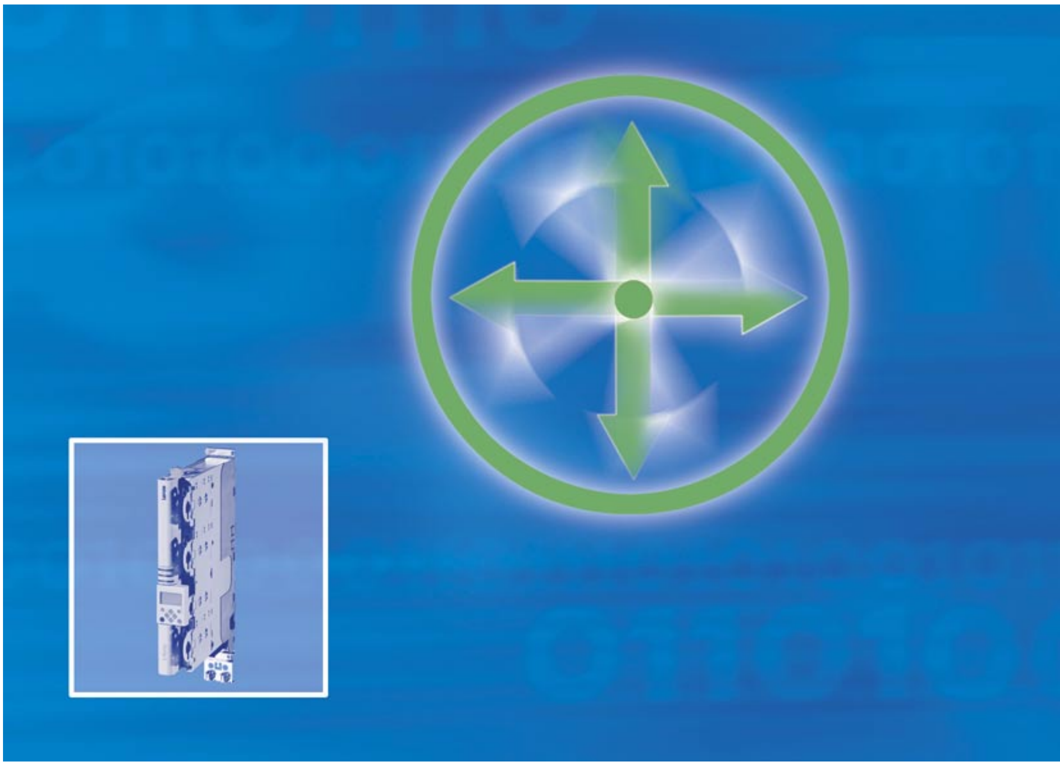
EDS94AxHExxxx
13331265

L-force Drives

Software Manual

9400



E94AxHExxxx

Servo Drives 9400 HighLine

Parameter setting & configuration

Lenze

Overview of technical documentation for Servo Drives 9400

Project planning, selection & order	
☐ Hardware manual 9400	
🖹 Catalogue / electronic catalogue (DSC - Drive Solution Catalogue)	

Mounting & wiring	
🖹 MA - 9400 StateLine/HighLine	
🖹 MA - communication module	
🖹 MA - extension module	
🖹 MA - safety module	
🖹 MA - accessories	
🖹 MA - remote maintenance components	

Parameter setting	
🖹 BA - keypad	
☐ SW - Lenze software »Engineer«	
☐ **SW - controller** (9400 StateLine/HighLine/PLC)	← **This documentation**
☐ SW - regenerative power supply module	
☐ KHB - communication module	
☐ SW - extension module	
☐ SW - safety module	
☐ SW - Lenze technology application	
☐ SW - function library 9400	

Configuring & programming	
☐ SW - Lenze software »Engineer«	
☐ SW - Lenze software »PLC Designer«	
☐ **SW - controller** (9400 HighLine/PLC)	← **This documentation**
☐ KHB - communication module	
☐ SW - extension module	
☐ SW - safety module	
☐ SW - Lenze technology application	
☐ SW - function library 9400	

Drive commissioning	
🖹 Commissioning guide	
☐ **SW - controller** (9400 StateLine/HighLine/PLC) → Chapter "Commissioning" (🕮 29) → Chapter "Oscilloscope" (🕮 590) → Chapter "Diagnostics & fault analysis" (🕮 611)	← **This documentation**
☐ Remote maintenance manual	

Networking structure	
☐ KHB - communication medium used	

Legend:	
🖹	Printed documentation
☐	Online documentation (PDF/Engineer online help)
Abbreviations used:	
BA	Operating instructions
KHB	Communication manual
MA	Mounting instructions
SW	Software Manual

Content

1 About this documentation

This documentation contains information on the 9400 HighLine controller.

Read the Mounting Instructions supplied with the controller before you start working!

The Mounting Instructions contain safety information which must be observed!

Target group

This documentation addresses to all persons who want to parameterise, configure, and diagnose the 9400 HighLine controller by means of the engineering software L-force »Engineer« and the keypad.

Validity information

The information in this documentation applies to the following standard devices:

Product series	Type designation	From software version
Servo Drives 9400	E94AxHExxxx	1.5

Document history

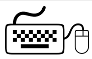
Version			Description
1.0	12/2006	TD05	First edition for 9400 HighLine V1.5
2.0	05/2007	TD05	Extended edition
3.0	11/2007	TD05	Supplemented with new functions for 9400 HighLine V3
4.0	06/2008	TD05	Supplemented with new functions for 9400 HighLine V4
4.1	07/2008	TD05	New main chapter: "CAN on board" system bus
5.0	11/2008	TD05	Extended by new functions for 9400 HighLine V5
5.1	12/2008	TD05	Error corrections
5.2	01/2009	TD05	Error corrections & supplements
6.0	08/2009	TD05	Extended by new functions for 9400 HighLine V7
6.1	08/2009	TD05	Error corrections & supplements

1.1 Conventions used

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

Type of information		Writing	Examples/notes
Numbers			
	Decimal separator	Point	The decimal point is always used. Example: 1234.56
Text			
	Version information	Text colour blue	The information valid for or from one specific software version of the controller is marked accordingly in this documentation. Example: This function extension is available from software version V3.0!
	Program name	» «	The Lenze PC software »Engineer«...
	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>**	Use **<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...

	Program code	`Courier`	`IF var1 < var2 THEN`
			`a = a + 1`
			`END IF`

	Keyword	**`Courier bold`**	
	Hyperlink	Underlined	Optically highlighted reference to another topic. It is activated with a mouse click in this online documentation.
Symbols			
	Page reference	(📖 19)	Optically highlighted reference to another page. It is activated with a mouse click in this online documentation.
	Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

The information valid for or from one specific software version of the controller is marked accordingly in this documentation.

1.2 Terminology used

Term	Meaning
»Engineer«	Lenze software which supports you throughout the whole machine life cycle - from planning to maintenance.
Code	"Container" for one or several parameters used for controller parameter setting or monitoring.
Subcode	If a code contains several parameters, the individual parameters are stored under "subcodes". This Manual uses a slash "/" as a separator between code and subcode (e.g. "C00118/3").
Function block editor	Graphical interconnection tool which is provided for controllers in the MotionControl HighLevel and TopLevel license level in the »Engineer« on the **FB editor** tab and by means of which the technology applications supplied can also be reconfigured and extended by individual functions.
Function block	A function block (FB) can be compared with an integrated circuit that contains a certain control logic and delivers one or several values when being executed. • An instance (reproduction, copy) of the function block is always inserted in the circuit. • It is also possible to insert several instances of a function block in a circuit. • Each instance has an unequivocal identifier (the instance name) and a processing number which defines the position at which the function block is calculated during the task cycle.
System block	System blocks provide interfaces to basic functions and hardware of the controller in the function block editor of the »Engineer« (e.g. to the digital inputs). • System blocks cannot be instanced in contrast to function blocks.
DIS code	Parameter that displays the current state or value of an input/output of a system block.
TA	Abbreviation for "Technology Application". Technology applications are applications prepared by Lenze which form the basis of solving typical applications. Each technology application is provided with an individual online documentation.

1.3 Definition of 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:

Danger!

(characterises 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	**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 simple handling
		Reference to another document

2 Introduction

The basis of every L-force application is an easy and quick parameter setting of prepared technology applications and solutions*.

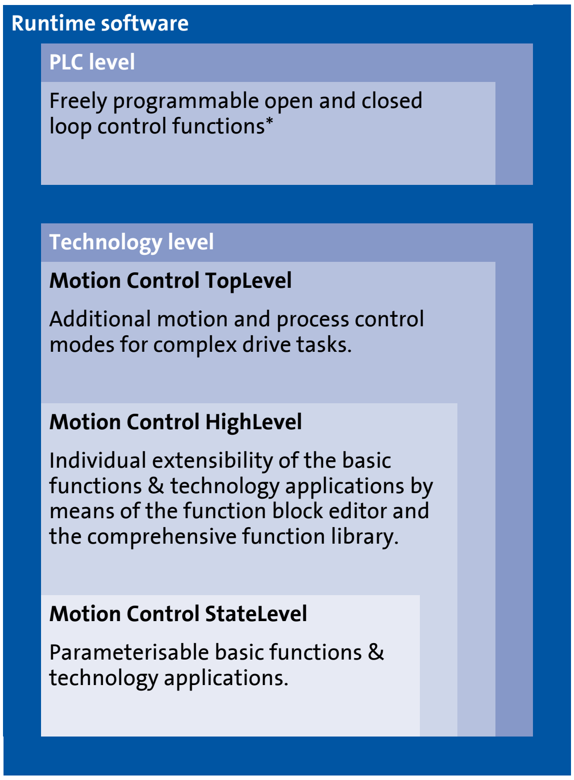
This chapter contains basic information on the runtime software model of L-force and on how you can establish an online connection between the PC and controller for parameter setting with the »Engineer« very easily.

At the end of this chapter you will find an overview of the different signal types & scaling which serve to process physical values (e.g. a speed or position) within the application.

* In preparation!

2.1 Parameter setting, configuring, or programming?

The graded runtime software model of L-force provides a simple and consistent solution for motion and process tasks as well as for complex machine functions:



Programming*

Configuring
The HighLevel and TopLevel licenses enable you to extend the provided technology applications by individual functions using the graphic function block editor of the »Engineer«. Here you can access the comprehensive function libraries of Lenze which among other things contain process controllers, arithmetic functions, logic blocks, and ramp generators and integrators.

Parameter setting
The StateLevel license includes a range of technology applications which can be put into operation easily with a keypad or via dialogs in the »Engineer«.

* In preparation!

2.1.1 Basic functionalities

Important basic drive functions and further basic functions are implemented in the firmware of the controller and thus are always provided, irrespective of the runtime software licence available.

Firmware

Motion Control basic drive functions

- Stop
- Quick stop
- Manual jog
- Homing
- Positioning
- Position follower
- Speed follower
- Torque follower
- Limiter
- Brake control

Further basic functionalities

- Drive interface
- Motor interface
- Encoder evaluation
- I/O terminals
- Safety engineering
- Logbook
- Oscilloscope

2.1.2 Technology applications

Technology applications (TAs) are applications prepared by Lenze which can serve as a basis for solving typical applications.

▶ The technology applications available for the Servo Drives 9400 can be selected in the »Engineer« from the application catalogue.

Runtime software

Technology level

Motion Control TopLevel

- TA "Positioning sequence control"
- TA "Electronic cam" *
- TA "Register control" *
- TA "Winding technology" *

Motion Control HighLevel

- TA "Electronic gearbox"
- TA "Synchronism with mark synchronisation"

Motion Control StateLevel

- TA "Actuator – speed"
- TA "Actuator – torque"
- TA "Table positioning"

Each higher license contains additional technology applications for further application fields.

* In preparation!

Tip!

Detailed information about the individual technology applications can be found in the corresponding software manuals.

2.2 Communicating with the controller

The following interfaces/communication modules can be used to establish communication between the PC and controller:

- Diagnostic interface X6/Going online via diagnostic adapter
- CAN on board interface/Going online via system bus (CAN on board) (📖 27)
- Optional interfaces which are provided by corresponding communication modules in the module slots MXI1/MXI2 of the controller.

Note!

For communication with the controller, at least the control electronics of the controller must be supplied with 24 V low voltage via plug X2. For detailed information, please see the Mounting Instructions for the controller.

Stop!

If you change parameters in the »Engineer« while the controller is connected online, the changes will be directly accepted by the controller!

Tip!

Detailed information about the individual interfaces can be found in the corresponding Communication Manuals (KHB).

2.2.1 Going online via diagnostic adapter

For initial commissioning of the controller you can for instance use the diagnostic adapter offered by Lenze:

Note!

Please observe the documentation for the diagnostic adapter!

Preconditions:

- The diagnostic adapter is connected to the controller at the diagnostic interface X6 and to the PC at a free USB port.
- The driver required for the diagnostic adapter is installed.
- The control electronics of the controller is supplied with 24 V low voltage via plug X2.

How to build up an online connection via the diagnostic adapter:

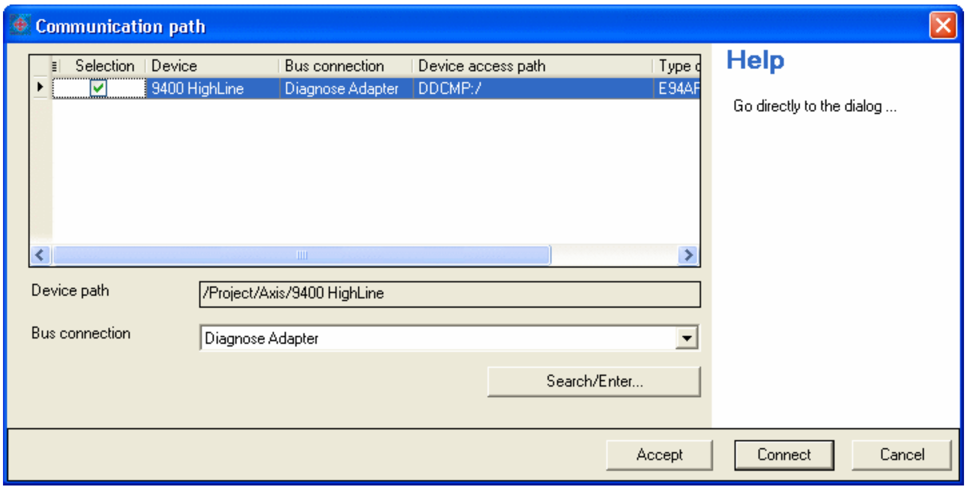
1. Select the 9400 HighLine controller to which you want to build up an online connection in the *Project view* of the »Engineer«:

2. Click the icon.

   If the changes you have made on the project have not been accepted yet, first a query on whether an update is to be carried out is effected.

   If an update is to be carried out:
   - Click on **Yes** to open the *Update project* dialog box.
   - Press the **Create** button in the *Update project* dialog box to update the changed project elements.
   - After the update a note is shown, saying whether the update was carried out successfully.

If no communication path was configured yet for the controller selected, the *Communication path* dialog box is shown after the update has been carried out:



- The "Diagnostic adapter" bus connection is already preset.

3. Click on **Connect**.
    - The dialog box is closed and the online connection with the controller is built up.
    - In the *Project view* a yellow icon indicates the online connection with the controller:

Now you can use the icons and to easily build up and end a connection with the controller. The communication settings are only required when communication with a controller is built up for the first time.

▶ If you want to change the configured communication path, select the command **Online → Set communication path and go online** to open the *Communication path* dialog box and change the settings.

▶ When an online connection has been established, the »Engineer« displays the current parameter settings of the controller with a yellow background colour.

2.2.2 Going online via system bus (CAN on board)

As an alternative to the diagnostic adapter, you can use the integrated system bus interface (CAN on board, terminal X1) of the controller for communication.

▶ 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!
- The online connection is established as described in the previous chapter "Going online via diagnostic adapter", only that this time the entry "CAN system bus" is to be selected in the **Bus connection** list field of the *Communication path* dialog box. (📖 25)

2.2.3 Use of other communication interfaces

The controller can be extended by further communication interfaces, if required, e.g. Ethernet, ETHERNET Powerlink, or PROFIBUS.

▶ For this the controller is provided with the module slots MXI1 and MXI2 for accepting communication modules.

▶ Detailed information on this subject can be found in the Hardware Manual and Communication Manual for the corresponding communication system.

2.3 Signal types & scaling

It is very helpful for the parameterisation & configuration of the controller to know the signal types and their scaling listed below, which serve to process physical quantities (e.g. a speed or position) within the function block interconnection.

Note!

From software version V3.0 the resolution of an encoder revolution can be parameterised in C00100 (Lenze setting: 16 bits/encoder revolution).

▶ Resolution of an encoder revolution (📖 44)

Signal type (data type)	Connection symbol in the FB editor	Resolution	Value range (external)	Decimal positions/ signal type suffix in the identifier	
Scaled (INT)	○	16 bits	± 199.99 %	2	_a
Scaled (DINT)	●	32 bits	± 200.00 %	2	_n
Speed (INT)	◁/▷	16 bits	± 30000.0 rpm	1	_v
Speed (DINT)	◆	32 bits	± 480000.0 rpm	1	_s
Position/angle (DINT)	◀/▶	32 bits	-2^{31} ... 2^{31}-1 increments	3	_p
Digital (BOOL)	□	1 bit	0 ≡ FALSE; 1 ≡ TRUE	0	
Acceleration (DINT)	■	32 bits	± 7.69 * 10^9 rpm/s	3	_x
Time	■	28 bits	0 ... 268435.456 s	3	
Other (BYTE)	■	8 bits	0 ... 255	0	
Other (WORD)	■	16 bits	0 ... 65535	0	
Other (DWORD)	■	32 bits	0 ... 4294967295	0	
Other (INT)	■	16 bits	-32768 ... 32767	0	
Other (DINT)	■	32 bits	-2147483648 ... 2147483647	0	

Scaling of physical quantities

Signal type	Connection symbol in the FB editor	Resolution	Scaling	
			External value	≡ internal value
Scaled (INT)	○	16 bits	100 %	≡ 2^{14} ≡ 16384
Scaled (DINT)	●	32 bits	100 %	≡ 2^{30} ≡ 1073741824
Speed (INT)	◁/▷	16 bits	15000 rpm	≡ 2^{14} ≡ 16384
Speed (DINT)	◆	32 bits	15000 rpm	≡ 2^{26} ≡ 67108864
Position/angle (DINT)	◀/▶	32 bits	1 encoder revolution	≡ 2^{16} increments
Acceleration (DINT)	■	32 bits	15000000 rpm/s	≡ 2^{22} ≡ 4194304

3 Commissioning

This documentation contains detailed information on parameter setting and configuration of the controller. Sequential reading is not required.

In order to obtain the information relevant for initial commissioning, this chapter describes different commissioning scenarios which can also be used as a guide through this manual:

A. Initial commissioning (📖 32)
   - **Target:** Adapting the controller to the electromechanics and the control system.

B. Standard set-up (📖 33)
   - **Target:** Taking over the application and parameter set of an already preconfigured "Engineer" project into several controllers.

C. Controller replacement (📖 34)
   - **Target:** Replacing a controller which has failed in a running system by a replacement device using the "old" memory module.

D. Motor replacement (📖 34)
   - **Target:** Replacing a motor which has failed in a running system.

3.1 General information

Note!

Some parameters of the controller have a setting range depending on the device type.

If parameterisation is carried out offline or if the memory module is exchanged between different 9400 HighLine device types, always check the settings of the parameters listed in the following table and adapt them, if required, to prevent a parameter error after the parameter set download or module change!

Parameter	Information	Lenze setting
C00018	Switching frequency	8 kHz variable
C00022	Maximum current Accepting/adapting plant parameters (132)	0.00 A
C00173 C00174	Mains voltage and undervoltage threshold (LU) Machine parameters (36)	400/415 V, LU = 285 V

Tip!

The rated data of the different device types can be found in the Hardware Manual in the "Rated data" chapter.

Term definition of "Plant parameters"

The term "plant parameters" which is frequently used in the following chapters summarises all parameters which result from the combination of motor and load. They characterise the transfer behaviour of the entire controlled system including the desired monitoring functions. The plant parameters depend on the application in which the controller and motor are used.

3.2 Notes on commissioning using the keypad

For a motor with an electronic nameplate (ENP)

- A display of the plant parameters offered by ENP via keypad is not provided. The plant parameters must be edited and optimised individually.
- To avoid that the motor starts unintentionally without adjusting the plant parameters, the maximum current in the Lenze setting is set to "0 A" in C00022.
- After setting the plant parameters, they have to be saved on the memory module of the controller with mains failure protection, just as the motor data that have been read out from the ENP (C00002 = "11: Save start parameters").

For a motor without an electronic nameplate (ENP)

- The motor data and plant parameters must be edited and set individually.
- To avoid that the motor starts unintentionally without adjusting the plant parameters, the maximum current is set to "0 A" in C00022 by the factory.
- After setting the motor data and plant parameters, they have to be saved on the memory module of the controller with mains failure protection (C00002 = "11: Save start parameters").

Commissioning of the application

- The application must already be stored on the memory module of the controller. Otherwise commissioning by only using the keypad is not possible.
- All application parameters which deviate from the factory adjustment have to be edited individually. For this the project planner has to provide a corresponding list to the commissioner (including the motor and plant data).
- In the case of a standard set-up, a pole position identification may have to be carried out for synchronous motors of a third party manufacturer or Lenze synchronous motors with a Stegmann absolute value encoder.
- After setting the parameters, they have to be saved on the memory module of the controller with mains failure protection (C00002 = "11: Save start parameters").

Tip!

Detailed information on the individual technology applications can be found in the corresponding Software Manual for the technology application and the »Engineer« online help in the chapter "L-force Servo Drives 9400 ➞ Technology applications".

3.3 Initial commissioning

Worksteps	
Parameterise motor control:	
1.	Read out the motor data of the controller or select them via the »Engineer« motor catalogue. • If the motor connected to the controller is provided with an electronic nameplate (ENP), all motor data are automatically read out from the ENP and a selection in the motor catalogue is not required. ▸ Reading out motor data from the controller (📖 125) • If a motor without ENP or a motor by a third-party manufacturer is used, the selection is carried out via the »Engineer« motor catalogue. ▸ Selecting motor in the »Engineer« motor catalogue (📖 126)
2.	Select motor control. (📖 129) • Servo control is preset for the synchronous motor.
3.	Adjusting motor and controller to each other (📖 131)
4.	Carry out settings for selected motor control. • For this see description for the corresponding motor control: – Servo control (SC) – Sensorless vector control (SLVC) (from software version V3.0) – V/f control (VFCplus) (from software version V3.0) – V/f control (VFCplus) (from software version V3.0)
Parameterise/configure application:	
5.	Load & parameterise technology application.
ℹ	Detailed information on the individual technology applications can be found in the corresponding Software Manual for the technology application and the »Engineer« online help in the chapter "L-force Servo Drives 9400 → Technology applications".
6.	If required, reconfigure the interconnection of the technology application with the function block editor.
Optimise control mode:	
7.	Optimise control mode of the selected motor control. • By means of traversing profile from the application and oscilloscope. • For this see description for the corresponding motor control: – Servo control (SC) – Sensorless vector control (SLVC) (from software version V3.0) – V/f control (VFCplus) (from software version V3.0) – V/f control (VFCplus) (from software version V3.0)
Save project and parameter set:	
8.	Execute device command C00002 = "11: Save start parameters".
9.	Save »Engineer« project.

More (optional) worksteps

Worksteps	
Establish network:	
1.	Insert network and machine application into the »Engineer« project.
2.	Interconnect port blocks reasonably to each other within the machine application.
3.	Configure network (set addresses, baud rate, and process data channels in a reasonable manner).
4.	Establish communication with the control system.
5.	Establish communication with other drive components (e.g. HMIs, I/O extensions and other controllers).

Worksteps	
Check & optimise application/DC-bus operation:	
1.	Traverse axis in manual operation. • See chapter Basic drive functions ▸ Manual jog (📖 398)
2.	Check area boundaries (path, speed, torque).
3.	Traverse axis in automatic operation with set-up speed, possibly together with coupled axes.
4.	Check coupling with other movements (master/slave axes, tools, ...).
5.	Optimisation of the process at higher speeds.
6.	Recording of typical signal characteristics using the oscilloscope function for the documentation. • See chapter Oscilloscope (📖 590)
Save & archive project and parameter set:	
1.	Execute device command C00002 = "11: Save start parameters".
2.	Save »Engineer« project.
3.	Deposit a backup copy of the »Engineer« project, e.g. on CD ROM, in the control cabinet.

3.4 Standard set-up

Worksteps	
Transfer application and parameter set to the controller:	
1.	Transfer the application preconfigured in the »Engineer« and the corresponding parameter set to the memory module of the controller.
2.	Execute device command C00002 = "11: Save start parameters".
For a motor with an electronic nameplate (ENP):	
3.	Restart controller with connected motor to read out the motor data from the electronic nameplate (ENP). • Either by switching off/switching on again the voltage supply or by means of device command C00002 = "11000: Restart controller". • See chapter Motor interface ▸ Reading out motor data from the controller (📖 125)
4.	Execute device command C00002 = "11: Save start parameters".
For a motor without an electronic nameplate (ENP):	
ℹ	**Note:** The motor is operated with the motor data and plant parameters identified during initial commissioning. ▸ Adjusting motor and controller to each other (📖 131)

3.5 Controller replacement

Scenario: The controller has failed in a running system.

Note!

For the procedure described in the following it is assumed that the memory module and possibly available extension modules in the controller, as well as the motor are not affected by the failure and that all parameters have been saved with mains failure protection.

Worksteps	
Replacement of the controller:	
1.	Replace controller. **See Mounting Instructions for the controller!**
2.	Insert the memory module of the failed controller into the replacement device.
3.	If further extension modules are plugged into the failed controller, they must be inserted into the replacement device as well.
Further steps are not required since all data required are on the memory module.	

3.6 Motor replacement

Scenario: The motor has failed in a running system.

Note!

For the procedure described in the following it is assumed that the controller is not affected by the failure.

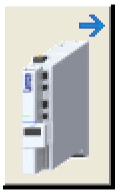
Worksteps	
Replacement of the motor:	
1.	Replace the motor. **See Mounting Instructions for the controller!**
	Note: The motor connection on the controller is accessible without having to remove the standard device from the installation backplane.
For a motor with an electronic nameplate (ENP):	
2.	Restart controller with connected motor to read out the motor data from the electronic nameplate. • Either by switching off/switching on again the voltage supply or by means of device command C00002 = "11000: Restart controller". • See chapter Motor interface ▸ Reading out motor data from the controller (📖 125)
3.	Execute device command C00002 = "11: Save start parameters".
For a motor without an electronic nameplate (ENP):	
	Note: The motor is operated with the motor data and plant data from the memory module.

4 Drive interface

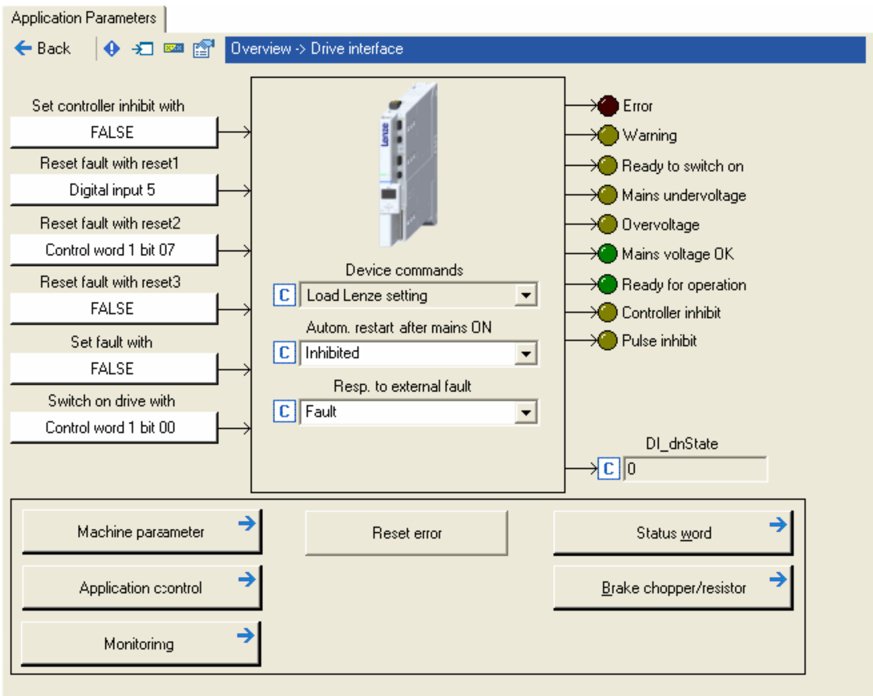
This chapter provides you with information on the drive interface via which you can control the drive controller into specific states and call different pieces of status information of the controller. Furthermore the machine constants for the motor end are entered via the drive interface.

How to get to the dialog for setting the drive interface parameters:

1. Go to the *Project view* of the »Engineer« and select the 9400 HighLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
3. Click the following button of the *Overview* dialog level:



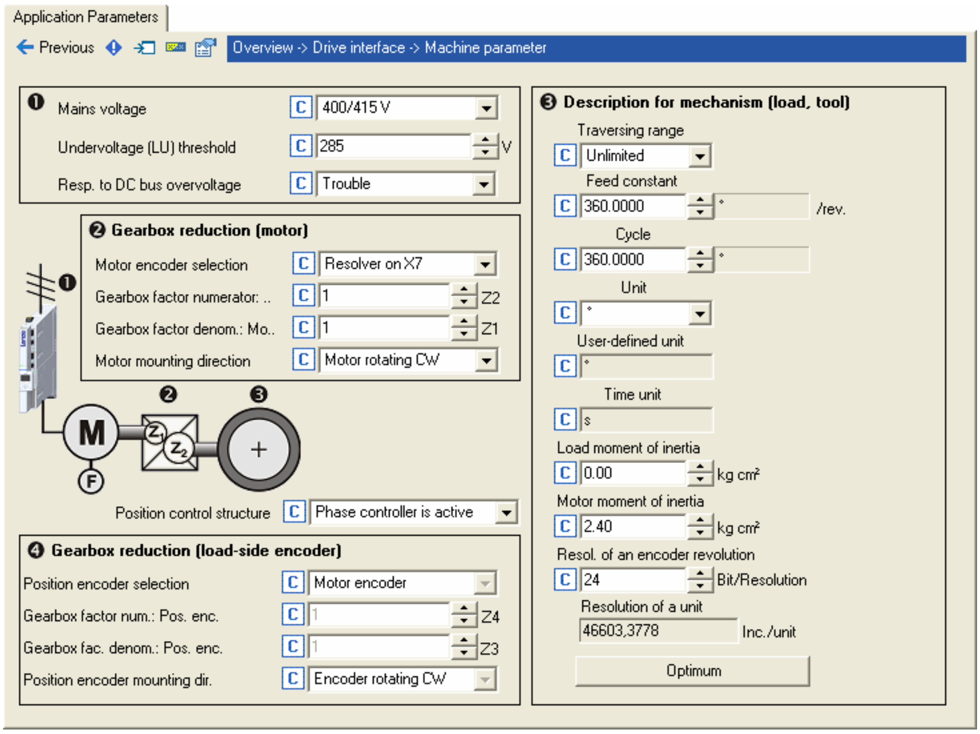
Parameterisation dialog in the »Engineer«



- The white buttons indicate the configuration of the drive interface inputs. ▶ Internal interfaces | "LS_DriveInterface" system block (📖 118)
  - The assignment is predefined by the technology application selected (in the example "Actuating drive – speed"). If required, this assignment configuration can be changed by clicking the corresponding buttons.
- If you click a button marked with the ➔ symbol, you go one level deeper in the corresponding parameterisation dialog.

4.1 Machine parameters

The global machine constants ("machine parameters") are set in the »Engineer» on the **Application parameters** tab in the dialog level *Overview* ➞ *Drive interface* ➞ *Machine parameters*:



Tip!

Detailed information on the different machine parameters can be obtained from the following subchapters.

4.1.1 Mains voltage

Via the **Mains voltage** list field (C00173) the mains voltage for the controller is set.

- If you set a mains voltage with an adjustable threshold for undervoltage ("LU adjustable"), this undervoltage threshold can be set in the **Undervoltage threshold (LU)** input field (C00174).
- In the **Resp. to DC-bus overvoltage** list field (C00600) you can select the response that is to be effected when a DC-bus overvoltage occurs.

Note!

Changing the setting in C00173 also affects the permissible device utilisation!

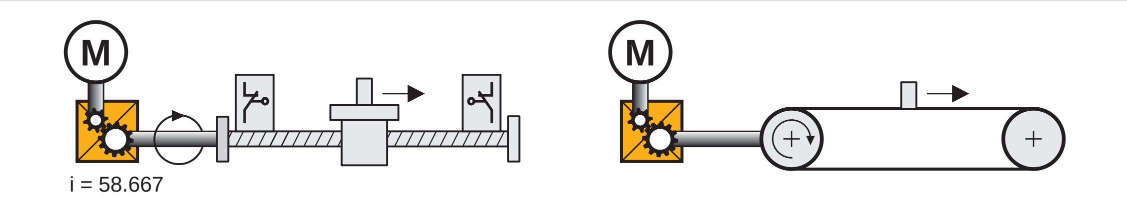
Tip!

In the chapter "Rated data" of the hardware manual the device types and their permissible device utilisation at a certain mains voltage and switching frequency are specified.

See also: Monitoring of the device utilisation (116)

4.1.2 Gearbox ratio

The gearbox ratio specifies the number of revolutions of the motor axis needed for one rotation of the load axis (e.g. spindle or drive roll).

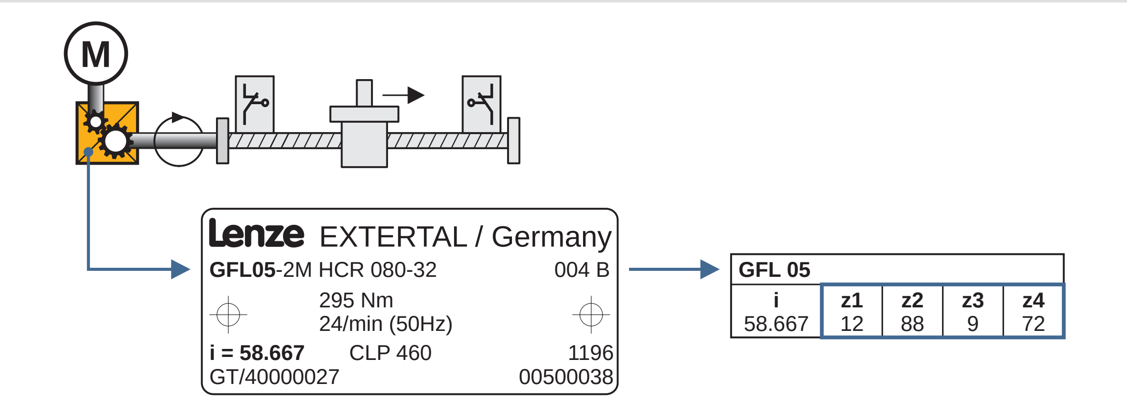


[4-1] Schematic diagram - gearbox ratio

- In the example shown in illustration [4-1] one revolution of the spindle is carried out at exactly 58,667 revolutions of the motor axis.

Selection of the gearbox ratio

- The gearbox ratio is to be defined in the form of a quotient (numerator/denominator); the data required can be found in the technical data for the gearbox:



[4-2] Example: Technical data for the gearbox (from the gearbox catalogue)

Tip!

For an exact selection of the gearbox ratio, use the number of teeth given in the data sheet or catalogue (see the following calculation) instead of the data on the nameplate.

In C02531/1 the gearbox factor is displayed in decimal format.

Example calculation on the basis of the technical gearbox data:

Gearbox factor numerator (C02520)	=	$z2 \times z4$	=	88×72	=	6336
Gearbox factor denominator (C02521)	=	$z1 \times z3$	=	12×9	=	108

[4-3] Calculation example

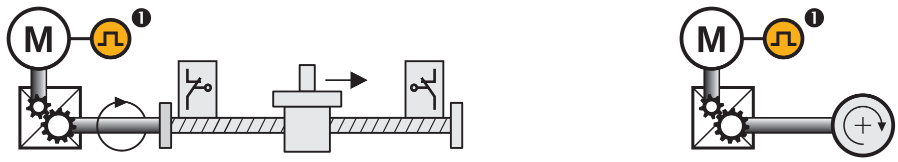
4.1.3 Motor mounting direction

Depending on the motor mounting position, you can carry out an inversion of the direction of rotation via the **Motor mounting direction** list field (C02527), if required:

- C02527 = "0": Clockwise rotating motor ≡ positive machine direction.
- C02527 = "1": Counter-clockwise rotating motor ≡ positive machine direction.

4.1.4 Feedback configuration

In most cases the system only has one motor encoder, i.e. no separate position encoder is installed on the load side. The motor position (angle of rotation) and motor speed are detected via the motor encoder selected in C00495 and converted with regard to the load side.



❶ Motor encoder

[4-4] Schematic diagram - feedback with position encoder = motor encoder

The actual position and speed values on the machine side result from the conversion via the Gearbox ratio on the motor side and the Feed constant.

Tip!

Detailed information on the parameterisation of the feedback systems for the motor control can be found in the chapter "Encoder evaluation". (□ 241)

4.1.5 Unit/user-defined unit

Via these machine parameters you define the real unit of the machine in which the feed constant and the parameters for a travel profile must be specified (e.g. position, speed, acceleration, and deceleration).

- If you for instance set the unit "mm" for a linear axis, the position must be specified in [mm] and the speed in [mm/s].
- By means of the user-defined unit, significant production units, like for example "bottles" can also be set.
  - For this, select the "User-defined" entry as unit in C02525 and then enter the desired user-defined unit in C02526.

Note!

In this documentation the term "unit" in the parameter unit data only serves as a wildcard for the real unit of the machine.

Display parameters

Parameter	Information
C02534	Used time unit
C02535	Used unit
C02537	Speed unit
C02538	Acceleration unit
Highlighted in grey = display parameter	

4.1.6 Traversing range

The selection of the traversing range ("Unlimited", "Limited", or "Modulo") in the **Traversing range** list field (C02528) serves to define the machine measuring system.

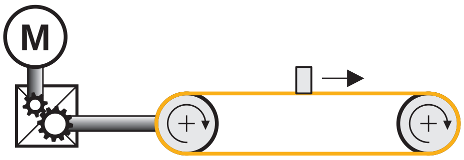
Note!

A change-over of the traversing range results in a loss of the reference information!

"Unlimited" traversing range

The drive can rotate continuously in one direction.

- By referencing and activating the software limit positions the traversing range can be limited.
- For positioning with absolute travel command the home position must be known.

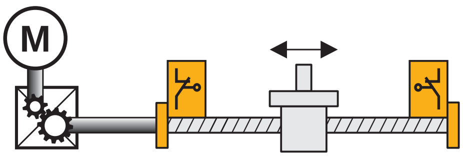


[4-5] Unlimited traversing range, taking the "feed control tape" as an example

"Limited" traversing range

The travel range is limited by positive and negative position limits (mechanical limits/ travel range limit switches/software limit positions). ▶ Limiter (📖 513)

- After a defined distance the drive must travel in the opposite direction again.
- For positioning in the limited traversing range the home position must be known.
- The software limit positions are basically monitored with regard to the maximum internally representable value range (±2^{31} increments), even if monitoring has been deactivated via C02700.
- An overflow of the value range results in a loss of the reference information.

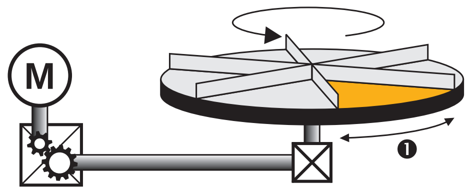


[4-6] Example: Limited traversing range - "spindle drive" (linear axis)

"Modulo" traversing range

The measuring system is repeated.

- If the cycle set in C02536 is exceeded, a defined overflow occurs. In a rotative system, the cycle typically corresponds to a revolution or tool distance.
- For positioning in the "Modulo" traversing range the home position must be known.
- Software limit positions are not effective.
- Absolute targets can be approached by exceeding the measuring system limit, e.g. from 10° to 350°.



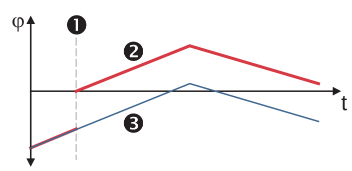
❶ C02536: Cycle (illustration = 60°)

[4-7] Example: Modulo traversing range - "rotary table""

Dependencies - traversing range/basic drive functions

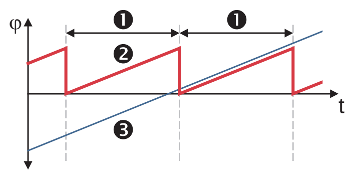
- The following table lists the different dependencies between the selected traversing range and the basic drive functions.

Basic drive function	Traversing range		
	Unlimited	Limited	Modulo
Position data for Encoder evaluation	continuously	continuously	clocked
Position data for Position follower	absolute	absolute	absolute (in time)
Positioning modes for Positioning	1, 2, 5, 6, 7, 8	1, 2, 5, 6, 7, 8	5, 6, 11 ... 16
Restrictions for Homing	none	none	home position must be in time
Limit positions (Limiter)	permitted	permitted	not permitted



Example 1: Unlimited/limited position display

❶ Reference setting
❷ Position in the machine measuring system
❸ Position in the motor measuring system



Example 2: Modulo position display

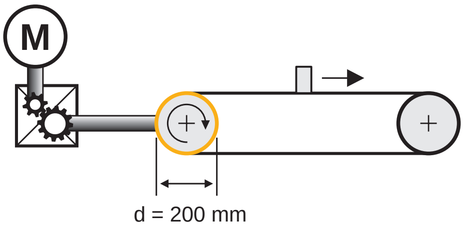
❶ Cycle
❷ Position in the machine measuring system
❸ Position in the motor measuring system

Lenze

4.1.7 Feed constant

The feed constant corresponds to the movement of the machine during one revolution of the gearbox output shaft.

- The entry in the **Feed constant** field (C02524) is made in the unit defined in C02525 relating to one revolution.
- In the case of a conveyor drive, the feed constant results from the circumference of the drive roll, which in the following example is calculated on the basis of the diameter specified:

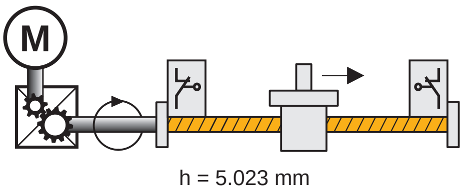


d = diameter

$$\text{Feed constant} = \pi \cdot d \frac{[\text{unit}]}{\text{Revolution}} = \pi \cdot 200 \frac{\text{mm}}{\text{Revolution}} = 628.3185 \frac{\text{mm}}{\text{Revolution}}$$

[4-8] Schematic diagram: Feed constant for a conveyor drive

- In the case of a spindle drive (linear axis), the feed constant results from the leadscrew pitch. The feed constant indicates the distance travelled by the slide during one revolution of the spindle (in the following example 5,023 mm).



h = leadscrew pitch (can be obtained from the technical data of the linear axis)

[4-9] Schematic diagram: Feed constant for a spindle drive

- For a rotary table and defined as an angle, the feed constant is = 360°/rev.

4.1.8 Resolution of an encoder revolution

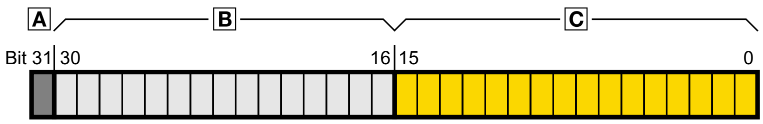
The following applies to software versions lower than V3.0:

The resolution of an encoder revolution and hence of a position value is constantly set to 16 bits/revolution, which corresponds to 65536 increments/revolution. At this resolution, the traversing range comprises ±32767 revolutions.

The following applies from software version V3.0:

C00100 serves to adjust the resolution according to the application.

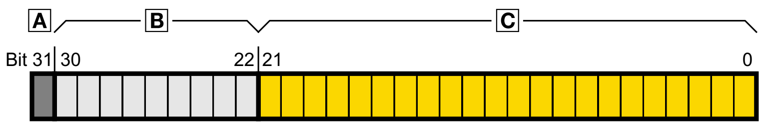
- The default resolution of 16 bits/revolution is sufficient for standard applications.



A Sign bit
B Number of revolutions: 15 bits ≡ ±32767 revolutions
C Resolution of one encoder revolution: 16 bits ≡ 65536 increments/revolution

[4-10] Example: standard resolution (16 bits/revolution)

- For more significant applications, a higher resolution of the position values can clearly improve the control properties and positioning accuracies:
  - Finer resolution of the position targets ➞ improved positioning accuracy
  - Finer quantisation of setpoints and actual values ➞ better control quality
  - Higher loop gain can be set ➞ less following errors
- However, a higher resolution at the same time causes a restricted number of encoder revolutions, and only smaller traversing distances can be displayed.



A Sign bit
B Number of revolutions: 9 bits ≡ ±512 revolutions
C Resolution of one encoder revolution: 22 bits ≡ 4194304 increments/revolution

[4-11] Example: Higher resolution (22 bits/revolution) with a restricted traversing range

Tip!

In the following subchapter "Determining the optimum resolution" (🕮 46) it is described how you can determine the optimum resolution of the position values.

Note!

The position values (e.g. setpoints, actual values, parameters, ...) in the signal flow always use the resolution set in C00100. In this connection it is irrelevant which resolution is delivered directly by the encoder.

Multi-axis systems

In an interconnection via the electrical shaft, at least two measuring systems (master and slave) are available in the drive.

- Each measuring system is provided with an individual setting of the resolution.
- The machine parameters (gearbox factors, feed constants, encoder resolution and cycle) for the master measuring system or master value must be set identically for all drives in the system.

Technology applications "Electronic gearbox" and "Synchronism"

For these two technology applications the machine parameters of the master measuring system are defined on the *Application parameters* tab in the "Master value scaling" dialog level.

Electronic cam

The machine parameters of the master measuring system for electronic cams can be defined on the *Measuring systems* tab for the electrical shaft.

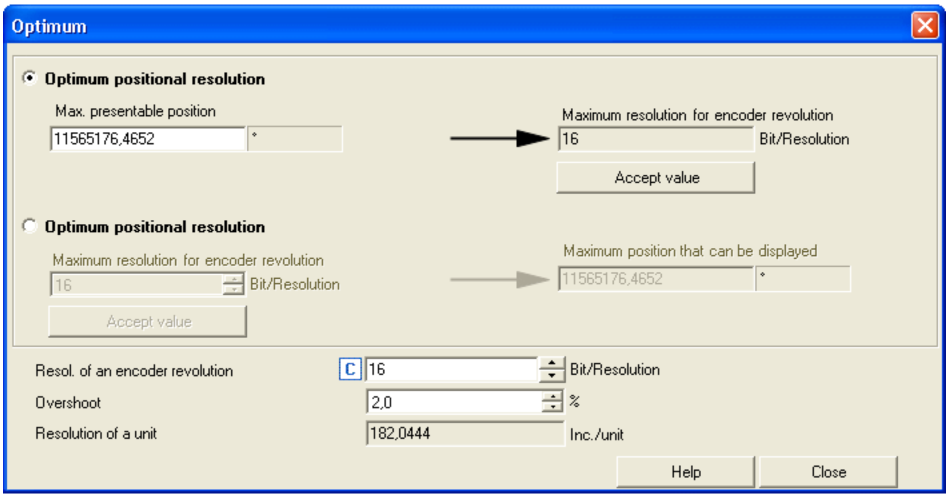
4.1.8.1 Determining the optimum resolution

This function extension is available from software version V3.0!

How to determine the optimum resolution:

In the dialog level *Overview* ➞ *Drive interface* ➞ *Machine parameters:*

1. Set gearbox factors.
2. Set real unit of the machine.
3. Set feed constant.
4. Press the **Optimum positional resolution** button.
   - The *Optimum positional resolution* dialog box is displayed:



5. Go to the **Max. presentable position** input field and enter the highest position which is to be entered in a parameter during operation.
   - If required, set a reserve in the **Overshoot** input field to take into account possible following errors (overshoot of actual values).

   Then the maximum resolution for the position entered is shown in the **Maximum resolution for encoder revolution** field.
6. Click on **Accept value** to accept the displayed resolution in C00100.
7. Click on **Close** to close the dialog box again.

Tip!

In order to have the maximally presentable position for a defined resolution displayed, activate the second option **Determine max. presentable position**. Then you can set the resolution for which the maximally presentable position is to be displayed in the **Maximum resolution for encoder revolution** input field.

4.1.9 Max. position, speed, and acceleration that can be displayed internally

By setting the following machine parameters, the connection between the real units (application units) of the machine and the internal units in the controller is described:

- Gearbox ratio (C02520, C02521, C02522, C02523)
- Feed constant (C02524)
- Resolution of an encoder revolution (C00100)

Possibly the defined values for position, speed, and acceleration cannot be represented in the internal units by the numerical 32-bit format used.

- The following display parameters show the values that can be maximally displayed:

Parameter	Information	Lenze setting	
		Value	Unit
C02539	Max. presentable position	-	Unit
C02540	Speed that can be maximally displayed	-	Unit/s
C02541	Acceleration that can be maximally displayed	-	Unit/s^2
Highlighted in grey = display parameter			

Behaviour when entering a value that cannot be displayed internally

If a position, speed, or acceleration which cannot be represented internally is defined via parameters, the value defined is limited to the maximum value that can be represented internally (±2147483647).

The following only applies to software version V3.0:

- If a position, speed, or acceleration which cannot be represented internally is defined via parameters, the value defined is rejected.
- If an internal counter overflow of a parameter value due to a subsequent change of the machine parameters for the gearbox ratio, feed constant, or resolution of an encoder revolution is detected, the "Fault" error response is triggered and a corresponding error message is entered in the logbook of the controller:

Error number	Error message
0x00B8001A	Int. overflow C02620 (manual speed 1)
0x00B8001B	Int. overflow C02621 (manual speed 2)
0x00B8001C	Int. overflow C02622 (manual acceleration)
0x00B8001D	Int. overflow C02624 (manual deceleration)
0x00B80020	Int. overflow C02701/1 (positive SW limit position)
0x00B80021	Int. overflow C02701/2 (negative SW limit position)
0x00B80022	Int. overflow C02703 (maximum speed)
0x00B80023	Int. overflow C02705 (maximum acceleration)
0x00B80024	Int. Overflow C02708/1 (Limited speed 1)
0x00B80025	Int. Overflow C02708/2 (Limited speed 2)
0x00B80026	Int. Overflow C02708/3 (Limited speed 3)
0x00B80027	Int. Overflow C02708/4 (Limited speed 4)
0x00B80028	Int. overflow C02710/1 (decel. limited speed 1)
0x00B80029	Int. overflow C02710/2 (decel. limited speed 2)
0x00B8002A	Int. overflow C02710/3 (decel. limited speed 3)
0x00B8002B	Int. overflow C02710/4 (decel. limited speed 4)
0x00B8002C	Int. overflow C02713 (maximum distance manual jog)
0x00B8002D	Int. Overflow C02642 (home position)
0x00B8002E	Int. Overflow C02643 (homing: target position)
0x00b8002f	Int. Overflow C02644 (homing: speed 1)
0x00B80030	Int. Overflow C02645 (homing: acceleration 1)
0x00B80031	Int. Overflow C02646 (homing: speed 2)
0x00B80032	Int. Overflow C02647 (homing: acceleration 2)
0x00B80033	Int. overflow C02670 (positioning: tolerance for target position)

Tip!

Possible measures for error correction:

- Plausibility check of the machine parameters set for gearbox ratio, feed constant, or resolution of an encoder revolution.
- Set parameters with a counter overflow to a value which can also be represented internally.

4.2 Device commands

In the following subchapters the device commands of the controller are described, which are provided in C00002 and which can be executed by means of the »Engineer« or alternatively with the keypad when an online connection has been established.

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!

The meaning of the status display in C00003 can be obtained from the subchapter for the corresponding device command.

Activating frequently required device commands via the toolbar

The simplest way to execute the frequently required device commands is directly via the *Toolbar* of the »Engineer« when an online connection has been established.

Icon	Function
	Enable controller
	Inhibit controller
	Start application
	Inhibit controller and Stop application

Note!

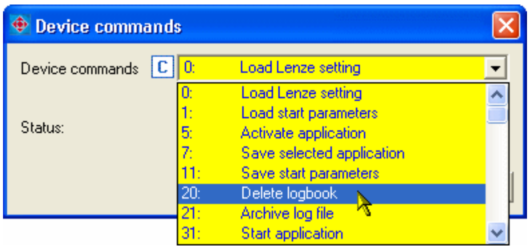
Device commands that can be executed via the *Toolbar* of the »Engineer« always affect the element currently selected in the *Project View* including all subelements!

- 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 executing the corresponding action a confirmation prompt is displayed, asking you whether the action should really be carried out.

Activating device commands via the "Device commands" dialog box

All device commands of the controller are available in the »Engineer« in the *Device Commands* dialog box:

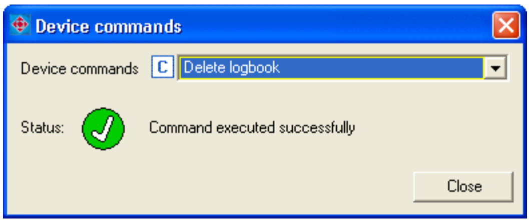


- The *Device commands* dialog box can be opened by clicking on the **Device commands** list field on the **Application parameters** tab in the dialog level *Overview* → *Drive interface*.
- The *Device commands* dialog box can also be opened by clicking on the setting of C00002 on the **All parameters** tab.

Note!

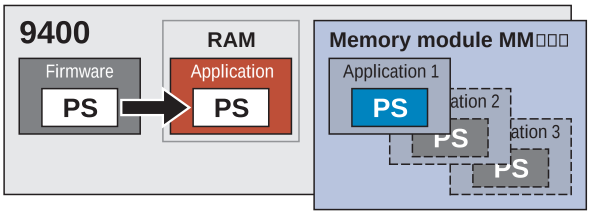
If you click on a device command in the list field of the *Device commands* dialog box, the corresponding device command is executed immediately!

- During and after the execution of the device command, the processing status is displayed in the *Device Commands* dialog box:



4.2.1 Load Lenze setting

The C00002 = "0: Load Lenze setting" device command is used to reset the parameters of the active application to the Lenze setting, which is stored in the controller firmware:



[4-12] "Load Lenze setting" function

- Only possible when the application has stopped and the controller is inhibited.
- All parameter changes made since the last saving of the parameter set will get lost!
- This device command only affects the settings of the operating system, application and module parameters, the active application or the configuration selected with the function block editor remain unchanged.

Possible status displays for this device command

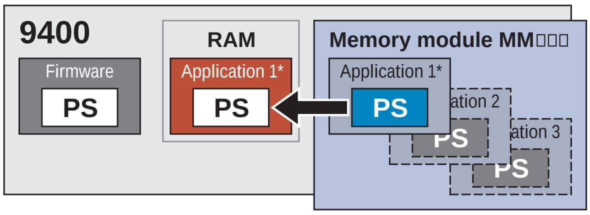
Status (C00003)		Meaning
	34050	Device command in process
	0	Device command executed successfully
	1	General fault
	39424	CAN fault
	...	...
	39679	CAN fault

Related device commands

- Load start parameters (52)
- Save start parameters (56)

4.2.2 Load start parameters

Via C00002 = "1: Load start parameters" the start parameters of the active application can be reloaded from the memory module to the controller:



* In this example, application 1 is the active application

[4-13] "Load start parameters" function

- Only possible when the application has stopped and the controller is inhibited.
- All parameter changes made since the last saving of the parameter set will get lost!
- This device command only affects the settings of the operating system, application and module parameters, the active application or the configuration selected with the function block editor remain unchanged.

Possible status displays for this device command

Status (C00003)		Meaning
⌛	99586	Device command in process
✓	65536	Device command executed successfully
✗	65537	General fault
	99371	Fault while reading the parameter set partition
	99374	No memory module available
	104960	CAN fault
	...	...
	105215	CAN fault

Related device commands

- Save start parameters (56)
- Load Lenze setting (51)

4.2.3 ENP:Load plant data

If the Lenze motor connected to the controller is provided with an electronic nameplate (ENP), all motor data are automatically read out from the electronic nameplate of the motor when the controller is switched on for the first time and are temporarily stored in the controller at first.

With the device command C00002 = "2: ENP: Load plant data" the motor data can be reread from the electronic nameplate (ENP) of the motor.

- Only possible when the application has stopped and the controller is inhibited.
- For a permanent acceptance of the motor data, the parameter set must be saved. ▶ Save start parameters (📖 56)
- The following plant data are read out from the ENP:

Parameter	Information
C00022	Maximum current
C00070	Speed controller gain
C00071	Speed controller reset time
C00596	Threshold max. speed reached

Note!

The two pieces of plant data C00011 and C00497 listed in the following table are not read out from the ENP and thus have to be checked and, if required, set manually after this device command has been executed!

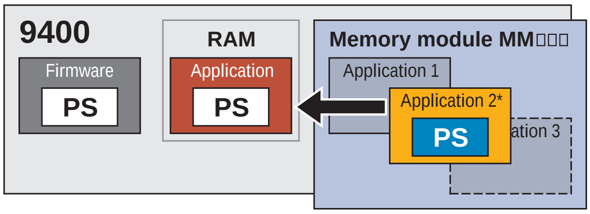
Parameter	Information
C00011	Motor reference speed
C00497	Speed act. val. time const.

Possible status displays for this device command

Status (C00003)		Meaning
	165122	Device command in process
	131072	Device command executed successfully
	131073	General fault

4.2.4 Activate application

If several applications are available on the memory module, the C00002 = "5: Activate application" device command can be used to activate the application the number of which has been set in C00005.



* In this example, application selection "2" is set in C00005

[4-14] "Activate application" function

- Only possible when the application has stopped and the controller is inhibited.
- Whether the application is started at the same time, depends on the auto-start setting selected in C02104.
- After mains switching, the preset application will be loaded into the controller.
- If after mains switching another application than the one preset by Lenze is to be loaded, it must be activated first and then the selected application must be saved with the device command "Save selected application" (📖 55).
- The number of the currently active application is displayed in C00007.

Note!

When the application is activated, the corresponding start parameter set is loaded automatically and parameter settings executed before will get lost unless the parameter set was saved before!

Possible status displays for this device command

Status (C00003)		Meaning
	361730	Device command in process
	327680	Device command executed successfully
	327681	General fault

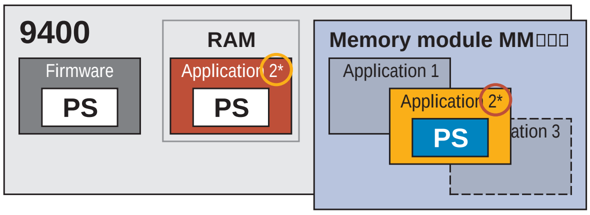
Related device commands

- Save selected application (📖 55)
- Start application (📖 60) / Stop application (📖 61)

4.2.5 Save selected application

After mains switching the controller always loads the preset start application from the memory module, even if a different application has been active before.

With the device command C00002 = "7: Save selected application" the active application can be defined as start application.



* In this example, the active application 2 is defined as start application

[4-15] "Save selected application" function

- When this device command is executed, the parameter set is also saved automatically.
- The number of the currently active application is displayed in C00007.

Possible status displays for this device command

Status (C00003)		Meaning
	492802	Device command in process
	458752	Device command executed successfully
	458753	General fault

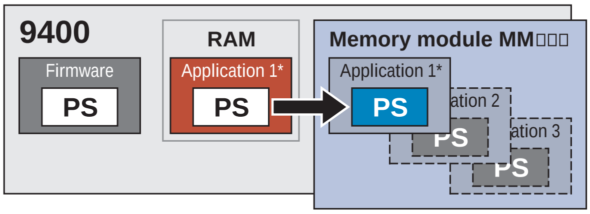
Related device commands

- Activate application (📖 54)
- Start application (📖 60)
- Stop application (📖 61)

4.2.6 Save start parameters

Controller parameter changes made via the »Engineer« or keypad will get lost after mains switching of the controller or loading of another application unless the settings have been explicitly saved.

With the device command C00002 = "11: Save start parameters" the current parameter settings of the active application can be saved with mains failure protection in the memory module of the controller:



* In this example, application 1 is the active application

[4-16] "Save start parameters" function

Tip!

With the keypad this device command can be executed via the left function key if it is currently assigned with the SAVE function.

Note!

The saving process can take several seconds. Before you switch off the supply voltage after having executed this device command, therefore be absolutely sure to check via the status display in C00003 whether the device command has been executed successfully!

Saving of the cam data

From software version V4.0, this device command also includes the powerfail-proof saving of the cam data on the memory module.

- The saving process is only carried out if the cam data in the controller and the memory module differ from each other (based on the time stamp/GUID of the cam data).
- For saving the cam data, you do not need to enter a possibly existing user password (C02900).
- The C00002 = "502: Save Cam Data" device command remains available. ▶ Save cam data (📖 97)

Possible status displays for this device command

Status (C00003)		Meaning
	754946	Device command in process
	720896	Device command executed successfully
	720897	General fault
	754718	Fault while writing into a file
	754734	No memory module available
	761857	Access to file has been denied since the file is already accessed from another position
	761861	I/O fault when accessing the file system
	761868	RAM is full
	761869	Access authorisation denied
	761884	No free memory on the memory module

Related device commands

- Load start parameters (📖 52)

4.2.7 Delete logbook

The C00002 = "20: Delete logbook" device command is used to delete all entries in the logbook.

Tip!

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

You can also delete all entries in the logbook by clicking the **Delete** button in the *Logbook* dialog box.

Further information on the logbook can be found in the chapter "Diagnostics & fault analysis". (611)

Possible status displays for this device command

Status (C00003)		Meaning
	1344770	Device command in process
	1310720	Device command executed successfully
	1310721	General fault

Related device commands

- Archive logbook (59)

4.2.8 Archive logbook

The C00002 = "21: Archive logbook" device command is used to archive the entries in the logbook.

Tip!

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

You can also export all entries available in the logbook into a file (*.log) by clicking the **Export** button in the *Logbook* dialog box.

Further information on the logbook can be found in the chapter "Diagnostics & fault analysis". (📖 611)

Possible status displays for this device command

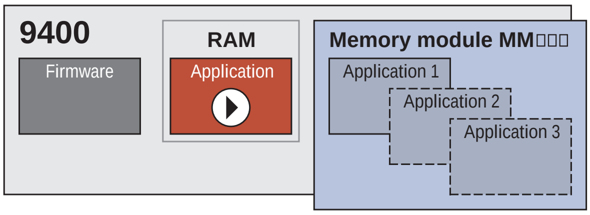
Status (C00003)		Meaning
	1410306	Device command in process
	1376256	Device command executed successfully
	1376257	General fault

Related device commands

- Delete logbook (📖 58)

4.2.9 Start application

The C00002 = "31: Start application" device command is used to start the active application in the controller.



[4-17] "Start application" function

- The number of the currently active application is displayed in C00007.
- The current program status is displayed in C02108.
- The active function state of the application is displayed in C02530.

Tip!

This device command can also be activated via the icon in the *Toolbar*.

Possible status displays for this device command

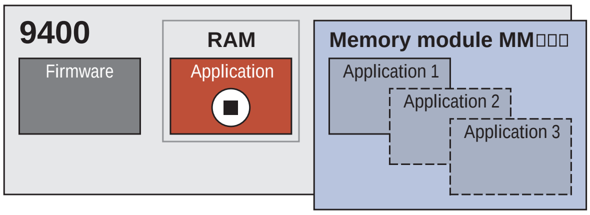
Status (C00003)		Meaning
	2065666	Device command in process
	2031616	Device command executed successfully
	2031617	General fault

Related device commands

- Stop application (📖 61)
- Activate application (📖 54)
- Save selected application (📖 55)

4.2.10 Stop application

The C00002 = "32: Stop application" device command can be used to stop the application started in the controller again.



[4-18] "Stop application" function

- Only possible when controller is inhibited.

Tip!

Via the icon in the *Toolbar* the controller can be inhibited, and at the same time the application in the controller can be stopped.

Possible status displays for this device command

Status (C00003)		Meaning
	2131202	Device command in process
	2097152	Device command executed successfully
	2097153	General fault

Related device commands

- Start application (60)
- Inhibit controller (67)
- Activate application (54)
- Save selected application (55)

4.2.11 Reset program

The C00002 = "33: Reset program" device command is used to reset the application program in the controller.

- All variables are reset to their initialisation value.
- The situation corresponds to the start of a new program loaded into the control (cold start).

Possible status displays for this device command

Status (C00003)		Meaning
⌛	2196738	Device command in process
✔	2162688	Device command executed successfully
✖	2162689	General fault

Related device commands

- Delete program (📖 63)
- Restart program (📖 64)

4.2.12 Delete program

The C00002 = "34: Delete program" device command is used to delete the application program in the controller and reset the controller to its original state.

- All variables are reset to their initialisation value.

Possible status displays for this device command

Status (C00003)		Meaning
	2262274	Device command in process
	2228224	Device command executed successfully
	2228225	General fault

Related device commands

- Reset program (📖 62)
- Restart program (📖 64)

4.2.13 Restart program

The C00002 = "35: Restart program" device command is used to restart the application program in the controller.

- All variables except the RETAIN variables are reset to their initialisation value.
- The situation corresponds to a power failure or switching the controller off/on (warm start) while the program is running.

Possible status displays for this device command

Status (C00003)		Meaning
	2327810	Device command in process
	2293760	Device command executed successfully
	2293761	General fault

Related device commands

- Reset program (62)
- Delete program (63)

4.2.14 Reset runtime measurement

When the application is started, the controller continuously carries out a runtime measurement for the interval-controlled application task, the interval-controlled user task, and the free-running idle task and displays the current and maximum task runtimes via parameters.

The C00002 = "36: Reset runtime measurement" device command is used to reset the runtime measurement, i.e. the memory for the maximum values is reset to "0".

Possible status displays for this device command

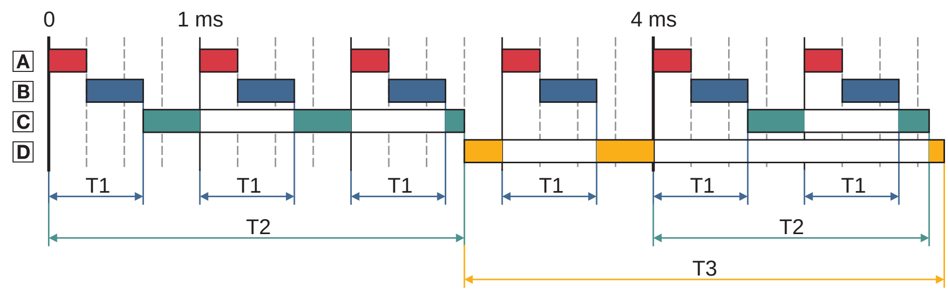
Status (C00003)		Meaning
	2393346	Device command in process
	2359296	Device command executed successfully
	2359297	General fault

Note!

The runtime measurement is also reset by the following actions:

- Start application
- Reset/delete/restart program

Example for runtime measurement



A System task
B Application task (here: task interval = 1 ms)
C User task (here: task interval = 4 ms)
D Free-running idle task

T1 = runtime of the application task
T2 = runtime of the user task
T3 = runtime of the idle task

[4-19] Example: runtimes of the different tasks

Display parameters

Parameter	Information	Lenze setting	
		Value	Unit
C02121/1	Current runtime - application task	-	µs
C02121/2	Maximum runtime - application task	-	µs
C02122/1	Current runtime - user task	-	µs
C02122/2	Maximum runtime - user task	-	µs
C02123/1	Current runtime - idle task	-	µs
C02123/2	Maximum runtime - idle task	-	µs
Highlighted in grey = display parameter			

4.2.15 Inhibit controller

The C00002 = "41: Inhibit controller" device command is used to inhibit the controller ("controller inhibit"), i.e. the power output stages in the controller are inhibited and the speed/current and position controllers of the motor control are reset. The motor becomes torqueless and coasts unless it is already at standstill.

- The controller can also be inhibited by other sources, e.g. via the digital input RFR or through the application.
- The sources or triggers that are active for controller inhibit are shown in C00158 in a bit coded manner.

Note!

This device command has no status display in C00003, i.e. the display remains unchanged showing the previous device command status.

Tip!

This device command can also be activated via the icon in the *Toolbar*.

Related device commands

- Enable controller (68)

4.2.16 Enable controller

The C00002 = "42: Enable controller" device command is used to re-enable an inhibited controller.

Note!

Please note that the controller will only be enabled if all sources for controller inhibit are reset!

- The sources or triggers that are active for controller inhibit are shown in C00158 in a bit coded manner.

This device command has no status display in C00003, i.e. the display remains unchanged showing the previous device command status.

Tip!

This device command can also be activated via the icon in the *Toolbar*.

Related device commands

▶ Inhibit controller (📖 67)

4.2.17 Reset error

The C00002 = "43: Reset error" device command is used to acknowledge an error message if the error cause has been eliminated and the error is thus no longer pending.

Tip!

An error message can also be acknowledged by clicking the **Reset error** button on the **Diagnostics** tab.

Further information on error messages can be found in the chapter "Diagnostics & fault analysis". (611)

Note!

This device command has no status display in C00003, i.e. the display remains unchanged showing the previous device command status.

4.2.18 Activate quick stop

The C00002 = "45: Activate quick stop" device command is used to activate the basic function "Quick stop", i. e. the drive is brought to standstill within the deceleration time set, irrespective of the setpoint defined.

- Quick stop can also be activated by other sources, e.g. by the application.
- The sources or triggers that are active for quick stop are shown in C00159 in a bit coded manner.

Note!

The activation of quick stop may cause following errors in superimposed controls (e.g. synchronous or position control). If several drives execute a coordinated movement, the quick stop function should therefore only be used for the motion master (master drive) in order to maintain the coordination.

This device command has no status display in C00003, i.e. the display remains unchanged showing the previous device command status.

Tip!

In contrast to the "stop" function, quick stop is required for a stop in case of error. Thus, quick stop can also be set as an error response ("quick stop by trouble) for many monitoring functions. Detailed information on this can be found in the chapter "Diagnostics & fault analysis". (📖 611)

Related device commands

- Reset quick stop (📖 71)

4.2.19 Reset quick stop

The C00002 = "46: Reset quick stop" device command is used to exit an active quick stop again.

Note!

Please note that the quick stop is only exited if all sources for quick stop are reset!

- The sources or triggers that are active for quick stop are shown in C00159 in a bit coded manner.

This device command has no status display in C00003, i.e. the display remains unchanged showing the previous device command status.

Related device commands

▶ Activate quick stop (📖 70)

4.2.20 Identify pole position (360°)

If no absolute value encoder is connected or a synchronous motor of a third-party manufacturer is driven by the controller, the C00002 = "51: Identify pole position (360°)" device command is used to determine the pole position with regard to the motor encoder currently activated in C00495.

- The function can only be activated if the controller is inhibited. Then the execution of the function starts automatically as soon as the controller inhibit is deactivated again.
- During the pole position identification, the motor makes one electrical revolution. This leads to a mechanical rotation of the motor shaft.
- The determined pole position is indicated under code C00058.

Note!

From software version V4.0 the response parameterised in C00640 (Lenze setting: "Fault") is triggered and the error message "Pole position identification cancelled" is entered in the logbook of the controller if the pole position identification process is aborted.

Tip!

Detailed information on the pole position identification can be found in the chapter "Motor interface", subchapter "Pole position identification". (📖 136)

Possible status displays for this device command

Status (C00003)		Meaning
⌛	3376386	Device command in process
✓	3342336	Device command executed successfully
✗	3342337	General fault
	3382023	Pole position identification cannot be executed because of wrong motor type (asynchronous motor).
	3382024	Pole position identification has been aborted
	3382025	Pole position identification cannot be executed because another identification is already active.
	3382026	Identification of pole position cannot be executed because U-rotation or I-rotation test mode is active.
	3382027	Identification of pole position cannot be executed because current controller optimisation mode is active.
	3382033	Pole position identification cannot be executed because the motor is blocked (e.g. by a mechanical brake), a motor phase is not connected, or a phase shifter is in the motor cable.
	3382047	Pole position identification cannot be executed because an error or trouble is active.
	3382065	Pole position identification cannot be executed because either the entire motor or a motor phase is not connected. • This error message is only available from software version V3.0 onwards.

Related device commands

▶ Identify pole position (min. motion) (📖 74)

4.2.21 Identify pole position (min. motion)

If no absolute value encoder is connected or a synchronous motor of a third-party manufacturer is driven by the controller, the C00002 = "52: Identify pole position (min. motion)" device command is used to determine the pole position with respect to the motor encoder currently activated in C00495.

- The function can only be activated if the controller is inhibited. Then the execution of the function starts automatically as soon as the controller inhibit is deactivated again.
- During the pole position identification, the rotor aligns itself. This is compensated by a position control.
- The determined pole position is indicated under code C00058.

Note!

From software version V4.0 the response parameterised in C00640 (Lenze setting: "Fault") is triggered and the error message "Pole position identification cancelled" is entered in the logbook of the controller if the pole position identification process is aborted.

Tip!

Detailed information on the pole position identification can be found in the chapter "Motor interface", subchapter "Pole position identification". (📖 136)

Possible status displays for this device command

Status (C00003)		Meaning
In process	3441922	Device command in process
Successful	3407872	Device command executed successfully
Fault	3407873	General fault
	3447559	Pole position identification cannot be executed because of wrong motor type (asynchronous motor).
	3447560	Pole position identification has been aborted
	3447561	Pole position identification cannot be executed because another identification is already active.
	3447562	Identification of pole position cannot be executed because U-rotation or I-rotation test mode is active.
	3447563	Identification of pole position cannot be executed because current controller optimisation mode is active.
	3447569	Pole position identification cannot be executed because the motor is blocked (e.g. by a mechanical brake), a motor phase is not connected, or a phase shifter is in the motor cable. • This error message is only available from software version V4.0 onwards.
	3447583	Pole position identification cannot be executed because an error or trouble is active.
	3447597	Identification of pole position cannot be executed because the rotor has moved too strongly.
	3447601	Pole position identification cannot be executed because either the entire motor or a motor phase is not connected. • This error message is only available from software version V3.0 onwards.

Related device commands

- Identify pole position (360°) (📖 72)

4.2.22 Resolver error identification

This function extension is available from software version V7.0!

The C00002 = "59: Resolver error identification" device command serves to detect resolver errors which are caused when sine and cosine tracks do not magnetise orthogonally. The identified resolver errors serve to compensate the resolver errors.

- Only possible with servo control.

Tip!

Detailed information on the resolver error compensation can be found in the chapter "Encoder evaluation" in the subchapter "Resolver error compensation". (🕮 261)

Possible status displays for this device command

Status (C00003)		Meaning
⌛	3900674	Device command in process
✓	3866624	Device command executed successfully
✗	3866625	General fault
	3906358	Resolver error identification cannot be executed since the wrong control type is active (no servo control).
	3906359	Resolver error identification cannot be executed since an error or trouble is active.
	3906360	Resolver error identification cannot be executed because another identification is already active.
	3906361	Resolver error identification cannot be executed because of too small speed (< 500 rpm).

4.2.23 Load Lenze inverter characteristic

This function extension is available from software version V4.0!

If it is not possible to determine the so-called "Inverter error characteristic" by means of the "Calculate inv. characteristic" device command, or if the results of the determination are incorrect, the C00002 = "70: Load Lenze inverter characteristic" device command can be used to load a characteristic typical for the device.

- Only possible when controller is inhibited.

Tip!

Detailed information about the determination of the inverter error characteristic can be found in the chapter "Motor interface" in the subchapter "Optimising the switching performance of the inverter". (143)

Possible status displays for this device command

Status (C00003)		Meaning
	4621570	Device command in process
	4587520	Device command executed successfully
	4587521	General fault

Related device commands

- Calculate inv. characteristic (78)

4.2.24 Calculate inv. characteristic

If a motor of a third-party manufacturer with unknown motor parameters is driven by the controller, the C00002 = "71: Determine inverter characteristic" device command can be used to determine the so-called "Inverter error characteristic" for optimising the inverter switching behaviour.

Tip!

Detailed information about the determination of the inverter error characteristic can be found in the chapter "Motor interface" in the subchapter "Optimising the switching performance of the inverter". (143)

From software version V4.0: If the inverter error characteristic cannot be determined by means of this device command, or if the results of the determination are incorrect, the device command "Load Lenze inverter characteristic" can be used to load a characteristic typical for the device. (77)

Possible status displays for this device command

Status (C00003)		Meaning
	4687106	Device command in process
	4653056	Device command executed successfully
	4653057	General fault
	4692754	The calculation of the inverter characteristic cannot be started since the current controller test mode is active.
	4692755	The calculation of the inverter characteristic cannot be started since the V/f test mode is active.
	4692756	The calculation of the inverter characteristic cannot be started since the pole position identification is active.
	4692757	Calculation of the inverter characteristic has been aborted.
	4692758	Calculation of the inverter characteristic has been interrupted by error.
	4692789	Determined inverter error characteristic exceeds internal limits. • This situation can for instance occur if the motor power is very much lower than the device power. • This error message is only available from software version V5.0 onwards.

Related device commands

- Load Lenze inverter characteristic (77)

4.2.25 Determine motor parameters

The C00002 = "72: Determine motor parameters" device command is used to automatically determine the motor parameters for a third-party motor that are listed in the following table – if they are not known:

Parameter	Information	ASM	SM
C00079	Mutual motor inductance	☑	
C00082	Motor rotor resistance	☑	
C00084	Motor stator resistance	☑	☑
C00085	Motor stator leakage induct.	☑	☑
C00091	Motor - cosine phi	☑	
C00092	Motor magnetising current	☑	

Tip!

Detailed information about the automatic determination of the motor parameters can be found in the chapter "Motor interface" in the subchapter "Determining the motor parameters". (📖 146)

Possible status displays for this device command

Status (C00003)		Meaning
⌛	4752642	Device command in process
✓	4718592	Device command executed successfully
✗	4718593	General fault
	4758290	Motor identification cannot be started since the current controller test mode is active.
	4758291	Motor identification cannot be started since the V/f test mode is active.
	4758292	Motor identification cannot be started because pole position identification is active.
	4758293	Motor identification has been aborted.
	4758294	Motor identification has been aborted by fault.
	4758332	Motor identification aborted due to inconsistent motor parameters. • This error message is only available from software version V7.0 onwards.

4.2.26 Calculate current controller parameters

This function extension is available from software version V5.0 onwards!

The device command C00002 = "77: Calculate current controller parameters" is used to calculate the gain and the reset time of the current controller for a third-party motor.

Precondition: The two motor parameters "stator resistance" (C00084) and "stator leakage inductance" (C00085) either have been parameterised manually on the basis of the manufacturer information before, or have been determined automatically via the device command "Determine motor parameters".

Note!

For a Lenze motor the calculation and the subsequent optimisation of the current controller parameters is not required, as the correct current controller parameters are accepted from the »Engineer« motor catalogue.

The device command is no identification procedure for determining the current controller parameters!

- The calculation is carried out according to the following formulas:

$$\text{Gain} = \frac{\text{Stator leakage inductance}}{340\ \mu\text{s}}$$

$$\text{Reset time} = \frac{\text{Stator leakage inductance}}{\text{Stator resistance}}$$

- After the device command has been executed successfully (see status in C00003), the two calculated values are set in C00075 and C00076. They serve as starting values for a subsequent optimisation of the current controller in the test mode.
- In case of error, codes C00075 and C00076 are not altered.

Tip!

Detailed information on the optimisation of the current controller in the test mode can be found in the chapter "Motor interface" in the subchapter for the respective motor control:

- Servo control (SC) ▶ Optimise current controller (153)
- Sensorless vector control (SLVC) ▶ Optimise current controller (187)
- V/f control (VFCplus) ▶ Optimise current controller (202)

Possible status displays for this device command

Status (C00003)		Meaning
	5080322	Device command in process
	5046272	Device command executed successfully
	5046273	General fault
	5086002	At least one calculated value is beyond the valid setting range.
	5086003	Stator resistance (C00084) too small (zero).

Related device commands

- Determine motor parameters (📖 79)
- Calculate speed controller parameters (📖 82)

4.2.27 Calculate speed controller parameters

This function extension is available from software version V5.0 onwards!

The device command C00002 = "78: Calculate speed controller parameters" is used to calculate the gain, reset time, and rate time of the speed controller.

Precondition: The moments of inertia for the motor (C00273/1) and load (C00273/2) have been parameterised correctly before.

Note!

The device command is no identification procedure for determining the speed controller parameters!

- The calculation is carried out according to the following formulas, taking the actual speed value filter time constant into consideration (C00497):

$$\text{Gain} = \frac{\text{Moment of inertia of motor+load}}{4 \cdot (\text{Actual speed value filter time constant} + 500\ \mu\text{s})} \cdot \frac{2\pi}{60}$$

$$\text{Reset time} = 4^2 \cdot (\text{Actual speed value filter time constant} + 500\ \mu\text{s})$$

$$\text{Rate time} = 0\ \text{ms}$$

- After the device command has been executed successfully (see status in C00003), the calculated values are set in the corresponding codes:
  - C00070: Speed controller gain
  - C00071: Speed controller reset time
  - C00072: Speed controller rate time
- In case of error, these codes are not altered.

Possible status displays for this device command

Status (C00003)	Meaning
5145858	Device command in process
5111808	Device command executed successfully
5111809	General fault
5151540	At least one calculated value is beyond the valid setting range.

Related device commands

- Calculate current controller parameters (📖 80)

4.2.28 CAN on board: Reset node

The C00002 = "91: CAN on board: reset node" device command is used to reinitialise the CANopen system bus interface of the controller ("CAN on board"), which is required, for instance, after the data transfer rate, node address, or identifiers have been changed.

Tip!

For detailed information about the "CAN on board" CANopen system bus interface, please see the "CAN" Communication Manual.

Possible status displays for this device command

Status (C00003)		Meaning
	5997826	Device command in process
	5963776	Device command executed successfully
	5963777	General fault
	6003200	CAN fault
	...	...
	6003455	CAN fault

Related device commands

- CAN on board: Pred.Connect.Set (85)
- CAN on board: Identify node (87)

4.2.29 CAN module: Reset node

The C00002 = "92: CAN module: reset node" device command is used to reinitialise the CANopen interface of a CANopen communication module in module slot MXI1 or MXI2, which is required, for instance, after the data transfer rate, node address, or identifiers have been changed.

Tip!

Detailed information on the CANopen communication module (E94AYCCA) can be found in the "CAN" Communication Manual.

Possible status displays for this device command

Status (C00003)		Meaning
	6063362	Device command in process
	6029312	Device command executed successfully
	6029313	General fault
	6068736	CAN fault
		...
	6068991	CAN fault

Related device commands

- CAN module: Pred.Connect.Set (86)
- CAN module: Identify node (88)

4.2.30 CAN on board: Pred.Connect.Set

The C00002 = "93: CAN on board: Pred.Connect.Set" device command is used to set the basic identifiers for the CANopen system bus interface of the controller ("CAN on board") according to the "Predefined Connection Set" (DS301V402).

Tip!

For detailed information about the "CAN on board" CANopen system bus interface, please see the "CAN" Communication Manual.

Possible status displays for this device command

Status (C00003)		Meaning
	6128898	Device command in process
	6094848	Device command executed successfully
	6094849	General fault

Related device commands

- CAN on board: Reset node (📖 83)
- CAN on board: Identify node (📖 87)

4.2.31 CAN module: Pred.Connect.Set

The C00002 = "94: CAN module: pred.connect.set" device command is used to set the basic identifiers for the CANopen system bus interface of a CANopen communication module in module slot MXI1 or MXI2 according to the "Predefined Connection Set" (DS301V402).

Tip!

Detailed information on the CANopen communication module (E94AYCCA) can be found in the "CAN" Communication Manual.

Possible status displays for this device command

Status (C00003)		Meaning
	6194434	Device command in process
	6160384	Device command executed successfully
	6160385	General fault

Related device commands

- CAN module: Reset node (84)
- CAN module: Identify node (88)

4.2.32 CAN on board: Identify node

The C00002 = "95: CAN on board: identify node" device command is used to determine the nodes connected to the CANopen system bus interface of the controller ("CAN on board").

- The result of the CAN bus scan is displayed in C00393.

Tip!

For detailed information about the "CAN on board" CANopen system bus interface, please see the "CAN" Communication Manual.

