IRInterface_bIRSignal_F3
16143	LS_IRInterface_bIRSignal_F4
16144	LS_IRInterface_bIRSignal_Down
16145	LS_IRInterface_bIRSignal_Up
16146	LS_IRInterface_bIRSignal_On
16147	LS_IRInterface_bIRSignal_Off
16148	LS_IRInterface_bIRSignal_0
16149	LS_IRInterface_bIRSignal_1

Selection list - digital signals	
16150	LS_IRInterface_bIRSignal_2
16151	LS_IRInterface_bIRSignal_3
16152	LS_IRInterface_bIRSignal_4
16153	LS_IRInterface_bIRSignal_5
16154	LS_IRInterface_bIRSignal_6
16155	LS_IRInterface_bIRSignal_7
16156	LS_IRInterface_bIRSignal_8
16157	LS_IRInterface_bIRSignal_9
16158	LS_ServiceSwitch_bSwitch_left
16159	LS_ServiceSwitch_bSwitch_right
16160	LS_ServiceSwitch_bSwitch_middle
16161	LS_CANManagement_bFail
16162	LS_CANManagement_bOperational
16167	LS_ServiceSwitch_bSwitch_Manual
16168	LS_ServiceSwitch_bSwitch_MainOff
20000	LS_ParFix_True
20001	bPar1
20002	bPar2
20003	bPar3
20004	bPar4
20005	bPar5
20006	bPar6
20007	bPar7
20008	bPar8
20009	bPar9
20010	bPar10
20011	bPar11
20012	bPar12
20013	bPar13
20014	bPar14
20015	bPar15
20016	bPar16
20033	b100Hz
20034	b10Hz
20035	b2Hz
20036	b1Hz
20037	b1HzFlash
20038	b2HzFlash
20039	bSingleFlash1
20040	bSingleFlash2
20041	bDoubleFlash
20042	bSquareWave
20043	bFirstCycle
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

Selection list - digital signals	
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
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

Selection list - digital signals	
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
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
36016	L_Nset_RfgEq0_1
36017	L_DigitalLogic_1_bOut
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
36131	L_JogCtrlExtension_bRfgOut
36132	L_JogCtrlExtension_bJog1Out
36133	L_JogCtrlExtension_bJog2Out
36135	L_Interpolator_1_bIPulse
36136	L_Interpolator_1_bSignalError
36138	L_Transient_1_bOut
36139	L_Transient_2_bOut
36140	L_Transient_3_bOut
36141	L_Transient_4_bOut
42000	LA_NCtrl_In_bCInh
42001	LA_NCtrl_In_bFailReset
42002	LA_NCtrl_In_bSetQuickstop
42003	LA_NCtrl_In_bSetDCBrake
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

Selection list - digital signals	
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
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
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

Selection list - digital signals	
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

15.3 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 10 ≡ 1 decimal position 100 ≡ 2 decimal positions 1000 ≡ 3 decimal positions 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	33	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	☑	☑	
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	Comparison value N_Act	24551	5FE7	E	1	INTEGER_16	100	☑	☑	
C00026	AINx: Offset	24549	5FE5	A	1	INTEGER_16	100	☑	☑	
C00027	AINx: Gain	24548	5FE4	A	1	INTEGER_32	10000	☑	☑	
C00028	AINx: Input voltage	24547	5FE3	A	1	INTEGER_16	100	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00029	AINx: Input current	24546	5FE2	A	1	INTEGER_16	100	☑		
C00033	AINx: Output value	24542	5FDE	A	1	INTEGER_16	100	☑		
C00034	AINx: Configuration	24541	5FDD	A	1	UNSIGNED_8	1	☑	☑	
C00036	DCB: 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	☑		
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	☑		
C00061	Heatsink temperature	24514	5FC2	E	1	INTEGER_16	1	☑		
C00064	Device utilisation (Ixt)	24511	5FBF	A	3	INTEGER_16	100	☑		
C00066	Thermal motor load (I²xt)	24509	5FBD	E	1	INTEGER_16	100	☑		
C00070	Vp speed controller	24505	5FB9	A	1	UNSIGNED_16	100	☑	☑	
C00071	Ti speed controller	24504	5FB8	A	1	UNSIGNED_16	10	☑	☑	
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	☑	☑	
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		☑		
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	☑	☑	
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	DCB: Hold time	24468	5F94	E	1	UNSIGNED_32	1000	☑	☑	
C00114	DIx: Polarity	24461	5F8D	E	1	UNSIGNED_16		☑	☑	
C00115	DI 1/2 & 5/6: Function	24460	5F8C	A	2	UNSIGNED_8	1	☑	☑	
C00116	DI 3/4 DO 1/2: Function	24459	5F8B	E	1	UNSIGNED_8	1	☑	☑	☑
C00118	DOx: Inversion	24457	5F89	E	1	UNSIGNED_8		☑	☑	
C00120	Motor overload threshold (I²xt)	24455	5F87	E	1	INTEGER_16	100	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00122	Initial value motor overload (I²xt)	24453	5F85	A	1	UNSIGNED_16	100	☑	☑	
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	Ramp smoothing main setpoint	24441	5F79	E	1	UNSIGNED_8	1	☑	☑	
C00136	Communication control words	24439	5F77	A	2	UNSIGNED_16		☑		
C00137	Device status	24438	5F76	E	1	UNSIGNED_16	1	☑		
C00138	Internal control signals	24437	5F75	A	3	UNSIGNED_16		☑		
C00142	Auto-start option	24433	5F71	E	1	UNSIGNED_8		☑	☑	
C00144	Thermal switching frequency reduction	24431	5F6F	E	1	UNSIGNED_8	1	☑	☑	
C00148	Config. group error	24427	5F6B	E	1	UNSIGNED_16		☑	☑	
C00150	Status word	24425	5F69	E	1	UNSIGNED_16		☑		
C00155	Extended status word	24420	5F64	E	1	UNSIGNED_16		☑		
C00158	Cause of controller inhibit	24417	5F61	E	1	UNSIGNED_16		☑		
C00159	Cause of quick stop QSP	24416	5F60	E	1	UNSIGNED_16		☑		
C00160	Status determining error (16-bit)	24415	5F5F	A	1	UNSIGNED_16	1	☑		
C00161	LS_SetError_x: Error number	24414	5F5E	A	4	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		☑		
C00166	Error information text	24409	5F59	A	6	VISIBLE_STRING		☑		
C00168	Status determining error	24407	5F57	E	1	UNSIGNED_32	1	☑		
C00169	Logbook setting	24406	5F56	E	1	UNSIGNED_16		☑	☑	
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	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	☑		
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	Setpoint arithmetic	24385	5F41	E	1	UNSIGNED_8	1	☑	☑	
C00199	Description data	24376	5F38	A	1	VISIBLE_STRING		☑	☑	
C00200	Firmware product type	24375	5F37	E	1	VISIBLE_STRING		☑		
C00201	Firmware	24374	5F36	A	6	VISIBLE_STRING		☑		
C00203	Product type code	24372	5F34	A	10	VISIBLE_STRING		☑		
C00204	Serial number	24371	5F33	A	10	VISIBLE_STRING		☑		
C00220	Accel. time - add. setpoint	24355	5F23	E	1	UNSIGNED_32	1000	☑	☑	
C00221	Decel. time - add. setpoint	24354	5F22	E	1	UNSIGNED_32	1000	☑	☑	
C00222	L_PCTRL_1: Vp	24353	5F21	E	1	INTEGER_16	10	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
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: Acceleration 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: Deceleration 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: ActEqSet window	24328	5F08	E	1	INTEGER_16	100	☑	☑	
C00249	L_PT1_1: Time constant	24326	5F06	E	1	UNSIGNED_16	1	☑	☑	
C00273	Moment of inertia	24302	5EEE	E	1	UNSIGNED_32	100	☑	☑	
C00275	Setpoint feedforward control filtering	24300	5EEC	E	1	UNSIGNED_16	10	☑	☑	
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	☑	☑	
C00324	CAN transmit blocking time	24251	5EBB	A	4	UNSIGNED_16	1	☑	☑	
C00338	L_Arithmetik_1: Function	24237	5EAD	E	1	UNSIGNED_8	1	☑	☑	
C00341	CAN management - error configuration	24234	5EAA	E	1	UNSIGNED_16		☑	☑	
C00342	CAN decoupling PDOInOut	24233	5EA9	A	2	UNSIGNED_16		☑	☑	
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	7	UNSIGNED_8	1	☑		
C00349	Setting- DIP switch SW1	24226	5EA2	E	1	UNSIGNED_16		☑		
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		☑	☑	
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		☑	☑	
C00368	CAN SYNC Tx identifier	24207	5E8F	E	1	UNSIGNED_16		☑	☑	
C00369	CAN sync transmission cycle time	24206	5E8E	E	1	UNSIGNED_16	1	☑	☑	
C00370	SyncTxRxTimes	24205	5E8D	A	2	INTEGER_16	1	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
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	7	UNSIGNED_8	1	☑	☑	
C00386	ConsumerTime HeartBeat producer	24189	5E7D	A	7	UNSIGNED_16	1	☑	☑	
C00400	LS_PulseGenerator	24175	5E6F	A	3	UNSIGNED_16	1	☑	☑	
C00401	CANxInOut: Inversion	24174	5E6E	A	6	UNSIGNED_16		☑	☑	
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		☑	☑	
C00413	L_SignalMonitor_a: Offs./gain	24162	5E62	A	8	INTEGER_16	100	☑	☑	
C00420	Number of encoder increments	24155	5E5B	A	2	UNSIGNED_16	1	☑	☑	
C00423	DOx: Delay times	24152	5E58	A	4	UNSIGNED_16	1000	☑	☑	
C00425	Encoder scanning time	24150	5E56	A	2	UNSIGNED_8	1	☑	☑	☑
C00440	LS_AnalogIn1: PT1 time constant	24135	5E47	A	1	UNSIGNED_16	1	☑	☑	
C00443	DIx: Level	24132	5E44	A	2	UNSIGNED_16		☑		
C00444	DOx: Level	24131	5E43	A	2	UNSIGNED_16		☑		
C00445	FreqInxx_nOut_v	24130	5E42	A	2	INTEGER_16	1	☑		
C00446	FreqInxx_nOut_a	24129	5E41	A	2	INTEGER_16	100	☑		
C00447	Decoupling DigOut	24128	5E40	E	1	UNSIGNED_16		☑	☑	
C00448	DigOut decoupling value	24127	5E3F	E	1	UNSIGNED_16		☑	☑	
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		☑	☑	
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: Fct. STOP key	24106	5E2A	E	1	INTEGER_32	1	☑	☑	☑
C00470	LS_ParFree_b	24105	5E29	A	16	UNSIGNED_8	1	☑	☑	
C00471	LS_ParFree	24104	5E28	A	4	UNSIGNED_16		☑	☑	
C00472	LS_ParFree_a	24103	5E27	A	4	INTEGER_16	100	☑	☑	
C00473	LS_ParFree_v	24102	5E26	A	4	INTEGER_16	1	☑	☑	
C00474	LS_ParFree_p	24101	5E25	A	4	INTEGER_32	1	☑	☑	
C00480	LS_DisFree_b	24095	5E1F	E	1	UNSIGNED_8		☑		
C00481	LS_DisFree	24094	5E1E	A	4	UNSIGNED_16		☑		
C00482	LS_DisFree_a	24093	5E1D	A	4	INTEGER_16	100	☑		
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: Text	24089	5E19	A	4	VISIBLE_STRING		☑	☑	
C00487	Application units	24088	5E18	A	4	INTEGER_32	100	☑		
C00488	L_JogCtrlEdgeDetect_1	24087	5E17	A	6	UNSIGNED_8	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	2	UNSIGNED_16	10	☑	☑	
C00505	Password data	24070	5E06	A	3	VISIBLE_STRING		☑	☑	
C00507	Current password protection	24068	5E04	A	1	UNSIGNED_16		☑		
C00517	User menu	24058	5DFA	A	32	INTEGER_32	1000	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00560	Fan switching status	24015	5DCF	A	2	UNSIGNED_8	1	☑		
C00561	Fan failure	24014	5DCE	A	2	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	☑	☑	
C00579	Resp. to max. speed/output freq. reached	23996	5DBC	E	1	UNSIGNED_8	1	☑	☑	
C00581	Resp. to LS_SetError_x	23994	5DBA	A	4	UNSIGNED_8	1	☑	☑	
C00582	Resp. to heatsink temp.> shutdown temp. -5°C	23993	5DB9	E	1	UNSIGNED_8	1	☑	☑	
C00585	Resp. to motor overtemp. PTC	23990	5DB6	E	1	UNSIGNED_8	1	☑	☑	
C00586	Resp. to encoder open circuit	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	☑	☑	
C00597	Resp. to motor phase failure	23978	5DAA	E	1	UNSIGNED_8	1	☑	☑	
C00598	Resp. to I/O monitoring	23977	5DA9	A	4	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	☑	☑	
C00606	Resp. to motor overload (I^2xt)	23969	5DA1	E	1	UNSIGNED_8	1	☑	☑	
C00607	Resp. to max. freq. feedb. DIG12/56	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	88	UNSIGNED_16	1	☑	☑	
C00621	System connection list: Bool	23954	5D92	A	162	UNSIGNED_16	1	☑	☑	
C00622	System connection list: Angle	23953	5D91	A	15	UNSIGNED_16	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		☑		
C00635	L_NSet_1: nMaxLimit	23940	5D84	E	1	INTEGER_16	100	☑	☑	
C00636	L_NSet_1: nMinLimit	23939	5D83	E	1	INTEGER_16	100	☑	☑	
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	☑		
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	☑	☑	
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	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00687	L_Compare_2: Window	23888	5D50	E	1	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	☑	☑	
C00720	L_DigitalDelay_1: Delay	23855	5D2F	A	2	UNSIGNED_32	1000	☑	☑	
C00725	Current switching frequency	23850	5D2A	E	1	UNSIGNED_8	1	☑		
C00727	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	☑	☑	
C00801	L_MPot_1: Lower limit	23774	5CDE	E	1	INTEGER_16	100	☑	☑	
C00802	L_MPot_1: Acceleration time	23773	5CDD	E	1	UNSIGNED_16	10	☑	☑	
C00803	L_MPot_1: Deceleration 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	☑	☑	
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	☑	☑	
C00830	16-bit inputs [%]	23745	5CC1	A	95	INTEGER_16	100	☑		
C00831	16-bit inputs	23744	5CC0	A	95	UNSIGNED_16		☑		
C00833	Binary inputs	23742	5CBE	A	123	UNSIGNED_8	1	☑		
C00834	32-bit inputs [incr]	23741	5CBD	A	7	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		☑		
C00843	Binary inputs I/O level	23732	5CB4	A	162	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		☑		
C00868	CAN output words	23707	5C9B	A	12	UNSIGNED_16		☑		
C00876	MCI input words	23699	5C93	A	16	UNSIGNED_16		☑		
C00877	MCI output words	23698	5C92	A	16	UNSIGNED_16		☑		
C00890	MCI_InOut: Inversion	23685	5C85	A	4	UNSIGNED_16		☑	☑	
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	☑		
C00950	L_Interpolator_1: Activation FB functions	23625	5C49	A	2	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: Filter	23622	5C46	E	1	UNSIGNED_8	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	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
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	☑	☑	
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	☑	☑	
C01082	LS_WriteParamList: Execute Mode	23493	5BC5	E	1	UNSIGNED_8	1	☑	☑	
C01083	LS_WriteParamList: FailState	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 1-3: Index	23485	5BBD	A	3	INTEGER_32	1000	☑	☑	
C01091	LS_ParReadWrite 1-3: CycleTime	23484	5BBC	A	3	UNSIGNED_16	1	☑	☑	
C01092	LS_ParReadWrite 1-3: FailState	23483	5BBB	A	3	UNSIGNED_16	1	☑		
C01093	LS_ParReadWrite 1-3: Arithmetic mode	23482	5BBA	A	3	UNSIGNED_8	1	☑	☑	
C01094	LS_ParReadWrite 1-3: Numerator	23481	5BB9	A	3	INTEGER_16	1	☑	☑	
C01095	LS_ParReadWrite 1-3: Denominator	23480	5BB8	A	3	INTEGER_16	1	☑	☑	
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	☑	☑	
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	☑	☑	
C01501	Resp. to communication error with MCI	23074	5A22	A	2	UNSIGNED_8	1	☑	☑	
C01770	Filter time - earth-fault detect. is running	22805	5915	E	1	UNSIGNED_8	1	☑	☑	
C01902	Diagnostics X70: Max. baud rate	22673	5891	E	1	UNSIGNED_16	1	☑	☑	
C01903	Diagnostics X70: Change baud rate	22672	5890	E	1	UNSIGNED_8	1	☑	☑	
C01905	Diagnostics X70: Current baud rate	22670	588E	E	1	UNSIGNED_32	1	☑		
C02580	Holding brake: Operating mode	21995	55EB	E	1	UNSIGNED_8	1	☑	☑	
C02581	Holding brake: Speed thresholds	21994	55EA	A	3	INTEGER_16	100	☑	☑	
C02582	Holding brake: Setting	21993	55E9	E	1	UNSIGNED_8		☑	☑	
C02589	Holding brake: Time system	21986	55E2	A	3	UNSIGNED_16	1	☑	☑	
C02593	Holding brake: Activation time	21982	55DE	A	2	UNSIGNED_32	1000	☑	☑	
C02607	Holding brake: Status	21968	55D0	E	1	UNSIGNED_16		☑		
C02610	MCK: Accel./decel. times	21965	55CD	A	3	UNSIGNED_32	1000	☑	☑	
C02611	MCK: Limitations	21964	55CC	A	4	INTEGER_16	100	☑	☑	
C02830	DIx: Debounce time	21745	54F1	A	6	UNSIGNED_8	1	☑	☑	
C02840	CountInx: Parameter	21735	54E7	A	4	UNSIGNED_32	1	☑	☑	
C02841	CountInx: Counter content	21734	54E6	A	2	UNSIGNED_32	1	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C02842	FreqInxx: Offset	21733	54E5	A	2	INTEGER_16	100	☑	☑	
C02843	FreqInxx: Gain	21732	54E4	A	2	INTEGER_16	100	☑	☑	
C02844	PosIn12: Function	21731	54E3	A	1	UNSIGNED_8	1	☑	☑	
C02845	PosIn12: Comparison	21730	54E2	E	1	INTEGER_32	1	☑	☑	

16 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 can be used to:

- carry out 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 "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.

16.1 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" version onwards, the experienced user has the opportunity to implement own drive solutions independent of the predefined technology applications by using the "free interconnection".

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 any time into the FB Editor of the »Engineer« even if the Engineer project is not available.

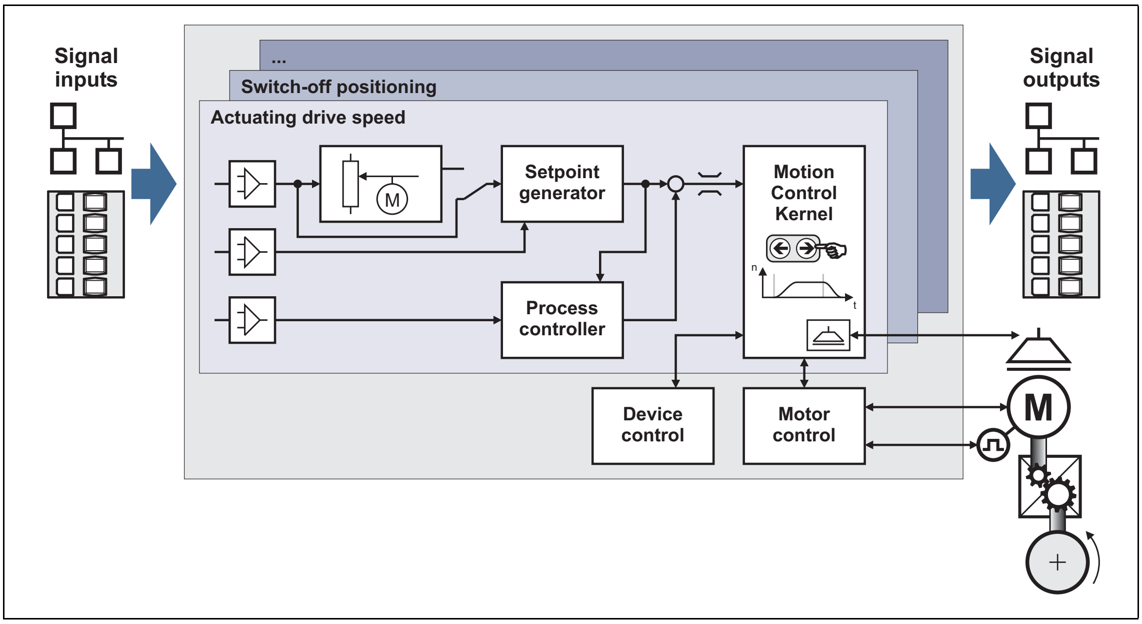
Note!

With the "StateLine" version, the interconnection shown in the application level cannot be edited.

16.1.1 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)



[16-1] Basic components of a drive solution

Regarding the 8400 device series, these three components are available for the FB interconnection and classified as follows:

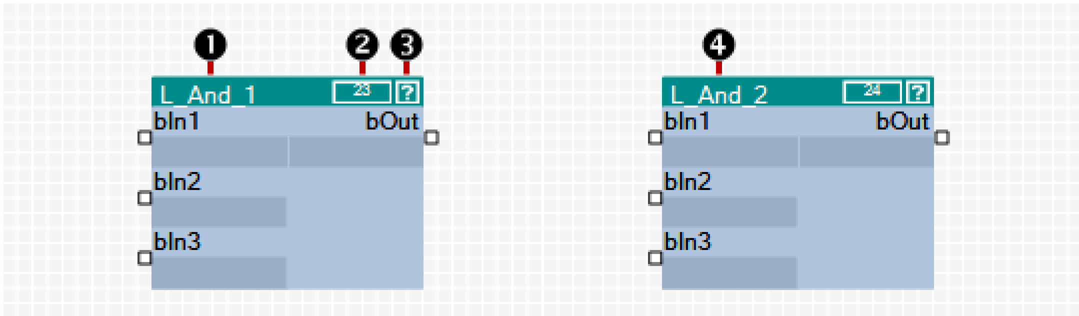
Module type	Name	Task	Example
Function block	L_*name*	General function block for free interconnection (only HighLine)	L_Compare_1 L_PCTRL_1
System block	LS_*name*	Signal interface to inverter-internal functions	LS_DigitalInput LS_DriveInterface
Port block	LP_*name*	• Process data communication via a fieldbus using a communication module • Process data communication via CAN on board	LP_CanIn1 LP_CanOut1 LP_MciIn LP_MciOut
Application block	LA_*name*	Block for a technology application	LA_NCtrl LA_SwitchPos

Further information on the individual modules can be obtained from the following subchapters!

16.1.1.1 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)

[16-2] 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". (📖 710)

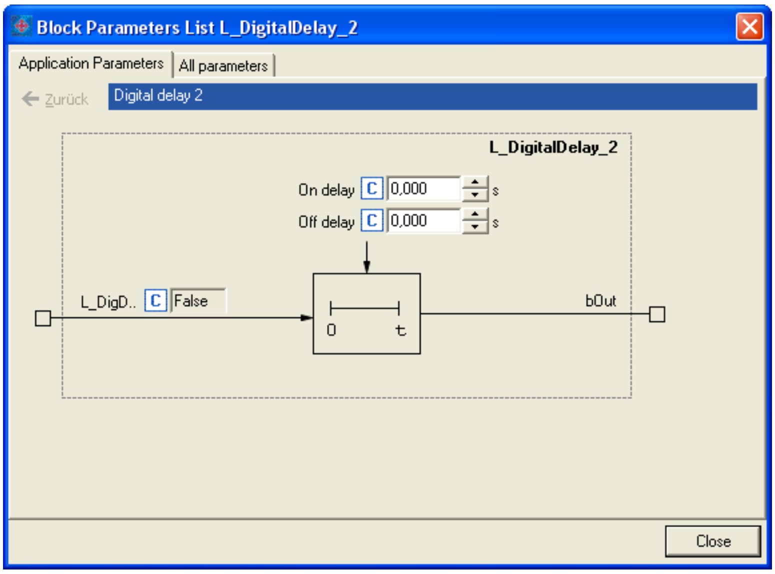
16.1.1.2 Parameterisable function blocks

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.