Possible status displays for this device command

Status (C00003)	Meaning
6259970	Device command in process
6225920	Device command executed successfully
6225921	General fault

Related device commands

- CAN on board: Reset node (83)
- CAN on board: Pred.Connect.Set (85)

4.2.33 CAN module: Identify node

The C00002 = "96: CAN module: identify node" device command is used to determine the nodes connected to the CANopen system bus interface of a CANopen communication module in module slot MXI1 or MXI2.

▶ The result of the CAN bus scan is displayed in C13393 (for MXI1) or in C14393 (for MXI2).

Tip!

Detailed information on the CANopen communication module (E94AYCCA) can be found in the "CAN" Communication Manual.

Possible status displays for this device command

Status (C00003)		Meaning
	6325506	Device command in process
	6291456	Device command executed successfully
	6291457	General fault

Related device commands

- CAN module: Reset node (📖 84)
- CAN module: Pred.Connect.Set (📖 86)

4.2.34 Ethernet module MXI2 unbind/bind

The C00002 = "101: Unbind/bind Ethernet module: MXI1" device command is used to reinitialise the Ethernet interface of an Ethernet communication module in module slot MXI1, e. g. to accept a newly set IP or gateway address without mains switching.

Tip!

Detailed information on the Ethernet communication module (E94AYCEN) can be found in the "Ethernet" Communication Manual.

Possible status displays for this device command

Status (C00003)		Meaning
	6653186	Device command in process
	6619136	Device command executed successfully
	6619137	General fault

Related device commands

- Ethernet module MXI2 unbind/bind (📖 90)

4.2.35 Ethernet module MXI2 unbind/bind

The C00002 = "102: Unbind/bind Ethernet module: MXI2" device command is used to reinitialise the Ethernet interface of an Ethernet communication module in module slot MXI2, e. g. to accept a newly set IP or gateway address without mains switching.

Tip!

Detailed information on the Ethernet communication module (E94AYCEN) can be found in the "Ethernet" Communication Manual.

Possible status displays for this device command

Status (C00003)		Meaning
	6718722	Device command in process
	6684672	Device command executed successfully
	6684673	General fault

Related device commands

- Ethernet module MXI2 unbind/bind (89)

4.2.36 Activate parameter set 1 ... 4

In addition to the start parameters, up to four further parameter sets can be stored in the memory module for each application. Like this you can for instance define different controller settings for an application, which are then simply activated via device command, if required.

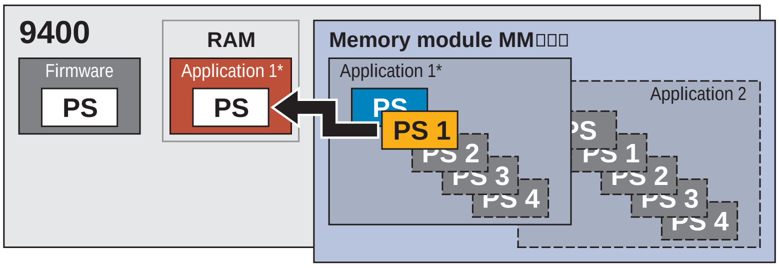
The following device commands can be used to activate the parameter set 1 ... 4 for the active application (if available on the memory module):

C00002 = "201: Activate parameter set 1"

C00002 = "202: Activate parameter set 2"

C00002 = "203: Activate parameter set 3"

C00002 = "204: Activate parameter set 4"



* In this example, application 1 is the active application

[4-20] Example: "Activate parameter set 1" function

- Only possible when the application has stopped and the controller is inhibited.
- All parameter changes of the previously active parameter set carried out since the last saving will get lost!
- These device commands only affect the settings of the operating system, application and module parameters, the active application or a configuration selected with the function block editor remain unchanged.

Possible status displays for these device commands

Status (C00003)					Meaning
	for command 201	for command 202	for command 203	for command 204	
	13206786	13272322	13337858	13403394	Device command in process
	13172736	13238272	13303808	13369344	Device command executed successfully
	13172731	13238273	13303809	13369345	General fault
	13206532	13272068	13337604	13403140	File could not be opened.
	13206557	13272093	13337629	13403165	Fault while reading out of a file.
	13206558	13272094	13337630	13403166	Fault while writing into a file.
	13206559	13272095	13337631	13403167	Invalid file type.
	13206560	13272096	13337632	13403168	Unexpected file end.
	13206562	13272098	13337634	13403170	Checksum error
	13212160	13277696	13343232	13408768	CAN fault
	...	...	...	...	...
	13212415	13277951	13343487	13409023	CAN fault
	13213697	13279233	13344769	13410305	Access to file has been denied since the file is already accessed from another position
	13213701	13279237	13344773	13410309	I/O fault when accessing the file system
	13213708	13279244	13344780	13410316	RAM is full
	13213709	13279245	13344781	13410317	Access authorisation denied
	13213724	13279260	13344796	13410332	No free memory on the memory module

Related device commands

▸ Activate parameter set 1 ... 4 (93)

4.2.37 Activate parameter set 1 ... 4

In addition to the start parameters, up to four further parameter sets can be stored in the memory module for each application. Like this you can for instance define different controller settings for an application, which are then simply activated via device command, if required.

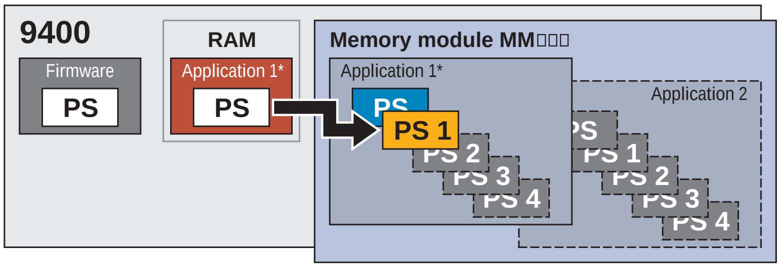
The following device commands are used to archive the current parameter settings of the controller for the active application in the memory module as parameter set 1 ... 4:

C00002 = "301: Archive parameter set 1"

C00002 = "302: Archive parameter set 2"

C00002 = "303: Archive parameter set 3"

C00002 = "304: Archive parameter set 4"



* In this example, application 1 is the active application

[4-21] Example: "Archive parameter set 1" function

▶ Previously archived parameter settings will be overwritten with the current parameter settings!

Possible status displays for these device commands

Status (C00003)					Meaning
	for command 301	for command 302	for command 303	for command 304	
	19760386	19825922	19891458	19956994	Device command in process
	19726336	19791872	19857408	19922944	Device command executed successfully
	19726337	19791873	19857409	19922945	General fault
	19760132	19825668	19891204	19956740	File could not be opened.
	19760157	19825693	19891229	19956765	Fault while reading out of a file.
	19760158	19825694	19891230	19956766	Fault while writing into a file.
	19760160	19825696	19891232	19956768	Unexpected file end.
	19767297	19832833	19898369	19963905	Access to file has been denied since the file is already accessed from another position
	19767301	19832837	19898373	19963909	I/O fault when accessing the file system
	19767308	19832844	19898380	19963916	RAM is full
	19767309	19832845	19898381	19963917	Access authorisation denied
	19767324	19832860	19898396	19963932	No free memory on the memory module

Related device commands

▶ Activate parameter set 1 ... 4 (📖 91)

4.2.38 Load cam data

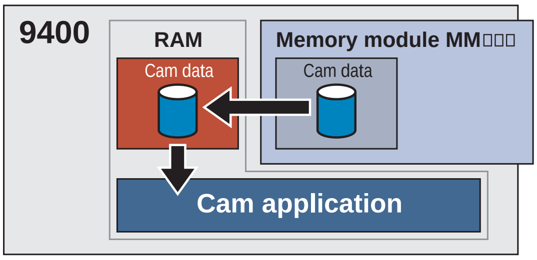
This function extension is available from software version V3.0!

The C00002 = "501: Load cam data" device command serves to reload cam data from the memory module into the controller.

Note!

If you transfer the parameter set or the application from the »Engineer« to the controller, the cam data are also transferred automatically to the controller.

- The new/altered cam data are accepted in the controller according to the online change mode set.
- Thus, normally this device command does not need to be executed manually.



[4-22] "Load cam data" function

- Only possible when the application has stopped and the controller is inhibited.
- If the cam data are provided with an access protection, the user password has to be entered in C02900 first.

Tip!

Detailed information on the online change mode and the access protection can be found in the chapter "Basic drive functions", subchapter "Cam data management". (📖 563)

Sequence

1. The cam data are completely loaded from the memory module into the main memory of the controller.
2. The present cam data in the application unit are converted to the internal unit [increments] and are reorganised.
3. The processed cam data are stored in a separate main memory that can be accessed by the cam application.

Possible status displays for this device command

Status (C00003)		Meaning
	32867586	Device command in process
	32833536	Device command executed successfully
	32833537	General fault
	32875521	No cam data available on the memory module
	32875523	Loading of the cam data failed
	32875525	Checksum error
	32875542	Wrong password entered
	32875545	The cam functionality is deactivated

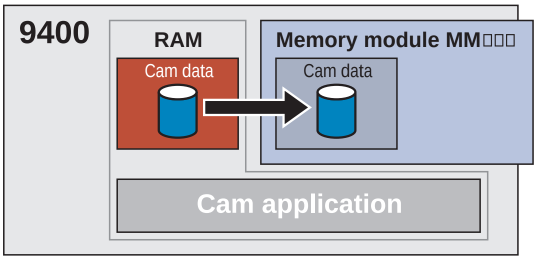
Related device commands

- Save cam data (97)
- Calculate cam data (99)
- Calculate cam data checksum (100)

4.2.39 Save cam data

This function extension is available from software version V3.0!

The C00002 = "502: Save cam data" device command serves to save the cam data available in the main memory of the controller with mains failure protection in the memory module.



[4-23] "Save cam data" function

- This function is executed in the background and is also possible when the controller is enabled and the application is running.
  - However, this function is only executed if valid cam data are available.
  - The cam data can also be saved if previously no cam data have been available on the memory module.
- While the function is executed, no online change and no change of the cam data via parameters can be carried out.

For software versions lower than V4.0 the following applies:

- If the cam data are provided with an access protection, the user password has to be entered in C02900 first.

The following applies from software version V4.0:

- For saving the cam data, you do not need to enter a possibly existing user password (C02900).

Tip!

Detailed information on the access protection can be found in the chapter "Basic drive functions", subchapter "Cam data management". (📖 563)

Possible status displays for this device command

Status (C00003)	Meaning
32933122	Device command in process
32899072	Device command executed successfully
32899073	General fault
32941057	No cam data to be saved are available in the RAM of the controller
32941060	Saving of the cam data failed
32941078	Wrong password entered
32941081	The cam functionality is deactivated

Related device commands

- Load cam data (95)
- Calculate cam data (99)
- Calculate cam data checksum (100)

4.2.40 Calculate cam data

This function extension is available from software version V3.0!

The C00002 = "503: Calculate cam data" device command converts the cam data stored in the main memory of the controller to the internal format and makes them available to the application. This, for instance, is necessary if one or more machine parameters affecting the internal scaling of cam data have been changed.

- The status signal *bNewDataAvailable* of the basic drive function "Cam data management" (**LS_CamInterface** system block) is set to TRUE and the cam data are accepted automatically or manually depending on the online change mode set. After successful data acceptance, the status signal *bNewDataAvailable* is automatically reset to FALSE.
- The user password does not have to be entered in C02900.
- While the function is executed, no online change and no change of the cam data via parameters can be carried out.
- This function is executed in the background and can also be activated when the controller is enabled and the application is running.

Tip!

Detailed information on the cam functionality can be found in the chapter "Basic drive functions", subchapter "Cam data management". (📖 563)

Possible status displays for this device command

Status (C00003)		Meaning
	32998658	Device command in process
	32964608	Device command executed successfully
	32964609	General fault
	33006617	The cam functionality is deactivated

Related device commands

- Load cam data (📖 95)
- Save cam data (📖 97)
- Calculate cam data checksum (📖 100)

4.2.41 Calculate cam data checksum

This function extension is available from software version V3.0!

The C00002 = "504: Calculate cam data checksum" device command is used to recalculate the checksum of the cam data available in the main memory of the controller. This is required if the cam data in the main memory of the controller have been changed via parameters. Afterwards the cam data can be converted to the internal format using the "503: Calculate cam data" device command, or they can be saved with mains failure protection in the memory module using the "502: Save cam data" device command.

- The user password does not have to be entered in C02900.
- This function is executed in the background and can also be activated when the controller is enabled and the application is running.

Tip!

Detailed information on the cam functionality can be found in the chapter "Basic drive functions", subchapter "Cam data management". (📖 563)

Possible status displays for this device command

Status (C00003)		Meaning
⌛	33064194	Device command in process
✔	33030144	Device command executed successfully
✖	33030145	General fault
	33072153	The cam functionality is deactivated

Related device commands

- Load cam data (📖 95)
- Save cam data (📖 97)
- Calculate cam data (📖 99)

4.2.42 Format file system

The C00002 = "1030: Format file system" device command is used to format the file system in the memory module.

Note!

By means of this device command all folders and files in the file system of the memory module are irrevocably deleted!

The application has to be downloaded again with the »Engineer«.

Possible status displays for this device command

Status (C00003)		Meaning
	67536130	Device command in process
	67502080	Device command executed successfully
	67502081	General fault

Related device commands

▸ Restore file system (📖 102)

4.2.43 Restore file system

The C00002 = "1040: Restore file system" device command is used to execute a low level formatting of the file system in the memory module.

Note!

By means of this device command all folders and files in the file system of the memory module and all internal information for the management of the file system are irrevocably deleted!

This device command has no status display in C00003, i.e. the display remains unchanged showing the previous device command status.

Stop!

The low level formatting of the file system by the user is only intended for the exceptional case when the standard formatting of the file system via the C00002 = "1030: Format file system" device command is no longer possible, e.g. due to damaged internal management information.

Related device commands

▸ Format file system (📖 101)

4.2.44 Prepare firmware update

Note!

For Lenze service only!

The C00002 = "10000: Prepare firmware update" device command is used to set the controller to the firmware update mode to update the firmware, if required, using the corresponding software.

▶ Only possible when the application has stopped and the controller is inhibited.

Possible status displays for this device command

Status (C00003)		Meaning
	655394050	Device command in process
	655360000	Device command executed successfully
	655360001	General fault

4.2.45 Restart controller

The C00002 = "11000: Restart controller" device command is used to restart the controller via parameter setting.

- Only possible when the application has stopped and the controller is inhibited.

Possible status displays for this device command

Status (C00003)		Meaning
	720930050	Device command in process
	720896001	General fault

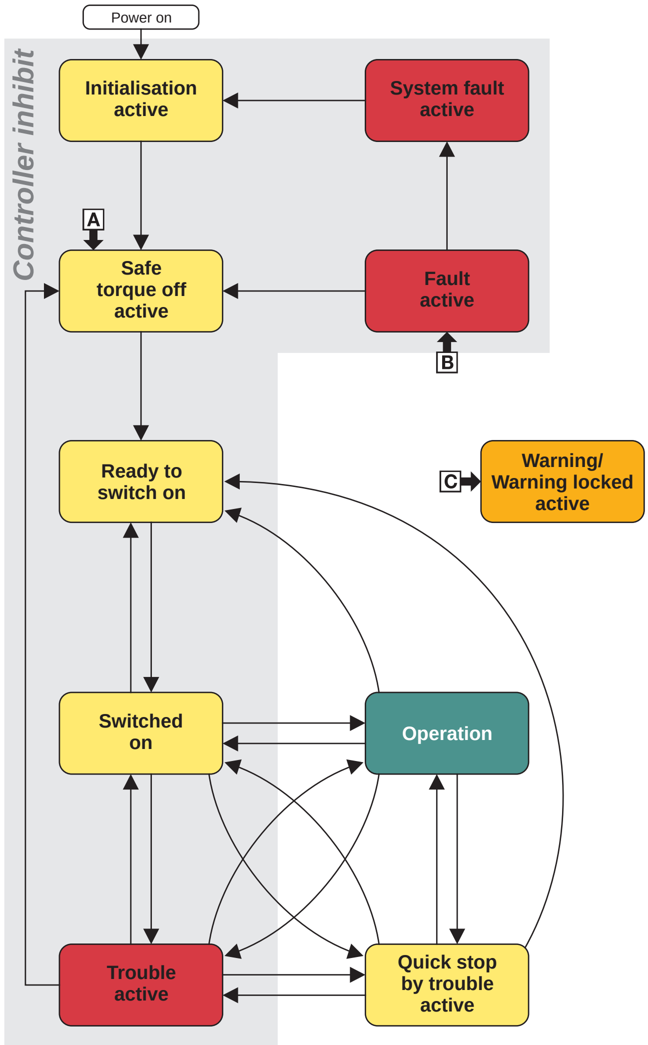
Note!

Due to the restart at the successful execution of the device command, this status is no longer displayed in C00003.

If this device command is used, the message "Undervoltage in the DC bus (0x007b000f)" may appear in the logbook.

4.3 Device states

The state control of the drive is controlled internally via a state machine which can adopt the following "device states":



A, B From all states

C Status "Warning active"/"Warning locked active" can occur simultaneously with other states.

[4-24] Device states of the state machine for the device control

Note!

The device states of the controller must not be confused with the function states of the Basic drive functions. (375)

- In the device state "Operation" the Basic drive functions define the motion control of the drive.

Display parameters for diagnostic purposes

- In C00183 the current device state is shown.
- In C00150 (status word 1) the current device state is shown bit coded via bits 8 ... 11:

Bit 11	Bit 10	Bit 9	Bit 8	Meaning
0	0	0	0	"Initialisation active" state
0	0	0	1	"Device is ready to switch on" state
0	0	1	0	-
0	0	1	1	"Device is switched on" state
0	1	0	0	-
0	1	0	1	-
0	1	1	0	"Operation" state
0	1	1	1	"Warning active" state or "Warning locked active" state The controller is ready to switch on, switched on or operation is enabled, and a warning is present.
1	0	0	0	"Trouble active" state
1	0	0	1	-
1	0	1	0	"Quick stop by trouble active" state
1	0	1	1	"Safe torque off active" state Observe LED of the safety module!
1	1	0	0	"Fault active" state
1	1	0	1	-
1	1	1	0	-
1	1	1	1	-

- C02530 displays the active function state.

LED status displays

The control of the two LEDs "DRIVE READY" and "DRIVE ERROR" in the middle of the controller's front panel depends on the device state. ▶ LED status displays for the device state (📖 612)

Influence of the status signals of the SB LS_DriveInterface by the device state

Device state	Status signals (Outputs of the SB LS_DriveInterface)						
	DI_bReady	DI_bFail Active	DI_bImp Active	DI_bCInh Active	DI_bWarning Active	DI_bReady ToSwitchOn	DI_bOperation Enabled
Initialisation active	FALSE	FALSE	**TRUE**	**TRUE**	FALSE	FALSE	FALSE
Safe torque off active	FALSE	FALSE	**TRUE**	**TRUE**	**TRUE**/FALSE	FALSE	FALSE
Device is ready to switch on	FALSE	FALSE	**TRUE**	**TRUE**	**TRUE**/FALSE	**TRUE**	FALSE
Device is switched on	**TRUE**	FALSE	**TRUE**	**TRUE**	**TRUE**/FALSE	FALSE	FALSE
Operation	**TRUE**	FALSE	FALSE	FALSE	**TRUE**/FALSE	FALSE	**TRUE**
Warning active	**TRUE**/FALSE	**TRUE**/FALSE	**TRUE**/FALSE	**TRUE**/FALSE	**TRUE**	**TRUE**/FALSE	**TRUE**/FALSE
Warning locked active	**TRUE**/FALSE	**TRUE**/FALSE	**TRUE**/FALSE	**TRUE**/FALSE	**TRUE**	**TRUE**/FALSE	**TRUE**/FALSE
Quick stop by trouble active	FALSE	**TRUE**	FALSE	FALSE	**TRUE**/FALSE	FALSE	FALSE
Trouble active	FALSE	FALSE	**TRUE**	**TRUE**	**TRUE**/FALSE	FALSE	FALSE
Fault active	FALSE	**TRUE**	**TRUE**	**TRUE**	**TRUE**/FALSE	FALSE	FALSE
System fault active	FALSE	**TRUE**	**TRUE**	**TRUE**	**TRUE**/FALSE	FALSE	FALSE

▸ Internal interfaces | "LS_DriveInterface" system block (📖 118)

4.3.1 "Initialisation active" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
OFF	OFF	10: Initialisation active

This is the controller's state directly after switching on the supply voltage.

- In this device state the operating system is initialised.
- The application is not processed yet.
- The monitoring is not active yet.
- Communication is not possible yet.
- The controller cannot be parameterised yet and no device commands can be executed.
- When the device initialisation is completed, the device state is automatically changed to "Safe torque off active".

4.3.2 "Safe torque off active" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
	OFF	101: Safe torque off active

This device state becomes active if the controller receives the "Safe torque off" request by the safety module.

- "Drive is torqueless" (0x00750003) is entered in the logbook.
- If no corresponding request by the safety module is available, a change to the subsequent state "Device is ready to switch on" is effected.

Note!

The "Safe torque off active" status is also passed through after an error has been acknowledged (see illustration [4-24]).

4.3.3 "Device is ready to switch on" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
	OFF	"Device is ready to switch on"

This is the device state of the controller directly after the initialisation has been completed and where no DC-bus voltage is applied yet.

- The bus systems are running and the terminals and encoders are evaluated.
- Monitoring is active.
- The controller can be parameterised and device commands can be executed to a limited extent.
- The application is basically executable.
- The functions of the user task can be used.
  - Precondition: The application has started (status display in C02108).
- The basic drive functions cannot be used yet.

Note!

The "Device is ready to switch on" status is not only activated after mains connection but also after reset of "Trouble", "Fault" or "Safe torque off active".

- In order to change from the "Device is ready to switch on" to the "Device is switched on" status when C00142 = "0: inhibited", at least one of the controller inhibit sources must be active.
- When C00142 = "1: Enabled", the "Device is ready to switch on" status directly changes to the "Device is switched on" status.

Danger!

If automatic restart is enabled (C00142 = "1: Enabled"), the drive can restart automatically from the "Trouble" and "Safe torque off" device states when the trouble or requirement for "Safe torque off active" has been eliminated!

▶ Automatic restart after mains connection/trouble... (📖 112)

4.3.4 "Device is switched on" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
	OFF	90: Drive is switched on

This is the device state of the controller if the DC-bus voltage is applied and the controller is still inhibited by the user (controller inhibit).

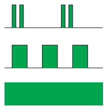
- The bus systems are running and the terminals and encoders are evaluated.
- Monitoring is active.
- The controller can be parameterised and device commands can be executed to a limited extent.
- The application is basically executable.
- The functions of the user task can be used.
  - Precondition: The application has started (status display in C02108).
- The basic drive functions cannot be used yet.
- If the controller is enabled, the motor builds up a torque.

4.3.5 "Operation" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
	OFF	0: Operation

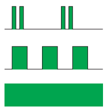
In this device state the motor follows its setpoint according to the basic drive function selected.

4.3.6 "Warning active" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
		1: Operation/warning active

This state can occur at the same time as the device states "Device is ready to switch on", "Device is switched on" and "Operation" if a monitoring function is activated for which the "Warning" error response has been parameterised.

4.3.7 "Warning locked active" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
		2: Operation/warning locked active

This state can occur at the same time as the device states "Device is ready to switch on", "Device is switched on" and "Operation" if a monitoring function is activated for which the "Warning locked" error response has been parameterised.

4.3.8 "Quick stop by trouble active" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
		151: Quick stop by trouble active

This device state becomes active as soon as a monitoring function responds for which the "Quick stop by trouble" error response has been parameterised.

- Irrespective of the setpoint defined, the drive is brought to standstill with a torque within the parameterised deceleration time for quick stop and can be held there.
- The device state can only be exited by acknowledging the error if the error cause has been eliminated.
- It is also possible to skip to the "Operation" state during the error status by setting controller inhibit, as controller inhibit has a higher priority. As long as the error is still available and has not been acknowledged, a change back to the "Quick stop by trouble active" state is effected when the controller is enabled afterwards.

4.3.9 "Trouble active" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
OFF		104: Trouble active

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

- The motor has no torque (is coasting).
- The device state is automatically exited if the error cause is eliminated:
  - "Trouble active" state < 500 ms: Return to the original device state.
  - "Trouble active" state > 500 ms: Return via the device state "Safe torque off active".

4.3.10 "Fault active" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
OFF		102: Fault active

This device state becomes active as soon as a monitoring function responds for which the "Fault" error response has been parameterised.

4.3.11 "System fault active" state

LED DRIVE READY	LED DRIVE ERROR	Display in C00183
OFF		20: System fault active

This device state becomes active if a system fault occurs.

- The device state can only be exited by mains switching.

4.4 Automatic restart after mains connection/trouble...

.../Fault/"safe torque off active"

In C00142, the starting performance of the controller after mains connection and reset of "Trouble", "Fault" or "safe torque off active" can be parameterised.

Note!

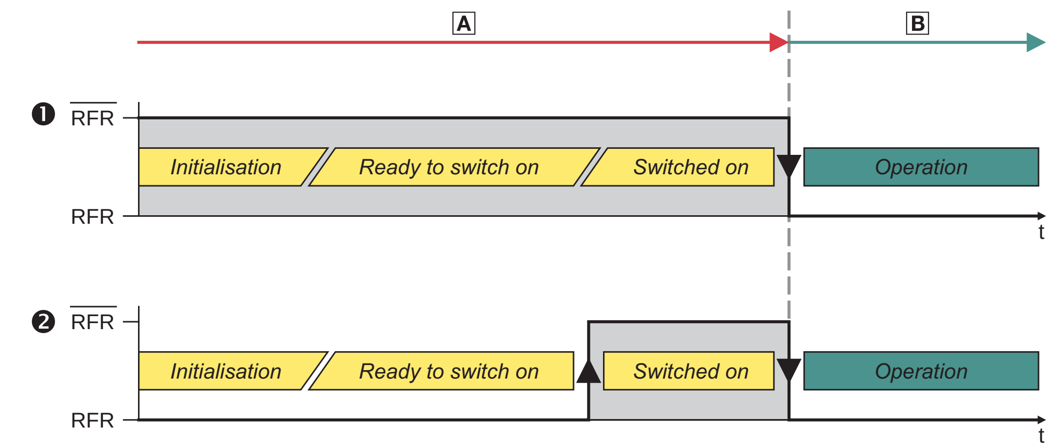
From software version V4.0 the automatic restart is inhibited in the Lenze setting! Set the selection "1: Enabled" in C00142 to obtain the former behaviour.

Danger!

If automatic restart is enabled (C00142 = "1: Enabled"), the drive can restart automatically from the "Trouble" and "Safe torque off" device states when the trouble or requirement for "Safe torque off active" has been eliminated!

Auto-start option 0: Auto restart inhibited after mains connection

Controller inhibit always has to be set if the controller is to change from the "Ready to switch on" state to the "Switched on" state after mains connection or reset of "Trouble", "Fault", or "Safe torque off active". The following change to the "Operation" state is performed when the controller is enabled:



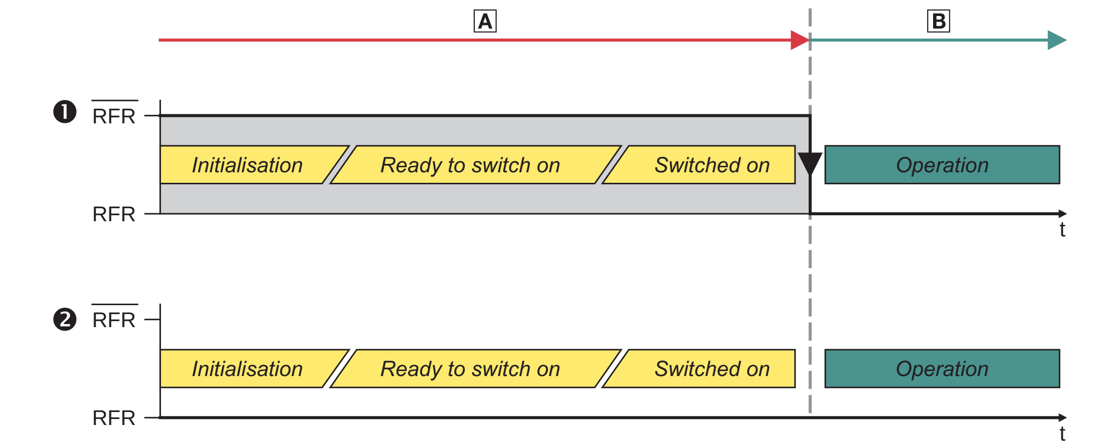
A Controller pulses are inhibited
B Controller pulses are enabled

❶ With controller inhibit after power-on
❷ Without controller inhibit after power-on

[4-25] State change when auto-restart is inhibited (C00142 = "0: Inhibited")

Auto-start option 1: Auto restart enabled after mains connection

The following illustration shows the state changes for the auto-start option 1 and their relationship to controller inhibit:



A Controller pulses are inhibited
B Controller pulses are enabled

❶ With controller inhibit after power-on
❷ Without controller inhibit after power-on

[4-26] State change when auto-restart is enabled (C00142 = "1: Enabled")

4.5 Behaviour after task overflow

Up to software version V5.0 the following applies:

- After a task overflow in the application or user task the "Error" response is effected.

The following applies from software version V5.0:

- In C02111 the error response after a task overflow in the application or user task can be parameterised. The Lenze setting "Error" corresponds to the previous behaviour of the controller with software versions lower than V5.0.

Tip!

For a hoist for instance the "Quick stop by trouble" error response with engagement of the brake can be set, so that the drive is brought to standstill within the shortest time possible.

From »Engineer« version 2.10 onwards, the function block editor can also be used to configure the behaviour of the analog and digital outputs and that of the brake control and the output ports after a task overflow in order to adapt it to the respective application. ▶ Configure exception handling of the outputs (📖 290)

4.6 Device output power

The parameters described in the following subchapters influence the output power of the controller.

4.6.1 Switching frequency

The controller uses a pulse-width modulation to generate its output voltage. The switching frequency is used to change the control factor of the pulse-width modulation.

Automatic switching frequency reduction

In the Lenze setting, the "variable" switching frequency "8 kHz" has been selected in C00018, which means that the controller automatically reduces the switching frequency depending on the setpoint current.

- Depending on the current amount, it is changed down to an assigned switching frequency.
- The switching thresholds are device-dependent (see 9400 hardware manual, chapter "Rated data").
- If a fixed switching frequency is selected in C00018 instead of a variable one, there is no switching frequency changeover, however, (due to the field frequency range 0...5 Hz) it can only be traversed at a low continuous current and low maximum currents (see 9400 hardware manual, chapter "Rated data").

Note!

If parameterisation is carried out offline or if the memory module is exchanged between different 9400 HighLine device types, always check the setting of the switching frequency in C00018 and adapt it, if required, to prevent a parameter error after the parameter set download or module change!

The maximum output frequency of the controller is limited to 1/8 of the switching frequency selected in C00018! (See the following table.)

Switching frequency (C00018):	1 kHz	2 kHz	4 kHz	8 kHz	16 kHz
Maximum output frequency:	125 Hz	250 Hz	500 Hz	1000 Hz	1999 Hz
Motor - number of pole pairs:	Maximum speed [rpm]				
1	7500	15000	30000	60000	120000
2	3750	7500	15000	30000	60000
3	2500	5000	10000	20000	40000
4	1875	3750	7500	15000	30000
5	1500	3000	6000	12000	24000
6	1250	2500	5000	10000	20000

Tip!

If a load profile and a fixed setting of the switching frequency (e. g. 8 kHz fixed) are given, an I x t disconnection due to a high device utilisation can be avoided by selecting the variable setting for the same switching frequency instead.
▸ Monitoring of the device utilisation

Reduced switching losses through switching frequency reduction

The advantage of a switching frequency reduction are the reduced switching losses in the controller, which are monitored via an I x t evaluation.

- A reduced switching frequency enables a greater current-time area at the output than it would be the case with a higher switching frequency. However, depending on the process, you always have to make a compromise between the torque ripple and the output power.

4.6.2 Monitoring of the device utilisation

In C00064 the device utilisation (I x t) is displayed over the last 180 seconds in [%].

- If the value displayed in C00064 exceeds the warning threshold set in C00123, the error message "device utilisation Ixt > C00123" is output and the fault response set in C00604 occurs (default setting: "Warning").
- If the value displayed in C00064 exceeds 100 %, the error message "device utilisation Ixt > 100 %" is output and the "Fault" error response occurs.
  – The fault can only be reset if the value displayed in C00064 is < 95 % again.

4.6.3 Operation with increased continuous power

This function extension is available from software version V3.0!

If required, operation with an increased continuous power for the switching frequencies 1 kHz and 2 kHz can be activated in C01199 for controllers from the device size 8S, if the following requirements are met:

- Controller is of E94AxxE1454 ... E94AxxE6954 type (device size 8S ... 10).
- The maximum current (C00022) is < 150 % of the rated device current.

Stop!

During operation with increased continuous power, the max. permissible ambient temperature is reduced to 40 °C.

The overload current must be reduced. An overload current of 180 % for 10 s is no longer permissible during operation with increased continuous power.

Note!

To activate the operation with increased continuous power, controller inhibit must be set in the controller.

The "activated" setting in C01199 is automatically reset to "deactivated" (without error message) if the previously mentioned requirements are not (no longer) met.

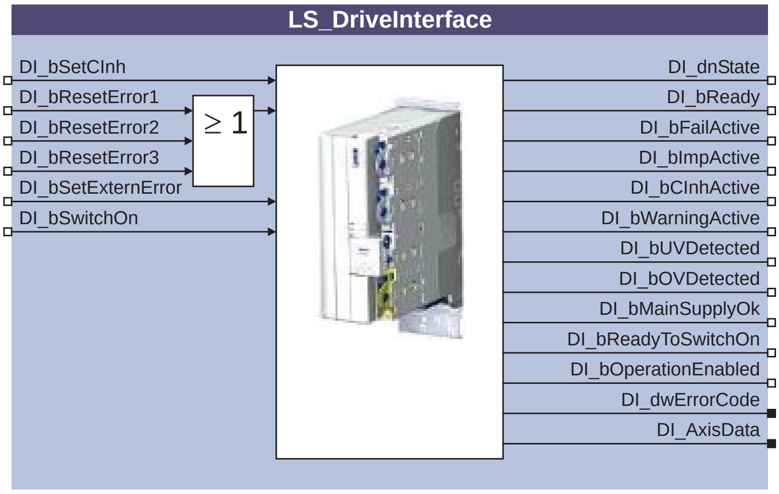
- This is the also the case if the memory module is plugged into a controller of the smaller design 8 (device exchange).

Tip!

The permissible output currents and overload factors for operation with increased continuous power for different device types can be found in the Hardware Manual in the "Rated data" chapter.

4.7 Internal interfaces | "LS_DriveInterface" system block

The **LS_DriveInterface** system block provides the internal interfaces to the drive interface in the function block editor.



Inputs

Identifier DIS code | data type	**Information/possible settings**	
DI_bSetCInh C02549/1 | BOOL	Set/remove controller inhibit • The controller can be inhibited by different sources, e.g. via the digital input RFR or using the device command "Inhibit controller". (📖 67) • The bit code under C00158 shows the source that inhibited the controller.	
	TRUE	Set controller inhibit. • The power output stages in the controller are inhibited and the speed, current, and position controller of the motor control are reset.
	TRUE↘FALSE	Remove controller inhibit. • Please note that the controller will only be enabled if all sources for controller inhibit are reset!
DI_bResetError1 C02548/1 | BOOL DI_bResetError2 C02548/2 | BOOL DI_bResetError3 C02548/3 | BOOL	Error message reset (acknowledgement) • This function resets an active error message if the cause of the error message has been eliminated. • The three inputs are linked via a logic OR gate.	
	TRUE	Reset (acknowledge) error message.
DI_bSetExternError C02548/4 | BOOL	Activation of error message "External error". ▸ Monitoring of external events (📖 122)	
	TRUE	Activate error message with the response selected in C00581.
DI_bSwitchOn C02549/4 | BOOL	Deactivate switch-on inhibit • If the automatic restart is inhibited (C00142 = "0"), the state machine remains in the "Device is ready to switch on" state after mains switching. ▸ "Device is ready to switch on" state (📖 108)	
	FALSE↗TRUE	The switch-on inhibit is deactivated and the controller changes to the device state "Device is switched on".

Outputs

Identifier DIS code | data type	Value/meaning	
DI_dnState C02547 | DINT	Status (bit coded)	
	Status signals of the currently enabled basic function (if available):	
	Bit 0	-
	Bit 1	Basic function is active (signal *bActive*).
	Bit 2	Basic function is completed (signal *bDone*).
	Bit 3	Acceleration/deceleration phase is active (signal *bAccDec*).
	Bit 4	-
	Bit 5	CCW rotation is active (signal *bCcw*).
	Bit 6	-
	Bit 7	Reference known.
	Bit 8	Brake is open.
	Bit 9	Waiting for clutch condition.
	Bit 10	Zero crossing detected or position = "0".
	Bit 11	-
	Bit 12	-
	Bit 13	-
	Bit 14	-
	Bit 15	Fault in active basic function (group signal).
	Status signals of the internal state machine for the basic functions:	
	Bit 16	Torque follower active.
	Bit 17	Speed follower active.
	Bit 18	Position follower active.
	Bit 19	Setpoint follower is active (group signal for bit 16 ...18).
	Bit 20	Positioning active.
	Bit 21	Homing active.
	Bit 22	Manual jog active.
	Bit 23	Brake test is active.
	Bit 24	Drive at standstill.
	Bit 25	Drive is stopped.
	Bit 26	Quick stop active.
	Bit 27	-
	Bit 28	Controller is not ready.
	Bit 29	Initialisation
	Bit 30	State "Fault active" (signal *DI_bFailActive*).
	Bit 31	State machine is not ready to receive setpoints. (Group signal for bit 28 ... 30)
DI_bReady C02549/6 | BOOL	Status signal "Controller is ready for operation"	
	TRUE	The controller is ready for operation.
DI_bFailActive C02549/7 | BOOL	Status signal "Error active - acknowledgement required"	
	TRUE	Monitoring with the "Fault" or "Quick stop by trouble" error response has responded and the controller is in the device state "Fault active" or "Quick stop by trouble active". For exiting the device state the fault has to be acknowledged, e. g. via the input *DI_bErrorReset1...3*.

Identifier DIS code | data type	Value/meaning	
DI_bImpActive C02549/8 | BOOL	Status signal "Pulse inhibit set"	
	TRUE	The power output stages are switched to high impedance.
DI_bCInhActive C02549/9 | BOOL	Status signal "Controller inhibit active"	
	TRUE	The controller inhibit is active.
DI_bWarningActive C02549/10 | BOOL	Status signal "Warning active"	
	TRUE	A warning is active in the controller.
DI_bUVDetected C02549/11 | BOOL	Status signal "Undervoltage detected" • The threshold for this monitoring function depends on the setting under C00173.	
	TRUE	Undervoltage detected in DC bus.
DI_bOVDetected C02549/12 | BOOL	Status signal "Overvoltage detected" • The threshold for this monitoring function depends on the setting under C00173.	
	TRUE	Overvoltage detected in DC bus.
DI_bMainSupplyOk C02549/13 | BOOL	Status signal "Mains voltage is applied"	
	TRUE	A voltage is applied to the mains voltage inputs L1, L2 and L3.
DI_bReadyToSwitchOn C02549/14 | BOOL	Status signal "Controller ready to switch on"	
	TRUE	The controller has completed the initialisation and is in the "Device is ready to switch on" device state.
DI_bOperationEnabled C02549/15 | BOOL	Status signal "Operation is enabled"	
	TRUE	The controller is in the "Operation" device state and the motor follows its setpoint according to the selected basic drive function or is at standstill due to stop or quick stop.
DI_dwErrorCode DWORD	Error number of the current error message ▸ Error messages of the operating system (📖 623)	
DI_AxisData	Data structure, which contains all required machine constants.	

4.7.1 Status signals

The following representation shows which status signals of the drive interface are set to TRUE in different typical cases:

Case 1:
Application has been transmitted to the controller.
No mains voltage available (LU fault).
Controller is inhibited (via RFR terminal).

Case 2:
Mains voltage has been connected.

Case 3:
Controller inhibit has been deactivated.

Case 4:
Fault active.

Case 5:
Quick stop by trouble is active

Status	Case 1	Case 2	Case 3	Case 4	Case 5	Status signal (output)
Ready for operation		•	•			*DI_bReady*
Fault active				•	•	*DI_bFailActive*
Pulse inhibit active	•	•		•		*DI_bImpActive*
Controller inhibit active	•	•		•		*DI_bCInhActive*
Warning active						*DI_bWarningActive*
Undervoltage detected	•					*DI_bUVDetected*
Overvoltage detected						*DI_bOVDetected*
Mains supply is ok		•	•	•	•	*DI_bMainSupplyOk*
Ready to switch on						*DI_bReadyToSwitchOn*
Operation enabled			•			*DI_bOperationEnabled*

4.7.2 Monitoring of external events

Use the input *DI_bSetExternError* of the LS_DriveInterface system block to monitor external events by means of corresponding logic operations and activate the error message "External error" in the controller.

Parameterising a response to an external error

The controller response to the error message "External error" can be selected under C00581.

Activation of error message "External error".

The error message "External error" is activated by setting the input *DI_bSetExternError* to TRUE.

- After this, the error number for the error message "External error" "0x20750000" (when "Fault" has been selected as response) will be stored in the internal fault memory (C00168).

Reset of error message

The error message "External error" and other active error messages are reset by setting the input *DI_bResetError* to TRUE.

- If the input *DI_bSetExternError* is still set to TRUE, the reset will not be carried out.
- Error messages can only be reset if the cause of the error has been eliminated.

5 Motor interface

This chapter provides you with information on initial commissioning of the motor and the parameterisation of the internal motor control of the controller.

Note!

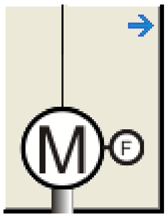
The motor interface contains all control functions that are not provided by other basic drive functions.

To select application-specific setpoints, the motor interface can be extended by appropriate interfaces using the basic functions "Speed follower", "Torque follower" and "Position follower".

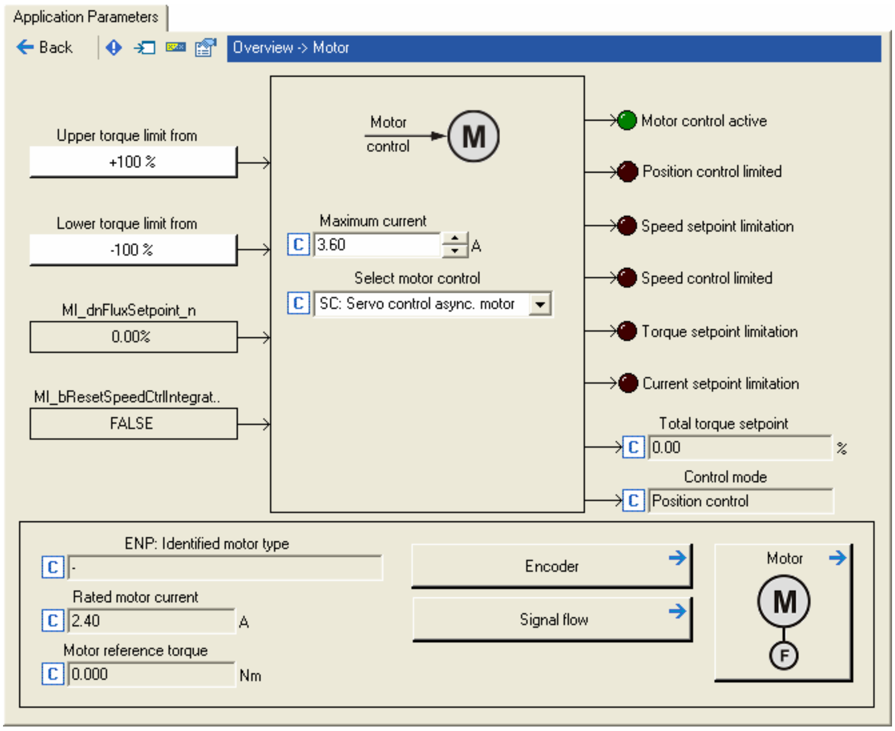
The application-specific conditioning of the encoder signals is executed with the basic function "Encoder evaluation".

How to get to the dialog for setting the motor interface parameters:

1. Go to the *Project view* of the »Engineer« and select the 9400 HighLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
3. Click the following button of the *Overview* dialog level:



Parameterisation dialog in the »Engineer«



- The white buttons indicate the configuration of the motor interface inputs. ▸ Internal interfaces | "LS_MotorInterface" system block (📖 236)
  - The configuration is predefined by the technology application selected (in this example "Actuating drive – speed"). If required, this configuration can be changed by clicking the corresponding buttons.
- If you click a button marked with the ➔ symbol, you go one level deeper in the corresponding parameterisation dialog.

5.1 General information

5.1.1 Reading out motor data from the controller

If the Lenze motor connected to the controller has an electronic nameplate (ENP), the motor does not need to be selected in the »Engineer« motor catalogue.

- With the first switch-on of the controller all motor data are automatically read out from the electronic nameplate of the motor and at first are saved temporarily within the controller.
- For a permanent acceptance of the motor data, the parameter set must be saved (C00002 = "11: Save start parameters").
- If there is an online connection between the »Engineer« and the controller, the motor data can be accepted from the controller to the »Engineer« project.

How to read out the motor data from the controller:

1. Establish an online connection between »Engineer« and controller.
2. Select the **Application parameters** tab and change to the *Overview* → *Motor* → *Motor* dialog level.
3. Click on the **From Drive** button.
   - Then the motor data are read out of the controller and directly written into the corresponding codes of the »Engineer« project.

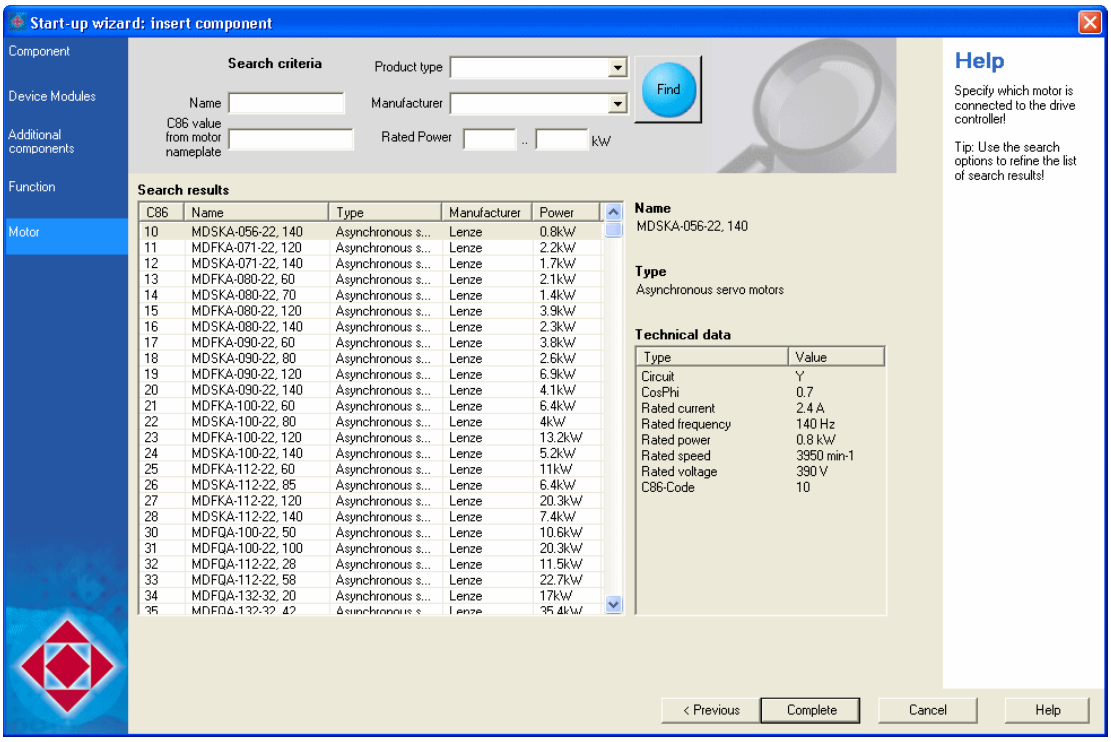
Display parameters for electronic nameplate (ENP)

Parameter	Information	Lenze setting Value	Unit
C00186	ENP: Identified motor type	-	
C00187	ENP: Identified serial number	-	
C00188	ENP: Status	-	

5.1.2 Selecting motor in the »Engineer« motor catalogue

If the Lenze motor does not have an electronic nameplate (ENP) or if a motor of a third-party manufacturer is used, select the motor in the »Engineer« via the motor catalogue and transfer the motor data to the controller.

- 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.

Tip!

If you use a motor of a third-party manufacturer, you can first select a suitable motor regarding the rated data for current, voltage and speed and then adapt the motor data exactly to the existing motor.

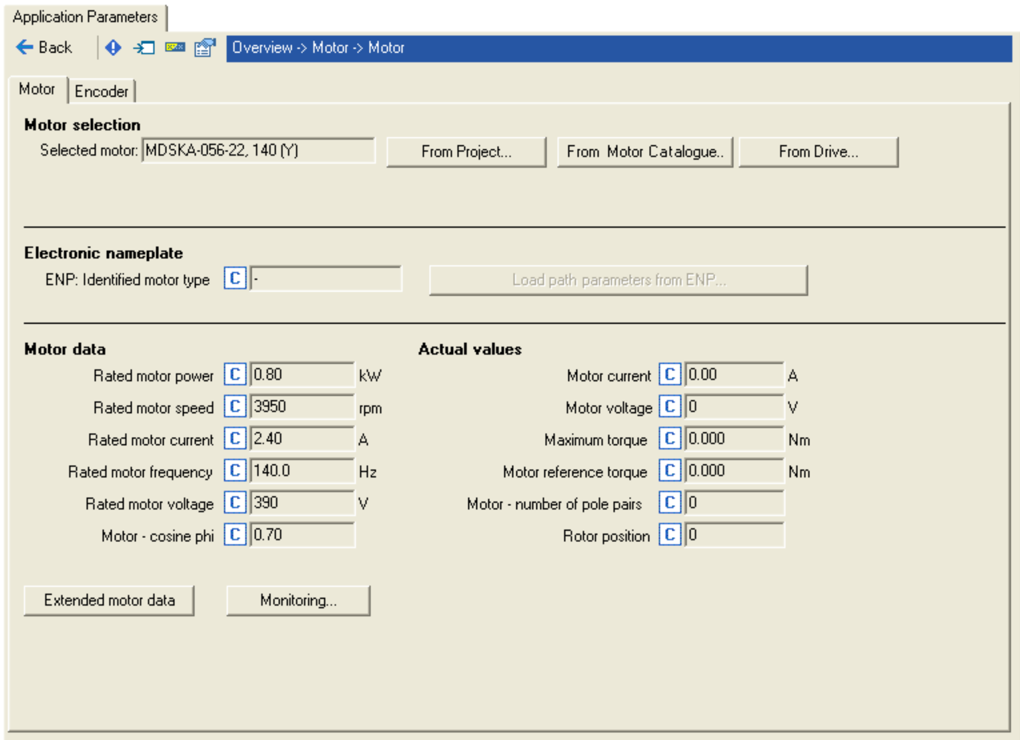
▸ Displaying/editing motor data in the »Engineer« (📖 127)

5.1.3 Displaying/editing motor data in the »Engineer«

The term "Motor data" combines all parameters that only depend on the motor. They solely characterise the electrical behaviour of the machine.

- The motor data do not depend on the application in which the controller and motor are used.
- The motor data are, if available in the »Engineer« via electronic nameplate or motor catalogue, accepted by the controller without confirmation prompt.

In the »Engineer« the motor data are shown on the **Application parameters** tab in the dialog level *Overview→Motor→Motor*:



- If you use a motor of a third-party manufacturer, the displayed motor data can be adapted exactly to the existing motor by clicking the **From Project** button and then selecting the "Own motor settings" entry in the **Motor selection** dialog box.
- Click **From Motor Catalogue** to open the motor catalogue and select a different motor.
  Selecting motor in the »Engineer« motor catalogue (126)
- If an online connection has been established, the motor data set in the controller can be accepted in the »Engineer« via the button **From Drive**. Reading out motor data from the controller (125)

Overview of motor data

Parameter	Information	Lenze setting * Value	Unit
C00052	Motor voltage	-	V
C00054	Motor current	-	A
C00057/1	Maximum torque	-	Nm
C00057/2	Motor reference torque	-	Nm
C00059	Motor - number of pole pairs	-	
C00060	Rotor position	-	
C00079	Mutual motor inductance	-	mH
C00081	Rated motor power		kW
C00082	Motor rotor resistance	-	Ohm
C00083	Motor - rotor time constant	-	ms
C00084	Motor stator resistance		Ohm
C00085	Motor stator leakage induct.		mH
C00087	Rated motor speed		rpm
C00088	Rated motor current		A
C00089	Rated motor frequency		Hz
C00090	Rated motor voltage		V
C00091	Motor - cosine phi		
C00092	Motor magnetising current	-	A
C00128/1	Therm. time constant coil		min
C00128/2	Therm. time constant plates		min
C00273/1	Motor moment of inertia		kg cm²
C01190	Motor thermal sensor		
C01191/1	Spec. characteristic: Temperature		°C
C01191/2	Spec. characteristic: Temperature		°C
C01192/1	Spec. characteristic: resistance		Ohm
C01192/2	Spec. characteristic: resistance		Ohm

Highlighted in grey = display parameter
* depending on the selected motor type

Note!

If the motor has been selected via the »Engineer« motor catalogue, or if the motor data have been adapted offline in the »Engineer«, all motor data have to be transferred to the controller afterwards when an online connection has been established and have to be saved in the memory module with mains failure protection (device command C00002 = "11: Save start parameters").

5.2 Select motor control

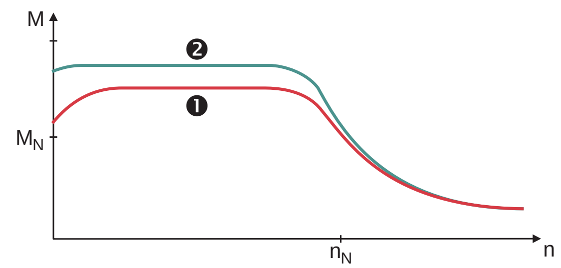
In C00006 the motor control is selected; the default is the servo control for synchronous motors.

Function extension from software version V3.0:

From software version V3.0, alternatively to the servo control also the V/f control and the sensorless vector control are provided as control types in C00006:

Open/closed loop control		Detailed information
1	**SC: Servo control for synchronous motor**	Servo control (SC) (151)
2	SC: Servo control for asynchronous motor	
4	SLVC: Sensorless vector control	Sensorless vector control (SLVC) (172)
6	VFCplus: V/f control open loop	V/f control (VFCplus) (190)
7	VFCplus: V/f control closed loop	V/f control (VFCplus) (206)

- The V/f control is the classic operating mode for standard applications.
- Compared to the V/f control, improved drive characteristics can be achieved with sensorless vector control by:
  - a higher torque across the entire speed range
  - a higher speed accuracy and a higher concentricity factor
  - Higher efficiency



❶ V/f control
❷ Sensorless vector control

[5-1] Comparison of V/f control and sensorless vector control

Note!

Sensorless vector control (SLVC) is only approved for powers up to 55 kW and horizontal applications (no hoists or lifting equipment)!

For the V/f control and sensorless vector control the following table helps with the selection of the correct control type:

	Selection of the motor control in C00006			
	Motor cable shielded ≤ 50 m unshielded ≤ 100 m		**Motor cable** shielded > 50 m unshielded > 100 m	
	recommended	optional	recommended	optional
Single drives				
With motor filter	6 / 7	-	6 / 7	-
With constant load	4	6 / 7	6 / 7	-
With extremely alternating loads	4	6 / 7	6 / 7	-
With high starting duty	4	6 / 7	6 / 7	-
Positioning and infeed drives	4	6 / 7	6 / 7	-
Winders/unwinders dancer	6 / 7	-	6 / 7	-
Pump and fan drives *	6 / 7	-	6 / 7	-
Three-phase reluctance motors	6 / 7	-	6 / 7	-
Three-phase sliding rotor motors	6 / 7	-	6 / 7	-
Three-phase AC motors with firmly assigned voltage/frequency characteristic	6 / 7	-	6 / 7	-
Vertical drive/hoist (up to 55 kW)	6 / 7 (with VCC**)	-	6 / 7 (with VCC**)	-
* For this application, we recommend a square-law voltage characteristic (C00950 = "1") ** VCC = voltage vector control				
Group drives Depending on the resulting motor cable length: $I_{res} = \sqrt{i} \cdot (I_1 + I_2 + ... + I_i)$				
Identical motors and loads	4	6 / 7	6 / 7	-
Different motors and/or alternating loads	6 / 7	-	6 / 7	-

Note!

For operation with motor encoder, we recommend to use the servo control!

For operation with motor filter, always use the V/f control!

5.3 Adjusting motor and controller to each other

This "initial commissioning" of the motor is required if no motor data suitable for the application is available yet in the memory module of the controller and in the »Engineer« project .

- The following step-by-step instructions can be used as a "check list" to correctly adjust the motor and controller to each other.
- Detailed information on the individual steps can be found in the following subchapters.

	Worksteps	Motor control* SC	SLVC	VFC plus
1.	Accepting/adapting plant parameters. (📖 132)	●	●	●
2.	Parameterise motor encoder. (📖 134) • Only required for the control types with speed feedback (servo control and V/f control).	●		(●)
3.	Pole position identification. (📖 136) • Only required: – For servo control with synchronous motor of a third-party manufacturer. – For servo control with synchronous motor and use of incremental encoders (TTL or sin/cos encoders as well as multi-pole-pair resolvers). – After changes of the motor feedback system, e.g. encoder exchange.	(●)		
4.	Optimising the switching performance of the inverter. (📖 143) • Only required for servo control if the motor parameters are to be defined by a motor from a third-party manufacturer! • Always required for sensorless vector control and open loop V/f control! – An optimum drive performance can only be achieved with the sensorless operating modes if the voltage errors in the inverter are compensated as exactly as possible.	(●)	●	●
5.	Determining the motor parameters. (📖 146) • Only required for servo control if the motor parameters are to be defined by a motor from a third-party manufacturer! • Always required for sensorless vector control! – An optimum drive performance can only be achieved with the sensorless vector control if the motor parameters correspond to the real motor as exactly as possible.	(●)	●	

* SC = servo control SLVC = sensorless vector control VFCplus = V/f control

5.3.1 Accepting/adapting plant parameters

The "plant parameters" summarise all parameters which result from the combination of motor and load. These characterise the transfer behaviour of the entire controlled system including the required monitoring modes.

- The plant parameters depend on the application in which the controller and motor are used.
- If a Lenze motor is selected in the »Engineer«, the plant parameters for no-load operation will be suggested for this motor.

Overview of plant parameters

Parameter	Information	Lenze setting		Motor control*		
		Value	Unit	SC	SLVC	VFC plus
C00011	Motor reference speed	3000	rpm	●	●	●
C00022	Maximum current	0.00	A	●	●	●
C00070	Speed controller gain	0.500	Nm/rpm	●	●	●[1]
C00071	Speed controller reset time	24.0	ms	●	●	●[1]
C00072	Speed controller rate time	0.00	ms	●		
C00497	Speed act. val. time const.	2.0	ms	●	●	●
C00596	Threshold max. speed reached	6500	rpm	●	●	●

* SC = servo control SLVC = sensorless vector control VFCplus = V/f control open loop [1] Only for V/f control closed loop

Note!

If plant data have been adapted offline in the »Engineer«, all plant data have to be transferred to the control afterwards when an online connection has been established and have to be saved in the memory module with mains failure protection (device command C00002 = "11: Save start parameters").

Motor reference speed

In C00011 the reference speed of the motor must be set.

Note!

From the perspective of the application it has to be ensured that a maximum of 100 % of the reference speed set in C00011 is requested as speed setpoint.

When using MCS motors, please observe the following:

The controller with software version V01.xx does not support a field weakening control for synchronous motors, so that for this version the operation of MCS motors at the voltage limit may present an undefined behaviour.

Therefore, it should be detected whether the motor used exceeds the voltage limit within the desired operating range up to the maximum current/reference speed. If so, the reference speed must be reduced to a value permissible with regard to voltage.

Maximum current

In C00022 the required maximum current must be set.

- To avoid that the motor starts unintentionally without adjusting the plant data, the maximum current in the Lenze setting is set to "0 A" in C00022.

Ultimate motor current I_{ULT}

C00620 serves to check the set ultimate motor current I_{ULT}.

Note!

When you select a Lenze motor from the catalogue and transfer the plant parameters of the motor to the controller, the setting in C00620 is automatically adjusted to the selected motor.

The ultimate motor current I_{ULT} is a limit value to protect the motor from destruction or influence of the rated data.

- This limit value must not be travelled cyclically during the drive process.
- The maximum current parameterisable in C00022 should have a sufficient distance from this limit value.
- If the instantaneous value of the motor current exceeds the limit value set in C00620, the response set in C00619 is executed for motor protection (Lenze setting: Fault).

Maximum motor speed

Adapt the maximum motor speed in C00596 and select the error response required when this speed limit has been reached in C00607.

5.3.2 Parameterise motor encoder

Note!

Only required for servo control and closed loop V/f control!

Tip!

Detailed information on the encoder evaluation and on the use of a separate position encoder can be found in the following main chapter "Encoder evaluation". (📖 241)

- The motor encoder can be parameterised on the **Application parameters** tab of the »Engineer« in the *Overview* → *Motor* → *Encoder* dialog level.
- The following table shows the required settings for different encoder types:

Encoder type: Motor type:	**Resolver** Tamagawa MCS MCA MDxKS MDXMA	**CDD50** MCA	**ITD21** MDFQA LMR	**ITD22** MDFQA LMR	**SEK... SEL...**	**SKS... SKM...**	**SCS70 SCM70** MDxKS	**SRS50 SRM50** MCS MCA	**ECN1313 EQN1325** MCS MCA	**EQI1329** MCS MCA
C00495 Motor encoder selection	0 Resolver	1 Encoder								
C00080 Resolver - number of pole pairs	1	-	-	-	-	-	-	-	-	-
C00422 Encoder type	-	0 Incremental encoder (TTL signal)		1 Sin/cos encoder	2 Absolute value encoder (Hiperface)				3 Absolute value encoder (EnDat)	
C00420 Encoder - number of increments	-	2048			16	128	512	1024	2048	32
C00421 Encoder voltage	-	5 V			8 V				5 V	

Danger!

If the encoder/resolver is used as motor encoder:
In case of error, safe operation of the motor is no longer guaranteed!

When servo control is used:

- For the (open circuit) monitoring of the encoder/resolver for reasons of safety always the "Fault" response (Lenze setting) should be set!

When V/f control is used:

- For this type of motor control, the drive basically is to coast down after an encoder failure and may not stop, therefore the "Warning" response is to be set for the (open circuit) monitoring in this case!

Short overview: Parameters for setting the response to (open circuit) monitoring

Parameter	Information	Lenze setting
C00580	Resp. to encoder open circuit	Fault
C00586	Resp. to resolver open circuit	Fault
C00601	Resp. to encoder comm. error	Fault

5.3.3 Pole position identification

Note!

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

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

- For Lenze motors with absolute value encoder or resolver, the pole position is already set correctly in C00058/1...3.
- When incremental encoders (TTL or sin/cos encoders) are used, a pole position identification (PPI) is always required after mains switching, even with Lenze motors.
- The controller can also evaluate multi-pole-pair resolvers.
  - When the number of motor pole pairs is an integer multiple of the number of pole pairs of the resolver, a pole position identification must only be executed once.
  - When the number of motor pole pairs is no integer multiple of the number of pole pairs of the resolver, a pole position identification must be executed after every mains switching.
- The device commands "Identify pole position (360°)" and "Identify pole position (min. motion)" serve to determine the pole position for the motor encoder currently activated in C00495 (see the following instructions).

Danger!

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

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

Stop!

Check the correct parameterisation of the max. motor current monitoring (C00619 and C00620) before carrying out the pole position identification to prevent the motor from being permanently damaged.

Note!

From software version V4.0 onwards:

If the pole position identification is aborted, the response parameterised in C00640 is activated (Lenze setting: "Fault").

- Pay attention to this changed behaviour in the Lenze setting when updating the firmware of existing systems!
- If this behaviour is not wanted, deactivate the monitoring by selecting "0: No response" in C00640.

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

- In the Lenze setting of the parameters, the pole position identification remains the same as in software versions < V4.0.

▶ Adjustment of the pole position identification (📖 140)

How to execute the pole position identification:

1. If the controller is enabled, inhibit the controller, e. g. with the device command C00002 = "41: Inhibit controller".
2. Execute device command C00002 = "51: Identify pole position (360°)" or

   device command C00002 = "52: Identify pole position (min. motion)".

   The procedure starts with controller enable, if
   - a synchronous machine is selected,
   - no other identification is active,
   - no error has occurred, and
   - no test mode is activated.

   If one of the above conditions is not met, the procedure is cancelled and the corresponding device command status is indicated under C00003.

   Note:
   By means of controller inhibit, the procedure started can be cancelled anytime, if required, without carrying out a change in C00058.

   For detailed information about the corresponding procedure, please see the following sections:

Tip!

For controller enable all sources for controller inhibit must be reset. In C00158 the sources for controller inhibit are displayed in a bit-coded manner.

The status of the device command activated under C00002 is indicated under C00003.

Procedure for "pole position identification 360°"

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

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

- The rotor is aligned through the current flow. This is absolutely necessary for the procedure.
- To ensure that the torque-neutral axis is not accidentally energised and the rotor stops, a 45° current vector is (electrically) generated for a short instant and then (electrically) switched back to 0° (≡ phase U).
  - Then a DC current of the above-mentioned value could be measured in this motor phase.

The next steps of the procedure depend on the feedback system used:

- If an absolute value encoder with Hiperface or EnDat protocol is used, the encoder position is set to zero and the procedure is cancelled.
- If a resolver or an optical encoder without absolute track is used, the difference between the preselected current angle and the mechanical rotor angle is determined. After this, the current vector is (electrically) turned by another 22.5° and the difference between current angle and rotor angle is determined once again.
  - The procedure is repeated 16 times. This corresponds to one electrical revolution. The machine rotates by 360° (mech.)/pole pair number.
  - Take the average value of the 16 measurements to compensate for asymmetries.

Procedure for "pole position identification with minimal movement"

If all conditions are met, the motor current is increased step by step to the smaller of the following two values:

$$25\,\% \cdot \sqrt{2} \cdot \text{Rated device current}$$
or
$$25\,\% \cdot \sqrt{2} \cdot \text{Rated motor current}$$

- By the current flow the rotor aligns itself, which, however, is compensated by a position control.
- If the rotor moves electrically by more than 20°, a fault message is output and the value measured is rejected. This may occur in the case of motors with a noticeable detent torque.
- In order to detect a non-permissible blocking of the machine, a positive and negative test angle (± 20°) relative to the current position are defined after the identification. The machine must align itself to these two test angles within a tolerance of 25 %.

Note!

In this procedure it is not written back into an optical absolute value encoder and all feedback systems are treated the same way.

Unlike in the "pole position identification 360°" procedure where, when an optical absolute value encoder is used, a "0" is entered into the encoder and into C00058/2, for this procedure nothing needs to be entered into the encoder and the identification result is entered into C00058/2.

After successful completion...

...the controller is inhibited automatically and the pole position determined for the activated feedback system is set in the corresponding subcode of C00058.

- For a permanent acceptance of the identified pole position, the parameter set must be saved (C00002 = "11: Save start parameters").
- The next controller inhibit and subsequent controller enable serve to cancel the controller inhibit automatically set by the procedure (e.g. by first executing the device command C00002 = "41: Inhibit controller" and then executing the device command C00002 = "42: Enable controller").

If an error occurs

If an error occurs during the procedure or the pulse inhibit gets active (e.g. due to short-time undervoltage), the procedure is terminated with controller inhibit without making a change in C00058.

If the machine was braked or blocked during the procedure, this will be recognised at the end of the measurement and no change is made in C00058.

From software version V4.0 onwards, the response parameterised in C00640 (Lenze setting: "Fault") is triggered and the error message "Pole position identification cancelled" is entered in the logbook of the controller if the pole position identification process is aborted.

Tip!

From software version V7.0 onwards, the pole position identification is additionally available as a basic function in the form of the LS_PolePositionIdentification system block.

Basic drive functions: ▸ Pole position identification (📖 581)

5.3.3.1 Adjustment of the pole position identification

This function extension is available from software version V4.0!

The two procedures for Pole position identification (PPI) described in the previous sections can be adjusted to the respective machine and the prevailing moments of inertia by means of the parameters described below.

In the Lenze setting of the parameters, the pole position identification remains the same as in software versions < V4.0.

Note!

The two procedures for the pole position identification should give the same results. But, due to e.g. friction, bearing forces and a trapezoidal field pattern, the results may differ. A proportional increase of the current amplitude in C00641 or C00646 can counteract this deviation.

Parameters for the pole position identification 360°

Parameter	Information	Lenze setting	
		Value	Unit
C00641	PolePosId 360° current amp.	100	%
C00642	PolePosId 360° ramp time	100	%
C00643	PolePosId 360° travel dir.	Clockwise rotating field	
C00644	PolePosId 360° fault tol.	0	°

- The current amplitude can be adjusted proportionally in C00641.
  - For large machines and high mass inertia values or for linear direct drives, the current amplitude usually has to be increased.
  - The Lenze setting "100 %" corresponds to the smaller of the two following values:

$\sqrt{2}$ · Rated device current
or
$\sqrt{2}$ · Rated motor current

Stop!

If there is no temperature monitoring in the motor and/or the I^2xt motor monitoring and the maximum current monitoring are not parameterised correctly, the motor may be permanently damaged when the current amplitude is set too high (e.g. to the maximum value)!

▸ Motor monitoring (I^2xt) (📖 224)
▸ Maximum current monitoring (📖 235)

Note!

If the current amplitude is set to 100 % in C00641 >, the device utilisation (Ixt) monitoring and/or one of the motor monitoring functions may respond and cause the abort of the pole position identification.

- The ramp time can be adjusted proportionally in C00642.
  - For large machines and high mass inertia values, the ramp time usually has to be increased.
  - For small machines, a reduction of the ramp time can speed up the pole position identification process.
- In some situations it may be helpful to reverse the travel direction (C00643) for the pole position identification (e.g. for linear motor at the end stop).
- The "pole position identification 360°" procedure comprises a plausibility check. If the rotor position determined via the encoder system does not correspond to the controlled output position:
  - the pole position identification procedure is aborted.
  - the response parameterised in C00640 (Lenze setting: "Fault") is activated.
  - the error message "Pole position identification cancelled" is entered into the logbook of the controller.
- The preset fault tolerance for the plausibility check can be changed via C00644.

Parameters for the pole position identification with minimal movement

Parameter	Information	Lenze setting	
		Value	Unit
C00646	PolePosId min.mov. cur. amp.	100	%
C00647	PolePosId min.mov. cur.rise rate	100	%
C00648	PolePosId min.mov. gain Vp	0	
C00649	PolePosId min.mov. reset time Tn	62.5	ms
C00650	PolePosId min.mov. max.perm.mov.	20	°

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

$$25\ \% \cdot \sqrt{2} \cdot \text{Rated device current}$$

or

$$25\ \% \cdot \sqrt{2} \cdot \text{Rated motor current}$$

Stop!

If there is no temperature monitoring in the motor and/or the I^2xt motor monitoring and the maximum current monitoring are not parameterised correctly, the motor may be permanently damaged when the current amplitude is set too high (e.g. to the maximum value)!

- Motor monitoring (I^2xt) (📖 224)
- Maximum current monitoring (📖 235)

Note!

If the current amplitude is set to 400 % in C00646 >, the device utilisation (Ixt) monitoring and/or one of the motor monitoring functions may respond and cause the abort of the pole position identification.

- The rate of the current rise for the pole position identification can be adjusted proportionally in C00647. The Lenze setting "100 %" corresponds to the fixed rise rate setting of the software versions < V4.0.
- The P component of the PI controller for the pole position identification can be adjusted in C00648. With the Lenze setting "0", the PI controller continues to work as an I controller (as in the previous software versions).
- The I component of the PI controller for the pole position identification can be adjusted in C00649. Please observe the following notes:
  - The variable *Position.dnActualMotorPos* can be used to monitor the deviation of the position from the start position with the Oscilloscope function in the »Engineer«.
  - In order to be able to compensate a position deviation faster, first the reset time in C00649 should be reduced. If this does not result in the desired behaviour, the proportional gain can be increased in C00648.
  - Ensure that the position control does not get unstable. We therefore recommend to use an I controller.
- The pole position identification comprises a monitoring function for the follow-up control. If a movement greater than the permissible movement set in C00650 is detected by the encoder system:
  - the pole position identification procedure is aborted.
  - the response parameterised in C00640 (Lenze setting: "Fault") is activated.
  - the error message "Pole position identification cancelled" is entered into the logbook of the controller.
- In order to detect a non-permissible blocking of the machine, a positive and negative test angle relative to the current position are defined after the identification. The machine must align itself to these two test angles within a tolerance of 25 %. The size of the test angle corresponds to the max. permissible movement set in C00650.

5.3.4 Optimising the switching performance of the inverter

Note!

Only required for servo control if the motor parameters are to be defined by a motor from a third-party manufacturer!

Always required for sensorless vector control and open loop V/f control!

- An optimum drive performance can only be achieved with the sensorless operating modes if the voltage errors in the inverter are compensated as exactly as possible.

An inverter generates a pulse-width-modulated, three-phase voltage system. Due to the design of the inverter, current-dependent and switching frequency-dependent losses inside of the inverter falsify the voltage that is output. As the voltage that is output is not measured, the losses have to be compensated by a suitable feedforward control. This compensation is based on an inverter error characteristic.

Among other things, the inverter error characteristic depends on the length of the motor cable and at least has to be individually determined once for the connected motor by means of the device command "Calculate inv. characteristic". For an automatic determination of the motor parameters, this ensures that the current has a sinusoidal form.

Danger!

This procedure may only be carried out during commissioning, not during operation!

- During the procedure the motor is energised so that:
  - it cannot be excluded that the connected mechanical components may move!
  - the windings heat up.
    If you repeat the procedure, ensure that the motor is not thermally overloaded (particularly if no temperature feedback is used).

For software versions lower than V4.0 the following applies:

- If the automatic brake operation is used, ensure that no basic function is requested or that the application is stopped before this procedure is called. Otherwise the applied holding brake could be released!
- For positioning applications you have to observe that the absolute position and the home position will get lost when this procedure is called. The loss of the home position is not signalled to the application. The following sequence has to be observed for positioning applications: 1.) Execute identification ➞ 2.) Save parameter set ➞ 3.) Restart controller ➞ 4.) Execute homing procedure.

Note!

For devices of the types 6 + 7 the Ixt monitoring may be activated during the inverter error characteristic is determined.

Remedy: Only start identification at a device utilisation (C00066) of 0 % and/or reduce rated motor current (C00088) and reset it to the original value after the identification.

How to determine the inverter error characteristic:

1. If the controller is enabled, inhibit the controller, e. g. with the device command C00002 = "41: Inhibit controller".
2. Execute device command C00002 = "71: Calculate inv. error characteristic".

   The procedure starts with controller enable, if
   - no other identification is active,
   - no error has occurred, and
   - no test mode is activated.

   If one of the above conditions is not met, the procedure is cancelled and the corresponding device command status is indicated under C00003.

   Note:
   By means of controller inhibit, the started procedure can be cancelled anytime, if required. Characteristic values that have already been determined in this case are rejected.

   For detailed information about the procedure, please see the following section "Sequence".

Tip!

For controller enable all sources for controller inhibit must be reset. In C00158 the sources for controller inhibit are displayed in a bit-coded manner.

The status of the device command activated under C00002 is indicated under C00003.

Sequence

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

$$\sqrt{2} \cdot \text{Rated device current}$$
or
$$\sqrt{2} \cdot 1.8 \cdot \text{Rated motor current}$$

- Ideally, the first value should be reached, the second value is to ensure that the load on the machine is not too high during this test.

During the procedure, the motor current rises up to the specified maximum value and falls back to "0" to repeat the cycle with a negative current sign.

- The maximum value is reached four times.
- The switching frequency is set to rated switching frequency and after the procedure, it is reset to the original value.
  - If the switching frequency should be changed later during operation, the characteristic will be adapted to the current switching frequency.

After successful completion...

...the controller is inhibited automatically and the detected characteristic is set in the controller.

- For a permanent acceptance of the characteristic, the parameter set must be saved afterwards (C00002 = "11: Save start parameters").
- With the device command C00002 = "42: Enable controller" the controller inhibit set automatically during the procedure can be deactivated again.

Tip!

The inverter error characteristic must only be detected again if the controller, motor, or motor cable has changed e.g. due to an exchange.

If an error occurs

If an error occurs during the procedure or the pulse inhibit gets active (e.g. due to short-time undervoltage), the procedure is terminated with controller inhibit and the detected characteristic is not considered.

Tip!

From software version V4.0: If it is not possible to determine the so-called "Inverter error characteristic", or if the results of the determination are incorrect, the device command C00002 = "70: Load Lenze inverter characteristic" can be used to load a characteristic typical for the device. ▶ Load Lenze inverter characteristic (📖 77)

5.3.5 Determining the motor parameters

Note!

Only required for servo control if the motor parameters are to be defined by a motor from a third-party manufacturer!

Always required for sensorless vector control!

- An optimum drive performance can only be achieved with the sensorless vector control if the motor parameters correspond to the real motor as exactly as possible.

To control an electrical machine, the motor parameters must be known.

- ▶ The motor parameters for Lenze motors are known and are already set accordingly by selecting them from the »Engineer« motor catalogue or reading out the ENP.
- ▶ The device command "Determine motor parameters" is used to automatically determine the motor parameters for a third-party motor that are listed in the following table – if they are not known:

Parameter	Information	ASM	SM
C00079	Mutual motor inductance	☑	
C00082	Motor rotor resistance	☑	
C00084	Motor stator resistance	☑	☑
C00085	Motor stator leakage induct.	☑	☑
C00091	Motor - cosine phi	☑	
C00092	Motor magnetising current	☑	

Danger!

This procedure may only be carried out during commissioning, not during operation!

- During the procedure the motor is energised so that:
  - it cannot be excluded that the connected mechanical components may move!
  - the windings heat up.
    If you repeat the procedure, ensure that the motor is not thermally overloaded (particularly if no temperature feedback is used).

For software versions lower than V4.0 the following applies:

- If the automatic brake operation is used, ensure that no basic function is requested or that the application is stopped before this procedure is called. Otherwise the applied holding brake could be released!
- For positioning applications you have to observe that the absolute position and the home position will get lost when this procedure is called. The loss of the home position is not signalled to the application. The following sequence has to be observed for positioning applications: 1.) Execute identification → 2.) Save parameter set → 3.) Restart controller → 4.) Execute homing procedure.

Requirements

- For the automatic determination of the motor parameters it is required that first the switching performance of the inverter has been optimised successfully, to ensure that the current has a sinusoidal form. Optimising the switching performance of the inverter (143)
- The motor parameters listed in the following table are excluded from the automatic determination and must be adapted to the motor used (see motor nameplate) before the determination.

Parameter	Information
C00081	Rated motor power
C00084	Motor stator resistance (Default setting is used as starting value for the automatic determination.)
C00087	Rated motor speed
C00088	Rated motor current (The current amount for the procedure is derived from this specification)
C00089	Rated motor frequency
C00090	Rated motor voltage

Note!

For devices of the types 9 + 10 (from 132 kW) the automatic determination of the motor parameters may fail and a corresponding status display is output.

Remedy: Parameterise the motor parameters manually by means of the manufacturer's data sheet.

How to determine the motor parameters:

1. If the controller is enabled, inhibit the controller, e. g. with the device command C00002 = "41: Inhibit controller".
2. Execute device command C00002 = "72: Determine motor parameters".

   The procedure starts with controller enable, if
   - no other identification is active,
   - no error has occurred, and
   - no test mode is activated.

   If one of the above conditions is not met, the procedure is cancelled and the corresponding device command status is indicated under C00003.

   Note:
   By means of controller inhibit, the started procedure can be cancelled anytime, if required, without altering the codes for the motor parameters.

   For detailed information about the procedure, please see the following section "Sequence".

Tip!

For controller enable all sources for controller inhibit must be reset. In C00158 the sources for controller inhibit are displayed in a bit-coded manner.

The status of the device command activated under C00002 is indicated under C00003.

Sequence

If all conditions are met, the impedance of the controlled system is determined for approx. 30 different frequencies. These values are used to determine the electrical machine parameters by means of a mathematical procedure.

- Since the procedure starts with very low frequencies and always considers several complete periods, the whole process takes approx. 3 minutes.
- During the procedure, the motor is energised with a current, the r.m.s. value of which corresponds to the lower of the following two values:

Rated device current

or

$\frac{1}{2} \cdot$ Rated motor current

After the parameters have been extracted from the impedance, they are checked for consistency with the required rated values. If an inconsistent parameter set is detected, is this an indication of faulty rated values on the nameplate.

Note!

During the procedure, the motor should not rotate.

With synchronous machines, this cannot always be ensured. Although the current flow is produced in the torque-neutral axis, asymmetries in the machine lead to a rotation of the rotor.

- In such a case, the measurement would be useless and would have to be repeated.
- As a remedy, we recommend to use a holding brake.

With asynchronous machines, slight rotations might possibly occur. Their influence on the measurements is, however, not worth mentioning.

- In case of uncertainties, the measurement should be repeated several times to check if the results for the stator resistance, the leakage inductance of the stator and the rotor resistance differ widely. This should not be the case.
- The magnetising inductance and the cos(φ) values are not that important for the diagnostics, because they are strongly non-linear.

After successful completion...

...controller inhibit is set automatically and the motor data determined are set in the corresponding codes.

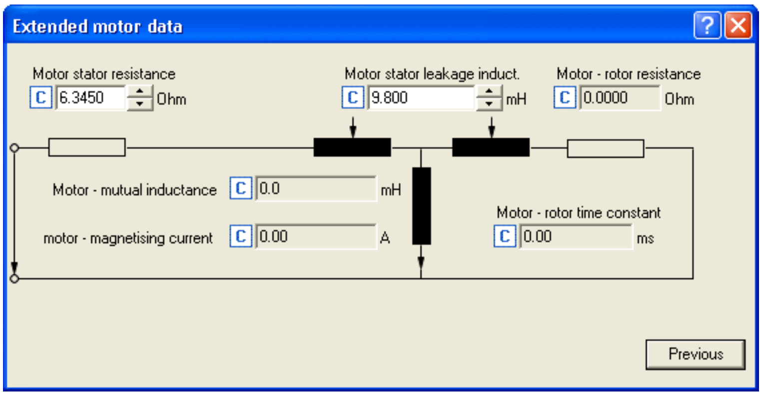
- For a permanent acceptance of the settings, the parameter set must be saved (C00002 = "11: Save start parameters").
- With the device command C00002 = "42: Enable controller" the controller inhibit set automatically during the procedure can be deactivated again.

If an error occurs

If an error occurs during the procedure or the pulse inhibit gets active (e.g. due to short-time undervoltage), the procedure is terminated with controller inhibit without changing the codes for the motor parameters.

Display and manual adjustment of motor data

In the »Engineer« you can have an equivalent circuit diagram with the motor parameters displayed by clicking on the **Further motor data...** button on the **Application parameters** in the dialog level *Overview→Motor→Motor*:



- The representation of the equivalent circuit diagram depends on the motor control selected (C00006).
- The motor stator resistance (C00084) and motor stator leakage inductance (C00085) can be altered directly via the input fields in the equivalent circuit diagram.
- The motor magnetising current (C00092) is displayed as comparison value to the motor current (C00054).
  - The motor magnetising current must especially be observed in case of a no-load operation, both at standstill and with rated speed.
  - The motor magnetising current is directly calculated from the rated motor current (C00088) and the motor power factor (C00091).
- The mutual motor inductance can be indirectly adapted via the parameter Lh adjustment (C02861) in the range of 50 ... 200 %. The mutual motor inductance evaluated in percent is shown in C00079.
- The motor rotor resistance can be indirectly adapted via the parameter Rr adjustment (C02860) in the range of 50 ... 200 %. The motor rotor resistance evaluated in percent is shown in C00082.