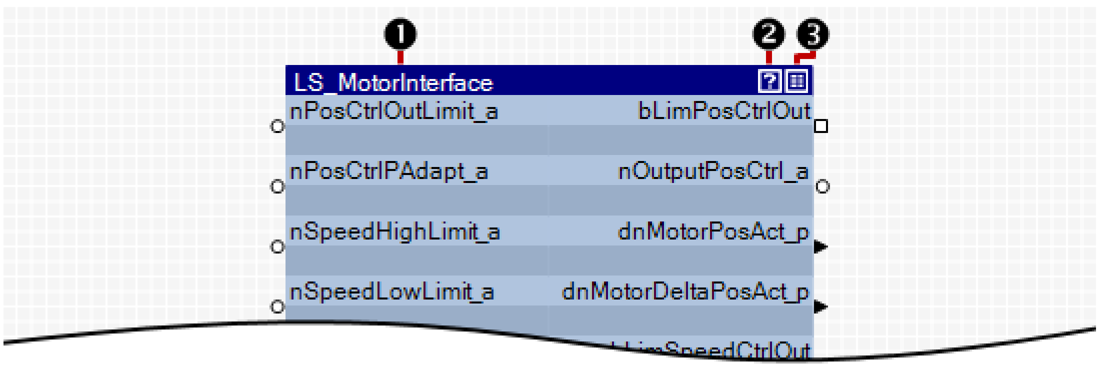
Example

Parameterisation dialog for the FB **L_DigitalDelay_2**:



16.1.1.3 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

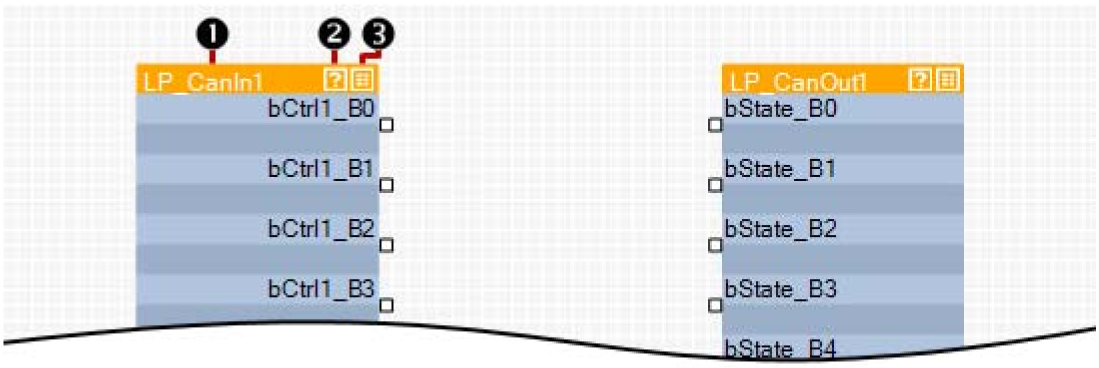
[16-3] Example: System block "LS_MotorInterface" for mapping the motor control

16.1.1.4 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 for communication option

If, for instance, the controller is to be controlled via CAN bus or a communication option (PROFIBUS or PROFINET), 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

[16-4] Example: Input port "LP_CanIn1" and output port "LP_CanOut1"

16.1.1.5 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.

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.

16.1.2 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	1 bit	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

16.1.3 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

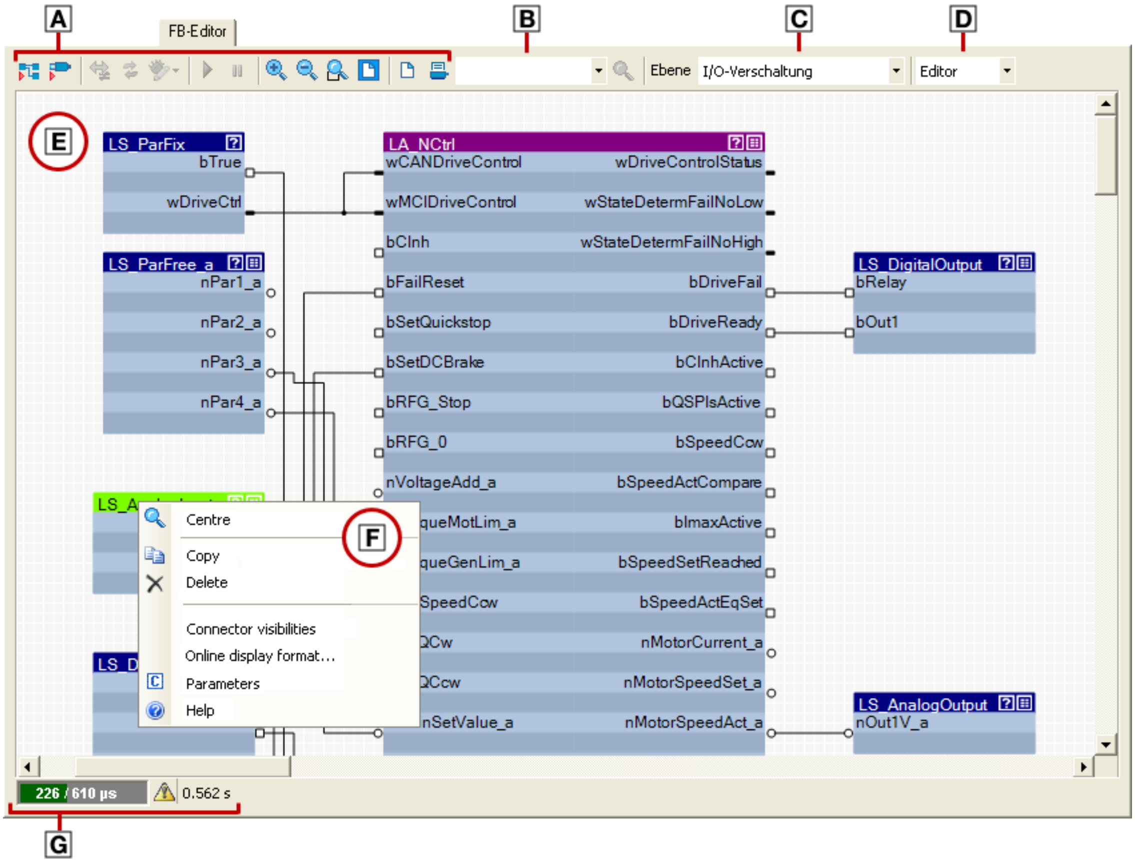
16.2 User interface

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.

16.2.1 Toolbar

The FB Editor is provided with an individual toolbar in the upper position which in the following text is called *FB Editor toolbar*.

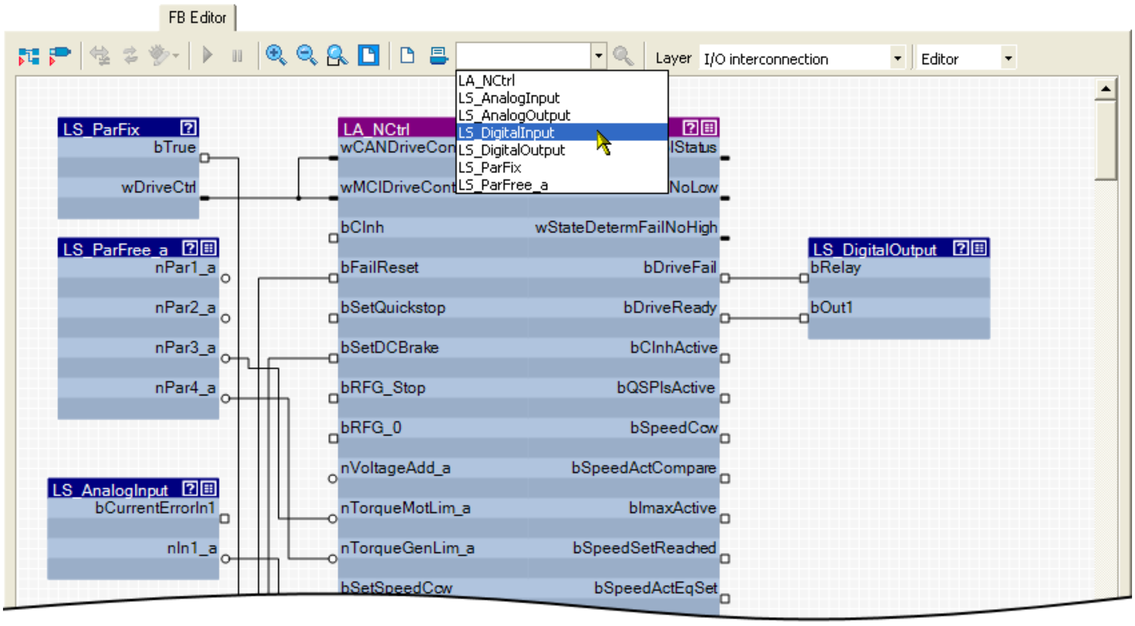
- Click an icon to execute the corresponding function.

Icon	Function
	Insert function block or system block ▸ Inserting a function block (📖 680) ▸ Inserting a system block (📖 682)
	Inserting a port block (📖 684)
	Adjusting online and offline interconnection (📖 703)
	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 (📖 704)
	Search function (📖 667)

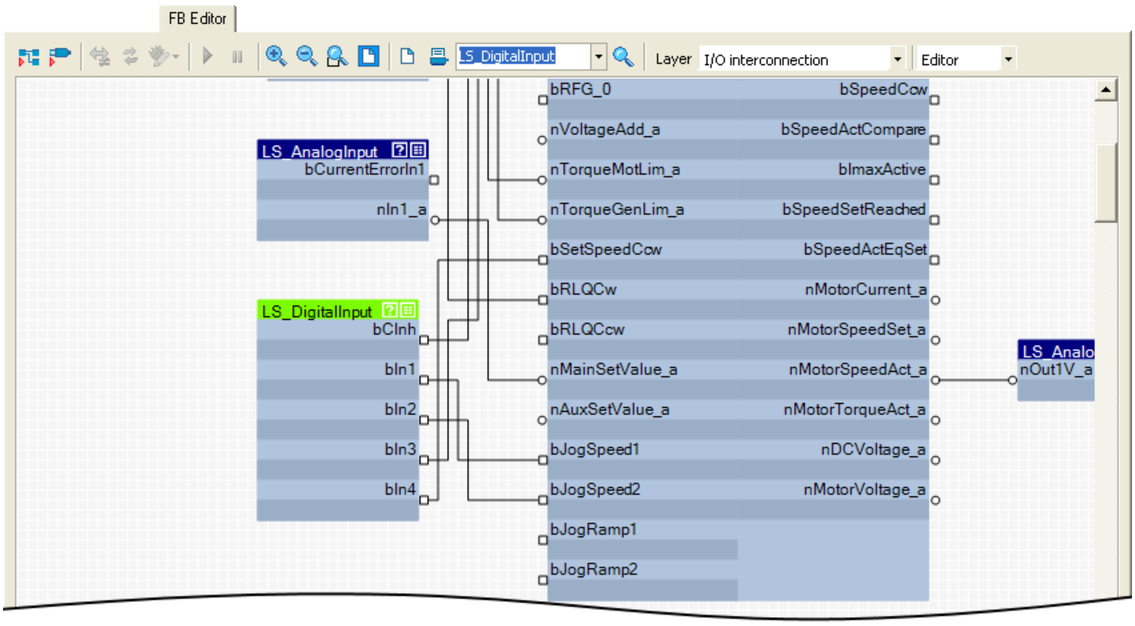
16.2.2 Search function

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.

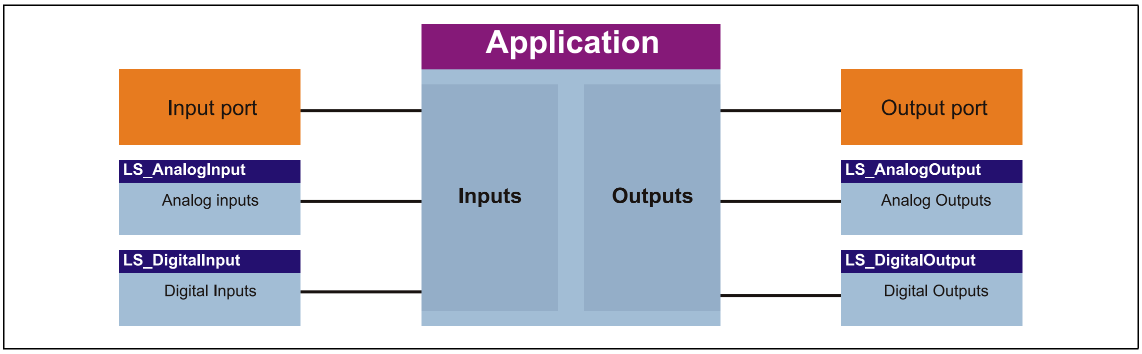
16.2.3 Level selection

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.

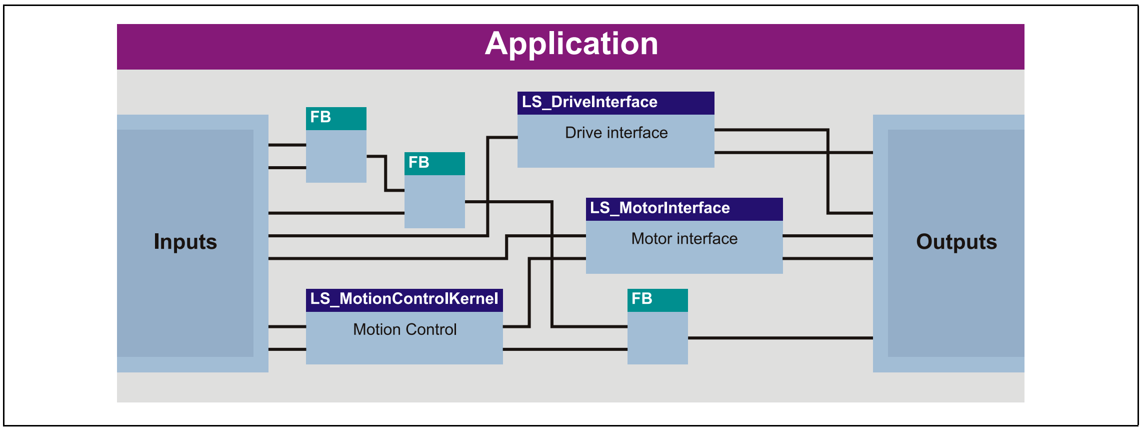


[16-5] Schematic diagram of "I/O interconnection"

"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.



[16-6] 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!

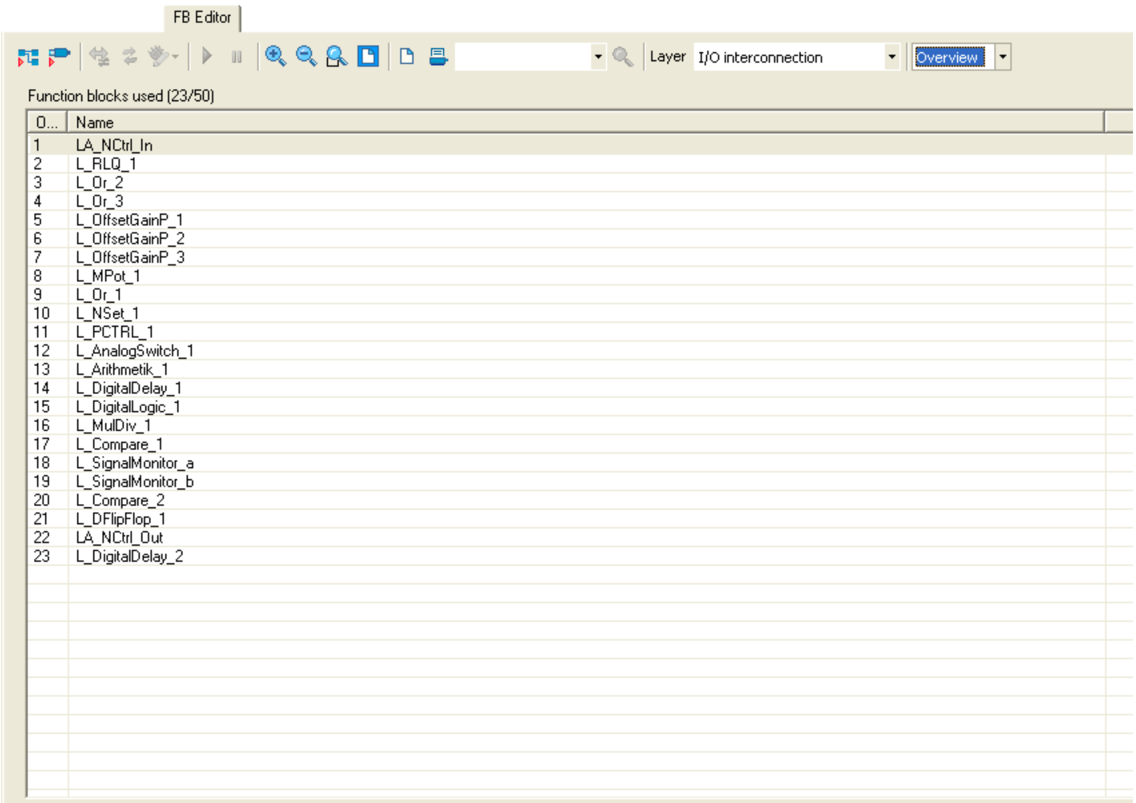
When you select the "Free interconnection" level for the first time, you are prompted to confirm whether the interconnection from the I/O level and the application level are to be combined and copied into this level.

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 (📖 702)

16.2.4 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.

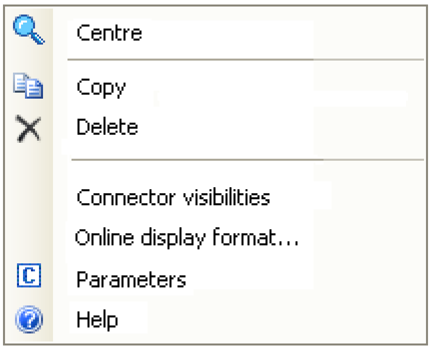


- The processing order of the function blocks can be optimised manually or according to an automatically generated selection. ▶ Changing the processing order (📖 697)

16.2.5 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:



16.2.6 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:

Icon	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	

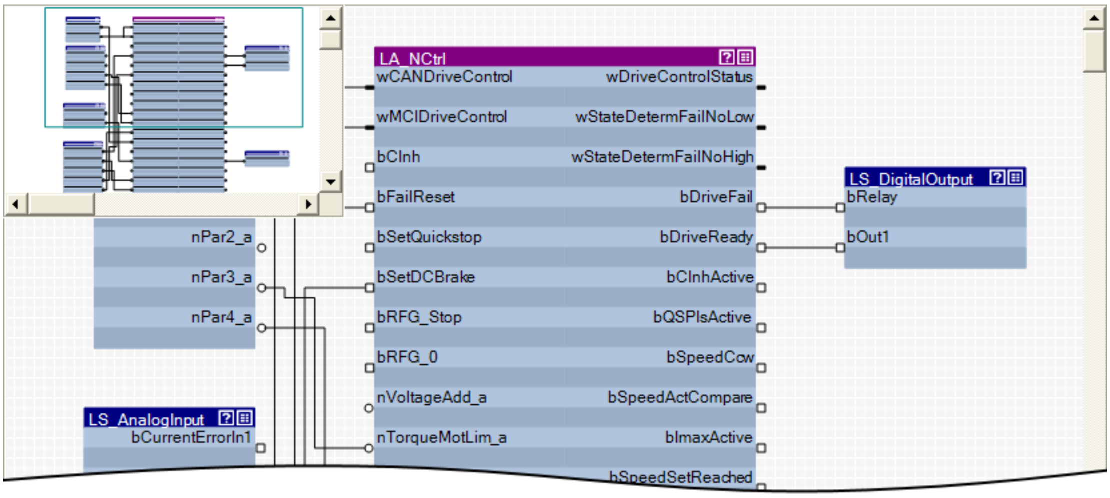
16.2.7 Overview window

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.



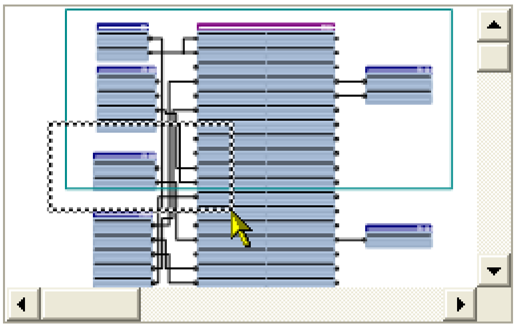
- The green frame in the overview window indicates the interconnection cutout that is currently displayed in the drawing area.
- Use the mouse pointer to shift and resize the cutout to be displayed.

How to shift the cutout presented in the drawing area:

1. Position the mouse pointer <u>to the green frame</u> 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:

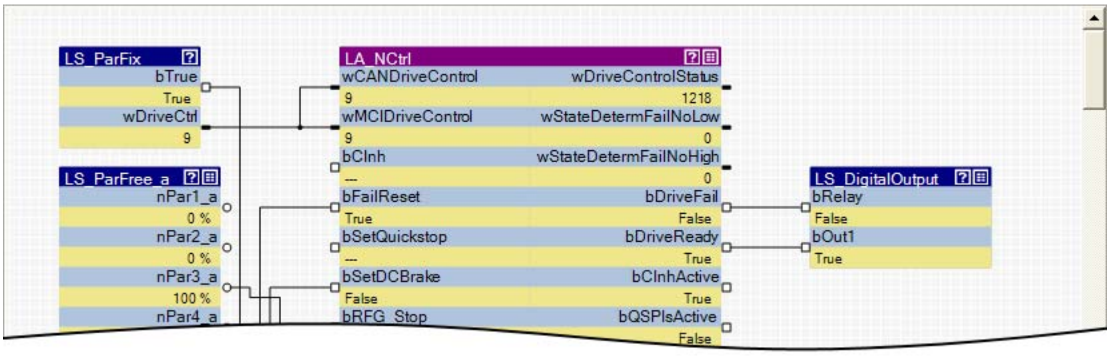
16.3 Using the FB Editor as "Viewer"

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 (📖 677)

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

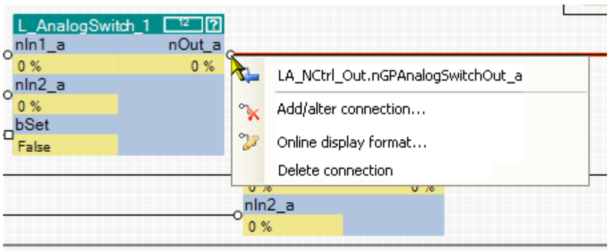
- The [▤] icon in the head of the module, a double-click on the module, or the **Parameter...** command in the *Context menu* of the module serve to open the parameterisation dialog or the parameter list for the module.

16.3.1 Following connections of inputs and outputs

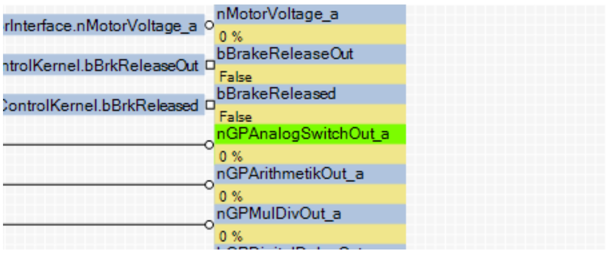
In addition to the Search function you can use the *context menu* of inputs and outputs to follow connections and quickly reach certain signals.

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:



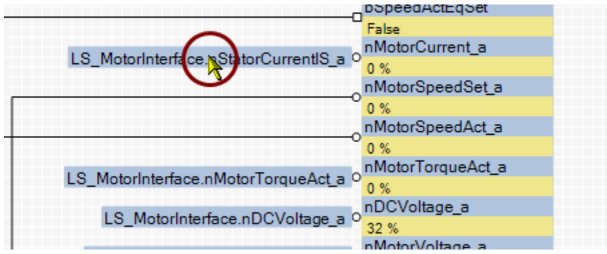
2. Select input in the *context menu* to which you want to navigate.
   - As a result, the selected input is displayed in the centre of the drawing area (in this example: nGPAnalogSwitchOut_a):



How to navigate from one input to another connected output:

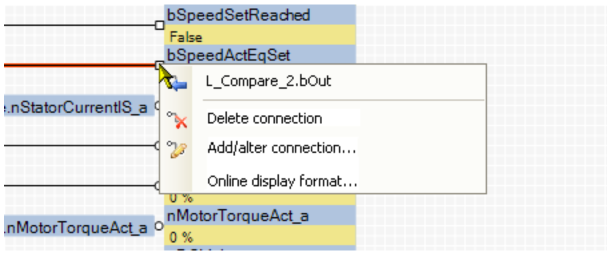
If the input is connected to a flag:

- 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.

16.3.2 Keyboard commands for navigation

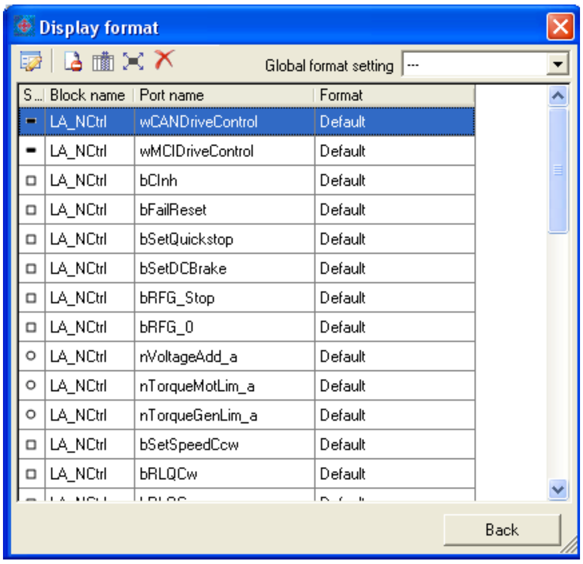
Keyboard command	Function
<Picture ▲ >	Scroll up
<Picture ▼ >	Scroll down
<Shift> + <picture ▲ >	Scroll to the left
<Shift> + <picture ▼ >	Scroll to the right
<POS1>	Scroll to the left edge of the interconnection
<END>	Scroll to the right edge of the interconnection
<Ctrl> + <Pos1>	Scroll to the left upper corner of the interconnection
<Ctrl> + <End>	Scroll to the right lower corner of the interconnection

16.3.3 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:

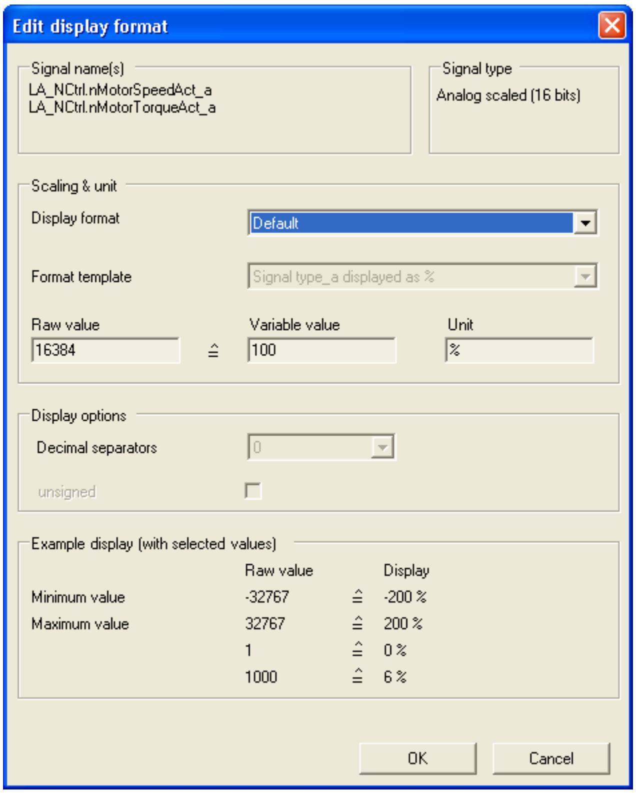
     Display masked out connections

     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.

16.4 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.

16.4.1 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.

Icon	Function
	Inserting a function block (📖 680)
	Inserting a system block (📖 682)
	Inserting a port block (📖 684)
	Inserting a comment (📖 686)

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) (📖 699)

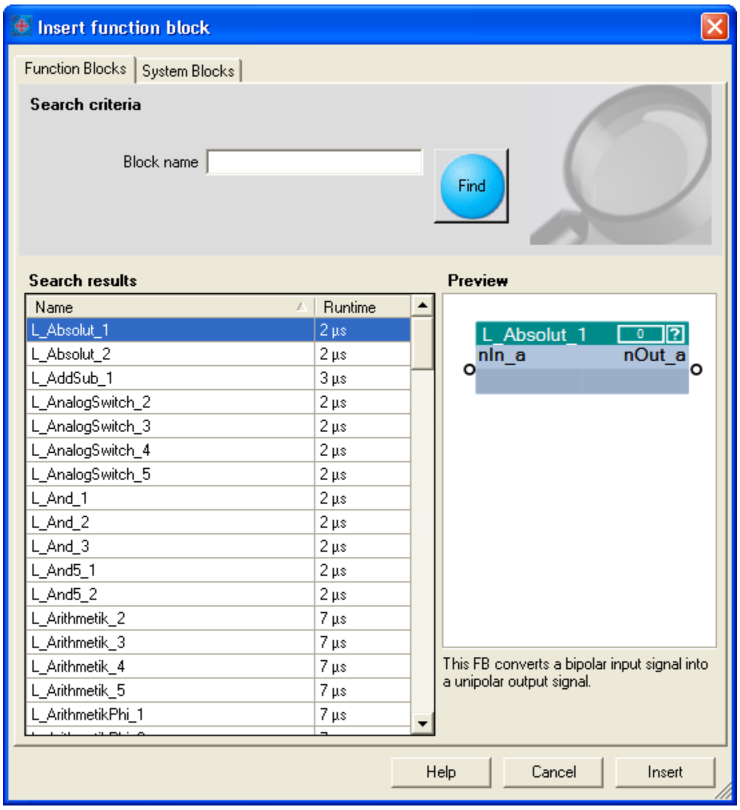
16.4.1.1 Inserting a function block

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". (📖 710)
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. ▸ Changing connector visibilities (📖 689)
	Online display format...	Adapt the display format of the input and output data of the block individually for online monitoring. ▸ Change online display format (📖 677)
	Parameter...	Open the parameter list/parameterisation dialog for the block. • Only if function block is parameterisable.
	Help	Show online help for the block.

Related topics

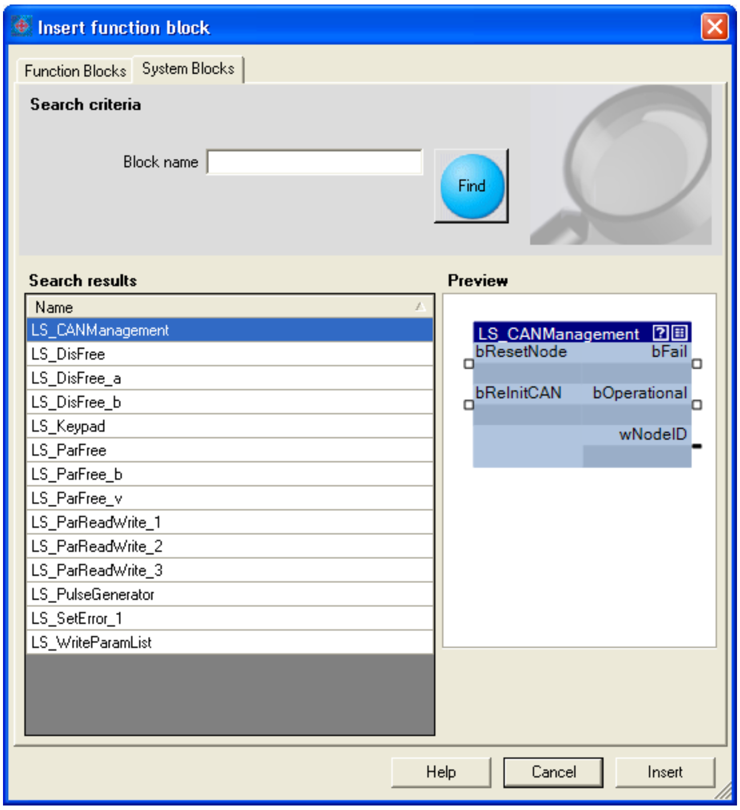
▸ Deleting objects that are no longer required (📖 688)
▸ Changing connector visibilities (📖 689)
▸ Arranging objects in the drawing area (📖 690)
▸ Creating/deleting connections (📖 691)
▸ Changing the processing order (📖 697)

16.4.1.2 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. ▸ Changing connector visibilities (📖 689)
	Online display format...	Adapt the display format of the input and output data of the block individually for online monitoring. ▸ Change online display format (📖 677)
	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 (📖 688)
▸ Changing connector visibilities (📖 689)
▸ Arranging objects in the drawing area (📖 690)
▸ Creating/deleting connections (📖 691)

16.4.1.3 Inserting a port block

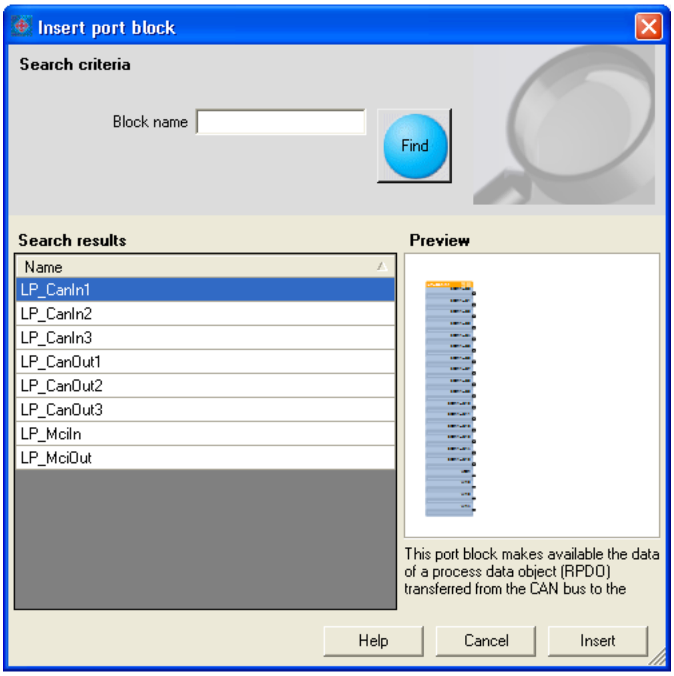
All input/output ports defined for the application on the **Ports** tab can be inserted into the interconnection in the form of port blocks in order to get access to the associated element variables.

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. ▸ Changing connector visibilities (📖 689)
	Online display format...	Adapt the display format of the input and output data of the block individually for online monitoring. ▸ Change online display format (📖 677)
	Parameter...	Open the parameter list/parameterisation dialog for the block.
	Help	Show online help for the block.

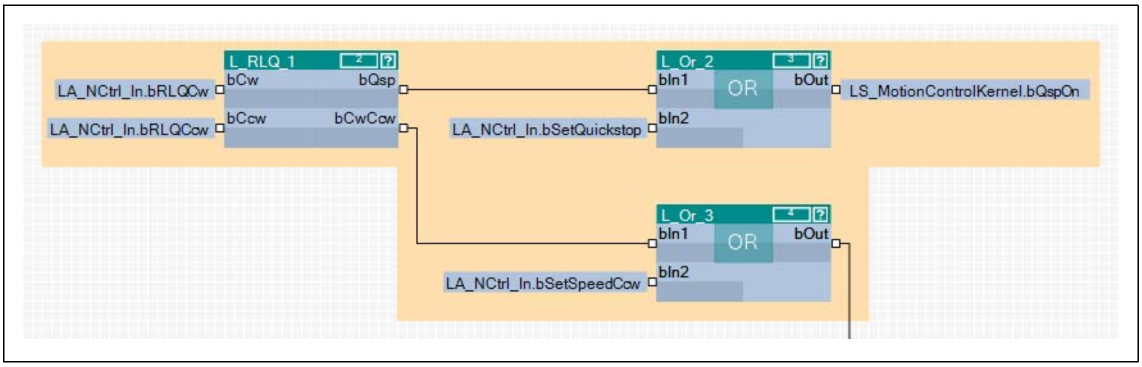
Related topics

▸ Deleting objects that are no longer required (📖 688)
▸ Changing connector visibilities (📖 689)
▸ Arranging objects in the drawing area (📖 690)
▸ Creating/deleting connections (📖 691)

16.4.1.4 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:



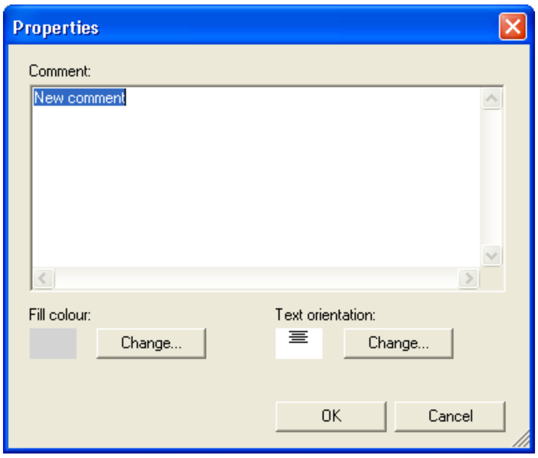
[16-7] Example: Graphical arrangement of FBs by means of two comments that overlap.

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 (📖 688)
- Arranging objects in the drawing area (📖 690)
- Creating/deleting connections (📖 691)

16.4.1.5 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

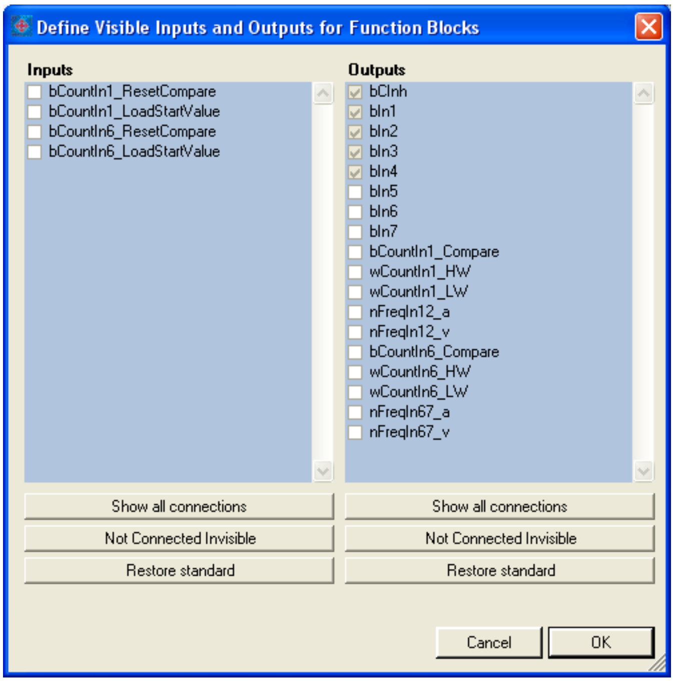
▶ Deleting connections that are no longer required (📖 696)

16.4.2 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.

16.4.3 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!

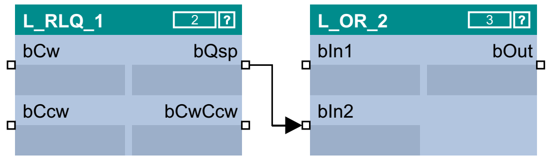
Arrange the objects so that no overlap occurs.

16.4.4 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.

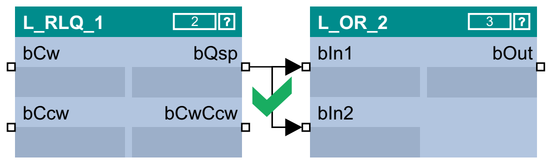
- An output represents a possible source in the interconnection.
- An input represents a possible target in the interconnection.



Permissible/impermissible connections

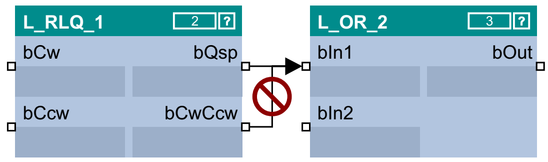
Several connections can lead from one output.

- Therefore it is always possible to start a new connection from an output.



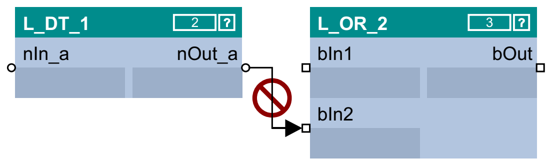
However, maximally one connection may end in an input.

- Therefore it is only possible to start a new connection from an input if there is no connection already ending in this input.



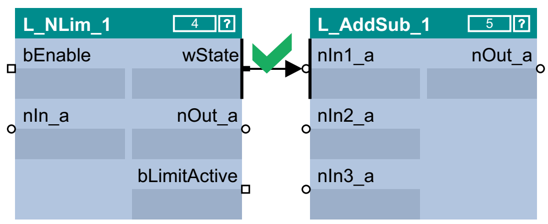
Only inputs/outputs of the same signal type can be connected.

- 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.

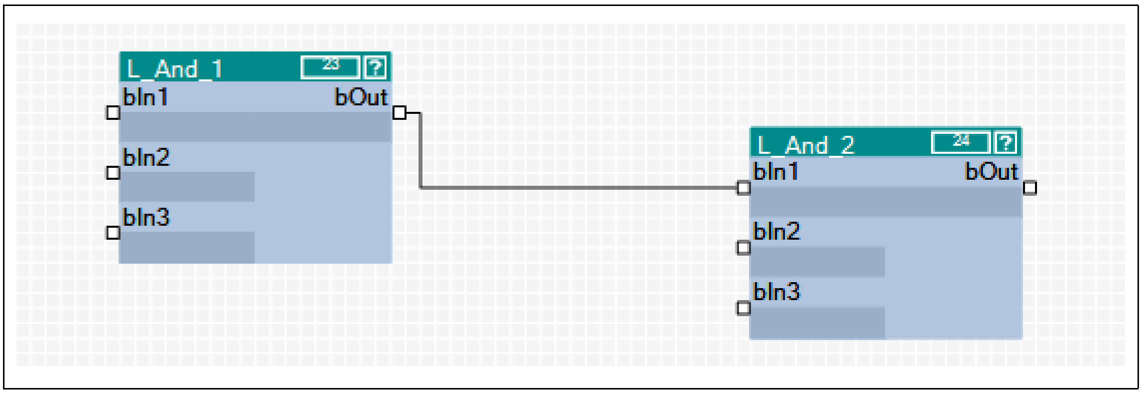
- The implicit type conversion is indicated by a vertical black bar at the port symbol.



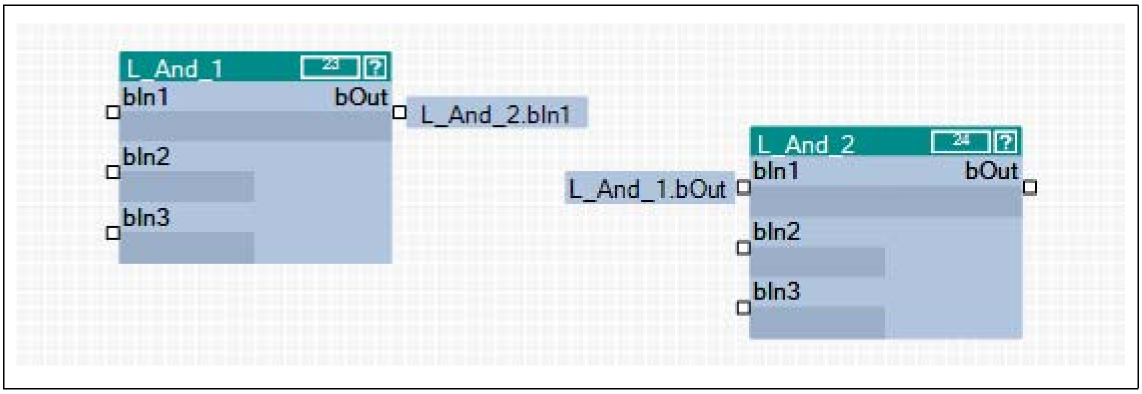
From the »Engineer« V2.13 "Analog/scaled" (_a) and "Angular velocity" (_v) signal types can also be interconnected.

Connection types

Connections can either be created by means of connection lines or port identifiers ("flags")



[16-8] Example 1: Connection via connection line



[16-9] 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.

16.4.4.1 Creating a connection using the connection line

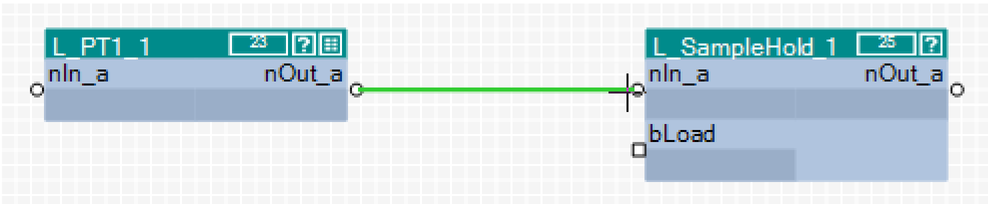
How to create a connection using the connection line:

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.

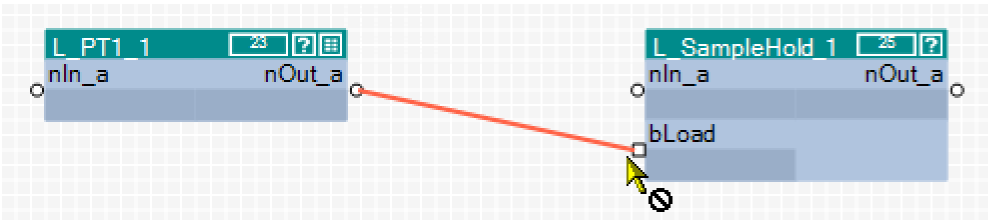
Tip!

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):

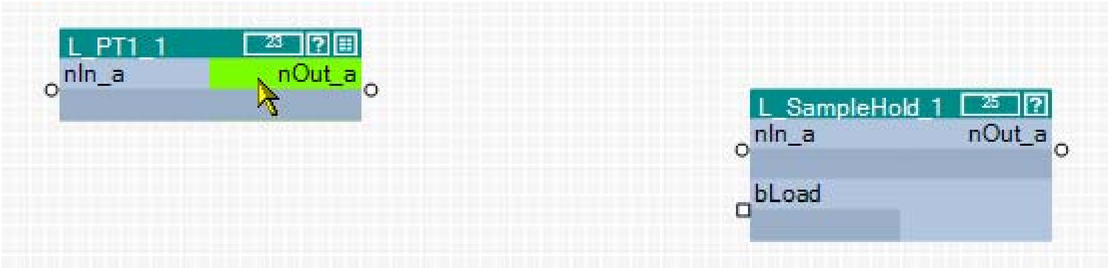


The command **Show as flag** in the *context menu* of a line serves to change the representation of the connection at any time.