5.4 Servo control (SC)

In the Lenze setting the servo control for synchronous motors is selected in C00006.

After the motor and controller are optimally adjusted to each other, no more basic settings are required for servo control.

Tip!

How to optimise the control behaviour and adjust it to the concrete application is described in the chapter "Optimise control mode". (📖 152)

From software version V2.0 the parameterisable additional function "Field weakening for synchronous machines" is provided for the servo control. (📖 215)

5.4.1 Optimise control mode

The "optimisation steps" given in the following table serve to further optimise the control behaviour of the servo control and adjust it to the concrete application.

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

Optimisation steps	
1.	Optimise current controller. (📖 153) • The current controller should always be optimised if a motor of a third-party manufacturer with unknown motor data is used!
	Parameterise selected technology application in the »Engineer« and load it into the controller. • See description for the corresponding technology application. • During operation (with setpoint selection) further steps can be carried out to optimise the motor control:
2.	Optimising the speed controller. (📖 156) • Via running a typical speed profile and recording the ramp response of the speed controller with the oscilloscope.
3.	If the speed controller optimisation did not achieve the intended result:
	Set current setpoint filter (band-stop filter). (📖 159) • In order to suppress or damp (mechanical) resonant frequencies, two current setpoint filters are integrated in the speed control loop of the controller which are switched off in the default setting but can be parameterised accordingly, if required.
	Then readjust the speed controller: Optimising the speed controller. (📖 156)
4.	Optimising phase controller. (📖 162) • Via running a typical speed profile and recording the ramp response of the phase controller with the oscilloscope.
5.	Optimise response to setpoint changes. (📖 163) • Via running a typical speed profile and recording the inputs and outputs of the speed controller with the oscilloscope.
6.	Setting the field weakening for asynchronous machines. (📖 165) • By means of traversing a speed profile 0 ↔ n_{max} and recording the speed, flow, and D-current setpoints/ actual values with the oscilloscope.
7.	Save »Engineer« project.

Tip!

To run a typical speed profile for optimising the motor control, you can also use the basic function "manual jog" with suitably adapted manual jog parameters if this basic function is supported by the technology application selected. ▶ Manual jog (📖 398)

5.4.1.1 Optimise current controller

Note!

The current controller should always be optimised if a motor of a third-party manufacturer with unknown motor data is used!

An optimisation of the current controller is useful, as the two controller parameters gain (C00075) and reset time (C00076) depend on the maximum current required and the switching frequency set.

For this purpose, the controller parameters only need to be adapted once at a fixed switching frequency.

We recommend to select a switching frequency

- as low as possible if the controller is to be operated frequently at the maximum current limit.
- of 8 kHz (up to and including model 7) or 4 kHz (from model 8 onwards) if the maximum current limit will not be reached or will only be reached rarely.

The controller parameters are then automatically adapted to the other switching frequencies.

In a test mode you can select current setpoint step-changes and optimise the setting of both control parameters by evaluating the step responses.

- The starting values for gain and reset time can be calculated with the following formula:

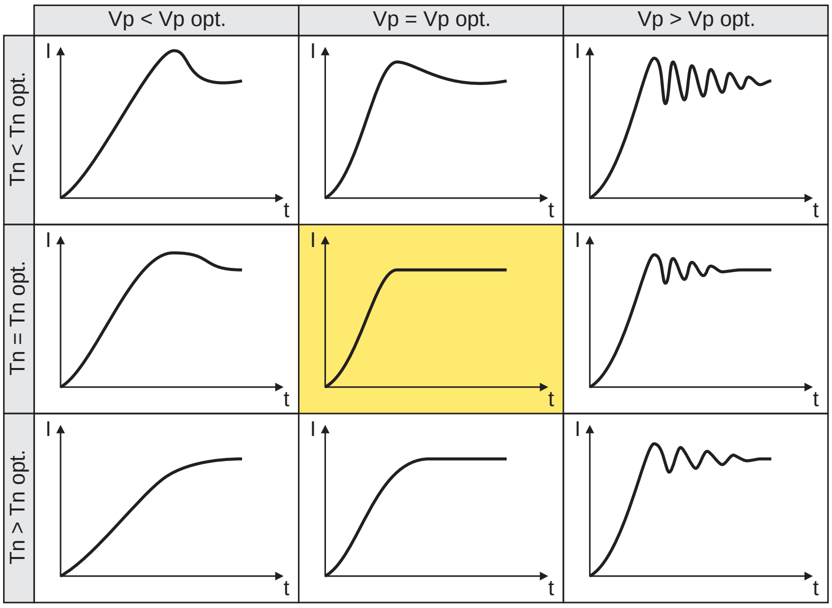
$$\text{Gain} = \frac{\text{Stator leakage inductance}}{340\ \mu\text{s}}$$

$$\text{Reset time} = \frac{\text{Stator leakage inductance}}{\text{Stator resistance}}$$

How to optimise the current controller in the test mode:

1. If the controller is enabled, inhibit the controller, e. g. with the device command C00002 = "41: Inhibit controller".
2. Set a fixed switching frequency using C00018. Observe the above mentioned recommendations.
3. Activate one of the two following optimisation modes for the current controller:
   - C00398 = "3: Current controller optimisation mode":
     After controller enable, the motor is supplied with current as long as the controller is enabled.
   - From software version V7.0:
     C00398 = "4: Current controller optimisation mode pulse":
     The motor is supplied with voltage for 50 ms after controller enable. Due to this time limit, the load of the machine is reduced. Afterwards, the controller is inhibited automatically.
4. Select the effective value of the current setpoint step change under C00022.
   - The peak value of the measurable motor current will be 1.41 times higher.
5. Enable the controller for a short time and measure the step response of the motor current in the motor phases using the oscilloscope and clamp-on ammeters or record the field-oriented direct-axis current using the Oscilloscope function in the »Engineer«. (📖 590)
   - Variable of the motor control to be recorded:
     Current.dnActualDirectCurrent (field-oriented direct-axis current)

6. Evaluate the step response:



7. Change the gain Vp under C00075 and the reset time Tn under C00076.
8. Repeat steps 4 ... 6 until the optimum step response of the motor current is reached.
   - In the optimised state the current rise time typically is 0.5 ... 1 ms.
   - If the adjustment results are not satisfactory, the decoupling network can be additionally activated via the setting C00074 = "1". After this, repeat the steps 2 ... 6.
   - In case of MCS, satisfying results may only be achieved with a current-dependent correction of the current controller parameters based on the saturation behaviour of the motor stator leakage inductance. For this purpose, it is required to use a motor with an electronic nameplate (ENP) or to set the saturation characteristic manually. ▶ Correction of the stator leakage inductance... (🕮 210)
9. After the optimisation has been completed, deactivate the test mode again (C00398 = "0: Test mode deactivated").
10. Save parameter set (C00002 = "11: Save start parameters").

5.4.1.2 Optimising the speed controller

The speed controller is designed as PID controller.

Gain setting

The proportional gain Vp is selected under C00070:

1. Select the speed setpoint.
2. Increase C00070 until the drive becomes unstable (observe motor noises).
3. Reduce C00070 until the drive becomes stable again.
4. Reduce C00070 to approx. half the value.

Reset time setting

The reset time is selected under C00071:

1. Reduce C00071 until the drive becomes unstable (observe motor noises).
2. Increase C00071 until the drive is stable again.
3. Increase C00071 to approx. double the value.

Rate time setting

The rate time Td is selected under C00072:

- ▶ Increase C00072 during operation until an optimum control behaviour is reached.

Using the ramp response for setting the speed controller

When operation of the mechanics at the stability limit is not possible, the ramp response can be used to set the speed controller. The proceeding is similar to optimising the current controller.

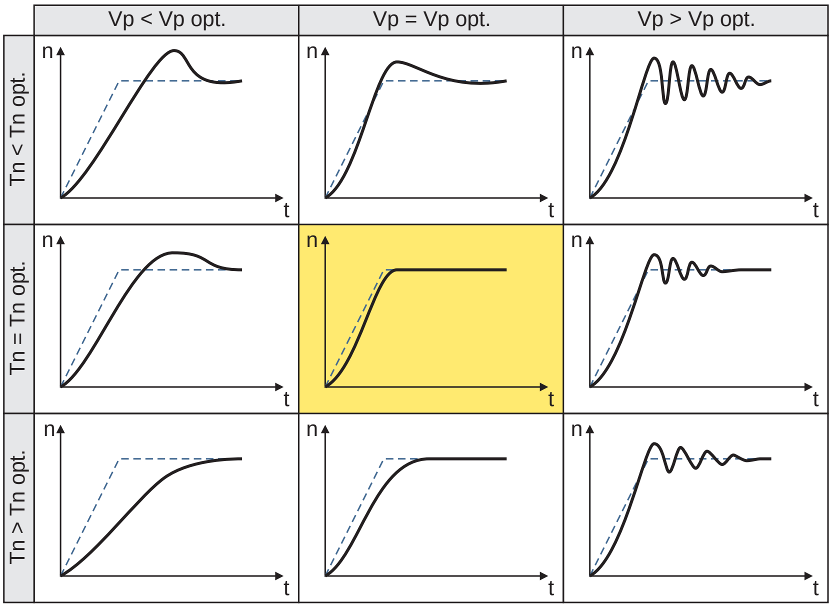
Stop!

If the controller parameters are preset unfavourably, the control can tend to heavy overshoots up to instability!

- Following and speed errors can adopt very high values.
- For sensitive mechanical systems, the corresponding monitoring systems have to be activated.

How to optimise the speed controller setting by means of the ramp response:

1. Run a typical speed profile and record the ramp response of the speed with the Oscilloscope. (📖 590)
   - Motor control variables to be recorded:
     Speed.dnSpeedSetpoint (speed setpoint)
     Speed.dnActualMotorSpeed (actual speed value)
2. Evaluate the ramp response:



   - Solid line = ramp response (actual speed value)
   - Dash line = speed setpoint
3. Change the gain Vp under C00070 and the reset time Tn under C00071.
4. Repeat steps 1 ... 3 until the optimum ramp response is reached.
5. Save parameter set (C00002 = "11").

Setting of actual speed filter

In order to maximise the dynamics of the speed control loop, the actual speed filter should be operated with a time constant as low as possible (C00497). The lower the time constant the higher the gain of the speed controller. Since actual value filters have the task to dampen measuring errors or interference components, it must be found a compromise between filter task and the resulting delay.

If a Lenze motor is selected from the motor catalogue, a time constant is automatically preset in C00497 which serves to operate the motor even with a faulty detection (e.g. in case of a bad shield connection).

When using EMC-compliant systems or high-quality encoders, you can reduce the preset time constant considerably. For this purpose, the running noise of the motor can be used for setting C00497 at constant speed.

If this is not possible, e.g. due to a too loud environment or because the motor is too far away, the noise of the actual speed value or the setpoint torque value can be used for evaluation by means of the Oscilloscope. Please observe that the speed controller gain Vp (C00070) in is used for the torque setpoint.

Dynamics of the actual value detection

Another element which influences the maximally achievable control dynamics, is the dynamics of the actual value detection itself. In case of optical encoders, the time delay by the actual value detection does not need to be considered. This does not apply to resolvers.

The resolver evaluation of the controller is adapted to the resolver types mounted in Lenze motors and offers a good compromise between the dynamic performance and interference suppression. If the resolver is used as a speed feedback system, the dynamic performance of the resolver evaluation determines, among other things, the maximum speed controller gain by means of which stable operation is possible.

From software version V5.0 onwards, it is possible to increase the dynamics of the resolver evaluation in C00417 in an EMC-compliant system (with low interference) without a quality loss in the speed signal.

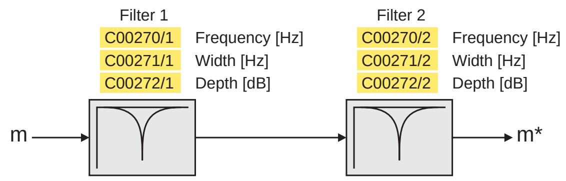
▸ Adaptation of the resolver evaluation dynamics (📖 252)

Lenze

5.4.1.3 Set current setpoint filter (band-stop filter)

Due to the high dynamic performance or the high limit frequency of the closed current control loop, mechanical natural frequencies can be excited, which can result in resonance and thus cause the speed control loop to become unstable.

In order to suppress or damp these resonant frequencies, two current setpoint filters are integrated in the speed control loop of the controller, which need to be parameterised. In the Lenze setting, these filters are switched off:



m = Output of the jerk limitation

[5-2] Optional current setpoint filters (filter cascade) in the speed control loop

Overview of parameters for current setpoint filter

Parameter	Information	Lenze setting	
		Value	Unit
C00270/1	Freq. - current setpoint filter 1	200.0	Hz
C00270/2	Freq. - current setpoint filter 2	400.0	Hz
C00271/1	Width current setp. filter 1	20.0	Hz
C00271/2	Width current setp. filter 2	40.0	Hz
C00272/1	Depth current setp. filter 1	0	DB
C00272/2	Depth current setp. filter 2	0	DB

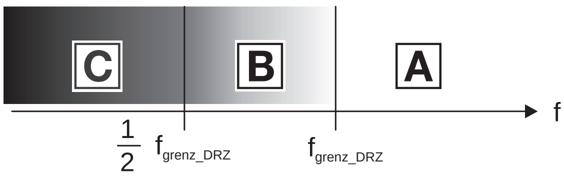
Use of the current setpoint filters depending on the resonant frequency

Stop!

If the filter parameters are set incorrectly, the impaired closed-loop control can respond with too large overshoots and cause the controller to become unstable, e.g. if the filter width is set to a value more than twice as large as the filter frequency.

After setting the filter parameters, the drive behaviour during stop and quick stop (QSP, Fail-QSP) must be checked. If impairments exist,

- the still running drive must either be coasted down by activating the controller inhibit or immediately be brought to a standstill via a brake.
- the speed controller must afterwards be optimised again.
- the test procedure must be repeated.



A Normal and intended use of the current setpoint filter
B Use of the current setpoint filter is only partially suitable
C The current setpoint filter should not be used

[5-3] Use of the current setpoint filter depending on the resonant frequency

▶ Resonant frequencies ≥ f_{limit_SPEED} = 70 Hz ... 110 Hz

This filter is suitable for use with resonant frequencies in the range around or above the limit frequency of the speed controller.

▶ Resonant frequencies < f_{limit_SPEED}

Please follow the Lenze recommendation and select suitable speed profiles, S-ramp or S-rounding, for avoiding resonances.

Setting of the current setpoint filter

Since the frequency response of the speed controlled system is only rarely known to such an extent that the current setpoint filters can be adjusted to the controlled system in the run-up, the following example describes an experimental procedure for setting the current setpoint filters:

How to set the current setpoint filters:

1. Adjust the current control loop.
2. Go to C00071 and adapt the reset time of the speed controller to the filter time constant of the speed filter (C00497) and the equivalent time constant of the current control loop: C00071 = 16 * (C00497 + 200 µs)

   Note: The setting of C00071 incorporates the equivalent time constant of the current control loop. The indicated 200 µs are typical in a power range of up to 20 kW. Beyond it, higher time constants may occur.
3. Slowly increase the proportional gain in C00070 until the speed control loop starts to become unstable (acoustic determination or measuring of the motor current).
4. Measure the oscillation frequency using an oscilloscope (observe current or speed).
5. Set the measured oscillation frequency in C00270/1 as filter frequency.
6. Set "50%" of the filter frequency in C00271/1 as filter width.
   - Example: Filter frequency = 200 Hz → filter width = 100 Hz.
7. Set "40 dB" in C00272/1 as filter depth.
   - If the filter depth is set to "0 dB" (default setting), the filter is not active.
8. Further increase the proportional gain in C00070 until the speed control loop starts to become unstable again.
   - If the oscillation frequency has changed now, readjust the filter frequency by trimming. The use of a second filter is ineffective here.
   - If the oscillation frequency remains the same, readjust the filter depth and/or the filter width by trimming (the first reduces the amplitude, the second lets the phase rotate faster).
   - Repeat step 8 until the desired behaviour or the limit of a sensible speed controller gain has been reached.
9. Save parameter set (C00002 = "11: Save start parameters").

Note!

Readjust the speed controller after setting the current setpoint filter.
▶ Optimising the speed controller. (📖 156)

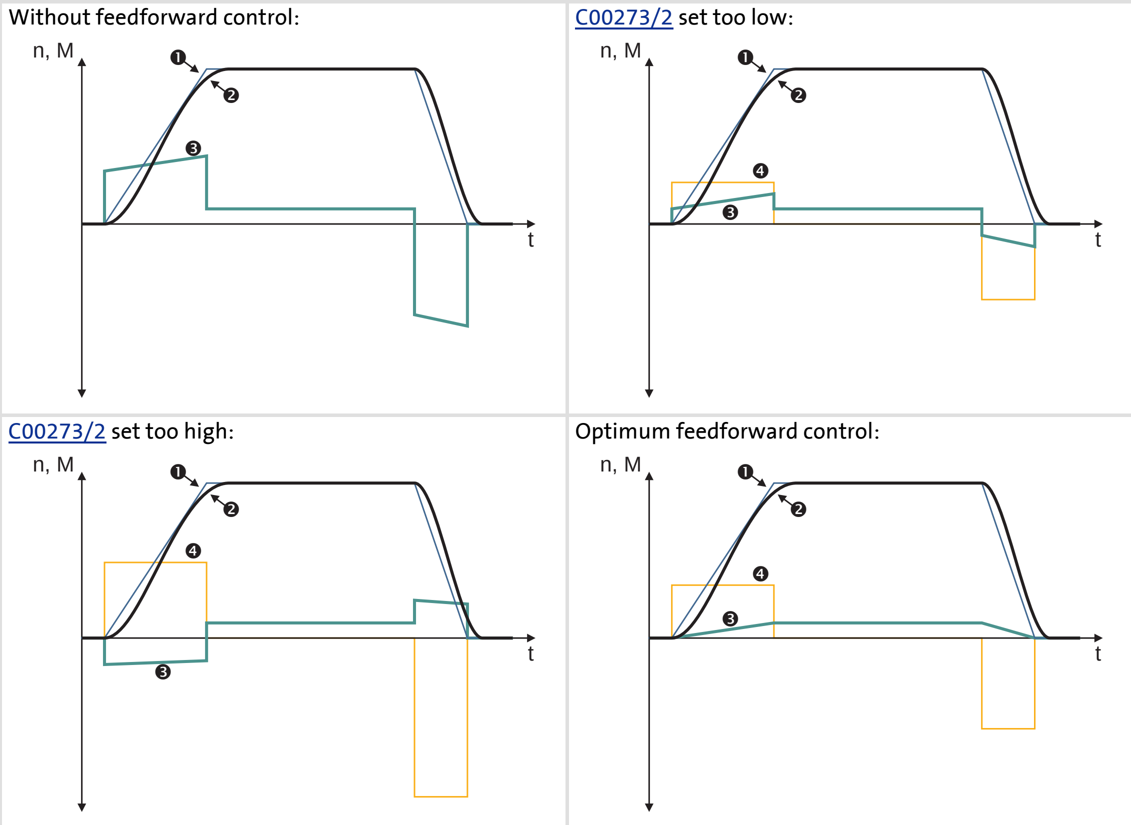
5.4.1.4 Optimising phase controller

How to optimise the phase controller setting by means of the ramp response:

1. Run a typical speed profile and record the ramp response of the phase controller with the Oscilloscope. (590)
   - Motor control variables to be recorded:
     Speed.dnSpeedSetpoint (speed setpoint)
     Speed.dnActualMotorSpeed (actual speed value)
     Speed.dnOutputPosCtrl (phase controller output)
     Position.dnEncounteringError (following error)
2. Adjust the gain Vp of the phase controller under C00254 and repeat oscilloscope recording until the intended following error behaviour is reached and the motor runs sufficiently smoothly during the constant travel phase.
3. Save parameter set (C00002 = "11: Save start parameters").

5.4.1.5 Optimise response to setpoint changes

Setting the load moment of inertia under C00273/2 does not always provide the optimum torque feedforward control. Depending on the application, an adjustment of the setting under C00273/2 may be necessary to optimise the response to position/speed setpoint changes by means of the torque feedforward control.



❶ Speed setpoint (*Speed.dnSpeedSetpoint* signal)
❷ Actual speed value (*Speed.dnActualMotorSpeed* signal)
❸ Speed controller output (*Torque.dnOutputSpeedCtrl* signal)
❹ Torque feedforward control (*Torque.dnTotalTorqueAdd* signal)

[5-4] Typical signal characteristics for different settings of the load moment of inertia

Apart from the load moment of inertia, effects can be compensated with C00273/2, which in the closed speed control loop are identified by the speed controller. These for example include the friction torques.

Below you will find a description of a procedure for optimising the feedforward control behaviour starting from the system's 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 Oscilloscope. (590)
   - Motor control variables to be recorded:
     Speed.dnSpeedSetpoint (speed setpoint)
     Speed.dnActualMotorSpeed (actual speed value)
     Torque.dnOutputSpeedCtrl (speed controller output)
     Torque.dnTotalTorqueAdd (torque feedforward control)
   - Application variable to be recorded (if available):
     L_LdMonitFollowError1.dnFollowErrorIn_p (following error)

   It is essential for optimising the response to setpoint changes to monitor the speed controller output (*Torque.dnOutputSpeedCtrl*) and the torque feedforward control (*Torque.dnTotalTorqueAdd*). The effect of the feedforward control can also be observed in the following error.
2. Select the signal source required for the torque setpoint (feedforward control path) under C00276.
3. Estimate the load moment of inertia and set it under C00273/2 with regard to the motor end (i.e. considering the gearbox factors).
4. Repeat the oscilloscope recording (see step 1).

   Now the oscillogram should show that part of the required torque is generated by the feedforward control (*Torque.dnTotalTorqueAdd*) and the speed controller output signal (*Torque.dnOutputSpeedCtrl*) should be correspondingly smaller. The resulting following error decreases.
5. Change the setting under C00273/2 and repeat the oscilloscope 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. [5-4]).
6. Save parameter set (C00002 = "11: Save start parameters").

5.4.1.6 Setting the field weakening for asynchronous machines

For the following setting instructions it is assumed that the drive has been adjusted in the base speed range (inverter error characteristic, motor parameters, current controller, speed controller, current setpoint filter, angle controller, torque feedforward control) and is running satisfactorily to the rated motor speed.

How to set the field weakening for an asynchronous machine:

1. Set the desired maximum speed (with field weakening) in C00011.
2. Carry out the following basic setting for the controller parameters for third-party motors:
   - Field controller gain (C00077) = 1 / (2 * C00082 * 500 μs)
   - Field controller reset time (C00078) = motor rotor time constant (C00083)
   - Field weakening controller gain (C00577) = 0 [Vs/V]
   - Field weakening controller reset time (C00578)
     = 4 ms / (0.3 ... 1.0 * 60) * C00059 * 2π * C00011 [rpm] * s
     (with factor 0.3 ... 1.0 for motor with rated power of 400 kW ... 0.4 kW)

   Optimising the static behaviour in the field weakening range:
3. By means of speed ramp (acceleration time several seconds), slowly accelerate to the field weakening range up to maximum speed (C00011),and decelerate to speed 0 again and record the signal characteristic using the Oscilloscope (see example oscillogram [5-5]).
   - From the entry into the field weakening range, the flow setpoint (output of the field weakening controller) should decrease with 1/n. Influences of the DC-bus voltage may be seen in the flow setpoint. The signal characteristic should preferably be "smooth".
   - From the entry into the field weakening range, the D-current setpoint (output of the field controller) should always decrease with 1/n. In the signal characteristic no heavy vibrations may occur.

Optimising the dynamic behaviour in the field weakening range:

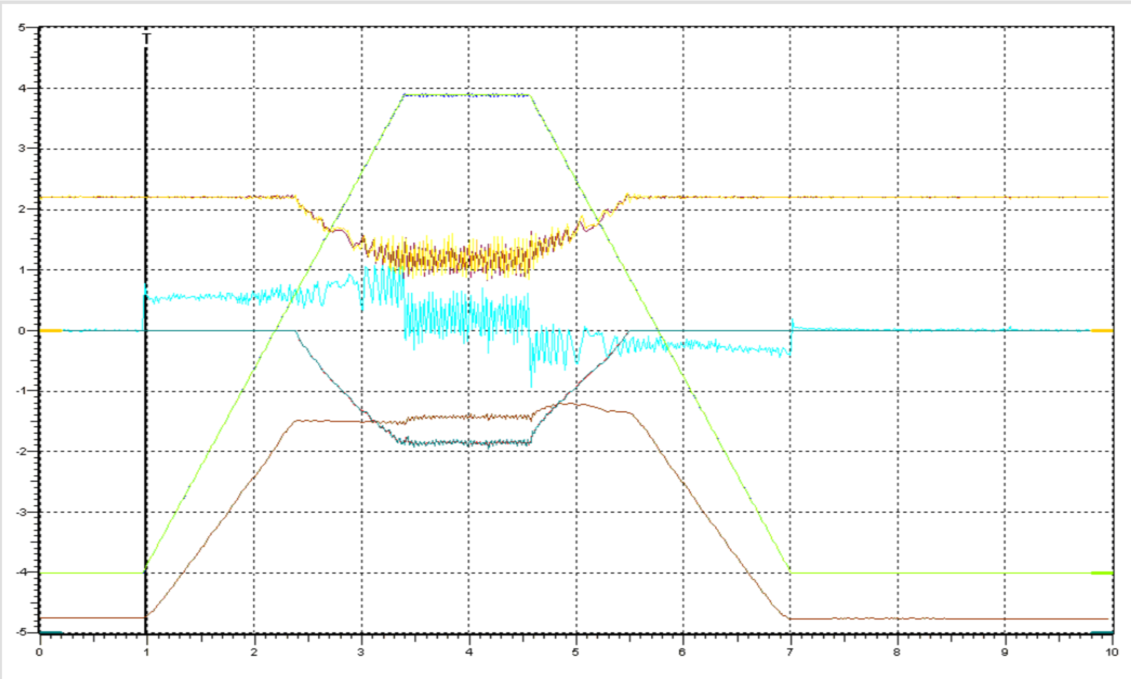
4. Adapt the dynamic performance to the behaviour required for the machine application.
5. Repeat the recording of the speed characteristic and record the small signal range in the field weakening range ($n_1 \leftrightarrow n_2$).
   - If the flow setpoint is adapted to the speed too slowly, increase the dynamic performance of the field weakening controller: reduce the reset time (C00578) in small steps, the gain (C00577) should remain set to 0 [Vs/V] for most machines.
   - If the flow setpoint in the field weakening range falls "too early" with $1/n^2$ (stability limit of the machine reached), the leakage inductance of the motor (C00085) may be reduced a bit.
   - If the actual flow value follows the flow setpoint too slowly, increase the dynamic performance of the field controller: increase gain (C00077), reduce reset time (C00078).
   - If the actual D-current value does not correspond enough to the D-current setpoint, the dynamic performance of the current controller has to be adapted. ▶ Optimise current controller (📖 153)
   - If the motor speed does not feature the desired characteristic, the speed controller has to be readjusted with maximum speed in the field weakening range. ▶ Optimising the speed controller (📖 156)

Checking the motor parameters:

6. Carry out dynamic measurement in the range $-n_{Max} \leftrightarrow +n_{Max}$ and record the motor speed using the Oscilloscope.
   - The objective is a preferably linear speed characteristic.
   - In particular check the ranges around the rated motor speed and speed 0 and, if required, improve them by adjusting R_R (C02860) or L_H (C02861)!

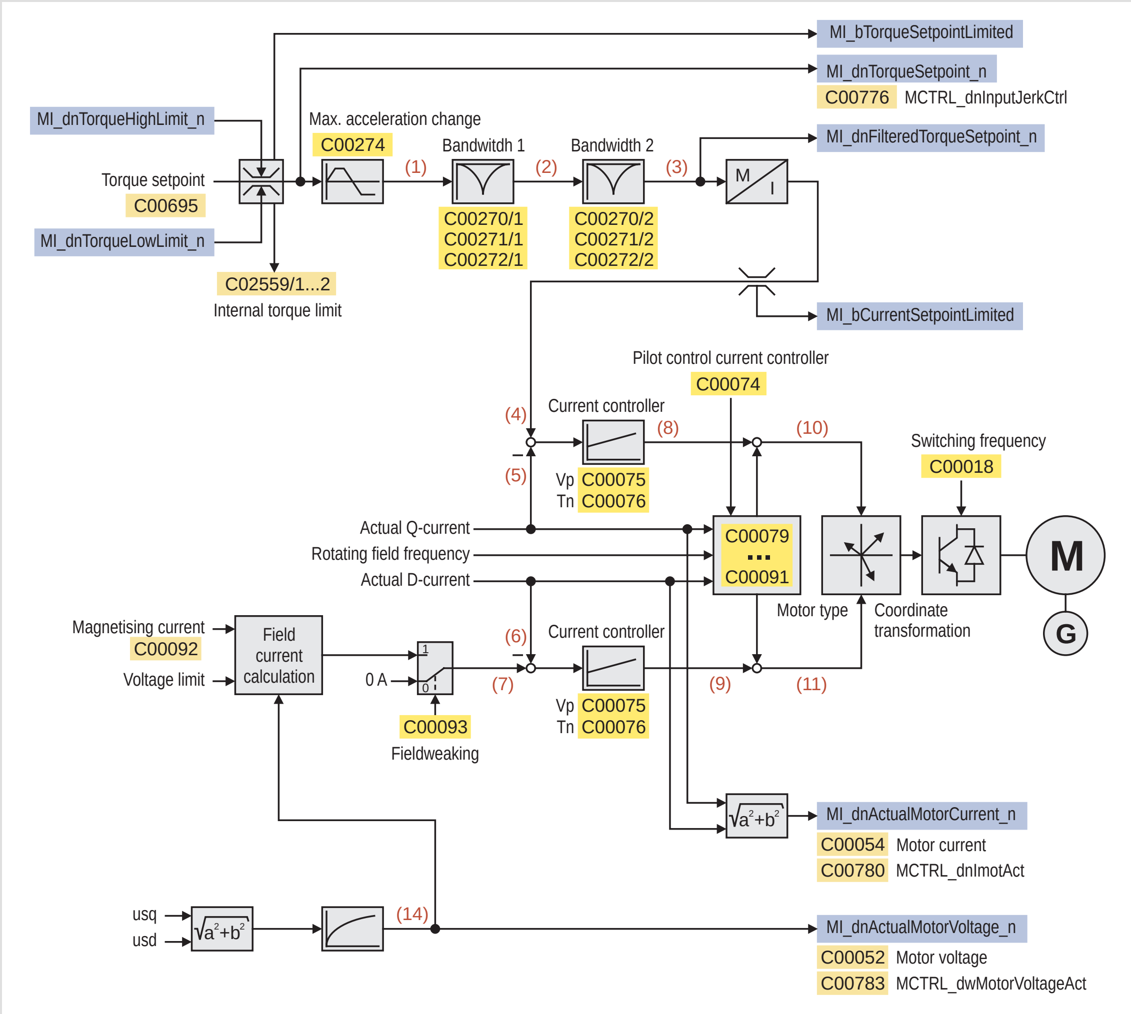
Example oscillogram

	Ch	Variable of the motor control	Unit	1/div	Offset	Position
	1	Speed.dnActualMotorSpeed (current speed)	rpm	1k	0	-4
	2	Voltage.dnActualMotorVoltage (current motor voltage)	V	100	0	-5
	3	Torque.dnActualMotorTorque (current motor torque)	Nm	500m	0	0
	4	Speed.dnSpeedSetpoint (speed setpoint)	rpm	1k	0	-4
	5	Common.dnFluxSet (flux setpoint)	%	20	0	-5
	6	Common.dnActualFlux (actual flux value)	%	20	0	-5
	7	Current.dnDirectCurrentSet (D-current setpoint)	A	1	0	0
	8	Current.dnActualDirectCurrent (actual D-current value)	A	1	0	0



[5-5] Example oscillogram

5.4.2 Signal flow (servo control for synchronous motor)



[5-6] Signal flow - servo control for synchronous motor

See also:

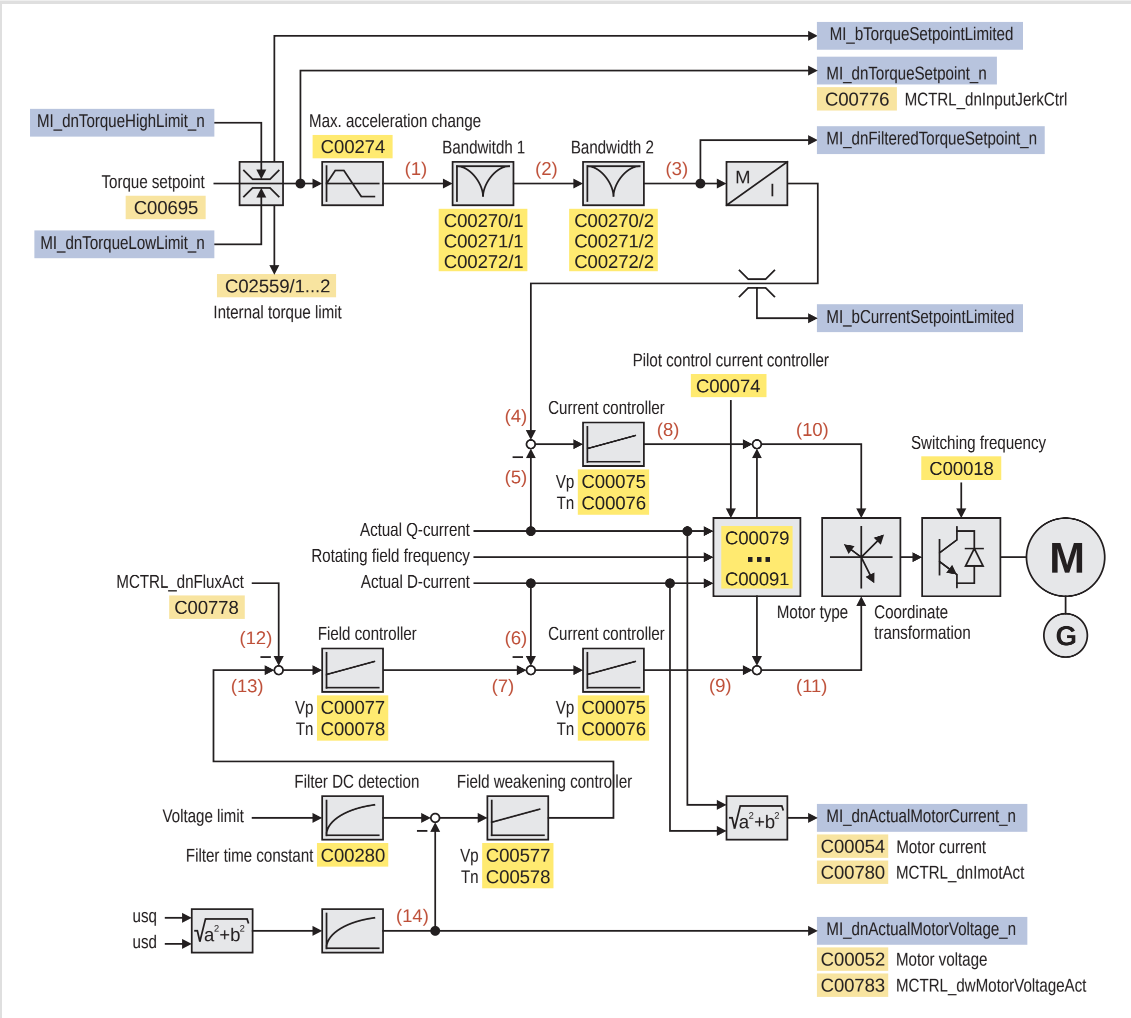
- Signal flow - encoder evaluation (📖 245)
- Signal flow - speed follower (📖 504)
- Signal flow - torque follower (📖 510)
- Signal flow - position follower (📖 497)

Internal variables of the motor control (oscilloscope signals)

- The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (📖 590)

No.	Variable of the motor control	Meaning
(1)	Torque.dnInputNotchFilter1	Torque setpoint at the band-stop filter input 1
(2)	Torque.dnInputNotchFilter2	Torque setpoint at the band-stop filter 2 input
(3)	Torque.dnFilteredTorqueSetpoint	Filtered torque setpoint
(4)	Current.dnQuadratureCurrentSet	Q current setpoint
(5)	Current.dnActualQuadratureCurrent	Actual Q current
(6)	Current.dnActualDirectCurrent	Actual D current
(7)	Current.dnDirectCurrentSet	D current setpoint
(8)	Voltage.dnOutputQuadratureCurrentCtrl	Q-output voltage of the current controller
(9)	Voltage.dnOutputDirectCurrentCtrl	D-output voltage of the current controller
(10)	Voltage.dnQuadratureVoltage	Q voltage
(11)	Voltage.dnDirectVoltage	D voltage
(12)	-	
(13)	-	
(14)	Voltage.dnActualMotorVoltage	Current motor voltage

5.4.3 Signal flow (servo control for asynchronous motor)



[5-7] Signal flow - servo control for asynchronous motor

See also:

- Signal flow - encoder evaluation (📖 245)
- Signal flow - speed follower (📖 504)
- Signal flow - torque follower (📖 510)
- Signal flow - position follower (📖 497)

Internal variables of the motor control (oscilloscope signals)

- The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (📖 590)

No.	Variable of the motor control	Meaning
(1)	Torque.dnInputNotchFilter1	Torque setpoint at the band-stop filter input 1
(2)	Torque.dnInputNotchFilter2	Torque setpoint at the band-stop filter 2 input
(3)	Torque.dnFilteredTorqueSetpoint	Filtered torque setpoint
(4)	Current.dnQuadratureCurrentSet	Q current setpoint
(5)	Current.dnActualQuadratureCurrent	Actual Q current
(6)	Current.dnActualDirectCurrent	Actual D current
(7)	Current.dnDirectCurrentSet	D current setpoint
(8)	Voltage.dnOutputQuadratureCurrentCtrl	Q-output voltage of the current controller
(9)	Voltage.dnOutputDirectCurrentCtrl	D-output voltage of the current controller
(10)	Voltage.dnQuadratureVoltage	Q voltage
(11)	Voltage.dnDirectVoltage	D voltage
(12)	Common.dnActualFlux	Actual flux value
(13)	Common.dnFluxSet	Flux setpoint
(14)	Voltage.dnActualMotorVoltage	Current motor voltage

5.5 Sensorless vector control (SLVC)

This function extension is available from software version V3.0!

If this motor control mode is set in C00006, a considerably higher torque and a lower current consumption in idle state can be achieved compared to the V/f control mode.

Note!

Observe the following application limits of the sensorless vector control:

- Only approved for power up to 55 kW and horizontal applications (no hoists or lifting equipment)
- For single drives only
- For asynchronous motors only
- Not suitable for operation in generator mode/braking operation (e. g. unwinders)

Tip!

For vertical drives/hoists, use the servo control (with feedback), or the V/f control with activated voltage vector control (VVC), which supports vertical drives/hoists up to 55 kW.

5.5.1 Basic settings

After the motor and controller have been optimally adjusted to each other, the "initial commissioning steps" described in the following table are sufficient for a quick initial commissioning.

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

Initial commissioning steps	
1.	Parameterising speed and torque controller. (📖 174)
2.	Additional "flying restart" function: • In the Lenze setting, this parameterisable additional function is activated. • If the flying restart function is not required, deactivate this function. ▸ Flying restart function (📖 218) STOP **Only deactivate the flying restart if it is ensured that the drive is always at standstill in the case of controller enable!**
3.	Additional "DC-injection braking" function: • In the Lenze setting, this parameterisable additional function is deactivated. • If DC-injection braking is required, activate this function. ▸ DC-injection braking (📖 221)

Tip!

A precise adjustment of the motor parameters for an improved concentricity factor and stability is described in the chapter "Optimising motor parameters". (📖 175)

How to optimise the control behaviour and adjust it to the concrete application is described in the chapter "Optimise control mode". (📖 181)

The parameterisable additional functions are described in the chapter of the same name "Parameterisable additional functions". (📖 209)

5.5.1.1 Parameterising speed and torque controller

Short overview: Parameters for controller settings

Parameter	Information	Lenze setting Value	Unit
C00070	Speed controller gain	0.500	Nm/rpm
C00071	Speed controller reset time	24.0	ms
C00987	SLVC: Torque controller gain	0.5000	Hz/A
C00988	SLVC: Torque controller reset time	10.00	ms

Typical controller settings

The following table contains typical guide values concerning the setting of the speed and torque control for different device types/motor powers:

Device type	Motor power	Speed controller		Torque controller	
E94ASx	(4-pole standard ASM)	Gain C00070 [Nm/rpm]	Reset time C00071 [ms]	Gain C00987 [Hz/A]	Reset time C00988 [ms]
E0024	0.37 kW	0.0122	50.00	5.0833	10.00
E0034	0.75 kW	0.0138	50.00	3.0500	10.00
E0044	1.50 kW	0.0264	50.00	1.9818	10.00
E0074	3.00 kW	0.0411	50.00	1.1077	10.00
E0134	5.50 kW	0.0674	50.00	0.5965	10.00
E0174	7.50 kW	0.1183	50.00	0.3303	10.00
E0244	11.00 kW	0.1183	50.00	0.3303	10.00
E0324	15.00 kW	0.2244	50.00	0.2368	10.00
E0474	22.00 kW	0.3442	50.00	0.1547	10.00
E0594	30.00 kW	1.1503	50.00	0.1232	10.00
E0864	45.00 kW	1.7400	50.00	0.0817	10.00
E1044	55.00 kW	2.1712	50.00	0.0661	10.00

[5-1] Typical controller settings

The gain for the field current controller (C00985) and the gain for the cross current controller (C00986) are initially set to "0.00".

5.5.2 Optimising motor parameters

Although the motor parameters have been determined before as described in the chapter "Adjusting motor and controller to each other", an additional optimisation may be required in the following cases using the adjustment processes described in this chapter:

- When the concentricity factor in the lowest speed range is to be improved.
- When the stability in the lower speed range is to be improved.
- When the rated torque is not reached in the rated point, i. e. at rated speed and rated current.
- When a too high magnetising current is injected in idle state.

General information on the motor parameter adjustment

The motor stator resistance can generally always be adjusted with a passive load since the motor is stopped when this parameter is set.

An optimisation of the mutual motor inductance, however, is only sensible when the motor rotates in the medium speed range. In the majority of cases, the no-load operation is sufficient for this adjustment. In contrast to the rated operation, the no-load operation is also possible for initial commissioning in many applications.

The motor rotor resistance can only be adjusted exactly if the current motor speed is available. Thus, for this adjustment only applications are considered where a speed measurement is possible, even with a manual tachometer. If, under operating conditions (e.g. at rated load), the motor consumes more than the rated motor current indicated, an adjustment can also be executed by reducing the mutual motor inductance.

Note!

To execute the adjustment processes described in the following subchapters, the controller must always be enabled!

5.5.2.1 Motor power factor

Together with the rated motor current, the motor power factor (C00091) defines the motor magnetising current (C00092) and thus the current consumption of the controller in idle state.

Note!

The following adjustment of the motor power factor should be executed after the motor parameters have been determined and when the value set in C00091 deviates more than 10 % from the data on the motor nameplate.

If the setting of the motor power factor in C00091 is changed, the setting of the mutual motor inductance also changes in C00079.

For the adjustment of the motor power factor, first the motor current in idle state at rated speed is determined in the control type "V/f control". Afterwards the motor power factor is set in the control type "Sensorless vector control" so that the motor magnetising current corresponds to the previously determined no-load current.

How to adjust the motor power factor:

1. If the controller is enabled, inhibit the controller, e. g. with the device command C00002 = "41: Inhibit controller".
2. Select the control type VFCplus: V/f control" in C00006.
3. Select setpoint speed 0 rpm
4. Enable controller.
5. Slowly increase setpoint speed to rated speed (no field weakening) and then keep it at rated speed constantly.
6. Take down the motor current displayed in C00054.
7. Slowly reduce the setpoint speed to 0 rpm again.
8. Inhibit controller.
9. Go to C00006 and select the "SLVC: Sensorless vector control" again.
10. Set Lh adjustment in C02861 to 100 %.
11. Set the motor power factor (C00091) so that the following applies:
    motor magnetising current (C00092) ≈ motor current taken down before.
12. Save parameter set (C00002 = "11: Save start parameters").

5.5.2.2 Motor stator resistance

For the adjustment of the motor stator resistance, first the motor current at standstill (without load of the motor) is compared to the motor magnetising current. Afterwards the setting of the motor stator resistance is changed step by step until the motor current stably reaches the motor magnetising current.

How to adjust the motor stator resistance:

1. Select setpoint speed 0 rpm or activate quick stop.
2. Enable controller.
3. Compare the motor current displayed in C00054 with the motor magnetising current displayed in C00092.
4. Inhibit controller.
5. If motor current > motor magnetising current:
   - Reduce the motor stator resistance stepwise in C00084.

   If motor current < motor magnetising current:
   - Increase the motor stator resistance stepwise in C00084.
6. Repeat steps 2 ... 5 until the following applies: Motor current ≈ motor magnetising current.
7. Save parameter set (C00002 = "11: Save start parameters").

5.5.2.3 Mutual motor inductance

Adjustment at rated operation

This adjustment is executed at rated speed and a defined load (e.g. measuring brake) which serves to define the rated torque. A condition for the adjustment is to know the real load torque. The motor current is compared to the rated current. At rated load, these two values should be almost identical.

If an adjustment at rated operation is not possible, alternatively execute the adjustment at no-load operation (see the following section "Adjustment at no-load operation").

How to adjust the mutual motor inductance at rated operation:

1. Set the maximum current in C00022 to 110 % of the rated motor current (C00088).
2. Select setpoint speed 0 rpm
3. Enable controller.
4. Slowly increase setpoint speed to rated speed (no field weakening) and then keep it at rated speed constantly.
5. Apply rated load to the motor.
6. Compare the motor current displayed in C00054 with the rated motor current displayed in C00088.
7. If motor current > rated motor current:
   - Reduce the mutual motor inductance stepwise and indirectly via the Lh adjustment in C02861 until the following applies: motor current ≈ rated motor current.

   If motor current < rated motor current:
   - Increase the mutual motor inductance stepwise and indirectly via the Lh adjustment in C02861 until the following applies: motor current ≈ rated motor current.
8. Unload the motor again and slowly reduce the setpoint speed to 0 rpm again.
9. Inhibit controller.
10. Save parameter set (C00002 = "11: Save start parameters").

Adjustment at no-load operation

If an adjustment at rated operation is not possible, alternatively execute the adjustment at no-load operation.

For the adjustment of the mutual motor inductance in no-load operation, first the motor current is compared to the motor magnetising current at a setpoint speed of approx. 75 % of the rated speed (without load of the motor). Afterwards the setting of the mutual motor inductance is changed step by step until the motor current just, but stably reaches the motor magnetising current.

How to adjust the mutual motor inductance at no-load operation:

1. Select setpoint speed 0 rpm
2. Enable controller.
3. Slowly increase the setpoint speed to approx. 75 % of the rated speed and keep this value constant.
   - If the controller oscillates, check the speed controller.
4. Compare the motor current displayed in C00054 with the motor magnetising current displayed in C00092.
5. If motor current > motor magnetising current:
   - Reduce the mutual motor inductance stepwise and indirectly via the Lh adjustment in C02861 (based on 100 %) until the following applies: motor current < motor magnetising current

   If motor current << motor magnetising current:
   - Increase the mutual motor inductance stepwise and indirectly via the Lh adjustment in C02861 (based on 100 %) until the following just applies: motor current < motor magnetising current
6. Slowly reduce the setpoint speed to 0 rpm again.
7. Inhibit controller.
8. Save parameter set (C00002 = "11: Save start parameters").

5.5.2.4 Motor rotor resistance

This adjustment is carried out at a setpoint speed of approx. 75 % of the rated speed and with a defined load (e. g. measuring brake). The precondition for the adjustment is that the actual speed is known (e. g. by the use of a manual tachometer). At constant setpoint speed first the actual speed is measured while the machine is unloaded. Afterwards the motor is loaded at the same setpoint speed until the rated torque is reached (rated current). The speed measured should preferably correspond in idle state and at rated speed.

How to adjust the motor rotor resistance:

1. Select setpoint speed 0 rpm
2. Enable controller.
3. Slowly increase the setpoint speed to approx. 75 % of the rated speed and keep this value constant.
4. Measure actual speed n_{Idle} (e.g. using a manual tachometer) and take it down.
5. Increase load of the motor until the motor current displayed in C00054 corresponds to the rated current.
6. Measure actual speed n_{Load}.
7. If $n_{Load} < n_{Idle}$:
   - Reduce motor rotor resistance stepwise and indirectly via the Rr adjustment in C02860 until the following applies: $n_{Load} \approx n_{Idle}$.

   If $n_{Load} > n_{Idle}$:
   - Increase motor rotor resistance stepwise and indirectly via the Rr adjustment in C02860 until the following applies: $n_{Load} \approx n_{Idle}$.
8. Unload the motor again and slowly reduce the setpoint speed to 0 rpm again.
9. Inhibit controller.
10. Save parameter set (C00002 = "11: Save start parameters").

5.5.3 Optimise control mode

A manual optimisation of the controller settings may be required for very dynamic applications and in the field weakening range.

Note!

The processes for optimising the controller settings described in the following subchapters can only be executed while the drive is rotating and never when being at standstill!

For all optimisation processes the magnetisation phase has to be completed!

Based on the typical controller settings which are listed in the chapter "Parameterising speed and torque controller" in table [5-1], first the field feedforward control and the speed controller are optimised in the base speed range. Afterwards, the torque controller is optimised in the field weakening range.

For optimisation, a suitable speed ramp must be selected for the drive and the acceleration must be recorded, e.g. using the Oscilloscope function in the »Engineer«. (📖 590)

Short overview: Parameters for controller settings

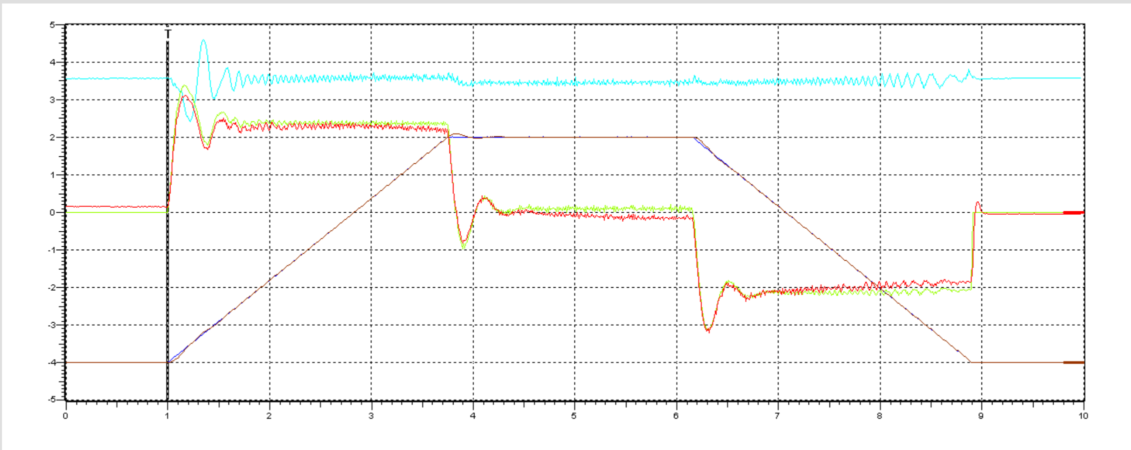
Parameter	Information	Lenze setting	
		Value	Unit
C00070	Speed controller gain	0.500	Nm/rpm
C00071	Speed controller reset time	24.0	ms
C00985	SLVC: Field controller gain	0.00	
C00986	SLVC: Cross current controller gain	0.00	
C00987	SLVC: Torque controller gain	0.5000	Hz/A
C00988	SLVC: Torque controller reset time	10.00	ms

5.5.3.1 Optimising field feedforward control

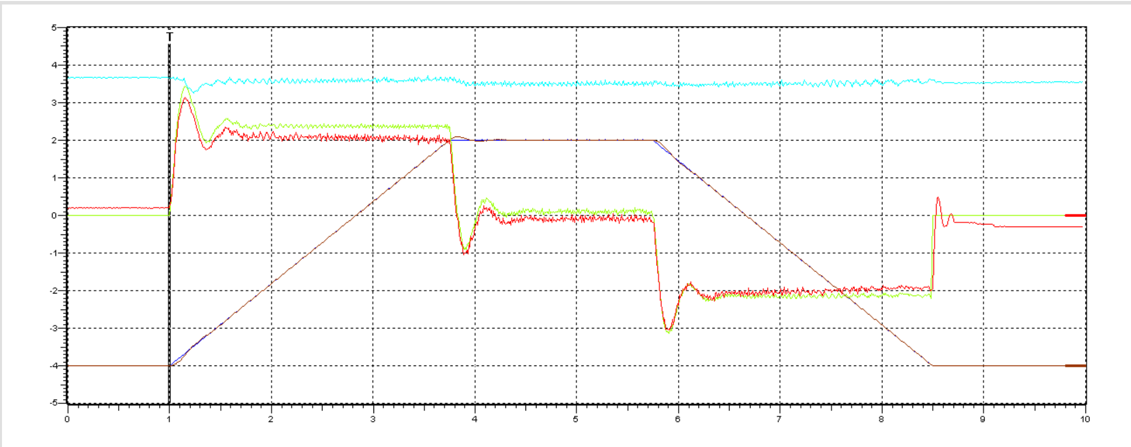
To optimise the field feedforward control, the drive must be accelerated in the base speed range with slow setpoint ramps (e.g. acceleration time = 5 s) to speed values below the rated speed and then decelerated again.

- If the field current oscillates at the beginning of the acceleration and at the end of the deceleration (Current.dnActualDirectCurrent), these oscillations can be reduced by increasing the gain for the field current controller in C00985.

	Ch	Variable of the motor control	Unit	1/div	Offset	Position
	1	Speed.dnSpeedSetpoint (speed setpoint)	rpm	0.2k	0	-4
	2	Speed.dnActualMotorSpeed (current speed)	rpm	0.2k	0	-4
	3	Current.dnActualDirectCurrent (actual field current)	A	10	0	0
	4	Current.dnQuadratureCurrentSet (cross current setpoint)	A	10	0	0
	5	Current.dnActualQuadratureCurrent (actual cross current)	A	10	0	0



[5-8] Oscillogram 1: speed ramp (motor 55 kw) – field controller gain = 0.00



[5-9] Oscillogram 2: speed ramp (motor 55 kw) – field controller gain = 2.00

5.5.3.2 Optimising the speed controller

To optimise the speed controller, the drive must be accelerated in the base speed range with slow setpoint ramps (e.g. acceleration time = 5 s) to speed values below the rated speed and then decelerated again.

Gain optimisation

The proportional gain Vp is selected under C00070:

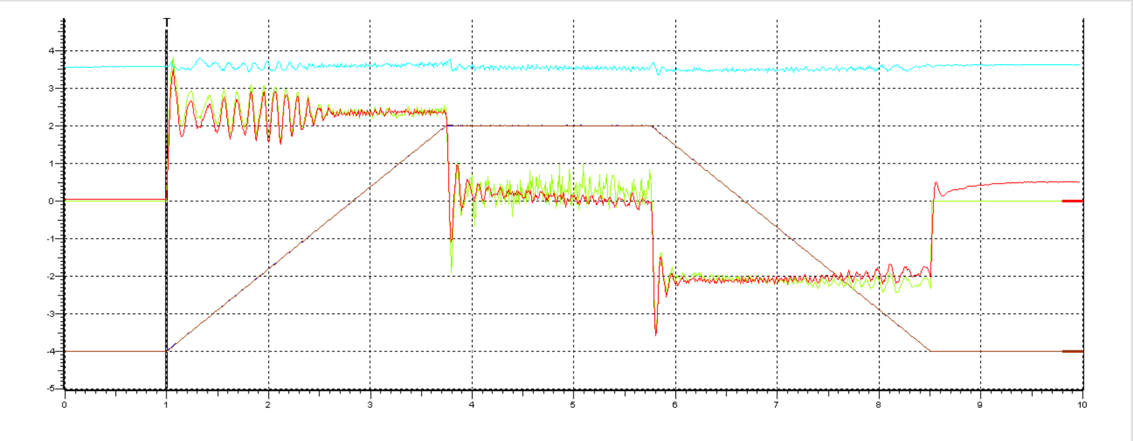
1. Increase C00070 until the drive oscillates slightly (see picture [5-10]).
2. Reduce C00070 until the drive runs stable again (see picture [5-11]).
3. Reduce C00070 to approx. half the value.

Optimise the reset time

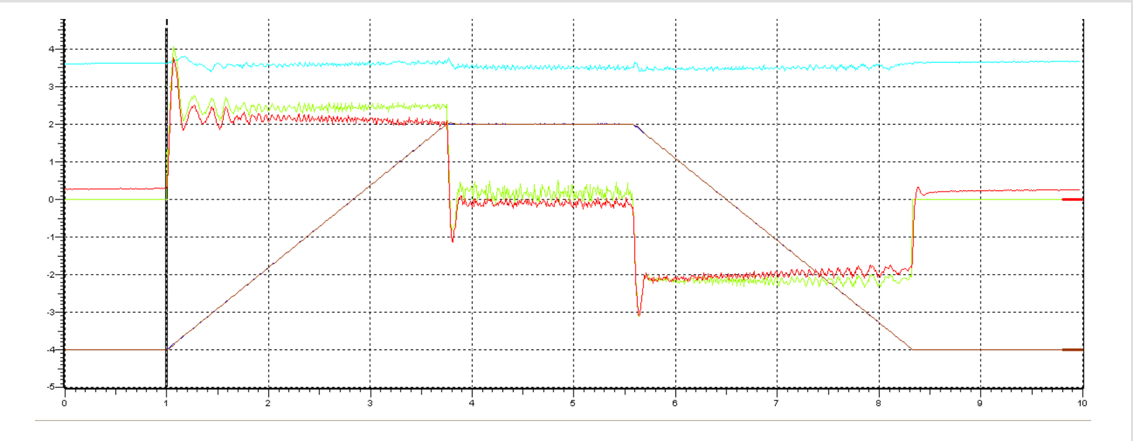
The reset time is selected under C00071:

1. Reduce C00071 until the drive oscillates slightly.
2. Increase C00071 until the drive is stable again.
3. Increase C00071 to approx. double the value.

	Ch	Variable of the motor control	Unit	1/div	Offset	Position
	1	Speed.dnSpeedSetpoint (speed setpoint)	rpm	0.2k	0	-4
	2	Speed.dnActualMotorSpeed (current speed)	rpm	0.2k	0	-4
	3	Current.dnActualDirectCurrent (actual field current)	A	10	0	0
	4	Current.dnQuadratureCurrentSet (cross current setpoint)	A	10	0	0
	5	Current.dnActualQuadratureCurrent (actual cross current)	A	10	0	0



[5-10] Oscillogram 1: speed ramp (motor 55 kw) – speed controller gain = 15.49



[5-11] Oscillogram 2: speed ramp (motor 55 kw) – speed controller gain = 7.49

Setting of actual speed filter

In order to maximise the dynamics of the speed control loop, the actual speed filter should be operated with a time constant as low as possible (C00497). The lower the time constant the higher the gain of the speed controller. Since actual value filters have the task to dampen measuring errors or interference components, it must be found a compromise between filter task and the resulting delay.

If a Lenze motor is selected from the motor catalogue, a time constant is automatically preset in C00497 which serves to operate the motor even with a faulty detection (e.g. in case of a bad shield connection).

When using EMC-compliant systems or high-quality encoders, you can reduce the preset time constant considerably. For this purpose, the running noise of the motor can be used for setting C00497 at constant speed.

If this is not possible, e.g. due to a too loud environment or because the motor is too far away, the noise of the actual speed value or the setpoint torque value can be used for evaluation by means of the Oscilloscope. Please observe that the speed controller gain Vp (C00070) in is used for the torque setpoint.

5.5.3.3 Optimising torque controller

To optimise the torque controller a steep speed ramp is required which reaches into the field weakening range (e.g. 1.2 * rated speed). For this purpose, the drive must be operated at its current and voltage limit.

Stop!

Reduce the maximum current in C00022 for this adjustment to approx. 130 % of the motor magnetising current (C00092) to prevent shocks on the drive!

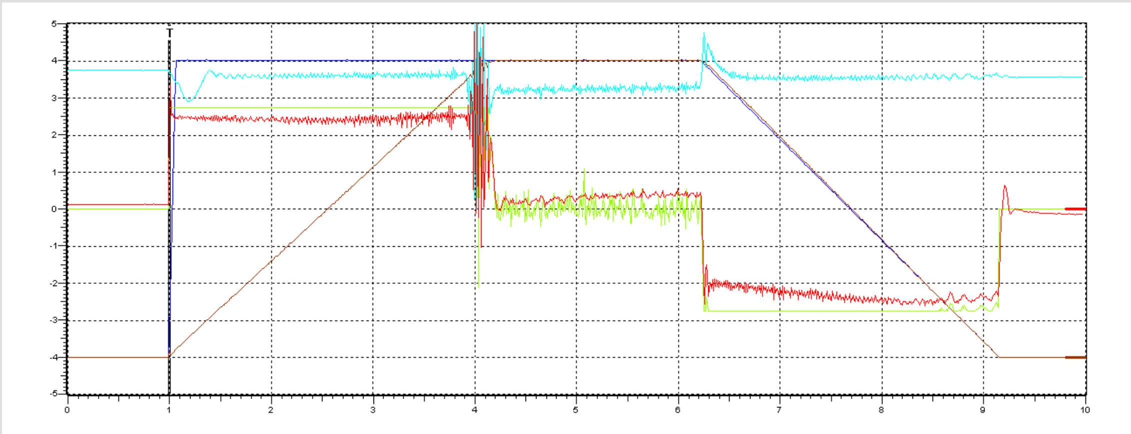
Note!

If no field weakening operation is required, the adjustment must be executed in the base speed range.

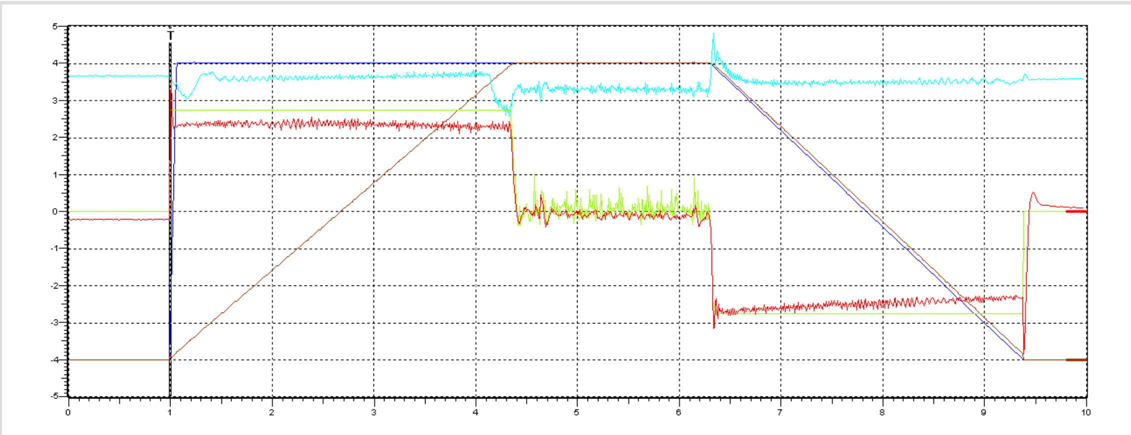
The gain (C00987) and reset time (C00988) of the torque controller are to be set so that the actual cross current can preferably follow the cross current setpoint.

- If oscillations occur during the cross current (see illustration [5-12]), the gain (C00987) is to be reduced until the drive is stable again (see illustration [5-13]).
- Afterwards the reset time (C00988) can be reduced as long as the drive accelerates in a stable way.

	Ch	Variable of the motor control	Unit	1/div	Offset	Position
	1	Speed.dnSpeedSetpoint (speed setpoint)	rpm	0.2k	0	-4
	2	Speed.dnActualMotorSpeed (current speed)	rpm	0.2k	0	-4
	3	Current.dnActualDirectCurrent (actual field current)	A	10	0	0
	4	Current.dnQuadratureCurrentSet (cross current setpoint)	A	10	0	0
	5	Current.dnActualQuadratureCurrent (actual cross current)	A	10	0	0



[5-12] Oscillogram 1: speed ramp (motor 55 kw) – torque controller gain = 0.0661



[5-13] Oscillogram 2: speed ramp (motor 55 kw) – torque controller gain = 0.0361

5.5.3.4 Optimise current controller

Note!

Only required for sensorless vector control if one of the following functions is used.

- Flying restart function (218)
- DC-injection braking (221)

In a test mode you can select current setpoint step-changes and optimise the parameter settings of the current controller (gain and reset time) by evaluating the step responses.

▶ The starting values for gain and reset time can be calculated with the following formula:

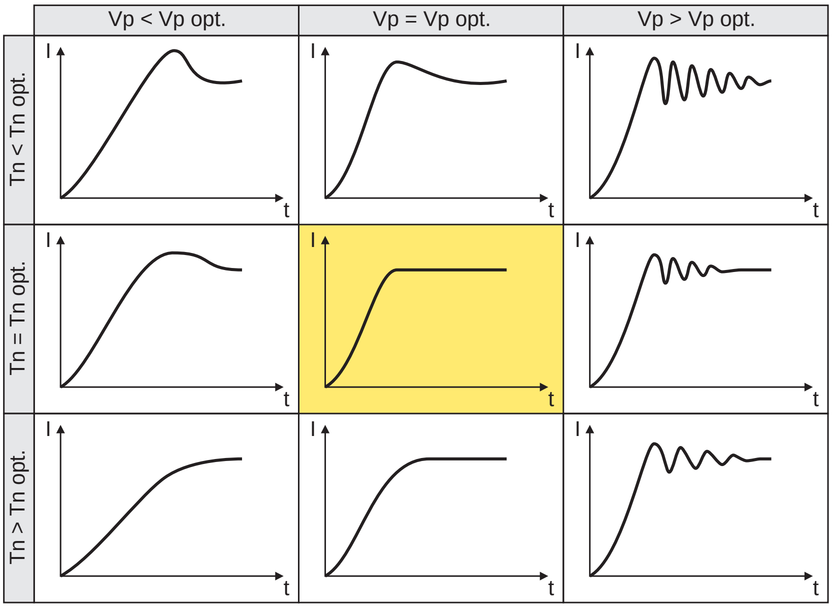
$$\text{Gain} = \frac{\text{Stator leakage inductance}}{340\ \mu s}$$

$$\text{Reset time} = \frac{\text{Stator leakage inductance}}{\text{Stator resistance}}$$

How to optimise the current controller in the test mode:

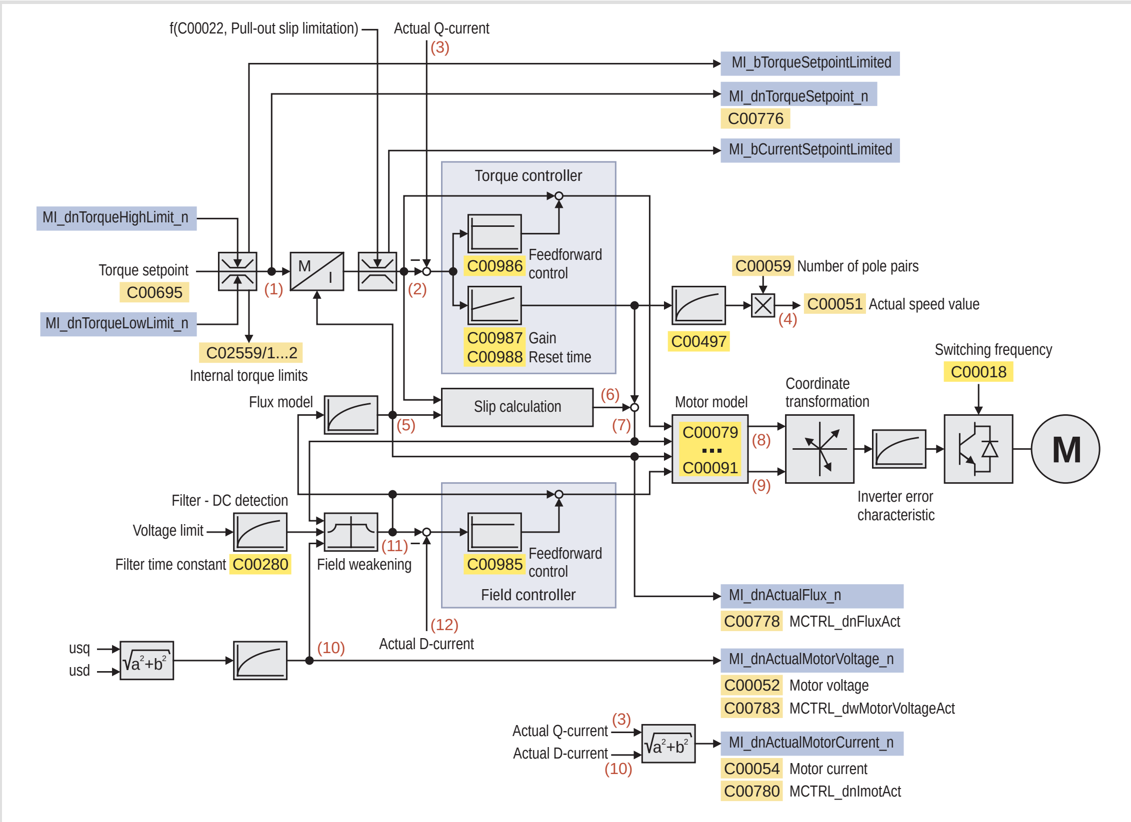
1. If the controller is enabled, inhibit the controller, e. g. with the device command C00002 = "41: Inhibit controller".
2. Activate one of the two following optimisation modes for the current controller:
   - C00398 = "3: Current controller optimisation mode":
     After controller enable, the motor is supplied with current as long as the controller is enabled.
   - From software version V7.0:
     C00398 = "4: Current controller optimisation mode pulse":
     The motor is supplied with voltage for 50 ms after controller enable. Due to this time limit, the load of the machine is reduced. Afterwards, the controller is inhibited automatically.
3. Select the effective value of the current setpoint step change under C00022.
   - The peak value of the measurable motor current will be 1.41 times higher.
4. Enable the controller for a short time and measure the step response of the motor current in the motor phases using the oscilloscope and clamp-on ammeters or record the field-oriented direct-axis current using the Oscilloscope function in the »Engineer«. (590)
   - Variable of the motor control to be recorded:
     Current.dnActualDirectCurrent (field-oriented direct-axis current)

5. Evaluate the step response:



6. Change the gain Vp under C00075 and the reset time Tn under C00076.
7. Repeat steps 4 ... 6 until the optimum step response of the motor current is reached.
   - In the optimised state the current rise time typically is 0.5 ... 1 ms.
8. After the optimisation has been completed, deactivate the test mode again (C00398 = "0: Test mode deactivated").
9. Save parameter set (C00002 = "11: Save start parameters").

5.5.4 Signal flow



[5-14] Signal flow - sensorless vector control

Internal variables of the motor control (oscilloscope signals)

▶ The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (📖 590)

No.	Variable of the motor control	Meaning
(1)	Torque.dnTorqueSetpoint	Torque setpoint
(2)	Current.dnQuadratureCurrentSet	Q current setpoint
(3)	Current.dnActualQuadratureCurrent	Actual Q current
(4)	Speed.dnActualMotorSpeed	Actual speed
(5)	Common.dnActualFlux	Actual flux value
(6)	Frequency.dnActualSlipFrequency	Actual slip frequency
(7)	Frequency.dnActualRotatingFieldFrequency	Actual field frequency
(8)	Voltage.dnQuadratureVoltage	Q voltage
(9)	Voltage.dnDirectVoltage	D voltage
(10)	Voltage.dnActualMotorVoltage	Current motor voltage
(11)	Current.dnDirectCurrentSet	D current setpoint
(12)	Current.dnActualDirectCurrent	Actual D current

5.6 V/f control (VFCplus)

This function extension is available from software version V3.0!

If this motor control mode is set in C00006, the output voltage of the controller follows a firmly defined characteristic.

Note!

The operation of vertical drives/hoists is only supported up to 55 kW by the V/f control!

5.6.1 Basic settings

After the motor and controller have been optimally adjusted to each other, the "initial commissioning steps" described in the following table 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. (📖 191)
2.	Setting the voltage boost. (📖 192)
3.	Parameterising load adjustment. (📖 194)
4.	Activating voltage vector control. (📖 195) • The *Voltage Vector Control* (VCC), which can be activated, serves to provide a torque at low field frequencies. This task is executed by a current controller the output voltage of which is added to the voltage from the characteristic.
5.	Defining the current limit (Imax controller). (📖 196)
6.	Additional "flying restart" function: • In the Lenze setting, this parameterisable additional function is activated. • If the flying restart function is not required, deactivate this function. ▸ Flying restart function (📖 218) STOP **Only deactivate the flying restart if it is ensured that the drive is always at standstill in the case of controller enable!**
7.	Additional "DC-injection braking" function: • In the Lenze setting, this parameterisable additional function is deactivated. • If DC-injection braking is required, activate this function. ▸ DC-injection braking (📖 221)

Tip!

How to optimise the control behaviour and adjust it to the concrete application is described in the chapter "Optimise control mode". (📖 197)

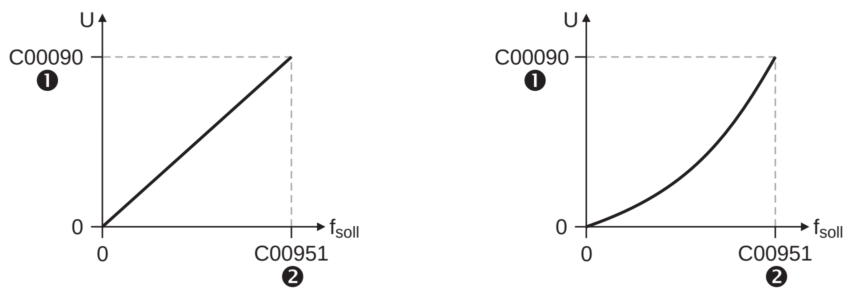
The parameterisable additional functions are described in the chapter of the same name "Parameterisable additional functions". (📖 209)

5.6.1.1 Defining the V/f characteristic

Linear/square-law characteristic

C00950 serves to select the shape of the characteristic to adjust the characteristic to different load profiles:

- Linear characteristic for drives with constant load torque over the speed.
- Square-law characteristic for drives with a linear or square-law load torque over the speed:
  - Square-law V/f characteristics are mostly used in centrifugal pump and fan drives. However, it must be checked in each individual case if your pump or fan drive can be used in this operating mode!
  - If your pump or fan drive is not suitable for operation with a square-law V/f characteristic, you have to use the linear or user-definable V/f characteristic or the sensorless vector control instead of the V/f control.



❶ C00090: Rated motor voltage
❷ C00951: V/f base frequency

[5-15] Linear/square-law V/f characteristic

- The calculation of the characteristic considers the rated motor voltage (C00090) and the V/f base frequency (C00951).

Short overview: Parameters for V/f characteristic

Parameter	Information	Lenze setting Value	Unit
C00950	VFC: V/f characteristic shape	Linear (V/f)	
C00951	VFC: V/f base frequency	50	Hz
C00952/1...11	VFC: Frequency interpol. point n	▸ Free definition of the V/f characteristic (📖 198)	
C00953/1...11	VFC: Voltage interpol. point n		
C00954/1...11	VFC: Activat. interpol. point n		

5.6.1.2 Setting the voltage boost

C00960 and the *MI_dnBoostSet_n* input of the motor interface serve to define a constant, load independent voltage boost at low speeds (below the V/f rated frequency) or at motor standstill to optimise the starting performance.

Stop!

If the motor is operated at standstill for a longer time - especially in case of smaller motors - the motor can be destroyed by overtemperature!

- Connect the thermal contact (NC contact), PTC, or KTY of the motor and activate the motor temperature monitoring of the controller.
- Operate self-ventilated motors with a blower, if required.

Note!

When device types > BF7 are used, the voltage boost only functions in a restricted way due to the hardware properties!

Depending on the required starting torque, the voltage boost must be set so that the required motor current will be available after controller enable (starting current ~ V_{min}).

Tip!

The required voltage can be calculated by multiplying the stator resistance by the rated magnetising current:

$$U_{min} = R_S \cdot I_{mN}$$

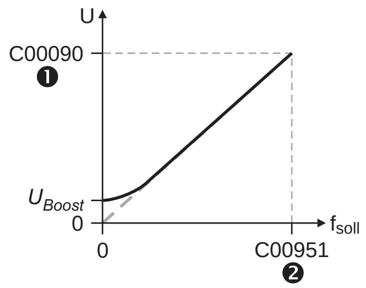
Optionally, the voltage can be determined empirically by increasing the value for the voltage boost until the rated magnetising current flows.

- ▶ Via the input *MI_dnBoostSet_n* the voltage has to be selected in [%] relating to the rated motor voltage (C00090).
- ▶ In C00960, however, the voltage must be set directly in [V].
- ▶ Only positive voltage values can be selected, negative values are limited to 0 V.
- ▶ Both selections are added:

$$U_{Boost} = MI\_dnBoostSet\_n \cdot \frac{C00090}{100\,\%} + C00960$$

- ▶ The resulting voltage V_{Boost} is added geometrically to the characteristic voltage:

$$U = \sqrt{U^2_{Characteristic} + U^2_{Boost}}$$



❶ C00090: Rated motor voltage
❷ C00951: V/f base frequency

[5-16] Voltage boost at linear/square-law V/f characteristic

Short overview: Parameters for voltage boost

Parameter	Information	Lenze setting	
		Value	Unit
C00960	VFC: V/f voltage boost	0	V

Tip!

For magnetising the motor, consider a sufficient time from the controller enable to the start of the speed ramp function generator.

- The bigger the motor the longer the time required for magnetisation. A motor with a power of 90 kW requires up to 2 seconds.

5.6.1.3 Parameterising load adjustment

C00962 serves to parameterise a load adjustment in [%] proportionally to the rated torque to obtain a correspondingly "rigid "drive behaviour even after the starting action.

- When starting torque = rated torque, a load adjustment of 50 % is suitable for most applications.

Stop!

If the load adjustment is too high, the motor current may increase in idle state and the motor may overheat!

The C00961 parameter serves to adjust the characteristic depending on the load at CW and CCW rotation:

Setting in C00961	Information
0: CW rotation in motor mode/CCW rotation in motor mode	The motor operates in motor mode in both directions.
1: CW rotation in motor mode/CCW rotation in generator mode	Application example: Hoist without counterweight
2: CW rotation in generator mode/CCW rotation in motor mode	Application example: Dancer-controlled unwinder

Short overview: Parameters for load adjustment

Parameter	Information	Lenze setting	
		Value	Unit
C00961	VFC: Load - cw/ccw-operation	CW: mot. / CCW: mot.	
C00962	VFC: Load adjustment	20.00	%

5.6.1.4 Activating voltage vector control

The *Voltage Vector Control* (VCC), which can be activated, is an alternative to the voltage boost. The voltage vector control is used if a high starting torque has to be ensured. The voltage vector control ensures that the motor current required for this purpose is available in the zero speed range.

Note!

A disadvantage of the voltage vector control is the increased current at low speeds. This causes higher losses and thus an increased heating of the machine.

The voltage vector control is activated by defining a current setpoint in C00957 and can be deactivated again by setting "0.0 A".

- The voltage vector control is additive to the voltage boost.
  ▶ Setting the voltage boost (📖 192)
- When the current setpoint is defined, provide a reserve of 20 % to prevent a motor stalling caused by sudden additional loads.
- Example for starting torque = rated motor torque:
  The current setpoint must be parameterised in C00957 to approx. 120 % of the load current.

Setting of the controller parameters

For the gain (C00958) and the reset time (C00959), accept the values that have been detected for the current controller gain (C00075) and reset time (C00076) in the test mode.
▶ Optimise current controller (📖 202)

Since the voltage vector control controls the current value which has a higher background noise due to its calculation, the reset time might possibly be increased.

Short overview: Parameters for voltage vector control

Parameter	Information	Lenze setting	
		Value	Unit
C00957	VFC: VVC current setpoint	0.00	A
C00958	VFC: VVC gain	0.00	V/A
C00959	VFC: VVC reset time	2000.00	ms

Tip!

For controllers with a power > 55 kW we recommend to use the voltage vector control for horizontal drives for improving the smooth running characteristics.

5.6.1.5 Defining the current limit (Imax controller)

The current limit for the Imax controller is defined by the maximum current which must be set in C00022. If the motor current exceeds the value set in C00022, the Imax controller gets active.

- The Imax controller changes the field frequency so that the motor current does not exceed the current limit. In motor mode, the frequency is reduced and in generator mode it is increased.
- Gain and reset time of the Imax controller can be parameterised in C00963 and C00964.

Short overview: Parameters for Imax controller

Parameter	Information	Lenze setting	
		Value	Unit
C00022	Maximum current	0.00	A
C00963	VFC: Gain - Imax controller	0.001	Hz/A
C00964	VFC: Reset time - Imax controller	100.0	ms

Optimise the Imax controller

- If oscillations occur during operation at the current limit, the Imax controller has to be decelerated:
  – Reduce gain (C00963)
  – Increase reset time (C00964)
- If the Imax controller does not operate fast enough after having exceeded the current limit, it must be accelerated:
  – Increase gain (C00963)
  – Reduce reset time (C00964).

5.6.2 Optimise control mode

The "optimisation steps" given in the following table serve to further optimise the control behaviour of the V/f control and adjust it to the concrete application.

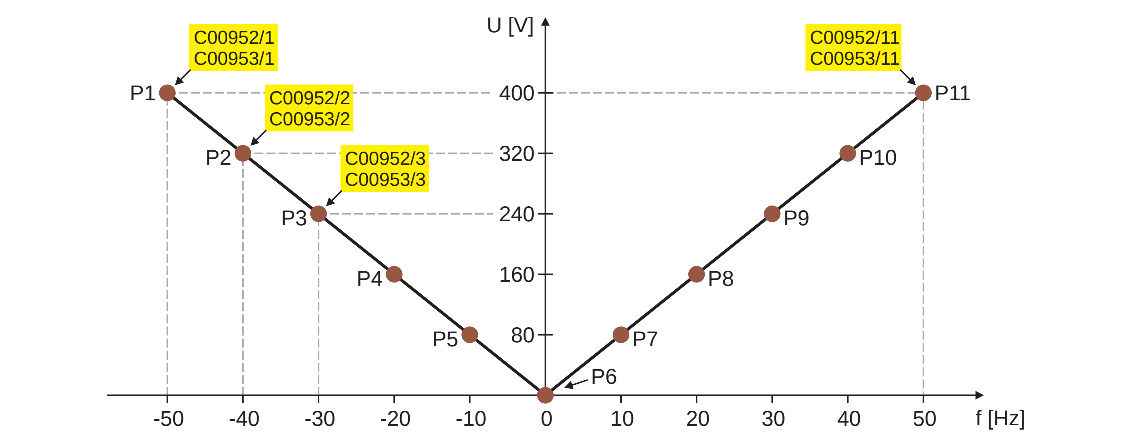
▶ Detailed information on the individual steps can be found in the following subchapters.

Optimisation steps	
1.	Free definition of the V/f characteristic. (📖 198) • Individual adjustment of the motor magnetisation to the concrete application if linear and square-law characteristics are not suitable.
2.	Parameterising slip compensation. (📖 199)
3.	Parameterising oscillation damping. (📖 200)
4.	When the flying restart function is used: Optimise flying restart process. ▸ Flying restart function (📖 218)
5.	Optimise current controller. (📖 202) • Only required if one of the following functions is used: – Voltage vector control (📖 195) – Flying restart function (📖 218) – DC-injection braking (📖 221)
6.	Save »Engineer« project.

5.6.2.1 Free definition of the V/f characteristic

To individually adjust the motor magnetisation to the real application, a user-definable characteristic can be selected in C00950 if the linear and square-law characteristic are not suitable.