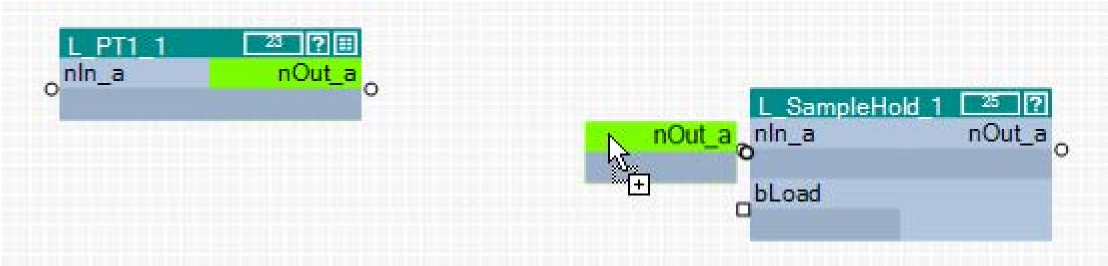
16.4.4.2 Creating a connection using port identifiers

How to create a connection with port identifiers:

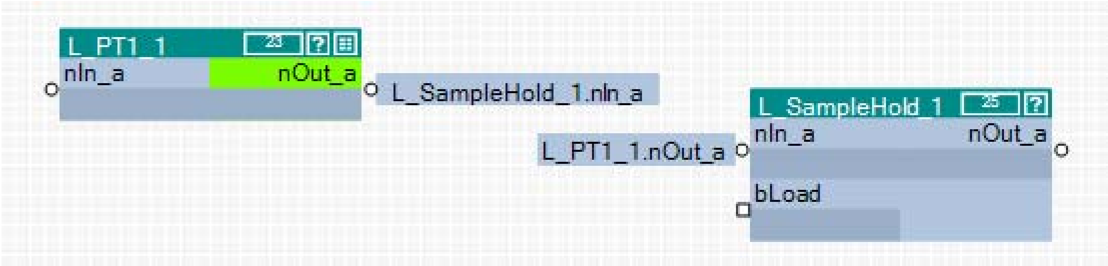
1. Click the port identifier.
   - The selected port is highlighted in light green:



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!

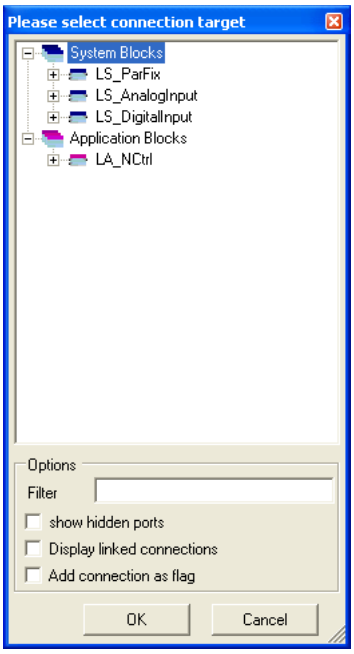
The command **Show as line** in the *context menu* of a flag serves to change the representation of the connection at any time.

16.4.4.3 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.

16.4.4.4 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>**.

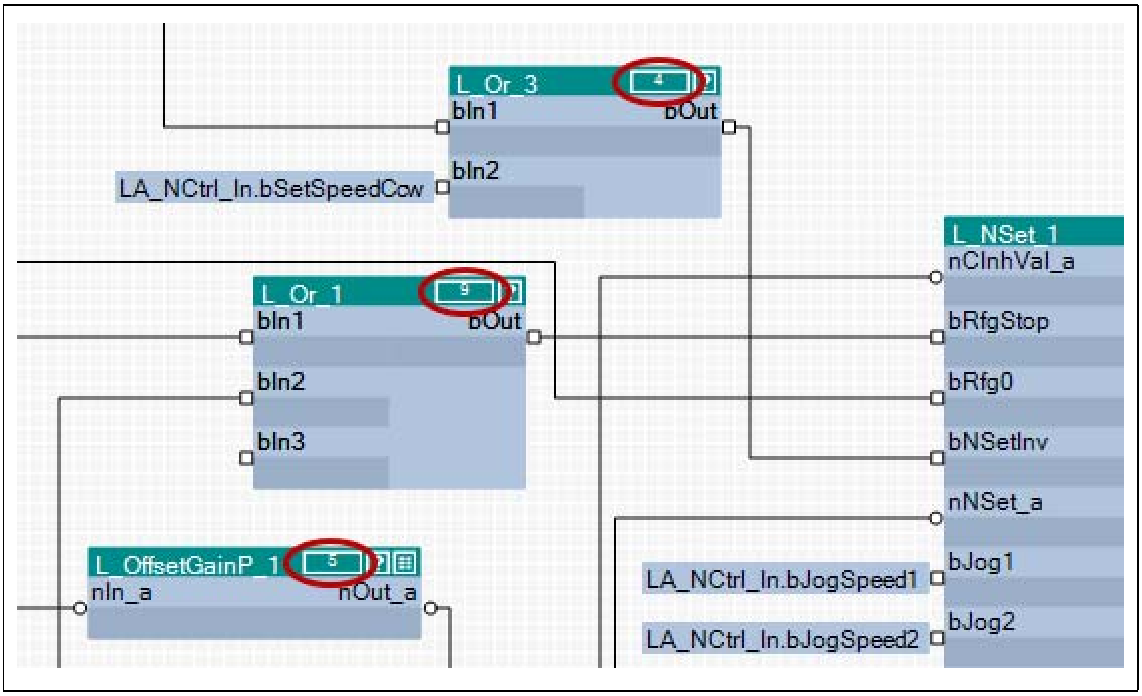
Related topics

▸ Deleting objects that are no longer required (📖 688)

16.4.5 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.



[16-10] Example: Function blocks with order index

Note!

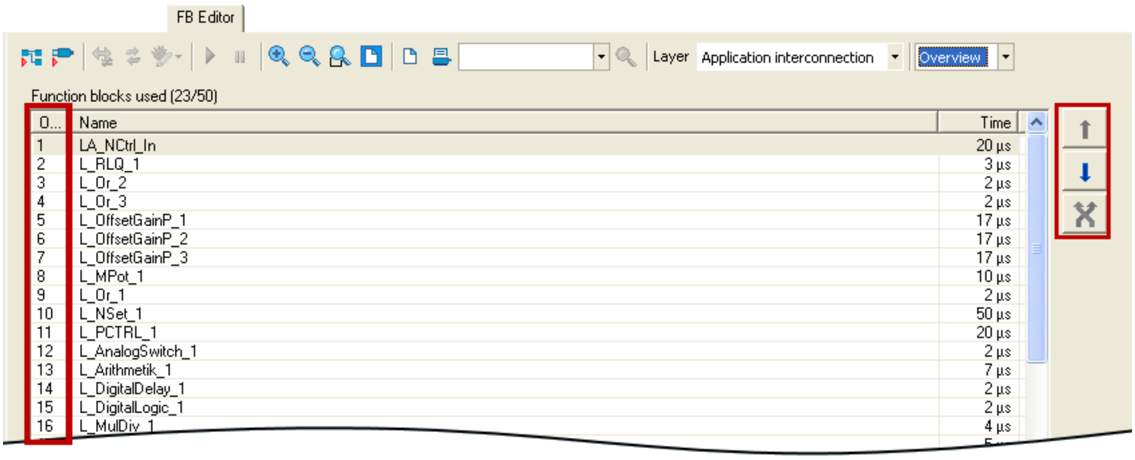
When a function block is shifted, its order index is maintained.

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.

16.4.6 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.

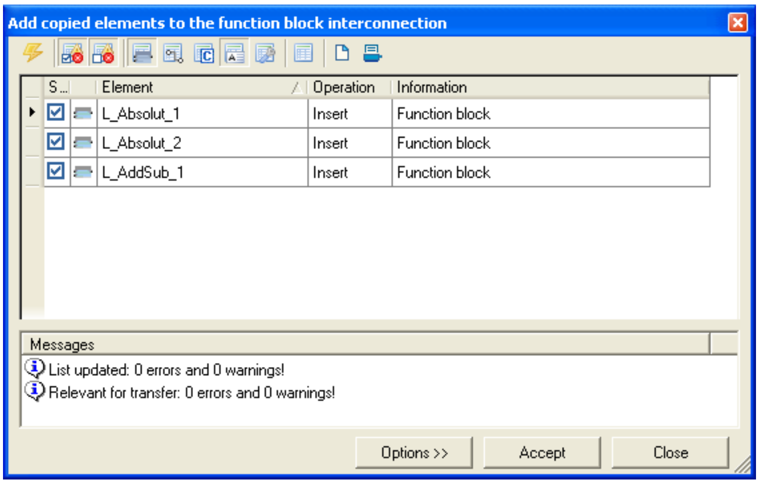
How to copy one or several interconnection elements:

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". (701)
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.

16.4.6.1 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:

Icon	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 • 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 added elements are simultaneously deleted from the list. The connections are added depending on the selected option.
Close	Close 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.

16.4.7 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:

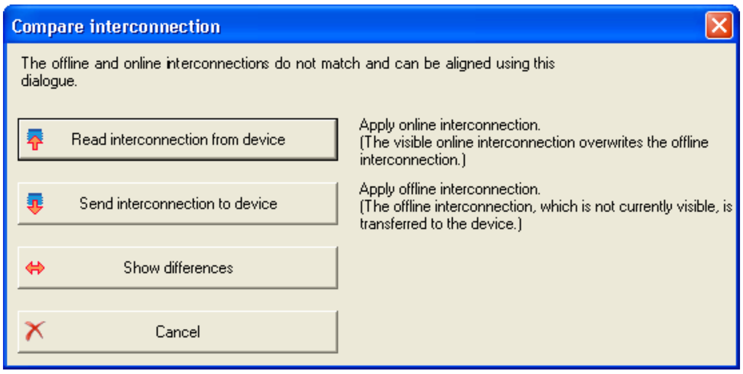
1. Go to the **Application parameters** tab.
2. Select the required application in the **Application** list field.

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.

16.5 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.

16.6 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.

16.7 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.

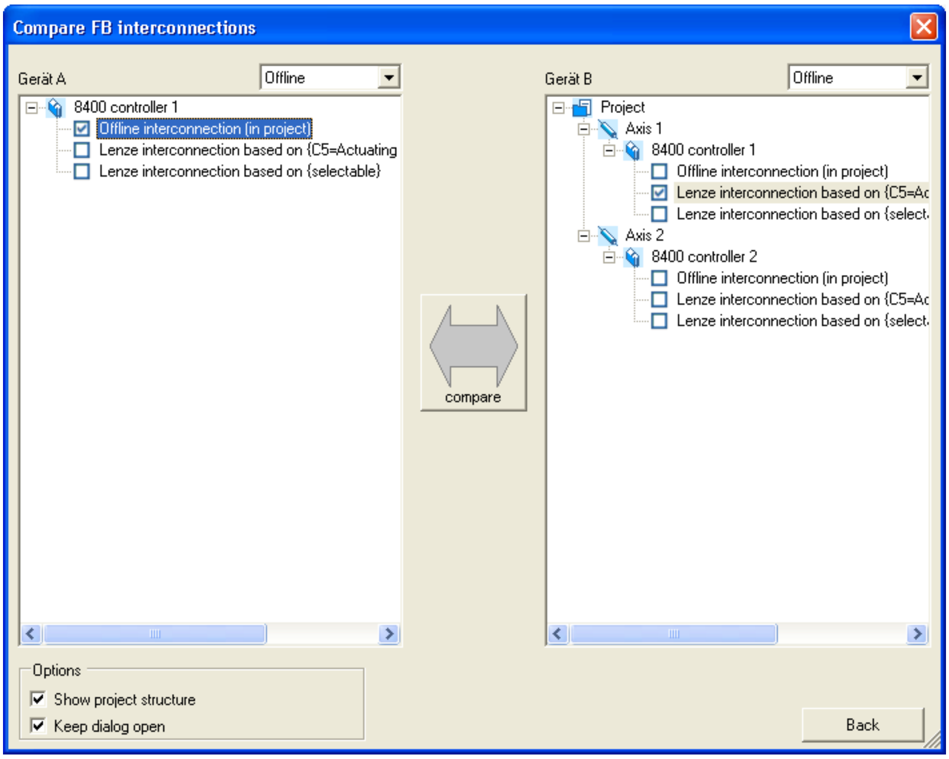
Note!

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:

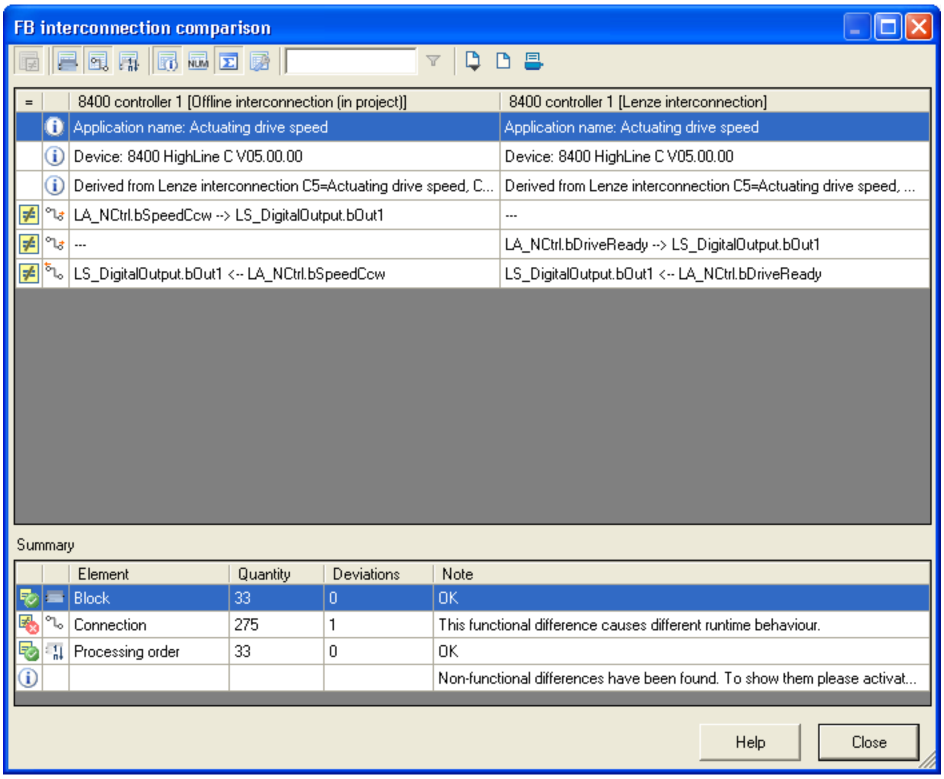
1. Select the command **Application data→Compare FB interconnections...**.
   - The *Compare FB interconnections* dialog box is displayed:



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.

Icon	Function
	Only show differences • Button can only be activated in expert mode.
	Show blocks
	Show connections
	Show processing order
	Show properties of blocks and connections • Function is only available in expert mode.
	Show comments • Function is only available in expert mode.
	Show block parameters • Function is only available in expert mode.
	Show application parameters • Function is only available in expert mode.
	Show all • Function is only available in expert mode.
	Show general information
	Show summary

Icon	Function
	Activate expert mode • In the expert mode, also non-functional differences are shown.
	Apply filter • Only show list entries which contain the text entered in the input field. • Function is only available in expert mode.
	Export shown list as comma-separated list (*.csv)
	Print view
	Print list

16.8 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.

16.9 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.

17 Function library

17.1 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 following chapter "System blocks". (📖 789)

Overview of function blocks available

Function block	Runtime	Function
L_Absolute_1	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 L_AnalogSwitch_2 L_AnalogSwitch_3	2 µs	... switches between two analog input signals.
L_And_1 L_And_2 L_And_3	2 µs	... ANDs three binary signals.
L_Arithmetik_1	7 µs	... combines two analog signals arithmetically.
L_Compare_1 L_Compare_2	5 µs	... compares two analog signals and can be used e.g. to implement a trigger.
L_DFlipFlop_1	1 µs	... provides two stable states depending on the input signals.
L_DigitalDelay_1	2 µs	... delays binary signals.
L_DigitalLogic_1	2 µs	... provides a binary output signal which is generated by the logic combination of three input signals.
L_GainOffset_1 L_GainOffset_2 L_GainOffset_3	3 µs	... can amplify an analog input signal and then add an offset to it. • Gain and offset can be set via FB inputs.
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_MPot_1	10 µs	... replaces a hardware motor potentiometer as setpoint source.
L_MulDiv_1	4 µs	... multiplies the analog input signal with a factor.
L_Negation_1	2 µs	... negates an analog input signal.
L_Not_1 L_Not_2 L_Not_3	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_OffsetGain_1 L_OffsetGain_2	4 µs	... can add an offset to an analog input signal and amplify it afterwards. • Offset and gain can be set via FB inputs.
L_OffsetGainP_1 L_OffsetGainP_2 L_OffsetGainP_3	17 µs	... can add an offset to an analog input signal and amplify it afterwards. • Offset and gain can be set via parameters.
L_Or_1 L_Or_2 L_Or_3	2 µs	... ORs three binary signals.

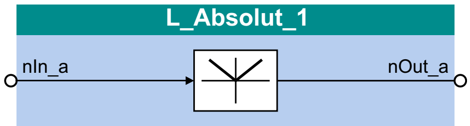
Function block	Runtime	Function
L_PCTRL_1	20 µs	... is a PID controller and can be used for various control tasks.
L_PT1_1	1 µs	... filters and delays analog signals.
L_RLQ_1	3 µs	... links a selected direction of rotation to the QSP function with wire-break protection.
L_SignalMonitor_a	15 µs	... serves 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_SMControlDecoder_1	2 µs	... decodes the control word of the integrated safety system into individual binary control signals for further processing in the application.
L_SMStateDecoder_1	2 µs	... decodes the status word of the integrated safety system into individual binary status signals for further processing in the application.
L_SMStateDecoderIO_1	2 µs	... decodes the I/O status word of the integrated safety system into individual binary status signals for further processing in the application.
L_Transient_1 L_Transient_2 L_Transient_3 L_Transient_4	3 µs	... evaluates digital signal edges and converts them into timed pulses.

Related topics:

- Overview of system blocks available (📖 789)
- Working with the FB Editor (📖 658)

17.1.1 L_Absolut_1

This FB converts a bipolar input signal into a unipolar output signal.



Inputs

Identifier Data type	Information/possible settings
nIn_a INT	Input signal

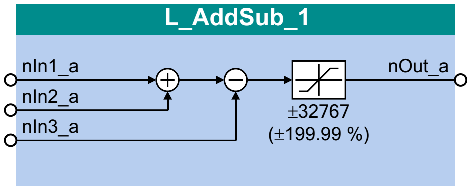
Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal

17.1.2 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

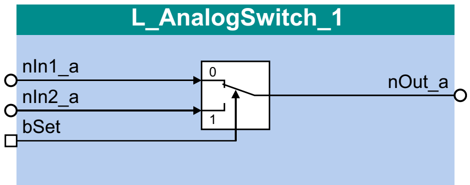
Identifier Data type	Information/possible settings
nIn1_a INT	Input signal 1 • This input is added
nIn2_a INT	Input signal 2 • This input is added
nIn3_a INT	Input signal 3 • This input is subtracted

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal • *nOut_a = nIn1_a + nIn2_a - nIn3_a* • Internal limitation to ±32767 (±199.99 %)

17.1.3 L_AnalogSwitch_1

This function block switches between two analog input signals. The switching is controlled via a boolean input signal.



Inputs

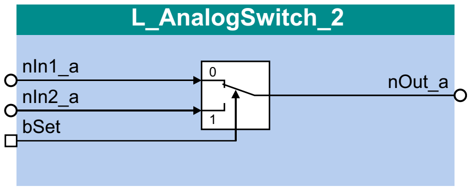
Identifier (Data type)	Information/possible settings	
nIn1_a (INT)	Input signal 1	
nIn2_a (INT)	Input signal 2	
bSet (BOOL)	Selection of the input signal for the output to *nOut_a*	
	FALSE	*nIn1_a*
	TRUE	*nIn2_a*

Outputs

Identifier (Data type)	Value/meaning
nOut_a (INT)	Output signal

17.1.4 L_AnalogSwitch_2

This function block switches between two analog input signals. The switching is controlled via a boolean input signal.



Inputs

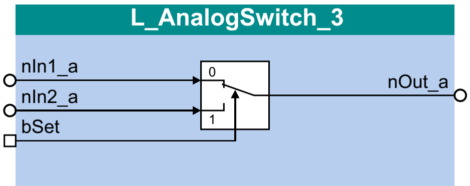
Identifier Data type	Information/possible settings	
nIn1_a INT	Input signal 1	
nIn2_a INT	Input signal 2	
bSet BOOL	Selection of the input signal for the output to *nOut_a*	
	FALSE	*nIn1_a*
	TRUE	*nIn2_a*

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal

17.1.5 L_AnalogSwitch_3

This function block switches between two analog input signals. The switching is controlled via a boolean input signal.



Inputs

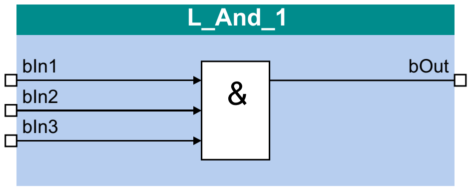
Identifier Data type	Information/possible settings	
nIn1_a INT	Input signal 1	
nIn2_a INT	Input signal 2	
bSet BOOL	Selection of the input signal for the output to *nOut_a*	
	FALSE	*nIn1_a*
	TRUE	*nIn2_a*

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal

17.1.6 L_And_1

This FB implements the ANDing of the input signals.



Inputs

Identifier Data type	Information/possible settings
bIn1 bIn2 bIn3 BOOL	Input signal

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output signal

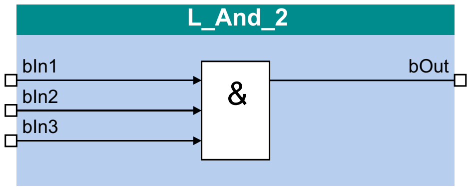
Function

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

[17-1] Truth table of the FB L_And_1

17.1.7 L_And_2

This FB implements the ANDing of the input signals.



Inputs

Identifier / Data type	Information/possible settings
bIn1 bIn2 bIn3 BOOL	Input signal

Outputs

Identifier / Data type	Value/meaning
bOut BOOL	Output signal

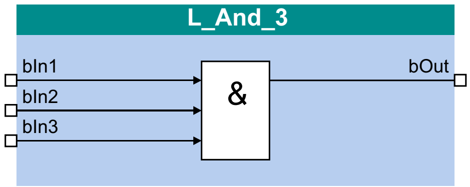
Function

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

[17-2] Truth table of the FB L_And_2

17.1.8 L_And_3

This FB implements the ANDing of the input signals.



Inputs

Identifier Data type	Information/possible settings
bIn1 bIn2 bIn3 BOOL	Input signal

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output signal

Function

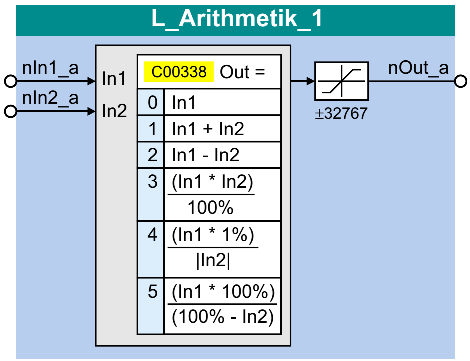
Inputs			Output
bIn3	**bIn2**	**bIn1**	**bOut**
FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	TRUE	
FALSE	TRUE	FALSE	
FALSE	TRUE	TRUE	
TRUE	FALSE	FALSE	
TRUE	FALSE	TRUE	
TRUE	TRUE	FALSE	
TRUE	TRUE	TRUE	TRUE

[17-3] Truth table of the FB L_And_3

17.1.9 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 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 • Internal limitation to ±32767 (±199.99 %)

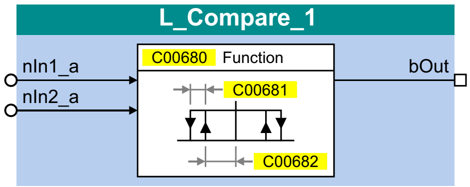
Parameter

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	$nOut\_a = \frac{nIn1\_a}{16384 - nIn2\_a} \cdot 16384$	

17.1.10 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 (Data type)	Information/possible settings
nIn1_a (INT)	Input signal 1
nIn2_a (INT)	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.