- The interpolation points (voltage/frequency values) of this characteristic are selected via the 11 subcodes of C00952 and C00953.
- If less interpolation points are required, the interpolation points that are not needed have to be deactivated via the subcodes of C00954.
- In the Lenze setting the 11 interpolation points represent a linear characteristic:



	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

[5-17] User-definable characteristic (Lenze setting)

5.6.2.2 Parameterising slip compensation

The slip compensation serves to automatically compensate a load-dependent speed loss. In order that the slip compensation can operate correctly, the rated slip of the motor is required. This is calculated from the rated frequency (C00089) and the rated speed (C00087), thus both parameters must be parameterised correctly.

- A percentage adjustment of the calculated slip can be made in C00965, e.g. when the real motor data deviate from the data given on the nameplate. A value of 100 % in C00965 corresponds to the rated slip of the machine.
- The time behaviour of the slip compensation can be parameterised in C00966.

Short overview: Parameters for slip compensation

Parameter	Information	Lenze setting	
		Value	Unit
C00965	VFC: Gain - slip compensation	0.00	%
C00966	VFC: Time constant - slip compens.	2000	ms

5.6.2.3 Parameterising oscillation damping

The oscillation damping serves to reduce the oscillations during no-load operation which are caused by energy oscillating between the mechanical system (mass inertia) and the electrical system (DC bus). Furthermore, the oscillation damping can also be used to compensate resonances.

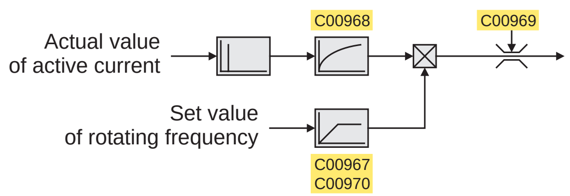
Note!

Observe the following restrictions:
- Oscillations occurring sporadically cannot be damped.
- Damping is possible only for constant oscillations at a steady-state operating point.
- The oscillation damping is not suitable for oscillations occurring during dynamic processes (e.g. accelerations or load changes).

Function

The determination of the oscillation is based on the active current. In order to obtain the alternating component of the active current, this current is differentiated. This signal is then passed through a PT1 filter.

- The base frequency of the PT1 filter has to be set in such a way that the oscillation can be damped and higher-frequency components are filtered out of the signal. For this purpose the time constant (C00968) is used.
- C00967 serves to parameterise the gain of the oscillation signal. The maximum amplitude of the frequency change determined by the oscillation damping is set via C00969.
- Oscillation damping is only active if the setpoint speed is greater than 10 rpm and the DC-bus voltage exceeds a value of 100 V.
- In the lower speed range, the oscillation damping may have a negative impact on the concentricity factor.
  - Therefore from software version V5.0 a ramp end frequency can be set in C00970, up to which the gain of the oscillation damping (C00967) from 10 rpm is slowly increased with increasing rotational frequency.



Identification of the oscillation

Before the oscillation damping can be parameterised, the oscillation has to be identified. One way is to examine the motor current while oscillation damping is switched off (C00967 = 0 %). At steady-state operation, a constant current flows. If the drive oscillates, these oscillations are also visible on the motor current. It is therefore possible to determine the frequency and the amplitude of the oscillation from the alternating component of the motor current. In the following, this alternating component is referred to as "current oscillation".

Parameter setting

- The time constant (C00968) is determined from the reciprocal value of twice the frequency of the current oscillation:

$$\text{Time constant} = \frac{1}{2 \cdot \text{Oscillation frequency}}$$

- The gain factor (C00967) is calculated with the following formula based on the relationship between the amplitude of the current oscillation and the maximum device current:

$$\text{Gain} = \frac{\text{Current amplitude}}{\sqrt{2} \cdot \text{Maximum device current (C00789)}} \cdot 100\ \%$$

- The maximum oscillation frequency (C00969) serves to the absolute limitation of the oscillation frequency calculated before it is added to the field frequency. It can be derived from the amplitude of the current oscillation, the rated motor current, and the slip frequency of the motor connected:

$$\text{Max. frequency} = \frac{2 \cdot \text{Amplitude of the current oscillation}}{\text{Rated motor current}} \cdot \text{Rated slip frequency}$$

- From software version V5.0: The ramp end frequency (C00970) defines the rotational frequency from which the gain factor is to have reached its nominal value (C00967).
  - The ramp end frequency refers to the rated motor frequency in percent (C00089).
  - Below a speed of 10 rpm, the oscillation damping remains deactivated.
  - For machines with a power greater than 55 kW a ramp end frequency of 20 % is recommended.

Short overview: Parameters for oscillation damping

Parameter	Information	Lenze setting Value	Unit
C00967	VFC: Gain - oscillation damping	20	%
C00968	VFC: Time const. - oscill. damp.	5	ms
C00969	VFC: Limitation - oscillation damping	0.2	Hz
C00970	VFC: Ramp end freq. - oscillation damping	0	%

5.6.2.4 Optimise current controller

Note!

Only required if one of the following functions is used:

- Voltage vector control (📖 195)
- Flying restart function (📖 218)
- DC-injection braking (📖 221)

In a test mode you can select current setpoint step-changes and optimise the parameter settings of the current controller (gain and reset time) by evaluating the step responses.

▶ The starting values for gain and reset time can be calculated with the following formula:

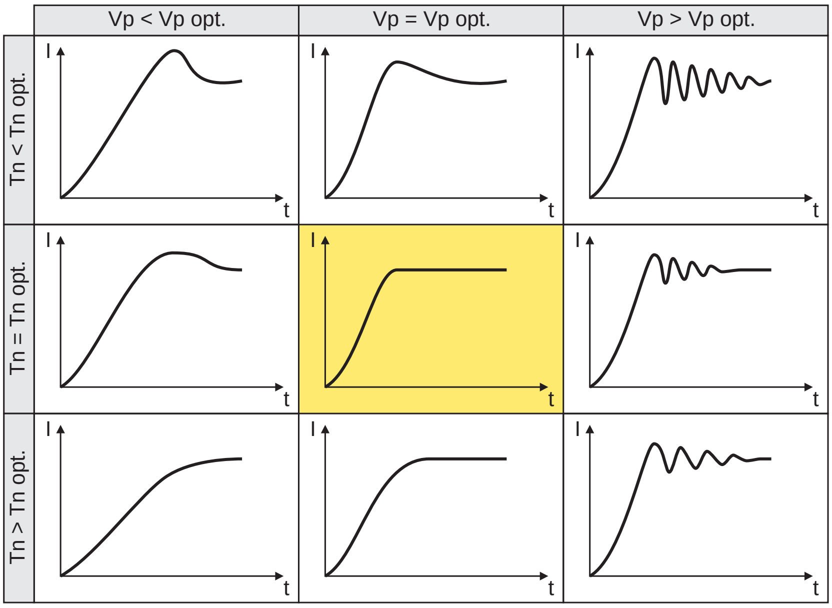
$$\text{Gain} = \frac{\text{Stator leakage inductance}}{340\ \mu\text{s}}$$

$$\text{Reset time} = \frac{\text{Stator leakage inductance}}{\text{Stator resistance}}$$

How to optimise the current controller in the test mode:

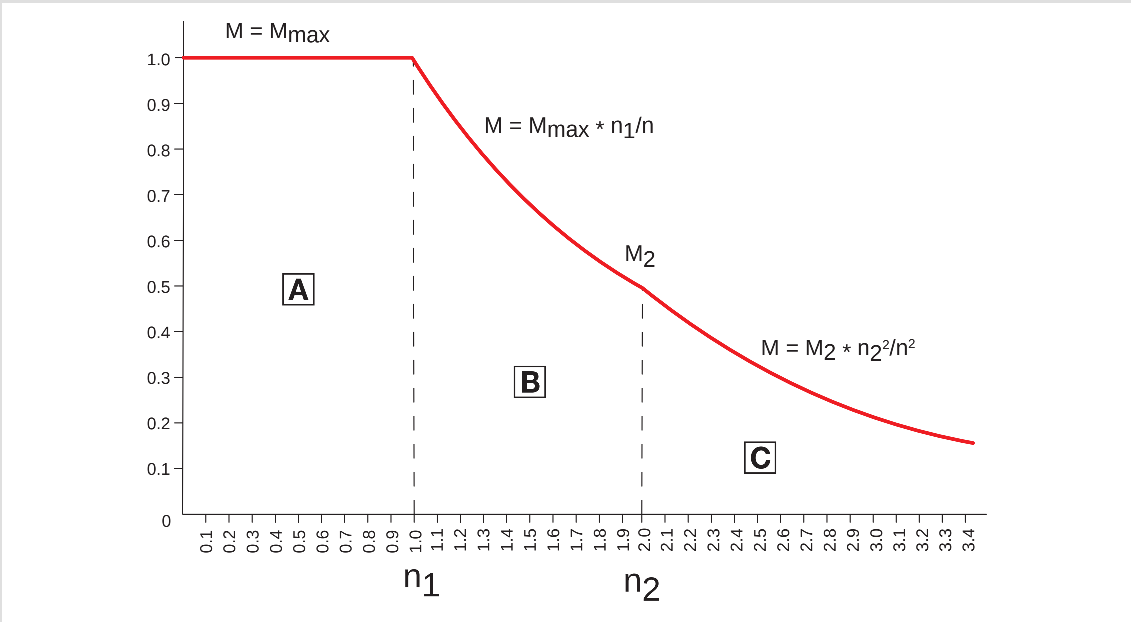
1. If the controller is enabled, inhibit the controller, e. g. with the device command C00002 = "41: Inhibit controller".
2. Activate one of the two following optimisation modes for the current controller:
   - C00398 = "3: Current controller optimisation mode":
     After controller enable, the motor is supplied with current as long as the controller is enabled.
   - From software version V7.0:
     C00398 = "4: Current controller optimisation mode pulse":
     The motor is supplied with voltage for 50 ms after controller enable. Due to this time limit, the load of the machine is reduced. Afterwards, the controller is inhibited automatically.
3. Select the effective value of the current setpoint step change under C00022.
   - The peak value of the measurable motor current will be 1.41 times higher.
4. Enable the controller for a short time and measure the step response of the motor current in the motor phases using the oscilloscope and clamp-on ammeters or record the field-oriented direct-axis current using the Oscilloscope function in the »Engineer«. (📖 590)
   - Variable of the motor control to be recorded:
     Current.dnActualDirectCurrent (field-oriented direct-axis current)

5. Evaluate the step response:



6. Change the gain Vp under C00075 and the reset time Tn under C00076.
7. Repeat steps 4 ... 6 until the optimum step response of the motor current is reached.
   - In the optimised state the current rise time typically is 0.5 ... 1 ms.
8. After the optimisation has been completed, deactivate the test mode again (C00398 = "0: Test mode deactivated").
9. If the Imin control is used, both calculated controller parameters can also be used for the Imin controller:
   - C00075 → C00958 (Imin controller: gain)
   - C00076 → C00959 (Imin controller: reset time)
10. Save parameter set (C00002 = "11: Save start parameters").

5.6.2.5 Optimise pull-out slip limitation



[5-18] Speed/torque curve of the asynchronous motor with two field weakening ranges B and C

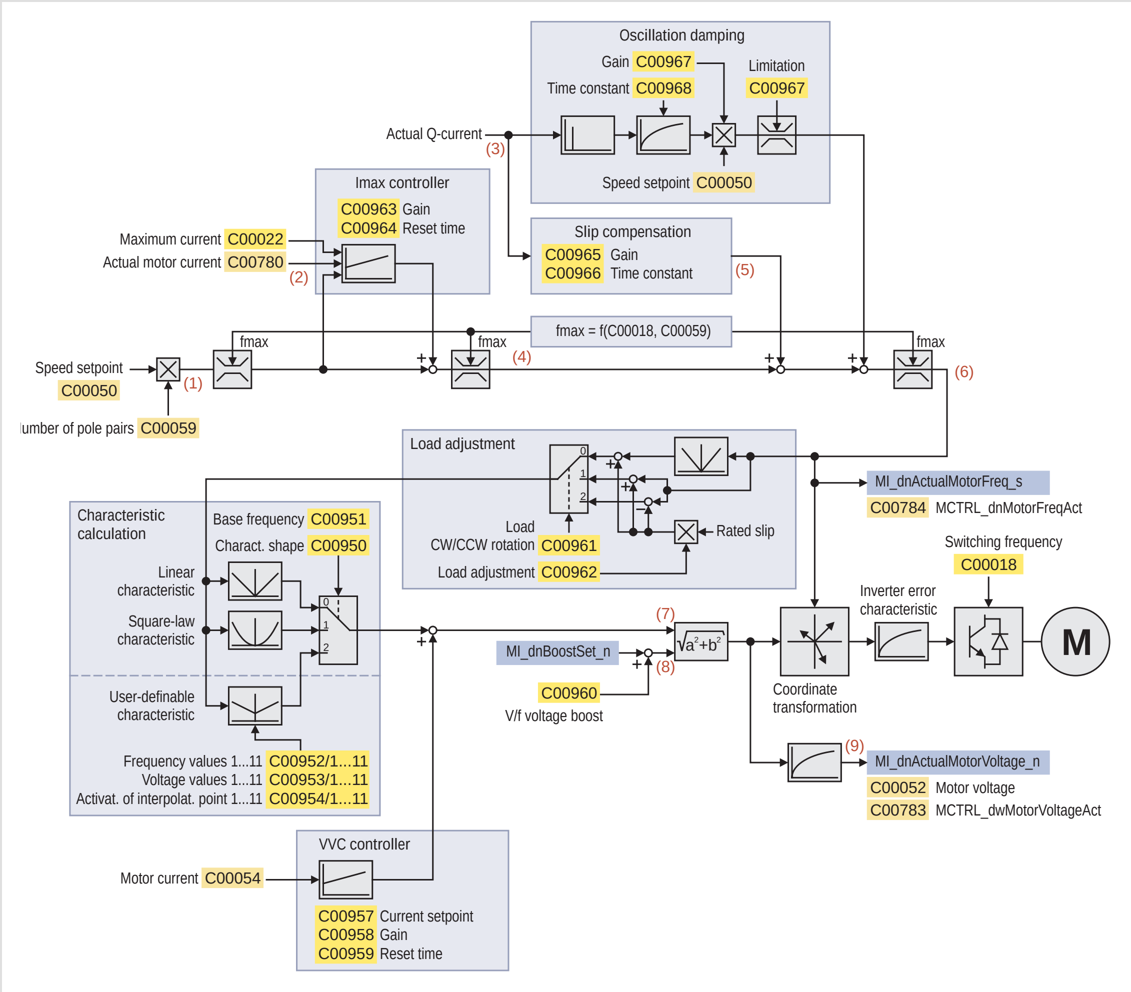
The operating range of an asynchronous motor consists of the voltage control range A and the field weakening range. The field weakening range again is divided into two ranges:

- In the first range B, the power can be kept constant without causing motor stalling.
- The second field weakening range C is characterised by the fact that the maximum permissible stator current (defined via C00022 "Maximum current") is reduced to prevent motor stalling.

The override point (n_2, M_2) can be influenced via C00980 ("VFC: Override point of field weakening"). If the motor stalls in the field weakening range, the override point (n_2, M_2) can be adjusted by decreasing C00980 so that motor stalling is avoided.

If the motor does not provide sufficient torque in the field weakening range, C00980 must be increased.

5.6.3 Signal flow



[5-19] Signal flow - V/f control

Internal variables of the motor control (oscilloscope signals)

▶ The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (📖 590)

No.	Variable of the motor control	Meaning
(1)	Speed.dnSpeedSetpoint	Speed setpoint
(2)	Current.Current.dnActualMotorCurrent	Actual motor current
(3)	Current.dnActualQuadratureCurrent	Actual Q current
(4)	Speed.dnActualMotorSpeed	Current motor speed
(5)	Frequency.dnActualSlipFrequency	Actual slip frequency
(6)	Frequency.dnActualRotatingFieldFrequency	Actual field frequency
(7)	Voltage.dnOutputQuadratureVoltage	Q voltage
(8)	Voltage.dnOutputDirectCurrentCtrl	D voltage
(9)	Voltage.dnActualMotorVoltage	Current motor voltage

5.7 V/f control (VFCplus)

This function extension is available from software version V3.0!

Note!

The descriptions in the chapter "V/f control (VFCplus)" also apply to the V/f control closed loop. (📖 190)

When this motor control mode is used, the operation can be continued if the encoder fails. For this, the encoder monitoring must be parameterised to "Warning". If the encoder fails, the speed controller is "frozen" so that the slip correction via the speed controller is maintained.

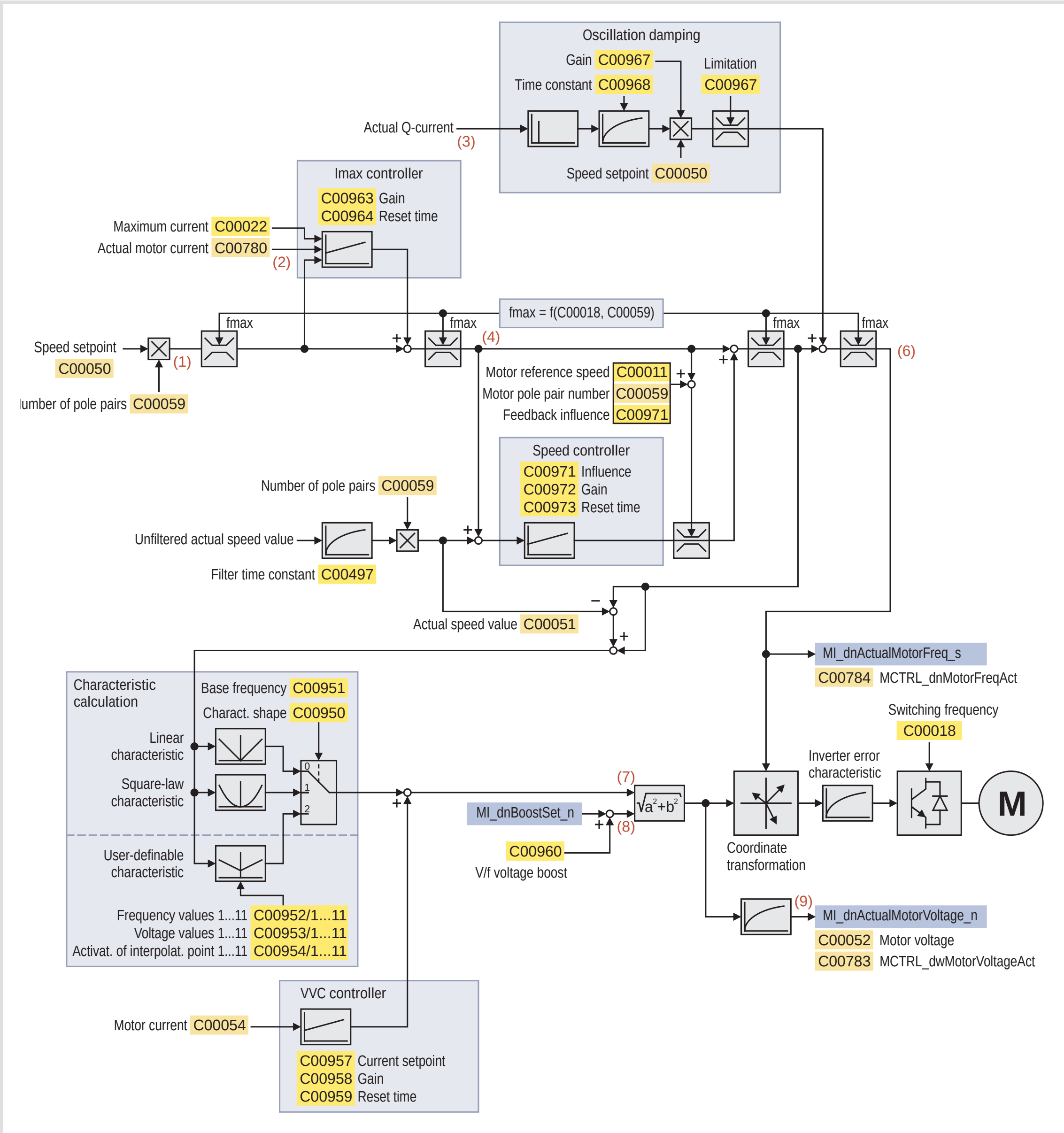
For the closed loop V/f control, the speed controller (also called "slip regulator") also has to be parameterised for the speed feedback.

- C00971 serves to define the influence of the speed controller in [%] with regard to the reference speed of the motor (C00011). If the influence is adjusted to the slip expected under normal operating conditions, the motor cannot accelerate in an uncontrolled way when the encoder fails.
- To activate the speed controller, parameterise the gain (C00972) and the reset time (C00973).

Short overview: Parameters for speed controller

Parameter	Information	Lenze setting	
		Value	Unit
C00971	VFC: Influence - speed controller	10.00	%
C00972	VFC: Gain - speed controller	0.000	Hz/rpm
C00973	VFC: Reset time - speed controller	6000.0	ms

5.7.1 Signal flow



[5-20] Signal flow for closed loop V/f control

Internal variables of the motor control (oscilloscope signals)

- The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (📖 590)

No.	Variable of the motor control	Meaning
(1)	Speed.dnSpeedSetpoint	Speed setpoint
(2)	Current.Current.dnActualMotorCurrent	Actual motor current
(3)	Current.dnActualQuadratureCurrent	Actual Q current
(4)	Speed.dnActualMotorSpeed	Current motor speed
(6)	Frequency.dnActualRotatingFieldFrequency	Actual field frequency
(7)	Voltage.dnOutputQuadratureVoltage	Q voltage
(8)	Voltage.dnOutputDirectCurrentCtrl	D voltage
(9)	Voltage.dnActualMotorVoltage	Current motor voltage

5.8 Parameterisable additional functions

Detailed information on the parameterisable additional functions can be found in the following subchapters:

Parameterisable additional functions	Available from software version	Motor control*		
		SC	SLVC	VFC plus
Correction of the stator leakage inductance... (210)	V1.0	●		
Field weakening for synchronous machines (215)	V2.0	●		
Flying restart function (218)	V3.0		●	●
DC-injection braking (221)	V3.0		●	●

* SC = servo control SLVC = sensorless vector control VFCplus = V/f control

5.8.1 Correction of the stator leakage inductance...

...and the current controller parameters by means of the saturation characteristic

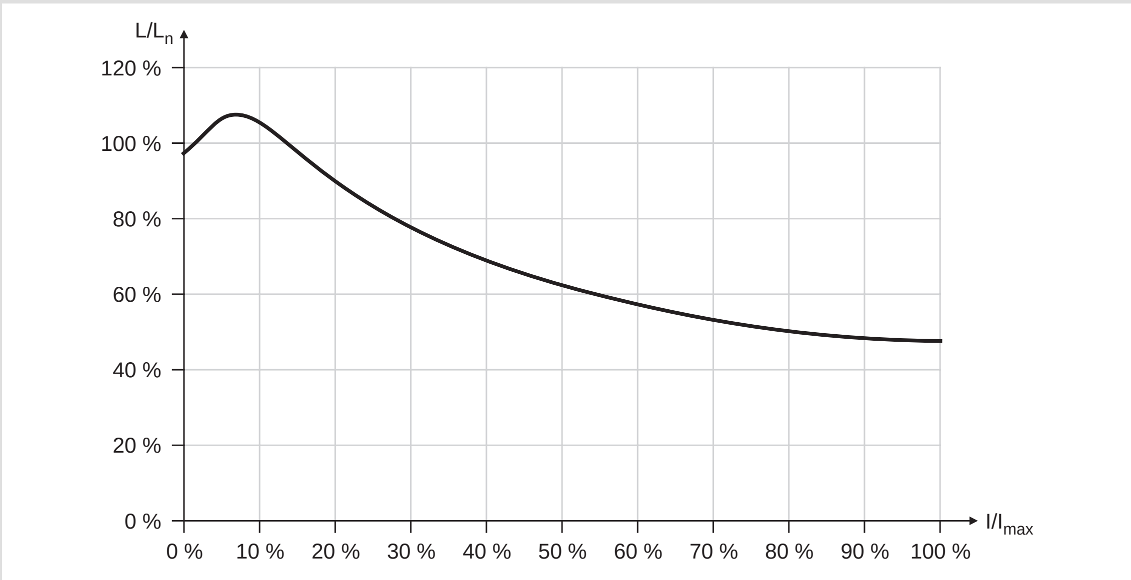
Note!

Function only possible for servo control!

The current controller must be adjusted to the electrical characteristics of the motor stator resistance (C00084) and stator leakage inductance (C00085). In case of modern motors, the stator leakage inductance changes with the height of the current so that a new current controller setting is required for each current height.

When the motor is operated with very low and very high currents (e.g. in *pick and place* applications), it is not always possible to achieve a satisfactory current controller setting for all operating points. For this purpose, the correction of the stator leakage inductance and current controller parameters is now possible via an adjustable saturation characteristic that can be set in C02853 (17 interpolation points).

The following picture shows a typical saturation characteristic of an MCS motor:



[5-21] Saturation characteristic: Inductance referring to the inductance for rated current

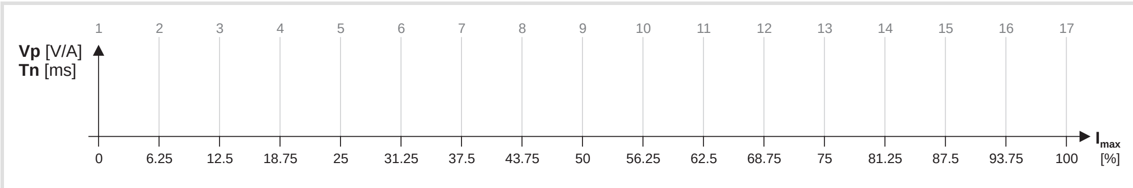
- By optimising the current controller with different current setpoints such a characteristic can be determined "by trial" and set in C02853.
- The correction by means of this saturation characteristic can be switched off or on via C02859.

Note!

The saturation characteristic is not only used for the correction of the current controller but also influences the current controller feedforward control (C00074).

Distribution of the interpolation points

- The saturation characteristic is defined by 17 interpolation points which are distributed linearly on the x axis.
- Interpolation point 17 represents 100 % of the maximum motor current in the process (C02855).



[5-22] Saturation characteristic: Distribution of the interpolation points

Example for determining the saturation characteristic

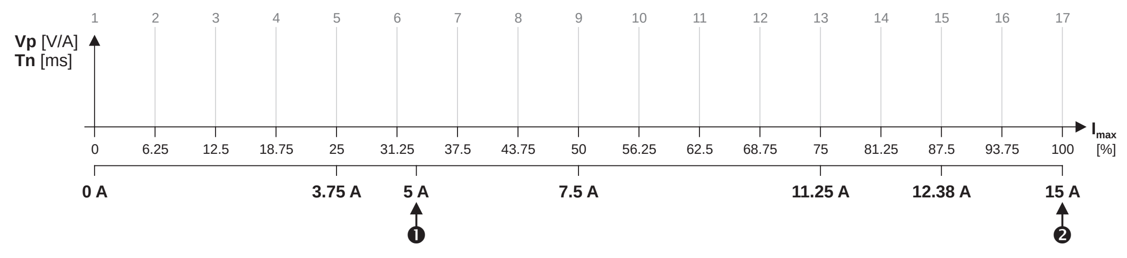
Given values:

- Rated motor current: 5 A
- Maximum motor current: 20 A
- Maximum process current: 15 A (must be set later in C00022)

Procedure:

1. Deactivate correction (C02859 = "OFF").
2. Set the maximum current up to which the motor is to be operated in the process in C02855 (in this example "15 A").
   – The value set in C02855 has to be greater or the same as C00022.
3. Adjust the current controller with different current setpoints and take down the corresponding settings for Vp and Tn.
   – The procedure for the adjustment is described in the chapter "Optimise current controller".
   – The current setpoints that are to be set for the respective adjustment in C00022 result from the scaling of the maximum process current to the x axis of the saturation characteristic.
   – The interpolation points which are required to define the saturation characteristic with a sufficient quality varies from motor to motor and thus has to be determined individually.

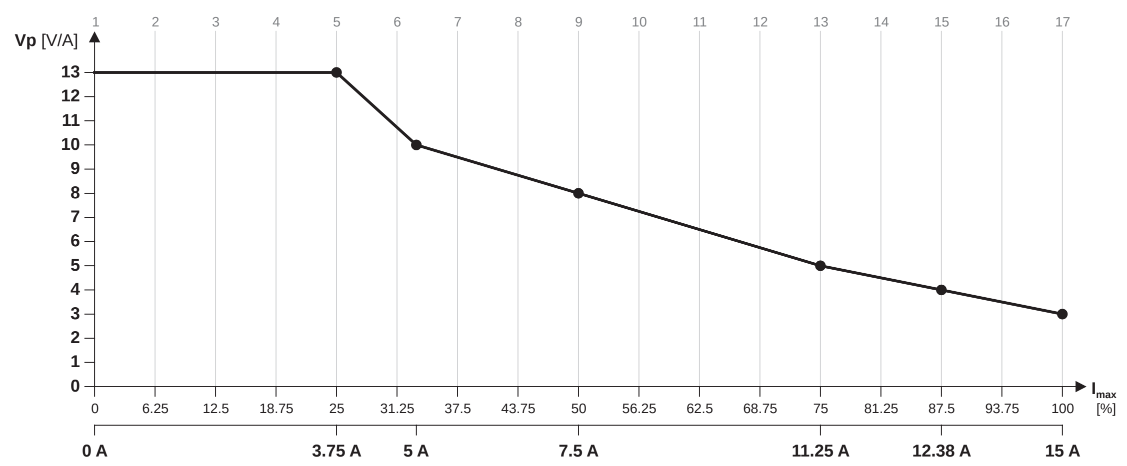
– For this example currents were selected that are part of the interpolation points 5, 9, 13, and 15, and a measurement at rated motor current was carried out:



[5-23] Saturation characteristic: Distribution of the interpolation points

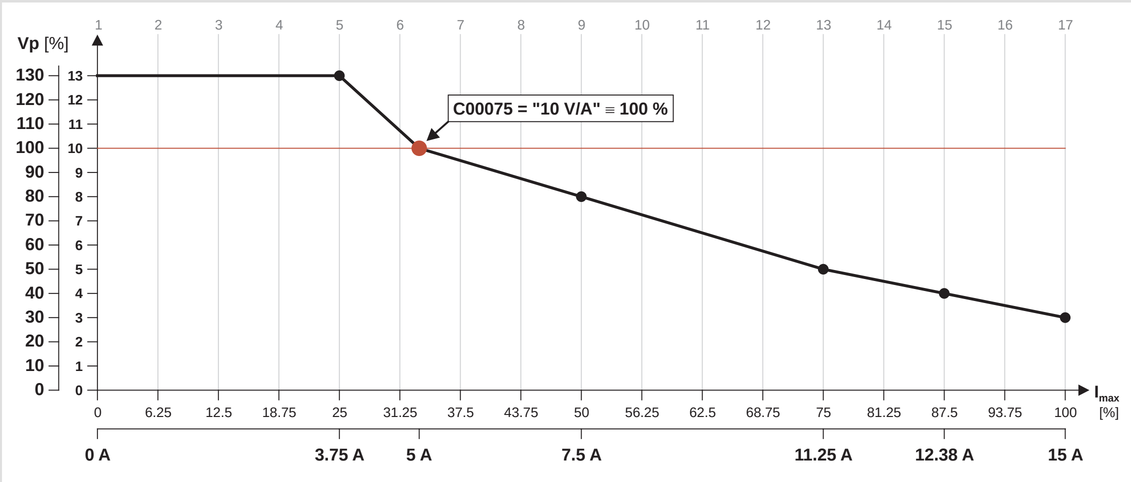
Specifications for adjustment			Measured values	
Interpolation point	Scaling	Setting in C00022	Vp [V/A]	Tn [ms]
5	0.25 * 15 A =	3.75 A	13	6.5
9	0.5 * 15 A =	7.5 A	8	4
13	0.75 * 15 A =	11.25 A	5	2.5
15	0.875 * 15 A =	12.38 A	4	2
17	1.0 * 15 A =	15 A	3	1.7
	Rated motor current =	5 A	10	5

4. Create a characteristic based on the values calculated for Vp.
   – Here, the values which have not been adjusted must be determined by interpolation between two values.
   – **Note:** In this example it was assumed that the inductance does not change considerably below 3.75 A. For this reason the same Vp value resulting from a measurement with a motor current of 3.75 A was used for all interpolation points below 3.75 A.



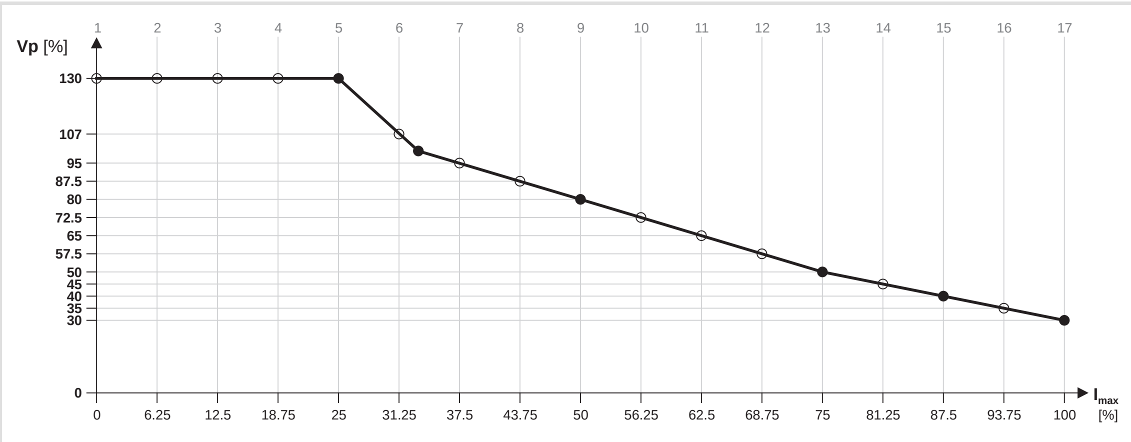
[5-24] Determined saturation characteristic

5. Set the gain Vp in C00075 and the reset time Tn in C00076, which have been determined during the adjustment with rated motor current (in this example "5 A"):
   – Set C00075 = "10 V/A".
   – Set C00076 = "5 ms".
6. Scale the Vp values on the Y axis of the characteristic to the 100 % Vp setting in C00075:



[5-25] Scaling of the determined saturation characteristic to the "100 %" setting in C00075

7. Enter the Vp values in percent, which are placed on the interpolation point, in C02853/1...17:



[5-26] Interpolation point values of the saturation characteristic determined

Interpolation point	Setting	Interpolation point	Setting
1	C02853/1 = 130 %	10	C02853/10 = 72.5 %
2	C02853/2 = 130 %	11	C02853/11 = 65 %
3	C02853/3 = 130 %	12	C02853/12 = 57.5 %
4	C02853/4 = 130 %	13	C02853/13 = 50 %
5	C02853/5 = 130 %	14	C02853/14 = 45 %
6	C02853/6 = 107 %	15	C02853/15 = 40 %
7	C02853/7 = 95 %	16	C02853/16 = 35 %
8	C02853/8 = 87.5 %	17	C02853/17 = 30 %
9	C02853/9 = 80 %		

8. Enter the maximum process current ("15 A") in C00022.
9. Switch on the correction (C02859 = "ON").
   – When the correction of the stator leakage inductance is switched on, the same current characteristic should occur, irrespective of the current magnitude.
   – Since the current controller gain is corrected actively, the step responses may differ slightly compared to the previous measurements. In this case C00075 and C00076 must be optimised one last time.
10. Save parameter set (C00002 = "11: Save start parameters").

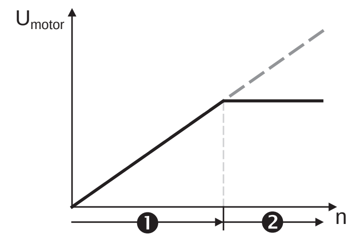
5.8.2 Field weakening for synchronous machines

This function is available from software version V2.0!

Note!

Function only possible for servo control!

If required, the field weakening mode can be switched on in C00093 for synchronous machines.



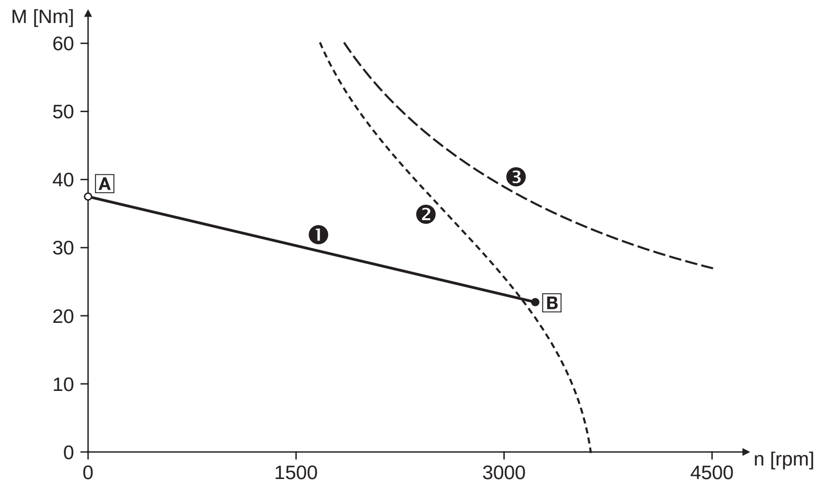
❶ Armature control range
❷ Field weakening range

- If field weakening is switched on and the DC-bus voltage limit is reached, the motor magnetising current is increased via an internal control loop from 0 A to the maximally effective magnetising current.
- The result is that a higher speed can be reached at the same motor voltage or DC-bus voltage.

[5-27] Voltage/speed characteristic with active field weakening

$$n_{max} = n_{rated\_mot} \cdot \frac{800\,V}{\sqrt{2} \cdot U_{rated\_mot}}$$

[5-28] Calculation of the maximum speed that can be reached when field weakening is switched on



A Standstill torque
B Rated torque
❶ Characteristic for continuous operation
❷ Limit characteristic at mains voltage of 400 V without field weakening
❸ Limit characteristic with field weakening

[5-29] Speed-torque characteristics of a synchronous servo motor with field weakening

- The maximally effective motor magnetising current is calculated based on the motor data set in C00084 to C00091.Then the calculated value is internally limited to 98 % of the maximum current set in C00022.
- When field weakening is switched on, the actually used maximum effective motor magnetising current is shown in C00092, if field weakening is switched off, "0 A" are displayed, as before.

Note!

If a Lenze motor is used:

The controller is parameterised automatically so that the field weakening works optimally and there is no danger to the devices.

Stop!

If a motor of a third-party manufacturer is used:

If pulse inhibit is set in the controller, the DC bus is charged with the voltage which corresponds to the current speed of the machine.

As higher speeds at an accordingly higher rotor voltage of the motor can be reached if the field weakening mode is switched on, the DC bus can be charged with a voltage higher than the set DC-bus voltage vdc-bus and possibly exceed the maximum permissible voltage of 800 V when pulse inhibit is set and the motor speed is high!

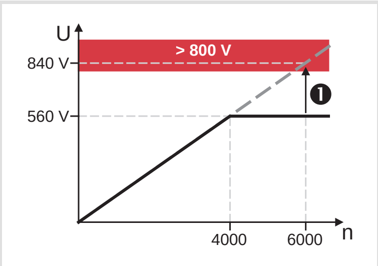
In order to protect the device, either use a brake chopper, or parameterise the speed monitoring via C00596 and C00607, so that only a maximum motor speed is possible that could also be reached with Vdc-bus = 800 V without field weakening.

Maximum motor speed (132)

Example: Voltage increase in the DC bus at field weakening loss

(For example by an active setting of controller inhibit, or by activation of trouble or fault at a high motor speed.)

Field weakening	Speed n	Motor voltage peak value
Switched off	4000 rpm	560 V
	5700 rpm	800 V
	6000 rpm	840 V
Switched on	6000 rpm	560 V



- If pulse inhibit is set at 6000 rpm and switched-on field weakening, the DC bus is charged to over 800 V (❶).
- A speed limitation to 5700 rpm is required, as this speed generates a DC-bus voltage of 800 V when the field weakening mode is switched off.

[5-30] Example: Possible DC-bus voltage > 800 V at field weakening loss

5.8.3 Flying restart function

This function is available from software version V3.0!

Note!

Function only possible for V/f control or sensorless vector control!

In the case of V/f control or sensorless vector control, the current motor speed is only provided to the controller if the motor control is active. However, if the controller is enabled, one cannot always assume that the drive is at standstill. The drive for example may still coast down, or be further operated by a load. It cannot always be assumed that fans are at standstill if the controller is enabled, e. g. if the fan impeller is further driven by an air flow in an undefined direction.

If the flying restart mode is activated in C00990, after controller inhibit is deactivated (or DC-injection braking is cancelled), a flying restart process is automatically started to determine the current motor speed if the following conditions are met:

- V/f control or sensorless vector control are selected as motor control in C00006.
- The position control structure is set to "Phase controller is active" in C02570.
- The *MI_bFlyingSyncBlocked* control input of the motor interface is not assigned or set to FALSE.
- The holding brake, if available, is not applied.

Stop!

If the flying restart function is deactivated and the controller is not enabled at standstill, the output voltage and output frequency does not match the current motor speed. High compensation currents may flow!

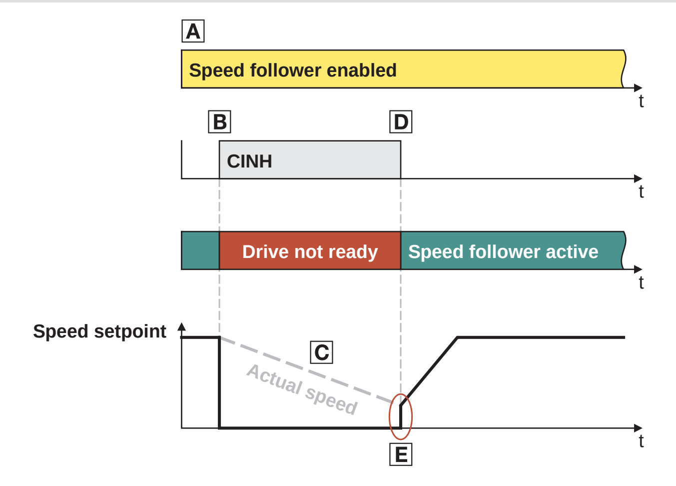
- The drive is first braked towards 0 Hz and is then accelerated again!
- This may cause the following error messages:
  - Controller: Overload during acceleration phases (fault)
  - Device utilisation Ixt > C00123 (warning)
  - Device utilisation Ixt > 100 % (fault)
  - Motor load I2xt > C00127 (warning)
  - Motor load I2xt > C00120 (fault)
  - Overcurrent detected (fault)
  - Overvoltage in DC bus (trouble)

Note!

The flying restart algorithm requires the motor voltage as exact as possible. Therefore it is absolutely necessary to predetermine the inverter error characteristic. ▶ Optimising the switching performance of the inverter (📖 143)

In addition to the exact motor voltage, the stator resistance must also be known exactly. If the flying restart function should not work as required, slightly adjust the setting of the stator resistance in C00084.

Sequence



A. Initial situation: basic function "Speed follower" is enabled and active.
B. The controller is inhibited while the drive is not at standstill.
C. The motor coasts down (torqueless).
D. The controller inhibit is deactivated again.
E. The flying restart process starts.

[5-31] Process example: Speed follower is active ➝ Controller inhibit ➝ Controller enable

Flying restart process

The controller calculates the output frequency required for the momentary speed of the coasting motor, then connects to the system, and accelerates the motor to the defined setpoint again.

- The flying restart process serves to prevent the motor from decelerating to zero speed with subsequent acceleration.
- The currently detected flying restart speed is provided to the application via the current motor speed *MI_dnActualMotorSpeed_s*.

Parameter setting

- The flying restart algorithm injects a current into the motor to identify the current speed. This flying restart current can be parameterised in C00991 in [%] relating to the rated motor current.
  - The higher the current the higher the torque imposed upon the motor.
  - If the current is too low, a wrong speed can be detected.
- The starting frequency of the flying restart algorithm can be set in C00992.
  - If it can be anticipated at which frequency the motor "caught" most frequently in a flying restart process, this frequency is to be set here.
- The integration time of the phase controller is set in C00993.
  - The Lenze setting "60 ms" is adapted for machines with a medium power (45 kW).
  - A guide value for the integration time can be calculated with the following equation as a function of the motor power (C00081):

$$T_i = 1.1\frac{\mu s}{W} \cdot \text{Motor power (C00081)} + 9.4\ \text{ms}$$

  - For accelerating the flying restart process, this guide value can be reduced.
  - If the flying restart frequency (*Frequency.dnActualRotatingFieldFrequency*) oscillates too much, the integration time has to be increased again.
  - A longer integration time increases the time for "catching" the drive.
- To avoid starting a flying restart process at short-time controller inhibit, a time can be set in C00995 for the minimum active controller inhibit time.
  - Since a pulse inhibit > 500 ms causes a controller inhibit, this also applies to the pulse inhibit.

Short overview: Parameters for flying restart process

Parameter	Information	Lenze setting	
		Value	Unit
C00990	Flying restart: Activation	Off	
C00991	Flying restart: Current	15	%
C00992	Flying restart: Start frequency	20.0	Hz
C00993	Flying restart: Integration time	60	ms
C00994	Flying restart: Min. deviation	5.00	°
C00995	Flying restart: Delay time	0	ms

5.8.4 DC-injection braking

This function is available from software version V3.0!

Note!

Function only possible for V/f control or sensorless vector control!

DC-injection braking can be divided into three functionalities:

Manual DC-injection braking

Braking can be activated and deactivated via the internal interface *QSP_bActivateDCBrake* of the basic function "Quick stop".

Tip!

A detailed description of this functionality can be found in the main chapter "Basic drive functions", subchapter "Quick stop":

▶ DC-injection braking". (📖 395)

DC-injection braking instead of quick stop

If DC-injection braking is activated in C00976 instead of quick stop, DC-injection braking is executed automatically when quick stop is activated.

Tip!

A detailed description of this functionality can be found in the main chapter "Basic drive functions", subchapter "Quick stop":

▶ DC-injection braking when quick stop is activated". (📖 397)

Automatic DC-injection braking

This functionality is part of the basic function "Brake control".

If mode 22 has been selected for the brake control in C02580, DC-injection braking is executed automatically if the current speed setpoint falls below the speed threshold set in C02581.

Tip!

A detailed description of this functionality can be found in the main chapter "Basic drive functions", subchapter "Brake control":

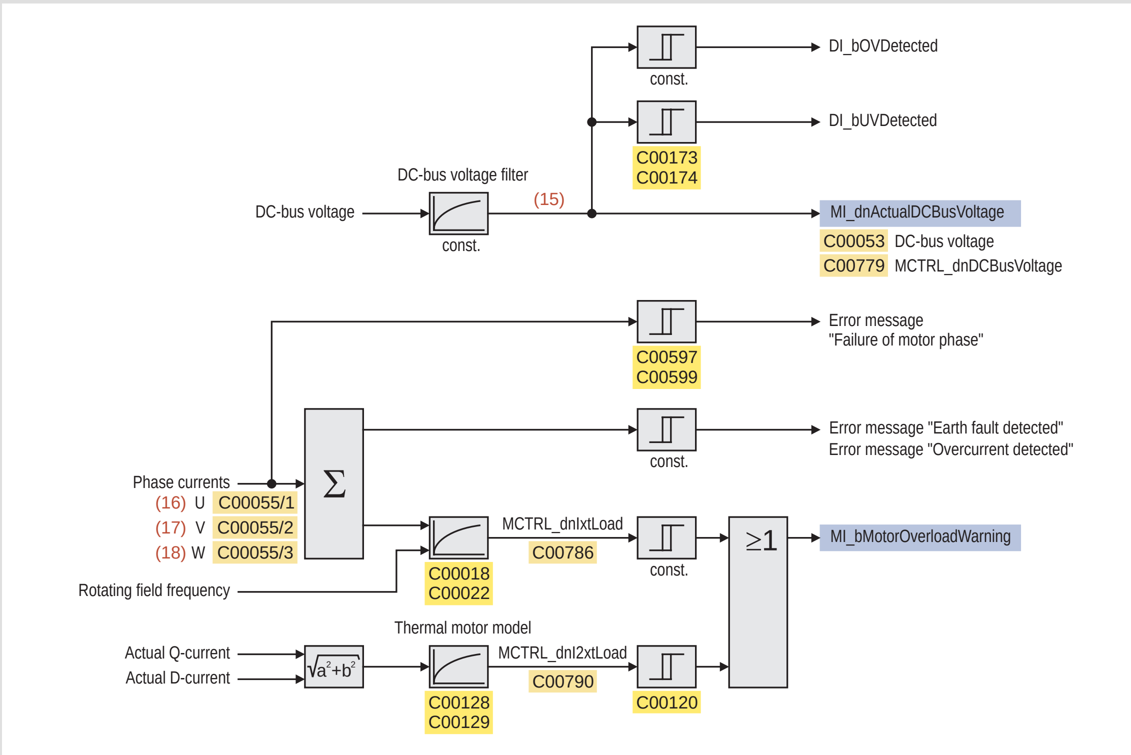
▶ Mode 22: Automatic DC-injection braking". (📖 395)

Short overview: Parameters for DC-injection braking

Parameter	Information	Lenze setting	
		Value	Unit
C00974	DC brakes: Current	0.00	A
C00975	DC brakes Current for quick stop	0.00	A
C00976	DC brake: Activat. by quick stop	Off	

5.9 Monitoring

5.9.1 Signal flow



[5-32] Signal flow of motor interface (monitoring)

Internal variables of the motor control (oscilloscope signals)

▶ The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (📖 590)

No.	Variable of the motor control	Meaning
(15)	Voltage.dnActualDCBusVoltage	Actual DC-bus voltage
(16)	Current.dnActualCurrentPhaseU	Actual motor current (phase U)
(17)	Current.dnActualCurrentPhaseV	Actual motor current (phase V)
(18)	Current.dnActualCurrentPhaseW	Actual motor current (phase W)

5.9.2 Motor monitoring (I^2xt)

The "Servo Drives 9400" are provided with an extended, sensorless thermal I^2xt motor monitoring function which is based on a mathematical model that calculates a thermal motor utilisation from the detected motor currents.

- The calculation considers the speed dependency of the torque (difference between standstill torque and rated torque).
- C00066 indicates the calculated motor utilisation in [%].
- If the motor utilisation exceeds the advance warning threshold set n C00127, the error message "I2t motor overload OC8" is output and the response (default setting: "Warning") set in C00606 is activated..
- If the switch-off threshold set in C00120 is exceeded, the error message "I2t motor overload OC6" is output and the "Fault" response is activated.

STOP **Stop!**

The I^2xt motor monitoring function is no full motor protection!

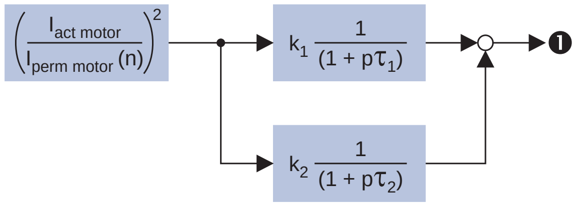
Since the motor utilisation calculated in the thermal model gets lost after mains switching, the following operating states cannot be determined correctly:

- Restart (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 situated directly in the winding of thermal contacts.

Structure of the I^2xt monitoring

The introduction of a two-component model with two time constants (one for the winding and the other for the housing/laminated core) serves to display the thermal behaviour of the motor up to 500% of the rated current:



❶ Thermal utilisation of the motor in [%]

[5-33] Structure of the motor monitoring

Parameter		Setting
$I_{act\ motor}$	Actual motor current	-
$I_{perm\ motor}$ (n)	Permissible motor current (speed-dependent)	-
τ_1	Therm. time constant coil	C00128/1
k_1	Percentage of the winding in the final temperature	C01195
τ_2	Therm. time constant plates	C00128/2
k_2	Percentage of the steel plates in the final temperature	100 % - C01195

Calculation with only one time constant

With C01195 = "0 %" the time constant for the winding is not considered and the thermal model is only calculated with the time constant set for the housing/laminated core.

- The setting C01195 = "0 %" is reasonable if for example the two time constants are not known.
- The calculation simplified due to this setting corresponds to the calculation in the previous Lenze devices (e.g. 9300 servo inverter or ECS).

Speed-dependent evaluation of the motor current

By selecting a characteristic in C01196/1...8 the permissible motor current is evaluated depending on speed for calculating the thermal motor utilisation.

Parameter	Characteristic point	
C01196/1	n_1/n_n	Speed = "0" (standstill)
C01196/2	I_1/I_n	Permissible motor current at standstill
C01196/3	n_2/n_n	Speed from which the torque must be reduced for self-ventilated motors. • Below this speed the cooling air flow of the integral fan is not sufficient anymore.
C01196/4	I_2/I_n	Permissible motor current at speed n_2 (torque reduction)
C01196/5	n_3/n_n	Rated speed
C01196/6	I_3/I_n	Permissible motor current at rated speed
C01196/7	n_4/n_n	Speed above the rated speed (in the field weakening range for asynchronous motors)
C01196/8	I_4/I_n	Permissible motor current at speed n_4 (field weakening)

▶ The speed-dependent evaluation can be more or less switched off by setting C01196/1...8 to "100 %" each. The calculation simplified due to this setting corresponds to the calculation in previous Lenze devices (e.g. 9300 servo inverter or ECS).

Note!

Self-ventilated standard motors are protected insufficiently at low speeds by setting C01196/1...8 to "100 %" each.

Servo motors, however, do not have a point from which the torque must be reduced due to a too low speed.

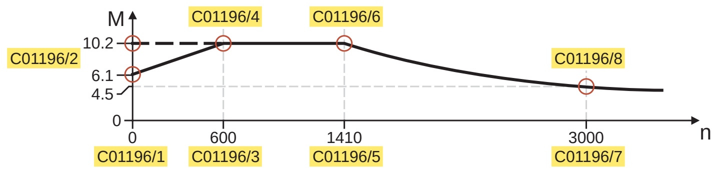
- When setting the characteristic in C01196/1...8 this point must not be ignored. Hence, point 2 is to be set ideally to point 1 or point 3.

5.9.2.1 Example for entry of the characteristic for asynchronous servo motor

Motor type: MDFMARS 090-32

Data from the catalogue:

- Rated speed n_N: 1410 rpm → Setting in C00087
- Rated current I: 6.1 A → Setting in C00088
- Rated torque M_{rated}: 10.2 Nm
- Characteristic of maximum torques (50 Hz, star connection):



[5-34] Torque/speed characteristic for motor type MDFMARS 090-32 from catalogue

Note!

At present, relative current values are still expected for the specification of the interpolation points in subcodes 2, 4, 6, 8 of C01196. This example, however, already uses relative torque values, the entry of which shall be possible at a later date.

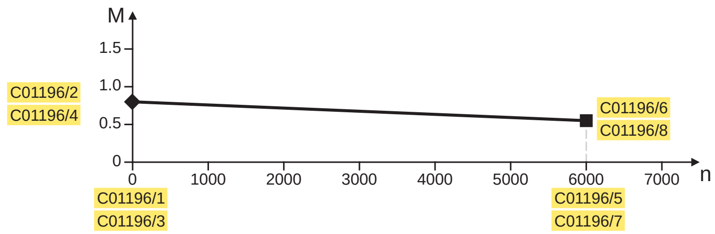
Parameter	Setting	Information
C00128/1	1.0 min	Thermal time constant - winding • Is unknown and is therefore deactivated by setting C01195 = "0 %".
C00128/2	5.0 min	Thermal time constant - laminated core/housing
C01195	0 %	Percentage of the winding in the final temperature.
C01196/1	0 %	Speed = "0" (standstill)
C01196/2	Permissible motor torque at standstill	
Self-ventilated:	60 %	= 6.1 Nm / 10.2 Nm * 100 %
Forced-ventilated:	100 %	= 10.2 Nm / 10.2 Nm * 100 %
C01196/3	Speed n_2 from which the torque must be reduced for self-ventilated motors.	
Self-ventilated:	43 %	= 600 rpm / 1410 rpm * 100 %
Forced-ventilated:	0 %	No torque reduction required.
C01196/4	100 %	Permissible motor torque at speed n_2 (torque reduction)
C01196/5	100 %	Rated speed (≡ 1410 rpm)
C01196/6	100 %	Permissible motor torque at rated speed (≡ 10.2 Nm)
C01196/7	213 %	Speed above the rated speed (in the field weakening range for asynchronous motors) = 3000 rpm / 1410 rpm * 100 %
C01196/8	44 %	Permissible motor torque at speed n_4 (field weakening) = 4.5 Nm / 10.2 Nm * 100 %

5.9.2.2 Example for entry of the characteristic for synchronous servo motor

Motor type: MCS 06C60

Data from the catalogue:

- Rated speed n_{rated}: 6000 rpm ➞ Setting in C00087
- Rated current I: 2.4 A ➞ Setting in C00088
- Rated torque M_N: 0.5 Nm (in S1 operation: 0.55 Nm)
- Characteristic - maximum torques:



[5-35] Torque/speed characteristic for motor type MCS 06C60 from the catalogue

Note!

At present, relative current values are still expected for the specification of the interpolation points in subcodes 2, 4, 6, 8 of C01196. This example, however, already uses relative torque values, the entry of which shall be possible at a later date.

Parameter	Setting	Information
C00128/1	1.0 min	Thermal time constant - winding
C00128/2	14.2 min	Thermal time constant - laminated core/housing
C01195	27 %	Percentage of the winding in the final temperature. (Only the laminated core percentage is known.)
C01196/1	0 %	Speed = "0" (standstill)
C01196/2	160 %	Permissible motor torque at standstill = 0.8 Nm / 0.5 Nm * 100 %
C01196/3	0 %	Speed n_2 from which the torque must be reduced for self-ventilated motors.
C01196/4	160 %	Permissible motor torque at speed n_2 (torque reduction)
C01196/5	100 %	Rated speed (≡ 6000 rpm)
C01196/6	100 %	Permissible motor torque at rated speed (≡ 0.5 Nm)
C01196/7	100 %	Speed above rated speed
C01196/8	100 %	Permissible motor torque at speed n_4 (field weakening)

5.9.3 Motor temperature monitoring

If the winding temperature detected by the motor temperature sensor exceeds the limit value set in C00121, the response set in C00584 is activated as advance warning.

- In the Lenze setting the "Warning" response occurs if the winding temperature exceeds 120 °C.
- As soon as the fixed limit value of 150 °C is exceeded, the response set in C00583 is activated (default setting: "Fault").
- If an open circuit is detected in the motor temperature sensor, the response set in C00594 (default setting: "Fault") is activated.

Note!

By setting C00583 = "0" the monitoring response and the temperature correction is switched off within the motor control (identification and parameter correction)

This setting for example is reasonable if no usable winding temperature signal is available.

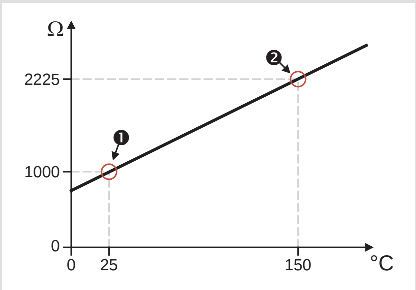
Tip!

The winding temperature currently detected by the motor temperature sensor is displayed in C00063.

5.9.3.1 Special characteristic for motor temperature sensor

If required, you can select and activate a special characteristic for the motor temperature sensor.

- The specific characteristic is selected on the basis of two interpolation points which are to be set in C01191 and C01192. Both interpolation points define a line which is extrapolated to the right and to the left.
- The special characteristic is activated by setting C01190 = "1".
- In the Lenze setting the special characteristic is defined as follows:



- Interpolation point ❶
  - C01191/1 = 25°C
  - C01192/1 = 1000 Ω
- Interpolation point ❷
  - C01191/2 = 150°C
  - C01192/2 = 2225 Ω

[5-36] Lenze setting of the special characteristic

Note!

- By selecting a motor from the motor catalogue the parameters C01190, C01191 and C01192 are overwritten!
- If the controller measures a resistance below 122 Ω, this is interpreted as a sensor error and a temperature of 255 °C is output.

The following applies from software version V4.0:

- Sometimes a short circuit is a desired state (e.g. temperature contact closed below 140 °C). For this purpose, the interpolation point 1 (C01191/1) must be below 122 Ω to prevent the triggering of sensor errors. In this case the temperature continues to be calculated.

5.9.3.2 Motor temperature monitoring (PTC)

For detecting and monitoring 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!

- This monitoring function is only active with line-side power supply of the controller.
- The controller can only evaluate one PTC thermistor! Do not connect several PTC thermistors connected in series or in parallel.
- If several motors are operated via one controller, use thermal contacts (NC contacts) connected in series.
- For full motor protection, you have to install an additional temperature monitoring system with separate evaluation.

Note!

- In the Lenze setting (C00585 = "1: Fault"), motor temperature monitoring is activated!
- In the delivery state, a wire jumper is installed between the terminals X106/ T1 and X106/T2.
- Lenze three-phase AC motor are factory-equipped with a thermal contact.

▶ The monitoring function responds at 1.6 kΩ < R < 4 kΩ across terminals X106/T1 and X106/T2, see functional test below.

▶ If the monitoring function responds:
  - the error response set in C00585 is executed (Lenze setting: "Fault").
  - the error message "PTC has triggered" (0x0077000f) is entered into the logbook of the controller.
  - the status output *MI_bMotorOverloadWarning* is set to TRUE.

▶ Setting C00585 = "0: No response" deactivates the monitoring function.

Tip!

For the operation of motors equipped with PTC thermistors or thermostats, we recommend always to activate the PTC input. This prevents the motor from being destroyed by overheating.

Functional test

Connect an invariable resistor to the PTC input:

▶ R > 4 kΩ : The fault message must be triggered.

▶ R < 1 kΩ : The fault message must not be triggered.

5.9.3.3 Temperature monitoring of a second motor

This function is available from software version V7.0!

From software version V7.0 onwards, two motor temperature sensors can be evaluated simultaneously via the two encoder inputs X7 and X8 when two motor are used (e.g. double motor for a storage and retrieval unit). For this purpose, the selection "X7 and X8 in parallel" must be set as feedback system for the motor temperature in C01193.

- In this case, always the higher temperature of the two detected temperatures is displayed as the current motor temperature on the **Diagnostics** tab and in C00063.
- Moreover, the following display parameters are available from software version V6.0 onwards:
  - C01200/1: Motor temperature via X7
  - C01200/2: Motor temperature via X8
- If one of the two detected temperatures exceeds the limit value set in C00121, the response set in C00584 is activated as advance warning.
  - In the Lenze setting the "Warning" response occurs if one of the two winding temperature exceeds 120 °C.
- As soon as one of the two detected temperatures exceeds the fixed limit value of 150 °C, the response set in C00583 is activated (default setting: "Fault").
- If an open circuit is detected in one of the two motor temperature sensors, the response set in C00594 (default setting: "Fault") is activated.
- It is not possible to set different responses for the two temperature monitoring modes.
- For the motor model in the controller, the mean value of both detected temperatures is used.

Related topics:

▸ Brake control ▸ Control of two motor holding brakes (📖 562)

5.9.4 Motor phase failure monitoring

5.9.4.1 Monitoring of the individual motor phases during operation

If a current-carrying motor phase (U, V, W) fails during operation, the response set in C00597 is executed (Lenze setting: "No response") if two conditions are fulfilled:

Condition 1: Monitoring is activated

To safely detect a motor phase failure, a certain motor current must flow for the current sensor system.

- Monitoring will therefore only be activated if, in the case of servo control the setpoint of the motor current, and in the case of sensorless vector control or V/f control the actual value motor current (display in C00054) has exceeded a certain current value.
- The current value for the activation can be set in C00599 in [%] with regard to the maximum device current (display in C00789).

Condition 2: A certain commutation angle has been covered

In this case monitoring works according to the principle of checking for each motor phase that a current flows depending on the commutation angle.

- Monitoring responds if a certain commutation angle (approx. 150°) is covered without the current having exceeded a (non-parameterisable) threshold that depends on the device power.

Note!

The dependence on the commutation angle also causes a dependence on the motor type used:

- The commutation angle and the angle at the shaft (number of pole pairs) of a synchronous machine are proportional. This makes it possible to predict which shaft angle is maximally covered in case of error.
- There still exists a slip between the commutation angle and the angle at the shaft of an asynchronous machine. This results in a load dependency which makes it impossible to predict a maximally covered shaft angle in case of error.
  – Thus, when a hoist is lowered (non-zero speed), it may happen that no rotating field is applied anymore but a DC current flows. In this case, condition 2 is not fulfilled anymore.

5.9.4.2 Checking the individual motor phases before operation is started

This function is available from software version V5.0!

From software version V5.0 a check via test signal application has been added. It injects a current into the machine before the actual operation is started, by means of which both a motor phase failure and the existence of the motor are checked. Only after the check has been carried out successfully, the actual operation is continued.

- The setpoint current amplitude corresponds to the lower of the two following values:

$$50\ \% \cdot \sqrt{2} \cdot \text{Rated device current}$$

or

$$50\ \% \cdot \sqrt{2} \cdot \text{Rated motor current}$$

- The test signal application is activated directly after controller enable if the following conditions are fulfilled:
  - In C00597 a response other than "No response" is set.
  - No test mode is activated (C00398 = 0).
  - No identification of pole position is active (by device command C00002 = "51" or "52").
  - No identification procedure is active (by device command C00002 = "71" or "72").
- The check actuates the response set in C00597 if one or more motor phase currents have not reached a certain threshold value within 5 ms after controller enable.
  - The threshold value depends on the device power and cannot be parameterised.
  - If only one motor phase current does not reach the threshold value, "Motor phase U/V/W not available" is entered in the logbook.
  - If several motor phases do not reach the threshold value, the motor is considered as not connected, and "Motor not connected" is entered in the logbook.
- The check is completed successfully if all three motor phase currents have exceeded the threshold value. Then the actual operation is continued immediately.

Note!

- As the check is cancelled immediately if all three motor phase currents have exceeded the threshold value, the setpoint current usually is not achieved.
- In order to be able to achieve the threshold value used for the check, the rated motor current must at least be 10 % of the maximum device current.
- This monitoring is independent of the further rotation of the commutation angle.

5.9.5 Maximum current monitoring

The ultimate motor current I_{ULT} to be parameterised in C00620 is a limit value to protect the motor from destruction or influence of the rated data.

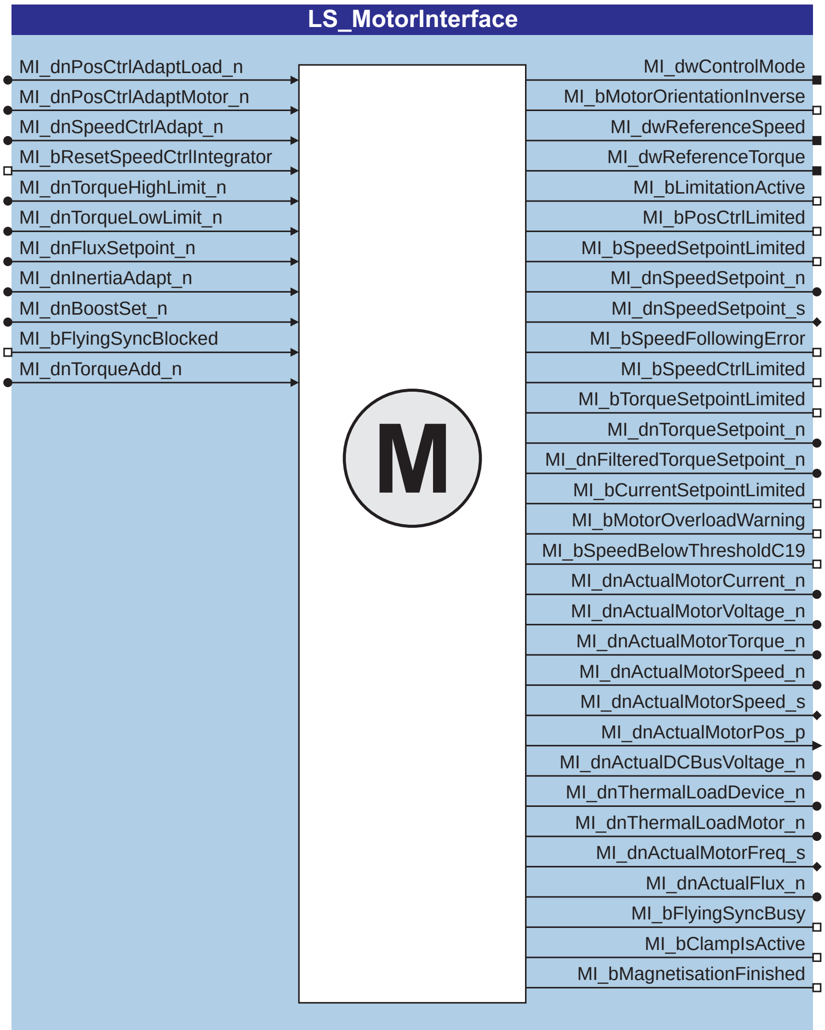
- This limit value must not be travelled cyclically during the drive process.
- The maximum current parameterisable in C00022 should have a sufficient distance from this limit value.
- If the instantaneous value of the motor current exceeds the limit value set in C00620, the response set in C00619 is executed for motor protection (Lenze setting: Fault).

Note!

When you select a Lenze motor from the catalogue and transfer the plant parameters of the motor to the controller, the setting in C00620 is automatically adjusted to the selected motor.

5.10 Internal interfaces | "LS_MotorInterface" system block

The **LS_MotorInterface** system block provides the internal interfaces to the driving machine, consisting of the phase controller, speed controller, and motor control in the function block editor.



Note!

All input and output signals of the motor interface directly refer to the motor!

Inputs

Identifier DIS code | data type	Information/possible settings	
MI_dnPosCtrlAdaptLoad_n C02568/1 | DINT	Dynamic change of the proportional gain Vp of the position controller during operation • For software versions lower than V5.0 the following applies: Internal limitation to 10 ... 200 % • From software version V5.0 the following applies: Internal limitation to 0 ... 200 %	
MI_dnPosCtrlAdaptMotor_n C02568/2 | DINT	Dynamic change of the proportional gain Vp of the phase controller during operation • For software versions lower than V5.0 the following applies: Internal limitation to 10 ... 200 % • From software version V5.0 the following applies: Internal limitation to 0 ... 200 %	
MI_dnSpeedCtrlAdapt_n C02568/3 | DINT	Dynamic change of the proportional gain Vp of the speed controller during operation • If the input is assigned, the following applies: V_P = *MI_dnSpeedCtrlAdapt_n* [%] * C00070 • If the input is not assigned, the following applies: V_P = 100 % * C00070 = C00070 • Internal limitation to 10 ... 200 % ▸ Optimising the speed controller (📖 156)	
MI_bResetSpeedCtrlIntegrator C02569/2 | BOOL	Reset integral action component in the speed controller	
	TRUE	Integral action component is reset to "0".
MI_dnTorqueHighLimit_n C02568/4 | DINT MI_dnTorqueLowLimit_n C02568/5 | DINT	Upper/lower limit value for correcting variable of the speed controller and total torque setpoint • These two inputs serve to select an external torque limitation. – If the motor torque reaches the selected limits, the drive can no longer follow the speed setpoint! – If the torque limitation is active, the output *MI_bTorqueSetpointLimited* is set to TRUE. • 100 % ≡ C00057/2 • Only a positive torque is permissible as upper limit value. • Only a negative torque is permissible as lower limit value. • The motor mounting position (C02527) defines the assignment to the limitation inputs of the motor control. • The internally effective torque limit values are displayed in C02559/1...2.	
MI_dnFluxSetpoint_n C02568/7 | DINT	Setpoint for the field controller	
MI_dnInertiaAdapt_n C02568/8 | DINT	Adaptation of the moment of inertia in [%] • If input is not assigned = 100 % • Internal limitation to 0 ... 200 %	
MI_dnBoostSet_n C02568/9 | DINT From V3.0	Boost voltage • 100 % ≡ 1000 V	
MI_bFlyingSyncBlocked C02569/16 | BOOL From V3.0	Block flying restart ▸ Flying restart function (📖 218)	
	FALSE	Flying restart is executed.
	TRUE	Flying restart process is blocked.

Identifier DIS code | data type	Information/possible settings	
MI_dnTorqueAdd_n C02568/10 | DINT From V8.0	Additional torque feedforward control value in [%] This input serves to provide an additional torque setpoint. In this way, you can provide an additional torque for the basic functions manual jog, positioning and homing besides the acceleration feedforward control. • 100 % = motor reference torque If the controller is enabled, the torque setpoints at this input have a direct effect on the drive! The user has to • apply the appropriate setpoint for every state of the drive. • avoid setpoint step-changes.	
	- 200 %	
	+ 200 %	

Outputs

Identifier DIS code | data type	Value/meaning	
MI_dwControlMode DWORD	Active control structure of the motor control • Displayed value is bit-coded:	
	Bit 1	Position control without feedback, external following error calculation
	Bit 2	Position control with encoder on the motor side
	Bit 3	Position control with encoder on the load side
	Bit 4	Speed control
	Bit 5	Torque control
MI_bMotorOrientationInverse BOOL	Parameterised motor mounting position	
	FALSE	Motor mounting position in the same direction, setpoints are not defined.
	TRUE	Motor mounting position in the opposite direction, setpoints are reversed.
MI_dwReferenceSpeed DWORD	Parameterised motor reference speed (C00011) in [rpm]	
MI_dwReferenceTorque DWORD	Reachable motor torque with I_{max_device} (C00022) in [mNm] • 1000 mNm ≡ 1 Nm • Display in C00057/2 in [Nm]	
MI_bLimitationActive C02569/3 | BOOL	Status signal "Internal limitation active" • Group signal for all limitation messages.	
	TRUE	One of the internal limitations is active.
MI_bPosCtrlLimited C02569/4 | BOOL	Status signal "Phase/position controller at the limit"	
	TRUE	The limitation of the phase and/or position controller is active.
MI_bSpeedSetPointLimited C02569/5 | BOOL	Status signal "Resulting speed setpoint at the limit"	
	TRUE	The resulting speed setpoint is limited to the limit values parameterised in C00909/1 and C00909/2.
MI_dnSpeedSetpoint_n DINT	Current speed setpoint from position control and speed feedforward control or direct setpoint selection in [%] • After limitation by the upper speed limit value (C00909/1) and lower speed limit value (C00909/2). • 100 % ≡ Motor reference speed (C00011)	
MI_dnSpeedSetpoint_s DINT	Current speed setpoint from position control and speed feedforward control or direct setpoint selection in [rpm] • After limitation by the upper speed limit value (C00909/1) and lower speed limit value (C00909/2).	

Identifier DIS code | data type	Value/meaning	
MI_bSpeedFollowingError C02569/10 | BOOL	Status signal "Impermissible speed control deviation"	
	TRUE	Speed control deviation is higher than the window set in C00576.
MI_bSpeedCtrlLimited C02569/6 | BOOL	Status signal "Speed controller at the limit"	
	TRUE	The speed controller limitation is active.
MI_bTorqueSetpointLimited C02569/7 | BOOL	Status signal "Total torque setpoint at the limit"	
	TRUE	The total torque setpoint is limited.
MI_dnTorqueSetpoint_n DINT	Current torque setpoint from speed control and torque feedforward control or direct setpoint selection • After limitation by *MI_dnTorqueLimit_n*. • 100 % ≡ C00057/2	
MI_dnFilteredTorqueSetpoint_n DINT	Filtered torque setpoint (after jerk limitation and band-stop filter) • 100 % ≡ C00057/2	
MI_bCurrentSetpointLimited C02569/8 | BOOL	Status signal "Setpoint for current controller at the limit"	
	TRUE	The setpoint for the current controller is limited to I_{max_device} (C00022).
MI_bMotorOverloadWarning C02569/11 | BOOL	Status signal "Motor overload" • Group signal for warning signals from temperature monitoring (KTY, PTC, thermal switch) or I^2xt monitoring.	
	TRUE	One of the monitoring modes for motor overload protection is active.
MI_bSpeedBelowThresholdC19 C02569/9 | BOOL	Status signal "Standstill reached"	
	TRUE	The current speed is below the threshold set in C00019.
MI_dnActualMotorCurrent_n DINT	Actual motor current • 100 % ≡ I_{max_device} (C00789) • Display in C00780 in [A]	
MI_dnActualMotorVoltage_n DINT	Current motor voltage • 100 % ≡ 1000 V • Display in C00783 in [V]	
MI_dnActualMotorTorque_n DINT	Current motor torque • 100 % ≡ C00057/2 • Display in C00774 in [Nm]	
MI_dnActualMotorSpeed_n DINT	Current speed of the motor shaft in [%] • 100 % ≡ Motor reference speed (C00011)	
MI_dnActualMotorSpeed_s DINT	Current speed of the motor shaft in [rpm] • Display under C00772	
MI_dnActualMotorPos_p DINT	Current position of the motor shaft in [increments] • Display under C00770	
MI_dnActualDCBusVoltage_n DINT	Actual DC-bus voltage • 100 % ≡ 1000 V	
MI_dnThermalLoadDevice_n DINT	Thermal device utilisation in [%] • Current result of the Ixt calculation. • Display in C00064 ▸ Monitoring of the device utilisation (🕮 116)	
MI_dnThermalLoadMotor_n DINT	Thermal motor utilisation in [%] • Current result of the I^2xt calculation. • Display in C00066 ▸ Motor monitoring (I^2xt) (🕮 224)	
MI_dnActualMotorFreq_s DINT From V3.0	Current motor frequency in [Hz] The motor frequency corresponds to the field frequency [Hz]. Field frequency [Hz] = motor speed × number of motor pole pairs	

Identifier DIS code | data type	Value/meaning	
MI_dnActualFlux_n DINT From V3.0	Actual flux value	
MI_bFlyingSyncBusy C02569/13 | BOOL From V3.0	Status signal "Flying restart is active" ▶ Flying restart function (📖 218)	
	TRUE	Flying restart is executed.
MI_bClampIsActive C02569/14 | BOOL From V3.0	Status signal "Clamping is active"	
	TRUE	Clamping is active.
MI_bMagnetisationFinished C02569/15 | BOOL From V3.0	Status signal "Motor magnetisation is completed"	
	TRUE	Motor magnetisation is completed.

6 Encoder evaluation

This chapter contains information on how to use feedback systems for the motor control.

Danger!

If the encoder/resolver is used as motor encoder:
In case of error, safe operation of the motor is no longer guaranteed!

When servo control is used:

- For the (open circuit) monitoring of the encoder/resolver for reasons of safety always the "Fault" response (Lenze setting) should be set!

When V/f control is used:

- For this type of motor control, the drive basically is to coast down after an encoder failure and may not stop, therefore the "Warning" response is to be set for the (open circuit) monitoring in this case!

Parameters for the (open circuit) monitoring:

- C00580: Response to open circuit of encoder
- C00586: Response to open circuit of resolver
- C00601: Response to communication error of encoder

Note!

The encoder position is stored safe against mains failure in the memory module and is therefore known to the drive control even after the mains has been switched.

With regard to their position resolution, higher-level applications are based on the resolution of the encoder which is activated for position control.

Behaviour of the home position after mains switching

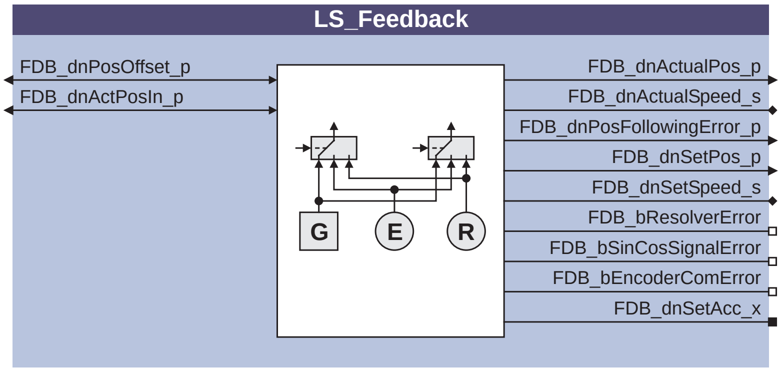
If the home position/information is also to be available after mains switching, the setting C02652 = "1: Received" is required.

- Another condition for keeping the home position/information after mains switching is the compliance with the maximum permissible angle of rotation of the encoder, which can be set in C02653.
- When resolvers or single-turn absolute value encoders are used and the mains is switched off (24 V supply off), the encoder may only be moved by ½ revolution since otherwise the home position will get lost due to the ambiguity of the encoder information.

See also: ▸ Parameterise motor encoder (📖 134)

6.1 Internal interfaces | "LS_Feedback" system block

The **LS_Feedback** system block provides the internal interfaces for the basic function "Encoder evaluation" in the function block editor.



Inputs

Identifier Data type	Information/possible settings
FDB_dnPosOffset_p DINT	Offset for position setpoint and actual position in [increments]
FDB_dnActPosIn_p DINT	External actual position in [increments] • For the selection of an external actual position with a corresponding position control. ▸ Use of an external position encoder (🕮 244)

Outputs

Identifier DIS code | data type	Value/meaning	
FDB_dnActualPos_p DINT	Current position of the position encoder in [increments]	
FDB_dnActualSpeed_s DINT	Current speed of the position encoder in [rpm]	
FDB_dnPosFollowingError_p DINT	Current following error in [increments]	
FDB_dnSetPos_p DINT	Set position calculated by active basic drive function in [increments] • Considering the motor mounting position. • In the case of an active speed or torque control (instead of position control) the actual position (*FDB_dnActualPos_p*) is shown at this output.	
FDB_dnSetSpeed_s DINT	Setpoint speed calculated by active basic drive function in [rpm] • Considering the motor mounting position.	
FDB_bResolverError C02579/1 | BOOL	Status signal "Resolver error"	
	TRUE	A resolver error (e.g. open circuit) has occurred.
FDB_bSinCosSignalError C02579/2 | BOOL	Status signal "sine/cosine encoder error"	
	TRUE	A sine/cosine encoder error has occurred.

<table>
<tr><th>Identifier
DIS code | data type</th><th colspan="2">Value/meaning</th></tr>
<tr><td rowspan="2">FDB_bEncoderComError
C02579/3 | BOOL</td><td colspan="2">Status signal "Encoder communication error"</td></tr>
<tr><td>TRUE</td><td>An encoder communication error has occurred.</td></tr>
<tr><td>FDB_dnSetAcc_x
DINT
From V7.0</td><td colspan="2">Setpoint acceleration calculated by active basic function
Considering the motor mounting position.
For the basic functions Stop, Manual jog, Homing and Positioning, the internal acceleration resulting from the profile generation is output.
For the basic functions Quick stop, Position follower and Speed follower, the acceleration from the differentiated setpoint speed is determined. C02562 serves to filter the determined acceleration.
With an active torque control or in a non-controlled operation (function states "Controller not ready" and "Error"), the value "0" is output.</td></tr>
</table>

6.1.1 Use of an external position encoder

The *FDB_dnActPosIn_p* input serves to evaluate an external encoder (CAN, SSI, Profibus) for the position control.

- Via this input, a current actual position of an external encoder in [increments] can be directly transferred to the encoder evaluation.