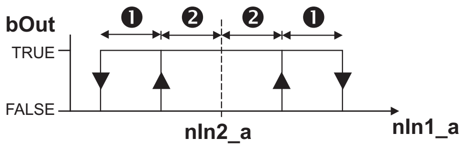
Parameter

Parameter	Possible settings			Info
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 • Lenze setting: 0.50 %
C00682	0.00	%	100.00	Window • Lenze setting: 2.00 %

17.1.10.1 Function 1: nIn1 = nIn2

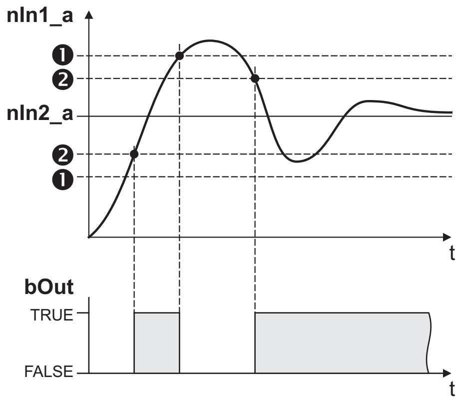
This function compares two signals with regard to equality. It can, for instance, provide the comparison "actual speed equals setpoint speed " ($n_{act} = n_{set}$).

- 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

[17-4] Function 1: Switching performance

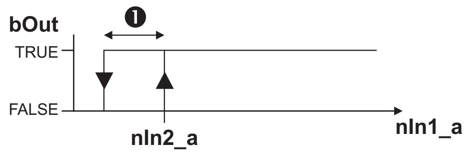


❶ C00681: Hysteresis
❷ C00682: Window

[17-5] Function 1: Example

17.1.10.2 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.

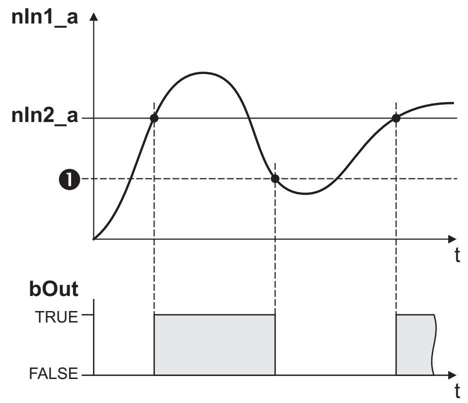


❶ C00681: Hysteresis

[17-6] 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.

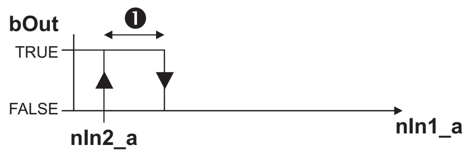


❶ C00681: Hysteresis

[17-7] Function 2: Example

17.1.10.3 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.

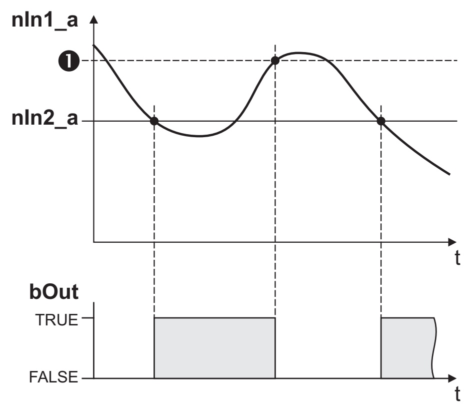


❶ C00681: Hysteresis

[17-8] 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

[17-9] Function 3: Example

17.1.10.4 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

17.1.10.5 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

17.1.10.6 Function 6: |nIn1| < |nIn2|

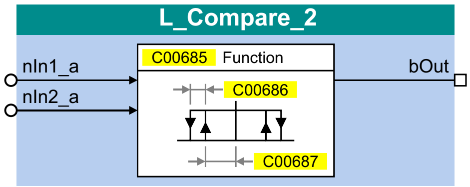
This function serves to implement the comparison "$|n_{act}| < |n_x|$" independent of the direction of rotation. This function is similar to function 3. However, the amount is generated by the input signals before signal processing (without sign).

▶ Function 3: nIn1 < nIn2

17.1.11 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 Data type	Information/possible settings
nIn1_a INT	Input signal 1
nIn2_a INT	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.

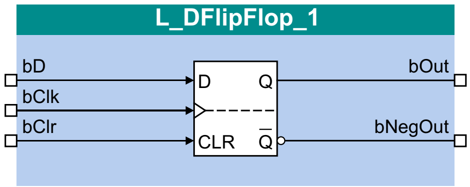
Parameter

Parameter	Possible settings			Info
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 • Lenze setting: 0.50 %
C00687	0.00	%	100.00	Window • Lenze setting: 2.00 %

For a detailed functional description see L_Compare_1.

17.1.12 L_DFlipFlop_1

The FB saves binary signals (DFlipFlop) in a clock-controlled way.



Inputs

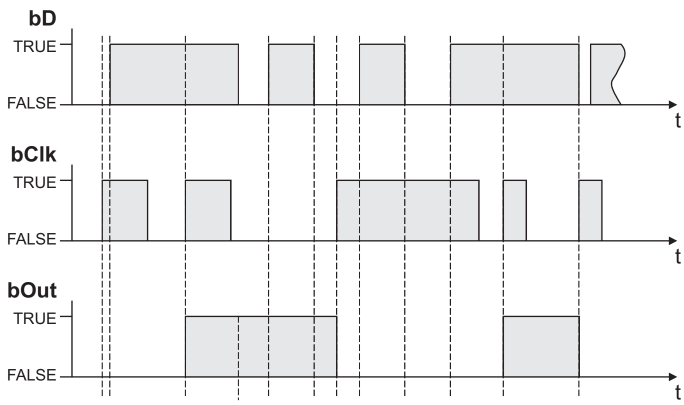
Identifier Data type	Information/possible settings
bD BOOL	Data input
bClk BOOL	Clock input • Only FALSE/TRUE edges are evaluated
bClr BOOL	Reset input TRUE: • The *bOut* output is set to FALSE. • The *bNegOut* output is set to TRUE.

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output signal
bNegOut 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:



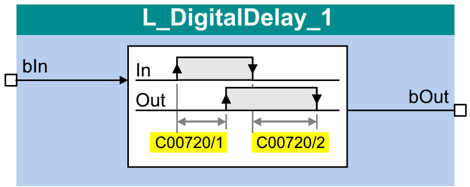
If the *bClr* input = TRUE:

- 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.

17.1.13 L_DigitalDelay_1

This FB delays binary signals.

- ON and OFF-deceleration can be parameterised separately.



Inputs

Identifier Data type	Information/possible settings
bIn BOOL	Input signal

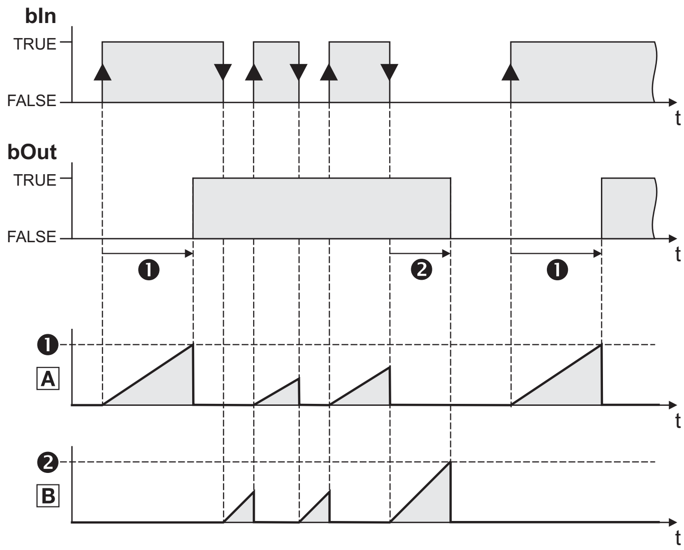
Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output signal (time-delayed input signal)

Parameter

Parameter	Possible settings			Info
C00720/1	0.000	s	3600.000	ON-deceleration • Lenze setting: 0.000 s
C00720/2	0.000	s	3600.000	OFF-deceleration • Lenze setting: 0.000 s

Function



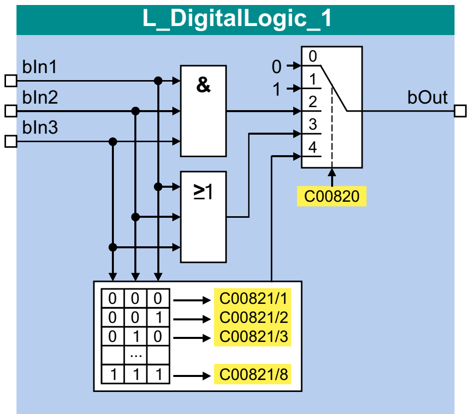
❶ ON delay
❷ OFF delay
A Internal timing element for ON delay (retriggerable)
B Internal timing element for OFF delay (retriggerable)

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*.

17.1.14 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
- AND operation of the inputs
- OR operation of the inputs
- Output depending on the combination of the input signals



Inputs

Identifier Data type	Information/possible settings
bIn1 BOOL	Input signal 1
bIn2 BOOL	Input signal 2
bIn3 BOOL	Input signal 3

Outputs

Identifier Data type	Value/meaning
bOut 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 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)

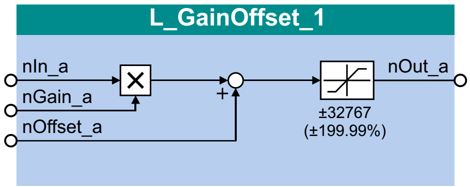
Example: If in case of the signal combination *bIn1* = FALSE, *bIn2* = FALSE and *bIn3* = TRUE, the output signal *bOut* is to be = TRUE, C00821/5 must be set to "TRUE":

bIn3	bIn2	bIn1	Output signal bOut
TRUE	FALSE	FALSE	C00821/5 (TRUE)

17.1.15 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 Data type	Information/possible settings
nIn_a INT	Input signal • Scaling: 16384 ≡ 100 %
nGain_a INT	Gain factor • Scaling: 16384 ≡ 100 % • 199.99 % ≈2
nOffset_a INT	Offset • Scaling: 16384 ≡ 100 %

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal • Internal limitation to ±199.99 %

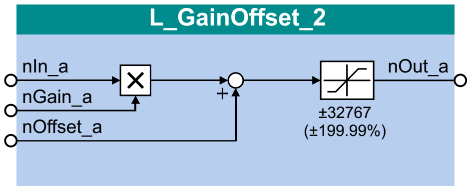
Function

$$nOut\_a = (nIn\_a \cdot Gain\ factor) + Offset$$

17.1.16 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 Data type	Information/possible settings
nIn_a INT	Input signal • Scaling: 16384 ≡ 100 %
nGain_a INT	Gain factor • Scaling: 16384 ≡ 100 % • 199.99 % ≈2
nOffset_a INT	Offset • Scaling: 16384 ≡ 100 %

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal • Internal limitation to ±199.99 %

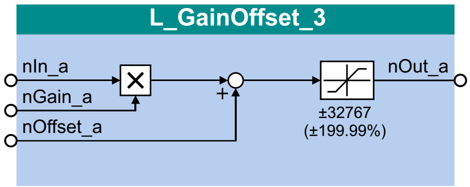
Function

$$\text{nOut\_a} = (\text{nIn\_a} \cdot \text{Gain factor}) + \text{Offset}$$

17.1.17 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 Data type	Information/possible settings
nIn_a INT	Input signal • Scaling: 16384 ≡ 100 %
nGain_a INT	Gain factor • Scaling: 16384 ≡ 100 % • 199.99 % ≈2
nOffset_a INT	Offset • Scaling: 16384 ≡ 100 %

Outputs

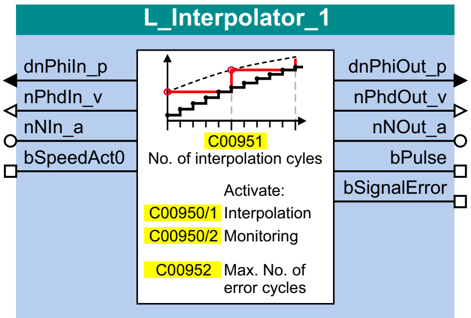
Identifier Data type	Value/meaning
nOut_a INT	Output signal • Internal limitation to ±199.99 %

Function

$$\text{nOut\_a} = (\text{nIn\_a} \cdot \text{Gain factor}) + \text{Offset}$$

17.1.18 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 DINT	Position setpoint • Is interpolated and completed when signal interpolation is activated.	
nPhdIn_v INT	Angular velocity • Is only passed through to the *nPhdOut_v* output.	
nNIn_a INT	Analog value • Is interpolated when signal interpolation is activated.	
bSpeedAct0 BOOL	Input for detecting the "Current speed is zero" status • This status signal needs to be transmitted by the setpoint source to ensure trouble-free operation.	
	TRUE	Current speed is zero.

Outputs

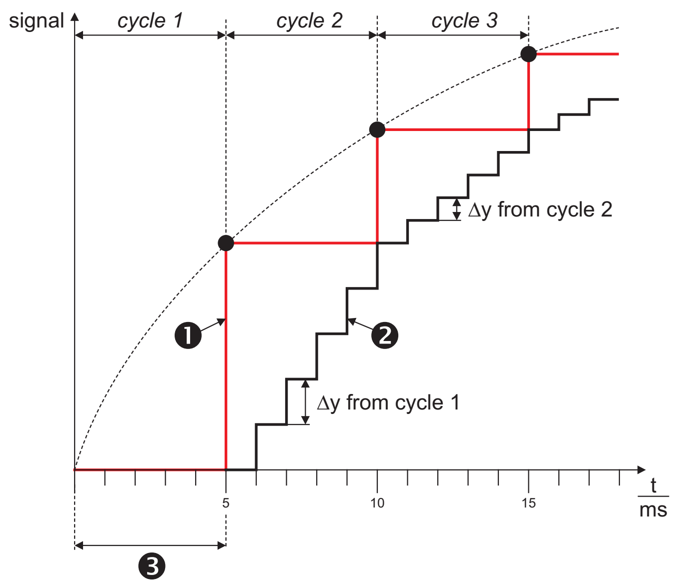
Identifier Data type	Value/meaning	
dnPhOut_p DINT	Output of the *dnPhIn_p* position setpoint which, if applicable, has been interpolated and completed	
nPhdOut_v INT	Output of the *nPhdIn_v* angular velocity	
nNOut_a INT	Output of the *nNIn_a* analog value which, if applicable, has been interpolated	
bPulse BOOL	"Input values have been accepted" status signal	
	TRUE	The input values have been accepted during this cycle.
bSignalError BOOL	"Signal error" status signal • 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 • Lenze setting: Off ▸ Signal interpolation (📖 737)
	0	Off		
	1	On		
C00950/2				Signal monitoring of the *dnPhIn_p* input signal • Lenze setting: Off ▸ Signal monitoring (📖 738)
	0	Off		
	1	On		
C00951	1		65535	Number of interpolation steps • Corresponds to the receive cycle of the data telegrams in [ms]. • Lenze setting: 1
C00952	0		65535	Limit value for missing data telegrams • Lenze setting: 5 ▸ Signal monitoring (📖 738)
C00953	0		0	Filters In preparation!

17.1.18.1 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)

[17-10] Signal characteristic

Note!

Do not change the number of interpolation steps during operation. Otherwise the interpolation becomes inaccurate.

17.1.18.2 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.

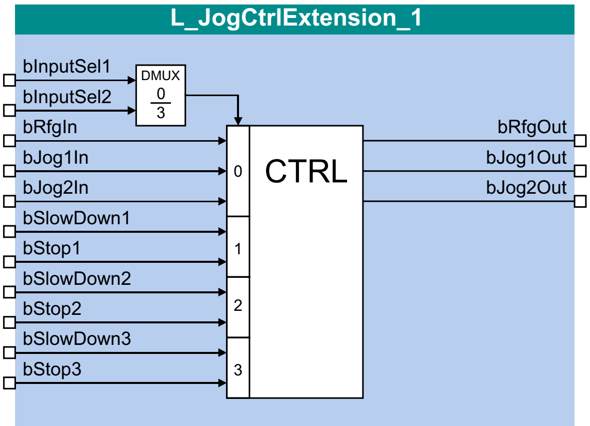
Note!

The *nNIn_a* analog value is not monitored!

17.1.19 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 Data type	Information/possible settings
bInputSel1 bInputSel2 BOOL	Activation of the *bSlowDown1/bStop1*, *bSlowDown2/bStop2* and *bSlowDown3/bStop3* signal pairs according to the Truth table
bRfgIn BOOL	Ramping down of the setpoint generator in the downstream L_NSet FB according to the Truth table
bJog1In bJog2In BOOL	Selection inputs for setting fixed speeds in the setpoint generator • 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. • To achieve the desired behaviour (starting at high speed, pre-switch off at low speed), both inputs must be set to TRUE. • 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. • 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 bSlowDown2 bSlowDown3 BOOL	Activation of fixed setpoint 2 in the downstream L_NSet FB • These inputs only fulfil a function if they have been activated via *bInputSel1* and *bInputSel2* previously (see Truth table).
bStop1 bStop2 bStop3 BOOL	Ramping down of the ramp function generator in the downstream L_NSet FB • These inputs only fulfil a function if they have been activated via *bInputSel1* and *bInputSel2* previously (see Truth table).

Outputs

Identifier Data type	Value/meaning
bRfgOut BOOL	Control signal for ramping down the setpoint generator • Connect this output to the *bRfg0* input of the L_NSet FB.
bJog1Out BOOL	Control signal for setting fixed speeds in the setpoint generator • Connect this output to the *bJog1* input of the L_NSet FB.
bJog2Out BOOL	Control signal for setting fixed speeds in the setpoint generator • Connect this output to the *bJog2* input of the L_NSet FB.

Truth table

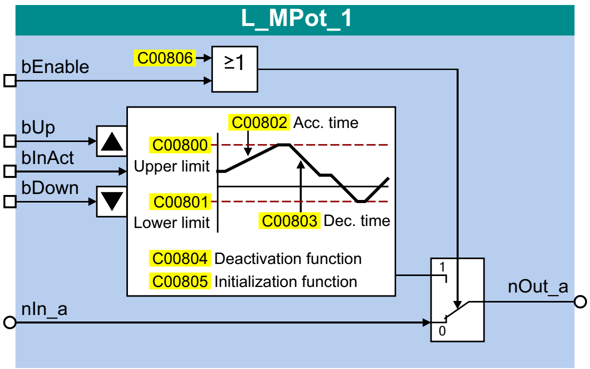
Input		Function	Response in the L_NSet FB
bInputSel1	**bInputSel2**		
FALSE	FALSE	Pre-switch off inactive	No response • The *bRfgIn* input signal is directly provided at the *bRfgOut* output. • The *bJogIn1* and *bJogIn2* input signals are directly output at the *bJog1Out* and *bJog2Out* outputs.
TRUE	FALSE	The *bSlowDown1* and *bStop1* inputs are evaluated.	**Pre-switch off can be activated** • 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. • 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.	

[17-1] Truth table for activating the pre-switch off

17.1.20 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 Data type	Information/possible settings	
bEnable BOOL	Switch over motor potentiometer function *bEnable* input and C00806 code are ORed.	
	TRUE	Motor potentiometer function is active, setpoint can be changed via *bUp* and *bDown*. • 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 INT	When bEnable = FALSE, the analog nIn_ is input signal switched to the nOut_a output.	
bUp BOOL	Approaching of the upper speed limit value set in C00800.	
	TRUE	The *nOut_a* output signal runs to its upper limit value (*nHighLimit*). • If the *bDown* input is simultaneously set to TRUE, the *nOut_a* output signal is not changed.
bDown BOOL	Approaching of the lower speed limit value set in C00801.	
	TRUE	The *nOut_a* output signal runs to its lower limit value (*nLowLimit*). • If the *bUp* input is simultaneously set to TRUE, the *nOut_a* output signal is not changed.
bInAct BOOL	Deactivate motor potentiometer function • This input has the highest priority. • 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 Data type	Value/meaning
nOut_a INT	Output signal

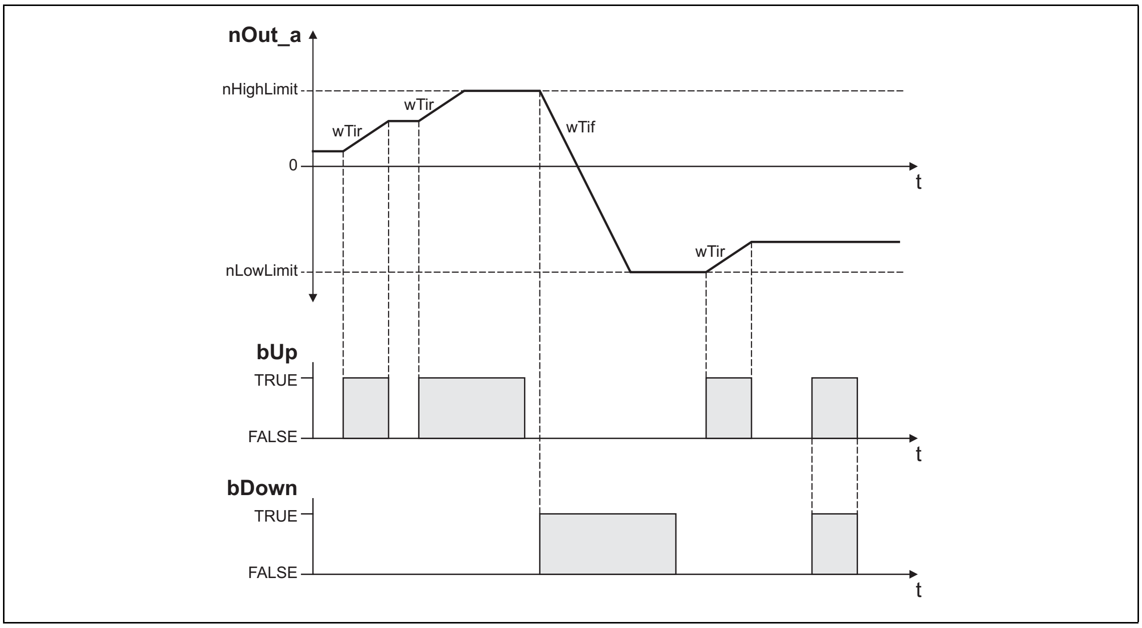
Parameter

Parameter	Possible settings			Info
C00800	-199.99	%	199.99	Upper limit • Lenze setting: 100.00 %
C00801	-199.99	%	199.99	Lower limit • Lenze setting: -100.00 %
C00802	0.1	s	6000.0	Acceleration time • Lenze setting: 10.0 s
C00803	0.1	s	6000.0	Deceleration time • Lenze setting: 10.0 s
C00804				Inactive function • Selection of response when deactivating the motor potentiometer via the input *bInAct*. • 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 • Selection of response when switching on the device. • 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 • When switching to 1: YES, the value applied to *nIn_a* is automatically transferred to the motor potentiometer. • Lenze setting: 0
	0	No		
	1	Yes		

17.1.20.1 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*.

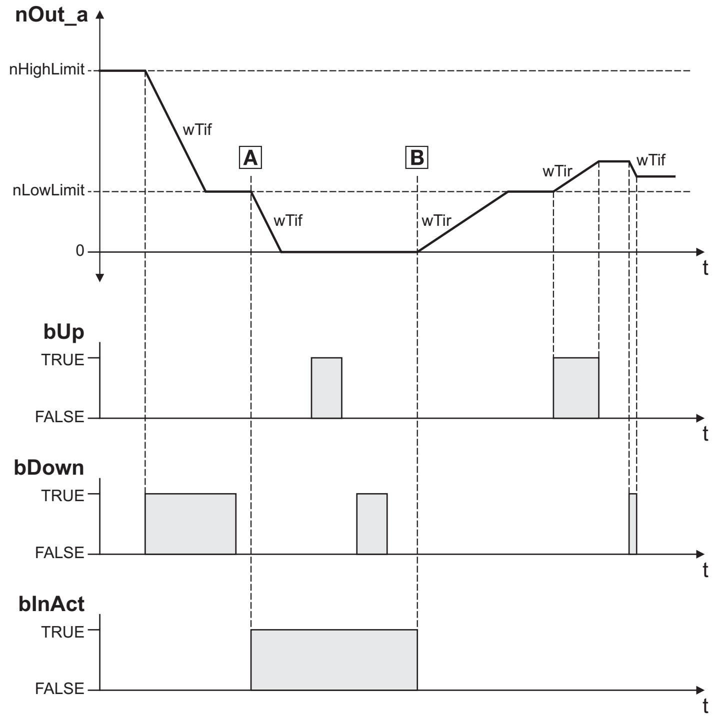


[17-11] Example: Control of the motor potentiometer

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*.

17.1.20.2 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*.



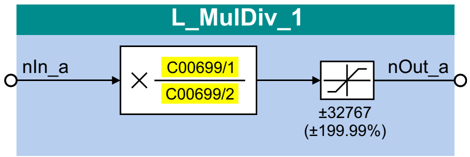
A When the motor potentiometer is deactivated, *nOut_a* is reduced to 0 % with the deceleration time *wTif*.
B If it is activated again, *nOut_a* is increased again to *nLowLimit* with the acceleration time *wTir*.

[17-12] Example: Deactivation of the motor potentiometer when the *Function* = 1 has been selected

17.1.21 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 Data type	Information/possible settings
nIn1 INT	Input signal

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Product value (result of the multiplication) • Internal limitation to ± 32767

Parameter

Parameter	Possible settings			Info
C00699/1	-32767		32767	Counter
C00699/2	-32767		32767	Denominator

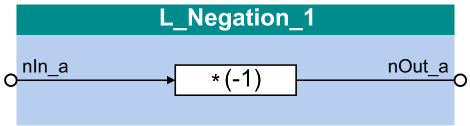
Function

$$\text{nOut\_a} = \text{nIn\_a} \times \frac{\text{C00699/1}}{\text{C00699/2}}$$

17.1.22 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

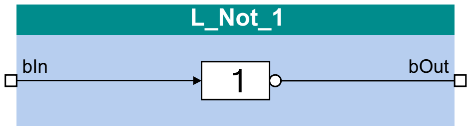
Identifier Data type	Value/meaning
nIn_a INT	Input signal

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal

17.1.23 L_Not_1

This FB negates a signal of BOOL data type.



Inputs

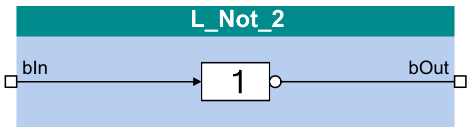
Identifier Data type	Value/meaning
bIn BOOL	Input signal

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Result of the NOT operation (negated input signal)

17.1.24 L_Not_2

This FB negates a signal of BOOL data type.



Inputs

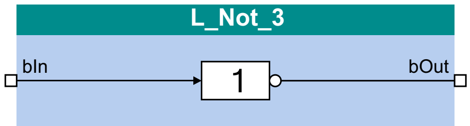
Identifier Data type	Value/meaning
bIn BOOL	Input signal

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Result of the NOT operation (negated input signal)

17.1.25 L_Not_3

This FB negates a signal of BOOL data type.



Inputs

Identifier Data type	Value/meaning
bIn BOOL	Input signal

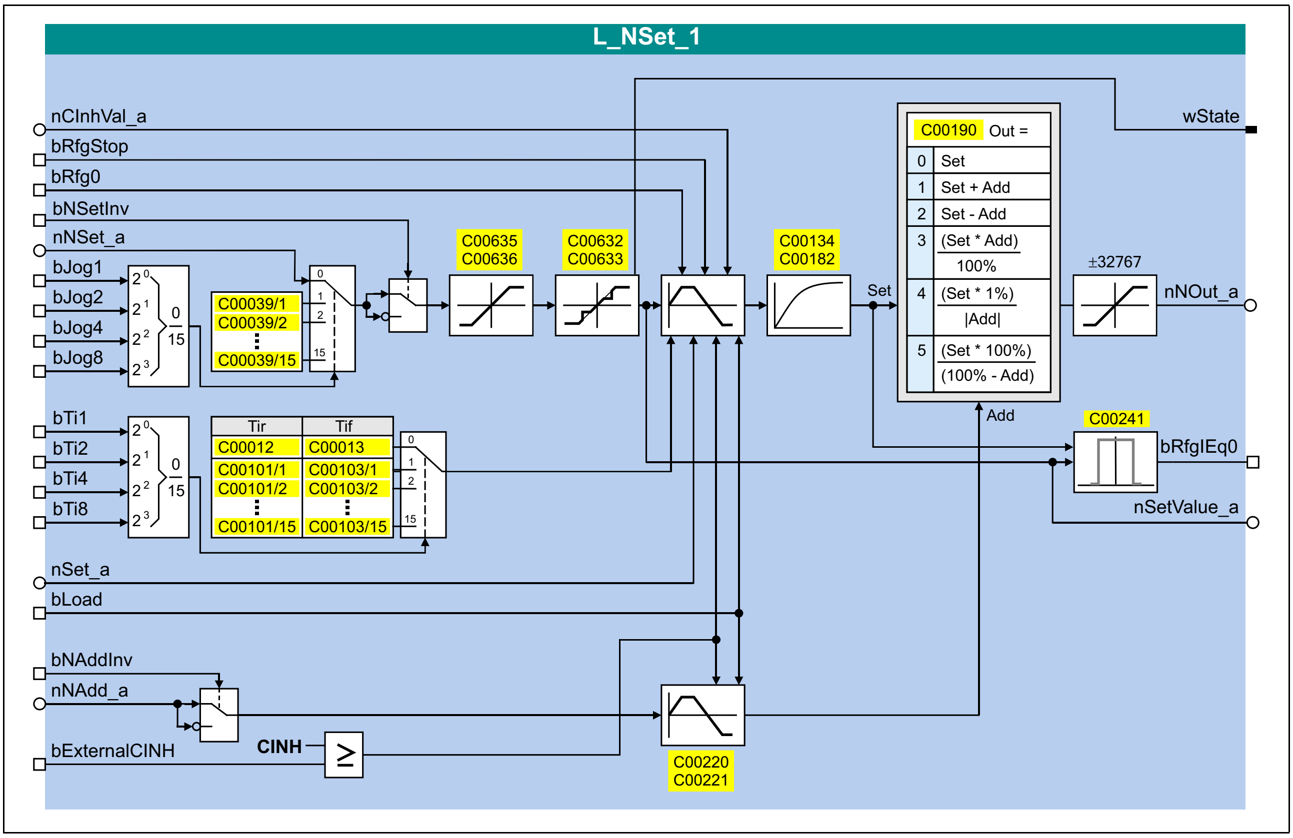
Outputs

Identifier Data type	Value/meaning
bOut BOOL	Result of the NOT operation (negated input signal)

17.1.26 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 INT	Main setpoint signal which is to be accepted by the main setpoint integrator when the controller is inhibited.	
bRfgStop BOOL	Holding (freezing) of the current value of the main setpoint integrator	
	TRUE	The current value of the main setpoint integrator is held.
bRfg0 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 BOOL	Signal inversion for the main setpoint	
	TRUE	Main setpoint signal is inverted.
nNset_a INT	Main setpoint signal • Other signals are also permitted	
bJog1 ... bJog8 BOOL	Selection inputs for fixed changeover setpoints (JOG setpoints) for the main setpoint • Selection inputs are binary coded.	
bTI1 ... bTI8 BOOL	Selection inputs for alternative acceleration/deceleration times for the main setpoint • Selection inputs are binary coded.	
nSet_a INT	Starting value which is loaded into the main setpoint integrator by setting *bLoad* to TRUE.	
bLoad 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 BOOL	Signal inversion for the additional setpoint	
	TRUE	Additional setpoint signal is inverted.
nNAdd_a INT	Additional setpoint signal • Other signals are also permitted	
bExternalCINH 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*. The additional setpoint integrator is set to "0". ▸ Application example for the additional load function (📖 759)

Outputs

Identifier Data type	Value/meaning
nNOut_a INT	Speed setpoint output signal • Scaling: 16384 ≡ 100 %
bRfgIEqO BOOL	Status signal "setpoint = 0"

Identifier Data type	Value/meaning	
wState WORD	Bit-coded status word • 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 INT	Speed-setpoint input signal of the ramp function generator • Scaling: 16384 ≡ 100 %	

Parameter

Parameter	Possible settings			Info
C00012	0.000	s	999.900	Acceleration time T_{ir} for the main setpoint • Lenze setting: 0.000 s
C00013	0.000	s	999.900	Deceleration time T_{if} for the main setpoint • Lenze setting: 0.000 s
C00039/1..15	-199.99	%	199.99	Fixed setpoints (JOG setpoints) • Lenze setting: 0.00 %
C00101/1..15	0.000	s	999.900	Alternative acceleration times (T_{ir}) for the main setpoint • Lenze setting: 0.000 s
C00103/1..15	0.000	s	999.900	Alternative deceleration times (T_{if}) for the main setpoint • Lenze setting: 0.000 s
C00134				Activates ramp rounding with PT1 behaviour for the main setpoint • The corresponding S-ramp time must be set in C00182. • Lenze setting: 0 (deactivated)
	0	Off		
	1	PT1 behaviour		
C00182	0.01	s	50.00	S-ramp time PT1 • Lenze setting: 20.00 s
C00190				Selection of the arithmetic function for combining main and additional setpoint
	0	NOut = NSet		Lenze setting • 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 • Lenze setting: 0.000 s

Parameter	Possible settings			Info
C00221	0.000	s	999.900	Deceleration time T_{if} for the additional setpoint • Lenze setting: 0.000 s
C00241	0.00	%	100.00	Hysteresis window for zero detection of speed output setpoint (output *bRfgIEqO*) • Lenze setting: 0.50 %
C00632/1...3	0.00	%	199.99	Maximum limit values for the speed blocking zones • Selection of the maximum limit values for the blocking zones in which the speed must not be constant. • Lenze setting: 0.00 %
C00633/1...3	0.00	%	199.99	Minimum limit values for the speed blocking zones • Selection of the minimum limit values for the blocking zones in which the speed must not be constant. • Lenze setting: 0.00 %
C00634				Status (bit-coded) • 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 • Maximum speed setpoint for speed setpoint limitation • Lenze setting: 199.99 %
C00636	-199.99	%	199.99	nMinLimit • Minimum speed setpoint for speed setpoint limitation • Lenze setting: -199.99 %

17.1.26.1 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.

17.1.26.2 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
bJog8	bJog4	bJog2	bJog1	Main setpoint
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

17.1.26.3 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.

17.1.26.4 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 %)

17.1.26.5 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!

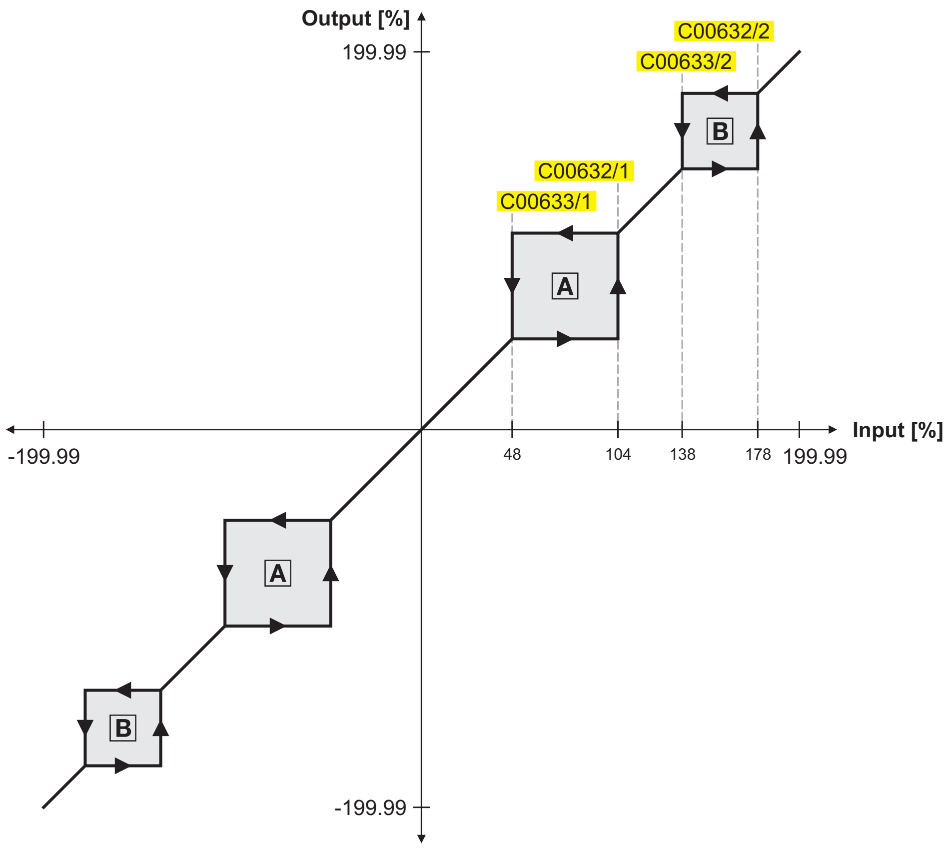
- Skip frequencies only affect main setpoints.
- It is not possible to exclude "0" speed if there is a sign reversal of the speed setpoint.

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

[17-13] 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.

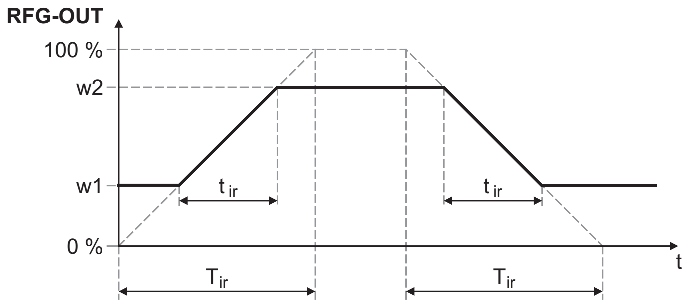
The same applies to a limit range of 0 ... xx %. During zero crossing of the speed setpoint, "0" speed is output as setpoint. It is possible to exclude "0" speed. However, in this case, the output speed will remain on the upper limit value when the input setpoint becomes "0".

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).

17.1.26.6 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

[17-14] 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:

$T_{ir} = t_{ir} \cdot \frac{100\ \%}{w2 - w1}$	$T_{if} = t_{if} \cdot \frac{100\ \%}{w2 - w1}$

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
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	

17.1.26.7 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.

17.1.26.8 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: • 100 % ≡ 16384 • 1 % ≡ 164
4	*nNOut_a* = (*nNSet_a* * 1 %) / |*nNAdd_a*|	
5	*nNOut_a* = (*nNSet_a* * 100 %) / (100 % - *nNAdd_a*)	

17.1.26.9 Application example for the additional load function

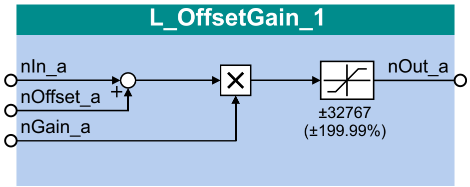
The motor control of the drive is provided with a function for automatically carrying along ramp function generators for "jerk-free" setpoint connection. For speed-controlled drive tasks, the LS_MotorInterface SB outputs the current actual speed value via the *nHlgSetValue_a* output (e.g. in case of 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*

17.1.27 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 Data type	Information/possible settings
nIn_a INT	Input signal • Scaling: 16384 ≡ 100 %
nOffset_a INT	Offset • Scaling: 16384 ≡ 100 %
nGain_a INT	Gain factor • Scaling: 16384 ≡ 100 % • 199.99 % ≈2

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal • Internal limitation to ±199.99 %

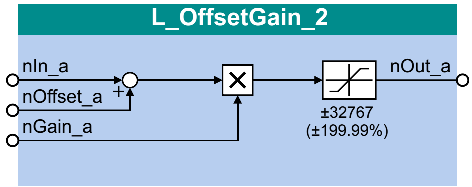
Function

$$\text{nOut\_a} = (\text{nIn\_a} + \text{Offset}) \cdot \text{Gain factor}$$

17.1.28 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 Data type	Information/possible settings
nIn_a INT	Input signal • Scaling: 16384 ≡ 100 %
nOffset_a INT	Offset • Scaling: 16384 ≡ 100 %
nGain_a INT	Gain factor • Scaling: 16384 ≡ 100 % • 199.99 % ≈2

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal • Internal limitation to ±199.99 %

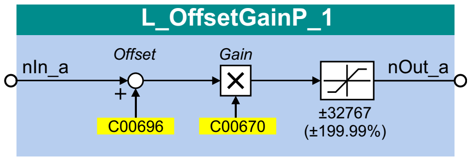
Function

$$nOut\_a = (nIn\_a + Offset) \cdot Gain\ factor$$

17.1.29 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 Data type	Information/possible settings
nIn_a INT	Input signal

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal • Internal limitation to ±199.99 %

Parameter

Parameter	Possible settings			Info
C00670	-100.0000		100.0000	Gain factor • High gain factor for further processing of smallest input signals. • Please observe the difference with regard to the gain factors of other blocks in percent (±199.99 % ≈ 2). • Lenze setting: 1.0000
C00696	-199.99	%	199.99	Offset • Lenze setting: 0.00 %

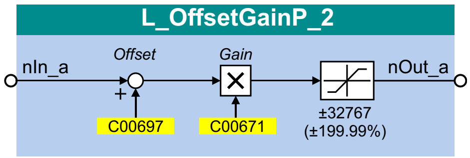
Function

$$nOut\_a = (nIn\_a + Offset) \cdot Gain\ factor$$

17.1.30 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 INT	Input signal

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal • Internal limitation to ±199.99 %

Parameter

Parameter	Possible settings			Info
C00671	-100.0000		100.0000	Gain factor • High gain factor for further processing of smallest input signals. • Please observe the difference with regard to the gain factors of other blocks in percent (±199.99 % ≈ 2). • Lenze setting: 1.0000
C00697	-199.99	%	199.99	Offset • Lenze setting: 0.00 %

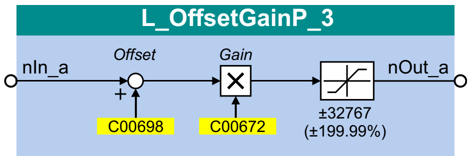
Function

$$nOut\_a = (nIn\_a + Offset) \cdot Gain\ factor$$

17.1.31 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 Data type	Information/possible settings
nIn_a INT	Input signal

Outputs

Identifier Data type	Value/meaning
nOut_a INT	Output signal • Internal limitation to ±199.99 %

Parameter

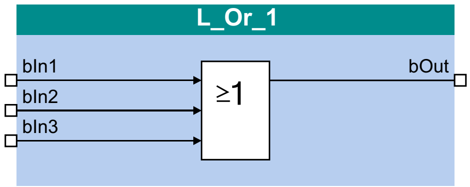
Parameter	Possible settings			Info
C00672	-100.0000		100.0000	Gain factor • High gain factor for further processing of smallest input signals. • Please observe the difference with regard to the gain factors of other blocks in percent (±199.99 % ≈ 2). • Lenze setting: 1.0000
C00698	-199.99	%	199.99	Offset • Lenze setting: 0.00 %

Function

$$nOut\_a = (nIn\_a + Offset) \cdot Gain\ factor$$

17.1.32 L_Or_1

This FB implements the ORing of the inputs signals.



Inputs

Identifier Data type	Information/possible settings
bIn1 bIn2 bIn3 BOOL	Input signal

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output signal

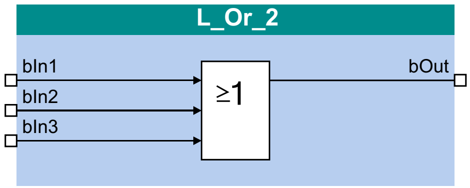
Function

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	

[17-15] Truth table of the FB L_Or_1

17.1.33 L_Or_2

This FB implements the ORing of the inputs signals.



Inputs

Identifier Data type	Information/possible settings
bIn1 bIn2 bIn3 BOOL	Input signal

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output signal

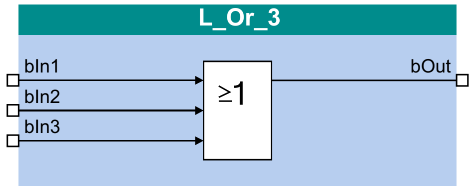
Function

	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	

[17-16] Truth table of the FB L_Or_2

17.1.34 L_Or_3

This FB implements the ORing of the inputs signals.



Inputs

Identifier Data type	Information/possible settings
bIn1 bIn2 bIn3 BOOL	Input signal

Outputs

Identifier Data type	Value/meaning
bOut 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	

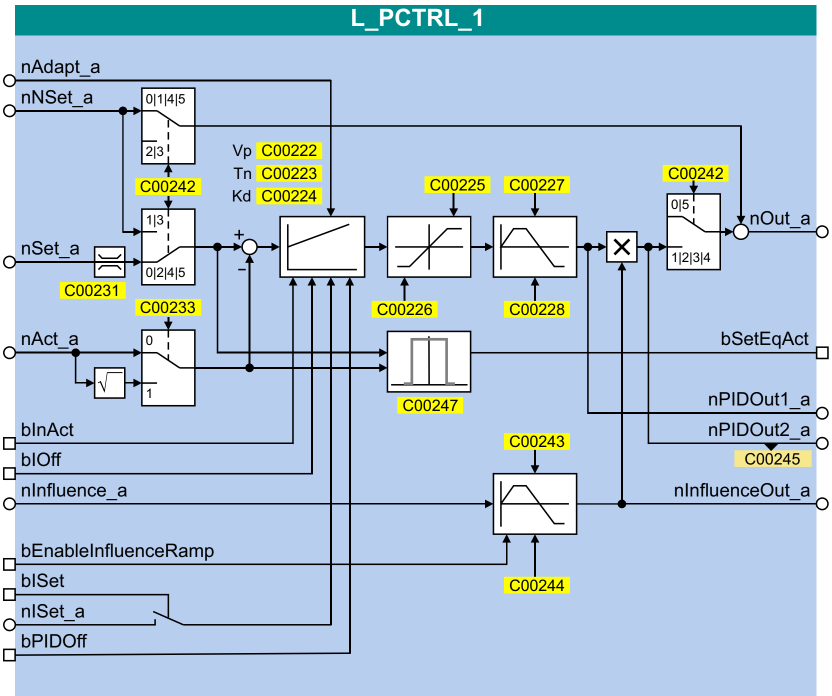
[17-17] Truth table of the L_Or_3 FB

17.1.35 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 provided 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 Data type	Information/possible settings
nAdapt_a INT	Adaptation of gain Vp set in C00222 in percent • Internal limitation to ± 199.99 % • Changes can be done online. • Display parameter: C00830/62
nNset_a INT	Speed setpoint • Scaling: 16384 ≡ 100 % • Internal limitation to ± 199.99 % • Display parameter: C00830/89
nSet_a INT	Sensor and process setpoint for operating modes 2, 4 and 5 • Scaling: 16384 ≡ 100 % • Internal limitation to ± 199.99 % • Display parameter: C00830/63

Identifier Data type		Information/possible settings
nAct_a INT		Speed or actual sensor value (actual process value) • Scaling: 16384 ≡ 100 % • Internal limitation to ± 199.99 % • Display parameter: C00830/61
bInAct BOOL		Deactivate process controller temporarily (stop) • Changes can be done online. • Display parameter: C00833/76 Note: This input is not interconnected in the LA_NCtrl application block.
	TRUE	• The current output value is frozen. • The internal control algorithm is stopped. • However, a setpoint selected via input *nNSet_a* is still provided in operating modes 0/1/4/5.
bIOff BOOL		Switch off the I-component of the process controller • Changes can be done online. • Display parameter: C00833/77
	TRUE	The I component of the process controller is set to zero.
nInfluence_a INT		Limitation of the influencing factor in percent • *nInfluence_a* serves to limit the influencing factor of the PID controller contained in the FB to a required value (- 199.99 % ... + 199.99 %). • Scaling: 16384 ≡ 100 % • Internal limitation to ± 199.99 % • Display parameter: C00830/64
bEnableInfluenceRamp BOOL		Activate ramp for influencing factor • 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 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 INT		Selection of I component of PID controller • With a TRUE signal at *bISet*, the assigned value is accepted in the PID controller. • Scaling: 16384 ≡ 100 % • Internal limitation to ± 199.99 %
bPIDOff BOOL (from version 06.00.00)		Reset the entire PID controller
	TRUE	• The I component of the controller is set to zero. • The controller output is set to zero. • The internal control algorithm is stopped.