How to activate the use of the external actual position:

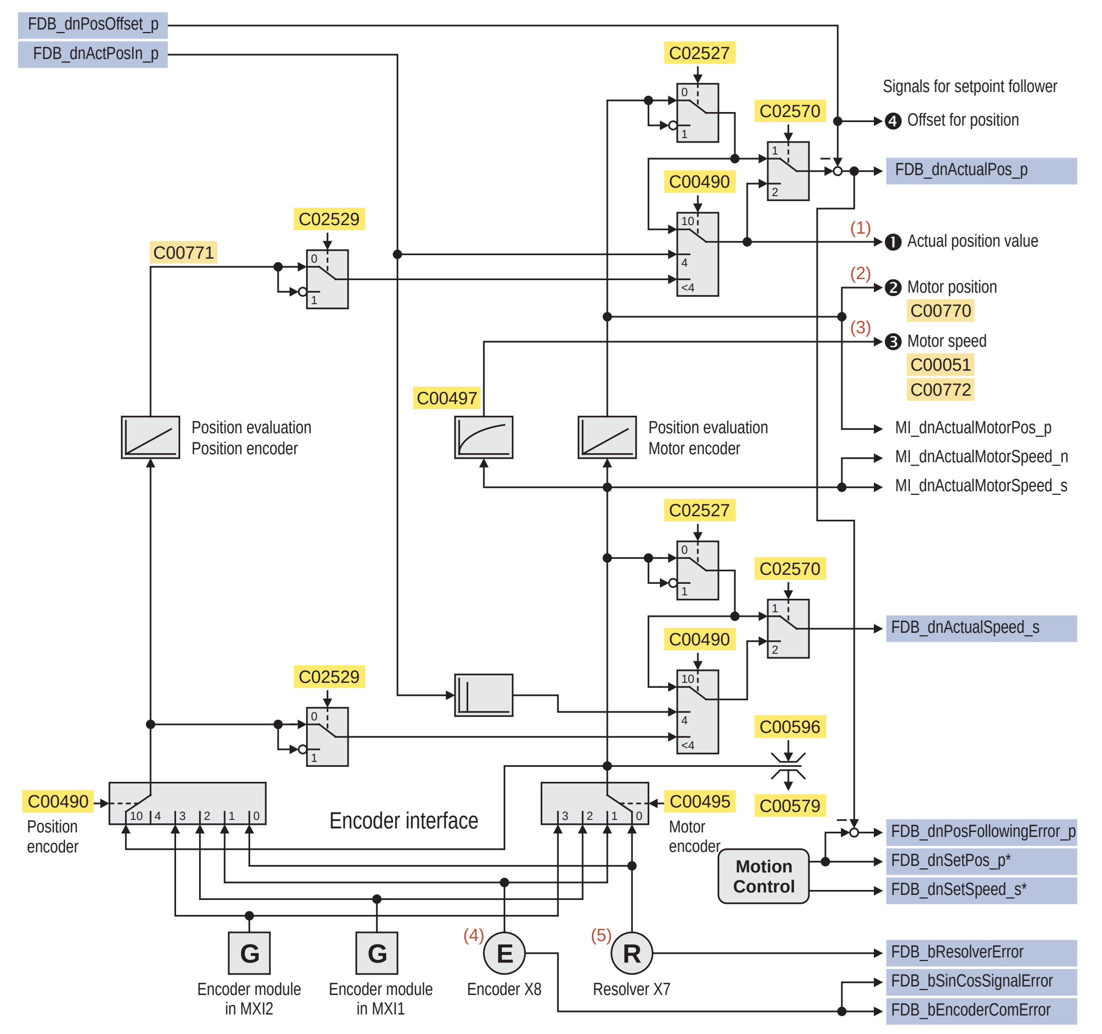
On the **Application parameters** tab in the dialog level *Overview* → *Drive interface* → *Machine parameters:*

1. Select the "Position controller active" setting in the **Position control structure** list field (C02570), so that the position encoder is evaluated.
2. Set "From application" in the **Position encoder selection** list field (C00490).

Note!

- Encoder inversion and offset selection *FDB_dnPositionOffset_p* also affect the external actual position.
- If the use of the external actual position preset via *FDB_dnActPosIn_p* is activated, the "Home position known" status (*HM_bHomePosAvailable* = TRUE) is automatically set and homing with the basic function "Homing" cannot be activated anymore.
- If the traversing range (C02528) is set to "Modulo", the external actual position also has to be defined as modulo (0 ... cycle-1).

6.2 Signal flow



MI_... = output signals of the motor interface. ▸ Internal interfaces | "LS_MotorInterface" system block (📖 236)
* Set position and setpoint speed are calculated from the basic function that is currently active.

[6-1] Signal flow - encoder evaluation

Internal variables of the motor control (oscilloscope signals)

▶ The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (📖 590)

No.	Variable of the motor control	Meaning
(1)	Position.dnActualLoadPos	Actual position
(2)	Position.dnActualMotorPos	Current motor position
(3)	Speed.dnActualMotorSpeed	Current motor speed
(4)	Speed.dnActualEncoderSpeed	Current encoder speed
(5)	Speed.dnActualResolverSpeed	Current resolver speed

6.3 Parameter setting

Short overview of parameters for the encoder evaluation:

Parameter	Information	Lenze setting Value	Unit
C00058/1	Pole position resolver	-90.0	°
C00058/2	Rotor displ. angle encoder	0.0	°
C00058/3	Pole position module	0.0	°
C00080	Resolver - number of pole pairs	1	
C00416	Resolver error correction	0	
C00417	Dynamics of the resolver evaluation	100	%
C00418	Activate resolver error compensation	Deactivated	
C00420	Encoder - number of increments	512	
C00421	Encoder voltage	5.0	V
C00422	Encoder type	Incremental encoder (TTL signal)	
C00423	SSI encoder: Bit rate	400	kbps
C00424	SSI encoder: Data word length	25	Bit
C00427	TTL encoder signal evaluation	4x evaluation (A, B)	
C00435/1...8	SSI encoder: Partword starting position	0	
C00436/1	SSI encoder: partword length (partword 1)	31	
C00436/2...8	SSI encoder: partword length (partwords 2...8)	0	
C00437/1...8	SSI encoder: partword coding	Binary coded	
C00490	Position encoder selection	Motor encoder	
C00495	Motor encoder selection	Resolver X7	
C00497	Speed act. val. time const.	2.0	ms
C00579	Resp. to speed monitoring	Off	
C00580	Resp. to encoder open circuit	Fault	
C00586	Resp. to resolver open circuit	Fault	
C00601	Resp. to encoder fault	Fault	
C00621	Resp. to angular drift of encoder	No response	
C02527	Motor mounting direction	Motor rotating CW	
C02529	Position encoder mounting direction	Encoder rotating CW	
C02570	Position control structure	Phase controller is active	
C02572	Speed setpoint (enc. eval.)	-	Unit/s
C02573	Position setpoint (enc. eval.)	-	Unit
C02574	Actual speed (encoder eval.)	-	Unit/s
C02575	Actual position (enc. eval.)	-	Unit
C02576	Following error	-	Unit
C02577	External actual position	-	Unit
C02578	Offset actual pos. value/setp.	-	Unit
C02760	Activate Encoder	Deactivated	
C02761	Resolution Multiturn	-	Rev
C02762	Encoder position	-	Steps.
C02763	Encoder position	-	Rev
C02764	Encoderspeed	-	rpm

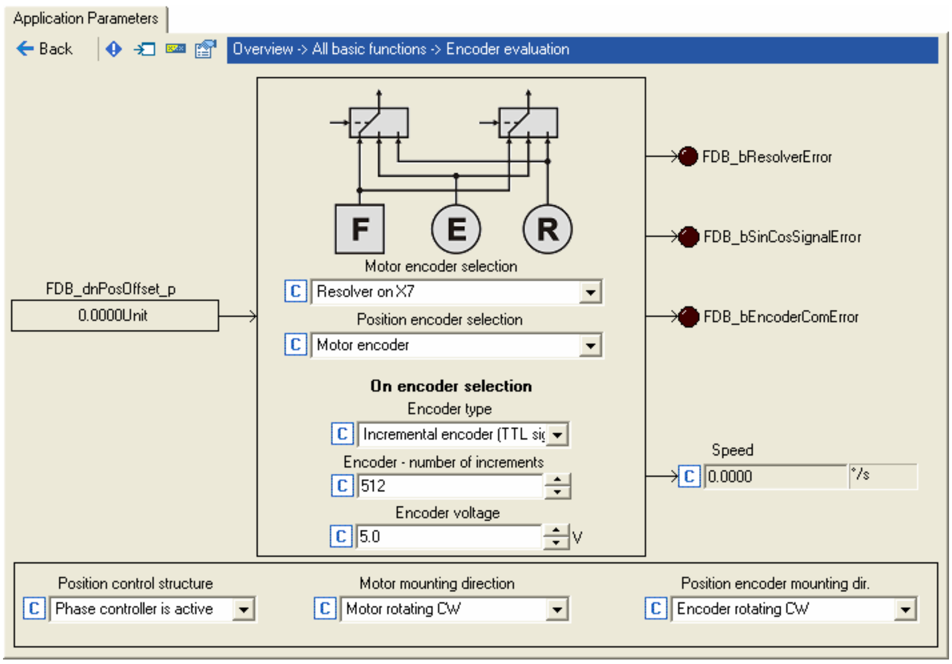
Highlighted in grey = display parameter

Parameter	Information	Lenze setting	
		Value	Unit
C02765	ENC_bError	-	
C02862/1	Resolver: Cos gain	100	%
C02862/1	Resolver: Sin gain	100	%
C02863	Resolver: Angle correction	0	
Highlighted in grey = display parameter			

How to get to the dialog for setting the encoder evaluation parameters:

1. Go to the *Project view* of the »Engineer« and select the 9400 HighLine controller.
2. Select the **Application parameters** tab from the *Workspace*.
3. Click the button **All basic functions** in the *Overview* dialog level.
4. Click the button **Encoder evaluation** in the dialog level *Overview* → *All basic functions*.

Parameterisation dialog in the »Engineer«



See also: ▸ Parameterise motor encoder (📖 134)

6.3.1 Controller configuration

The device interfaces for the encoder on the motor side and, if available, on the load side are directly assigned to the corresponding control according to the structure of the position control selected (C02570):

	Phase control (Lenze setting)	Position control
Cycle time:	250 µs	Application-dependent
Dead time:	Smaller dead time in the actual value channel	Same dead time for position setpoint and actual position
Use:	In positioning technology and single-axis applications or if only one encoder is used.	In multi-axis applications or if a position encoder is used in addition to the motor encoder.

- If only an encoder on the motor side is available, this "motor encoder" provides the actual value signals for the phase/position control and the speed control.
  - In this case both the angle control and the position control can be selected.
  - When selecting the position control, make sure that the position encoder selection "10: Motor encoder (C00495)" is set in C00490. With this selection, the mounting position and the resulting gearbox factor are already considered.
  - The motor encoder supports the secondary servo control irrespective of the use for position and speed control (commutation).
- If an additional encoder is available on the load side, this "position encoder" only supports the position control and C02570 accordingly has to be set to "Position controller active", so that the position encoder is evaluated.
  - The used position encoder must be set in C00490.
  - The position encoder mounting direction must be set in C02529.
  - The starting position of the position encoder can be set via the basic function "Homing".

Note!

When the basic function "Quick stop" is activated, the controller configuration is always switched over to angle control internally, irrespective of the setting in C02570.

- If the basic function "Quick stop" is to be used, the gain of the phase controller (C00254) must also be set correctly for the "Position control" controller configuration.

For the technology applications for the interconnection via the "Electrical shaft", the controller configuration is set to position control in the default setting.

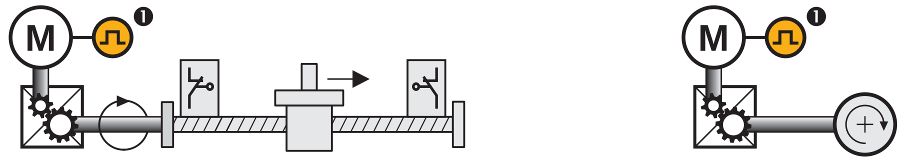
From software version V7.0 onwards, the selection "3: Position controller active" is available in C02570.

- In contrast to the already existing selection "2: Position controller active (<= FW V5.xx)", this selection considers the gearbox factor.
- Further explanations on this can be obtained from the following table:

C02570 = **2: Position controller active (<= FW V5.xx)**	C02570 = **3: Position controller active**
When the separate position encoder at the output end is used, the reference speed to the tool is assumed. This causes the acceleration and deceleration times not to refer to the motor but to the encoder. In order to re-establish the motor reference, the desired acceleration time of the corresponding function must be multiplied by the resulting gearbox factor.	When the separate position encoder at the output end is used, the reference speed is referred to the motor. Thus, all acceleration and deceleration times are calculated with regard to the reference speed at the motor.
Example: • Motor reference speed (C00011) = 3000 rpm • Resulting gearbox factor = 10 • Acceleration time = 1 s With 10 % setpoint selection:	
• Motor speed = 100 % (3000 rpm) • Tool speed = 300 rpm • Acceleration time up to 10 % setpoint selection (100 % motor speed) = 0.1 s	• Motor speed = 10 % (300 rpm) • Tool speed = 30 rpm • Acceleration time up to 10 % setpoint selection (10 % motor speed) = 0.1 s

6.3.2 System with motor encoder

No encoder is installed on the load side. The motor position (angle of rotation) and motor speed are detected via the motor encoder selected in C00495 and are converted with regard to the load side.



❶ Motor encoder

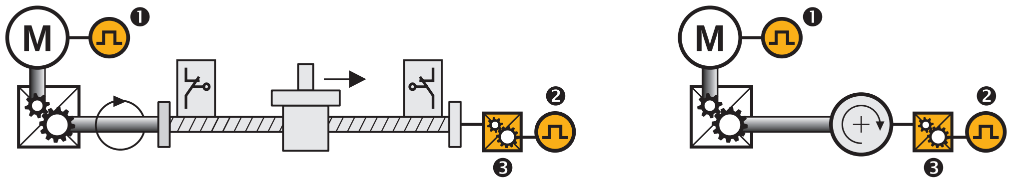
[6-2] Schematic diagram - feedback with position encoder = motor encoder

The actual position and actual speed values on the machine side result from the conversion via the gearbox factor on the motor side (C02520/C02521) and the feed constant (C02524).

See also:
- Gearbox ratio (📖 38)
- Feed constant (📖 43)

6.3.3 System with motor encoder and position encoder

The optional position encoder is used as a feedback for the position control and transmits the position of slide or drive roll to the controller.



❶ Motor encoder
❷ Position encoder
❸ Gearbox ratio for position encoder

[6-3] Schematic diagram - feedback with separate position encoder

In this case, the actual position and actual speed values on the machine side result from the conversion of the position encoder position via the resulting gearbox factor (ratio of the motor speed to the position encoder speed; display in C02531/3) and the feed constant (C02524).

How to activate the use of a separate position encoder:

On the **Application parameters** tab in the dialog level *Overview* → *Drive interface* → *Machine parameters:*

1. Select the "Position controller active" setting in the **Position control structure** list field (C02570), so that the position encoder is evaluated.
2. Select the position encoder available in the **Position encoder selection** list field (C00490).
3. Select the gearbox ratio of the position encoder (ratio of load speed to encoder speed) as a quotient (numerator/denominator) which results from the resulting teeth number:
   - Enter numerator in the input field **Gearbox fact. num.: Pos. enc.** (C02522).
   - Enter denominator in the input field **Gearbox fact. denom.: Pos. enc.** (C02523).
4. If required, adapt the position encoder mounting direction via the **Position encoder mounting direction** list field (C02529). The position encoder mounting direction is preset to "Encoder rotating CW".

Tip!

In C02531/2 the gearbox factor is displayed in decimal format.

Important reference variables converted to the load side:

- Motor reference speed (C00011) → Load reference speed (C02542)
- Reference torque (C00057/2) → Load reference torque (C02543)

See also: ▸ Feed constant (📖 43)

6.3.4 Adaptation of the resolver evaluation dynamics

This function extension is available from software version V5.0 onwards!

The resolver evaluation of the controller is adapted to the resolver types mounted in Lenze motors and offers a good compromise between the dynamic performance and interference suppression. If the resolver is used as a speed feedback system, the dynamic performance of the resolver evaluation determines, among other things, the maximum speed controller gain by means of which stable operation is possible.

In a system with an EMC-compliant structure (low interference), you can increase the dynamics of the resolver evaluation in C00417 without a loss in quality in the speed signal. By increasing C00417, the evaluation gets more dynamic and thus the speed controller gain Vp (C00070) also increases without leaving the stable operating range.

The acceleration of the evaluation depends on the cable length, the resolver, and the quality of the electrical shielding. In many cases, a setting of C00417 = 300 % is possible which can double the speed controller gain. The higher gain in the speed controller may reduce following errors.

See also:

- Servo control (SC): Optimising the speed controller (156)
- Sensorless vector control (SLVC): Optimising the speed controller (183)

6.3.5 Parameterisation of an unknown Hiperface encoder

This function extension is available from software version V4.0!

Application: A Hiperface encoder is to be used with the controller, the current firmware of which has not (yet) been stored permanently.

How to parameterise a Hiperface encoder unknown to the controller:

1. Determine the type code of the encoder.
   - If the encoder has already been connected and read out, the type code is indicated in C00413.
   - Alternatively, the type code can be obtained from the manufacturer or gathered from the documentation for the encoder.
2. Set the type code of the encoder in C00414.
   - Please observe that the decimal format has to be used for the setting. The type codes provided by the manufacturer, however, are in the hexadecimal format.
3. If a multi-turn encoder is used, set the number of displayable resolutions in C00415.
   - This value can also be gathered from the documentation for the encoder.
4. Set the number of encoder increments in C00420.
   - This step should be the last step because it initiates a new readout of the encoder.

6.3.6 Support of the laser measuring system (DME4000/5000)

This function extension is available from software version V4.0!

From software version V4.0, the laser measuring system (DME4000/5000) is supported. The laser measuring system provides the opportunity to set the period length on the encoder.

Note!

Due to its properties, the laser measuring system may not be used as a motor encoder!

Within the controller the evaluation of optical encoders is dimensioned for rotary-operating encoders. Code C00420 describes the number of encoder increments per mechanical motor revolution. In order to be able to correctly evaluate the laser measuring system, or an optical linear measuring scale in general, a back calculation from a linear movement to one motor revolution has to be carried out.

Calculation of the number of increments for rotary machines

For rotary machines driving a slide via a spindle or similar, to which the length measuring system is mounted, the number of encoder increments (C00420) is calculated as follows:

$$\text{Number of increments} = \frac{\text{Distance covered on the encoder during one motor revolution}}{\text{Period length of the encoder}}$$

Calculation of the number of increments for linear direct drives

If the controller activates a linear direct drive, one mechanical motor revolution is defined as follows:

$$\text{One mech. motor revolution} = 2 \cdot \text{length of a pole pitch} \cdot \text{number of pole pairs}$$

The length of a pole pitch describes the distance between the north and south pole at the linear direct drive. The number of encoder increments (C00420) thus is calculated as follows:

$$\text{Number of increments} = \frac{2 \cdot \text{length of a pole pitch} \cdot \text{number of pole pairs}}{\text{Period length of the encoder}}$$

6.3.7 Use of an SSI encoder at X8

This function extension is available from software version V5.0 onwards!

From software version V5.0 all encoders at X8 using the Stegmann SSI protocol are supported.

- Supported bit rates for the SSI communication: 150 ... 1000 kbits
- Supported data word widths: 1 ... 31 bits (effectively)
- Supported output code of the SSI encoder: Gray or binary
- The SSI encoder can be used as position encoder or master encoder with a minimum cycle time of 1 ms.
- The SSI encoder can be supplied with a voltage of up to 12 V and a current of 0.25 A via X8.
- The SSI data words received are provided to the application via the LS_SsiEncoderX8 system block for further processing within the function block editor.

Note!

The LS_SsiEncoderX8 SB is only provided within controllers with a MM3xx or MM4xx memory module.

How to parameterise the SSI encoder at X8:

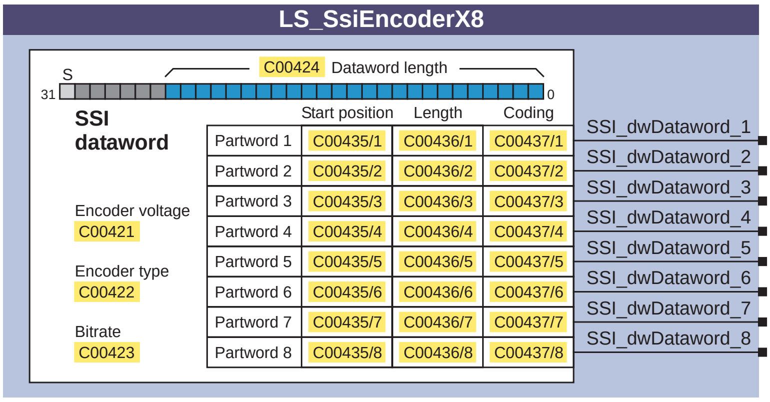
1. Set the supply voltage of the SSI encoder used in C00421.
2. Set the selection "4: SSI encoder" as encoder type in C00422.
3. Set the bit rate for SSI communication in C00423.
   - In the case of the SSI protocol, the permissible baud rate decreases with an increasing cable length. Depending on the length of the encoder cable used and the electromagnetic interference level, a safe bit rate must be set here.
   - Lenze setting: 400 kbits (for encoder cables with a length of up to ≈ 50 meters)
4. Set the data word width in C00424, i. e. the number of data bits that is used for the transmission of a complete SSI data packet.
   - Lenze setting: 25 bits (Stegmann multiturn SSI encoder)
5. Optionally: Split the SSI data word received into partwords and connect a data conversion from Gray into binary code, which may be required (see the following subchapters).

6.3.7.1 "LS_SsiEncoderX8" system block

The **LS_SsiEncoderX8** system block provides the SSI data words received to the application for further processing in the function block editor.

Note!

- The **LS_SsiEncoderX8** SB is only provided within controllers with a MM3xx or MM4xx memory module.
- If a position is transmitted in the SSI data word, it is output in an unchanged manner with regard to the position format by the **LS_SsiEncoderX8** SB. For a use of the SSI encoder as position encoder the position has to be converted into the 9400 format afterwards by means of the **L_EsEncoderConv** FB.



Outputs

Identifier Data type	Value/meaning
SSI_dwDataword_1 DWORD	SSI partword 1 • In the Lenze setting the complete SSI data word received is shown at this output without a conversion of the data format.
SSI_dwDataword_2 ... SSI_dwDataword_8 DWORD	SSI partwords 2 ... 8 • In the Lenze setting these outputs are deactivated. ▸ Division of the SSI data word into partwords (📖 257)

Gray-binary conversion

If an SSI encoder with Gray coding is used, a data conversion of Gray-to-binary code can be connected in C00437/1...8 individually for each output of the **LS_SsiEncoderX8** SB, and thus for each partword.

▶ In the Lenze setting "Binary coded" there is no conversion, i. e. an SSI encoder with binary coding is expected.

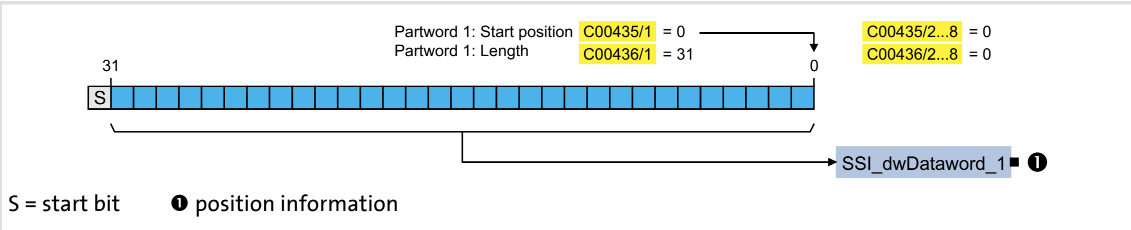
6.3.7.2 Division of the SSI data word into partwords

The LS_SsiEncoderX8 SB can be configured so that it splits up the SSI data word received by the encoder interface into several partwords.

- A separation into partwords is reasonable if the SSI data word also contains other data (like for instance fault or status information) in addition to the position.
- The max. 8 possible partwords are fixedly assigned to the outputs *SSI_dwDataword_1* ... *SSI_dwDataword_8*.
- The partwords are configured via the following parameters:

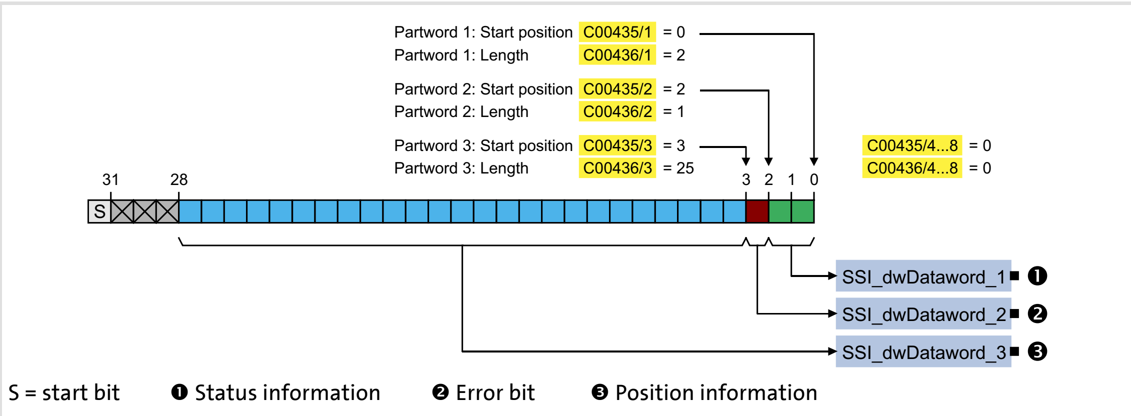
Parameter	Information
C00435/1...8	**Starting position for partwords 1 ... 8** In order to be able to display the individual components of the SSI data word received at different outputs, this code serves to specify the bit position with which the partword for the respective output starts for the eight possible outputs of the LS_SsiEncoderX8 SB . Subcode 1 is fixedly assigned to the first output, subcode 2 to the second output, etc.
C00436/1...8	**Length of the partwords 1 ... 8** Apart from the position of the first bit, also the bit length of each partword is important for the separation. A length of zero means that no partword is to be shown at the corresponding output (output = 0). Here also subcode 1 is fixedly assigned to the first output, subcode 2 to the second output, etc.

- In the Lenze setting the complete SSI data word received is shown at the output *SSI_dwDataword_1*:



[6-4] Example 1: Lenze setting

- The following example shows the parameterisation required to split up the SSI data word received into three partwords (here status information, error bit, and position information):



[6-5] Example 2: Splitting up the SSI data word received into three partwords

6.3.8 Provision of the encoder signal of input X8

This function extension is available from software version V7.0!

The **LS_EncoderX8** system block serves to provide the encoder signal of input X8 to the application, independent of the selected feedback system for the motor encoder and position encoder.

Application cases:

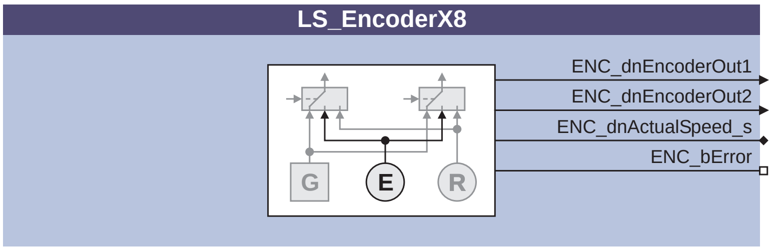
- High-resolution speed encoder as master encoder /value, correcting signal, ...
- Absolute value encoder for length measurements
- Display of the absolute encoder value without considering an offset

Note!

For SSL encoders, the LS_SsiEncoderX8 system block must be used. Use of an SSI encoder at X8 (255)

6.3.8.1 System block "LS_EncoderX8"

The **LS_EncoderX8** system block provides the input X8 to the application in the function block editor.



Outputs

Identifier DIS code | data type	Value/meaning
ENC_dnEncoderOut1 C02762 |DINT	Display of the current encoder position (steps) within one revolution • 1 revolution ≡ 2^{32} bits **Note:** In order to convert the encoder information/position into a position_p in the internal measuring system, connect both outputs *ENC_dnEncoderOut1* and *ENC_dnEncoderOut2* with the inputs *dnEncoderIn* and *dnEncoderIn2* of the FB **L_EsEncoderConv**. A storage with mains failure protection of the position signal is also processed via this FB.
ENC_dnEncoderOut2 C02763 |DINT	Display of all revolutions of the encoder (only with Multiturn) • After the max. presentable revolutions have been reached, the value jumps back to "0". • C02761 shows the max. presentable revolutions of the MultiTurn encoder (encoder-dependent). • In case of SingleTurn, the value "0" is always output.

Identifier DIS code | data type	Value/meaning	
ENC_dnActualSpeed_s C02764 | DINT	Current encoder speed in [rpm]	
ENC_bError C02765 | BOOL	Status signal "Encoder error"	
	TRUE	An encoder error has occurred.

6.3.8.2 Activate evaluation

C02760 serves to activate the evaluation of the encoder signal of input X8.

- When the evaluation is activated, the encoder parameterised in C00422 is read in.
  - At the same time, the monitoring functions are active. If no encoder is available, the corresponding monitoring functions are triggered.
- When the evaluation is deactivated, the outputs of the system block are reset.
  - Monitoring is deactivated depending on the position encoder selection (C00490) and the motor encoder selection (C00495).

Monitoring

The monitoring functions depend on the encoder type selected inC00422 and do not differ from the existing monitoring functions:

- Open circuit of encoder (response: C00580)
- Encoder angular drift monitoring (📖 263)
- Encoder communication error (*FDB_bEncoderComError*; response: C00601)
- Sine/cosine encoder error (*FDB_bSinCosSignalError*)
- Group signal for errors as process date (*ENC_bError*)

Conditioning of the encoder signal

- The encoder signal is conditioned to a position (including a storage with mains failure protection) within the application using the FB **L_EsEncoderConv**:
  - Interconnection of the *ENC_dnEncoderOut1* output signal with the *dnEncoderIn* input of the FB **L_EsEncoderConv**.
  - Interconnection of the *ENC_dnEncoderOut2* output signal with the *dnEncoderIn2* input of the FB **L_EsEncoderConv**.
  - Additional parameter setting of the FB **L_EsEncoderConv**:
    Mode selection: Cxxxxx = 1
    Number of revolutions transmitted from C02761
  - The (optional) reconstruction of the position after mains switching is also made by the FB **L_EsEncoderConv**.
- In contrast, the conditioning of the encoder signal to a speed is directly made in the LS_EncoderX8 system block.
  - The current encoder speed is provided at the *ENC_dnActualSpeed_s* output in [rpm] (display parameter: C02764).
- TouchProbe function is not supported (continues to be only available for motor and position encoders).

- If the encoder at X8 is simultaneously used as motor or position encoder, the "raw value" of the encoder is continued to be output.

Display parameters

Parameter	Information	Lenze setting	
		Value	Unit
C02761	Resolution Multiturn	-	Rev
C02762	Encoder position: Steps within one revolution	-	Steps
C02763	Encoderrev: Number of revolutions	-	Rev
C02764	Encoderspeed	-	rpm
C02765	Encoder error	-	
Highlighted in grey = display parameter			

6.3.9 Resolver error compensation

This function extension is available from software version V7.0!

Resolver errors typically occur in form of the 1st and 2nd harmonic. They have two different causes:

1. The inductances of the sine and cosine track of the resolver have slightly different values.
2. The sine and cosine track do not magnetise orthogonally to each other.

The resolver error which is based on cause 1 can be corrected by adapting the gains of the digital/analog converters which feed the resolver tracks. In the Lenze setting, the gains for both resolver tracks are preset identically.

A resolver error based on cause 2, can be compensated by a slight correction of the angle which serves to feed the two resolver tracks relatively to each other.

By executing the device command C00002 = "59: Resolver error identification", the gain of the digital/analog converter for feeding the resolver and the angle which serves to feed the two resolver tracks relatively to each other are corrected so that the resolver error is minimised.

- A precondition for the execution of the device command is that the machine is in speed-controlled operation (servo control). The speed amount during the identification must be constant and higher than 500 rpm.
- After the resolver error identification has been executed successfully, the resolver error compensation is activated automatically (C00418 = "1: Activated"). Now the resolver operates with the following resolver error parameters which have been identified during the procedure:

Parameter	Information	Lenze setting	
		Value	Unit
C02862/1	Resolver: Cos gain	100	%
C02862/1	Resolver: Sin gain	100	%
C02863	Resolver: Angle correction	0	

- The detected gain can take values between 0 ...100 %.
  - With a setting of 0 %, the gain of the corresponding resolver track is only 95 % of the default setting.
  - With a sensible resolver error compensation only one of the two gains is adapted. The other remains at 100 %.
- For a permanent acceptance of the identified resolver error parameters, the parameter set must be saved (C00002 = "11: Save start parameters").
- When the resolver error compensation is deactivated (C00418 = "0: Deactivated"), the resolver operates with the Lenze setting again. The identified resolver error parameters remain stored.

The resolver error identification can fail due to the following:

- Wrong control mode is active (no servo control)
- Error or fault is active
- Another identification is active
- The speed is too low (< 500 rpm)
- Time-out while the algorithm is processed

6.3.10 Encoder angular drift monitoring

The optional encoder angular drift monitoring monitors a potential deviation between the actual encoder angle and the angle calculated by the counting of increments in the encoder evaluation.

The encoder angular drift monitoring is activated by parameterising an error response that is not "0: No response" in C00621.

If a deviation greater than 45° (electrical) is detected while monitoring is activated:

- The error message "Encoder monitoring: pulse deviation detected" is entered in the logbook of the controller.
- The error response parameterised in C00621 is triggered.
- The "Reference known" status of the basic drive function "Homing" is reset (provided that this status was set before).

Tip!

A deviation can for instance occur by incorrect parameter setting of the encoder increments, by additional increments due to EMC interferences, or by an EMC-related loss of increments.

The encoder angular drift monitoring is implemented for encoders with and without absolute information by two different principles which are explained in detail in the following subchapters.

6.3.10.1 Angular drift monitoring for encoders without absolute information

When an encoder without absolute information is used, the number of increments between two zero pulses (one revolution) is monitored. This value must equal the encoder increments set in C00420.

Note!

After mains switching, the monitoring function is only active after the second zero pulse, since the first difference in increments to be used can only be calculated with the second and first zero pulse.

When the motor (and thus the encoder) is replaced, an angular drift error is likely to occur within the first revolution after the encoder error is acknowledged, since the monitoring function cannot recognise that the encoder has been replaced.

6.3.10.2 Angular drift monitoring for encoders with absolute information

When an encoder with absolute information is used, cyclic communication with the encoder takes place and the angle is read out digitally. This angle is compared with the angle from the encoder evaluation.

Note!

If monitoring is deactivated (C00621 ="0: No response"), there is no cyclic communication with the encoder, and therefore no communication errors with the encoder can occur.

If monitoring is activated, it is only carried out for speeds below 100 rpm due to runtimes for communication .

- If increments get lost at higher speeds, this deviation can only be detected if the speed is below 100 rpm for at least 80 ms.

After every detected angular drift error of the encoder, the position is automatically read out again and this angle is written into the encoder evaluation. This makes it possible to acknowledge the error. When synchronous machines are used, the pole position is corrected again simultaneously.

7 Braking operation

The 9400 HighLine controller as a single-axis controller (Single Drive) is provided with an integrated brake transistor.

- The required brake resistor must be connected externally (see Mounting Instructions/ Hardware Manual).
- The rated values for the internal brake transistor are given in the Hardware Manual in the "Rated data" chapter.

Stop!

If the brake resistor actually connected is smaller than the brake transistor parameterised, the brake chopper can be destroyed!

The brake resistor can be overloaded thermally. Carry out protective measures suitable for the installation, e.g.:

- Parameterisation of an error response in C00574 and evaluation of the parameterised error message within the application or within the machine control. I2t utilisation - brake resistor (269)
- External wiring using the temperature contact on the brake resistor (e.g. interruption of the supply via mains contactor and activation of the mechanical brakes).

Note!

The brake chopper control is also guaranteed if, for example, the application stands still or the 24-V supply is not connected and the controller is only fed by the DC bus.

7.1 Parameter setting

Short overview: Parameters for braking operation

Parameter	Information	Lenze setting	
		Value	Unit
C00129	Brake resistor value	180.0	Ohm
C00130	Rated power of brake resistor	5600	W
C00131	Rated heat quantity of brake resistor	485	kWs
C00133	Ref.: Brake chopper utilisation	Minimum resistance (C00134)	
C00134	Min. brake resistance	-	Ohm
C00137	Brake transistor utilisation	-	%
C00138	Brake resistor utilisation	-	%
C00173	Mains voltage	400/415 V	
C00181	Reduced brake chopper threshold	0	V
C00569	Resp. brake trans. ixt > C00570	Warning	
C00570	Warning thres. brake transistor	90	%
C00571	Resp. brake res. i2t > C00572	Warning	
C00572	Warning thres. brake resistor	90	%
C00573	Resp. to brake transistor overload	No response	
C00574	Resp. to overtemp. brake resist.	No response	
C00600	Resp. to DC bus overvoltage	Trouble	
Highlighted in grey = display parameter			

7.1.1 Setting the voltage threshold for braking operation

The voltage threshold for the braking operation is set via C00173 (mains voltage) and C00181 (reduced brake chopper threshold). If the brake chopper threshold in the DC bus is exceeded, the brake transistor is switched on.

Mains voltage selected in C00173	Effective brake chopper threshold
230 V	390 V - value in C00181 (0 ... 100 V)
400/415 V	725 V - value in C00181 (0 ... 100 V)
460/480 V	765 V - value in C00181 (0 ... 100 V)
500 V	790 V - value in C00181 (0 ... 100 V)

7.2 Monitoring

7.2.1 Overcurrent protection

The brake chopper hardware is monitored with regard to overcurrent (short circuit or earth fault).

Note!

The monitoring with regard to overcurrent can only be triggered if a braking current is actually available. It is not possible to carry out a test in idle state (without connected brake resistor).

- If monitoring responds:
  - The brake chopper is switched off immediately.
  - The "Fault" response is activated.
  - The "Brake transistor: overcurrent" error message is entered into the logbook of the controller.

Note!

The error can only be acknowledged after 2 seconds at the earliest to resume the braking operation.

Tip!

In addition to the overcurrent protection the controller is provided with two further monitoring functions for the braking operation, which are also activated if no brake resistor is connected at all (testing mode for checking the parameterisation):

▸ Ixt utilisation - brake transistor (📖 268)

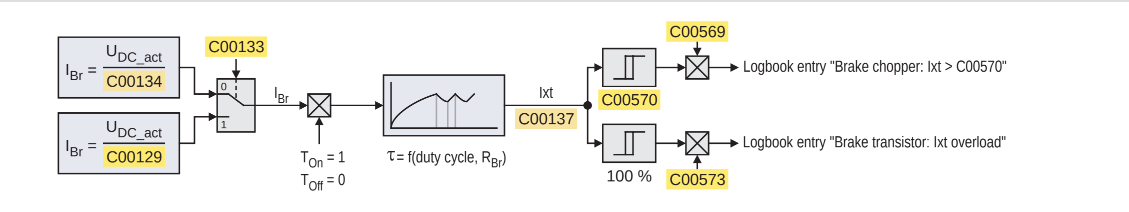
▸ I2t utilisation - brake resistor (📖 269)

7.2.2 Ixt utilisation - brake transistor

The controller is provided with a monitoring function for the Ixt utilisation of the internal brake transistor.

Note!

The braking operation will never be switched off by this monitoring function.



[7-1] Signal flow of Ixt utilisation - brake chopper

- Monitoring is based on a mathematical model which calculates the braking current from the current DC-bus voltage and the brake resistance parameterised.
  - Hence, monitoring can be activated although no brake resistor is connected and can therefore also be used for a testing mode to check the parameterisation.
- During the calculation the thermal utilisation of the brake transistor is taken into consideration by the use of an accordingly adapted time constant.
- In C00133 it can be selected whether the minimum brake resistance (display in C00134) which depends on the network setting in C00173) or the brake resistor value parameterised in C00129 is to be used as a reference for calculating the utilisation.
- C00137 displays the calculated utilisation of the brake transistor in [%].
  - A 100 % utilisation corresponds to the continuous braking power which is provided by the integrated brake chopper at a DC-bus voltage of 790 V (or 390 V at a mains voltage of 230 V).
  - The maximum braking power (assuming that the utilisation starts at 0 %) can be provided for a time period depending on the device.
  - The calculated utilisation is provided as oscilloscope signal *Common.dnIxtBrakeChopper* to check the braking operation while the system is running (scaling: $2^{30} \equiv 100$ %).
- If the utilisation exceeds the advance warning threshold set in C00570, "Brake chopper: Ixt > C00570" is entered in the logbook and the response set in C00569 (default setting: "Warning") is activated.
- When the utilisation reaches the limit value (100 %):
  - The activation of the brake chopper is reset to the permanently permissible mark-to space ratio (taking the parameterised brake resistance into consideration). (The brake chopper is activated with 4 kHz, which means that it can be switched on/off at minimum intervals of 250 µs.)
  - The response set in C00573 (default setting: "No response") is activated with the corresponding effects on the state machine and the inverter.

Note!

If the DC-bus voltage exceeds the overvoltage threshold due to a too high braking energy, the monitoring function for overvoltage in the DC bus responds.

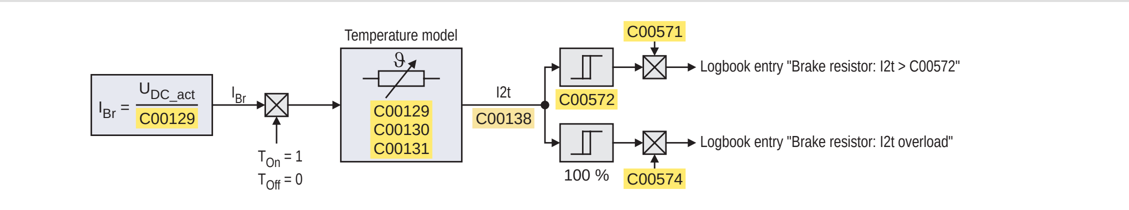
▶ DC bus overvoltage (📖 271)

7.2.3 I2t utilisation - brake resistor

The controller is provided with a monitoring function of the I^2t utilisation of the brake resistor which is proportional to the converted braking power.

Note!

The braking operation will never be switched off by this monitoring function.



[7-2] Signal flow - I2t utilisation - brake resistor

- The monitoring function is based on the mathematical model which calculates the braking current from the current DC-bus voltage and the brake resistance parameterised in C00129.
  - Hence, monitoring can be activated although no brake resistor is connected and can therefore also be used for a testing mode to check the parameterisation.
- The calculation considers the thermal utilisation of the brake resistor based on the following parameters:
  - Resistance value (C00129)
  - Continuous power (C00130)
  - Thermal capacity (C00131)
- C00138 displays the calculated utilisation of the brake resistor in [%].
  - A 100 % utilisation corresponds to the continuous power of the brake resistor which results at the maximum permissible temperature limit of the brake resistor.
  - The calculated utilisation is provided as oscilloscope signal *Common.dnI2tBrakeResistor* to check the braking operation while the system is running (scaling: $2^{30} \equiv 100$ %).
- If the utilisation exceeds the advance warning threshold set in C00572, "Brake resistor: I2t > C00572" is entered in the logbook and the response set in C00571 (default setting: "Warning") is activated.

- When the utilisation reaches the limit value (100 %):
  - The response set in C00574 (default setting: "No response") is activated with the corresponding effects on the state machine and the inverter.
  - Only applies to software versions lower than V3.0:

    The activation of the brake chopper is reset to the permanently permissible mark-to space ratio (taking the parameterised brake resistance into consideration). (The brake chopper is activated with 4 kHz, which means that it can be switched on/off at minimum intervals of 250 µs.)

Stop!

The brake resistor can be overloaded thermally. Carry out protective measures suitable for the installation, e.g.:

- Parameterisation of an error response in C00574 and evaluation of the parameterised error message within the application or within the machine control.
- External wiring using the temperature contact on the brake resistor (e.g. interruption of the supply via mains contactor and activation of the mechanical brakes).

Note!

If the system is dimensioned correctly, this monitoring should not respond. If individual rated data of the actually connected brake resistor are not known, they have to be determined "empirically".

7.2.4 DC bus overvoltage

If, due to a too high braking energy, the DC-bus voltage exceeds the overvoltage threshold which results from the mains voltage setting in C00173, the "Overvoltage in the DC bus" error message is output and the response set in C00600 is activated (default setting: "Trouble").

Note!

For hoist applications, the "Fault" response should be selected in C00600 (in combination with an emergency stop via mechanical brakes).

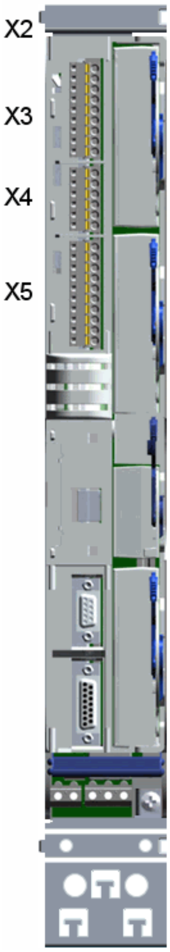
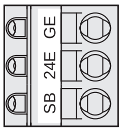
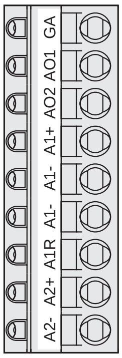
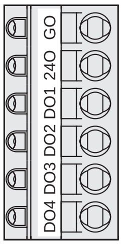
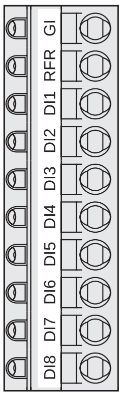
8 I/O terminals

This chapter provides information about options for parameter setting and configuration of the controller input and output terminals.

Tip!

Information on wiring the terminals can be found in the Mounting Instructions for the controller!

8.1 Overview

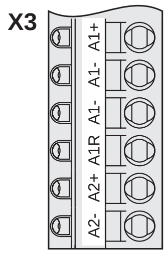
Front view	Terminal assignment	Information
	X2 	▸ "State bus" monitoring function (📖 283)
	X3 	▸ Analog inputs (📖 274) ▸ Analog outputs (📖 277)
	X4 	▸ Digital outputs (📖 281)
	X5 	▸ Digital inputs (📖 279) ▸ Touch probe detection (📖 285)

8.2 Analog inputs

The controller has two analog inputs that can be used to detect differential voltage signals in the range of ±10 V, e.g. analog speed setpoint selections or the voltage signals of an external sensor (temperature, pressure, etc.).

- ▶ Optionally, analog input 1 can also be used to detect current setpoints.

8.2.1 Terminal assignment/electrical data



Terminal	Use	Electrical data	
X3/A1- X3/A1+	Differential voltage input 1 (no jumper between A1R and A1-	Level:	-10 V ... +10 V
		Resolution:	11 bits + sign
		Scaling:	When C00034 = "0": ±10 V ≡ $\pm 2^{30}$
		Conversion rate:	1 kHz
	Current input (jumper between A1R and A1-)	Level:	-20 mA ... +20 mA
		Resolution:	10 bits + sign
		Scaling:	When C00034 = "1": -20 mA ... -4 mA = -2^{30} ... 0 +4 mA ... +20 mA = 0 ... 2^{30}
			When C00034 = "2": ±20 mA ≡ $\pm 2^{30}$
		Conversion rate:	1 kHz
X3/A2- X3/A2+	Differential voltage input 2	Level:	-10 V ... +10 V
		Resolution:	11 bits + sign
		Scaling:	±10 V ≡ $\pm 2^{30}$
		Conversion rate:	1 kHz

8.2.2 Parameter setting

Short overview of parameters for the analog inputs:

Parameter	Information
C00034	Config. analog input 1
C00598	Resp. to open circuit AIN1
C02730/1	Analog input 1: Gain
C02730/2	Analog input 2: Gain
C02731/1	Analog input 1: Offset
C02731/2	Analog input 2: Offset
C02732/1	Analog input 1: Dead band
C02732/2	Analog input 2: Dead band
C02800/1	Analog input 1: Input signal (-16384 ≡ -100 %, 16383 ≡ 100 %)
C02800/2	Analog input 2: Input signal (-16384 ≡ -100 %, 16383 ≡ 100 %)
Highlighted in grey = display parameter	

8.2.3 Reconfiguring analog input 1 into current input

By means of the following two steps, analog input 1 can be reconfigured into a current input:

1. Bridge the terminals A1R and A1- at terminal strip X3 by means of wiring.
2. Select the corresponding current loop under C00034.

Tip!

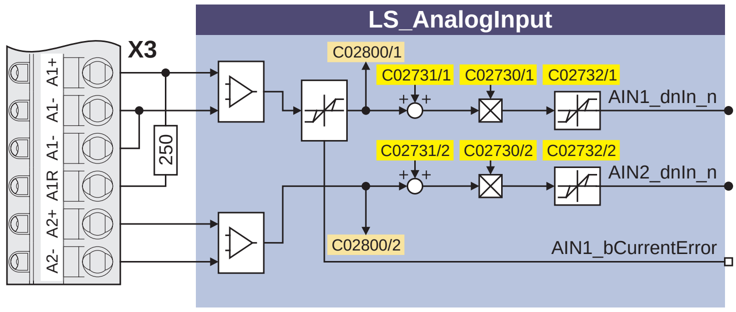
Like this you can implement a 4 ...20 mA current loop, e.g. for speed setpoint selection.

Open-circuit monitoring

Under C00598 you can set an error response to open circuit for the 4 ...20 mA current loop.

8.2.4 "LS_AnalogInput" system block

The **LS_AnalogInput** system block displays the analog inputs in the function block editor.



Output Data type	Value/meaning	
AIN1_dnIn_n DINT	Analog input 1 • Scaling: $\pm 2^{30} \equiv \pm 10$ V for use as a voltage input $\pm 2^{30} \equiv \pm 20$ mA for use as a current input	
AIN2_dnIn_n DINT	Analog input 2 • Scaling: $\pm 2^{30} \equiv \pm 10$ V	
AIN1_bCurrentError BOOL	Status signal "Current input error" • Only when analog input 1 is used as current input. • Application: Monitoring of the 4 ...20 mA circuit with regard to cable break.	
	TRUE	$\lvert I_{AIN1} \rvert$ < 2 mA

8.3 Analog outputs

The controller has two analog outputs that can be used to output internal analog signals as voltage signals, e.g. for the control of analog indicating instruments or as a setpoint for slave drives.

Note!

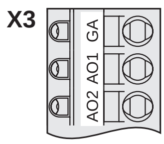
Initialisation behaviour:

- After mains switching until the application is started, the analog outputs remain on 0 V.

Exceptional behaviour:

- In the case of a critical exception within the application (e. g. reset), the analog outputs are set to 0 V.

8.3.1 Terminal assignment/electrical data



Terminal	Use	Electrical data	
X3/AO1	Voltage output 1	Level:	-10 V ... +10 V (max. 2 mA)
		Resolution:	11 bits + sign
		Scaling:	$\pm 2^{30} \equiv \pm 10$ V
		Conversion rate:	1 kHz
X3/AO2	Voltage output 2	Level:	-10 V ... +10 V (max. 2 mA)
		Resolution:	11 bits + sign
		Scaling:	$\pm 2^{30} \equiv \pm 10$ V
		Conversion rate:	1 kHz
X3/GA	Reference potential (analog ground)		

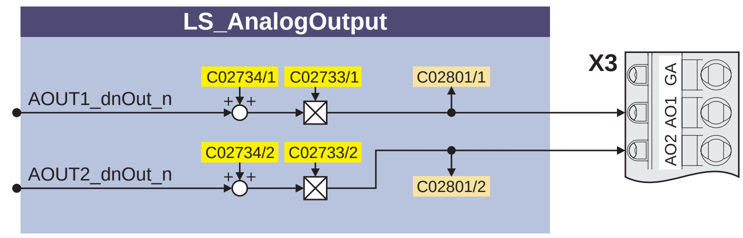
8.3.2 Parameter setting

Short overview of parameters for the analog outputs:

Parameter	Information
C02733/1	Analog output 1: Gain
C02733/2	Analog output 2: Gain
C02734/1	Analog output 1: Offset
C02734/2	Analog output 2: Offset
C02801/1	Analog output 1: Output signal
C02801/2	Analog output 2: Output signal
Highlighted in grey = display parameter	

8.3.3 "LS_AnalogOutput" system block

In the function block editor the **LS_AnalogOutput** system block provides the interface to the analog outputs.



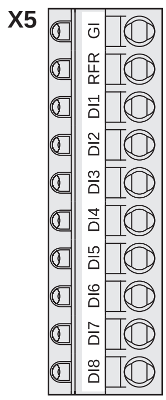
Input Data type	Information/possible settings
AOUT1_dnOut_n DINT	Analog output 1 • Scaling: $\pm 2^{30} \equiv \pm 10$ V
AOUT2_dnOut_n DINT	Analog output 2 • Scaling: $\pm 2^{30} \equiv \pm 10$ V

8.4 Digital inputs

The controller is provided with eight freely configurable digital inputs.

- All digital inputs can be used for touch probe. Touch probe detection (285)
- The control input RFR of terminal strip X5 for controller enable is fixedly connected to the device control.

8.4.1 Terminal assignment/electrical data



Terminal	Use	Electrical data	
X5/DI1 X5/DI8	Digital input 1 ... 8	LOW level:	0 ... +5 V
		HIGH level:	+15 ... +30 V
		Input current:	8 mA per input (at 24 V)
		External-voltage protection:	Max. ±30 V
		Conversion rate:	1 kHz
X5/RFR	Controller enable	See digital inputs	
X5/GI	Reference potential (digital ground)		

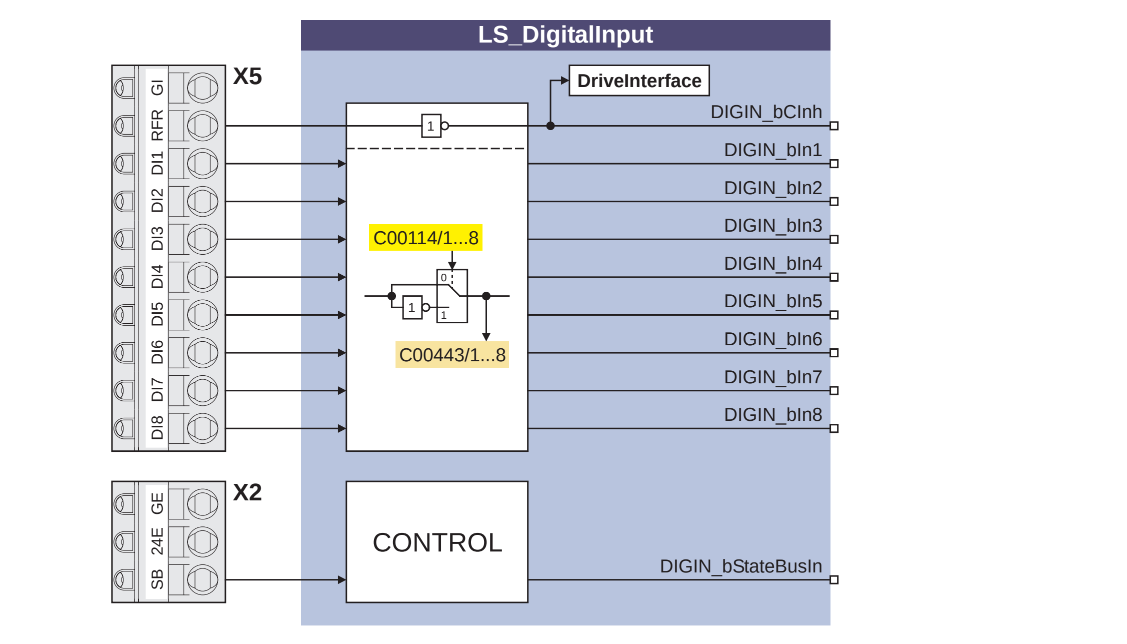
8.4.2 Parameter setting

Short overview of parameters for the digital inputs:

Parameter	Information
C00114	Digital input x - terminal pol.
C00443	Status: Digital inputs
C02803	Status word: Digital inputs
C02830	Digital inputs: Delay time
Highlighted in grey = display parameter	

8.4.3 "LS_DigitalInput" system block

The **LS_DigitalInput** system block displays the digital inputs and the status of the state bus in the function block editor.



Output DIS code | data type	Value/meaning	
DIGIN_bCInh C00443/9 | BOOL	Status signal "Controller inhibit" • The control input RFR (X5/pin 9) for setting/deactivating controller inhibit is fixedly connected to the device control (DCTRL) via an inverter.	
	TRUE	Controller inhibit active
DIGIN_bIn1 C00443/1 | BOOL ... DIGIN_bIn8 C00443/8 | BOOL	Digital input 1 ... 8	
DIGIN_bStateBusIn C00443/12 | BOOL	State bus status ▸ "State bus" monitoring function (📖 283)	
	TRUE	A node connected to the state bus has set the state bus to LOW level and the "Error" state has been set. • The "Error" state is also set if a node connected to the state bus is not supplied with voltage.

8.5 Digital outputs

The controller is provided with four freely configurable digital outputs.

Note!

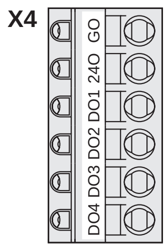
Initialisation behaviour:

- After mains switching until the application is started, the digital outputs remain on FALSE.

Exceptional behaviour:

- In the case of a critical exception within the application (e. g. reset), the digital outputs are set to FALSE, taking the terminal polarity parameterised in C00118 into consideration.

8.5.1 Terminal assignment/electrical data



Terminal	Use	Electrical data	
X4/DO1 ... X4/DO4	Digital output 1 ... 4	LOW level:	0 ... +5 V
		HIGH level:	+15 ... +30 V
		Output current:	Max. 50 mA per output (external resistance > 480 Ω at 24 V)
		Conversion rate:	1 kHz
X4/24O	External 24 V voltage supply for the digital outputs		
X4/GO	Reference potential (digital ground)		

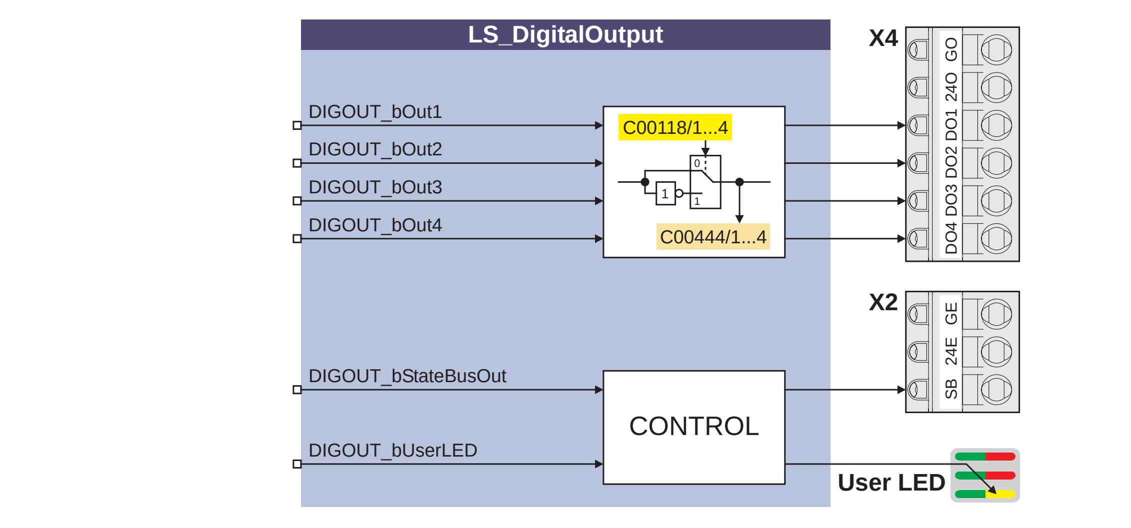
8.5.2 Parameter setting

Short overview of parameters for the digital outputs:

Parameter	Information
C00118	Digital output x - terminal pol.
C00444	Status: Digital outputs
C02802	Status word: Digital outputs
Highlighted in grey = display parameter	

8.5.3 "LS_DigitalOutput" system block

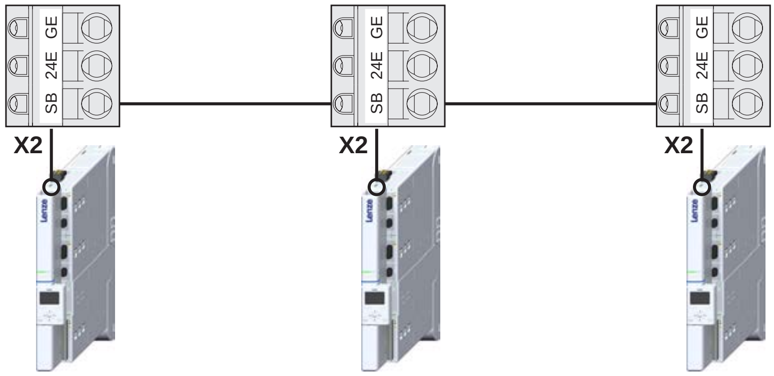
In the function block editor the **LS_DigitalOutput** system block provides the interface to the digital outputs, the state bus, and the yellow user LED at the front of the controller.



Input DIS code | data type	Information/possible settings	
DIGOUT_bOut1 C00444/1 | BOOL ... DIGOUT_bOut4 C00444/4 | BOOL	Digital output 1 ... 4	
DIGOUT_bStateBusOut C00444/18 | BOOL	Setting the state bus to the "Error" state "State bus" monitoring function (283)	
	TRUE	The state bus is set to LOW level, all nodes connected to the state bus start their pre-programmed response.
DIGOUT_bUserLED C00444/9 | BOOL	Control of yellow user LED on the front of the controller	
	TRUE	LED on

8.6 "State bus" monitoring function

The state bus is a bus system that is solely designed for Lenze controllers, via which up to 20 controllers can be connected to each other, and by means of which the function of a "release cord" can be simulated:



X2/SB = State bus In/Out (reference potential GE)

[8-1] Schematic diagram: Networking via state bus

- The state bus only knows the states "OK" and "Error".
- The state bus is a bus with multi-master capability, i.e. each node connected to the state bus can set the state bus to the "Error" state by setting it to LOW level.
- In the "Error" status, all nodes start their adjustable response, e.g. synchronised braking of the drive system.
- The "Error" state is also set if a node connected to the state bus is not supplied with voltage.

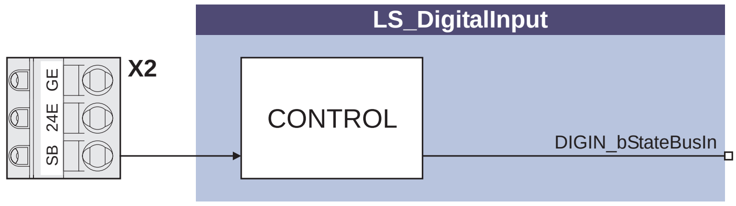
Note!

Exceptional behaviour:

- In the case of a critical exception within the application (e. g. reset), the "release cord" is not triggered, the state bus remains in the "OK" status.

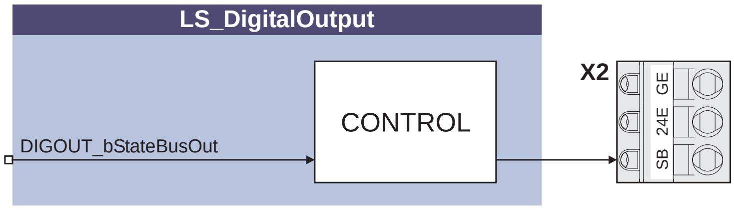
8.6.1 Detecting the current state

Via the output *DIGIN_bStateBusIn* of the LS_DigitalInput system block, the current status of the state bus can be queried. In case of error the output *DIGIN_bStateBusIn* is set to TRUE.



8.6.2 Setting the state bus to the "Error" state

If the input *DIGOUT_bStateBusOut* of the LS_DigitalOutput system block is set to TRUE, the state bus is set to "Error" and all connected nodes start their pre-programmed response.



8.7 Touch probe detection

A "Touch probe" is an event which can for instance be actuated in an edge-controlled manner via a digital input to detect an actual value (that changes quickly) at the time of activation and to process it further within the program afterwards.

Overview of the touch probe channels

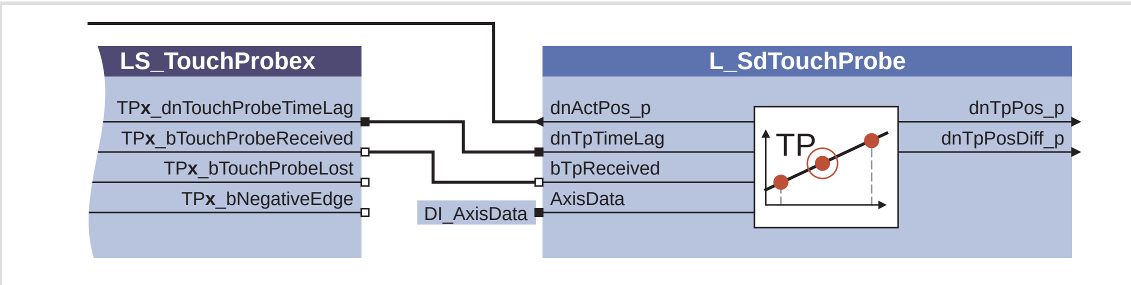
For the touch probe detection 12 touch probe channels are provided, which can be configured independently of each other:

Touch probe channel	Activating event	System block
1	Edge change at digital input 1	LS_TouchProbe1...8 (288)
2	Edge change at digital input 2	
3	Edge change at digital input 3	
4	Edge change at digital input 4	
5	Edge change at digital input 5	
6	Edge change at digital input 6	
7	Edge change at digital input 7	
8	Edge change at digital input 8	
9	Motor encoder zero pulse	LS_TouchProbeMotor (289)
10	Position encoder zero pulse	LS_TouchProbeLoad (289)
11	DFIN zero pulse	LS_TouchProbeDFIN
12	DFOUT zero pulse	LS_TouchProbeDFOUT

- Each touch probe channel is assigned to a system block which provides the application with a scaled time stamp.
- The time stamp refers to the sampling time of the encoder signals and outputs the difference with regard to the touch probe event.

Further processing of the touch probe

For further processing of the touch probe event the time stamp is to be transmitted to an instance of the **L_SdTouchProbe** FB:

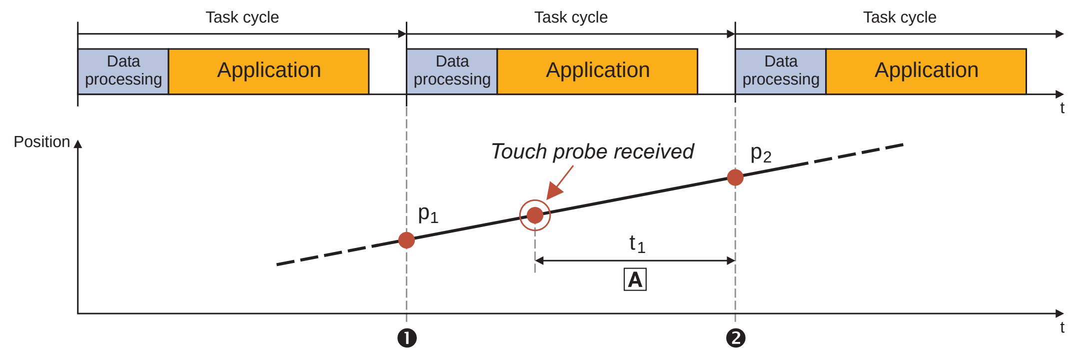


[8-2] Transfer of the time stamp to the L_SdTouchProbe FB

- The **L_SdTouchProbe** FB takes over the interpolation of the input signal on the basis of the time stamp and outputs the interpolated value and the difference to the last input signal.

8.7.1 Actual value interpolation (principle)

If a touch probe is detected, the (remaining) time until the following task cycle is determined and from this a time stamp is generated. On the basis of this time stamp, the **L_SdTouchProbe** FB can then carry out a linear interpolation between the two actual position interpolation points; the result is the precise actual position at the time of the physical touch probe event.



❶ Actual position point 1

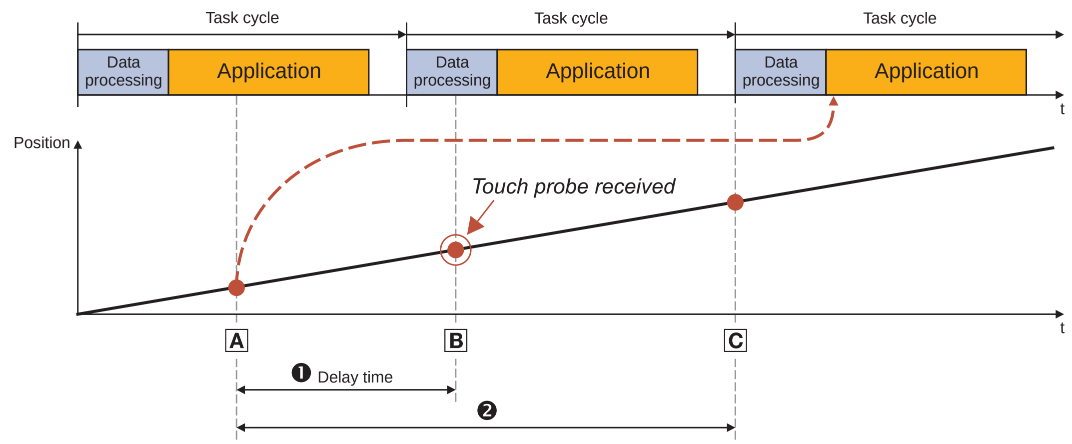
❷ Actual position point 2

A Residual time (*dnTouchProbeTimeLag*)

[8-3] Actual value determination through linear interpolation (principle)

8.7.2 Dead time compensation

For dead time compensation during the detection of the touch probe event, it is possible to select a delay time (*Touch probe delay*) in C02810 for each touch probe channel, which will be considered in the touch probe calculation.



🄰 Mechanical event: Switch is closed

🄱 Electrical event: Closed switch contact is registered
- Can for example be delayed by flying times of the mechanical contact or filter.

🄲 Value and position determination for the real (mechanical) event, the determined values are transferred to the application

❶ Delay time (dead time) between mechanical and electrical event
- This dead time can be compensated by a corresponding setting in C02810.

❷ Time considered by touch probe calculation
- If more than one task cycle has elapsed, the two actual position values are searched within a history buffer, between which the physical event has taken place.

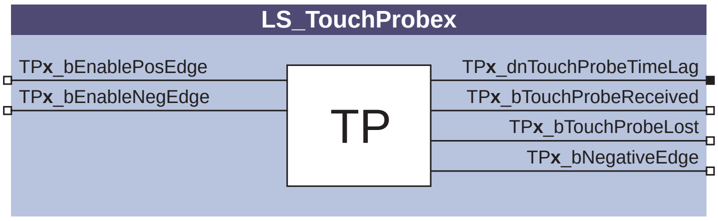
[8-4] Dead time compensation (principle)

▶ The filtering of the digital inputs has an impact on the electrical detection of the touch probe, i. e. the delay time for the digital inputs set in C02830 has to be taken into consideration within the delay time C02810.

▶ For the optional digital frequency input/output the setting of the delay times is effected via separate parameters:
  – C13021 or C14021: TP delay time - digital frequency input.
  – C13061 or C14061: TP delay time - digital frequency output.

8.7.3 "LS_TouchProbe1...8" system block

In the function block editor the **LS_TouchProbe1** ... **LS_TouchProbe8** system blocks display the touch probe channels 1 ... 8 which are assigned to the digital inputs **DI1** ... **DI8**.



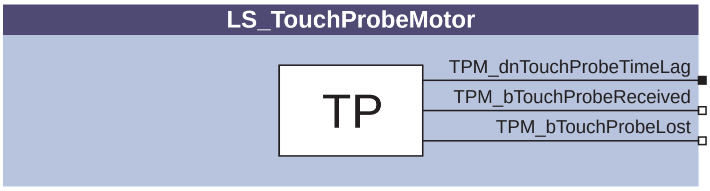
x = 1 ... 8

Input Data type	Value/meaning	
TPx_bEnablePosEdge BOOL	Enable response to positive edge **Note:** • If several positive edges occur within the basic cycle time (HighLine: 1 ms), only the first positive edge initiates the touch probe event and no status signal "touch probe(s) lost" is generated.	
	TRUE	A touch probe event is activated by a positive edge at the digital input DIx.
TPx_bEnableNegEdge BOOL	Enable response to negative edge **Note:** • If several negative edges occur within the basic cycle time (HighLine: 1 ms), only the first negative edge initiates the touch probe event. • If a positive and negative edge occur within the basic cycle time (1 ms), and if the response to both edges is enabled, only the positive edge initiates the touch probe event. • In both cases no status signal "touch probe(s) lost" is generated.	
	TRUE	A touch probe event is activated by a negative edge at the digital input DIx.

Output Data type	Value/meaning	
TPx_dnTouchProbeTimeLag DINT	Scaled time stamp for further processing of the touch probe event with the **L_SdTouchProbe** FB. • 1 ms ≡ 20 bits	
TPx_bTouchProbeReceived BOOL	Status signal "Touch probe detected" • State is only set for one task cycle.	
	TRUE	Touch probe event has been activated.
TPx_bTouchProbeLost BOOL	Status signal "Touch probe(s) lost" • State is only set for one task cycle.	
	TRUE	More than one touch probe event was actuated within the task runtime. The time stamp that is output only refers to the first touch probe event.
TPx_bNegativeEdge BOOL	Status signal "Negative edge detected" • State is only set for one task cycle.	
	TRUE	A negative edge has been detected at the digital input DIx.

8.7.4 "LS_TouchProbeMotor" system block

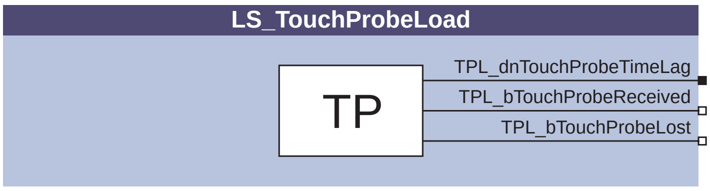
In the function block editor the **LS_TouchProbeMotor** system block represents the touch probe channel that is assigned to the motor encoder zero pulse.



Output Data type	Value/meaning	
TPM_dnTouchProbeTimeLag DINT	Scaled time stamp for further processing of the touch probe event with the **L_SdTouchProbe** FB.	
TPM_bTouchProbeReceived BOOL	Status signal "Touch probe detected" • State is only set for one task cycle.	
	TRUE	Touch probe event has been activated.
TPM_bTouchProbeLost BOOL	Status signal "Touch probe(s) lost" • State is only set for one task cycle.	
	TRUE	More than one touch probe event was actuated within the task runtime and therefore could not be detected anymore.

8.7.5 "LS_TouchProbeLoad" system block

In the function block editor the **LS_TouchProbeLoad** system block represents the touch probe channel that is assigned to the position encoder zero pulse.



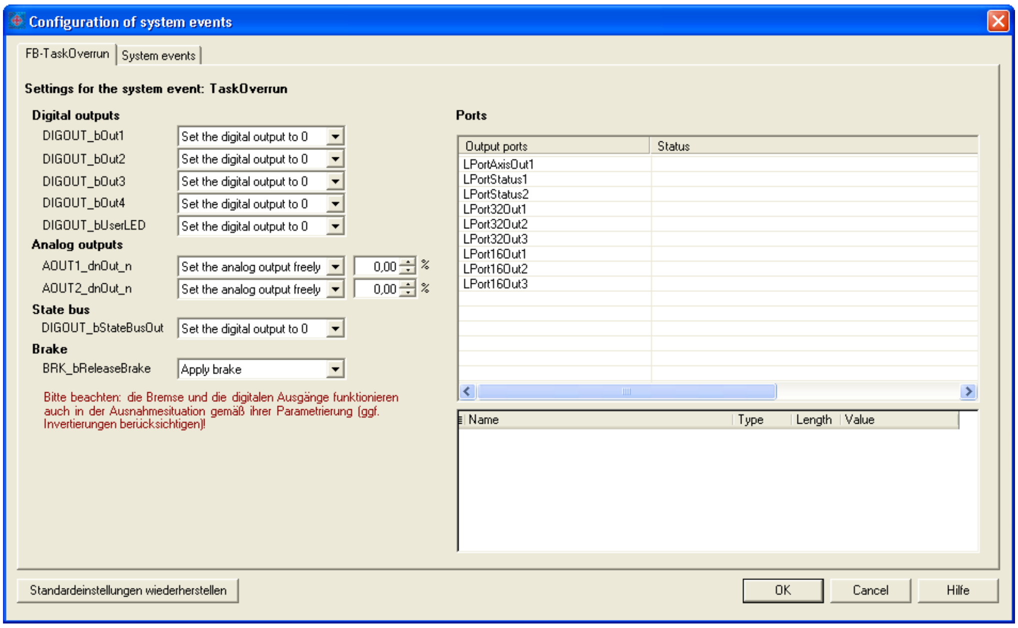
Output Data type	Value/meaning	
TPL_dnTouchProbeTimeLag DINT	Scaled time stamp for further processing of the touch probe event with the **L_SdTouchProbe** FB.	
TPL_bTouchProbeReceived BOOL	Status signal "Touch probe detected" • State is only set for one task cycle.	
	TRUE	Touch probe event has been activated.
TPL_bTouchProbeLost BOOL	Status signal "Touch probe(s) lost" • State is only set for one task cycle.	
	TRUE	More than one touch probe event was actuated within the task runtime and therefore could not be detected anymore.

8.8 Configure exception handling of the outputs

From »Engineer« version 2.10 onwards, the function block editor for the controller can also be used to configure the behaviour of the analog and digital outputs and that of the brake control and the output ports after a task overflow in order to adapt it to the respective application.

How to configure the exception handling:

1. Go to the *Project view* of the »Engineer« and select the 9400 HighLine controller.
2. Change to the **FB-Editor** tab in the *Workspace*.
3. Click on the icon in the *FB editor toolbar* to open the *Configure exception handling* dialog box:



- On the **FB TaskOverrun** tab the behaviour of the controller outputs and that of the output ports defined in the application in the case of a task overflow can be configured.
- On the **FB System Events** tab, the behaviour of the outputs of the drive controller and the application is only displayed and cannot be configured.

4. Carry out the desired configuration.
   - Each output can be configured individually. A free value can be set for the analog outputs (-200.00 ... 200.00 %).
   - If you select an output port in the "Ports" area on the right, all application variables for this output port are shown in the table below. In the "Value" column a value can be set for each application variable to which it is to be set if an event occurs can be specified.
   - If a value has been set for at least one application variable, the status "Exceptional behaviour parameterised" is shown for the corresponding output port.
   - If the **Restore standard settings** button is clicked, the default setting for task overflow is restored. In this case, all output terminals would be set to LOW level or 0 V in the event of a task overflow and the output ports would retain their last value.
5. Click **OK** to accept the configuration and close the dialog box.

Danger!

In case a task overflow occurs, the brake can be configured to "open". This setting should be used with caution as the brake is then forcibly opened and does not close even if the drive controller is inhibited!

Note!

- To render the changes effective within the controller, the project has to be updated, and the changed application has to be transferred to the controller.
- During the reset or download of an application, all output signals are set to LOW level or 0 V for a short time (the state bus, in contrast, is set to HIGH level due to hardware inversion).

See also: Behaviour after task overflow (114)

9 "CAN on board" system bus

The controller is provided with an implemented CANopen system bus interface ("CAN on board") via which, for instance, process data and parameter values can be exchanged between the nodes. In addition, the interface enables the connection of further modules, such as distributed terminals, operator and input devices ("HMIs") or 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!

- The parameters relevant for the CANopen system bus interface are assigned to different subcategories in the parameter list in the »Engineer« and in the keypad in the **CAN** category.
- Information on CAN communication modules and CANopen system bus interfaces of other Lenze devices can be found in the Lenze library in the "CAN" communication manual.

9.1 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 old system bus are lower as compared with CANopen. For parameter setting, two parameter data channels are always available to the user while CANopen provides only one active parameter data channel (along with the possibility to establish further channels).

The system bus (CANopen) of the Servo Drives 9400 has been developed from the system bus (CAN) of the controller of the 9300 series with the following properties:

- Full compatibility according to CANopen DS301 V4.02.
- Support of the NMT master/slave function "Node Guarding" (DS301 V4.02).
- Support of the NMT slave function "Heartbeat" (DS301 V4.02).
- There are no restrictions regarding the selection of the node addresses.
- Number of parameterisable server and client SDO channels:
  - max. 10 channels with 1 ... 8 bytes
- Number of parameterisable PDO channels:
  - max. 4 Transmit-PDOs (TPDOs) with 1 ... 8 bytes
  - max. 4 Receive-PDOs (RPDOs) with 1 ... 8 bytes
- All PDO channels are functionally equivalent.
- Monitoring of the RPDOs for data reception
- Telegram counters for SDOs and PDOs
- Bus status diagnostics
- Boot-up telegram generation
- Emergency telegram generation
- Reset node telegram generation (with master configuration).
- Sync telegram generation and response to sync telegrams:
  - Transmit/receive data
  - Synchronisation of the device-internal time base
- Abort codes
- All "CAN on board" functions can be parameterised via codes.
- Object directory (all mandatory functions, optional functions, indexes)

9.1.1 General data and operating conditions

Field	Values
Communication profile	CANopen (DS301 V4.02)
Communication medium	DIN ISO 11898
Network topology	Line closed on both sides (e.g. termination by Sub-D plug, type EWZ0046)
Adjustable node addresses	1 ... 127 • Adjustable via DIP switch on the memory module (exception: memory module MM1xx) or via code C00350.
Max. number of nodes	127
Baud rate	10, 20, 50, 125, 250, 500, 800, 1000 kbit/s or automatic recognition • Adjustable via DIP switch on the memory module (exception: memory module MM1xx) or via code C00351.
Process data	• max. 4 TPDOs with 1 ... 8 bytes • max. 4 RPDOs with 1 ... 8 bytes
Parameter data	Max. 10 client and server SDO channels with 1 ... 8 bytes
Transfer mode for TPDOs	• With change of data • Time-controlled, 1 to x ms • After the reception of 1 to 240 sync telegrams

9.1.2 Supported protocols

Category	Protocol
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 (slave)
Monitoring protocols	Node guarding (master and slave)
	Heartbeat (heartbeat producer and heartbeat consumer)

9.1.3 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
- telegram run time (baud rate / telegram length)
- bus load (especially if the bus is loaded with PDOs and SDOs at a low baud rate.)

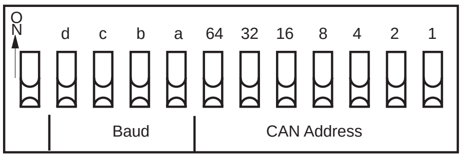
Servo Drives 9400 processing time

Parameter data and process data are independent of each other.

- Parameter data:
  - For controller-internal parameters: approx. 30 ms ± 20 ms tolerance (typically)
  - For some codes the processing time can be longer.
- Process data are transported in real time.

9.2 Possible settings by DIP switch

The DIP switches on the front of the memory serve to set the baud rate and the node address.



[9-1] DIP switch

9.2.1 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.
- Valid address range: 1 ... 127

Note!

- The addresses of the nodes must differ from each other.
- All twelve DIP switches = OFF (Lenze setting):
  - At switching on, the settings under code C00350 (node address) and C00351 (baud rate) will become active.
- Switch the voltage supply of the basic device off and then on again to activate altered settings.

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**						

Tip!

The node address resulting from the setting of the DIP switches at the last mains switching is displayed in C00349/1.

9.2.2 Setting the baud rate

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

Switch positions				Baud rate
d	**c**	**b**	**a**	
OFF	ON	ON	OFF	10 kbps
OFF	ON	OFF	ON	20 kbps
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
OFF	ON	OFF	OFF	1 Mbps
OFF	ON	ON	ON	Automatic recognition

Note!

Switch the voltage supply of the basic device off and then on again to activate altered settings.

Tip!

The baud rate resulting from the setting of the DIP switches at the last mains switching is displayed in C00349/2.

9.3 LED status displays for the system bus

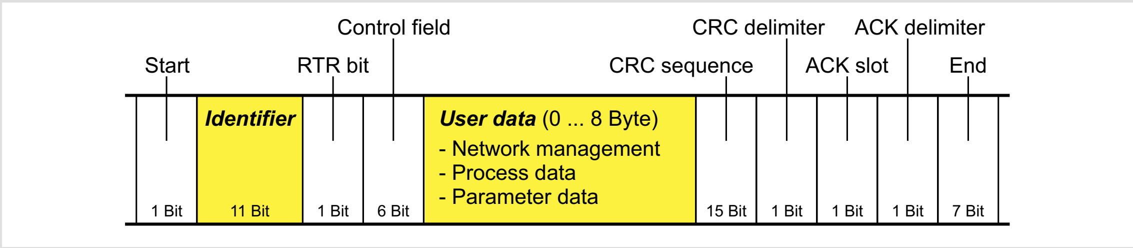
Both upper LEDs LEDs "CAN-RUN" and "CAN-ERR" on the front of the controller inform about the CANopen state and report CANopen errors.