Outputs

Identifier Data type		Value/meaning
nOut_a INT		Output signal • Internal limitation to ±32767 (±199.99 %) • Scaling: 16384 ≡ 100 %
bSetEqAct INT		Status output "Setpoint and actual value are identical"
	TRUE	Setpoint and actual value are identical, i.e. no system deviation available.
nPIDOut1_a INT		PID controller output without influencing factor *nInfluence_a* • Inputs *bEnableInfluenceRamp* and *nInfluence_a* do not have any effect here, the limited PID output value influenced by the internal ramp times is output. • There is no connection with the additive input *nNSet_a*. • Scaling: 16384 ≡ 100 %

Identifier Data type	Value/meaning
nPIDOut2_a INT	PID controller output with influencing factor *nInfluence_a*. • There is no connection with the additive input *nNSet_a*. • Scaling: 16384 ≡ 100 % • Display parameter: C00245
nInfluenceOut_a INT	Current influencing factor ("ramp status") on the PID output value • Scaling: 16384 ≡ 100 %

Parameter

Parameter	Possible settings			Info
C00222	0.1	0.1	500.0	Gain Vp • Lenze setting: 1.0
C00223	20	ms	6000	Reset time Tn • Lenze setting: 400 ms
C00224	0.0	0.1	5.0	Differential component Kd • Lenze setting: 0.0
C00225	-199.99	%	+199.99	MaxLimit • Maximum value of the PID operating range • Lenze setting: 199.99 %
C00226	-199.99	%	+199.99	MinLimit • Minimum value of the PID operating range • 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) • Lenze setting: 0.010 s
C00228	0.000	s	999.999	Deceleration time for the ramp at the PID output • Lenze setting: 0.010 s
C00231/1 (Pos. Maximum) C00231/2 (Pos. Minimum) C00231/3 (Neg. Minimum) C00231/4 (Neg. Maximum)	0.00	%	199.99	Operating range • Determination of the operating range for the PID process controller by limiting the input signal *nSet_a*. • Lenze setting: No limitation (-199.99 % ... +199.99 %)
C00233				Root function • 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 • 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 • Acceleration time T_{ir} for the influencing factor. • Lenze setting: 5.000 s
C00244	0.000	s	999.999	Influence deceleration time • Deceleration time T_{if} for the influencing factor. • Lenze setting: 5.000 s
C00245	-199.99	%	+199.99	Display of PID output value *nPIDOut_a*
C00247 (from version 06.00.00)	0	%	100	Window for comparison function "Actual value = setpoint" • Lenze setting: 2 % • Hysteresis: 1 % (fixed)

17.1.35.1 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.

Reset time

The adjustment time Tn is set under C00223.

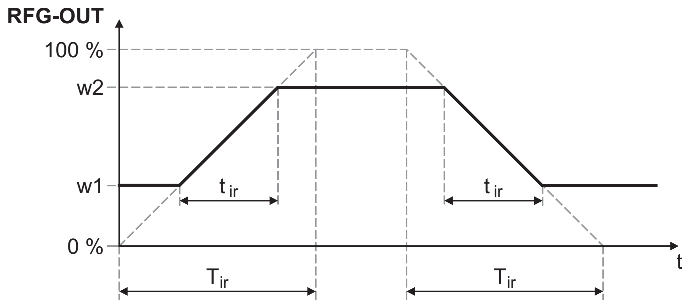
Differential component Kd (D component)

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.

17.1.35.2 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

[17-18] 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:

$T_{ir} = t_{ir} \cdot \frac{100\ \%}{w2 - w1}$	$T_{if} = t_{if} \cdot \frac{100\ \%}{w2 - w1}$

- The ramp function generator is immediately set to "0" by setting *bInAct* to TRUE.

17.1.35.3 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 %)

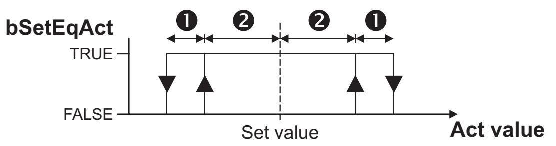
17.1.35.4 Evaluation of the output signal

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).

17.1.35.5 Comparison function "Actual value = setpoint"

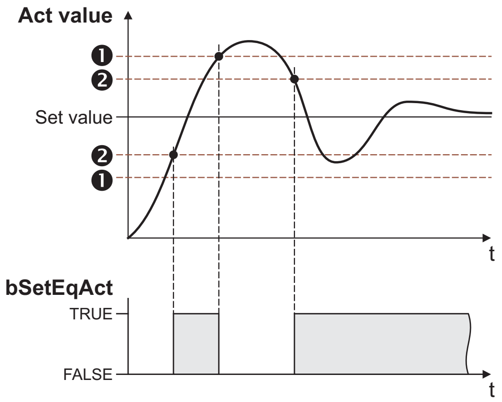
If setpoint and actual value are identical and there is no system deviation, the *bSetEqAct* status output is set to TRUE.

- The hysteresis of the comparison function has a fixed value of 1 %.
- From version 06.00.00 onwards, 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

[17-19] Comparison function: Switching performance



❶ Hysteresis (1 %, fixed value)
❷ C00247: Window

[17-20] Comparison function: Example

17.1.35.6 Control functions

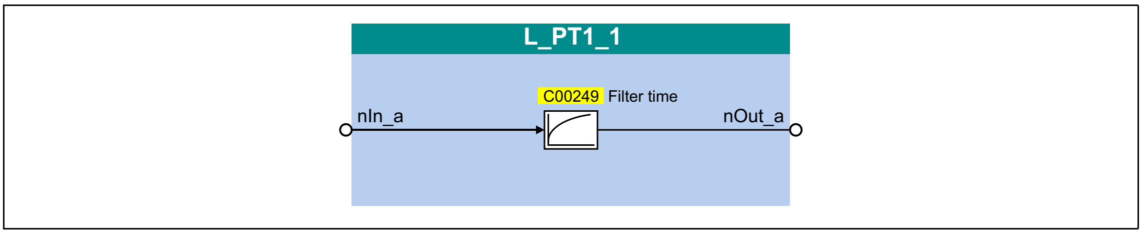
The process controller has various digital inputs for controlling the FB:

Identifier Data type	Information/possible settings	
bInAct BOOL	Deactivate process controller temporarily (stop) • Changes can be done online. • Display parameter: C00833/76 Note: This input is not interconnected in the LA_NCtrl application block.	
	TRUE	• The current output value is frozen. • The internal control algorithm is stopped. • However, a setpoint selected via input *nNSet_a* is still provided in operating modes 0/1/4/5.
bIOff BOOL	Switch off the I-component of the process controller • Changes can be done online. • Display parameter: C00833/77	
	TRUE	The I component of the process controller is set to zero.
bPIDOff BOOL (from version 06.00.00)	Reset the entire PID controller	
	TRUE	• The I component of the controller is set to zero. • The controller output is set to zero. • The internal control algorithm is stopped.

17.1.36 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 Data type	Information/possible settings
nIn_a INT	Input signal

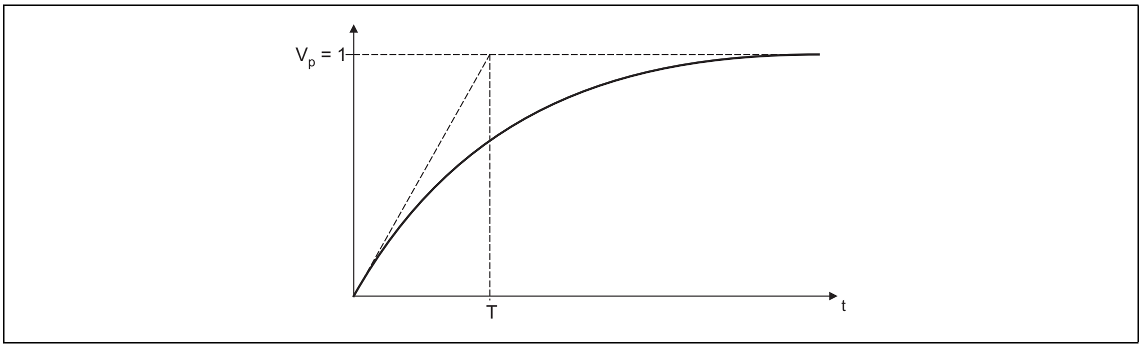
Outputs

Identifier Data type	Value/meaning
nOut INT	Output signal

Parameter

Parameter	Possible settings			Info
C00249	0	ms	5000	Filter time constant • The filter is not active with a setting of "0 ms". The input signal is passed through one-to-one to the output. • Lenze setting: 2000 ms

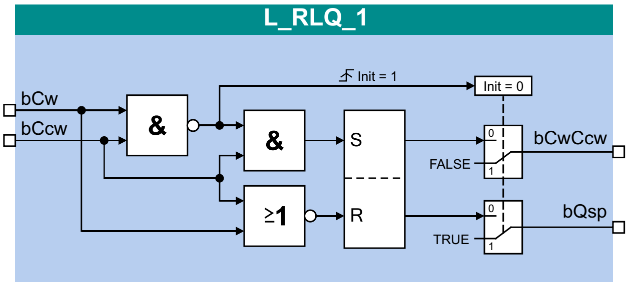
Function



[17-21] Filter time constant T of the first-order delay element

17.1.37 L_RLQ_1

This FB links a selected direction of rotation to the quick stop function with wire-break protection.



Inputs

Identifier Data type	Information/possible settings
bCw BOOL	Input • TRUE = CW rotation
bCCw BOOL	Input • TRUE = CCW rotation

Outputs

Identifier Data type	Value/meaning
bQSP BOOL	Output signal for quick stop (QSP)
bCwCcw BOOL	Output signal for CW/CCW rotation • TRUE = CCW rotation

Function

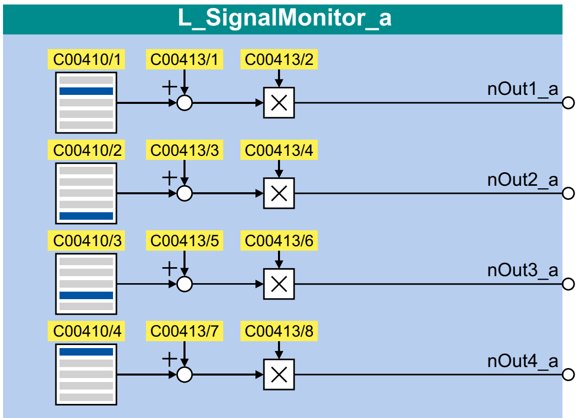
Inputs		Outputs		Notes
bCw	bCCw	bCwCcw	bQSP	
TRUE	TRUE	FALSE	TRUE	The inputs have this status only if a TRUE signal is being applied to both inputs at the moment of switch-on! See also FB illustration above, "Init" = 1.
If *one* of the inputs has the TRUE status, the following truth table applies:				
FALSE	FALSE	FALSE	TRUE	See also FB illustration above, "Init" = 0.
TRUE	FALSE	FALSE	FALSE	
FALSE	TRUE	TRUE	FALSE	
TRUE	TRUE	X (save)		

[17-22] Truth table of the FB L_RLQ, 0 = FALSE, 1 = TRUE

17.1.38 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 Data type	Value/meaning
nOut1_a INT	Output signal • Internal limitation to ±32767
nOut2_a INT	Output signal • Internal limitation to ±32767
nOut3_a INT	Output signal • Internal limitation to ±32767
nOut4_a INT	Output signal • Internal limitation to ±32767

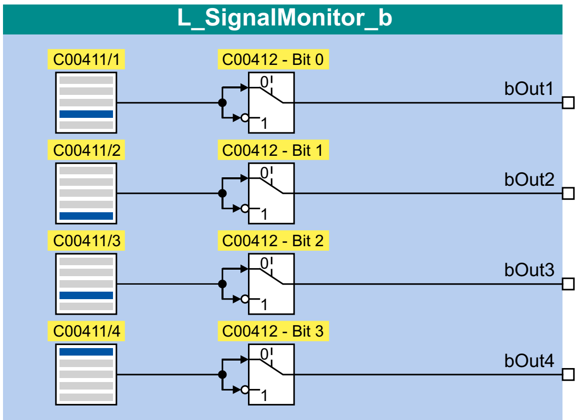
Parameter

Parameter	Possible settings			Info
C00410/1 ... 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 C00413/3 C00413/5 C00413/7	-199.99	%	+199.99	Offset
C00413/2 C00413/4 C00413/6 C00413/8	-199.99	%	+199.99	Gain

17.1.39 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 Data type	Value/meaning
bOut1 ... bOut4 BOOL	Output signal FALSE / TRUE

Parameter

Parameter	Possible settings		Info
C00411/1 ... 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 • 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	

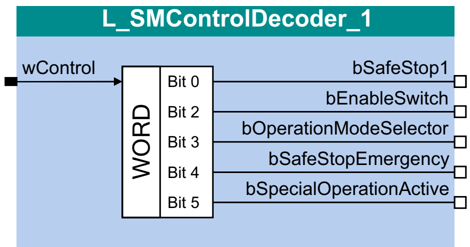
17.1.40 L_SMControlDecoder_1

The integrated safety system transmits information on requested or active safety functions via the bit-coded *wControl* control signal of the SB LS_SMInterface. This FB decodes the control signal into individual boolean control signals for further use in the function block interconnection.

Note!

The application in the controller has to evaluate the *wControl* control signal and execute the corresponding action. This action (e.g.braking to a standstill) has to be implemented by a corresponding application interconnection, which has to be carried out by an application engineer!

- Several safety functions can be requested/active at the same time.
- Which safety functions are supported depends on the safety module used or the available safety option.
- The functionality of this FB corresponds to a converter "WORD➞BOOL", but the outputs of this FB are named according to their meaning.



Inputs

Identifier Data type	Information/possible settings
wControl WORD	Input for the bit-coded control signal of the integrated safety system • The control signal of the integrated safety system is provided at the *wControl* output of the LS_SMInterface system block.

Outputs

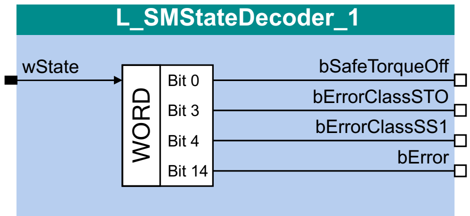
Bit*	Identifier	Requested/active safety function in case of TRUE
0	bSafeStop1	Safe stop 1 (SS1)
2	bEnableSwitch	Safe enable switch (ES)
3	bOperationModeSelector	Safe operation mode selector (OMS)
4	bSafeStopEmergency	Emergency stop (SSE)
5	bSpecialOperationActive	Special operation is active.

* with regard to the *wControl* input signal.

17.1.41 L_SMStateDecoder_1

The integrated safety system transmits the status of safety functions to the application via the bit coded *wState* status signal of the SB LS_SMInterface. This FB decodes the status signal into individual boolean status signals for further use in the function block interconnection.

- The functionality of this FB corresponds to a converter "WORD➞BOOL", but the outputs of this FB are named according to their meaning.



Inputs

Identifier Data type	Information/possible settings
wState WORD	Input for the bit-coded status signal of the integrated safety system • The status signal of the integrated safety system is provided at the *wState* output of the LS_SMInterface system block.

Outputs

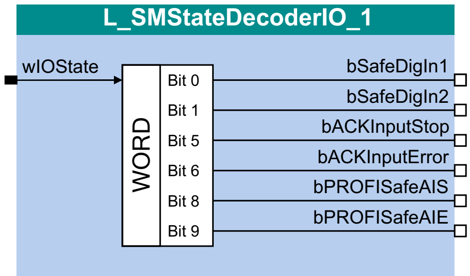
Bit*	Identifier	Meaning in case of TRUE
0	bSafeTorqueOff	"Safe torque off (STO)" function is active. • The drive is safely switched to torqueless operation.
3	bErrorClassSTO	Error stop category 0: "Safe torque off (STO)" function is active.
4	bErrorClassSS1	Error stop category 1: "Safe stop 1 (SS1)" function is active.
14	bError	Integrated safety system in error status (trouble or warning).

* with regard to the *wState* input signal.

17.1.42 L_SMStateDecoderIO_1

The bit-coded status signal *wIOState* of the SB LS_SMInterface enables the integrated safety system to transfer the status of the safe inputs to the application. This FB decodes the status signal in single boolean status signals for further processing in the FB interconnection.

- The functionality of this FB corresponds to a converter "WORD➞BOOL", but the outputs of this FB are named according to their meaning.



Inputs

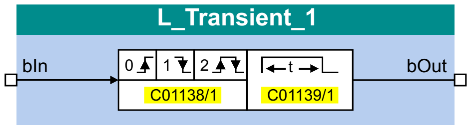
Identifier Data type	Information/possible settings
wIOState WORD	Input for the bit coded I/O status signal of the integrated safety system • The I/O status signal of the integrated safety system is provided at the *wIOState* output of the LS_SMInterface system block.

Outputs

Bit*	Identifier	Meaning in case of TRUE
0	bSafeDigIn1	Sensor input 1 in ON state.
1	bSafeDigIn2	Sensor input 2 in ON state.
5	bACKInputStop	Restart acknowledgement via terminal is done (if TRUE↘FALSE).
6	bACKInputError	Error acknowledgement via terminal is done (if TRUE↘FALSE).
8	bPROFISafeAIS	Restart acknowledgement via safety bus is done (if FALSE↗TRUE).
9	bPROFISafeAIE	Error acknowledgement via safety bus is done (if FALSE↗TRUE).
* With regard to the *wIOState* input signal.		

17.1.43 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 Data type	Information/possible settings
bIn BOOL	Input for edge evaluation • The function depends on the selection of edge evaluation in C01138/1.

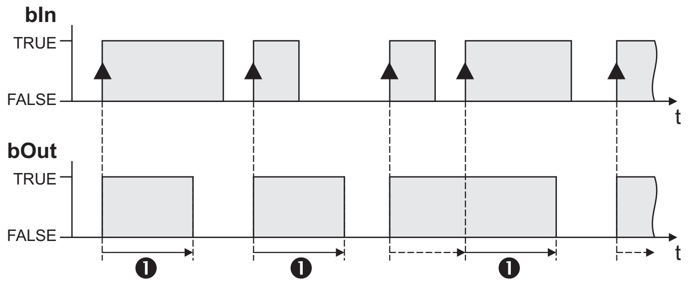
Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output (retriggerable)

Parameter

Parameter	Possible settings			Info
C01138/1				Function • 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 • Lenze setting: 0.001 s

17.1.43.1 Function 0: Evaluate rising signal edges



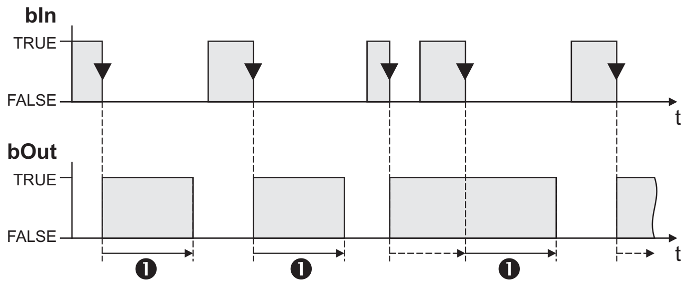
❶ Pulse duration

[17-23] 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.

17.1.43.2 Function 1: Evaluate falling signal edges



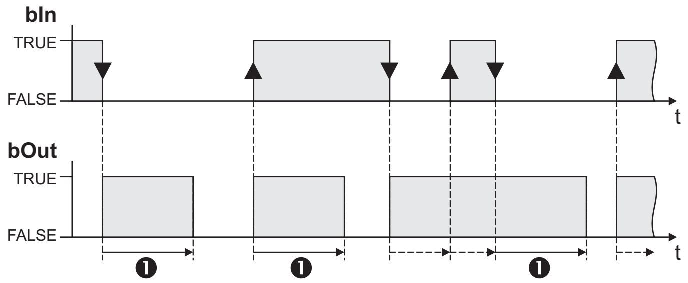
❶ Pulse duration

[17-24] 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.

17.1.43.3 Function 2: Evaluate rising and falling signal edges



❶ Pulse duration

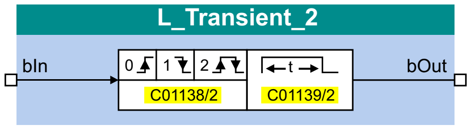
[17-25] Switching performance for function selection "2: High and low edge"

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.

17.1.44 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 Data type	Information/possible settings
bIn BOOL	Input for edge evaluation • The function depends on the selection of edge evaluation in C01138/2.

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output (retriggerable)

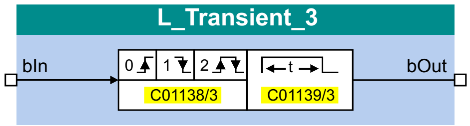
Parameter

Parameter	Possible settings			Info
C01138/2				Function • 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 • Lenze setting: 0.001 s

For a detailed functional description see L_Transient_1.

17.1.45 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 Data type	Information/possible settings
bIn BOOL	Input for edge evaluation • The function depends on the selection of edge evaluation in C01138/3.

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output (retriggerable)

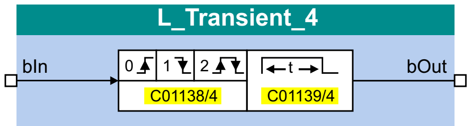
Parameter

Parameter	Possible settings			Info
C01138/3				Function • 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 • Lenze setting: 0.001 s

For a detailed functional description see L_Transient_1.

17.1.46 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 Data type	Information/possible settings
bIn BOOL	Input for edge evaluation • The function depends on the selection of edge evaluation in C01138/4.

Outputs

Identifier Data type	Value/meaning
bOut BOOL	Output (retriggerable)

Parameter

Parameter	Possible settings			Info
C01138/4				Function • 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 • Lenze setting: 0.001 s

For a detailed functional description see L_Transient_1.

17.2 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 previous chapter "Function blocks". (📖 710)

Overview of system blocks available

System block	Function	can be inserted into level: I/O	Appl.
LS_AnalogInput	Interface to the analog input terminals ▸ Analog terminals (📖 221)	●	
LS_CANManagement	Control of internal functions of the CAN driver and display of the "Operational" status as well as the node address ▸ CANopen option (📖 371)	●	
LS_DataAccess	*Lenze internal only*		●
LS_DeviceMonitor	Motor control status signals ▸ Motor control (MCTRL) (📖 98)		●
LS_DigitalInput	Interface to the digital input terminals ▸ Digital terminals (📖 198)	●	
LS_DigitalOutput	Interface to the digital output terminals ▸ Digital terminals (📖 198)	●	
LS_DisFree	Display of 4 arbitrary 16-bit signals of the application on display codes	●	●
LS_DisFree_a	Display of 4 arbitrary analog signals of the application on display codes	●	●
LS_DisFree_b	Display of 8 arbitrary digital signals of the application on a bit coded display code	●	●
LS_DriveInterface	Interface to drive control (DCTRL) ▸ Device control (DCTRL) (📖 67)		●
LS_IRInterface	SB in preparation!	●	
LS_Keypad	Control via keypad	●	
LS_MotionControlKernel	Interface to the basic drive function implemented in the Motion Control Kernel (MCK) ▸ Basic drive functions (MCK) (📖 295)		●
LS_MotorInterface	Interface to motor control (MCTRL) ▸ Motor control (MCTRL) (📖 98)		●
LS_ParFix	Output of frequently used constants (TRUE, FALSE, 100 %, etc.) to be used in the interconnection	●	
LS_ParFree	Output of 4 parameterisable 16-bit signals	●	
LS_ParFree_a	Output of 4 parameterisable analog signals	●	
LS_ParFree_b	Output of 16 parameterisable digital signals	●	
LS_ParFree_v	Output of 4 parameterisable speed signals	●	
LS_ParReadWrite_1 ... LS_ParReadWrite_3	Reading/Writing of local parameters	●	
LS_PulseGenerator	Output of 9 fixed frequencies and 1 parameterisable frequency	●	
LS_ServiceSwitch	Output of the switching status of the optional service switch	●	
LS_SetError_1	Parameterisable responses to user-defined events are tripped ▸ Diagnostics & error management (📖 322)	●	

System block	Function	can be inserted into level:	
		I/O	Appl.
LS_SMInterface	Interface to the integrated safety system	●	
LS_WriteParamList	Interface to the basic "Parameter change-over" function ▶ Parameter change-over (📖 459)	●	

Related topics:

▶ Overview of function blocks available (📖 710)
▶ Working with the FB Editor (📖 658)

17.2.1 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" (225)

17.2.2 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" (441)

17.2.3 LS_DataAccess

Only for Lenze-internal use.

17.2.4 LS_DeviceMonitor

Motor control status signals.

For a detailed description see the main chapter "Motor control (MCTRL)":
- Internal status signals | System block "LS_DeviceMonitor" (195)

17.2.5 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" (213)

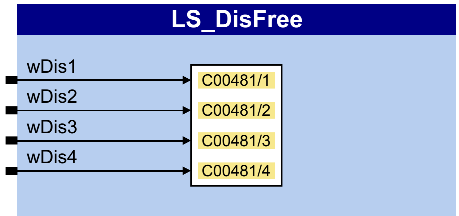
17.2.6 LS_DigitalOutput

Interface to the digital output terminals.

For a detailed description see the main chapter "I/O terminals":
- Internal interfaces | System block "LS_DigitalOutput" (220)

17.2.7 LS_DisFree

This system block displays 4 arbitrary 16-bit signals of the application on display codes.



Inputs

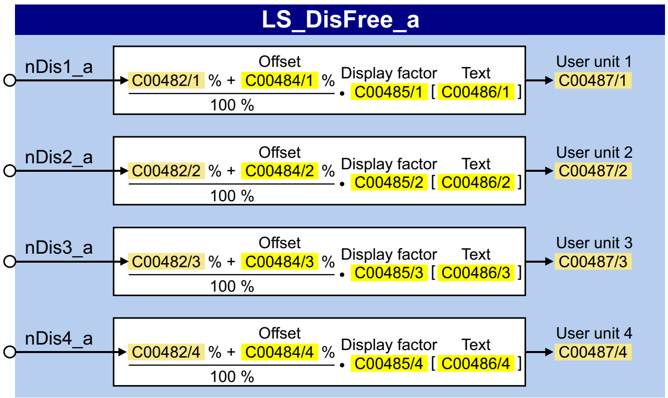
Identifier Data type	Information/possible settings
wDis1 ... wDis4 WORD	Inputs for any 16-bit signals of the application

Parameter

Parameter	Possible settings			Info
C00481/1...4	0x0000		0xFFFF	Display of the 16-bit signals which are applied at the *wDis1 ... wDis4* inputs

17.2.8 LS_DisFree_a

This system block displays 4 arbitrary analog signals of the application on display codes.



Inputs

Identifier Data type	**Information/possible settings**
nDis1_a ... nDis4_a INT	Inputs for arbitrary analog signals of the application

Parameter

Parameter	**Possible settings**			**Info**
C00482/1...4	-199.99 %		199.99 %	Display of the analog signals which are applied at the *nDis1_a ... nDis4_a* inputs
C00484/1...4 ... C00487/1...4	From version 06.00.00 onwards: ▶ Display of internal process factors in application units			

17.2.8.1 Display of internal process factors in application units

This function extension is available from version 06.00.00!

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 • See formula [17-26]. • Lenze setting: 0.00 %
C00485/1...4	-65536.0000		65536.0000	Display factor 1 ... 4 • Scaling of the input variable for the display. • See formula [17-26]. • Lenze setting: 1.0000
C00486/1...4	String of digits			Text 1 ... 4 • For each display value, an individual unit (e.g. "bottles") can be set.

$$\text{User unit 1} = \frac{\text{nDis1\_a [\%] + Offset 1 [\%]}}{\text{100 [\%]}} \cdot \text{Display factor 1 [text 1]}$$

[17-26] 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"

$$\text{User unit 1} = \frac{100\ [\%] + 0\ [\%]}{100\ [\%]} \cdot 123.45\ [\text{bottles}] = 123.45\ \text{bottles}$$

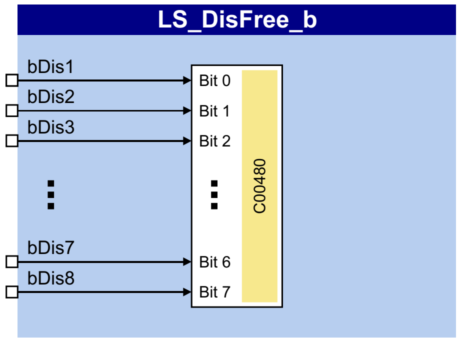
Example 2:

- Input variable *nDis2_a* = 40 %
- Offset 2 (C00484/2) = 35 %
- Display factor 2 (C00485/2) = 20
- Text 2 (C00486/2) = "kg"

$$\text{User unit 2} = \frac{40\ [\%] + 35\ [\%]}{100\ [\%]} \cdot 20\ [\text{kg}] = 15.00\ \text{kg}$$

17.2.9 LS_DisFree_b

This system block displays 8 arbitrary digital signals of the application on a bit coded display code.



Inputs

Identifier Data type	Information/possible settings
bDis1 ... bDis8 BOOL	Inputs for arbitrary digital signals of the application

Parameter

Parameter	Possible settings		Info
C00480	0x0000	0xFFFF	Display of the digital signals applied at the *bDis1* ... *bDis8* inputs in the form of hexadecimal values
	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 7	Signal level at the *bDis8* input	

17.2.10 LS_DriveInterface

Interface to internal device control.

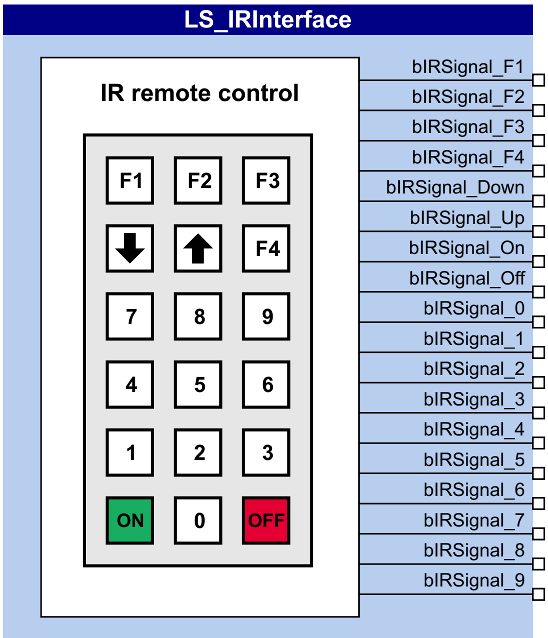
For a detailed description see main chapter "Device control (DCTRL)":

▸ Internal interfaces | "LS_DriveInterface" system block (📖 93)

17.2.11 LS_IRInterface

SB in preparation!

This system block makes the signals of the optional infrared remote control available in the FB Editor. There, the signals can be linked e.g. with control inputs of the technology application.



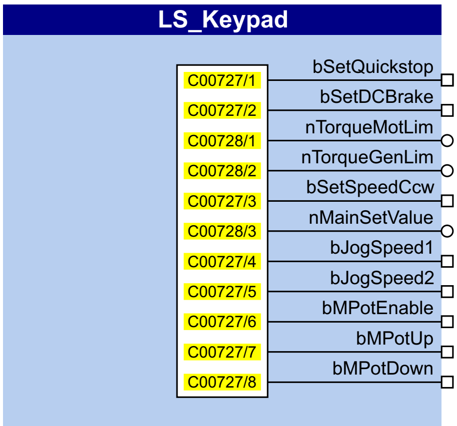
Outputs

Identifier Data type	Meaning in case of TRUE:
bIRSignal_F1 ... bIRSignal_F4 BOOL	Key F1 ... F4 pressed
bIRSignal_Down bIRSignal_Up BOOL	Key ▲ / ▼ pressed
bIRSignal_On bIRSignal_Off BOOL	Key ON / OFF pressed
bIRSignal_0 ... bIRSignal_9 BOOL	Key 0 ... 9 pressed

17.2.12 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 Data type	Value/meaning
bSetQuickstop BOOL	C00727/1 = "1" ≡ Request quick stop
bSetDCBrake BOOL	C00727/2 = "1" ≡ Request DC-injection braking
nTorqueMotLim INT	Torque limit in motor mode set in C00728/1 • Lenze setting: 100.00 %
nTorqueGenLim INT	Torque limit in generator mode set in C00728/2 • Lenze setting: 100.00 %
bSetSpeedCcw BOOL	C00727/3 = "1" ≡ Request reversal
nMainSetValue INT	Setpoint speed set in C00728/3 • Lenze setting: 0.00 %
bJogSpeed1 BOOL	C00727/4 = "1" ≡ Request fixed speed setpoint 1
bJogSpeed2 BOOL	C00727/5 = "1" ≡ Request fixed speed setpoint 2
bMPotEnable BOOL	C00727/6 = "1" ≡ Motor potentiometer: Request activation
bMPotUp BOOL	C00727/7 = "1" ≡ Motor potentiometer: Request positive acceleration
bMPotDown BOOL	C00727/8 = "1" ≡ Motor potentiometer: Request negative acceleration

Parameter

Parameter	Possible settings			Info
C00727/1...8	0		1	Keypad digital values • Execution of control commands for keypad operation • See the "Outputs" table for the meaning of the individual subcodes
C00728/1...3	-199.99	%	199.99	Analog values - keypad • Selection of various setpoints for operation via keypad • See the "Outputs" table for the meaning of the individual subcodes

17.2.13 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" (📖 297)

17.2.14 LS_MotorInterface

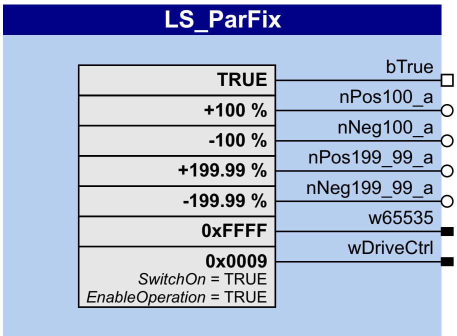
Interface to internal motor control.

For a detailed description see the main chapterl "Motor control (MCTRL)":

▸ Internal interfaces | System block "LS_MotorInterface" (📖 190)

17.2.15 LS_ParFix

This system block outputs various fixed values (constants) to be used in the interconnection.

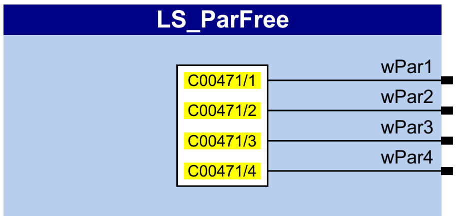


Outputs

Identifier Data type	Value/meaning
bTrue BOOL	1 ≡ TRUE
nPos100_a INT	16384 ≡ + 100 %
nNeg100_a INT	-16384 ≡ - 100 %
nPos199_99_a INT	32767 ≡ + 199.99 %
nNeg199_99_a INT	-32767 ≡ - 199.99 %
w65535 WORD	65535 ≡ 0xFFFF
wDriveCtrl WORD	9 ≡ 0x0009 • Bit 0, SwitchOn = TRUE • Bit 3, EnableOperation = TRUE • All others: FALSE See also: ▸ wCANControl/wMCIControl control words (📖 96)

17.2.16 LS_ParFree

This system block outputs 4 parameterisable 16-bit signals.



Outputs

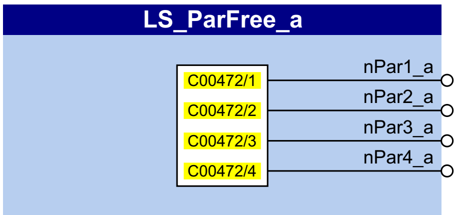
Identifier Data type	**Value/meaning**
wPar1 ... wPar4 WORD	Output of the 16-bit signals parameterised in C00471/1...4

Parameter

Parameter	**Possible settings**			**Info**
C00471/1...4	0x0000		0xFFFF	Setting of the 16-bit signals to be output

17.2.17 LS_ParFree_a

This system block outputs 4 parameterisable analog signals.



Outputs

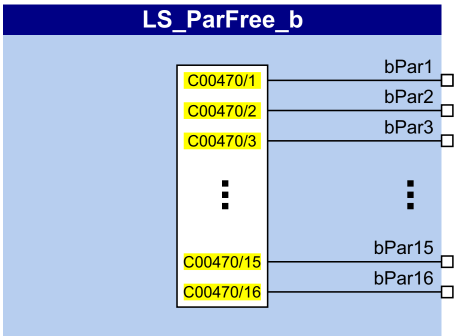
Identifier Data type	Value/meaning
nPar1_a ... nPar4_a INT	Output of the analog signals parameterised in C00472/1...4

Parameter

Parameter	Possible settings			Info
C00472/1...4	-199.99	%	+199.99	Selection of analog signals to be output

17.2.18 LS_ParFree_b

This system block outputs 16 parameterisable digital signals.



Outputs

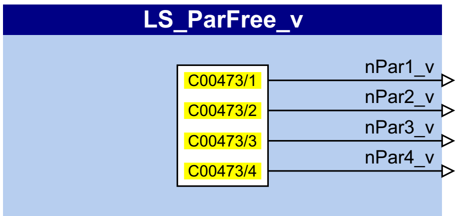
Identifier Data type	Value/meaning
bPar1 ... bPar16 BOOL	Output of the signal levels (FALSE/TRUE) parameterised in C00470/1...16

Parameter

Parameter	Possible settings		Info
C00470/1...16			Selection of signal levels to be output • Bit 0 ... 15 = *bPar1* ... *bPar16*
	0	"FALSE" signal is output	
	1	"TRUE" signal is output	

17.2.19 LS_ParFree_v

This system block outputs 4 parameterisable speed signals.



Outputs

Identifier Data type	Information/possible settings
nPar1_v ... nPar4_v INT	Output of the speed signals parameterised in C00473/1...4

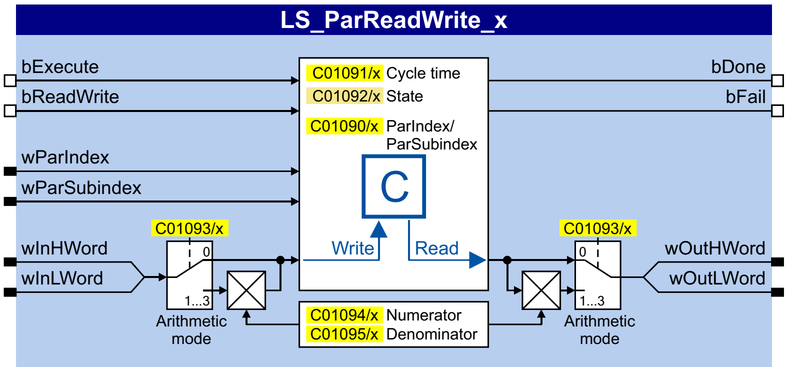
Parameter

Parameter	Possible settings			Info
C00473/1...4	-32767	Incr/ms	+32767	Selection of speed signals to be output

17.2.20 LS_ParReadWrite_1-3

The **LS_ParReadWrite_1** ... **LS_ParReadWrite_3** system blocks serve to read and write 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 Data type	Information/possible settings	
bExecute BOOL	Trip read/write request	
	FALSE↗TRUE	If cycle time (C01091) = "0 ms": One-time reading/writing of the parameter value which has been addressed via the*wParIndex* and *wParSubindex* inputs.
		If cycle time (C01091) > "0 ms": Cyclic reading/writing of the parameter value which has been addressed via the *wParIndex* and *wParSubindex* inputs.
	TRUE↘FALSE	Deactivate cyclic reading/writing again.
bReadWrite BOOL	Selection: Read or write request	
	FALSE	Read request
	TRUE	Write request
wParIndex WORD	Code to be read or written. • From version 06.00.00 onwards, this selection can be optionally carried out via C01090.	
wParSubindex WORD	Subcode to be read or written. • From version 06.00.00 onwards, this selection can be optionally carried out via C01090.	
wInHWord wInLWord WORD	Value to be written (DataHigh/DataLow portion)	

Outputs

Identifier Data type	Value/meaning	
bDone BOOL	"Read/Write request successfully completed" status signal • 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: 1. There is no active read/write request. 2. The read/write request has not been completed yet. 3. An error has occurred (if *bFail* = TRUE).
bFail BOOL	"Error" status	
	TRUE	An error has occurred (group signal). • See display parameter (C01092) for details.
wOutHWord wOutLWord WORD	Value which was read (DataHigh/DataLow portion) after read request	

Parameter

Parameter	Possible settings			Info
C01090/1...3 (from version 06.00.00)	0,000		16000,000	Parameter to be read or written. • For a setting of "0,000", inputs *wParIndex* and *wParSubindex* are effective for addressing purposes instead. • Lenze setting: 0.000
	Format: <Code number>,<subcode number>			
C01091/1...3				Cycle time • Subcode 1 = **LS_ParReadWrite_1** Subcode 2 = **LS_ParReadWrite_2** Subcode 3 = **LS_ParReadWrite_3** • 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...3				Error status • If *bFail* = TRUE: Error status is displayed. • Subcode 1 = **LS_ParReadWrite_1** Subcode 2 = **LS_ParReadWrite_2** Subcode 3 = **LS_ParReadWrite_3**
	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		
C01093/1...3 (from version 06.00.00)				Arithmetic mode • Lenze setting:"0: No arithmetic" ▸ Arithmetic function
	0	No arithmetic		
	1	In16Bit: LW=+/-32767		
	2	In16Bit: HW=+/-; LW=0..65535		
	3	In32Bit: HW_LW=+/-2147483647		
C01094/1...3 (from version 06.00.00)	-32767		32767	Counter • For internal conversion in arithmetic modes 1 ... 3. • Lenze setting: 1
C01095/1...3 (from version 06.00.00)	1		32767	Denominator • For internal conversion in arithmetic modes 1 ... 3. • Lenze setting: 1

17.2.20.1 Arithmetic function

This function extension is available from version 06.00.00!

The implemented arithmetic function enables easy arithmetic conversion of the process values to be written or read via parameterisable factors into the format of the target parameter without the need of an additional arithmetic function block.

- In C01093, the interpretation of the *wInHWord* and *wInLWord* inputs can be set to be able to write to parameters:

<table>
<tr><th colspan="2">Arithmetic mode</th><th>wInHWord</th><th>wInLWord</th><th>Internal conversion</th></tr>
<tr><td rowspan="2">0</td><td rowspan="2">No arithmetic
(Lenze setting)</td><td colspan="2">INTEGER_32
(4 bytes with sign)</td><td rowspan="2">No
(same behaviour as before)</td></tr>
<tr><td>DataHigh portion</td><td>DataLow portion</td></tr>
<tr><td>1</td><td>In16Bit:
LW=+/-32767</td><td>-</td><td>INTEGER_16
(2 bytes with sign)</td><td rowspan="4">Yes
(see the following section)</td></tr>
<tr><td>2</td><td>In16Bit: HW=+/-;
LW=0..65535</td><td>Sign
(0 ≡ positive value)</td><td>UNSIGNED_16
(2 bytes without sign)</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">In32Bit: HW_LW=
+/-2147483647</td><td colspan="2">INTEGER_32
(4 bytes with sign)</td></tr>
<tr><td>DataHigh portion</td><td>DataLow portion</td></tr>
</table>

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.

$$\text{parameter value to be written} = \text{Input value}_{[32]} \cdot \frac{\text{Counter}_{[16]}}{\text{Denominator}_{[16]}}$$

C01094: Numerator
C01095: Denominator

[17-27] Internal conversion with write access

$$\text{Output value}_{[32]} = \text{Read parameter value} \cdot \frac{\text{Counter}_{[16]}}{\text{Denominator}_{[16]}}$$

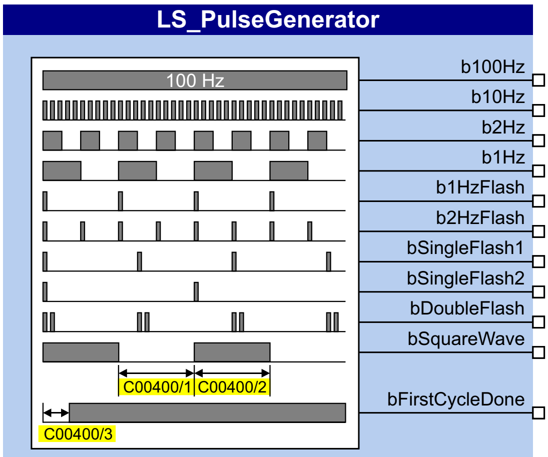
C01094: Numerator
C01095: Denominator

[17-28] Internal conversion with read access

17.2.21 LS_PulseGenerator

This system block outputs 9 different fixed frequencies and 1 frequency with parameterisable pulse/dead time.

From version 06.00.00 onwards, the SB provides a TRUE signal at the *bFirstCycleDone* output when the first 1-ms processing cycle is completed and the time set in C00400/3 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 Data type	Value/meaning
b100Hz BOOL	Rectangular signal 100 Hz
b10Hz BOOL	Rectangular signal 10 Hz
b2Hz BOOL	Rectangular signal 2 Hz
b1Hz BOOL	Rectangular signal 1 Hz
b1HzFlash BOOL	80 ms-pulse, repetition rate every second
b2HzFlash BOOL	80 ms-pulse, repetition rate every 0.5 seconds
bSingleFlash1 BOOL	80 ms pulse, repetition rate every 1.25 seconds
bSingleFlash2 BOOL	80 ms pulse, repetition rate every 2 seconds
bDoubleFlash BOOL	80 ms-double pulse, repetition rate every 1.25 seconds
bSquareWave BOOL	Output frequency with pulse/dead time set in C00400/1...2
bFirstCycleDone BOOL (from version 06.00.00)	Status signal "First processing cycle completed" 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).

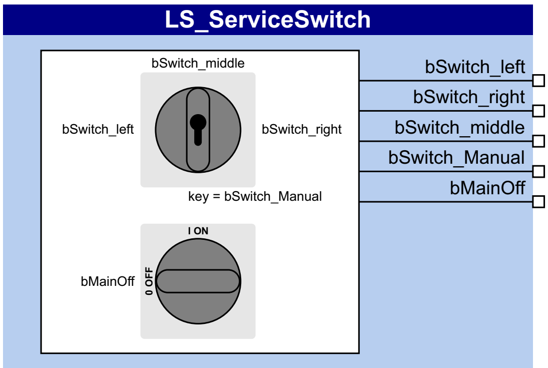
Parameter

Parameter	Possible settings			Info
C00400/1	0	ms	60000	Length of LOW level (break) • For output *bSquareWave* • Lenze setting: 1000 ms
C00400/2	0	ms	60000	Length of HIGH level • For output *bSquareWave* • Lenze setting: 1000 ms
C00400/3 (from version 06.00.00)	0	ms	60000	Delay of the *bFirstCycleDone* status • Lenze setting: 100 ms

17.2.22 LS_ServiceSwitch

This system block provides the switching states of the optional service switch in the FB Editor.

- Which outputs are effective depends on the service switch type available (service switch without/with protective function, service switch with operating unit)
- If no service switch is available, all outputs have to be permanently set to FALSE.



Outputs

Identifier Data type	Meaning in case of TRUE
bSwitch_left BOOL	Operating unit: Left switch position
bSwitch_right BOOL	Operating unit: Right switch position
bSwitch_middle BOOL	Operating unit: Central switch position
bSwitch_Manual BOOL	Operating unit: Key-operated switch in "manual" position
bMainOff BOOL	Service switch: "Off" switch position or overcurrent protective function has been triggered

Note!

For flexible customer solutions, the user has to configure the system block for the desired functionality correctly.

17.2.23 LS_SetError_1

Parameterisable responses to user-defined events are tripped.

For a detailed description see the main chapter "Diagnostics & error management":

▸ "LS_SetError_1" system block (📖 370)

17.2.24 LS_WriteParamList

Writing to a configurable list which contains up to 32 local parameters.

For a detailed description see the main chapterl "Parameter change-over". (📖 459)

Numbers

A

B

C

M

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

8400 protec StateLine · Reference manual · EDS84DSxxxx · 13350162 · DMS 4.2 EN · 03/2013 · TD05

Lenze Drives GmbH
Breslauer Straße 3
D-32699 Extertal
Germany
+49 (0)51 54 / 82-0
+49 (0)51 54 / 82-28 00
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