[9-2] LED status displays CAN-RUN and CAN-ERR

The meaning can be obtained from the following table:

LED status display		Meaning	
CAN-RUN	CAN-ERR	CANopen state	CANopen error
Only CAN-ERR is on		-	**Bus off**
CAN-RUN and CAN-ERR jitter		Automatic detection of the baud rate is active.	
CAN-RUN is blinking every 0.2 seconds | CAN-ERR is off		**Pre-operational**	-
CAN-RUN is blinking every 0.2 seconds | CAN-ERR 1 x blinking, 1 s off			**Warning limit reached**
CAN-RUN is blinking every 0.2 seconds | CAN-ERR 2 x blinking, 1 s off			**Node guard event**
Only CAN-RUN is on		**Operational**	-
CAN-RUN on | CAN-ERR 1 x blinking, 1 s off			**Warning limit reached**
CAN-RUN on | CAN-ERR 2 x blinking, 1 s off			**Node guard event**
CAN-RUN on | CAN-ERR 3 x blinking, 1 s off			**Sync message error**
CAN-RUN is blinking every 1 second | CAN-ERR is off		**Stopped**	-
CAN-RUN is blinking every 1 second | CAN-ERR 1 x blinking, 1 s off			**Warning limit reached**
CAN-RUN is blinking every 1 second | CAN-ERR 2 x blinking, 1 s off			**Node guard event**

9.4 Structure of the CAN data telegram



[9-3] Basic structure of the CAN telegram

The identifier and the user data are described in detail in the following subchapters. The other signals refer to the transfer characteristics of the CAN telegram that are not described in this documentation.

9.4.1 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" (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 the user data.

The identifier consists of a so-called 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 be the network configurator) or is firmly allocated.

Node address (node ID)

For purposes of definite identification, a node address, also referred to as "node ID", in the valid address range (1 ... 127) is to be assigned to each node within the system bus network.

- A node address may not be assigned more than once within a network.
- The own node address can be configured via the DIP switch of the memory module (exception: MM1xx memory module) or via code C00350. ▶ Setting the node address (📖 296)

Identifier assignment

The system bus is message-oriented and not node-oriented. Each message has a definite identification, the identifier. In the case of CANopen, a node-orientation is achieved by the fact that for each message there is only one transmitter.

- The basic identifiers for network management (NMT) and Sync as well as the basic SDO channel (SDO1) are specified in the CANopen protocol and cannot be changed.
- The basic identifiers of the PDOs are preset in the Lenze setting 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 (📖 307)

Object		Direction		Basic identifier	
		from the device	to the device	dec	hex
Network management (NMT)				0	0
Sync				128	80
Emergency		●		128	80
PDO1 (Process data channel 1)	TPDO1	●		384	180
	RPDO1		●	512	200
PDO2 (Process data channel 2)	TPDO2	●		640	280
	RPDO2		●	768	300
PDO3 (Process data channel 3)	TPDO3	●		896	380
	RPDO3		●	1024	400
PDO4 (Process data channel 4)	TPDO4	●		1152	480
	RPDO4		●	1280	500
SDO1 (Basic SDO channel)		●		1408	580
			●	1536	600
SDO2 ... SDO10 (Parameter data channel 2 ... 10)		●		1472	5C0
			●	1600	640
Node guarding, heartbeat		●		1792	700

9.4.2 User data

All nodes communicate with each other by exchanging data telegrams via the system bus. The user data area of the CAN telegram either contains network management data, parameter data, or process data:

Network management data

(NMT data)

- Control information on start, stop, reset, etc. of communication to certain or all nodes of the CAN network.

Process data

(PDOs – Process Data Objects)

- Process data are transferred via the process data channel.
- The controller can be controlled using process data.
- Process data are not saved in the controller.
- Process data are transferred between the host and the nodes to ensure a 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.

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 sender 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 in the controller until the mains is switched.
- In general, the parameter transfer is not time-critical.
- Parameter data are, for instance, operating parameters, diagnostic information and motor data.

9.5 Communication phases/network management

Regarding communication via the system bus, the drive distinguishes between the following states:

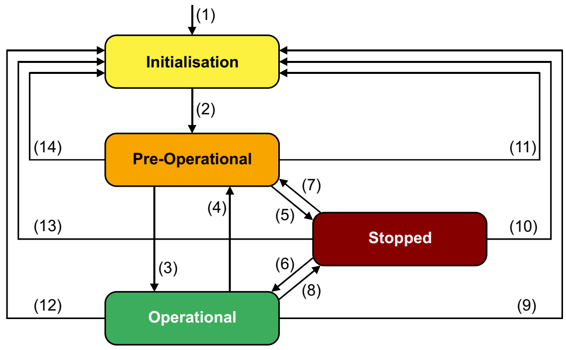
Status	Explanation
"Initialisation" (Initialisation)	After power-up, initialisation is executed. • During this phase, the drive does not take part in the data transfer. • All CAN-relevant parameters are written with their standard values again. • After initialisation has been completed, the controller is automatically set to the "pre-operational" status.
"Pre-operational" (before being ready for operation)	Parameter data can be received, process data is 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!

In every state, the initialisation can be re-executed partly or completely by transmitting appropriate network management telegrams.

9.5.1 State transitions

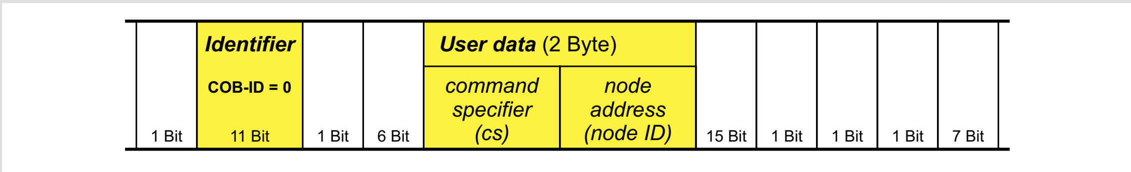


[9-4] NMT state transitions in the CAN network

Transition	NMT command	State after change	Effect on process or parameter data after state change
(1)	-	Initialisation	Initialisation starts automatically when the mains is switched on. • During initialisation, the controller does not take part in the data transfer. • After initialisation has been 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 the communication.
[i]	From here, the states are changed over by the master for the entire network. • A target address contained in the NMT command specifies the receiver(s). • If the 9400 controller has been configured as CAN master, the state automatically changes to "Operational" after the waiting time has expired (C00378) and the NMT command 0x0100 ("Start Remote Node") is sent to all nodes. • Data can only be exchanged via process data objects if the state is "Operational".		
(3), (6)	0x01 xx Start Remote Node	Operational	Network management, sync and emergency telegrams as well as process data (PDO) and parameter data (SDO) are active. Optional: When the state is changed, event and time-controlled process data (PDOs) will be sent once.
(4), (7)	0x80 xx Enter Pre-Operational	Pre-operational	Network management, sync and emergency telegrams as well as 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	Initialisation of all CAN-relevant parameters (CiA DS 301) with the saved values.
(12), (13), (14)	0x82 xx Reset communication		Initialisation of all CAN-relevant parameters (CiA DS 301) with the saved values.
[i]	Meaning of the node address in the NMT command: • xx = 0x00: With this assignment, all nodes are addressed by the telegram (Broadcast telegram). The state can be changed for all nodes at the same time. • xx = Node-ID: If a node address is indicated, the state will only be changed for the node addressed.		

9.5.2 Network management telegram (NMT)

The telegram for the network management contains the identifier "0" and the command included in the user data, which consists of the command byte and the node address.



[9-5] Network management telegram for changing 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 communication phases are changed over by a node, the CAN master, for the entire network. The CAN master can also be a controller. ▶ Parameterising the controller as CAN master (📖 305)

Example:

Data can only be exchanged via the process data objects in the "Operational" state. To change all nodes on the bus from "Pre-Operational" to "Operational" via the CAN master, the following identifiers and user data must be set as follows in the transmit telegram:

- Identifier: 0x00 (network management)
- User data: 0x0100 (NMT command "Start Remote Node" to all nodes)

9.5.3 Parameterising the controller as CAN master

If the initialisation of the system bus and the connected status change from "Pre-operational" to" Operational' is not carried out by a higher-level host system, the controller can be defined to be a "quasi" master to take over this task.

The controller is configured as CAN master in C00352.

- As a CAN master, the controller sets " all nodes on the bus (broadcast telegram) to the "Operational" communication status using the "Start Remote Node" NMT telegram. Only this communication status enables a data exchange via the process data objects.
- In C00378, you can set a delay time which must elapse after power-up before the controller applies the "Start Remote Node" NMT telegram to the bus.

Parameter	Information	Lenze setting	
		Value	Unit
C00352	CAN slave/master	Slave	
C00378	CAN delay boot-up - operational	3000	ms

Note!

Changing the master/slave operation in C00352 will only be effective

- by repeated mains switching of the controller

or

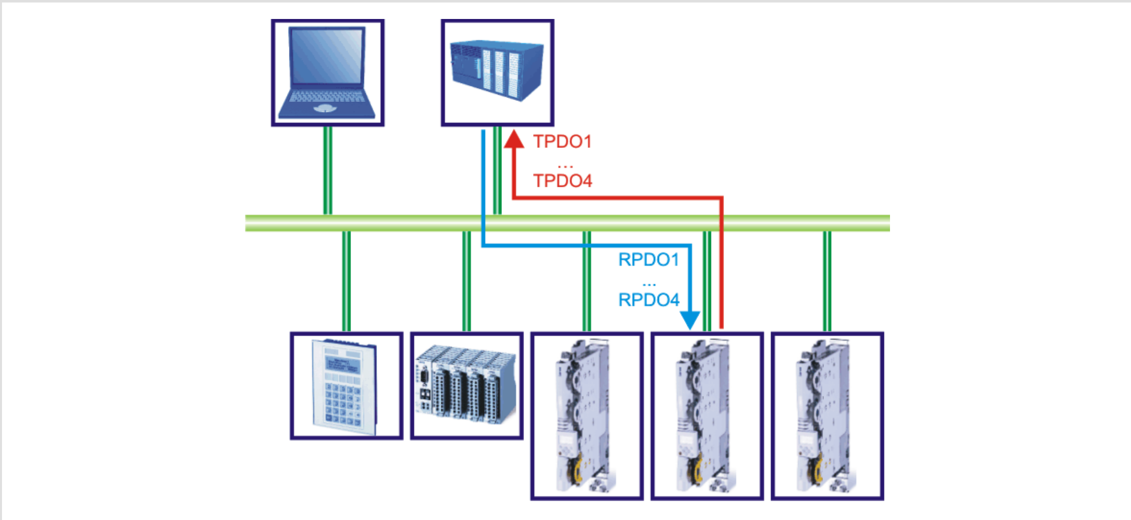
- by sending the NMT telegram "Reset Node" or "Reset Communication" to the controller.

As an alternative to the "Reset Node" NMT telegram, the device command C00002 = "91: CAN on board: Reset Node" can be used to reinitialise the CAN-specific device parameters.

Tip!

The master functionality is only required for the initialisation phase of the drive system.

9.6 Process data transfer



[9-6] PDO data transfer from / to the higher-level host system

For the transfer of process data, four separated process data channels (PDO1 ... PDO4) are available.

Definitions

- Process data telegrams between the host and the devices are distinguished 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 PDO (RPDOx): process data object received by a node
  - Transmit PDO (TPDOx): process data object transmitted by a node

Note!

Data can only be exchanged via the process data objects if the state is "Operational".

Communication phases/network management (302)

9.6.1 Identifiers of the process data objects

The identifiers for the process data objects PDO1 ... PDO4 in the Lenze setting result from the basic identifier and the node address set in C00350.

Identifier (COB-ID) = basic identifier + node address (node ID)

- The basic identifiers of the PDOs are preset in the Lenze setting according to the "Predefined Connection Set" of DS301 V4.02.
- The identifiers for the PDOs can be set individually via the Lenze codes and CANopen indexes listed in the following table. Thus, you can also set an identifier independent of the node address for certain PDOs.

Process data object	Basic identifier		Individual setting	
	dec	hex	Lenze code	CANopen index
PDO1				
RPDO1	512	0x200	C00321/1	I-1400/1
TPDO1	384	0x180	C00320/1	I-1800/1
PDO2				
RPDO2	768	0x300	C00321/2	I-1401/1
TPDO2	640	0x280	C00320/2	I-1801/1
PDO3				
RPDO3	1024	0x400	C00321/3	I-1402/1
TPDO3	896	0x380	C00320/3	I-1802/1
PDO4				
RPDO4	1280	0x500	C00321/4	I-1403/1
TPDO4	1152	0x480	C00320/4	I-1803/1

Note!

After the node address (C00350) has changed and a subsequent CAN reset node, the identifiers which result from the corresponding basic identifiers and the set node address are automatically set again in the subcodes of C00320 and C00321.

Tip!

The "Predefined Connection Set" can be re-established anytime using the following device commands (C00002):

- "93: CAN on board: Pred.Connect.Set" for CAN on board
- "94: CAN module: Pred.Connect.Set" for E94AYCCA communication module

9.6.2 Transmission type

The process data objects are transmitted in an event-controlled or time-controlled way.

- **Event-controlled:** The PDO is sent if a special device-internal event has occurred, for instance, when the data contents of the TPDO have changed or when a transmission cycle time has elapsed.
- **Synchronous:** A TPDO (or RPDO) is transmitted (or received) after the device has received a sync telegram (with identifier 0x80).
- **Cyclically:** The PDOs are transmitted in fixed time intervals after the transmission cycle time has elapsed.

The table shows that combinations of logic operations (AND, OR) are also possible between the different transmission modes:

Transmission type	PDO transmission			Logic operation
	cyclic	synchronous	event-controlled	
0		●	●	AND
1 ... 240		●		-
254, 255	●		●	OR

Transmission type	Description
0	The PDO is transmitted at every sync in an event-controlled manner (e. g. by means of a bit change within the PDO).
1 ... 240	**SYNC (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.
254, 255	**Event-controlled (with mask) with cyclic overlay** If this value is entered, the PDO transmission is event-controlled or cyclic. (Note: The values "254" and "255" have the same meaning). For cyclic overlay, a cycle time must be set for the respective PDO. In this case, cyclic transmission takes place in addition to event-controlled transmission (e.g. through a bit change in the PDO).

The communication parameters (as e.g. transmission mode and cycle time) can be freely adjusted for any PDO and independent of the settings of other PDOs:

Parameter	Information	Lenze setting	
		Value	Unit
C00322/1...4	CAN TPDOx Tx mode	254	
C00323/1...4	CAN RPDOx Rx mode	254	
C00324/1...4	CAN TPDOx delay time	0	1/10 ms
C00356/1...4	CAN TPDOx cycle time	0	ms

Tip!

The setting can also be made via the following CANopen projects:

- I-1400 ... I-1403: Communication parameters for RPDO1 ... RPDO4
- I-1800 ... I-1803: Communication parameters for TPDO1 ... TPDO4

9.6.3 Masking of the TPDOs for event control

For TPDO1 ... TPDO4, a mask can be parameterised for every byte. In case of the event-controlled transmission of a PDO, only the masked bits are used for the event control.

- Mask "0x0" means that no bit of the corresponding byte actuates the transmission.
- Mask "0xff" means that every bit of the corresponding byte can actuate the transmission.

Short overview: Parameters for masking the TPDOs

Parameter	Information	Lenze setting
C00311/1...8	CAN TPDO1 mask byte x	0x00
C00312/1...8	CAN TPDO2 mask byte x	0x00
C00313/1...8	CAN TPDO3 mask byte x	0x00
C00314/1...8	CAN TPDO4 mask byte x	0x00

9.6.4 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 monitoring the RPDOs

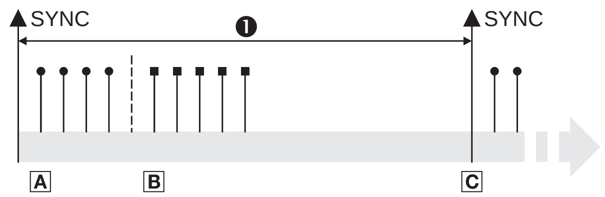
Parameter	Information	Lenze setting	
		Value	Unit
C00357/1...4	CAN RPDOx monitoring time	3000	ms
C00591/1...4	Resp. to CAN-RPDOx error	No response	

9.6.5 Synchronisation of PDOs via sync telegram

In case of cyclic transmission, one or several PDOs are transmitted/received at fixed time intervals. For synchronising the cyclic process data, an additional special telegram, the sync telegram, is used.

- The sync telegram is the trigger point for the transmission of process data from the frequency inverters to the master and for the acceptance of process data from the master by 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 transmission type selected.
  Transmission type (308)

General procedure



❶ Sync cycle time (C01121)

[9-7] Sync telegram

A. After the sync telegram has been received, the slaves send the synchronous process data to the master (TPDOs). The master reads them as process input data.
B. When the sending process has been completed, the slaves (RPDOs) receive the process output data (from the master).
   – All other telegrams (e.g. parameters or the event-controlled process data) are accepted acyclically by the slaves after the transmission has been completed successfully.
   – The acyclic data are not shown in illustration [9-7] They must be considered when the cycle time is dimensioned.
C. The data in the slave is accepted with the next sync telegram if the Rx mode is set to 1 ... 240. When the Rx mode is set to 254 or 255 the data is accepted in the next device cycle independent of the sync telegram.

9.6.5.1 Parameter setting

Short overview: Parameters for synchronisation via sync telegram

Parameter	Information	Lenze setting		Assignment	
		Value	Unit	Sync master	Sync slave
C00367	CAN SYNC Rx identifier	128			●
C00368	CAN SYNC Tx identifier	128		●	
C00369/1	CAN sync transmission cycle time	0	ms	●	
C01120	Sync source	Off			●
C01121	Sync cycle time	1000	µs		●
C01122	Sync phase position	400	µs		●
C01123	Sync tolerance	0	µs		●
C01124	Sync PLL increment	109	ns		●
C01130	Sync application cycle	1000	µs		●

Sync source

C01120 is used to select the source of the synchronisation signals. Only one source can synchronise the node.

Sync cycle time

Time with which the internal phase-locking loop (PLL) visualises the synchronisation signals. The time must be set in C01121 in accordance with the cycle of the synchronisation source selected in C01120.

Note!

If the synchronisation takes place via the system bus, it is reasonable to only enter integer multiples of 1000 µs in C01121.

Example: For the system bus, the distance between two synchronisation signals is set to 2 ms. If the system bus is to be used as synchronisation source, a sync cycle time of 2000 µs must be set in C01121.

Sync phase position

The phase position defines the zero point of time for the application relating to the synchronisation signal (bus cycle). Since PDO processing is integrated in the system part of the application, the instant of the PDO acceptance also changes if the phase position is changed.

- If 0 is set, the application is started together with the synchronisation signal.
- If a value > 0 is set, the application starts by the set time interval before the synchronisation signal arrives (the phase position acts negatively).

Example: If the phase position is set to 400 μs, the system part of the application starts 400 μs before the synchronisation signal arrives.

> **Note!**
>
> From software version V3.0:
>
> The effect of the sync phase position can be affected by the application cycle set in C01130. For the Lenze setting of C01130 the behaviour remains as before.

Sync tolerance

Time slot for monitoring the synchronisation signal via the system block **LS_SyncInput**.
System block "LS_SyncInput" (365)

- If the last synchronisation signal amounted to approx. the expected value within this time slot, the *SYNC_bSyncInsideWindow* output of the **LS_SyncInput** system block is set to TRUE.
- This setting does not affect the synchronisation process.

Sync PLL increment

If the cycle times of the synchronisation signal and the phase-locking loop (PLL) differ from each other, the setting in C01124 defines the increment with which the phase-locking loop can be reset.

- The recommended reset time for the system bus as synchronisation source with regard to occurring deviations is 109 ns (Lenze setting).

Sync application cycle

This parameter influences the effect of the sync phase position (C01122) with regard to the instant of acceptance of the synchronous PDOs by the application or the instant of transmission of the synchronous PDOs to the system bus.

The following applies to software versions lower than V3.0:

- The sync application cycle is permanently set to 1000 µs.
- The resulting PDO delay can be calculated with the following formula taking into consideration an internal processing time of 150 s: PDO delay= (sync cycle time - sync phase position + 150 µs) modulo 1000

The following applies from software version V3.0:

- The sync application cycle can be set in C01130. The set value is automatically rounded down to full 1000 µs.
- The resulting PDO delay can be calculated with the following formula taking into consideration an internal processing time of 150 s: PDO delay= (sync cycle time - sync phase position + 150 µs) modulo C01130

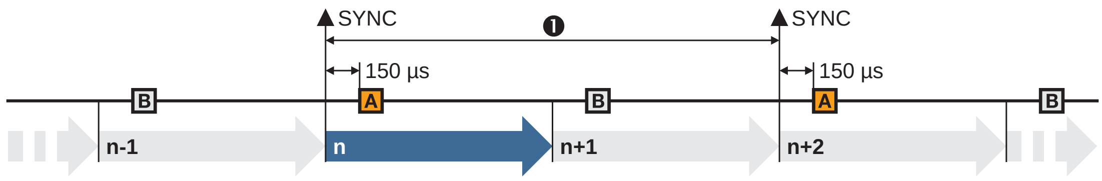
Note!

If the sync application cycle in C01130 is set higher than the sync cycle time (C01121), the behaviour is undefined. The same applies if the sync phase position (C01122) is set higher than the sync cycle time (C01121).

Usually, no synchronous PDOs are then applied to the system bus anymore.

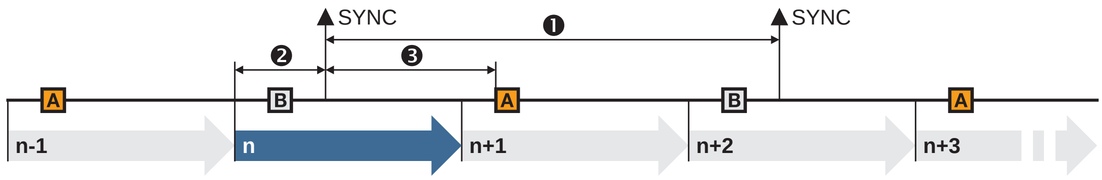
9.6.5.2 Effect of C01130 on the sync phase position

Example 1:



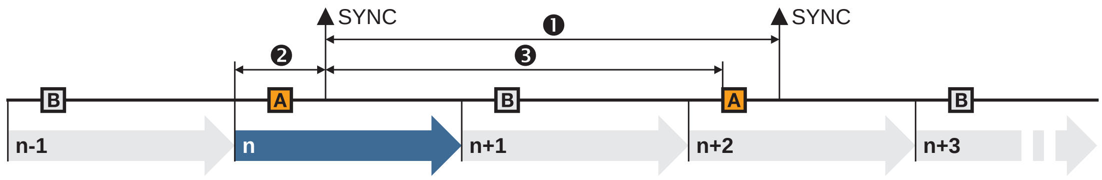
❶	Sync cycle time (C01121)	= 2000 µs
	Sync phase position (C01122)	= 0 µs
	Sync application cycle (C01130)	= 1000 µs
A	Instant of acceptance/transmission of synchronous PDOs	
B	Inactive instant of acceptance	
n	Device cycle	

Example 2:



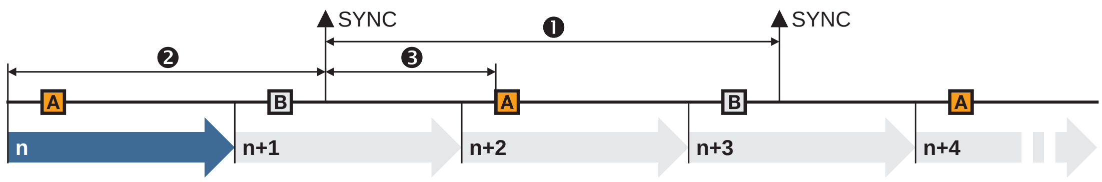
❶	Sync cycle time (C01121)	= 2000 µs
❷	Sync phase position (C01122)	= 400 µs
	Sync application cycle (C01130)	= 1000 µs
❸	PDO delay	= (C01121 - C01122 + 150 µs) modulo C01130 = 750 µs
A	Instant of acceptance/transmission of synchronous PDOs	
B	Inactive instant of acceptance	
n	Device cycle	

Example 3:



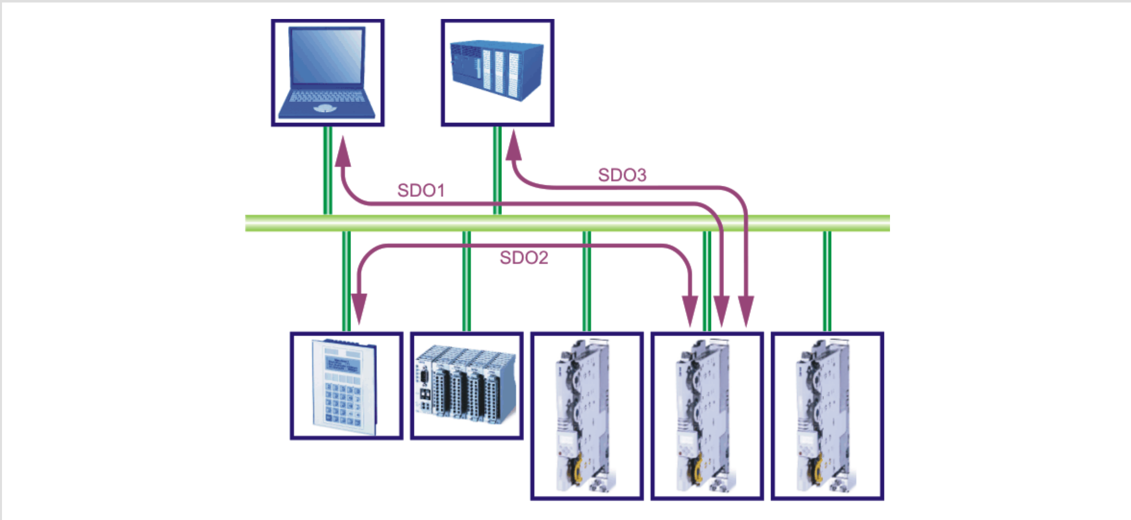
❶	Sync cycle time (C01121)	= 2000 µs
❷	Sync phase position (C01122)	= 400 µs
	Sync application cycle (C01130)	= 2000 µs
❸	PDO delay	= (C01121 - C01122 + 150 µs) modulo C01130 = 1750 µs
A	Instant of acceptance/transmission of synchronous PDOs	
B	Inactive instant of acceptance	
n	Device cycle	

Example 4:



❶	Sync cycle time (C01121)	= 2000 µs
❷	Sync phase position (C01122)	= 1400 µs
	Sync application cycle (C01130)	= 1000 µs
❸	PDO delay	= (C01121 - C01122 + 150 µs) modulo C01130 = 750 µs
A	Instant of acceptance/transmission of synchronous PDOs	
B	Inactive instant of acceptance	
n	Device cycle	

9.7 Parameter data transfer



[9-8] Parameter data transfer via available parameter data channels

Parameters are values stored in codes on Lenze controllers.

Ten separate parameter data channels are available for parameter setting, enabling the simultaneous connection of several devices for configuration.

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 the object directory. Indexes (e.g. I-1000) ensure access to 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.

9.7.1 Identifiers of the parameter data objects

The identifiers for the parameter data objects SDO1 ... SDO10 in the Lenze setting result from the basic identifier and the node address set in C00350.

Identifier (COB-ID) = basic identifier + node address (node ID)

- The basic identifiers of the SDOs are preset in the Lenze setting according to the "Predefined Connection Set" of DS301 V4.02.

Parameter data object	Direction		Basic identifier	
	from the device	to the device	dec	hex
SDO1 (Parameter data channel 1)	●		1408	580
		●	1536	600
SDO2 ... 10 (Parameter data channel 2 ... 10)	●	●	Deactivated	
Node guarding, heartbeat	●		1792	700
Boot up	●		1792	700

Note!

Please observe that the parameter data channels 2 ... 10 are deactivated in the Lenze setting.

The procedure for activating these parameter data channels is explained in the description of parameters C00372 and C00373 and the description for the implemented CAN object I-1201. ▶ Example (📖 353)

Lenze

9.7.2 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!

User data are displayed in the Motorola format.

▶ Examples for a parameter data telegram (📖 323)

The following subchapters describe the user data in detail.

9.7.2.1 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	Information
	hex	dec		
Write request	0x23	35	4 byte	Writing a parameter to the controller.
	0x2B	43	2 bytes	
	0x2F	47	1 byte	
	0x21	33	Block	
Write response	0x60	96	4 byte	Acknowledgement by the controller regarding a write request.
Read request	0x40	64	4 byte	Reading a parameter from the controller.
Read response	0x43	67	4 byte	Response from the controller 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 byte	Response from the controller when the read/write request could not be executed correctly. Error messages (321)

In detail, the command byte contains 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 bytes
e: expedited (shortened block service)
s: segmented (normal block service)

Tip!

Further commands are defined in the CANopen specification DS301, V4.02 (e. g. segmented transfer).

9.7.2.2 Addressing through 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 (Lenze code) is addressed is addressed according to the following formula:
Index = 24575 - (Lenze code number)

Example

The parameter C00011 (motor reference speed) is to be addressed.

Calculation:

- Index:
  - Decimal: 24575 - 11 = 24564
  - Hexadecimal: 0x5FFF - 0xB = 0x5FF4
- Subindex: 0x00 (subindex 0, since the parameter has no 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**				

9.7.2.3 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

A maximum of 4 bytes is available for parameter value entries. The bytes are assigned as follows, depending on the data format:

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 table of attributes contains a scaling factor for all Lenze parameters in the "factor" column. The scaling factor is important for the transfer of parameter values which are shown with one or several decimal positions in the parameter list.

When the scaling factor is > 1, the value must be multiplied by the scaling factor before being transmitted to be able to transmit the value as an integer. On the SDO client side, the integer must by divided by the scaling factor again to obtain the original value with decimal positions.

Example

For a code with the scaling factor "100" and the data format U32 the value "123.45" is to be transmitted.

Calculation:

- Value to be transmitted = scaling factor x value
- Data $_{(1 \ldots 4)}$ = 100 x 123.45 = 12345 (0x00 00 30 39)

Entries:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
				0x39	0x30	0x00	0x00

9.7.2.4 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 case of an error, the addressed node generates a telegram with the "Error Response" (0x80) command.

- The telegram contains the index and subindex of the code in which an error has occurred.
- Bytes 5 ... 8 contain the error code.
  - The error codes are standardised acc. to DS301, V4.02.
  - The representation of the error codes is reversed to the read direction (see the following example).

Example

Representation of the error codes "0x06 04 00 41" in the 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

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 (block mode only).
0x0504 0003	Invalid processing number (block mode only).
0x0504 0004	CRC error (block mode only).
0x0504 0005	Memory does not suffice.
0x0601 0000	Object access not supported.
0x0601 0001	Attempted read access to a writable only object.
0x0601 0002	Attempted write access to a readable 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.
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 too short).
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 or saved for application.
0x0800 0021	Data cannot be transferred or saved for application due to local control.
0x0800 0022	Data cannot be transferred or 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).

9.7.3 Examples for a parameter data telegram

9.7.3.1 Reading parameters

Task: The heatsink temperature of 43 °C (code C00061, data format INTEGER32, scaling factor 1) is to be read from the controller with node address 5.

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 of the telegram to the drive	
Identifier	= 1536 + node address = 1536 + 5 = 1541 = 0x0605 (1536 = basic identifier SDO1 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 has no subcodes)

Response telegram from drive (if data has been transmitted correctly)

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
0x0585	0x43	0xC2	0x5F	0x00	0x2B	0x00	0x00	0x00

Explanations of the telegram from the drive	
Identifier	= 1408 + node address = 1408 + 5 = 1413 = 0x0585 (1408 = basic identifier SDO1 from the controller)
Command	= 0x43 = "Read Response" (response to read request with current value)
Index	as in telegram to the drive
Subindex	
Data 1 ... 4	= 0x0000002B = 43 [°C]

9.7.3.2 Writing parameters

Task: The rated current of the connected motor with I_{rated} = 10.2 A (code C00088) is to be entered in the controller.

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	0x66	0x00	0x03	0x00

Explanations of the telegram to the drive	
Identifier	= 1536 + node address = 1536 + 2 = 1538 = 0x0602 (1536 = basic identifier SDO1 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 has no subcodes)
Data 1 ... 4	= 10,2 x 10 = 102 = 0x00030066 (Value for motor current, data type U32; display factor 1/10)

Response telegram from drive (if data has been transmitted correctly)

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 of the telegram from the drive	
Identifier	= 1408 + node address = 1408 + 2 = 1410 = 0x0582 (1408 = basic identifier SDO1 from the controller)
Command	= 0x60 = "Write Response" (acknowledgement of the write access from the controller)
Index	as in telegram to the drive
Subindex	

9.7.3.3 Reading 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 of the telegram to the drive	
Identifier	= 1536 + node address = 1536 + 12 = 1548 = 0x060C (1536 = basic identifier SDO1 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 has no subcodes)

Response telegram 1 from the drive: Data 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 of the telegram from the drive	
Identifier	= 1408 + node address = 1408 + 12 = 1420 = 0x058C (1408 = basic identifier SDO1 from the controller)
Command	= 0x41 = "Read Response" (response is a block telegram)
Index	as in telegram to the drive
Subindex	
Data 1 ... 4	= 0x0000000B = data length of 11 characters in ASCI 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 of the telegram to the drive	
Command	= 0x60 = "Read Segment Request" (request: read data block) • Bit 4 = 0 (toggle bit)
	Influence of the toggle bit to the request command The individual blocks are toggled successively, i.e. first the request with command "0x60" is effected (= $0110\mathbf{0}{*}0000_{bin}$), then command "0x70" (= $0111\mathbf{1}{*}0000_{bin}$), then "0x60" again, etc. * Toggle bit

Response telegram 2 from the drive: Send 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 of the telegram to the drive	
Command	= 0x00 = 00000000_{bin} • Bit 4 = 0 (toggle bit)
	Influence of the toggle bit to the transmission command • The first response of the controller in the command byte is "$0x000\mathbf{0}{*}0000_{bin}$" if bytes 2 ... 8 are completely assigned with data and further telegrams are following. • The second response of the controller in the command byte is "$0x000\mathbf{1}{*}0000_{bin}$" if bytes 2 ... 8 are completely assigned with data and further 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 of the 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: Send 2nd data block with 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 of the 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 end bit and the residual data length on the transmission command • The end of transmission is reported by the set end bit 0. • By means of bit 1 ... 3, the number of bytes which do not contain data anymore is announced. * Toggle bit
Data 1 ... 7	= "0.00" (ASCII representation) The result of the data block transfer is: "01.00.00.00"

9.8 Diagnostics

The display parameters listed in the following table serve to request current information on the system bus for diagnostic purposes, e.g. using the keypad, via a bus system, or using the »Engineer« (when an online connection has been established to the controller).

- The »Engineer« parameter list and the keypad contain these parameters in the category **CAN → CAN management**.
- A detailed description of these parameters can be found in the chapter "Parameter reference". (📖 705)

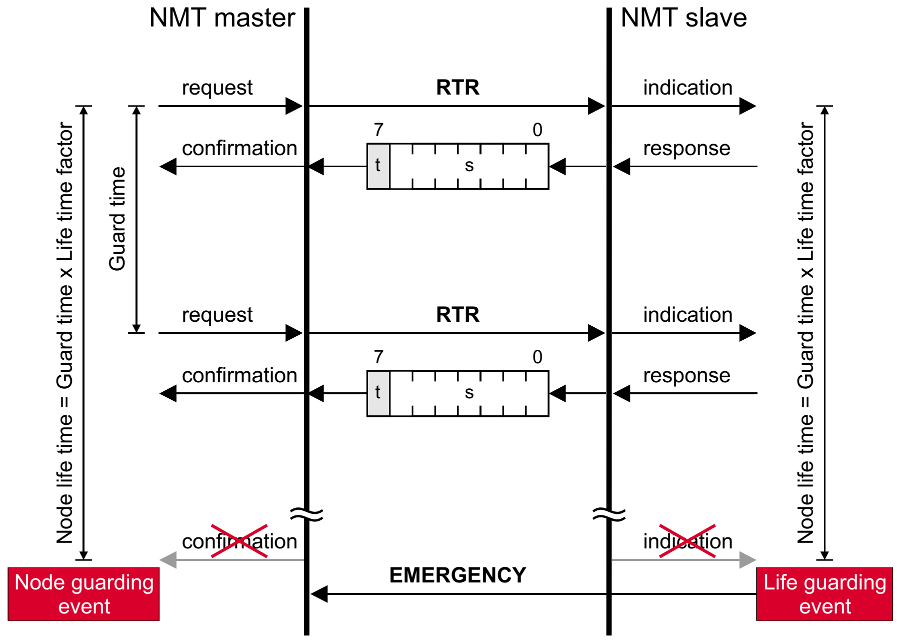
Parameter	Display
C00345	CAN error
C00359	CAN status
C00360/1	CAN stuffing bit error counter
C00360/2	CAN format error counter
C00360/3	CAN acknow. error counter
C00360/4	CAN bit 1 error counter
C00360/5	CAN bit 0 error counter
C00360/6	CAN CRC error counter
C00360/7	CAN Tx telegram counter
C00360/8	CAN Rx telegram counter
C00361/1	CAN bus load: Current node load in Tx direction
C00361/2	CAN bus load: Current node load in Rx direction
C00361/3	CAN bus load: Current node load of faulty telegrams
C00361/4	CAN bus load: Node peak load in Tx direction
C00361/5	CAN bus load: Node peak load in Rx direction
C00361/6	CAN bus load: Node peak load of faulty telegrams
C00390	CAN Error Register (DS301V402)

9.9 Monitoring

9.9.1 Node guarding protocol

In a CAN network, the node guarding protocol serves to monitor the connection between the NMT master and the NMT slave(s). If the controller was parameterised as NMT master, it can monitor up to 32 NMT slaves.

General procedure



t: Toggle bit
s: NMT slave status (4: Stopped, 5: Operational, 127: Pre-Operational)

[9-9] Node guarding protocol

1. The NMT master within cyclic time intervals sends a data telegram to the NMT slave, which is referred to as "Remote Transmission Request" (RTR).
2. The NMT slave then returns a response telegram ("Response") to the NMT master.

9.9.1.1 Telegram structure

RTR telegram

- The RTR telegram from the NMT master has the following identifiers:
  Identifier (COB-ID) = 1792 + node address of the NMT slave
- The RTR telegram does not contain any user data.
- The RTR bit in the arbitration field of the RTR telegram is set to the valency LOW (dominant level).

Response telegram

- The response telegram from the requested NMT slave has the same identifier as the RTR telegram received by the NMT master.
- The user data (1 byte) contains the NMT slave status and the toggle bit (see the following description).

NMT slave state (s)

NMT slave status		Data							
Communication status	Decimal value (s)	(t)	NMT slave state (s)						
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Stopped	4	0/1	0	0	0	0	1	0	0
Operational	5	0/1	0	0	0	0	1	0	1
Pre-operational	127	0/1	1	1	1	1	1	1	1

Toggle bit (t)

- The toggle bit (t) in the response telegram has the value "0" when the node guarding protocol is activated for the first time.
- The toggle bit (t) must change its value with each response.

Note!

The toggle bit is monitored by the NMT master.

If a telegram is received with a toggle bit value that has not changed compared to the previously received telegram, it will be treated as if it were not received, i.e. the monitoring time is not reset and elapses further.

The toggle bit can only be reset to the value "0" by the "Reset Communication" telegram of the NMT master.

9.9.1.2 Parameter setting

Short overview of parameters for the "Node Guarding" monitoring function:

Parameter	Information	Lenze setting		Assignment	
		Value	Unit	Master	Slave
C00382	CAN guard time	0	ms		●
C00383	CAN Life Time Factor	0			●
C00386/1...32	CAN node guarding	0x00000000		●	
C00387	CAN Node Guarding Activity	-		●	
C00388/1...32	CAN node guarding status	-		●	
C00612/1...32	Resp. to CAN node guarding error	No response		●	
C00614	Resp. to CAN life guarding error	No response			●
C00625	CAN behaviour in case of fault	Pre-operational state		●	●
Highlighted in grey = display parameter					

Guard time

The time interval with which the NMT master transmits the RTR telegram is the "Guard Time".

- For each NMT slave to be monitored an individual time interval can be set.
- The RTR telegram prompts the NMT slave to send its current status.

Node life time

The "Node Life Time" is the product of the "Guard Time" and the "Life Time Factor":
Node Life Time = Guard Time x Life Time Factor

- "Life time factor" and "Guard time" have to be known to the NMT master. For this, the values from the NMT slave are read at each reboot, or defined values are sent to the NMT slave at each reboot.
- It is possible to select a different "node life time" for each NMT slave to be monitored.

OK status

The status of the connection is ok (OK status) if within the "Node life time"

- the NMT slave has received an RTR telegram from the NMT master and
- the NMT master has received a correct response from the requested NMT slave.

In the OK status the monitoring times for the NMT master and the NMT slave are reset and the node guarding protocol is continued.

Life guarding event

The "Life Guarding Event" is triggered in the NMT slave if this has not received an RTR telegram from the NMT master within the "Node Life Time":

- In the Lenze setting, the NMT slave changes from the "Operational" communication status into the "Pre-Operational" communication status.
  - C00625 or the I-1029 object serve to set a status change.
- An emergency telegram with the emergency error code 0x8130 is transmitted to the NMT master.
- The response parameterised in C00614 takes place (Lenze setting: "No response").

> **Note!**
>
> The "Life Guarding Event" can only be triggered in the NMT slave if at least one RTR telegram has been received successfully from the NMT master.

Node guarding event

The "Node Guarding Event" is triggered in the NMT master if this has not received any response to its RTR telegram from the requested NMT slave within the "Node Life Time" or the toggle bit in the response telegram has not changed within the "Node Life Time".

- In the Lenze setting, the NMT master changes from the "Operational" communication status into the "Pre-Operational" communication status.
  - C00625 or the I-1029 object serve to set a status change.
- The response parameterised in C00612/1...32 takes place (Lenze setting: "No response"). The response in the NMT master can be set individually for each monitored node.

> **Note!**
>
> The "Node Guarding Event" can only be triggered in the NMT master if at least one response has been received successfully from the requested NMT slave.

9.9.1.3 Commissioning example

Task

A 9400 controller configured as NMT master (node 1) is to monitor another 9400 controller (node 2).

- The node guarding telegram is to be transmitted from the NMT master to the NMT slave in intervals of 1 s:
  - Guard time = 1000 ms
- The node life time is to amount to 5 seconds:
  - Node life time = guard time (1000 ms) x life time factor (5)
- If an error occurs, an error response is to be activated both in the NMT master and the NMT slave.

Parameter setting of the NMT master (node 1)

1. Set heartbeat producer time (C00381) to 0 ms to deactivate the heartbeat monitoring (node guarding and heartbeat must not be used simultaneously in a CANopen device).
2. Configure controller as NMT master: Set C00352 = "1: Master".
3. Set guard time (C00382) to 0 ms (slave parameter).
4. Set life time factor (C00383) to 0 (slave parameter).
5. Configure monitoring for the node guarding in C00386.
   - The value to be entered into a free subcode (1 ... 32) is "0x050203E8". It consists of the following:

Bit 31 ... bit 24 **Life time factor**	Bit 23 ... bit 16 **Node address of slave**	Bit 15 ... bit 0 **Guard time**
0x05	**0x02**	**1000 [ms] = 0x03E8**

6. Go to C00612/1...32 and set the response required for the monitoring functions parameterised in C00386/1...32 which are to take place in case of a "Node Guarding Event" in the NMT master.

Tip!

- C00387 displays the activity of every monitoring function parameterised in C00386/1...32 in a bit-coded form.
- C00388/1...32 displays the node guarding status of the monitored NMT slaves.
- C00625 serves to set which status change is to occur in the NMT master in case of a "Node Guarding Event".

Parameterise NMT slave (node 2)

1. Accept the settings made in the NMT master in C00386 of the life time factor and the guard time for the NMTslave:
   - Set guard time (C00382) to 1000 ms.
   - Set life time factor (C00383) to 5.
2. Go to C00614 and set the response required in case of a "Life Guarding Event" in the NMT slave.

Tip!

C00625 serves to set which status change is to occur in case of a "Life Guarding Event" in the NMT slave.

Node guarding telegrams

- Remote Transmission Request:
  The RTR telegram from the NMT master has the following identifiers:
  Identifier (COB-ID) = 1792 + node address of slave = 1792 + 2 = 1794 = 0x702
- Remote Transmission Response:
  The response telegram from the NMT slave has the same identifier and the "Operational" NMT status in the user data (s = 5). Telegram structure (330)

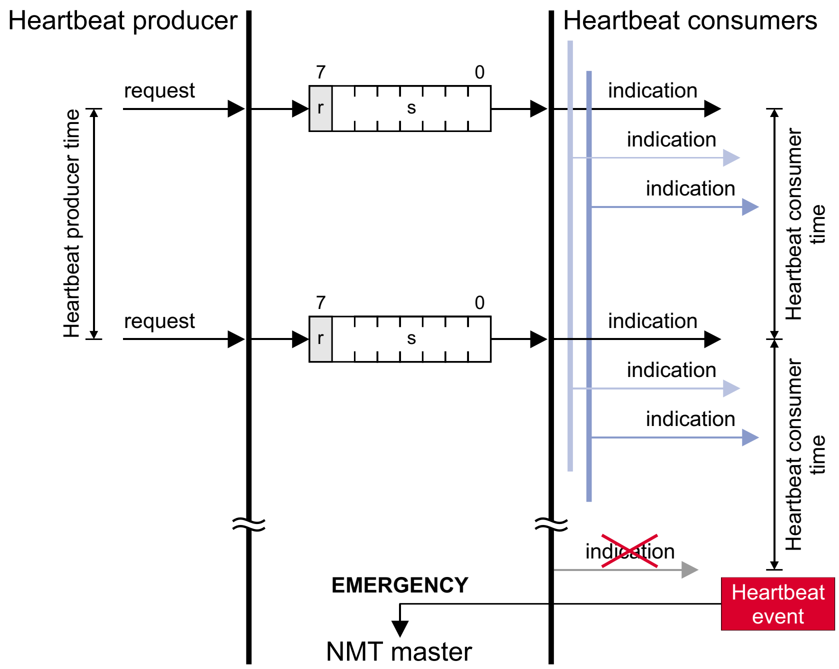
9.9.2 Heartbeat protocol

The heartbeat protocol can be used optionally to the node guarding protocol for monitoring nodes within a CAN network. Unlike the node guarding, this monitoring does not require a polling by means of an RTR telegram (Remote Transmission Request) from the NMT master.

Note!

Heartbeat and node guarding protocols must not be used simultaneously in a CANopen device. If the heartbeat producer time is set > 0 ms, the heartbeat protocol is used.

General procedure



r: reserved (always 0)
s: Status of the producer (0: Boot-up, 4: Stopped, 5: Operational, 127: Pre-Operational)

[9-10] Heartbeat protocol

1. A heartbeat producer cyclically sends a heartbeat telegram to one or several heartbeat consumers.
2. One or several consumers monitor the regular arrival of the heartbeat telegram.

9.9.2.1 Telegram structure

- The heartbeat telegram from the producer has the following identifier:
  Identifier (COB-ID) = 1792 + node address of the producer
- 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

9.9.2.2 Parameter setting

Short overview of the parameters for the "Heartbeat" monitoring function:

Parameter	Information	Lenze setting		Assignment	
		Value	Unit	Consumer	Producer
C00346	CAN heartbeat activity	-		●	
C00347/1...32	CAN heartbeat status	-		●	
C00381	CAN heartbeat producer time	0	ms		●
C00385/1...32	CAN heartbeat consumer time	0x00000000		●	
C00613/1...32	Resp. to CAN heartbeat error	No response		●	
C00625	CAN behaviour in case of fault	Pre-operational state		●	●
Highlighted in grey = display parameter					

Heartbeat producer time

Time interval for the transmission of the heartbeat telegram to one or several consumers.

- Can be parameterised in C00381 or via the object I-1017. The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is automatically transmitted as soon as a time > 0 ms is set.

Note!

Heartbeat and node guarding protocols must not be used simultaneously in a CANopen device. If the heartbeat producer time is set > 0 ms, the heartbeat protocol is used.

Heartbeat consumer time

Monitoring time for the node (producer) to be monitored.

- Can be parameterised in C00385/1...32 or via the object I-1016.
- The parameterised time is rounded down to an integer multiple of 5 ms and must have a higher value than the heartbeat producer time of the node to be monitored.
- A consumer can monitor up to 32 producers.

Heartbeat event

The "Heartbeat Event" is triggered in the consumer if this has not received a heartbeat telegram from the producer within the "Heartbeat Consumer Time":

- In the Lenze setting, the consumer changes from the "Operational" communication status into the "Pre-Operational" communication status.
  - C00625 or the I-1029 object serve to set a status change.
- An emergency telegram with the emergency error code 0x8130 is transmitted to the NMT master.
- The response parameterised in C00613/1...32 for the corresponding producer takes place (Lenze setting: "No response").

Note!

The heartbeat monitoring starts when the first heartbeat telegram of a monitored producer has been successfully received and the "pre-operational" NMT status has been assumed.

The boot-up telegram counts as the first heartbeat telegram!

9.9.2.3 Commissioning example

Task

A 9400 controller (node 2) configured as a heartbeat consumer is to monitor another 9400 controller (Heartbeat Producer; node 1).

- The heartbeat producer is to transmit every 10 seconds a heartbeat telegram to the heartbeat consumer.
- The heartbeat consumer monitors the arrival of the heartbeat telegram. In case of an error, a response is to take place.

Parameter setting of the heartbeat producer (node 1)

1. Set heartbeat producer time (C00381) to 10 ms.

Parameter setting of the heartbeat consumer (node 2)

1. Configure monitoring for the heartbeat in C00385.
   - Note: The heartbeat consumer time must be higher than the heartbeat producer time set in C00381 for the node to be monitored.
   - The value to be entered into a free subcode (1 ... 32) is "0x0001000F". It consists of the following:

Bit 31 ... bit 24 **Reserved**	Bit 23 ... bit 16 **Node address of the producer**	Bit 15 ... bit 0 **Heartbeat consumer time** (integer multiple of 5 ms)
0x00	**0x01**	**15 [ms] = 0x000F**

2. Go to C00613/1...32 and set the response required for the monitoring functions parameterised in C00385/1...32 which are to take place in case of a "Heartbeat Event" in the consumer.

Tip!

- C00346 displays the activity of every monitoring function parameterised in C00385/1...32 in a bit-coded form.
- C00347/1...32 displays the node guarding status of the monitored NMT slaves.
- C00625 serves to set which status change is to occur in case of a "Heartbeat Event".

Heartbeat telegram

- The heartbeat telegram from the producer has the following identifier:
  Identifier (COB-ID) = 1792 + node address of the producer = 1792 + 1 = 1793 = 0x701

9.9.3 Emergency telegram

If the error status changes due to the occurrence or elimination of an internal device error, an emergency telegram with the following structure is sent once to the NMT master:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Emergency error code		**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			• With emergency error code 0x1000: Lenze error number (displayed value in C00168) • With all other emergency error codes, the value "0" is entered here.				

Emergency error code	Error register	Cause
0x0000	0xxx	One of several errors eliminated
	0x00	Single error eliminated (afterwards no more errors)
0x1000	0x01	Generic error • In the standard device, an error has occurred with the error response "Fault", "Trouble", "Quick stop by trouble", "Warning", "Warning locked" or "System fault". • Error message is the Lenze error number (C00168). • For error cause see fault error description (C00166).
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 is shorter than the expected PDO length
0x8220	0x11	PDO length is longer than the expected PDO length
0x8700	0x11	Monitoring of the sync telegram

Example

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Emergency error code		**Error register**	**Manufacturer-specific error message**				
0x00	0x10	0x01	0x00 (reserved)	0x1b	0x00	0x7b	0x00
Generic error				Lenze error message 0x007b001b: Encoder wire breakage. ▶ Error messages of the operating system Corresponding "error-free" message: value "0x00000000"			

Tip!

A detailed description can be gathered from the CAN specification DS301, V4.02.

9.10 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. The following table lists the related Lenze codes in the column "Relationship to Lenze codes".

Note!

Some of the terms used here derive from the CANopen protocol.

Overview of the CANopen indexes 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	C00390
I-1003	Pre-defined 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-100C	0	Guard time	C00382
I-100D	0	Life time factor	C00383
I-1010	Store parameters		
	0	Highest subindex supported	-
	1	Save all parameters	-
I-1011	Restore default parameters		
	0	Highest subindex supported	-
	1	restore all default parameters	-
I-1014	0	COB-ID EMCY	C00391
I-1015	0	Inhibit time EMCY	C00392
I-1016	Consumer heartbeat time		
	0	Highest subindex supported	-
	1 ... 32	Consumer heartbeat time	C00385/1...32
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	-

CANopen object			Relationship to Lenze code
Index	Subindex	Name	
I-1029	Error behaviour		
	0	Highest subindex supported	-
	1	Communication error	C00625
I-1200	SDO1 server parameter		
	0	Highest subindex supported	-
	1	COB-ID client –> server (rx)	C00372/1
	2	COB-ID server –> client (tx)	C00373/1
I-1201 ... I-1209	SDO2 ... SDO10 server parameter		
	0	Highest subindex supported	-
	1	COB-ID client –> server (rx)	C00372/2...10
	2	COB-ID server –> client (tx)	C00373/2...10
	3	Node-ID of the SDO client	-
I-1400 ... I-1403	RPDO1 ... RPDO4 communication parameter		
	0	Highest subindex supported	-
	1	COB-ID used by RPDO	C00321/1...4
	2	Transmission type	C00323/1...4
	3	Inhibit time	-
	4	Compatibility entry	-
	5	Event timer	-
I-1600 ... I-1603	RPDO1 ... RPDO4 mapping parameter		
	0	Number of mapped application objects in PDO	-
	1 ... 8	Application object 1 ... 8	-
I-1800 ... I-1803	TPDO1 ... TDDO4 communication parameter		
	0	Highest subindex supported	-
	1	COB-ID used by TPDO	C00320/1...4
	2	Transmission type	C00322/1...4
	3	Inhibit time	C00324/1...4
	4	Reserved	-
	5	Event timer	C00356/1...4
I-1A00 ... I-1A03	TPDO1 ... TDDO4 mapping parameter		
	0	Number of mapped application objects in PDO	-
	1 ... 8	Application object 1 ... 8	-

I-1000

Index **I-1000**	Name: **Device type**					
Subindex	Default setting	Display area (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**	

[9-1] Data telegram assignment

I-1001

Index: **I-1001**	Name: **Error register**					
Subindex	Default setting	Display area (min. value | unit | max. value)			Access	Data type
0: Error register	-	0		255	ro	U8

Error register

- This object is related to the Lenze code C00390.
- The error status in the data byte (U8) is bit coded (see the following table). Currently only bit 0 and bit 4 in the data byte contain the corresponding information.

Bit	Meaning if bit is set:
Bit 0	Generic error
Bit 1	Current error (not used)
Bit 2	Voltage error (not used)
Bit 3	Temperature error (not used)
Bit 4	Communication error
Bit 5	Device profile spec. error (not used)
Bit 6	Reserved
Bit 7	Manufacturer-specific error (not used)

I-1003

Index: **I-1003**	Name: **Pre-defined 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 the error messages stored
1 ... 10	Display of the error list The error messages (U32) consist of a 16-bit error code and a 16-bit manufacturer-specific information field.

Note!

The values in the "standard error field" under subindexes 1 ... 10 will be deleted if the subindex "number of recorded errors" is overwritten with the value "0".

Emergency error code	Cause	Entry in the error register (I-1001)
0x0000	One of several errors eliminated	0xxx
	Elimination of an individual error (afterwards error-free state)	0x00
0x1000	Standard device is in error status (error response "Trouble", "Message", "Warning", "Fault", "Quick stop by trouble" or "System fault")	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 also is used for transmission.	0x11
0x8210	PDO length is shorter than the expected PDO length	0x11
0x8220	PDO length is longer than the expected PDO length	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 serves to activate the creation of sync telegrams and write the value of the identifier.

- This object is related to the Lenze codes C00367 and C00368.

Creating sync telegrams

To create sync telegrams, the bit 30 (see below) must be set to "1". The interval of the sync telegrams can be set with 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 the CANopen specification) . This means that all modules are by default set to the same sync telegram.

- If sync telegrams are only to be received from specific communication modules, their identifiers can be entered with a value of up to and including 0x07FF.
- The identifier may only be changed if the communication module does not transmit a sync telegram (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".

[9-2] 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.

- A cycle time can be defined with the entry "1000" or an integer multiple of it.
- With "0 µs" (Lenze setting), no sync telegrams are created.
- This object is related to the Lenze code C00369.

I-100C

Index: **I-100C**	Name: **Guard time**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: Guard time	0 ms	0	ms	65535	rw	U16

Monitoring time for node guarding. Node guarding protocol (329)

- Time within the NMT slave expects the RTRs from the NMTmaster.
- The "Node Life Time" is the product from "Guard Time" and "Life Time Factor":
  Node Life Time = Guard Time (I-100C) **x Life Time Factor** (I-100D)
- The "Life Guarding Event" occurs in the NMT slave if this has not been triggered from the NMT master trough an RTR within the "Node Life Time".
- With "0 ms" (Lenze setting) monitoring is not supported by the slave.
- This object is related to the Lenze code C00382.

I-100D

Index: **I-100D**	Name: **Life time factor**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: Life time factor	0	0		255	rw	U8

Life Time Factor for node guarding. Node guarding protocol (329)

- The "Node Life Time" is the product from "Guard Time" and "Life Time Factor":
  Node Life Time = Guard Time (I-100C) **x Life Time Factor** (I-100D)
- The "Life Guarding Event" occurs in the NMT slave if this has not been triggered from the NMT master trough an RTR within the "Node Life Time".
- With "0" (Lenze setting) monitoring is not supported by the slave.
- This object is related to the Lenze code C00383.

I-1010

Index: **I-1010**	Name: **Store parameters**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: highest subindex supported	1	- (only read access)			ro	U32
1: Save all parameters	-	0		4294967295	rw	U32

Save parameters with mains failure protection.

- Corresponds to the device command C00002 = "11: Save start parameters".
- This command serves to save the current parameter settings of the active application with mains failure protection in the memory module of the controller.

Subindex	Meaning	
	Read	Write
0	Max. supported subindex: 1	- (a write attempt triggers the error message 0x06010002.)
1	Reading memory functions of all parameters.	Save parameters with mains failure protection.

Read subindex 1

8th byte	7th byte	6th byte	5th byte	
Data 4	Data 3	Data 2	Data 1	
Bit 31 ... bit 2			Bit 1	Bit 0
0			**0**/1	0/**1**

[9-3] Assignment of the data telegram (read access)

Bit	Meaning	
Bit 0	0	No saving of parameters on command.
	1	Saving of parameters on command (Lenze).
Bit 1	0	No automatic saving of parameters (Lenze).
	1	Automatic saving of parameters.

Write subindex 1

In addition to the index and subindex, the telegram data must also include the "save" signature (ASCII characters; ISO 8859) so that the parameters are stored.

- A response according to the DS301 V4.02 specification occurs while writing with a wrong identifier.

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
"e" = 0x65	"v" = 0x76	"a" = 0x61	"s" = 0x73

[9-4] Assignment of the data telegram (write access)

I-1011

Index: **I-1011**	Name: **Restore default parameters**				
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: highest subindex supported	1	- (only read access)			ro	U32
1: restore all default parameters	-	0		4294967295	rw	U32

Load Lenze setting.

- Corresponds to the device command C00002 = "0: Load Lenze setting".
- This command serves to reset the parameters of the active application to the Lenze setting which is stored in the firmware.

Subindex	Meaning	
	Read	Write
0	Max. supported subindex: 1	- (a write attempt triggers the error message 0x06010002.)
1	Loading of all parameters possible.	Load Lenze setting.

Read subindex 1

8th byte	7th byte	6th byte	5th byte	
Data 4	Data 3	Data 2	Data 1	
Bit 31 ... bit 1				Bit 0
0				0/1

[9-5] Assignment of the data telegram (read)

Bit	Setting	
Bit 0	0	Parameters cannot be loaded.
	1	Parameters can be loaded (Lenze).

Write subindex 1

In addition to the index and subindex, the telegram data must include the "load" signature (ASCII characters; ISO 8859) so that the Lenze setting can be loaded.

- A response according to the DS301 V4.02 specification occurs while writing with a wrong identifier.

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
"d" = 0x64	"a" = 0x61	"o" = 0x6F	"l" = 0x6C

[9-6] Assignment of the data telegram (write)

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

If a communication error or an internal error of the communication module or the controller occurs or is acknowledged (e. g. "trouble"), an error message is sent via the system bus. For each error, the telegram is interrupted once. By means of bit 31 this function can be activated or deactivated.

- This object is related to the Lenze code C00391.

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".				

[9-7] Data telegram assignment

Bit	Setting	
Bit 31	0	Emergency object valid.
	1	Emergency object invalid.

Note!

The identifier can only be changed in the "Emergency object invalid" status (bit 31 = 1).

I-1015

Index: **I-1015**	Name: **Inhibit time EMCY**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: inhibit time EMCY	0	0	0.1 ms	65535	rw	U32

Time which must elapse after an error message (I-1014) has been transmitted before further error messages can be sent via the bus.

- The entered value multiplied by "0.1" gives the delay time in [ms]. The values are automatically rounded up to whole values in [ms].
- This object is related to the Lenze code C00392.

I-1016

Index: **I-1016**	Name: **Consumer heartbeat time**				
Subindex	Default setting	Setting range (min. value | unit | max. value)		Access	Data type
0: highest subindex supported	32	- (only read access)		ro	U32
1 ... 32: consumer heartbeat time	0	0	4294967295	rw	U32

Monitoring time for the nodes 1 ... 32 to be monitored via heartbeat. ▸ Heartbeat protocol (📖 335)

- The parameterised time is rounded down to an integer multiple of 5 ms and must have a higher value than the heartbeat producer time of the node to be monitored.

Subindex	Meaning	Lenze code
0	Number of nodes to be monitored	
1 ... 32	Node-ID and heartbeat time of the node 1 ... 32 to be monitored	C00385/1...32

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]	

[9-8] 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	U32

Time interval for the transmission of the heartbeat telegram to one or several consumers. ▸ Heartbeat protocol (📖 335)

- The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is transmitted automatically as soon as a time > 0 ms is set. The monitoring function "Node Guarding" is deactivated in this case.
- This object is related to the Lenze code C00381.

I-1018

Index: **I-1018**	Name: **Identity object**					
Subindex	Default setting	Display area (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 identification number • The identification number allocated to Lenze by the "CAN in Automation e. V." organisation is "0x0000003B".	
2	Product code	
	0x94001	9400 StateLine
	0x94002	9400 HighLine / ServoPLC
	0x94004	9400 regenerative power supply module
3	Main version and subversion of the firmware	
4	Serial number	

I-1029

Index: **I-1029**	Name: **Error behaviour**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: highest subindex supported	1	- (only read access)			ro	U8
1: Communication error	0	0		2	rw	U8

This object serves to set the communication status to which the controller is to change after a bus off, a node/life guarding event or a heartbeat event.

Subindex	Meaning		Lenze code
1	Status change after bus off, node/life guarding event or heartbeat event:		C00625
	0	State change from "Operational" to "Pre-operational"	
	1	No state change	
	2	State change to "Stopped"	

I-1200

Index: **I-1200**	Name: **SDO1 server parameter**					
Subindex	Default setting	Display area (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

Identifier for the SDO server channel 1 (basic SDO channel).

▶ The basic SDO channel can neither be changed nor deactivated according to DS301 V4.02.

Subindex	Meaning
1	Receive identifier specification • For SDO server channel 1: node address (C00350) + 0x600
2	Send identifier specification • For SDO server channel 1: node address (C00350) + 0x580

8th byte		7th byte	6th byte	5th byte
Data 4		Data 3	Data 2	Data 1
Bit 31	Bit 30	Bit 29 ... bit 11		Bit 10 ... bit 0
0	**0**	**Extended identifier***		**11-bit identifier**

* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".

[9-9] 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	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for the SDO server channel 2.

- The server SDO parameter is only valid if the bit 31 is set to "0" for both transmission directions (subindex 1 and 2).
- In the Lenze setting, the SDO server channels 2 ... 10 are deactivated (bit 31 = "1").
- The identifier may only be changed if the SDO is invalid (bit 31 = "1").

Subindex	Meaning
1	Receive identifier specification
2	Send identifier specification
3	Node address of the client

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".				

[9-10] Data telegram assignment

Bit	Setting	
Bit 31	0	SDO valid.
	1	SDO 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

The parameter data channel 2 of the controller with the node address 4 is to be activated.

- For this purpose, the bit 31 in the subindexes 1 and 2 of the object I-1201 must be set to the value "0" (≡ "SDO valid").
- Both "Write request" commands must be transmitted from the master to the nodes via the basic SDO channel.

Calculation of the identifiers

- Identifier (COB-ID) = basic identifier + node address (node ID)
- Basic identifier SDO2 from the master to the drive: 1600 (0x640)
  → identifier = 0x640 + 0x4 = 0x644
- Basic identifier SDO2 from the drive to the master: 1472 (0x5C0)
  → identifier = 0x5C0 + 0x4 = 0x5C4

Resulting data (data 1 ... data 4)

8th byte		7th byte	6th byte	5th byte
Data 4		Data 3	Data 2	Data 1
Bit 31	Bit 30	Bit 29 ... bit 11		Bit 10 ... bit 0
0	0	Extended identifier = 0		11-bit identifier = 0x644
0x00		**0x00**	**0x06**	**0x44**

[9-11] Data telegram assignment for subindices 1

8th byte		7th byte	6th byte	5th byte
Data 4		Data 3	Data 2	Data 1
Bit 31	Bit 30	Bit 29 ... bit 11		Bit 10 ... bit 0
0	0	Extended identifier = 0		11-bit identifier = 0x5C4
0x00		**0x00**	**0x05**	**0xC4**

[9-12] Data telegram assignment for subindices 2

Assignment of the 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
0x23	**0x01**	**0x12**	**0x01**	**0x44**	**0x06**	**0x00**	**0x00**

[9-13] Assignment of the user data for writing the 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**

[9-14] Assignment of the user data for writing the subindex 2

I-1202

Index: **I-1202**	Name: **SDO3 server parameter**				
Subindex	Default setting	Setting range (min. value | unit | max. value)		Access	Data type
0: highest subindex supported	3	- (only read access)		ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0	4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0	4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)		ro	U32

Setting of the identifiers for the SDO server channel 3. For description see object I-1201.

I-1203

Index: **I-1203**	Name: **SDO4 server parameter**				
Subindex	Default setting	Setting range (min. value | unit | max. value)		Access	Data type
0: highest subindex supported	3	- (only read access)		ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0	4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0	4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)		ro	U32

Setting of the identifiers for the SDO server channel 4. For description see object I-1201.

I-1204

Index: **I-1204**	Name: **SDO5 server parameter**				
Subindex	Default setting	Setting range (min. value | unit | max. value)		Access	Data type
0: highest subindex supported	3	- (only read access)		ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0	4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0	4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)		ro	U32

Setting of the identifiers for the SDO server channel 5. For description see object I-1201.

I-1205

Index: **I-1205**	Name: **SDO6 server parameter**				
Subindex	Default setting	Setting range (min. value | unit | max. value)		Access	Data type
0: highest subindex supported	3	- (only read access)		ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0	4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0	4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)		ro	U32

Setting of the identifiers for the SDO server channel 6. For description see object I-1201.

I-1206

Index: **I-1206**	Name: **SDO7 server parameter**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for the SDO server channel 7. For description see object I-1201.

I-1207

Index: **I-1207**	Name: **SDO8 server parameter**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for the SDO server channel 8. For description see object I-1201.

I-1208

Index: **I-1208**	Name: **SDO9 server parameter**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for the SDO server channel 9. For description see object I-1201.

I-1209

Index: **I-1209**	Name: **SDO10 server parameter**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for the SDO server channel 10. For description see object I-1201.

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	- (only read access)			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	-	- (for RPDOs unused)			rw	U16
4: compatibility entry	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	-	- (for RPDOs unused)			rw	U16

Communication parameter for the reception of process data via RPDO1

Subindex	Meaning	Lenze code
1	Identifier RPDO1 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x200 + node ID	C00321/1
2	RPDO Transmission type according to DS301 V4.02 ▶ Transmission type (📖 308)	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".				

[9-15] Data telegram assignment

Bit	Setting	
Bit 30	0	RTR to this PDO possible (cannot be set).
	1	RTR to this PDO not possible (Lenze).
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").

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	- (only read access)			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	-	- (for RPDOs unused)			rw	U16
4: compatibility entry	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	-	- (for RPDOs unused)			rw	U16

Communication parameter for the reception of process data via RPDO2

Subindex	Meaning	Lenze code
1	Identifier RPDO2 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x300 + node ID	C00321/2
2	RPDO Transmission type according to DS301 V4.02 ▸ Transmission type (📖 308)	C00323/2

▶ For assignment of the data telegram see object I-1400.

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	- (only read access)			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	-	- (for RPDOs unused)			rw	U16
4: compatibility entry	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	-	- (for RPDOs unused)			rw	U16

Communication parameter for the reception of process data via RPDO3

Subindex	Meaning	Lenze code
1	Identifier RPDO3 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x400 + node ID	C00321/3
2	RPDO Transmission type according to DS301 V4.02 ▸ Transmission type (📖 308)	C00323/3

▶ For assignment of the data telegram see object I-1400.

I-1403

Index: **I-1403**	Name: **RPDO4 communication parameter**				
Subindex	Default setting	Setting range (min. value | unit | max. value)		Access	Data type
0: highest subindex supported	5	- (only read access)		ro	U8
1: COB-ID used by RPDO	0x500 + node ID	0	4294967295	rw	U32
2: transmission type	254	0	255	rw	U8
3: inhibit time	-	- (for RPDOs unused)		rw	U16
4: compatibility entry	-	- (reserved, write or read access results in error message 0x06090011)		rw	U8
5: event timer	-	- (for RPDOs unused)		rw	U16

Communication parameter for the reception of process data via RPDO4

Subindex	Meaning	Lenze code
1	Identifier RPDO4 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x500 + node ID	C00321/4
2	RPDO Transmission type according to DS301 V4.02 ▸ Transmission type (📖 308)	C00323/4

▶ For assignment of the data telegram see object I-1400.

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 ... 8: application object 1 ... 8	0	0	4294967295	rw	U32

The object I-1600 serves to receive parameter data as RPDO1.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO1

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**

[9-16] Data telegram assignment

IEC 61131 process data words are mapped. Only whole bytes can be mapped (the granularity of the mapping entries is one byte).

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 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

The object I-1601 serves to receive parameter data as RPDO2.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO2

▶ For assignment of the data telegram see object I-1600.

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 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

The object I-1602 serves to receive parameter data as RPDO3.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO3

▶ For assignment of the data telegram see object I-1600.

I-1603

Index: **I-1603**	Name: **RPDO4 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 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

The object I-1603 serves to receive parameter data as RPDO4.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO4

▶ For assignment of the data telegram see object I-1600.

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	- (only read access)			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, write or read access results in error message 0x06090011)			rw	U8
5: event timer	0 ms	0	ms	65535	rw	U16

Communication parameter for the transmission of process data via TPDO1

Subindex	Meaning	Lenze code
1	Identifier TPDO1 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x180 + node ID	C00320/1
2	TPDO transmission type according to DS301 V4.02 ▸ Transmission type (📖 308)	C00322/1
3	Minimum time between the transmission of two equal TPDOs (see DS301, V4.02).	C00324/1
5	Cycle time with which the PDOs are transmitted with the transmission type "254".	C00356/1

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".				

[9-17] Data telegram assignment

Bit	Setting	
Bit 30	0	RTR to this PDO possible (Lenze).
	1	RTR to this PDO not possible (cannot be set).
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 3 - inhibit time

The delay time can only be changed if the PDO is inactive (subindex 1, bit 31 = 1). The entered value multiplied by 0.1 results in the delay time in [ms]. The calculated delay time is always rounded down to an inter value.

Example:

- Entered value: 26
- Calculated time = 26 * 0.1 [ms] = 2.6 [ms] → delay time = 2 [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	- (only read access)			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, write or read access results in error message 0x06090011)			rw	U8
5: event timer	0 ms	0	ms	65535	rw	U16

Communication parameter for the transmission of process data via TPDO2

Subindex	Meaning	Lenze code
1	Identifier TPDO2 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x280 + node ID	C00320/2
2	TPDO transmission type according to DS301 V4.02 ▸ Transmission type (📖 308)	C00322/2
3	Minimum time between the transmission of two equal TPDOs (see DS301, V4.02).	C00324/2
5	Cycle time with which the PDOs are transmitted with the transmission type "254".	C00356/2

- For assignment of the data telegram see object I-1800.

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	- (only read access)			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, write or read access results in error message 0x06090011)			rw	U8
5: event timer	0 ms	0	ms	65535	rw	U16

Communication parameter for the transmission of process data via TPDO3

Subindex	Meaning	Lenze code
1	Identifier TPDO3 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x380 + node ID	C00320/3
2	TPDO transmission type according to DS301 V4.02 ▸ Transmission type (📖 308)	C00322/3
3	Minimum time between the transmission of two equal TPDOs (see DS301, V4.02).	C00324/3
5	Cycle time with which the PDOs are transmitted with the transmission type "254".	C00356/3

▶ For assignment of the data telegram see object I-1800.

I-1803

Index: **I-1803**	Name: **TPDO4 communication parameter**					
Subindex	Default setting	Setting range (min. value | unit | max. value)			Access	Data type
0: highest subindex supported	5	- (only read access)			ro	U8
1: COB-ID used by TPDO	0x480 + 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, write or read access results in error message 0x06090011)			rw	U8
5: event timer	0 ms	0	ms	65535	rw	U16

Communication parameter for the transmission of process data via TPDO4

Subindex	Meaning	Lenze code
1	Identifier TPDO4 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x480 + node ID	C00320/4
2	TPDO transmission type according to DS301 V4.02 ▸ Transmission type (📖 308)	C00322/4
3	Minimum time between the transmission of two equal TPDOs (see DS301, V4.02).	C00324/4
5	Cycle time with which the PDOs are transmitted with the transmission type "254".	C00356/4

▶ For assignment of the data telegram see object I-1800.

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 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

The object I-1A00 serves to transmit parameter data as TPDO1.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entry 1 ... 8 for TPDO1

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

[9-18] Data telegram assignment

IEC 61131 process data words are mapped. Only whole bytes can be mapped (the granularity of the mapping entries is one byte).

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 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

The object I-1A01 serves to transmit parameter data as TPDO2.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO2

▶ For assignment of the data telegram see object I-1A00.

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 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

The object I-1A02 serves to transmit parameter data as TPDO3.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO4

▶ For assignment of the data telegram see object I-1A00.

I-1A03

Index: **I-1A03**	Name: **TPDO4 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 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

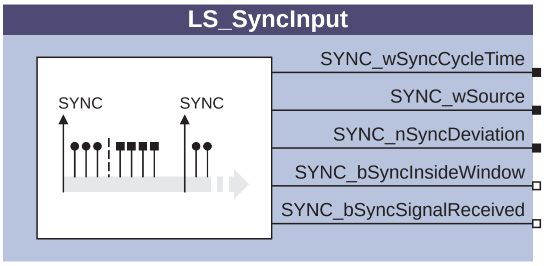
The object I-1A03 serves to transmit parameter data as TPDO4.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for TPDO4

▶ For assignment of the data telegram see object I-1A00.

9.11 System block "LS_SyncInput"

The **LS_SyncInput** system block provides status information in the function block editor about the sync telegram received via the system block.



Outputs

Identifier Data type	Value/meaning	
SYNC_wSyncCycleTime WORD	Sync cycle time in [µs] • Time with which the internal phase-locking loop (PLL) visualises the synchronisation signals. The time must be set in C01121 in accordance with the cycle of the synchronisation source selected in C01120.	
SYNC_wSource WORD	Synchronisation source selected in C01120:	
	0	Off
	1	CAN on board
	2	CAN module
	4	Module in MXI1
	5	Module in MXI2
	6	Digital input 1
	7	Digital input 2
	8	Digital input 3
	9	Digital input 4
	10	Digital input 5
	11	Digital input 6
	12	Digital input 7
	13	Digital input 8
SYNC_nSyncDeviation INT	Deviation of the synchronisation signal in [increments] • ±16000 increments ≡ ±1 ms	
SYNC_bSyncInsideWindow BOOL	Status signal "Synchronisation signal within time slot" **Note!** If you use this signal in the application, observe the change in behaviour from software version V7.= onwards described in the following subchapter! ▸ Behaviour of the status signal bSyncInsideWindow (📖 366)	
	TRUE	The last synchronisation signal has been around the expected value within the time slot parameterised in C01123.
SYNC_bSyncSignalReceived BOOL	Status signal "Receive synchronisation signal"	
	TRUE	Synchronisation signal has been received.

▸ Synchronisation of PDOs via sync telegram (📖 310)

9.11.1 Behaviour of the status signal bSyncInsideWindow

C01123 serves to set a time slot for monitoring the synchronisation signal. If the synchronisation signal received via the bus is in this time slot (around the expected time of the synchronisation signal), the *bSyncInsideWindow* output is set to TRUE.

Up to and including software version V6.0 the following applies:

Due to an error in the implementation, the phase position set in C01122 is included in the calculation of the time slot. The time slot effective for monitoring around the expected time of the synchronisation signal is thus increased by the amount of the set phase position.

Example:

- Sync phase position (C01122) = 400 µs
- Sync tolerance (C01123) = 20 µs

→ The time slot for monitoring has a size of 420 µs!

The following applies from software version V7.0 onwards:

The faulty inclusion of the phase position set in C01122 into the calculation of the time slot has been removed. The time slot for monitoring the synchronisation signal only corresponds to the sync tolerance set in C01123.

Example:

- Sync phase position (C01122) = 400 µs
- Sync tolerance (C01123) = 20 µs

→ The time slot for monitoring has a size of 20 µs!

Feedbacks and their remedies

If the *bSyncInsideWindow* status signal is used in existing systems, this remedy reduces the monitoring window by the amount of the phase position if it is non-zero. This may cause an unwanted activation of the monitoring of the synchronisation signal programmed by the user.

Remedy: When the syn tolerance is increased (C01123) by the amount of the phase position set in C01122, the compatible state is restored.

10 Safety engineering

The controller can be equipped with a safety module. The individual safety module types have a different range of functions to optimally meet different requirements.

"Integrated safety technology" stands for user-related safety functions that are applicable to the protection of persons working with machines and the machine protection.

The motion functions are furthermore executed by the controller. The safety modules monitor the reliable compliance with limit values and provide safe inputs and outputs. If limit values are exceeded, the safety modules start control functions for the fault scenario in accordance with EN 60204-1 directly in the controller.

The safety functions are suitable for applications according to IEC 61508, SIL 3 and, depending on the module, meet the requirements of EN 954, part 1, up to control category 4.

Note!

For detailed information about the integrated safety technology, please see the manual for the safety module.

10.1 Integration into the application

If a safety function is requested, the safety engineering activates a corresponding safe monitoring function. The standstill function, however, is only executed directly if the "Safe torque off" function (STO) is activated. For the other safety functions, an action of the controller is required, which is safely monitored. The implementation of the corresponding action (e.g. braking, braking to standstill, holding of the standstill position) must be carried out by the application.

System block "LS_SafetyModuleInterface"

For the transmission of the control and status information from the safety module to the application, the **LS_SafetyModuleInterface** system block is provided in the function block editor of the »Engineer«. (📖 370)

System block "LS_Limiter"/basic function "Limiter"

Furthermore the **LS_Limiter** system block which contains the basic function "Limiter" is provided in the function block editor for the connection of safety engineering to the application. (📖 513)

For one thing, the basic function "Limiter" provides a parameterisation interface in the »Engineer« for a comfortable setting of limit positions, limited speeds, and limit values, and for another it enables the drive to be braked specifically **after request** through the safety module.

General procedure

1. Activation of the safety function on the safety module (e. g. SS1 - safe stop 1).
   → Monitoring starts.
2. The safety module informs the controller via a control word that the safety function has been activated.
3. The application evaluates the control word and starts the required motion sequence (e.g. braking).

Note!

If communication to the controller is interrupted, e.g. by switching off the controller, the safety module responds as follows:

- Fault stop with STO is activated.
- "Warning" error message is transmitted.
- The LED "ME" is blinking.

The required fault acknowledgement (AIE) is possible via terminal or safety bus.

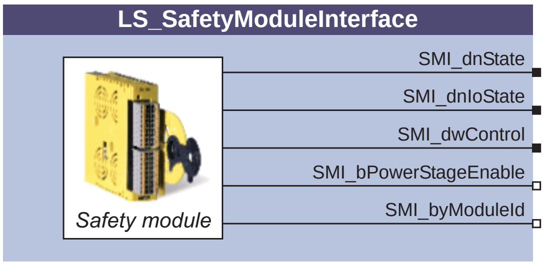
10.2 Selecting the required safety module

The safety module expected by the application and the controller is selected in C00214.

- In the »Engineer« this setting is carried out automatically by assigning the device modules to the controller, i.e. the »Engineer« automatically sets C00214 according to the safety module selected.
- If the safety module selected in C00214 does not correspond to the safety module type connected, an error will be activated.

10.3 System block "LS_SafetyModuleInterface"

The **LS_SafetyModuleInterface** system block is the interface to the safety module in the function block editor.



Outputs

Identifier Data type	Value/meaning	
SMI_dnState DINT	Bit coded status information from the safety module ▸ Status information (🕮 371)	
SMI_dnIoState DINT	Bit coded I/O status information from the safety module ▸ I/O status information (🕮 372)	
SMI_dwControl DWORD	Bit coded control information from the safety module ▸ Control information (🕮 373)	
SMI_bPowerStageEnable BOOL	Status signal "Inverter enable"	
	TRUE	Inverter is enabled by the safety module.
SMI_byModuleId BYTE	ID of the safety module available in the controller	

10.3.1 Status information

Via the bit-coded status signal *SMI_dnState* of the **LS_SafetyModuleInterface** SB, the SM3xx safety module transmits the status of safety functions to the application.

▶ Which bits are supported depends on the safety module used.

Bit coding of the status signal SMI_dnState		
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.
5	EC_SS2	Error stop category 2: "Safe stop 2 (SS2)" function is active.
8	SLS1 monitored	Safely limited speed 1 is activated and complied with.
9	SLS2 monitored	Safely limited speed 2 is activated and complied with.
10	SLS3 monitored	Safely limited speed 3 is activated and complied with.
11	SLS4 monitored	Safely limited speed 4 is activated and complied with.
12	SDIpos monitored	Safe positive direction of rotation (SDIpos) is activated and complied with.
13	SDIneg monitored	Safe negative direction of rotation (SDIneg) is activated and complied with.
14	Fault active	SM3xx safety module has the error status (trouble or warning).
Bits not listed are reserved for future extensions!		

Tip!

For decoding the status signal into individual boolean status signals, simply connect the output *SMI_dnState* to the **L_DevSMStateDecoder** FB which is available in the function library from V2.0.

10.3.2 I/O status information

Via the bit coded status signal *SMI_dnIoState* of the **LS_SafetyModuleInterface** SB, the SM3xx safety module transmits the status of the safe inputs and outputs to the application.

▶ Which bits are supported depends on the safety module used.

Bit coding of the SMI_dnIoState status signal		
Bit	**Name**	**Meaning**
0	SD-In1	Sensor input 1 in ON state.
1	SD-In2	Sensor input 2 in ON state.
2	SD-In3	Sensor input 3 in ON state.
3	SD-In4	Sensor input 4 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)
12	SD-Out1	Safe output 1 (feedback output) in the ON state.
Bits not listed are reserved for future extensions!		

Tip!

For decoding the status signal into individual boolean status signals, simply connect the output *SMI_dnIoState* to the **L_DevSMStateDecoderIO** FB which is available in the function library from V2.0.

10.3.3 Control information

Via the bit coded control signal *SMI_dwControl* of the **LS_SafetyModuleInterface** SB, the SM3xx safety module transmits information on safety functions requested, or on active safety functions to the application.

- Several safety functions can be requested/active at the same time.
- Which bits are supported depends on the safety module used.

Note!

The corresponding actions (e.g. braking, braking to standstill, holding of the standstill position) must be executed by the application, e.g. via the basic function "Limiter". (513)

- To integrate the basic function "Limiter", the output *SMI_dwControl* is to be connected to the input *LIM_dwControl* of the **LS_Limiter** system block.

Bit coding of the control signal SMI_dwControl

Bit	Name	Meaning
1	SS1 active	"Safe stop 1 (SS1)" function is active. • After the parameterised stopping time has elapsed, bit 0 of the status signal *SMI_dnState* (STO active) is set.
2	SS2 active	"Safe stop 2 (SS2)" function is active. • After the parameterised stopping time has elapsed, bit 16 (SOS monitored) is set.
3	SLS1 active	"Safely limited speed 1 (SLS1)" function is active. • After the parameterised braking time has elapsed, bit 8 of the status signal *SMI_dnState* (SLS1 monitored) is additionally set.
4	SLS2 active	"Safely limited speed 2 (SLS2)" function is active. • After the parameterised braking time has elapsed, bit 9 of the status signal *SMI_dnState* (SLS2 monitored) is additionally set.
5	SLS3 active	"Safely limited speed 3 (SLS3)" function is active. • After the parameterised braking time has elapsed, bit 10 of the status signal *SMI_dnState* (SLS3 monitored) is additionally set.
6	SLS4 active	"Safely limited speed 4 (SLS4)" function is active. • After the parameterised braking time has elapsed, bit 11 of the status signal *SMI_dnState* (SLS4 monitored) is additionally set.
7	SDIpos active	"Safe positive direction of rotation (SDIpos)" function is active.
8	SDIneg active	"Safe negative direction of rotation (SDIneg)" function is active.
9	ES active	"Confirm button (ES)" function for motion functions in special operation is active.
10	SLI active	"Safely limited increment (SLI)" function is active.
11	OMS active	"Operation mode selector (OMS)" function for special operation is active.
12	SLP1 active	"Safely-limited position 1 (SLP1)" function is active.
13	SLP2 active	"Safely-limited position 2 (SLP2)" function is active.
14	SLP3 active	"Safely-limited position 3 (SLP3)" function is active.
15	SLP4 active	"Safely-limited position 4 (SLP4)" function is active.

Bits not listed are reserved for future extensions!

Bit coding of the control signal SMI_dwControl		
Bit	**Name**	**Meaning**
16	SOS active	"Safe operating stop (SOS)" function is active. • The safe operating stop is monitored. • The function becomes active after the "Safe stop 2 (SS2)" function has been executed.
23	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_dnState* (STO active) is set according to the emergency stop function parameterised.
Bits not listed are reserved for future extensions!		

Tip!

For decoding the control signal into individual boolean control signals, simply connect the output *SMI_dwControl* to the **L_DevSMControlDecoder** FB which is available in the function library from V2.0.

11 Basic drive functions

In this chapter the basic (drive) functions of the "Servo Drives 9400" are described, to which the active application can access via defined, internal interfaces, and which can be carried out in the following way, depending on the controller type (StateLine or HighLine) and the Motion Control licence available:

Parameter setting by means of »Engineer« or keypad

In each licence level the basic functions can be parameterised in the »Engineer« via a corresponding dialog or alternatively via the keypad.

In the case of the 9400 StateLine (licence level Motion Control StateLevel), the interconnection of the internal interfaces is defined by the technology application selected.

Configuration in the »Engineer« function block editor

The »Engineer« additionally provides the graphic function block editor for the 9400 HighLine which can be used to reconfigure and extend the application interconnection by individual functions using the function library.

Programming according to IEC 61131-3 in the »PLC Designer«*

For the 9400 HighLine with the licence level Motion Control PLC the basic functions are also provided as separate system blocks in the »PLC Designer«, which, if required, can be integrated in the control configuration, and which then can be accessed from the IEC 61131-3 program via the corresponding system variables.

* In preparation!

11.1 General information

11.1.1 Internal state machine

The execution of the different basic functions is internally controlled by a state machine which can adopt the following "function states":

[11-1] Function states of the state machine "Basic functions"

The state machine ensures that:

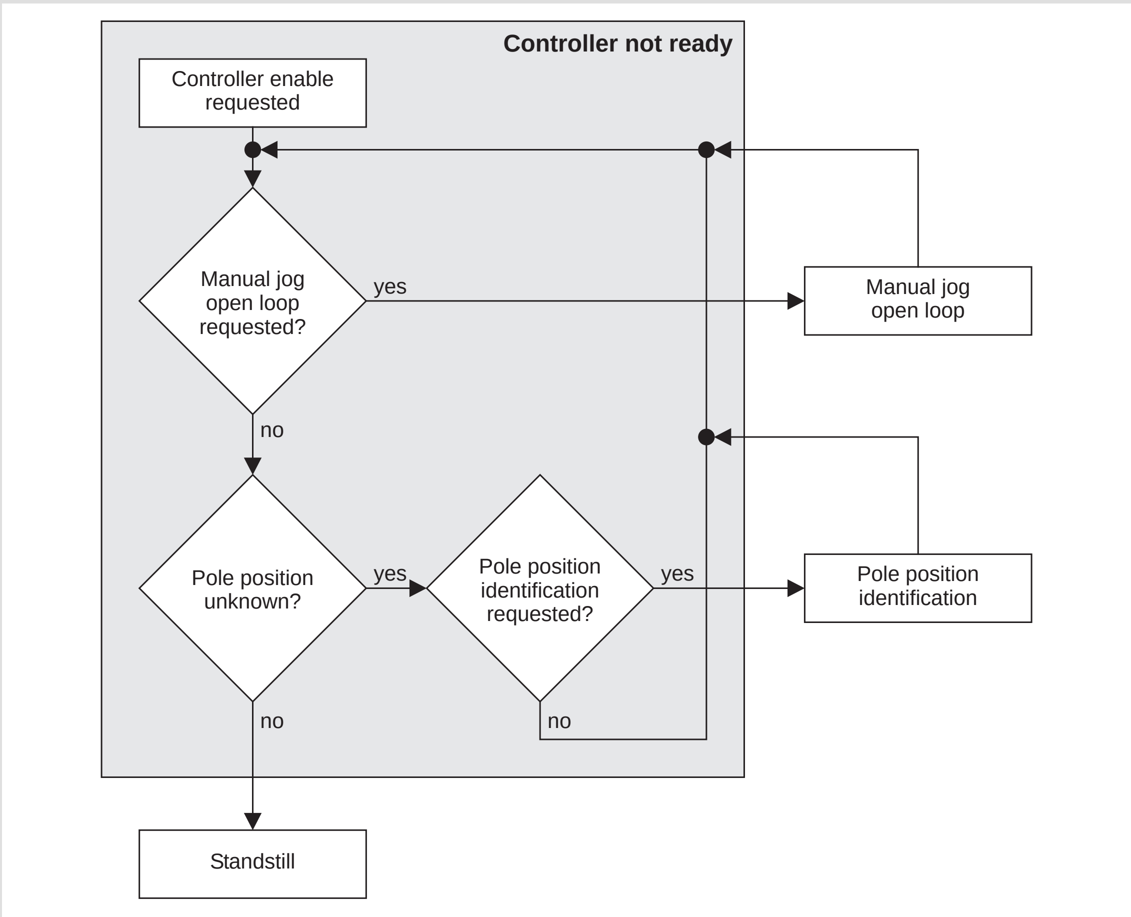
- one basic function at a time adopts the control of the drive.
- only the basic function with the highest priority (= smallest number) is executed if several basic functions are activated at the same time. ▶ Priorities (📖 381)
- the drive always has a defined state both in case of error and in normal operation.

Note!

The basic functions "Limiter" and "Brake control" run autonomously, but are able to control the state machine to a certain function state, if required.

The function states are not to be confused with the device states ("Operation", "Fault active", "Device is switched on", etc.) of the controller. ▶ Device states (📖 105)

From software version V7.0 onwards, the basic functions "Manual jog, encoderless" and "Pole position identification" are additionally available for the setting-up operation. Both basic functions can only be requested when the controller is inhibited and with the "Controller not ready" function state:



[11-2] Sequence diagram for basic functions "Manual jog, encoderless" and "Pole position identification"

11.1.2 Function states

Tip!

In C02530 the currently active function state is displayed.

State "Initialisation"

If the controller has completed the device initialisation, the function state "Initialisation" is passed through automatically.

- In the "Initialisation" function state, the process data required for processing the basic functions are initialised.
- The monitoring is not active yet.
- The basic functions are not yet processed (e.g. brake control) and cannot yet be parameterised or activated either.
- If the initialisation of the basic functions is completed and no fault is available, a change-over to the basic state "Drive at standstill" is automatically effected.

State "Controller not ready"

In this function state the pulse inhibit is set in the controller, which means that the power output stages are high-resistance and the drive cannot be controlled.

State "Manual jog encoderless active"

From software version V7.0 onwards, the drive can be controlled without feedback (encoderless) in this function state, e.g. for a setting-up operation or in the event of service when the feedback system fails. ▶ Manual jog, encoderless (📖 410)

State "Identification of pole position active"

From software version V7.0 onwards, an identification of pole position can be executed in this function state in order to detect the pole position for the motor encoder that is currently activated in C00495. ▶ Pole position identification (📖 581)

State "Drive at standstill"

This "basic state" is automatically adopted if no other state is active.

- The setpoints for speed and acceleration are set to "0".
- The drive is position-controlled.
- No error has occurred and quick stop is not active.
- Every basic function can be activated out of this state.

State "Drive is stopped"

This function state is automatically passed through when a basic function is deactivated.

- If the drive is not yet in the standstill state, it is decelerated to standstill via a parameterisable deceleration ramp.
- If a basic function is activated during the "stopping" process, this basic function takes over the control of the drive from the current speed on and the function state "Drive is stopped" is abandoned.
- If the drive is at standstill, a change-over to the basic state "Drive at standstill" is automatically effected.

State "Manual jog active"

In this function state the drive can be traversed manually to the right or left direction of rotation ("Inching mode"). Manual jog (398)

- If the home position is known to the controller, the software limit positions set and a potentially connected travel range limit switch are monitored.
- "Retracting" from an activated travel range limit switch is also possible.

State "Homing active"

In this function state the home position and the machine measuring system for the drive can be determined. Homing (419)

- The home position can be specified by an active homing or by reference setting.
- A redetermination of the home position is only required in case of recommissioning or in case of service (e.g. when drive components are exchanged) or after travel commands have been executed which reset the reference.

State "Positioning active"

In this function state all positioning types (absolute, relative, modulo, continuous, touch probe etc.) can be executed. Positioning (482)

- In the position-controlled mode, the drive executes a time-controlled point-to-point setpoint generation based on the defined motion profile.

State "Setpoint follower active"

In this function state the drive directly follows the defined setpoint.

- The setpoint can be optionally defined as speed, torque, or position via three separate basic functions:
  - Speed follower (502)
  - Torque follower (508)
  - Position follower (494)

State "Quick stop active"

This function state is active if quick stop has been activated by the user. Quick stop (391)

- The drive is brought to standstill within the deceleration time parameterised, irrespective of the setpoint defined.
- If the quick stop is cancelled again by the user, a change-over to a setpoint-generating basic function (e. g. "Speed follower") is effected, if requested.

Note!

For the encoderless motor control types (from software version V3.0) **the following applies:**

The "Quick stop active" function state is also activated when DC-injection braking is executed.

Tip!

Quick stop can also be set as error response for many monitoring functions ("quick stop by trouble"). Detailed information can be found in the chapter "Diagnostics & fault analysis".

The source from which the quick stop was activated is shown in a bit coded manner in C00159.

"Fault" state

This function state is active if a fault has occurred and the controller is in the "Fault active" or "Quick stop by trouble active" device state.

- The function state can only be abandoned by acknowledging the error if the error is removed.

11.1.3 Interrupting/replacing states

An active function state cannot be interrupted or replaced by the activation of another function state. However, the following exceptions apply:

- The "Initialisation" state replaces all other states.
- The "Fault" state can replace all other states except "Initialisation".
- The "Controller not ready" state can replace all other states except "Error" and "Initialisation".
- The "Quick stop active" state can replace all other states except "Initialisation", "Error" and "Controller not ready".

11.1.4 Priorities

The function states are assigned to priorities so that, if several basic functions are activated at the same time, it is always changed to the function state with the highest priority:

Priority	Function state	Executable basic function
1	Initialisation	-
2	Fault	-
3	Controller is not ready	-
4	Quick stop active	▸ Quick stop (📖 391)
5	Manual jog active	▸ Manual jog (📖 398)
6	Homing active	▸ Homing (📖 419)
7	Positioning active	▸ Positioning (📖 482)
8	Setpoint follower (position) active	▸ Position follower (📖 494)
9	Setpoint follower (speed) active	▸ Speed follower (📖 502)
10	Setpoint follower (torque) active	▸ Torque follower (📖 508)
10	Brake check	▸ Brake control (📖 528)
12	Drive is stopped	▸ Stop (📖 387)
13	Manual control open loop active	▸ Manual jog, encoderless (📖 410)
14	Pole position identification active	▸ Pole position identification (📖 581)
1 ≡ highest priority; 14≡ lowest priority		

Note!

The basic state "Drive at standstill" is automatically adopted if no other state is active.

11.1.5 Requesting control via a basic function

Enable input "bEnable"

The basic functions "Manual jog", "Homing" and "Positioning" and the three setpoint followers each possess an *bEnable* enable input, via which the control of the corresponding basic function can be requested.

- If no other basic function and no error status is active, a change-over to the corresponding function state is effected, and the basic function can be controlled now.
- If several enable inputs are set to TRUE at the same time, the change-over to the function state is effected with the highest priority.

Status outputs "bEnabled", "bActive" and "bDone"

If the basic function is enabled, the *bEnabled* status output of the basic function is set to TRUE and the corresponding drive motion can be started via the control inputs of the basic function.

- If the basic function is currently carrying out a drive movement, this is shown by a TRUE signal at the status output *bActive*.

- The basic functions "Speed follower", "Torque follower", and "Position follower" are only provided with the status output *bEnabled*, as the drive directly follows the setpoint selection after being enabled.
- The basic functions "Homing" and "Positioning" are furthermore provided with a status output *bDone* showing that the drive movement started (Homing or positioning) has been completed.

Priority	Basic function	Status outputs		
		bEnabled	*bActive*	*bDone*
1	Manual jog	☑	☑	
2	Homing	☑	☑	☑
3	Positioning	☑	☑	☑
4	Speed follower	☑		
5	Torque follower	☑		
6	Position follower	☑		

Re-deactivating the enable of a basic function

When the *bEnable* enable input of the active basic function is reset to FALSE, the control inputs of the basic function are inhibited and the status outputs *bEnabled*, *bActive* and *bDone* are reset to FALSE.

- If the drive is not at standstill, it is brought to standstill within the deceleration time for Stop if no other basic function takes over the control of the drive. Here a change-over from the active function state via the function state "Drive is stopped" back to the basic state "Drive at standstill" is effected.
- When the enable input of another basic function is set to TRUE, this basic function adopts the control of the drive immediately.

11.1.6 Start acceleration/acceleration reduction when the basic function changes

In order to make the transitions during the changeover between the single basic functions as jerk-free as possible, i.e. preventing acceleration steps, the current setpoint acceleration is used as starting value for the new basic function (see the following table).

	to						
from	Position, speed or torque follower	Manual jog	Homing	Positioning	Error/controller not ready	Stop	Quick stop*
Position follower	0	A	A	A	0	A	0
Speed follower	0	A	A	A	0	A	0
Torque follower	0	0	0	0	0	0	0
Manual jog	0	-	B	B	0	B	0
Homing	0	B	-	B	0	B	0
Positioning	0	B	B	-	0	B	0
Error/controller not ready	0	0	0	0	-	0	0
Stop	0	B	B	B	0	-	0
Quick stop*	0	A	A	A	0	A	-

Legend:

0	The start acceleration is defined with "0", thus no acceleration reduction is required.
A	• Acceleration value is generated from the differentiation and filtering (C02562) of the active setpoint speed. • Jerk = Maximum value from transition jerk (defined via C02545) and jerk of the "new" profile data.
B	• Acceleration value is taken from the setpoint generator (e.g. profile generator). • Jerk = maximum value from the jerk of the "old" and "new" profile data.

* Also quick stop by trouble

Reduction of the start acceleration

Depending on the acceleration and S-ramp time parameterised in the basic function, the start acceleration is reduced.

The following applies for software versions lower than V7.0:

▶ The start acceleration is always reduced with the jerk of the new basic function.
Setting the S-ramp time (385)

Note!

Very low jerks cause very high speeds!
See also the following chapter "Setting the S-ramp time". (385)

Since this behaviour is mostly not wanted or expected, the acceleration is reduced from software version V7.0 as described in the following section.

The following applies from software version V7.0 onwards:

- The start acceleration is reduced with the maximum jerk of the old or new basic function.

Transition of a profile-generating to a profile-generating basic function

The corresponding jerk results from the profile data:

$$\text{Jerk} = \frac{\text{Profile acceleration/deceleration}}{\text{Verschliffzeit}}$$

Transition of a non-profile-generating to a profile-generating basic function

- The jerk of the profile-generating basic function results from the profile data:

$$\text{Jerk} = \frac{\text{Profile acceleration/deceleration}}{\text{Verschliffzeit}}$$

- Since a non-profile-generating basic function has no defined jerk, a "transition jerk" is used which results from the reference acceleration and the reference S-ramp time parameterised in C02545.

$$\text{Transition jerk} = \frac{\text{Bezugsacceleration}}{\text{Bezugsverschliffzeit}} = \frac{\text{C00011 / 1 ms}}{\text{C02545}}$$

- With a Lenze setting C02545 = 0.001 s, a maximum jerk occurs, i.g. the start acceleration is reduced in one cycle (1 ms).
- The setting C02545 = 0.000 s results in a compatible behaviour lower than V7.0.

Tip!

Profile-generating basic functions are:
"Stop", "Manual jog", "Homing", "Positioning"

Non-profile-generating basic functions are:
"Quick stop", "Position follower", "Speed follower", "Torque follower"

11.1.7 Setting the S-ramp time

For path planning, various basic functions serve to build up or reduce the acceleration linearly. The motion profile causes less structural vibrations and the gearboxes are protected.

The smoothening (jerk) is calculated via the S-ramp time and the maximum acceleration permitted in the profile:

$$\text{Jerk} = \frac{\text{maximum acceleration}}{\text{S-ramp time}}$$

[11-3] Formula for calculating the jerk for acceleration and deceleration phases

S-ramp times can be set in the given parameters for the following basic functions:

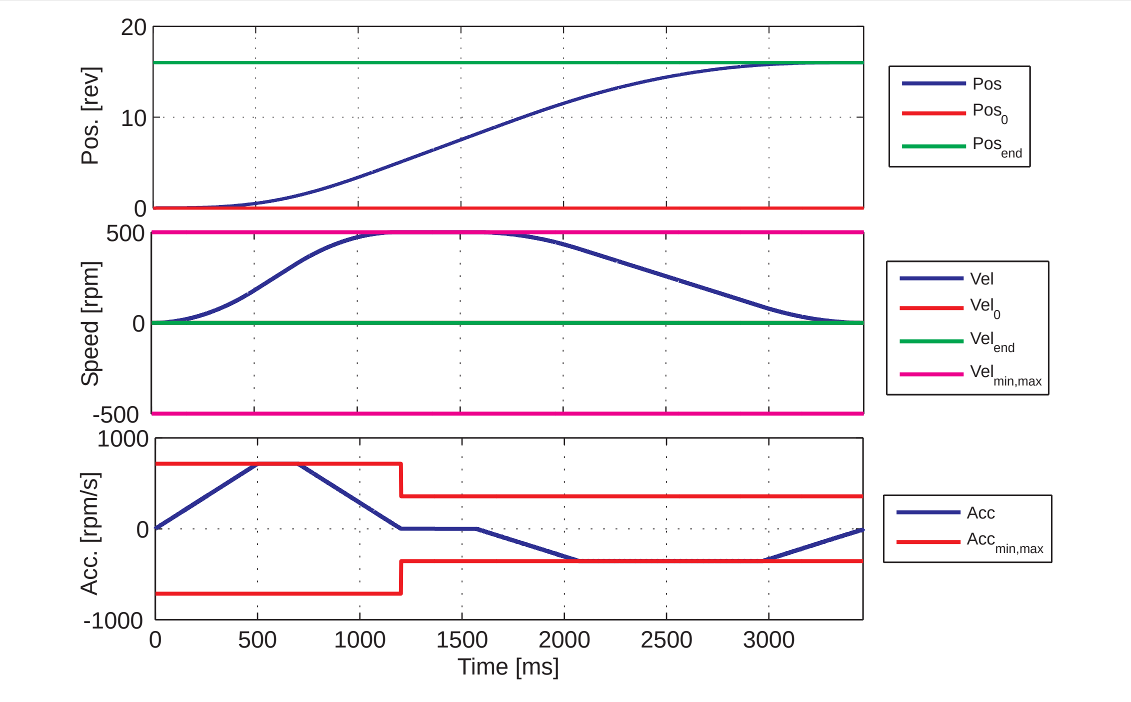
Basic function	Parameter for S-ramp time
Stop	C02611
Quick stop	C00106
Manual jog	C02624
Homing	C02648
Positioning	The S-ramp time is selected via FB L_PosPositionerTable or FB L_PosProfileTable.

Stop!

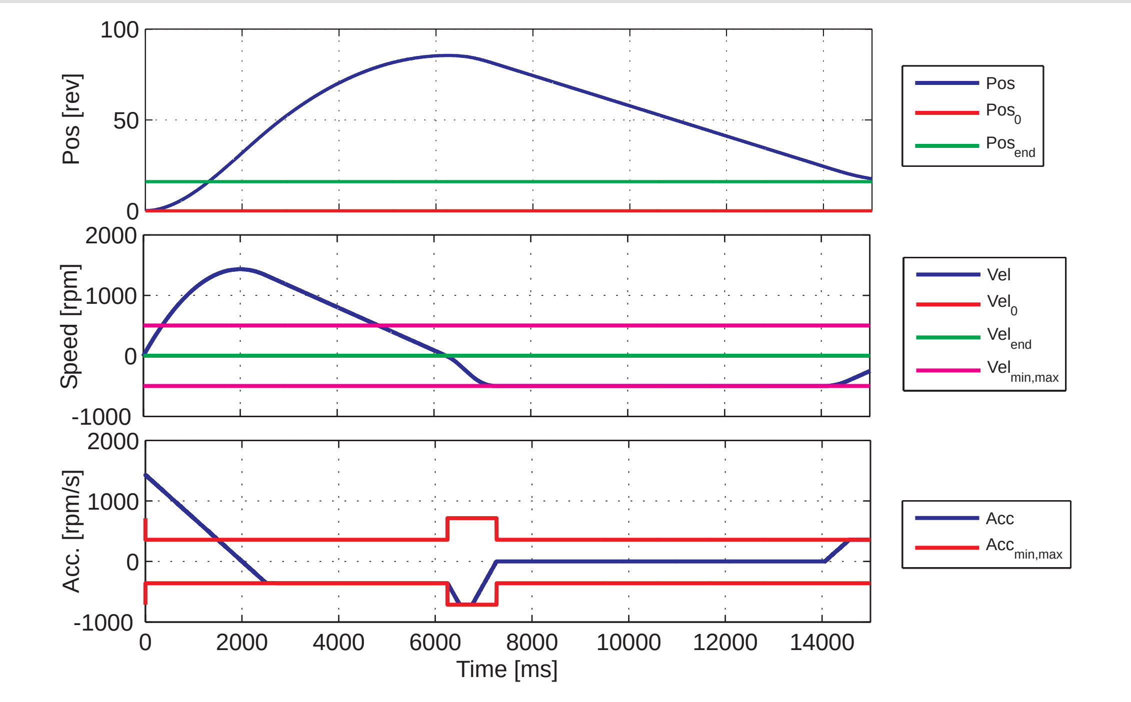
When it is switched to another basic function, the start acceleration is reduced with the jerk of the new basic function. **A small jerk causes very high speeds!**

Remedy: Avoid unnecessarily long S-ramp times. Set the profile parameters of the different basic functions so that the jerk is roughly the same for all basic functions.

Examples



[11-4] Example 1: Point-to-point positioning from standstill without start acceleration



[11-5] Example 2: Point-to-point positioning from standstill with start acceleration

In the example 2, the slow reduction of the start acceleration results in very high speeds!

11.2 Stop

The standard stop (in the following called "stop") of the drive will be automatically activated by the internal state machine if a basic function is deactivated and the drive is not yet at standstill.

- The drive is braked to standstill via a parameterisable deceleration ramp.
  - While the drive is braked to standstill, the state machine is in the "Drive is stopped" function state.
  - If meanwhile another basic function is activated, this basic function takes over the control of the drive from the current speed on and the function state "Drive is stopped" is abandoned.
  - If the drive is at standstill, a change-over to the basic state "Drive at standstill" is automatically effected.
- An acceleration phase active at the time of activating the stopping process is considered by the standard stop, i.e. the current acceleration is first led to "0" with the parameterised S-ramp time before the real deceleration process starts.
- If the controller is enabled while the shaft is coasting (controller inhibit and pulse inhibit are deactivated), the drive is braked from the current speed to standstill.

STOP

Stop!

The basic functions "Speed follower", "Torque follower", and "Position follower" do not take over the control of the from the current speed, but immediately with the setpoint defined, which may cause a jerk!

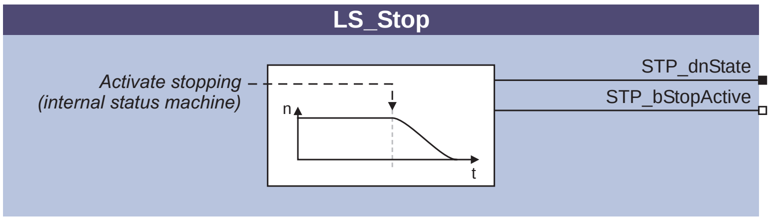
Note!

- As the stop function takes into account the acceleration active at the moment of activation, the deceleration of the stop function should always be set greater than the deceleration of the active process in order to avoid a possible overshoot.
- If the stop function is activated while the basic function "Torque follower" or the states "Controller inhibited" or "Error" are active, the drive is braked to standstill starting from the current speed and without taking into account the current acceleration.

Start acceleration/acceleration reduction when the basic function changes (📖 383)

11.2.1 Internal interfaces | "LS_Stop" system block

The **LS_Stop** system block provides the internal interfaces for the basic function "Stop" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Outputs

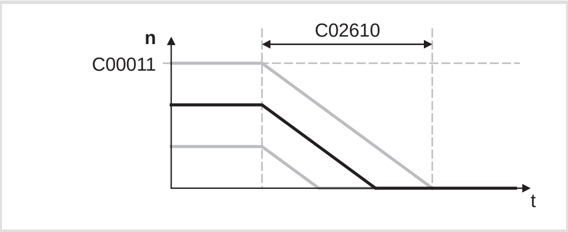
Identifier DIS code | data type	Value/meaning	
STP_dnState C02616 | DINT	Status (bit coded) • When the basic function is not active, all bits are set to "0". • Bits which are not listed are not assigned with a status (always "0").	
	Bit 1	Drive is braked to standstill. • The internal state machine is in the "Drive is stopped" function state.
	Bit 2	Drive is at standstill. • The internal state machine is in the "Drive at standstill" function state.
	Bit 3	Deceleration phase is active.
	Bit 5	CCW rotation is active.
STP_bStopActive C02617 | BOOL	Status signal "Stop is active"	
	TRUE	The drive is braked to standstill or is at standstill. • The internal state machine is in the "Drive is stopped" or "Drive at standstill" function state.

11.2.2 Parameter setting

- Parameterisation dialog in the »Engineer«: Tab **Application parameter** → Dialog level *Overview* → *All basic functions* → *Stop*
- Short overview of parameters for standard stop :

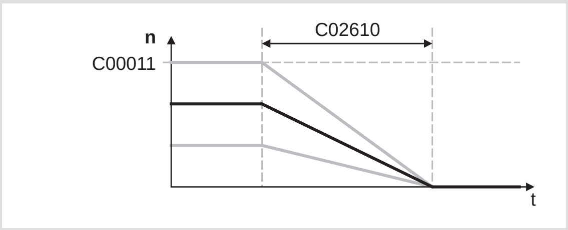
Parameter	Information
C02610	Deceleration time for stop
C02611	S-ramp time for stop
C02612	Ref. for decel. time of stop

Parameter setting of stop



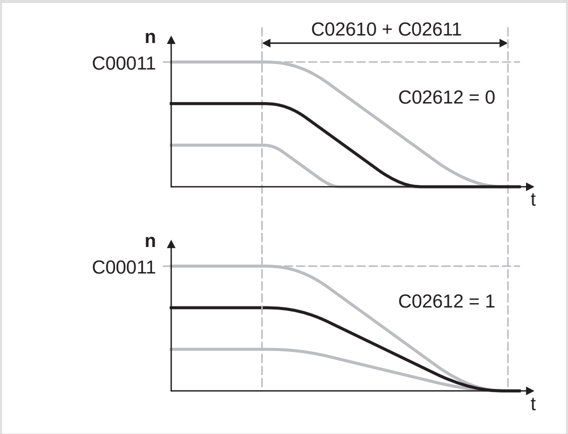
[11-6] Deceleration time referred to the motor reference speed

- The deceleration time for stop set in C02610 refers to a speed variation from the motor reference speed (C00011) to standstill, i. e. the deceleration is constant.



[11-7] Deceleration time referred to the current speed

- When C02612 is set = "1", the deceleration time refers to the current speed, i. e. the braking time is constant.



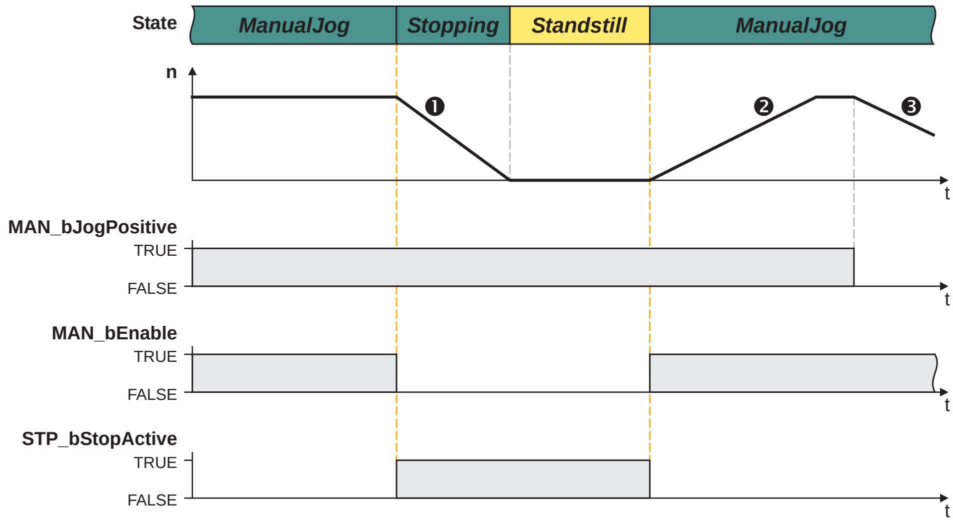
[11-8] S-shaped deceleration ramp through selection of a relative S-ramp time

- By entering an S-ramp time in C02611, the deceleration ramp can be set in an S-shaped manner for purposes of jerk limitation; the total time until standstill is then extended by the S-ramp time set. ▶ Setting the S-ramp time (📖 385)
- Braking time at motor reference speed or C02612 = "1":

  C02610 [s] + C02611 [s]

11.2.3 Behaviour of the function (example)

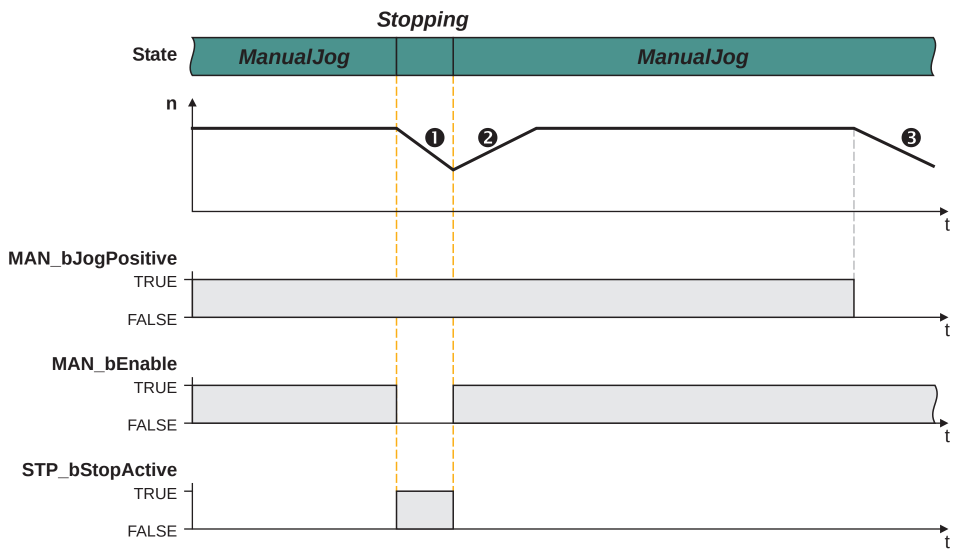
In the following example the enable of manual jog is deactivated during an active manual jog. Then the drive is braked to standstill within the deceleration time ❶ set for stop.



❶ C02610: Deceleration time for stop function
❷ C02622: Manual jog: Acceleration
❸ C02623: Manual jog: Deceleration

[11-9] Example: Stop with reaching standstill

If the basic function "Manual Jog" is reactivated within the deceleration time ❶, this basic function takes over the control of the drive from the current speed and the function state "Drive is stopped" is abandoned immediately:



❶ C02610: Deceleration time for stop function
❷ C02622: Manual jog: Acceleration
❸ C02623: Manual jog: Deceleration

[11-10] Example: Stop without reaching standstill

11.3 Quick stop

In contrast to Stop, the purpose of quick stop (QSP) is a stop in case of error. If quick stop is activated, the drive is brought to standstill within the deceleration time set irrespective of the setpoint that is preselected.

Note!

Through this, superimposed controls (e.g. synchronous or position control) may produce following errors. If several drives execute a coordinated motion, the quick stop function should only be used for the motion master (master drive) in order to maintain the coordination.

When the basic function is activated, a start acceleration is considered. ▶ Start acceleration/acceleration reduction when the basic function changes (📖 383)

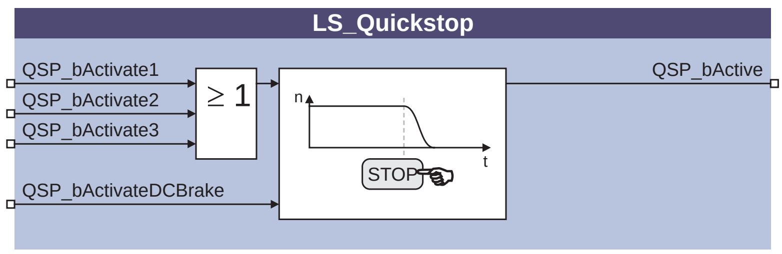
Tip!

Quick stop can also be set as error response for many monitoring functions ("quick stop by trouble"). Detailed information can be found in the chapter "Diagnostics & fault analysis". (📖 611)

The source from which the quick stop was activated is shown in a bit coded manner in C00159.

11.3.1 Internal interfaces | "LS_Quickstop" system block"

The **LS_Quickstop** system block provides the internal interfaces for the basic function "Stop" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Inputs

Identifier DIS code | data type	Information/possible settings	
QSP_bActivate1 C02619/1 | BOOL QSP_bActivate2 C02619/2 | BOOL QSP_bActivate3 C02619/3 | BOOL	Activate quick stop • The three inputs are linked via a logic OR gate.	
	TRUE	If one of the three inputs is set to TRUE, a change-over to the "Quick stop active" function state is effected and the drive is brought to standstill within the deceleration time set for quick stop.
	TRUE↘FALSE	If all three inputs are reset to FALSE, a change-over to a setpoint-generating basic function (e. g. "Speed follower") via the "Drive is stopped" function state is effected.
QSP_bActivateDCBrake C02619/5 | BOOL From V3.0	Activate DC-injection braking. (📖 395) • Only possible if V/f control or sensorless vector control is selected as motor control type in C00006! • This input has a higher priority than the three inputs *QSP_bActivate1 ... 3*.	
	TRUE	A change-over to the "Quick stop active" function state is effected and the drive is decelerated with the braking current set in C00974.
	TRUE↘FALSE	DC-injection braking is activated again. • If flying restart is activated in C00990, a flying restart process is automatically started to determine the current motor speed.

Outputs

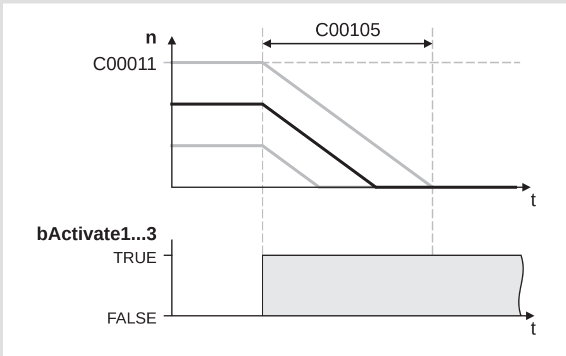
Identifier DIS code | data type	Value/meaning	
QSP_bActive C02619/4 | BOOL	Status signal "Quick stop by application active" • *QSP_bActive* is not set to TRUE if quick stop has been activated by another source, e. g. via device command or as an error response ("Quick stop by trouble").	
	TRUE	Quick stop has been requested via one of the three inputs *QSP_bActivate1 ... 3* and is active. - or - DC-injection braking has been requested via *QSP_bActivateDCBrake* and is active (only for motor control mode without encoder).

11.3.2 Parameter setting

- ▶ Parameterisation dialog in the »Engineer«: Tab **Application parameter** → Dialog level *Overview* → *All basic functions* → *Quick stop*
- ▶ Short overview of the parameters for quick stop:

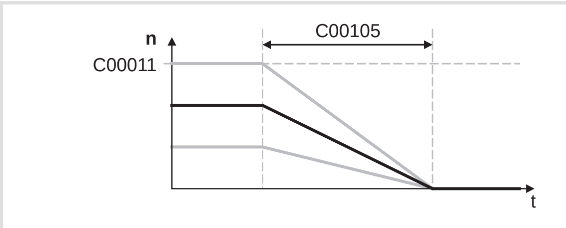
Parameter	Information
C00105	Quick stop decel. time
C00106	Quick stop S-ramp time
C00107	Reference for quick stop deceleration time

Parameter setting of quick stop



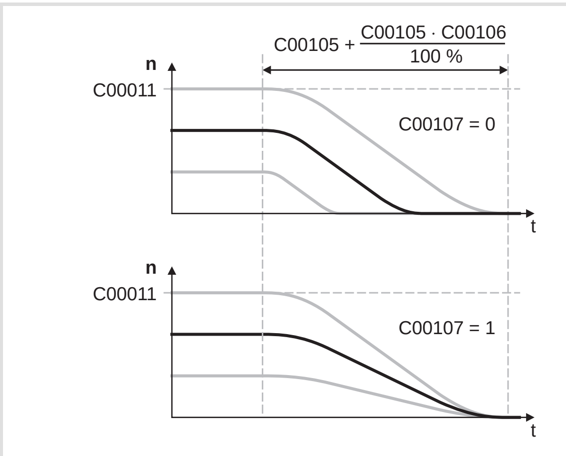
[11-11] Deceleration time referred to the motor reference speed

- The deceleration time for the quick stop function set in C00105 refers to a speed variation from the motor reference speed (C00011) to standstill.



[11-12] Deceleration time referred to the current speed

- When C00107 is set = "1", the deceleration time refers to the current speed.



[11-13] S-shaped deceleration ramp through selection of a relative S-ramp time

- By entering a relative S-ramp time in C00106, the deceleration ramp can be set in an S-shaped manner for purposes of jerk limitation; the total time until standstill is then extended by the S-ramp time set. ▸ Setting the S-ramp time (📖 385)
- Braking time at motor reference speed or C00107 = "1":

$$\text{C00105} + \frac{\text{C00105} \cdot \text{C00106 [\%]}}{100\ \%}$$

Tip!

After reaching standstill, the standstill position can be maintained while a torque is applied.

- For this purpose, select the phase controller gain in C00254 > "0".
- With C00254 > "0" the phase control is automatically activated after the standstill is reached.

11.3.3 Activating/deactivating quick stop

For activation/deactivation of quick stop by the application, the three inputs *QSP_bActivate1...3* are provided. (📖 391)

- The three control inputs are linked via a logic OR gate, i.e. in order to activate quick stop, only one of the three inputs must be set to TRUE. To deactivate quick stop, though, all three inputs must be reset to FALSE.
- The control inputs can be linked with terminals (digital inputs) and/or process data in the function block editor.

Note!

In the standard technology applications the control input *QSP_bActivate1* is linked with the digital input DI1 in the Lenze setting.

Further options for activating quick stop

- Via device command "Activate quick stop" (C00002 = "45"), e.g. via a corresponding SDO of a higher-level control, an HMI or the »Engineer«.
- Via the stop key at the keypad, unless the Lenze setting of C00469 (assignment of the key) has been changed.
- Through the response "quick stop by trouble" parameterised for monitoring.

11.3.4 DC-injection braking

This function extension is available from software version V3.0!

Note!

DC-injection braking is only possible if V/f control or sensorless vector control is selected as motor control type in C00006!

Activate DC-injection braking

To activate DC-injection braking through the application, the *bActivateDCBrake* control input must be set to TRUE.

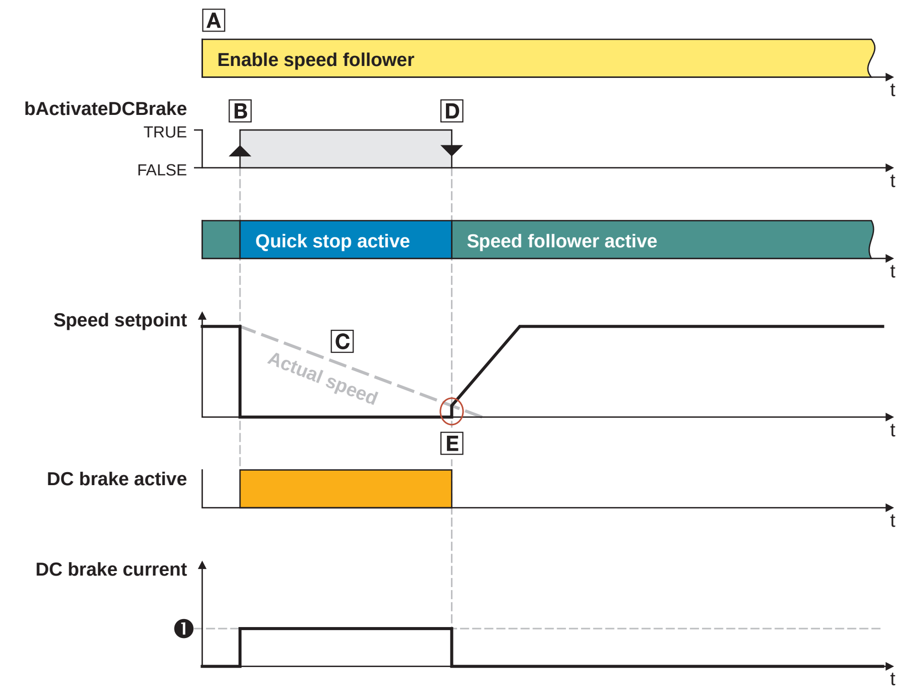
- A change-over to the "Quick stop active" function state is effected and DC-injection braking with the braking current set in C00974 is carried out.

Flying restart process after cancelling DC injection braking

If the flying restart mode is activated in C00990 and DC-injection braking is cancelled, a flying restart process is automatically started to determine the current motor speed if the following conditions are met:

- V/f control or sensorless vector control are selected as motor control in C00006.
- The position control structure is set to "Phase controller is active" in C02570.
- The *MI_bFlyingSyncBlocked* control input of the motor interface is not assigned or set to FALSE.
- The holding brake, if available, is not applied.

11.3.4.1 DC-injection braking and flying restart process



❶ C00974: Braking current

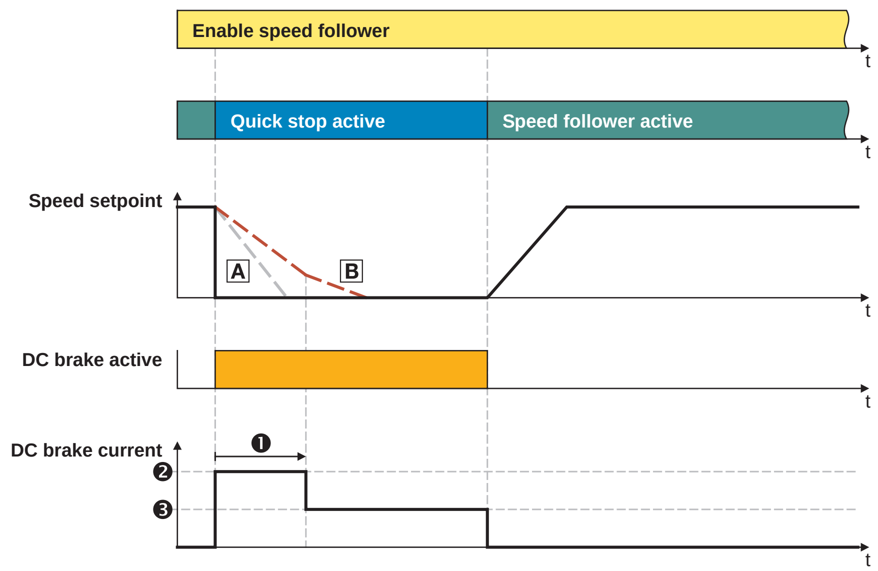
[11-14] Process example: speed follower is active ➔ QSP_bActivateDCBrake is active ➔ speed follower is active

A. Initial situation: basic function "Speed follower" is enabled and active.
B. *QSP_bActivateDCBrake* control input is set to TRUE through the application to activate DC-injection braking.
C. DC-injection braking is executed with the braking current set in C00974.
D. *QSP_bActivateDCBrake* control input is reset to FALSE through the application to deactivate DC-injection braking.
E. The flying restart process starts, i.e. the controller calculates the output frequency required for the momentary motor speed, then connects to the system, and accelerates the motor to the defined setpoint again.

11.3.4.2 DC-injection braking when quick stop is activated

If DC-injection braking is activated in C00976 instead of quick stop, DC-injection braking is executed automatically when quick stop is activated.

- After activating quick stop, a change-over to the "Quick stop active" function state is effected, and for the quick stop deceleration time set in C00105 a DC-injection braking process with the braking current set in C00975 is carried out.
- After this time has elapsed, a change-over to the braking current parameterised in C00974 is carried out and DC-injection braking is continued with this braking current.
- The DC-injection braking in this case is also carried out when the "Quick stop by trouble" error response is actuated; however, instead of the "Quick stop active" function state, the "Fault" function state is active, and the controller is in the "Quick stop by trouble active" device state.



A Actual speed value
B Actual speed value if the quick stop braking current is parameterised too low (C00975)
❶ C00105: Quick stop decel. time
❷ C00975: Quick stop braking current
❸ C00974: Braking current

[11-15] Process example: speed follower is active → quick stop activation → speed follower is active

Note!

The quick stop braking current in C00975 has to be set so that the drive can be decelerated from the maximum operating speed to standstill within the deceleration time for quick stop set in C00105!

11.4 Manual jog

The basic function "Manual jog" serves to traverse the drive manually, e.g. to clean or exchange the tool.

- Optionally a change-over to a second speed can be carried out during the traversing process.
- "Retracting" from activated (travel range) limit switches is also supported, then traversing can automatically only be carried out in the corresponding retracting direction.

Danger!

During manual jog operations, specially assigned profile parameters are effective. If these parameters are not set correctly, the drive may execute uncontrolled movements!

Stop!

In manual mode a travel range monitoring via limit switches and software limit positions is carried out via the basic function "Limiter". (513)

If no limit switches are connected and no software limit positions are set, and the reference is not known, the drive can travel into a mechanical barrier during manual mode and machine parts can be destroyed or damaged!

Note!

For manual jog setpoint speeds greater than 30000 rpm are not possible. The speeds defined for these basic function are internally limited to 30000 rpm.

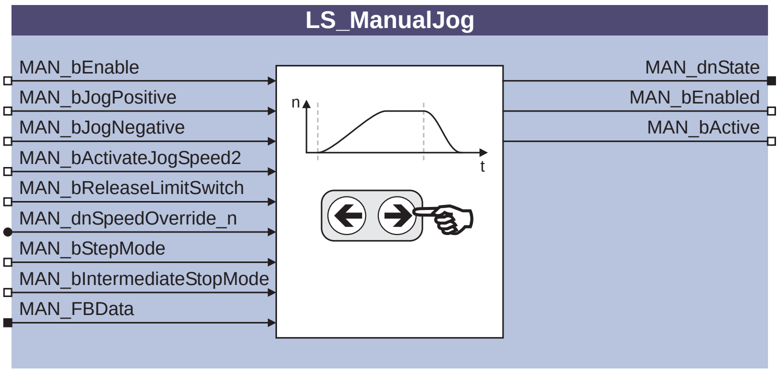
When the basic function is activated, a start acceleration is considered. ▶ Start acceleration/acceleration reduction when the basic function changes (383)

For the encoderless motor control types (from software version V3.0) **the following applies:**

If no position controller has been selected for position control in case of V/f control or sensorless vector control (C02570 = "1: Phase controller is active"), the manual jog is only executed via the speed profile resulting from the manual jog parameters.

11.4.1 Internal interfaces | "LS_ManualJog" system block"

The **LS_ManualJog** system block provides the internal interfaces for the basic function "Manual jog" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Inputs

Identifier DIS code | data type	Information/possible settings	
MAN_bEnable C02639/1 | BOOL	Requesting control via the basic function	
	TRUE	If no other basic function is active, a change-over to the "Manual jog active" function state is effected and manual jog can be carried out via the control inputs.
	TRUE↘FALSE	Active manual jog is stopped, i. e. a change-over from the active "Manual jog active" function state via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.
MAN_bJogPositive C02639/2 | BOOL	▸ Manual jog in positive/negative direction (📖 405)	
MAN_bJogNegative C02639/3 | BOOL		
MAN_bActivateJogSpeed2 C02639/4 | BOOL	Change-over to speed 2 for manual jog	
	FALSE	Speed 1 (C02620) active.
	TRUE	Speed 2 (C02621) active.
MAN_bReleaseLimitSwitch C02639/5 | BOOL	Retracting of an activated limit switch	
	TRUE	Retracting of the accordingly operated limit switch in the corresponding retracting direction until the limit switch is cleared again (no longer operated) and the drive is within the software limit positions again. Afterwards the drive is braked to standstill with the deceleration set unless the control input *MAN_bJogPositive* or *MAN_bJogNegative* is activated for the corresponding retracting direction.

Identifier DIS code | data type	Information/possible settings	
MAN_dnSpeedOverride_n C02637 | DINT From V5.0	Value of speed override • Percentage multiplier for the currently active speed (C02620 or C02621). • In the case of active manual jog, the speed override is always active and does not have to be activated separately. • Changes are accepted in each cycle. • 2^{30} ≡ 100 % of the speed parameterised in C02620 or C02621. • For values ≤ 1 % the status bit 19 is set. • Values ≤ 0 % are set to 0 % internally and lead to the standstill of the drive.	
MAN_bStepMode C02639/8 | BOOL From V5.0	▶ Manual jog with step limitation (📖 406) • Only possible if the "Manual jog with intermediate stop" mode is not active.	
	TRUE	Manual jog with step limitation active.
MAN_ bIntermediateStopMode C02639/9 | BOOL From V5.0	▶ Manual jog with intermediate stop (📖 407) • This mode has a higher priority than the "Manual jog with step limitation" mode.	
	TRUE	Manual jog with intermediate stop active.
MAN_FBData From V5.0	Interface for the transfer of the function block instance data for determining the positions for intermediate stop • Connect this input to the output *FBData* of the function block instance of type **L_PosPositionerTable** or **L_PosProfileTable**.	

Outputs

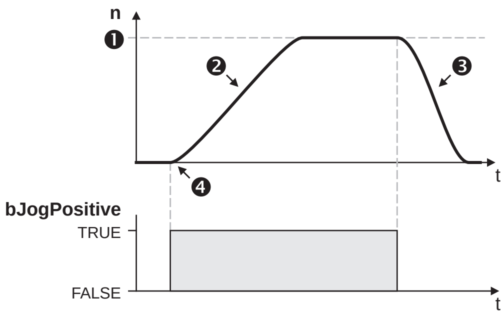
Identifier DIS code | data type	Value/meaning	
MAN_dnState C02638 | DINT	Status (bit coded) • When the basic function is not enabled, all bits are set to "0". • Bits which are not listed are not assigned with a status (always "0").	
	Bit 1	Manual jog is active.
	Bit 2	Manual jog is completed.
	Bit 3	Acceleration/deceleration phase is active.
	Bit 5	CCW rotation is active.
	Bit 15	Fault in basic function active (group signal).
	Bit 16	Stop by simultaneous selection of negative direction and retraction of limit switch.
	Bit 17	Stop by simultaneous selection of positive and negative direction.
	Bit 18	Stop by simultaneous selection of positive direction and retraction of limit switch.
	Bit 19	Speed override ≤1 % • This status is only available from software version V5.0.
	Bit 20	Speed 2 (C02621) active.
	Bit 21	Speed 1 (C02620) active.
	Bit 22	Stop by selection of positive direction and simultaneous activation of the positive software limit position or the positive limit switch.
	Bit 23	Stop by selection of negative direction and simultaneous activation of the negative software limit position or the negative limit switch.
	Bit 24	General abort process (ramp down of the speed setpoint) • Takes place e.g. when a manual direction initiator is released or due to an impermissible state (see bit 16, 17, 18, 22, 23).
	Bit 25	Stopping is active. • Basic function enabled for the first time but no manual jog has been requested/is active yet or current speed is higher than the manual jog speed.
	Bit 26	Home position is not known. • This status is only available from software version V5.0.
	Bit 27	No intermediate stop position available. • This status is only available from software version V5.0.
	Bit 30	Profile generation error.
MAN_bEnabled C02639/6 | BOOL	Status signal "Basic function is enabled"	
	TRUE	Manual jog via the control inputs is possible. • The *MAN_bEnable* enable input is set to TRUE and the controller is in the "Manual jog active" function state.
MAN_bActive C02639/7 | BOOL	Status signal "Basic function is active"	
	TRUE	Manual jog is active (the drive axis is moving).

11.4.2 Parameter setting

- Parameterisation dialog in the »Engineer«: Tab **Application parameters** ➞ Dialog level *Overview* ➞ *All basic functions* ➞ *Manual jog*
- Short overview of the parameters for manual jog:

Parameter	Information
C02620	Manual jog: Speed 1
C02621	Manual jog: Speed 2
C02622	Manual jog: Acceleration
C02623	Manual jog: Deceleration
C02624	Manual jog: S-ramp time
C02625	Manual jog: Step distance
C02626/1...16	Manual jog: Index stop position
C02627/1...16	Manual jog: Selected stop position
Highlighted in grey = display parameter	

11.4.2.1 Smooth start and quick stop of the drive



❶ C02620: Manual jog: Speed 1
❷ C02622: Manual jog: Acceleration
❸ C02623: Manual jog: Deceleration
❹ C02624: Manual jog: S-ramp time

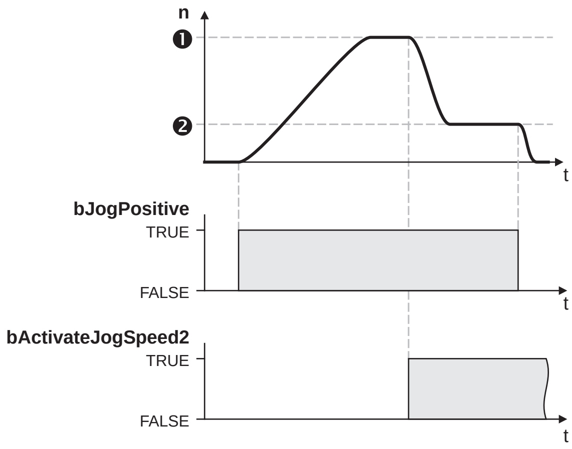
[11-16] Example: Smooth start and quick stop

- Different values for acceleration and deceleration can be set in C02622/C02623 in order to implement a smooth start and a quick stop of the drive.
- By entering a relative S-ramp time in C02624 both ramps can be set in S-shape for jerk reduction. ▶ Setting the S-ramp time (📖 385)

Tip!

A quick deceleration (C02623) reduces the time from letting go of the "Jog key" to the actual stop of the drive, so that the drive can thus be better positioned "by eye" and the desired stop position is not overtravelled.

11.4.2.2 Second speed



❶ C02620: Manual jog: Speed 1
❷ C02621: Manual jog: Speed 2

[11-17] Example: Change-over to second speed

- By setting the input *MAN_bActivateJogSpeed2* to TRUE, a change-over to a second speed (C02621) can be carried out during the traversing process.

11.4.3 Executing manual jog

Requirements

- The controller is in the "Operation" device state.
- The basic function "manual jog" is part of the active application.
- No other basic function is active.

Activation

To request the control via the basic function, the *MAN_bEnable* enable input in the application must be set to TRUE.

- If no other basic function is active, a change-over to the "Manual jog active" function state is effected and manual jog can be carried out via the control inputs.
- A successful change to the function state "Manual jog active" is displayed by a TRUE signal at the *MAN_bEnabled* status output.

Deactivation

If the *MAN_bEnable* enable input is reset to FALSE, an active manual jog is reset, i.e. the control inputs for manual jog are inhibited and the drive is braked to standstill within the deceleration time for Stop. (📖 387)

- The status output *MAN_bEnabled* is reset to FALSE and a change-over from the active "Manual jog active" function state via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.

11.4.3.1 Manual jog in positive/negative direction

In the "Manual jog active" function state the drive can be traversed manually according to the following truth table via the control inputs:

MAN_bJogNegative	MAN_bJogPositive	MAN_bActivateJogSpeed2	Function
FALSE	FALSE	-	Stop • The drive is braked to standstill with the deceleration set.
FALSE	TRUE	FALSE	Manual jog • In positive direction • Using speed 1 (C02620)
		TRUE	Manual jog • In positive direction • Using speed 2 (C02621)
TRUE	FALSE	FALSE	Manual jog • In negative direction • Using speed 1 (C02620)
		TRUE	Manual jog • In negative direction • Using speed 2 (C02621)
TRUE	TRUE	-	When both inputs are set to TRUE at the same time: • The drive is braked to standstill with the deceleration set. If not both inputs are set to TRUE at the same time: • The drive continues to traverse in the direction that was selected first.

Note!

In the standard technology applications "Actuating drive – speed" and "Actuating drive – torque", in the Lenze setting the control inputs are linked to the following digital inputs:

- **DI6**: Activate manual mode
- **DI7**: Manual jog in positive direction
- **DI8**: Manual jog in negative direction

11.4.3.2 Manual jog with step limitation

This function extension is available from software version V5.0 onwards!

This mode can be activated via the control input *MAN_bStepMode*.

In the "Manual jog with step limitation" mode the drive traverses by the "step distance" parameterised in C02625 if a direction is requested via the control inputs *MAN_bJogPositive/MAN_bJogNegative*. After traversing this distance, the drive stops.

- A new positive edge for the routing request causes a restart of the function or a reset of the distance counter, even if the drive is not at standstill yet.
- If the routing request is reset before the distance is reached, the drive stops immediately (with the deceleration set).

Note!

The two modes "Manual jog with step limitation" and "Manual jog with intermediate stop" cannot be active at the same time.

If there is a simultaneous request via the control inputs *MAN_bStepMode* and *MAN_bIntermediateStopMode*, only the "Manual jog with intermediate stop" mode is active. ▶ Manual jog with intermediate stop (🕮 407)

11.4.3.3 Manual jog with intermediate stop

This function extension is available from software version V5.0 onwards!

This mode can be activated via the control input *MAN_bIntermediateStopMode*.

In the "Manual jog with intermediate stop" mode, in the case of a routing request via the control inputs *MAN_bJogPositive/MAN_bJogNegative* the drive traverses to the defined "Intermediate stop position" that is next in the corresponding direction.

MAN_bJogNegative	MAN_bJogPositive	Function
FALSE	TRUE	Drive traverses from the current position to the next target in positive direction of the intermediate positions defined.
TRUE	FALSE	Drive traverses from the current position to the next target in negative direction of the intermediate positions defined.

- The drive stops on the intermediate position that is approached.
- If the routing request is reset before the intermediate position is reached, the drive stops immediately (with the deceleration set).
- After the drive has stopped on the intermediate position, it can only continue after a new positive edge for the routing request.
- If the drive is in the outmost intermediate position defined and a new routing request is effected, the drive stops.

Note!

Requirements for manual jog with intermediate stop:
- The home position is known (otherwise status bit 26 is set).
- At least one intermediate stop position is defined (otherwise status bit 27 is set).

Selection of the intermediate stop positions

The max. 16 intermediate stop positions are selected/defined via a function block instance of type **L_PosPositionerTable** or **L_PosProfileTable**.

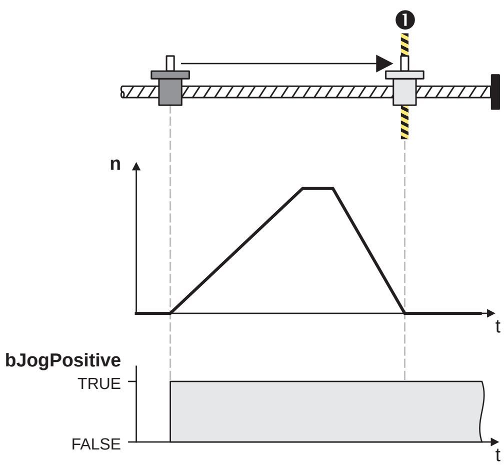
- For the transfer of the intermediate stop positions the output *FBData* of the respective function block instance is to be connected to the input *MAN_FBData* of the **LS_ManualJog** SB.
- The positions defined by the function block instance, which are to be used as intermediate stop positions are selected via C02626/1...16.
  - In connection with a function block instance of type **L_PosPositionerTable**: In C02626/x the index [1...75] of the table position in the VTPOS table has to be specified, which contains the intermediate stop position x that is to be used.
  - In connection with a function block instance of type **L_PosProfileTable**: In C02626/x the index [1...4] of the profile data set has to be specified, which contains the intermediate stop position x that is to be used.
- The positions selected are shown in C02627/1...16.

11.4.3.4 Manual jog to limit position

Note!

Detailed information on travel range monitoring via limit switches and software limit positions can be found in the description of the basic function "Limiter". (513)

Manual jog to software limit position

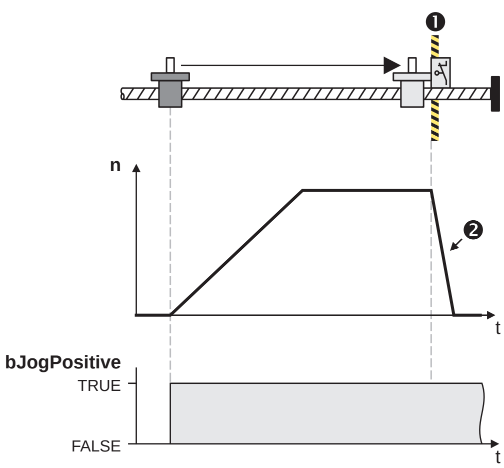


❶ Positive software limit position

[11-18] Example: Manual jog to positive software limit position

- If the reference is known and the software limit positions are set, a positioning to the corresponding software limit position is carried out, if manual jog was not exited manually before by resetting *MAN_bJogNegative* or *MAN_bJogPositive*.
- The drive brakes with the deceleration set (C02623) to the position of the corresponding software limit position.

Manual jog to hardware limit position (limit switch)



❶ Positive limit switch
❷ Deceleration ramp of the quick stop function

[11-19] Example: Manual jog to positive limit switch

- When a limit switch is approached during manual jog, the drive is braked to standstill within the deceleration time set for the quick stop function.

See also:

- Software limit positions (519)
- Hardware limit positions (limit switch) (522)

11.4.3.5 Retracting of an activated limit switch

By setting *MAN_bReleaseLimitSwitch* to TRUE, retracting from an operated limit switch is possible. The traversing process in the corresponding retracting direction is carried out until the limit switch is no longer activated.

- If, while retracting, a direction is selected additionally via the control inputs *MAN_bJogPositive* or *MAN_bJogNegative* towards the retracting direction, the travel is continued even after the limit switch has been abandoned until *MAN_bJogPositive* or *MAN_bJogNegative* are reset to FALSE.
- If the direction opposite to the retracting direction is preselected instead, the drive stops, and a corresponding status is shown via the status output *MAN_dnState*.

Note!

Retracting from a limit switch is only possible if it is still activated, i. e. if the corresponding limit switch input of the limiter is still activated. Therefore ensure that if a limit switch is approached, its trigger mechanics is not "overtravelled", for instance by a too great mass or drive, so that by this the limit switch is no longer activated.

Tip!

An activated limit switch can also be exited again by manual jog in the retracting direction via the control input *MAN_bJogPositive* or *MAN_bJogNegative*.

See also: Hardware limit positions (limit switch) (522)

11.5 Manual jog, encoderless

This function extension is available from software version V7.0!

In various applications where drives are operated with feedback in the production process, it may be required to execute a controlled operation without feedback of position values and speed and torque values derived from the position values as well.

- For initial commissioning and test of the system hardware.
  (Control cabinet function, connection system, test of trailing cables)
- In case of feedback failure in order to move the drive/machine to a service position afterwards.

The U and I rotation test modes implemented in the motor control of the drive controller make it possible for the user to move the drive in a controlled manner. In the case of an asynchronous motor, a load-dependent speed is adopted. A synchronous motor rotates with a predefined field frequency provided the motor can generate the necessary load torque.

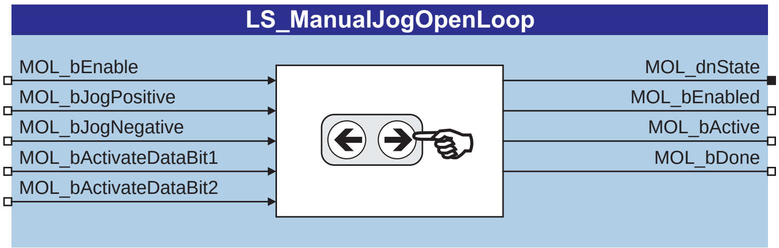
The basic function "Manual jog, encoderless" described in this chapter is available together with the corresponding Engineer dialogs for controlled manual jogging on the basis of the I rotation test mode.

Note!

The activation and control of the basic function "Manual jog encoderless" can either be executed via process data (system block inputs or code accesses (e.g. via keypad).

11.5.1 Internal interfaces | "LS_ManualJogOpenLoop" system block

The **LS_ManualJogOpenLoop** system block provides the internal interfaces for the basic function "Manual jog encoderless" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Inputs

Identifier DIS code | data type	Information/possible settings	
MOL_bEnable C02781/1 | BOOL	Requesting control via the basic function. • Request is also possible via C02770/1.	
	TRUE	If no other basic function is active, it is changed to the "Manual jog encoderless active" and the drive can be traversed in a controlled way via the control inputs.
	TRUE↘FALSE	Active controlled traversing is stopped, i.e. a change-over from the active "Manual jog encoderless active" function state back to the "Controller not ready" basic state is effected.
MOL_bJogPositive C02781/2 | BOOL	▸ Manual jog open loop in positive/negative direction (📖 418) • Control is also possible via C02770/2 and C02770/3.	
MOL_bJogNegative C02781/3 | BOOL		
MOL_bActivateDataBit1 C02781/4 | BOOL	▸ Selection of the profile parameter set (📖 418) • Selection is also possible via C02770/4 and C02770/5.	
MOL_bActivateDataBit2 C02781/5 | BOOL		

Outputs

<table>
<tr><th>Identifier
DIS code | data type</th><th colspan="2">Value/meaning</th></tr>
<tr><td rowspan="3">Mol_dnState
C02780 | DINT</td><td colspan="2">Status (bit coded)
• When the basic function is not enabled, all bits are set to "0".
• Bits which are not listed are not assigned with a status (always "0").</td></tr>
<tr><td>Bit 1</td><td>Manual jog open loop active.</td></tr>
<tr><td>Bit 2</td><td>Profile executed.</td></tr>
<tr><td rowspan="2">MOL_bEnabled
C02781/6 | BOOL</td><td colspan="2">Status signal "Basic function is enabled"</td></tr>
<tr><td>TRUE</td><td>Controlled traversing via the control inputs is possible.
• The MOL_bEnable enable input is set to TRUE and the controller is in the "Manual jog open loop active" function state.</td></tr>
<tr><td rowspan="2">MOL_bActive
C02781/7 | BOOL</td><td colspan="2">Status signal "Basic function is active"</td></tr>
<tr><td>TRUE</td><td>Controlled traversing is active (the drive axis moves according to the defined profile).</td></tr>
<tr><td rowspan="2">MOL_bDone
C02781/8 | BOOL</td><td colspan="2">Status signal "Max. activation time reached"
• The status signal indicates that the respective parameterised max. activation time has been reached.
• The counter of the max. activation time is reset every time there is a TRUE-FALSE edge at MOL_bJogPositive/MOL_bJogNegative.
• The setpoint frequency and therefore the rotating field at the motor terminals is generated within the acceleration time parameterised in C02774/1...4 for the selected profile and the drive movement starts accordingly.
• At the instant of "Max. activation time reached", the drive is still moving and ramping down within the deceleration time parameterised in C02775/1...4 for the selected profile is initiated.</td></tr>
<tr><td>TRUE</td><td>Controlled manual jogging is active
• The max. activation time parameterised for the selected profile in C02776/1...4 has expired.
• MOL_bJogPositive or MOL_bJogNegative is TRUE
• The setpoint frequency still corresponds to the setpoint frequency parameterised in C02771/1...4 for the selected profile.</td></tr>
</table>

11.5.2 Parameter setting

- Parameterisation dialog in the »Engineer«: Tab **Application parameters** → Dialog level *Overview* → *All basic functions* → *Manual jog open loop*
- Short overview of the parameters for manual jog open loop:

Parameter	Information	Lenze setting Value	Unit
C02770/1	EnableManualMode	0: Deactivate	
C02770/2	JogPositive	0: Deactivate	
C02770/3	JogNegative	0: Deactivate	
C02770/4	SelectTab1	0: Deactivate	
C02770/5	SelectTab2	0: Deactivate	
C02771/1...4	Frequency • Field frequency f_d with which the current vector rotates.	1.0	Hz
C02772/1...4	Starting angle	0.0	°
C02773/1...4	Current • R.m.s. value of the current vector which is injected with the parameterised frequency/starting angle. • 100 % ≡ I_{max_device} (C00022)	10.00	%
C02774/1...4	Acceleration time	1.000	s
C02775/1...4	Deceleration time	1.000	s
C02776/1...4	Duration (max. activation time)	1.000	s
C02779	MoI_SetpointCurrent	-	A
C02780	MoI_dnState	-	
C02781	ManualJogOpenLoop: Dig. signals	-	

Highlighted in grey = display parameter

Stop!

At the frequency f_d = 0 Hz, the r.m.s. value increases to 141 % of the current parameterised in C02773/x. As a result, the connected motor can be destroyed!

Remedy: Activation of a derating curve in i2xt monitoring or limitation of the parameterised current in C02773/x to 71 % of the rated motor current. ▶ Motor monitoring (I^2xt) (📖 224)

11.5.2.1 Necessary basic parameter settings

Before encoderless manual jogging is performed, the following basic parameter settings must be made:

1. Optimisation of the current controller in the case of non-Lenze motors or motors that are not included in the »Engineer« motor catalogue. ▶ Optimise current controller (📖 187)
2. Parameterisation of motor monitoring corresponding to the existing motor with the monitoring response "Fault". ▶ Motor monitoring (I^2xt) (📖 224)
   – This function is especially important for monitoring the permissible r.m.s. current load on motors at f_d = 0 Hz.
3. Activation of maximum current monitoring with the monitoring response "Fault" in the case of non-Lenze motors or motor that are not included in the »Engineer« motor catalogue. ▶ Maximum current monitoring (📖 235)
4. Activation of motor temperature monitoring via PTC and/or KTY. ▶ Motor temperature monitoring (📖 229)

11.5.2.2 Profile parameters

For controlled traversing, four different profiles (profile parameter sets) can be parameterised:

Setpoint	Profile parameter set 1	Profile parameter set 2	Profile parameter set 3	Profile parameter set 4
Frequency	C02771/1	C02771/2	C02771/3	C02771/4
Starting angle	C02772/1	C02772/2	C02772/3	C02772/4
Current	C02773/1	C02773/2	C02773/3	C02773/4
Acceleration time	C02774/1	C02774/2	C02774/3	C02774/4
Deceleration time	C02775/1	C02775/2	C02775/3	C02775/4
Duration time	C02776/1	C02776/2	C02776/3	C02776/4

The profile parameter set to be used is selected via the control inputs *MOL_bActivateDataBit1* and *MOL_bActivateDataBit2* or alternatively via the parameters C02770/4 and C02770/5.

11.5.3 Executing manual jog open loop

Requirements

- The basic parameter settings described in chapter [11.5.2.1] are made. ▶ Necessary basic parameter settings (📖 414)
- The controller inhibit is active.
- The controller has the "Controller not ready" function state.
- The basic function "Manual jog open loop" is part of the active application.
- No other basic function is active.

Activation

In order to request the control via the basic function, the *MOL_bEnable* enable input in the application must be set to TRUE or C02770/1 to "1".

- If no other basic function is active, it is changed to the "Manual jog open loop active" function state and a controlled traversing can take place via the control inputs or by writing C02770/2...5.
- A successful change to the function state "Manual jog open loop active" is displayed by a TRUE signal at the *MOL_bEnabled* status output.

Stop!

If a quick stop is requested with the "Manual jog, encoderless" function active, ramping-down of the setpoint frequency occurs with a fixed deceleration time t = 0 s.

- The thus generated braking effect at the drive is not comparable with the braking effect in the controlled motor mode!
- Under certain conditions, this can also lead to vibrations in the drive.

The user must select suitable setpoint frequencies to reduce the speeds of the machine to a minimum so that shutdown by means of the basic "Quick stop" function can be ensured in the case of encoderless manual jogging as well.

Deactivation

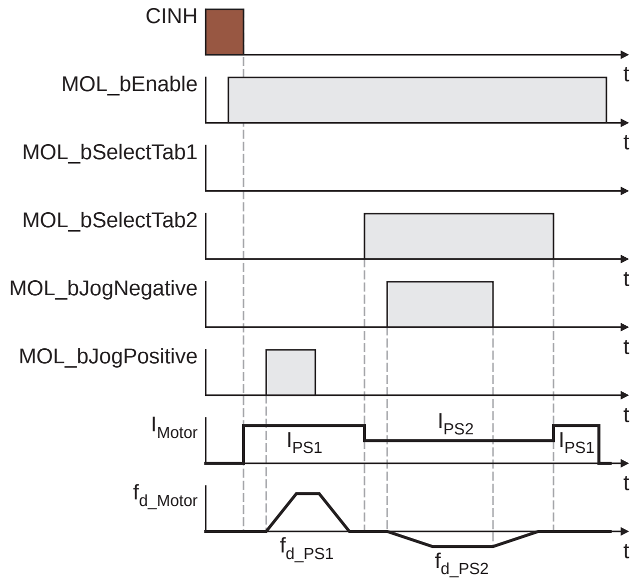
Note!

The controller must be inhibited before the basic function is deactivated!

Otherwise, the consequence is unsteady drive behaviour via the function state "Controller not ready" into the basic "Stop" function.

If the enable input *MOL_bEnable* is reset to FALSE, active manual jogging is stopped according to the following different cases that are distinguished.

Case 1: If *MOL_bEnable* is reset to FALSE and the drive controller is still enabled, the function state "Manual jog encoderless, active" is exited and the basic "Stop" function is activated. A changeover is then made back the basic state "Drive at a standstill" via the function state "Drive is being stopped".

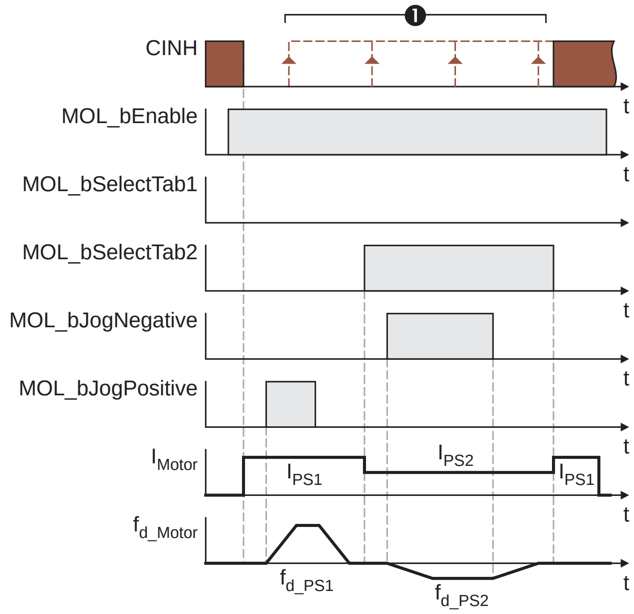


CINH = controller inhibit

[11-20] Signal characteristic, case 1: *MOL_bEnable* is reset to FALSE while the drive controller is still enabled

- In line with the motor control selected in C00006, additional basic conditions are necessary in order to avoid unsteady drive behaviour:
  - C00006 = "1: SC: Servo control sync. motor":
    Parameterised and connected motor encoder as well as a correct pole position in C00058/x
    - or -
    activation of the basic "Pole position identification" function and its monitoring in C02785 (Lenze recommendation).
  - C00006 = "2: SC: Servo control async. motor":
    Parameterised and connected motor encoder.

Case 2: If *MOL_bEnable* is reset to FALSE and the controller inhibit has been set at least at the same time, a changeover takes place from the function state "Manual jog encoderless active" to the function state "Drive controller not ready":



CINH = controller inhibit
❶ Setting of the controller inhibit not permissible here!

[11-21] Signal characteristic, case 2: *MOL_bEnable* is reset to FALSE and the controller inhibit is set at least at the same time

- ▶ With this signal sequence, drives can be operated for which the function of the motor encoder and/or the pole position do not exist.

11.5.3.1 Selection of the profile parameter set

For a controlled traversing, four different profiles can be parameterised. The profile parameter set to be used is selected via the control inputs given in the following truth table:

MOL_bActivateData Bit2 C02770/5	MOL_bActivateData Bit1 C02770/4	Selected profile parameter set
FALSE	FALSE	Profile parameter set 1
FALSE	TRUE	Profile parameter set 2
TRUE	FALSE	Profile parameter set 3
TRUE	TRUE	Profile parameter set 4

Note!

The four profile parameter sets may only be processed sequentially, i.e. a selected profile parameter set must be completed with f_d = 0 Hz before another profile parameter set can be activated.

- See signal characteristic (Fig. [11-20]) in the previous chapter. (🕮 416)
- Lenze recommends to control the basic function in accordance with this signal characteristic.

11.5.3.2 Manual jog open loop in positive/negative direction

In the "Manual jog open loop active" function state the drive can be traversed manually via the control inputs given in the following truth table:

MOL_bJogNegative C02770/3	MOL_bJogPositive C02770/2	Function
FALSE	FALSE	Stop • The drive is braked to standstill with the deceleration time in (C02775/x) set for the selected profile.
FALSE	TRUE	Controlled traversing in positive direction • The drive is led to the setpoint frequency with the acceleration time (C02774/x) set for the selected profile (C02771/x).
TRUE	FALSE	Controlled traversing in negative direction • The drive is led to the setpoint frequency with the acceleration time (C02774/x) set for the selected profile (C02771/x).
TRUE	TRUE	When both inputs are set to TRUE at the same time: • The drive is braked to standstill with the deceleration time in (C02775/x) set for the selected profile. If not both inputs are set to TRUE at the same time: • The drive continues to traverse in the direction that was selected first.
		x = number (1...4) of the selected profile parameter set

11.6 Homing

With the basic function "Homing" the measuring system of the machine is transmitted to the controller within the travel range that is physically possible.

- The reference (e.g. zero position of the drive axis in the machine measuring system) can be defined by reference search or reference setting.
- In case of reference search, the drive travels according to the defined homing mode to detect the reference in the measuring system independently.
  - In the reference point, the home position parameterised in C02642 is set as the current position. Afterwards, an absolute positioning to the target position parameterised in C02643 takes place (if C02641 = "0").
- If the reference is set in the homing mode "100: Set reference directly" or via the control input *HM_bLoadHomePos*, the drive can also be referenced manually if the motor is at standstill. The measuring system is set by means of the home position parameterised in C02642 or applied at the input *HM_dnHomePos_p*.

Danger!

During homing operations, specially assigned profile parameters are effective. If these parameters are not set correctly, the drive may execute uncontrolled movements!

Note!

Normally homing is only required once during commissioning of systems for which the machine cycle can be represented in the display area of the encoder, e. g. if multiturn absolute value encoders or singleturn absolute value encoders/ resolvers are used during the machine cycle on one motor revolution.

- The encoder position is stored safe against mains failure in the memory module and is therefore known to the drive control even after the mains has been switched. ▶ Behaviour of the home position after mains switching (📖 424)
- A renewed reference setting is only required in case of a renewed commissioning or in case of service (e.g. when drive components are exchanged).

Note!

For homing, setpoint speeds greater than 30000 rpm are not possible. The speeds defined for these basic function are internally limited to 30000 rpm.

When the basic function is activated, a start acceleration is considered. ▶ Start acceleration/acceleration reduction when the basic function changes (📖 383)

For the encoderless motor control types (from software version V3.0) **the following applies:**

The basic function "Homing" can only be activated for V/f control or sensorless vector control if the position controller has been selected for the position control (C02570 = "2: position controller active").

- The homing modes 14 & 15 are not permissible for the V/f control. If the selection is impermissible, the error message "Homing mode not allowed" is output.

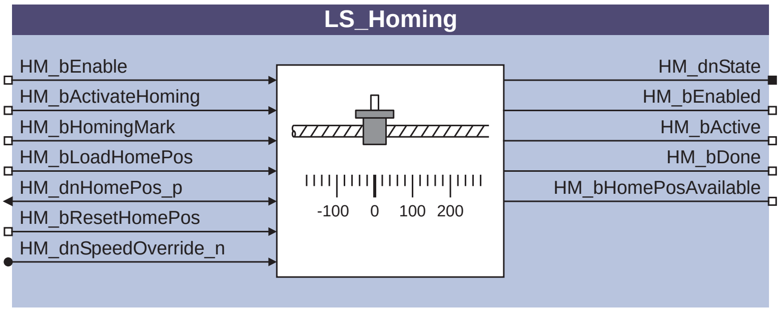
For the encoderless motor control types (from software version V5.0) **the following applies:**

If the V/f control or sensorless vector control is selected, the basic function "Homing" can be activated, irrespective of the use of the position controller.

- If no position controller has been selected for the position control in case of V/f control or sensorless vector control (C02570 = "1: Phase controller is active"), homing is only carried out via the speed profile resulting from the homing parameters. Because of this, the target positions set will only "roughly" be reached.

11.6.1 Internal interfaces | "LS_Homing" system block

The **LS_Homing** system block provides the internal interfaces for the basic function "Homing" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Inputs

Identifier DIS code | data type	Information/possible settings	
HM_bEnable C02659/1 | BOOL	Requesting control via the basic function.	
	TRUE	If no other basic function is active, a change-over to the "Homing active" function state is effected and homing can be carried out via the control inputs.
	TRUE↘FALSE	An active reference search is stopped, i. e. a change-over from the active "Homing active" function state via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.
HM_bActivateHoming C02659/2 | BOOL	Starting homing/directly setting reference	
	TRUE	Reference search is started in the homing mode selected (C02640). In the homing mode "100: Set reference directly" no reference search is started, but the home position set in C02642 is directly accepted.
	TRUE↘FALSE	Active reference search is completed/cancelled.
HM_bHomingMark C02659/3 | BOOL	Input for reference switch	
	TRUE	Reference switch is activated.
HM_bLoadHomePos C02659/4 | BOOL	Loading home position	
	FALSE↗TRUE	The position applied to input *HM_dnHomePos_p* is accepted as home position.
HM_dnHomePos_p C02658 | DINT	Home position in [increments] for acceptance with *HM_bLoadHomePos*	

Identifier DIS code | data type	Information/possible settings	
HM_bResetHomePos C02659/5 | BOOL	Resetting the status "Reference known"	
	FALSE↗TRUE	The internal status "Reference known" is reset. • The status outputs *HM_bDone* and *HM_bHomePosAvailable* are reset to FALSE.
HM_dnSpeedOverride_n C02655 | DINT From V5.0	Value of speed override • Percentage multiplier for the currently active speed (C02644 or C02646). • In the case of active homing, the speed override is always active and does not have to be activated separately. • Changes are accepted in each cycle. • $2^{30} \equiv 100$ % of the speed parameterised in C02644 or C02646. • For values ≤ 1 % the status bit 19 is set. • Values ≤ 0 % are set to 0 % internally and lead to the standstill of the drive.	

Outputs

Identifier DIS code | data type	Value/meaning	
HM_dnState C02657 | DINT	Status (bit coded) • When the basic function is not enabled, all bits are set to "0". • Bits which are not listed are not assigned with a status (always "0").	
	Bit 1	Reference search is active.
	Bit 2	Reference search is completed.
	Bit 3	Acceleration/deceleration phase is active.
	Bit 5	CCW rotation is active.
	Bit 7	Reference known.
	Bit 15	Fault in basic function active (group signal).
	Bit 16	Pre-switch-off (reference switch) has been detected.
	Bit 17	Touch probe/zero pulse has been detected.
	Bit 19	Speed override ≤1 % • This status is only available from software version V5.0.
	Bit 21	Profile data are limited by basic function "Limiter".
	Bit 22	Traversing direction is inhibited by basic function "Limiter".
	Bit 23	Abort by basic function "Limiter".
	Bit 25	Stopping is active. • Basic function is enabled for the first time but no referencing has been requested / is active yet or speed ≠ 0.
	Bit 30	Profile generation error.
HM_bEnabled C02659/6 | BOOL	Status signal "Basic function is enabled"	
	TRUE	Homing via the control inputs is possible. • The *HM_bEnable* enable input is set to TRUE and the controller is in the "Homing active" function state.
HM_bActive C02659/7 | BOOL	Status signal "Basic function is active"	
	TRUE	Reference search is active (the drive axis is moving).
HM_bDone C02659/8 | BOOL	Status signal "Basic function is ready"	
	TRUE	Reference search is completed. • Output is reset to FALSE when input *HM_bActivateHoming* is reset to FALSE.
HM_bHomePosAvailable C02659/9 | BOOL	Status signal "Reference is known"	
	TRUE	The drive knows the home position.

11.6.2 Parameter setting

- Parameterisation dialog in the »Engineer«: Tab **Application parameters** → Dialog level *Overview* → *All basic functions* → *Homing*
- Short overview of the parameters for homing:

Parameter	Information
C02528	Traversing range
C02640	Homing mode
C02641	Action after detect Home position
C02642	Home position
C02643	Homing: Target position
C02644	Homing: Speed 1
C02645	Homing: Acceleration 1
C02646	Homing: Speed 2
C02647	Homing: Acceleration 2
C02648	Homing: S-ramp time
C02649	Homing: Torque limit
C02650	Homing: Blocking time
C02651	Homing: TP configuration
C02652	Ref. pos. after mains switching
C02653	Max. rot. ang. aft. mns. swtch.
C02656	Actual position (homing)

11.6.2.1 Behaviour of the home position after mains switching

If the home position/information is also to be available after mains switching, the setting C02652 = "1: Received" is required.

Another condition for keeping the home position/information after mains switching is the compliance with the maximum permissible angle of rotation of the encoder.

- ▶ The maximally permissible angle of rotation can be set in C02653 in angular degree [°] with regard to the encoder shaft (360° ≡ one encoder shaft rotation).

Note!

Due to the internal numerical format and the resolution of one encoder revolution according to C00100, the position may not be reconstructed over the complete encoder range!

- The possible number of revolutions can be calculated as follows:
  Number of revolutions = $2^{(31\text{ - }C00100)}$
- Example: For a standard multiturn absolute value encoder with an absolute display area of 4096 revolutions (±2048), a maximum position resolution of 20 bits per revolution can be used!

When resolvers or single-turn absolute value encoders are used and the mains is switched off (24 V supply off), the encoder may only be moved by ½ revolution since otherwise the home position will get lost due to the ambiguity of the encoder information.

11.6.2.2 Homing mode

The zero position, also called reference, can be defined by a reference search or reference setting:

- In case of a reference search the drive travels according to a defined mode to detect the reference independently.
- In case of reference setting the reference is manually set when the drive has stopped.

Tip!

A reference search is mainly used in the case of continuously running systems, or if the traversing range or machine cycle of the drive cannot be represented in the display area of the encoder, e. g. if incremental encoders are used at the motor, or singleturn absolute value encoders or resolvers are used at the gearbox.

A reference is mainly set in systems/machines that bear the risk of collisions, or every time no homing can be executed (e.g. in case of a cross cutter having material in the machine).

- For reference setting, select the homing mode "100" in C02640.
- For a reference search the homing modes 0"..."15" are provided in C02640.
  - For process descriptions see the chapter "Overview of the Lenze homing modes". (📖 430)
  - From software version V3.0 according to the DS402 device profile, additionally the homing modes 1001"..."1035" are provided in C02640. Process descriptions for these homing modes can be found in the chapter "Overview of DS402 homing modes". (📖 430)

Note!

Profile data set change-over

For the reference search, two profile data sets with different speeds and accelerations can be parameterised. In this way, the homing time can be reduced and at the same time the accuracy can be increased.

▸ Profile data set change-over (📖 427)

The process descriptions in the following chapters provide information on the time the change-over to the profile data set 2 takes place in the corresponding homing mode.

▸ Overview of the Lenze homing modes (📖 430)
▸ Overview of DS402 homing modes (📖 445)

If the speed 2 (C02646) is set to "0" (Lenze setting), no change-over to the profile data set 2 takes place and the reference search and positioning to the target position is only executed with the profile parameters of the profile data set 1.

Note!

Drive behaviour after setting the reference

From software version V4.0 onwards, C02641 serves to parameterise the drive behaviour after setting the home position.

▶ Drive behaviour after setting the home position (📖 426)

In the Lenze setting (C02641 = "0"), the drive traverses to the absolute target position set in C02643 similarly to the behaviour known from the previous versions.

11.6.2.3 Home position & target position

When the home position is set during the reference search, the position detected in the machine measuring system corresponds to the value set in C02642.

For software versions lower than V4.0 the following applies:

- Afterwards the drive travels to the target position set in C02643.

The following applies from software version V4.0:

- The subsequent drive behaviour is determined by the mode parameterised in C02641. See the below chapter "Drive behaviour after setting the home position".

11.6.2.4 Drive behaviour after setting the home position

This function extension is available from software version V4.0!

C02641 can be used to select the drive behaviour after setting the home position.

Selection "0: Move absolute on Target position"

After setting the home position (C02642), the drive moves to the absolute target position set in C02643.

This selection is the Lenze setting and corresponds to the behaviour known from the previous versions.

Selection "1: Move relative by Target position"

After setting the home position (C02642), the drive moves relatively by the target position set in C02643.

Selection "2: Stop immediately"

After setting the home position (C02642), the drive stops immediately.

11.6.2.5 Profile data set change-over

For the reference search two profile data sets can be parameterised to reduce the homing time and increase the accuracy:

Profile data set 1		Profile data set 2	
C02644	Speed 1	C02646	Speed 2
C02645	Acceleration 1 (and deceleration 1)	C02647	Acceleration 2 (and deceleration 2)
C02648	S-ramp time (identical in both profile data sets) ▸ Setting the S-ramp time (📖 385)	C02648	S-ramp time (identical in both profile data sets) ▸ Setting the S-ramp time (📖 385)

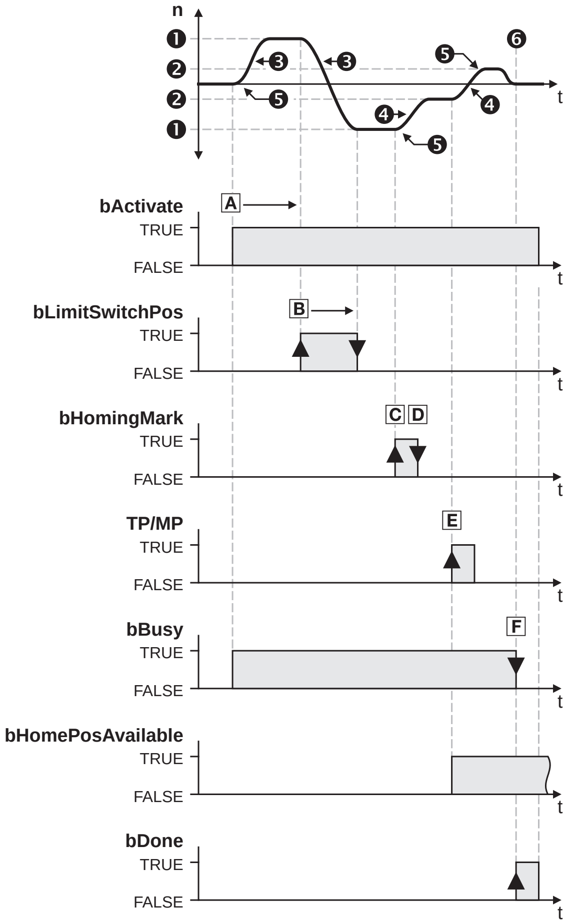
- With the profile data set 1 first the limit switch/reference switch (depending on the mode selected) is quickly approached.
- After reversing on the limit switch/reference switch, the slower – but more accurate – approach of the encoder zero pulse/touch probe sensor and the positioning to the target position (C02643) are effected with profile data set 2.

Note!

The change-over to profile data set 2 is only effected if speed 2 (C02646) is set > "0"!

In the Lenze setting (C02646 = "0") no change-over to the profile data set 2 takes place and the reference search and positioning to the target position is only executed with the profile parameters of the profile data set 1.

The process descriptions of the homing modes provide information about when the change-over to profile data set 2 takes place in the respective homing mode.
▸ Overview of the Lenze homing modes (📖 430)



❶ C02644: Speed 1
❷ C02646: Speed 2
❸ C02645: Acceleration 1 (and deceleration 1)
❹ C02647: Acceleration 2 (and deceleration 2)
❺ C02648: S-ramp time
❻ C02643: Target position

Example: Procedure of mode 2:

A. Movement in positive direction with profile data set 1.

B. Reversing to positive travel range limit switch.

C. Positive edge at *HM_bHomingMark* activates profile data set 2 for the further reference search.

D. Negative edge at *HM_bHomingMark* enables home position detection.

E. The following positive edge of the encoder zero pulse (MP) sets the reference.

F. Drive has reached defined target position.

11.6.2.6 Homing to end stop

By selecting the homing modes 14 & 15, homing to end stop can be executed as follows:

1. The drive travels with reduced torque in positive direction (mode 14) or negative direction (mode 15).
2. When the drive hits an end stop so that the torque limit set in C02649 is exceeded for the blocking time defined in C02650, the reference is set.
   – If a reference offset is set, traversing takes place around this offset in a correctly signed manner.

▸ Mode 14: positive direction to torque limit (📖 443)
▸ Mode 15: negative direction to torque limit (📖 443)

11.6.2.7 Connection of reference switch

For the homing modes with reference switch, the *HM_bHomingMark* control input must be connected to the digital input which is connected to the reference switch.

11.6.2.8 Touch probe interface configuration

The touch probe channel to be used for homing with touch probe detection in the »Engineer« is selected on the **Application parameters** tab in the dialog level *Overview* → *All basic functions* → *Homing* → *TP interface*.

- The setting carried out in this parameterisation dialog directly affects the setting of C02651 ("Homing: TP configuration") and vice versa.
- For directly setting C02651 (e. g. by means of the keypad) the corresponding decimal values are listed for all configuration options in the following table:

Selection Touch probe channel	Touch probe response		
	Positive edge	Negative edge	Both edges
Digital input 1	1	2	3
Digital input 2	4	8	12
Digital input 3	16	32	48
Digital input 4	64	128	192
Digital input 5	256	512	768
Digital input 6	1024	2048	3072
Digital input 7	4096	8192	12288
Digital input 8	16384	32768	49152
Motor encoder zero pulse	65536		
Position encoder zero pulse	262144		

- Example: For selecting the touch probe channel "Digital input 1" and a response only to a negative edge, the decimal value "2" has to be set in C02651.

11.6.3 Overview of the Lenze homing modes

In the following subchapters the procedures of homing modes 0 ... 15 are described, which can be selected in C02640.

Homing mode C02640	Evaluated signals/sensors			
	Touch probe sensor/ encoder zero pulse	Travel range limit switch		Reference switch at *HM_bHomingMark*
		Negative limit switch	Positive limit switch	
0	☑			☑
1	☑			☑
2	☑		☑	☑
3	☑	☑		☑
4	☑			☑
5	☑			☑
8	☑			
9	☑			
10	☑		☑	
11	☑	☑		
12			☑	
13		☑		
14	Positive direction of rotation to torque limit.			
15	Negative direction of rotation to torque limit.			
100	Set home position directly			

The switches/sensors are evaluated via the following internal interfaces:

Switch/sensor	Internal interface for digital input signal
Touch probe sensor	*DIGIN_bIn1 ... DIGIN_bIn8* • Alternatively the motor encoder or position encoder zero pulse can be evaluated. ▶ Touch probe interface configuration (📖 429)
Positive travel range limit switch	*LIM_bLimitSwitchPositive* (basic function "Limiter")
Negative travel range limit switch	*LIM_bLimitSwitchNegative* (basic function "Limiter")
Reference switch	*HM_bHomingMark* (basic function "Homing")

Note!

Profile data set change-over

For the reference search, two profile data sets with different speeds and accelerations can be parameterised. In this way, the homing time can be reduced and at the same time the accuracy can be increased.

▶ Profile data set change-over (📖 427)

The following process descriptions give information about the time the change-over to the profile data set 2 takes place in the corresponding homing mode.

If the speed 2 (C02646) is set to "0" (Lenze setting), no change-over to the profile data set 2 takes place and the reference search and positioning to the target position is only executed with the profile parameters of the profile data set 1.

Note!

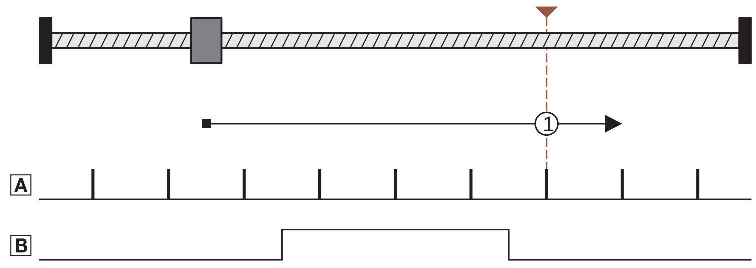
Drive behaviour after setting the reference

From software version V4.0 onwards, C02641 serves to parameterise the drive behaviour after setting the home position.

▶ Drive behaviour after setting the home position (📖 426)

In the Lenze setting (C02641 = "0"), the drive traverses to the absolute target position set in C02643 similarly to the behaviour known from the previous versions.

Mode 0: pos. direction - via home mark - to TP

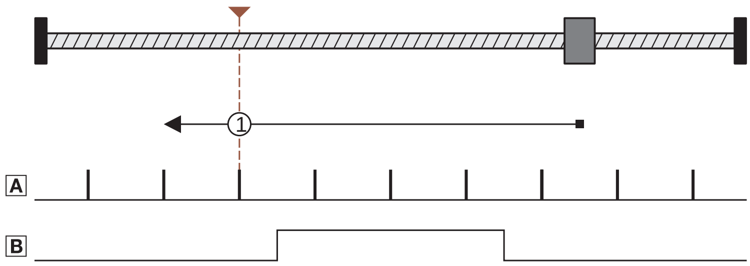


A Touch probe/zero pulse
B Reference switch

Procedure:

1. Movement in positive direction with profile data set 1.
2. Positive edge at *HM_bHomingMark* activates profile data set 2 for the further reference search.
3. Negative edge at *HM_bHomingMark* enables home position detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1: neg. direction - via home mark - to TP

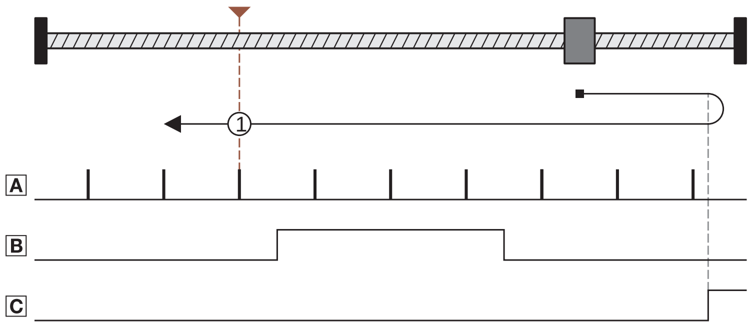


A Touch probe/zero pulse
B Reference switch

Procedure:

1. Movement in negative direction with profile data set 1.
2. Positive edge at *HM_bHomingMark* activates profile data set 2 for the further reference search.
3. Negative edge at *HM_bHomingMark* enables home position detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 2: pos. direction - reversing to limit switch - via home mark - to TP

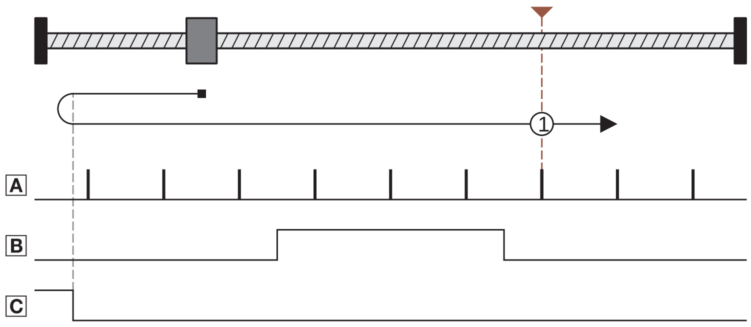


A Touch probe/zero pulse
B Reference switch
C Positive travel range limit switch

Procedure:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch.
3. Positive edge at *HM_bHomingMark* activates profile data set 2 for the further reference search.
4. Negative edge at *HM_bHomingMark* enables home position detection.
5. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 3: neg. direction - reversing to limit switch - via home mark - to TP

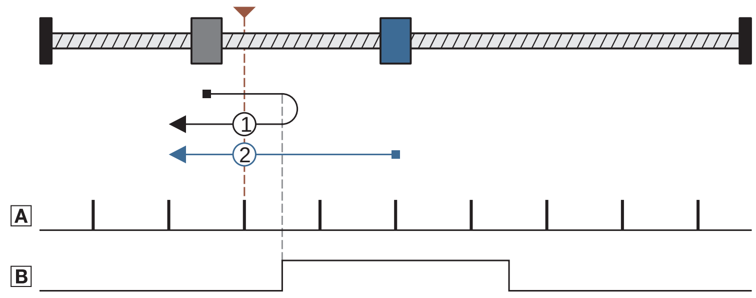


A Touch probe/zero pulse
B Reference switch
C Negative travel range limit switch

Procedure:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch.
3. Positive edge at *HM_bHomingMark* activates profile data set 2 for the further reference search.
4. Negative edge at *HM_bHomingMark* enables home position detection.
5. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 4: pos. direction - reversing to home mark - to TP



A Touch probe/zero pulse
B Reference switch

Procedures:

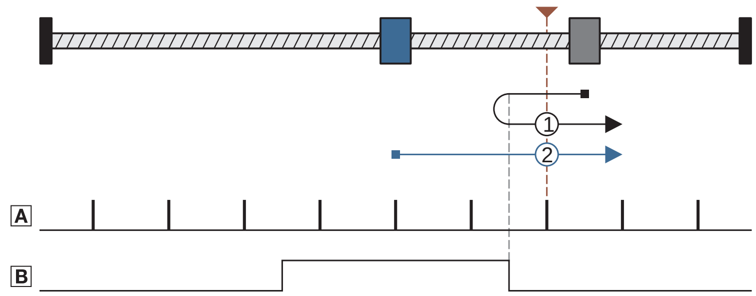
Case 1: Axis has not yet activated the reference switch:

1. Movement in positive direction with profile data set 1.
2. Reversing with positive edge at *HM_bHomingMark* and simultaneous activation of the profile data set 2 for further reference search.
3. Negative edge at *HM_bHomingMark* enables home position detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Negative edge at *HM_bHomingMark* enables home position detection.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 5: neg. direction - reversing to home mark - to TP



A Touch probe/zero pulse
B Reference switch

Procedures:

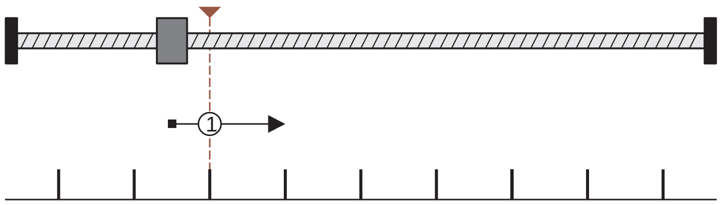
Case 1: Axis has not yet activated the reference switch:

1. Movement in negative direction with profile data set 1.
2. Reversing with positive edge at *HM_bHomingMark* and simultaneous activation of the profile data set 2 for further reference search.
3. Negative edge at *HM_bHomingMark* enables home position detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Negative edge at *HM_bHomingMark* enables home position detection.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 8: positive direction to touch probe

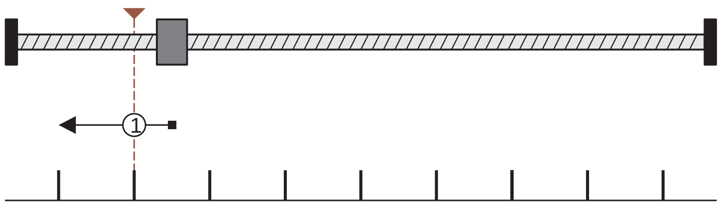


A Touch probe/zero pulse

Procedure:

1. Movement in positive direction with profile data set 1.
2. The following positive edge of the touch probe sensor sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 9: negative direction to touch probe

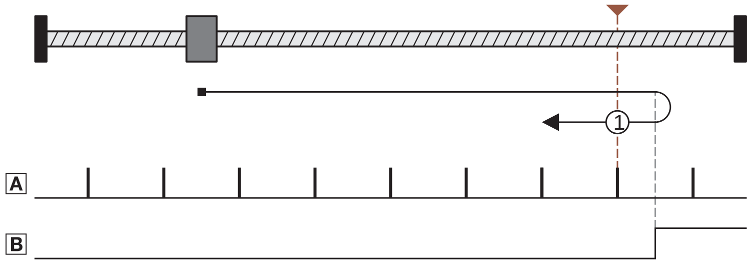


A Touch probe/zero pulse

Procedure:

1. Movement in negative direction with profile data set 1.
2. The following positive edge of the touch probe sensor sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 10: pos. direction - reversing to limit switch - to TP



A Touch probe/zero pulse
B Positive travel range limit switch

Procedure:

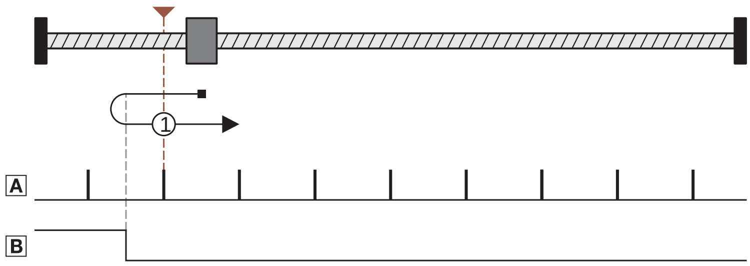
1. Movement in positive direction with profile data set 1.
2. Reversing with positive edge of the positive travel range limit switch and simultaneous activation of the profile data set 2 for further reference search.
3. The following positive edge of the touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Note!

The touch probe detection is already activated after reversing to the travel range limit switch, i.e. the home position may be set to the travel range limit switch.

- We therefore recommend to set a target position (C02643) unequal to the home position (C02642), in order to reenable the activated limit switch. Otherwise, the positioning process to the target position may be aborted by the basic function "Limiter" (see status signal *HM_dnState*).
- If the touch probe detection is to be activated after the travel range limit switch has been left, we recommend to use the DS402 homing methods 1 or 2. ▶ Overview of DS402 homing modes (📖 445)

Mode 11: neg. direction - reversing to limit switch - to TP



A Touch probe/zero pulse
B Negative travel range limit switch

Procedure:

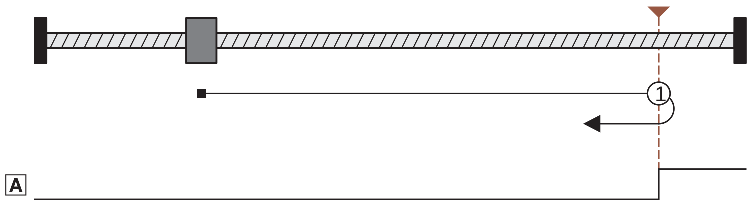
1. Movement in negative direction with profile data set 1.
2. Reversing with positive edge of the negative travel range limit switch and simultaneous activation of the profile data set 2 for further reference search.
3. The following positive edge of the touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Note!

The touch probe detection is already activated after reversing to the travel range limit switch, i.e. the home position may be set to the travel range limit switch.

- We therefore recommend to set a target position (C02643) unequal to the home position (C02642), in order to reenable the activated limit switch. Otherwise, the positioning process to the target position may be aborted by the basic function "Limiter" (see status signal *HM_dnState*).
- If the touch probe detection is to be activated after the travel range limit switch has been left, we recommend to use the DS402 homing methods 1 or 2. ▶ Overview of DS402 homing modes (📖 445)

Mode 12: positive direction to limit switch



A Positive travel range limit switch

Procedure:

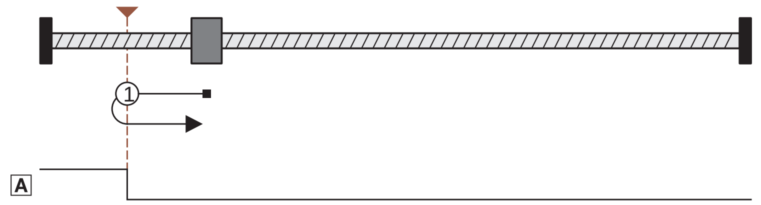
1. Movement in positive direction with profile data set 1.
2. Positive edge of the travel range limit switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Note!

The load mechanics is able to leave the travel range limit switch. It is then travelled back to the home position which has been set with the positive edge of the travel range limit switch.

- The mechanics may possibly stop on an activated limit switch.
- Therefore it is recommended to set a target position (C02643) that is unequal to the home position (C02642) to release the activated limit switch again.

Mode 13: negative direction to limit switch



A Negative travel range limit switch

Procedure:

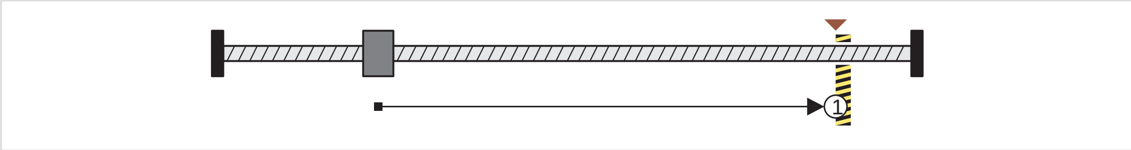
1. Movement in negative direction with profile data set 1.
2. Positive edge of the travel range limit switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Note!

The load mechanics is able to leave the travel range limit switch. It is then travelled back to the home position which has been set with the positive edge of the travel range limit switch.

- The mechanics may possibly stop on an activated limit switch.
- Therefore it is recommended to set a target position (C02643) that is unequal to the home position (C02642) to release the activated limit switch again.

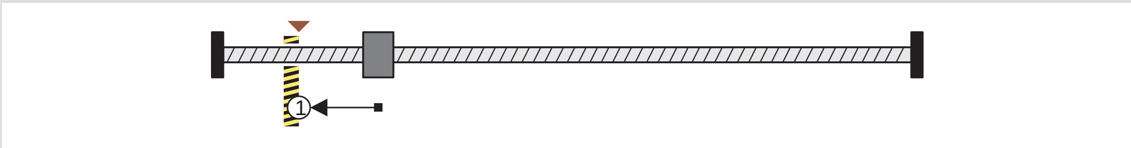
Mode 14: positive direction to torque limit



Procedure:

1. Movement in positive direction with reduced torque and profile data set 1.
2. The reference is set if the two following conditions for the time set in C02650 are fulfilled at the same time:
   – The current speed is lower than the threshold for standstill detection set in C00019.
   – Current torque is greater than the torque limit set in C02649 ("Homing to end stop").
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

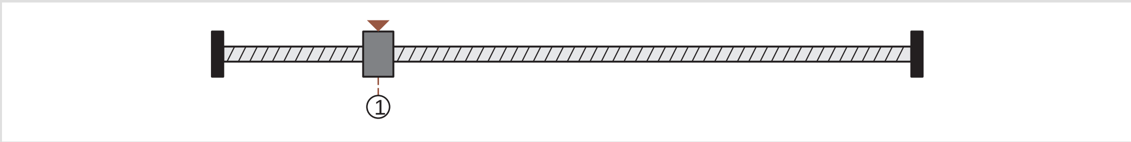
Mode 15: negative direction to torque limit



Procedure:

1. Movement in negative direction with reduced torque and profile data set 1.
2. The reference is set if the two following conditions for the time set in C02650 are fulfilled at the same time:
   – The current speed is lower than the threshold for standstill detection set in C00019.
   – Current torque is greater than the torque limit set in C02649 ("Homing to end stop").
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 100: Set reference directly



During the drive is at standstill, the measuring system is set by means of the home position parameterised in C02642.

11.6.4 Overview of DS402 homing modes

This function extension is available from software version V3.0!

In addition to the homing modes described in the previous subchapter "Overview of the Lenze homing modes", from software version V3.0 also the homing modes described in the following can be selected for a homing in C02640, according to the DS402 device profile.

DS402 homing method	Evaluated signals/sensors			
	Touch probe sensor/ encoder zero pulse	Travel range limit switch		Reference switch at *HM_bHomingMark*
		Negative limit switch	Positive limit switch	
01	☑	☑		
02	☑		☑	
03	☑			☑
04	☑			☑
05	☑			☑
06	☑			☑
07	☑		☑	☑
08	☑		☑	☑
09	☑		☑	☑
10	☑		☑	☑
11	☑	☑		☑
12	☑	☑		☑
13	☑	☑		☑
14	☑	☑		☑
15	Reserved: no homing is executed.			
16	Reserved: no homing is executed.			
17		☑		
18			☑	
19				☑
20				☑
21				☑
22				☑
23			☑	☑
24			☑	☑
25			☑	☑
26			☑	☑
27		☑		☑
28		☑		☑
29		☑		☑
30		☑		☑
31	Reserved: no homing is executed.			
32	Reserved: no homing is executed.			
33	☑			
34	☑			
35	Direct reference setting.			

The switches/sensors are evaluated via the following internal interfaces:

Switch/sensor	Internal interface for digital input signal
Touch probe sensor	*DIGIN_bIn1 ... DIGIN_bIn8* • Alternatively the motor encoder or position encoder zero pulse can be evaluated. ▶ Touch probe interface configuration (📖 429)
Positive travel range limit switch	*LIM_bLimitSwitchPositive* (basic function "Limiter")
Negative travel range limit switch	*LIM_bLimitSwitchNegative* (basic function "Limiter")
Reference switch	*HM_bHomingMark* (basic function "Homing")

Note!

Profile data set change-over

For the reference search, two profile data sets with different speeds and accelerations can be parameterised. In this way, the homing time can be reduced and at the same time the accuracy can be increased.

▶ Profile data set change-over (📖 427)

The following process descriptions give information about the time the change-over to the profile data set 2 takes place in the corresponding homing mode.

If the speed 2 (C02646) is set to "0" (Lenze setting), no change-over to the profile data set 2 takes place and the reference search and positioning to the target position is only executed with the profile parameters of the profile data set 1.

Note!

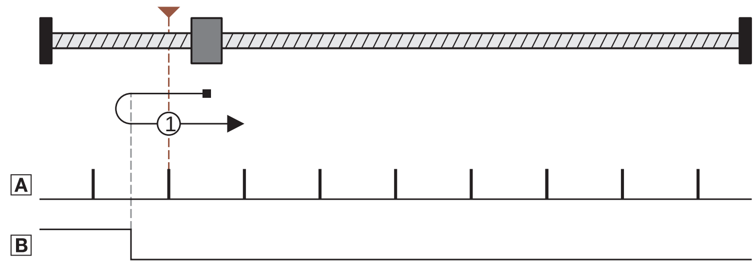
Drive behaviour after setting the reference

From software version V4.0 onwards, C02641 serves to parameterise the drive behaviour after setting the home position.

▶ Drive behaviour after setting the home position (📖 426)

In the Lenze setting (C02641 = "0"), the drive traverses to the absolute target position set in C02643 similarly to the behaviour known from the previous versions.

Mode 1001: DS402 homing method 01

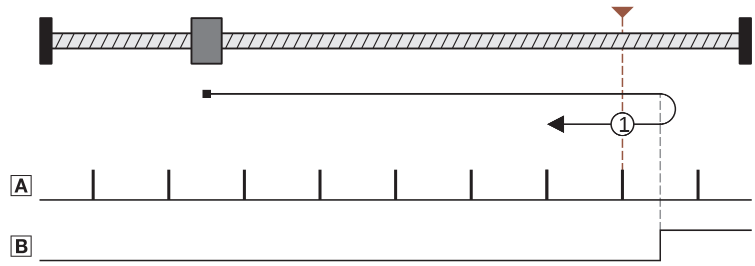


A Touch probe/zero pulse
B Negative travel range limit switch

Procedure:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch and change to profile data set 2.
3. Negative edge of the travel range limit switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1002: DS402 homing method 02

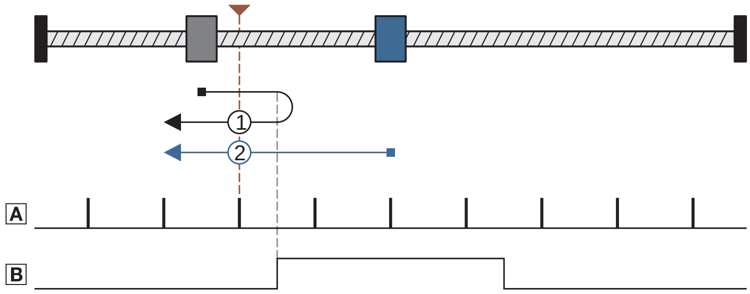


A Touch probe/zero pulse
B Positive travel range limit switch

Procedure:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch and change to profile data set 2.
3. Negative edge of the travel range limit switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1003: DS402 homing method 03



A Touch probe/zero pulse
B Reference switch

Procedures:

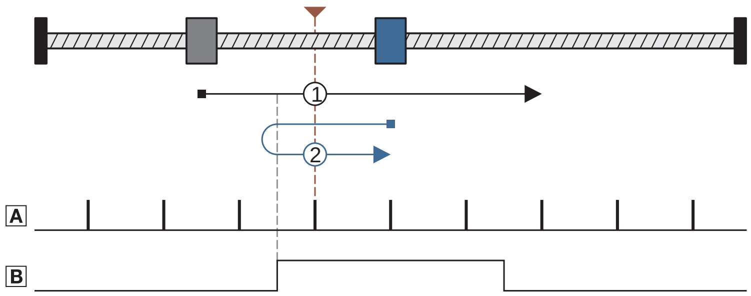
Case 1: Axis has not yet activated the reference switch:

1. Movement in positive direction with profile data set 1.
2. Reversing with positive edge of the reference switch and change to profile data set 2.
3. Negative edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Negative edge of the reference switch activates touch probe detection.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1004: DS402 homing method 04



A Touch probe/zero pulse
B Reference switch

Procedures:

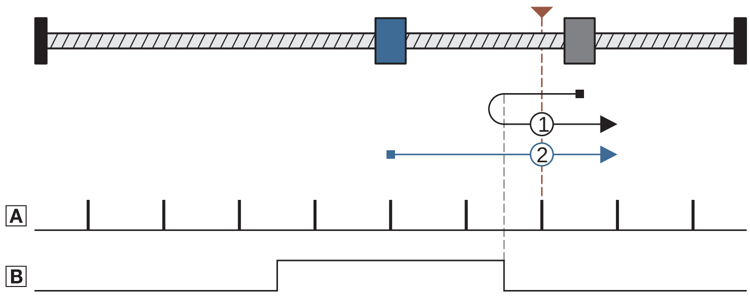
Case 1: Axis has not yet activated the reference switch:

1. Movement in positive direction with profile data set 1.
2. Positive edge of the reference switch activates touch probe detection and change to profile data set 2.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. Positive edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1005: DS402 homing method 05



A Touch probe/zero pulse
B Reference switch

Procedures:

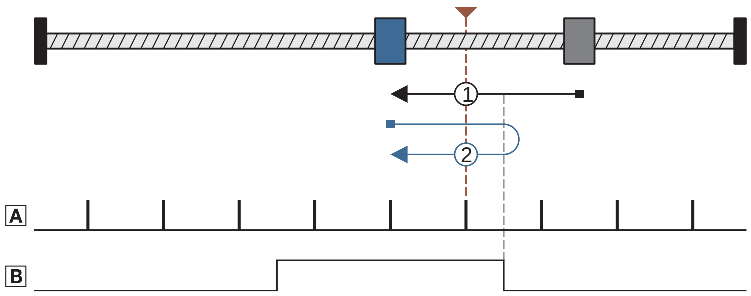
Case 1: Axis has not yet activated the reference switch:

1. Movement in negative direction with profile data set 1.
2. Reversing with positive edge of the reference switch and change to profile data set 2.
3. Negative edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Negative edge of the reference switch activates touch probe detection.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1006: DS402 homing method 06



A Touch probe/zero pulse
B Reference switch

Procedures:

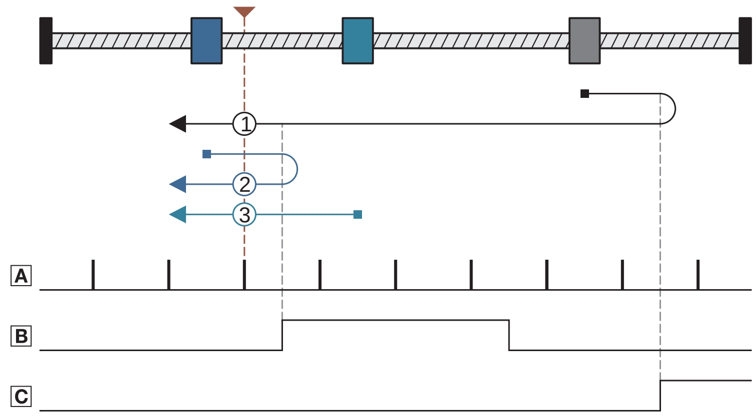
Case 1: Axis has not yet activated the reference switch:

1. Movement in negative direction with profile data set 1.
2. Positive edge of the reference switch activates touch probe detection and change to profile data set 2.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. Positive edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1007: DS402 homing method 07



A Touch probe/zero pulse
B Reference switch
C Positive travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch.
3. Positive edge of the reference switch activates profile data set 2.
4. Negative edge of the reference switch activates touch probe detection.
5. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

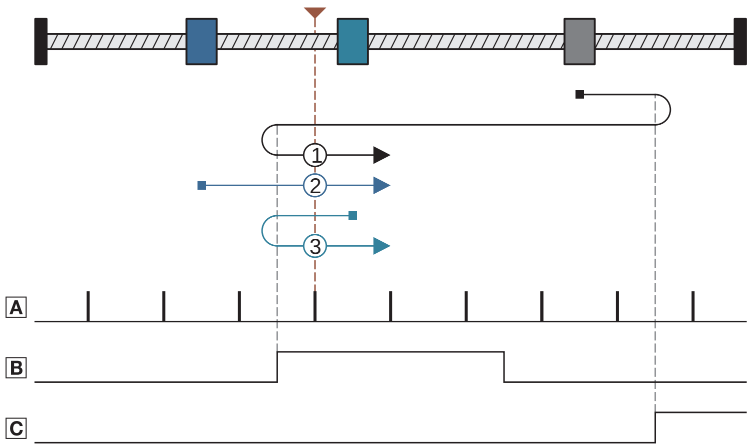
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing with positive edge of the reference switch and change to profile data set 2.
3. Negative edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Negative edge of the reference switch activates touch probe detection.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1008: DS402 homing method 08



A Touch probe/zero pulse
B Reference switch
C Positive travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch.
3. Positive edge of the reference switch activates profile data set 2.
4. Reversing with negative edge of the reference switch.
5. Positive edge of the reference switch activates touch probe detection.
6. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
7. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

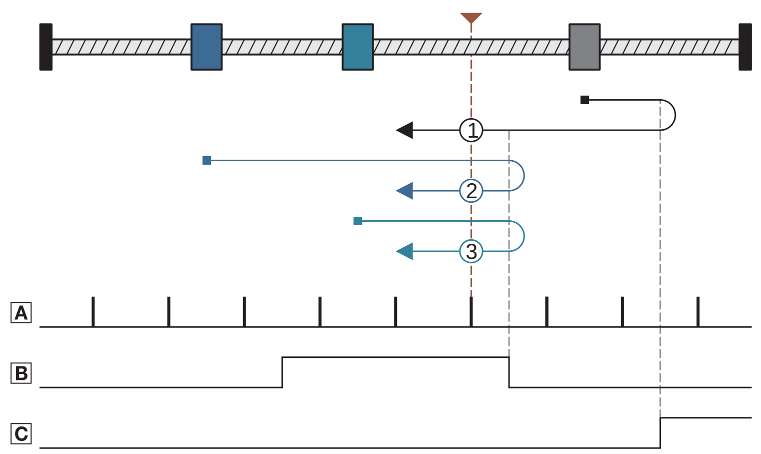
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Positive edge of the reference switch activates touch probe detection and change to profile data set 2.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. Positive edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1009: DS402 homing method 09



A Touch probe/zero pulse
B Reference switch
C Positive travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch.
3. Positive edge of the reference switch activates touch probe detection and change to profile data set 2.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

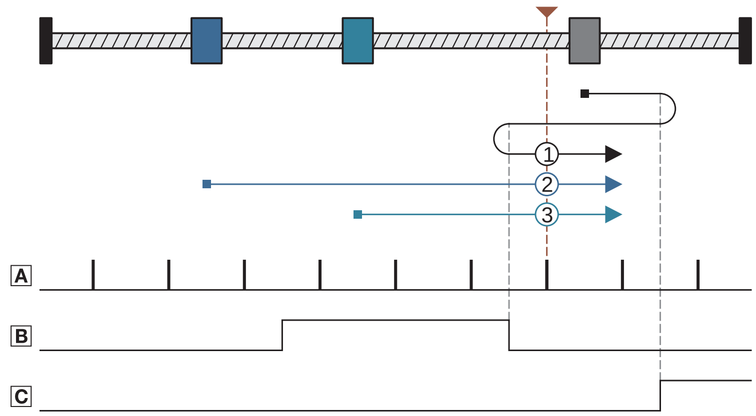
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Positive edge of the reference switch activates profile data set 2.
3. Reversing with negative edge of the reference switch.
4. Positive edge of the reference switch activates touch probe detection.
5. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. Positive edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1010: DS402 homing method 10



A Touch probe/zero pulse
B Reference switch
C Positive travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch.
3. Reversing with positive edge of the reference switch and change to profile data set 2.
4. Negative edge of the reference switch activates touch probe detection.
5. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

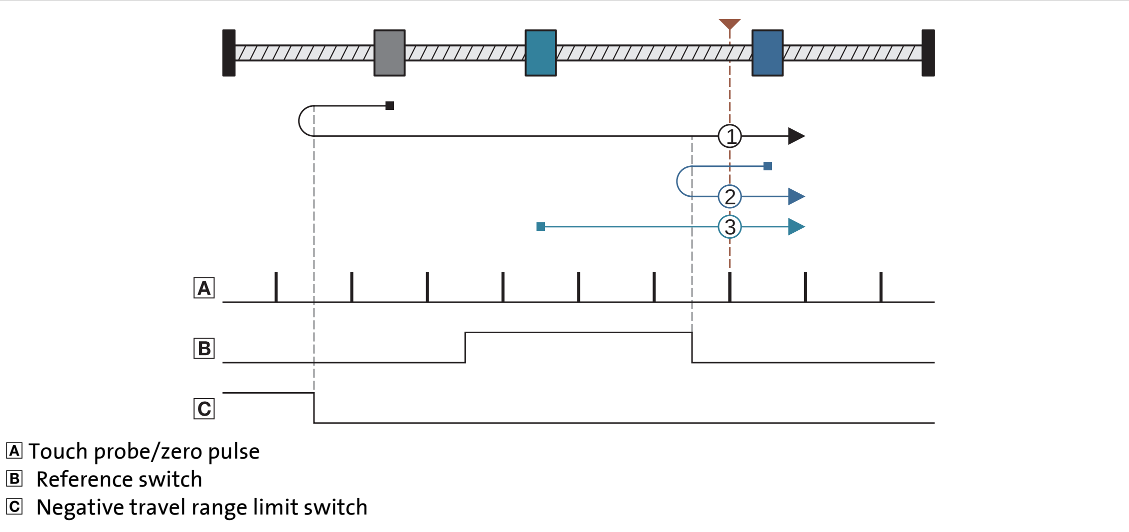
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Negative edge of the reference switch activates touch probe detection.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Negative edge of the reference switch activates touch probe detection.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1011: DS402 homing method 11



Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch.
3. Positive edge of the reference switch activates profile data set 2.
4. Negative edge of the reference switch activates touch probe detection.
5. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

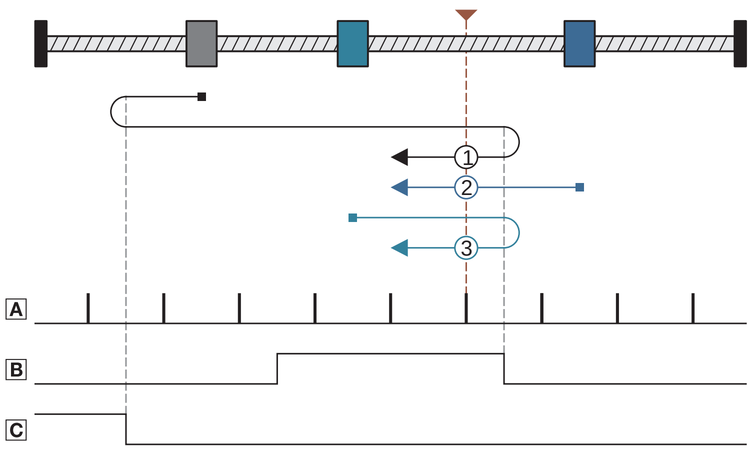
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing with positive edge of the reference switch and change to profile data set 2.
3. Negative edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Negative edge of the reference switch activates touch probe detection.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1012: DS402 homing method 12



A Touch probe/zero pulse
B Reference switch
C Negative travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch.
3. Positive edge of the reference switch activates profile data set 2.
4. Reversing with negative edge of the reference switch.
5. Positive edge of the reference switch activates touch probe detection.
6. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
7. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

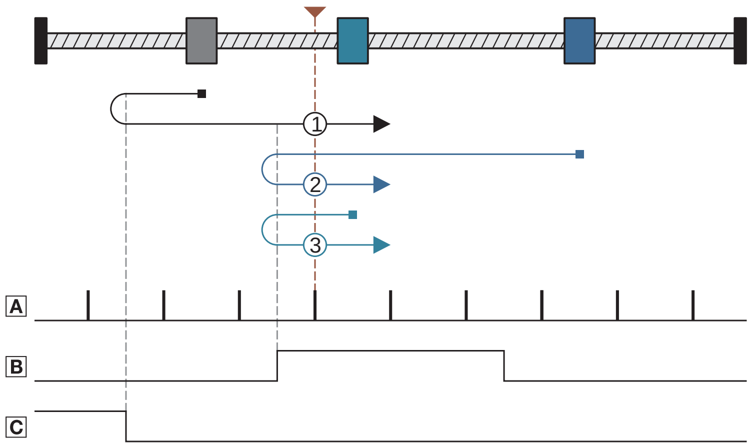
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Positive edge of the reference switch activates touch probe detection and change to profile data set 2.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. Positive edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1013: DS402 homing method 13



A Touch probe/zero pulse
B Reference switch
C Negative travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch.
3. Positive edge of the reference switch activates touch probe detection and change to profile data set 2.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

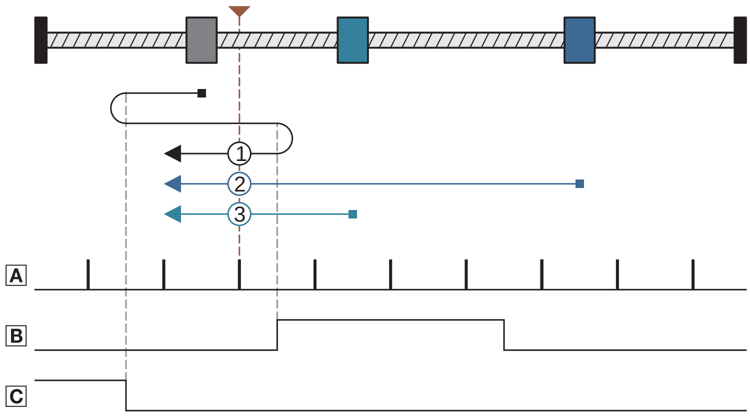
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Positive edge of the reference switch activates profile data set 2.
3. Reversing with negative edge of the reference switch.
4. Positive edge of the reference switch activates touch probe detection.
5. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. Positive edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1014: DS402 homing method 14



A Touch probe/zero pulse
B Reference switch
C Negative travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch.
3. Reversing with positive edge of the reference switch and change to profile data set 2.
4. Negative edge of the reference switch activates touch probe detection.
5. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Positive edge of the reference switch activates profile data set 2.
3. Negative edge of the reference switch activates touch probe detection.
4. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Negative edge of the reference switch activates touch probe detection.
3. The following positive edge of the encoder zero pulse/touch probe sensor sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

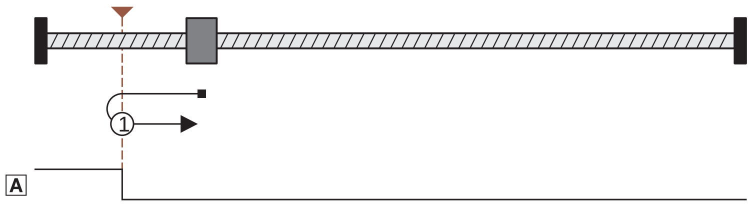
Mode 1015: DS402 homing method 15

Reserved: no homing is executed.

Mode 1016: DS402 homing method 16

Reserved: no homing is executed.

Mode 1017: DS402 homing method 17

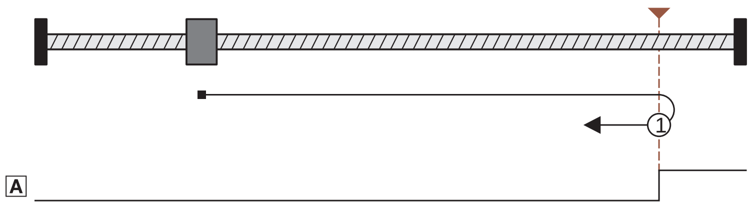


A Negative travel range limit switch

Procedure:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch and change to profile data set 2.
3. The following negative edge of the travel range limit switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1018: DS402 homing method 18

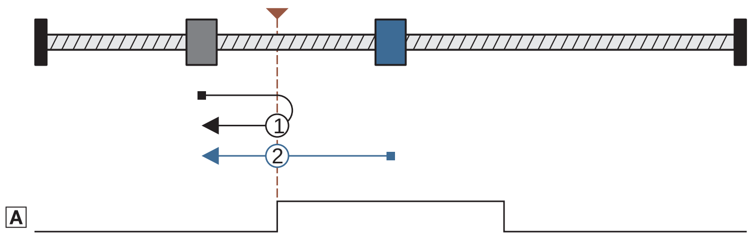


A Positive travel range limit switch

Procedure:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch and change to profile data set 2.
3. The following negative edge of the travel range limit switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1019: DS402 homing method 19



A Reference switch

Procedures:

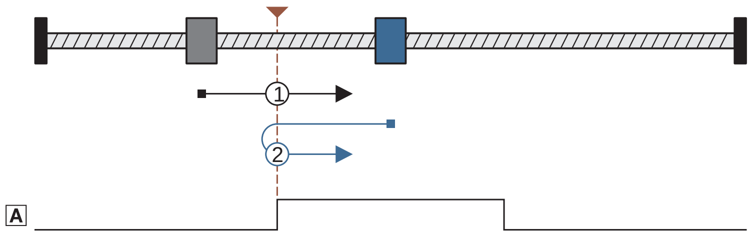
Case 1: Axis has not yet activated the reference switch:

1. Movement in positive direction with profile data set 1.
2. Reversing with positive edge of the reference switch and change to profile data set 2.
3. The following negative edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. The following negative edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1020: DS402 homing method 20



A Reference switch

Procedures:

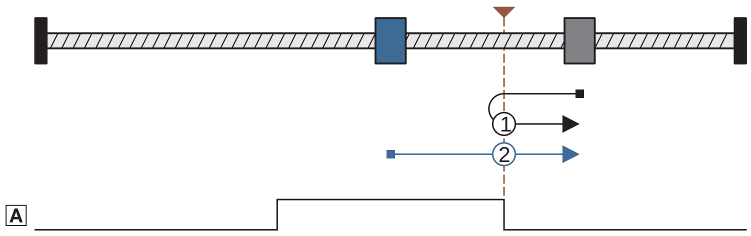
Case 1: Axis has not yet activated the reference switch:

1. Movement in positive direction with profile data set 1.
2. The following positive edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. The following positive edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1021: DS402 homing method 21



A Reference switch

Procedures:

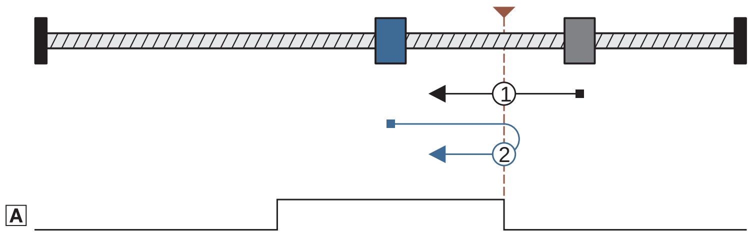
Case 1: Axis has not yet activated the reference switch:

1. Movement in negative direction with profile data set 1.
2. Reversing with positive edge of the reference switch and change to profile data set 2.
3. The following negative edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. The following negative edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1022: DS402 homing method 22



A Reference switch

Procedures:

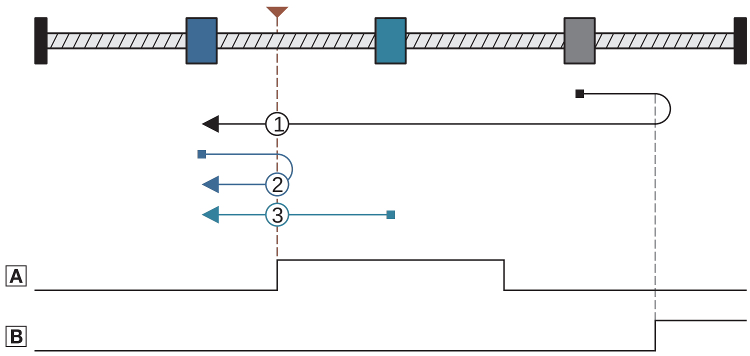
Case 1: Axis has not yet activated the reference switch:

1. Movement in negative direction with profile data set 1.
2. The following positive edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. The following positive edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1023: DS402 homing method 23



A Reference switch
B Positive travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch.
3. Positive edge of the reference switch activates profile data set 2.
4. The following negative edge of the reference switch sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

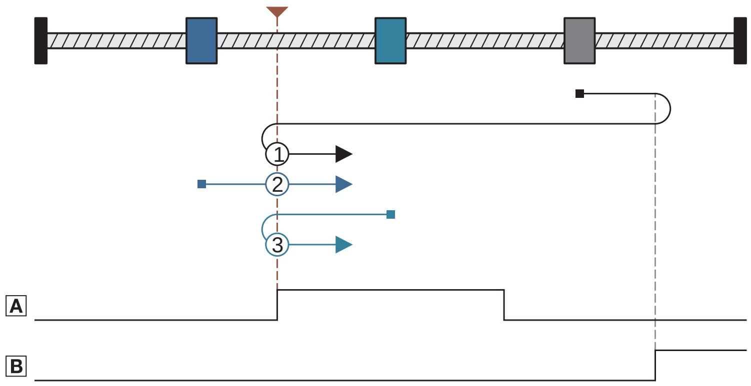
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing with positive edge of the reference switch and change to profile data set 2.
3. The following negative edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. The following negative edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1024: DS402 homing method 24



A Reference switch
B Positive travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch.
3. Positive edge of the reference switch activates profile data set 2.
4. Reversing with negative edge of the reference switch.
5. The following positive edge of the reference switch sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

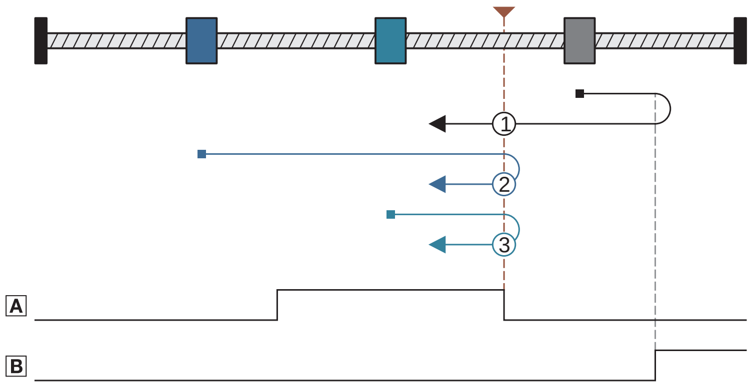
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. The following positive edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. The following positive edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1025: DS402 homing method 25



A Reference switch
B Positive travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch.
3. The following positive edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

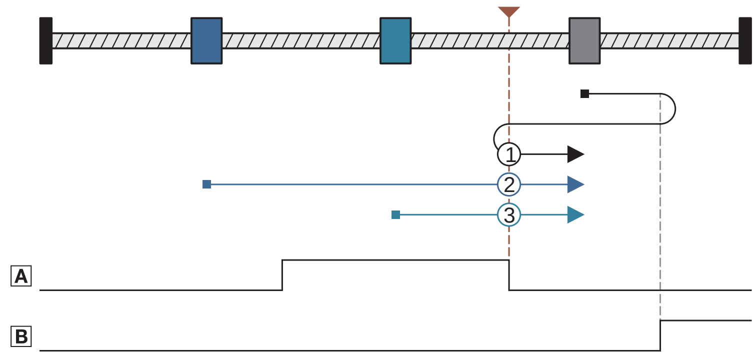
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. The following positive edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. The following positive edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1026: DS402 homing method 26



A Reference switch
B Positive travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Reversing to positive travel range limit switch.
3. Reversing with positive edge of the reference switch and change to profile data set 2.
4. The following negative edge of the reference switch sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

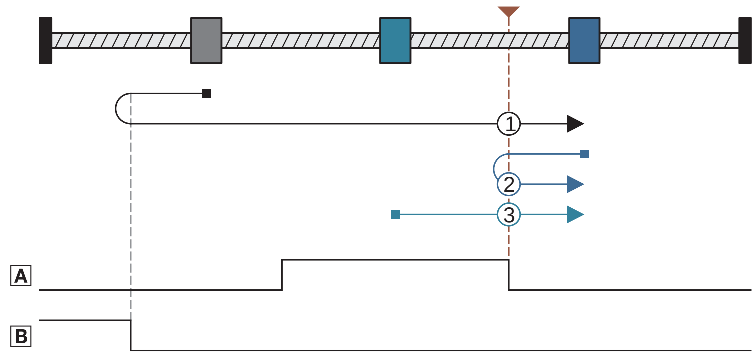
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in positive direction with profile data set 1.
2. Positive edge of the reference switch activates profile data set 2.
3. The following negative edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. The following negative edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1027: DS402 homing method 27



A Reference switch
B Negative travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch.
3. Positive edge of the reference switch activates profile data set 2.
4. The following negative edge of the reference switch sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

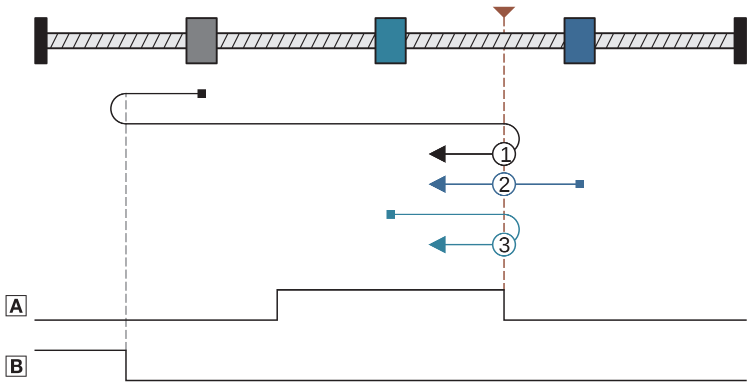
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing with positive edge of the reference switch and change to profile data set 2.
3. The following negative edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. The following negative edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1028: DS402 homing method 28



A Reference switch
B Negative travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch.
3. Positive edge of the reference switch activates profile data set 2.
4. Reversing with negative edge of the reference switch.
5. The following positive edge of the reference switch sets the reference.
6. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

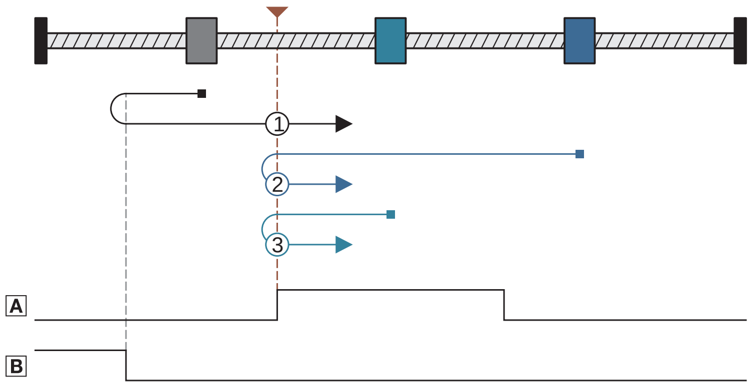
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. The following positive edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in positive direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. The following positive edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1029: DS402 homing method 29



A Reference switch
B Negative travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch.
3. The following positive edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

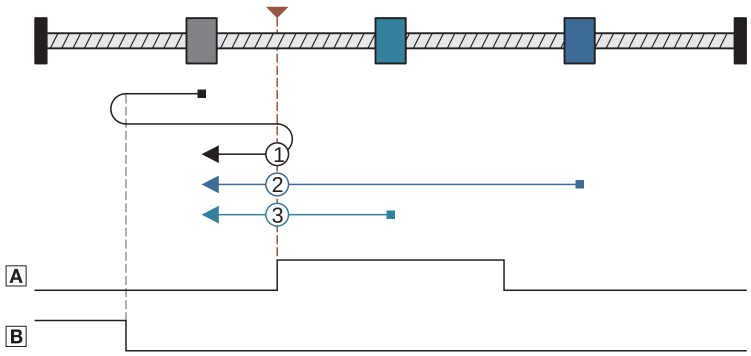
Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Positive edge of the reference switch activates profile data set 2.
3. Reversing with negative edge of the reference switch.
4. The following positive edge of the reference switch sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. Reversing with negative edge of the reference switch.
3. The following positive edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1030: DS402 homing method 30



A Reference switch
B Negative travel range limit switch

Procedures:

Case 1: Axis does not activate the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Reversing to negative travel range limit switch.
3. Reversing with positive edge of the reference switch and change to profile data set 2.
4. The following negative edge of the reference switch sets the reference.
5. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 2: Axis first activates the reference switch while moving towards the limit switch:

1. Movement in negative direction with profile data set 1.
2. Positive edge of the reference switch activates profile data set 2.
3. The following negative edge of the reference switch sets the reference.
4. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Case 3: Axis already stands on the reference switch:

1. Movement in negative direction with profile data set 2.
2. The following negative edge of the reference switch sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

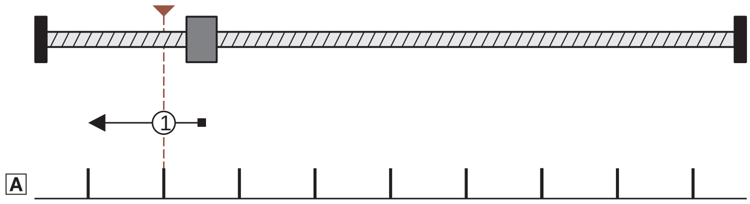
Mode 1031: DS402 homing method 31

Reserved: no homing is executed.

Mode 1032: DS402 homing method 32

Reserved: no homing is executed.

Mode 1033: DS402 homing method 33

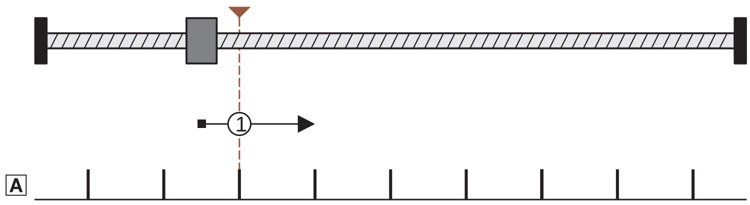


A Touch probe/zero pulse

Procedure:

1. Movement in negative direction with profile data set 1 and activation of the touch probe detection.
2. The following positive edge of the touch probe sensor sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1034: DS402 homing method 34

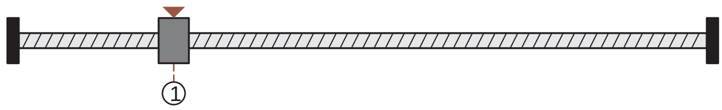


A Touch probe/zero pulse

Procedure:

1. Movement in positive direction with profile data set 1 and activation of the touch probe detection.
2. The following positive edge of the touch probe sensor sets the reference.
3. Absolute positioning to target position (C02643) with profile data set 2 (if C02641 = "0").

Mode 1035: DS402 homing method 35



Direct reference setting.

11.6.5 Execute homing

Requirements

- The controller is in the "Operation" device state.
- The basic function "Homing" is part of the active application.
- No other basic function is active.

Activation

To request the control via the basic function, the HM_*bEnable* enable input in the application must be set to TRUE.

- If no other basic function is active, a change-over to the "Homing active" function state is effected and homing can be carried out via the control inputs.
- A successful change to the "Homing active" function state is displayed by a TRUE signal at the *HM_bEnabled* status output.

Deactivation

When the *HM_bEnable* enable input is reset to FALSE, an active homing is stopped, i.e. the control inputs for homing are inhibited and the drive is braked to standstill within the deceleration time for stop.

- The status output *HM_bEnabled* is reset to FALSE and a change-over from the active "Homing active" function state via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.

11.6.5.1 Starting reference search/setting the reference directly

By setting the *HM_bActivateHoming* control input to TRUE, the reference search in the selected homing mode is started.

- During reference search, the *HM_bActive* status output is set to TRUE.
- By setting the status output *HM_bHomePosAvailable* to TRUE, it is already signalised during the reference search that the home position has been found. Depending on the homing mode selected, the drive traverses further on to the target position set in C02643.
- When the reference search is completed, the *HM_bActive* status output is reset to FALSE and the *HM_bDone* status output is set to TRUE.

Note!

In the homing mode "100: Set reference directly" no reference search is started, but the home position set in C02642 is directly accepted.

11.6.5.2 Loading home position via input

By setting the control input *HM_bLoadHomePos* to TRUE, the "Tool position" that is pending at input *HM_dnHomePos_p* is manually accepted as home position during the drive is at standstill. This is also possible if the controller is inhibited.

- The *HM_bDone* status output is set to TRUE for one cycle.
- The *HM_bHomePosAvailable* status output is set to TRUE.

Note!

For the encoderless motor control types (from software version V3.0) **the following applies:**

If V/f control or sensorless vector control has been selected, this function is only effective if the position controller has been selected for the position control (C02570 = "2: position controller active").

For the encoderless motor control types (from software version V5.0) **the following applies:**

If the V/f control or sensorless vector control is selected, this function can be activated irrespective of the use of the position controller.

11.6.5.3 Resetting home position

By setting the control input *HM_bResetHomePos* to TRUE, the "Home position known" status can be reset.

- The status outputs *HM_bDone* and *HM_bHomePosAvailable* are reset to FALSE.

Note!

For the encoderless motor control types (from software version V3.0) **the following applies:**

If V/f control or sensorless vector control has been selected, this function is only effective if the position controller has also been selected for the position control (C02570 = "2: position controller active").

For the encoderless motor control types (from software version V5.0) **the following applies:**

If the V/f control or sensorless vector control is selected, this function can be activated irrespective of the use of the position controller.

11.7 Positioning

The basic function "Positioning" provides the functions for executing the (travel) profiles and supports an "override" of speed and acceleration.

- A profile describes a motion request which can be implemented by this basic function into a rotary motion.
- A profile is described via the following profile parameters: Mode (type of positioning), position, speed, acceleration, deceleration, S-ramp time, final speed, standard sequence profile, TP sequence profile, TP window starting and end position and touch probe signal source(s).

Note!

For positioning, setpoint speeds greater than 30000 rpm are not possible. The speeds defined for these basic function are internally limited to 30000 rpm.

If the basic function is activated for a speed greater than 30000 rpm (e. g. if the basic function "Speed follower" is replaced), the internal limitation of the speed setpoint causes a speed step.

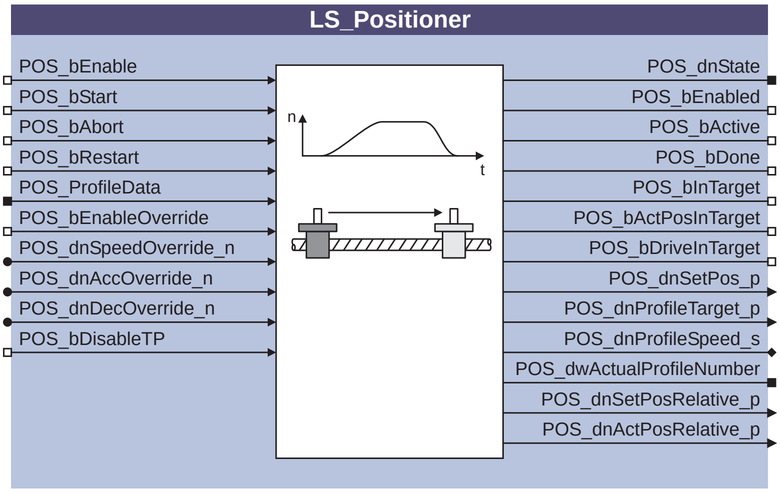
When the basic function is activated, a start acceleration is considered. ▶ Start acceleration/acceleration reduction when the basic function changes (📖 383)

For the encoderless motor control types (from software version V3.0) **the following applies:**

If no position controller has been selected for the position control in case of V/f control or sensorless vector control (C02570 = "1: Phase controller is active"), positioning is only executed via the speed profile resulting from the profile parameters. Because of this, the target positions set will only "roughly" be reached.

11.7.1 Internal interfaces | "LS_Positioner" system block

The **LS_Positioner** system block provides the internal interfaces for the basic function "Positioning" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Inputs

Identifier DIS code | data type	Information/possible settings	
POS_bEnable C02679/1 | BOOL	Requesting control via the basic function	
	TRUE	If no other basic function is active, a change-over to the "Positioning active" function state is effected and positioning can be carried out via the control inputs.
	TRUE↘FALSE	Active positioning is stopped, i. e. a change-over from the active "Positioning active" function state via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.
POS_bStart C02679/2 | BOOL	Starting positioning	
	FALSE↗TRUE	The profile *POS_ProfileData* is executed.
	FALSE↗TRUE (once again)	"Restart" • During an active positioning process, another profile can be specified via the input *POS_ProfileData* which is executed after restart. • Distances of a relative positioning that have already been covered are not taken into consideration.

Identifier DIS code | data type	**Information/possible settings**	
POS_bAbort C02679/3 | BOOL	Aborting or interrupting positioning	
	FALSE↗TRUE	The current profile is not completed but braked to standstill with the deceleration defined in the profile data.
	TRUE	A restart via *POS_bStart* or the continuation of an interrupted positioning via *POS_bRestart* is inhibited.
	FALSE	A restart via *POS_bStart* or the continuation of an interrupted positioning via *POS_bRestart* is possible again. • If the restart signal *POS_bRestart* is already effected during the deceleration phase, positioning is continued immediately.
POS_bRestart C02679/4 | BOOL	Continue interrupted positioning • Only possible if *POS_bAbort* has been reset from TRUE to FALSE.	
	TRUE	The positioning interrupted before via *POS_bAbort* is completed. • Distances of a relative positioning that have already been covered are taken into consideration.
	FALSE↗TRUE (once again)	"Restart" • During an active positioning process, another profile can be specified via the input *POS_ProfileData* which is executed after restart. • Distances of a relative positioning that have already been covered are taken into consideration.
POS_ProfileData	Pointer to the profile to be executed in internal units (increments) • A profile linkage results from the fact that a pointer to the sequence profile is contained within the profile.	
POS_bEnableOverride C02679/5 | BOOL	Activating override	
	TRUE	Override of the speed, acceleration, and deceleration is active.
POS_dnSpeedOverride_n C02677/1 | DINT	Value of speed override • Percentage multiplier for the current profile parameter "Speed". • Changes are accepted in each cycle. • $2^{30} \equiv 100$ % of the speed defined in the profile. • For values ≤ 1 % the status bit 18 is set. • Values ≤ 0 % are set to 0 % internally and lead to the standstill of the drive.	
POS_dnAccOverride_n C02677/2 | DINT	Value for acceleration override • Percentage multiplier for the current profile parameter "Acceleration". • Changes are accepted in each cycle. • $2^{30} \equiv 100$ % of the acceleration defined in the profile. • For values ≤ 1 % the status bit 19 is set. • Values ≤ 0 % are internally set to 0 % ("no acceleration").	
POS_dnDecOverride_n C02677/3 | DINT From V5.0	Value for deceleration override • Percentage multiplier for the current profile parameter "Deceleration". • Changes are accepted in each cycle. • $2^{30} \equiv 100$ % of the acceleration defined in the profile. • For values ≤ 1 % the status bit 19 is set. • Values ≤ 0 % are internally set to 0 % ("no deceleration").	
POS_bDisableTP C02679/6 | BOOL	Deactivating touch probe positioning	
	TRUE	Detected touch probes are ignored. There is no automatic change-over to the TP sequence profile defined in the profile data.

Outputs

Identifier DIS code | data type	Value/meaning	
POS_dnState C02675 | DINT	Status (bit coded) • When the basic function is not enabled, all bits are set to "0". • Bits which are not listed are not assigned with a status (always "0").	
	Bit 1	Positioning is active.
	Bit 2	Positioning is completed (all profiles have been executed).
	Bit 3	Acceleration/deceleration phase is active.
	Bit 4	Actual position in the target • The actual position value of the drive has reached the target of the last profile to be traversed within the tolerance window set in C02670.
	Bit 5	CCW rotation is active.
	Bit 6	Set position reached (in case of sequence profiles the drive continues to travel).
	Bit 10	Zero crossing in the positioning mode "modulo".
	Bit 11	Positioning cannot be continued.
	Bit 12	Drive in the target (actual position and set position in the target). • This status is available from software version V5.0.
	Bit 15	Fault in basic function active (group signal).
	Bit 16	Positioning is aborted.
	Bit 17	Reversing phase is active.
	Bit 18	Speed override ≤1 %
	Bit 19	Acceleration or deceleration override ≤ 1 %
	Bit 20	Position is limited by basic function "Limiter".
	Bit 21	Profile data are limited by basic function "Limiter".
	Bit 22	Direction is inhibited by basic function "Limiter".
	Bit 23	Abort by basic function "Limiter".
	Bit 24	Home position is not known.
	Bit 25	Stopping is active. • Basic function is enabled for the first time but no positioning has been requested / is active yet.
	Bit 26	Cycle is not known.
	Bit 27	Invalid positioning mode.
	Bit 28	Invalid change of the positioning mode.
	Bit 29	Profile data are not plausible or incorrect.
	Bit 30	Profile generation error.
POS_bEnabled C02679/7 | BOOL	Status signal "Basic function is enabled"	
	TRUE	Positioning via the control inputs is possible. • The *POS_bEnable* enable input is set to TRUE and the controller is in the "Positioning active" function state.
POS_bActive C02679/8 | BOOL	Status signal "Basic function is active"	
	TRUE	Positioning is active (the drive axis is moving).
POS_bDone C02679/9 | BOOL	Status signal "Basic function is ready"	
	TRUE	Positioning is completed. • The profile is executed and no sequence profile is defined.

Identifier DIS code | data type	**Value/meaning**	
POS_bInTarget C02679/10 | BOOL	Status signal "Setpoint has reached target position"	
	FALSE	Positioning is still active or has been aborted.
	TRUE	The current position setpoint has reached the target position.
POS_bActPosInTarget C02679/11 | BOOL	Status signal "Actual position in the target" • In case of sequence profiles: the target position of the last profile to be traversed. ▸ Actual value-based evaluation "Target position reached" (📖 488)	
	FALSE	Positioning is still active or has been aborted.
	TRUE	The current actual position value of the drive has reached the target of the last profile to be traversed within the tolerance window set in C02670.
POS_bDriveInTarget C02679/12 | BOOL From V5.0	Status signal "Drive in the target" • In case of sequence profiles: the target position of the last profile to be traversed. • The status is also output if the basic function "Positioning" is deactivated. ▸ Actual value- and setpoint-based evaluation "Drive in the target" (📖 489)	
	FALSE	Positioning is still active or has been aborted.
	FALSE↗TRUE	The current actual position value of the drive has reached the target position within the profile to be traversed last within the tolerance window set in C02671. At this time, the current position setpoint has already reached the target position. • In the case of positioning processes with sequence profiles, the output is only set to TRUE after the last profile to be traversed.
	TRUE↘FALSE	The current actual position value of the drive has exited the tolerance and hysteresis window set in C02671 and C02672 again after a positioning process has been completed.
	FALSE↗TRUE (once again)	If C02673 = "1", a modulo evaluation is carried out in all cycles (Lenze setting): • The output is set to TRUE again if the current actual position value of the drive enters the tolerance and hysteresis window <u>in an optional modulo cycle</u> again.
		If C02673 = "0", a modulo evaluation is only carried out in the modulo cycle of the target setpoint: • The output is set to TRUE again if the current actual position value of the drive enters the tolerance and hysteresis window <u>in the same modulo cycle</u> again.
POS_dnSetPos_p C02678/1 | DINT	Current position setpoint in [increments] • Reference point is the home position.	
POS_dnProfileTarget_p C02678/2 | DINT	Target position of the current profile in [increments] • Reference point is the home position.	
POS_dnProfileSpeed_s C02676 | DINT	Current setpoint speed of the current profile as speed in [rpm] • Taking a speed override into consideration.	
POS_dwActualProfileNumber C02674 | DWORD	Profile number (1 .... 100) of the current profile	

Identifier DIS code | data type	Value/meaning
POS_dnSetPosRelative_p C02678/3 | DINT From V5.0	Current relative position setpoint of the current positioning in [increments] • The value is also output if the basic function "Positioning" is deactivated. • Reference point is the starting position of the current profile. • After a positioning process has been completed, the output keeps the last relative value of the setpoint profile. • The output is reset when a new positioning is started, or when the home position is set.
POS_dnActPosRelative_p C02678/4 | DINT From V5.0	Current relative actual position value of the current positioning in [increments] • The value is also output if the basic function "Positioning" is deactivated. • Reference point is the starting position of the current profile. • The output follows the current position even if the basic function "Positioning" is no longer active. • The output is reset when a new positioning is started, or when the home position is set.

11.7.1.1 Possibilities for the selection of the profile

For specifying as well as storing and managing (travel) profiles, the following function blocks are available:

Function block	Function
L_PosPositionerTable	...serves to store and manage up to 100 (travel) profiles and to "Teach" positions, speeds, accelerations/decelerations and S-ramp times. • A further important task of this FB is the conversion of the table values according to the preselected scaling in the **LS_DriveInterface** SB.
L_PosProfileTable	...serves to store and manage up to four (travel) profiles and allows the "teaching" of target positions. • In contrast to the FB **L_PosPositionerTable** this FB does not use any variable tables but the data of the profile parameters are entered directly into the assigned codes. • The position at the input *dnExtPos_p* is used as target position as a further specific feature for the selection of profile no. 1.
L_PosProfileInterface	...provides a profile data set for the **LS_Positioner** SB.

Related topics:

▶ Setting the S-ramp time (📖 385)

11.7.2 Parameter setting

Setting parameters is not required for the basic function "Positioning".

- After activating the function, the profile is executed which has been transferred from the application to the basic function via the input *POS_ProfileData*.
- For profiles with touch probe positioning mode (residual path positioning) touch probe is detected implicitly.

Related topics:

▸ Setting the S-ramp time (📖 385)

11.7.2.1 Actual value-based evaluation "Target position reached"

An actual value-based evaluation on whether the drive has reached the target position can be carried out by means of the output *POS_bActPosInTarget* and parameterisation of C02670.

- The output *POS_bActPosInTarget* is set to TRUE if the current actual position value of the drive has reached the target position of the profile to be traversed last within the tolerance window set in C02670.
  – Hence, for sequence profiles the evaluation is only valid for the target position of the last profile.
- If C02670 is set to "0" (Lenze setting), the evaluation is setpoint-based and the signal at the *POS_bActPosInTarget* output corresponds to the *POS_bDone* signal.

Tip!

In many cases the signal *POS_bActPosInTarget* only has to be evaluated if the setpoint has also reached the target position. This can for instance be implemented in the function block editor by a logic "AND" operation with the signal *POS_bDone*.

From software version V5.0 it is displayed whether the set position and the actual position are in the target via the output *POS_bDriveInTarget*. ▸ Actual value- and setpoint-based evaluation "Drive in the target" (📖 489)

11.7.2.2 Actual value- and setpoint-based evaluation "Drive in the target"

This function extension is available from software version V5.0 onwards!

An actual value- and setpoint-based evaluation on whether the drive is in the target can be carried out by means of the output *POS_bDriveInTarget* and parameterisation of C02671, C02672, and C02673.

- The output *POS_bDriveInTarget* is set to TRUE if the current actual position value of the drive has reached the target position of the profile to be traversed last within the tolerance window set in C02671.
  - At this time, the current setpoint value has already reached the target position, i. e. the actual position and set position are in the target.
  - In the case of positioning processes with sequence profiles, the output is only set to TRUE after the last profile to be traversed.
- The output *POS_bDriveInTarget* is reset to FALSE if the current actual position value of the drive has exited the tolerance and hysteresis window set in C02671 and C02672 again after a positioning process has been completed.
- How the modulo evaluation is to be carried out if the actual position value enters the tolerance and hysteresis window again can be set in C02673:
  - Modulo evaluation in all cycles (Lenze setting):
    The output *POS_bDriveInTarget* is set to TRUE again if the current actual position value of the drive enters the tolerance window again in an optional modulo cycle.
  - Modulo evaluation only in the modulo cycle of the target setpoint:
    The output *POS_bDriveInTarget* is set to TRUE again if the current actual position value of the drive enters the tolerance window again in the same modulo cycle.
- A new FALSE↗TRUE edge at the output *POS_bDriveInTarget* after a positioning process has been completed can for instance occur when the basic function is deactivated afterwards, and if the drive axis is skewed so that the tolerance and hysteresis window is exited and then the tolerance range is entered again.

Short overview of the parameters for the actual value- and setpoint-based evaluation:

Parameter	Information	Lenze setting	
		Value	Unit
C02671	Tolerance for target position	2.0000	Unit
C02672	Hysteresis for target position	1.0000	Unit
C02673	Activate DriveInTarget Modulo	All cycles	

11.7.3 Carrying out positioning

Requirements

- The controller is in the "Operation" device state.
- The basic function "Positioning" is part of the active application.
- No other basic function is active.

Activation

To request the control via the basic function, the POS_*bEnable* enable input in the application must be set to TRUE.

- If no other basic function is active, a change-over to the "Positioning active" function state is effected and positioning can be carried out via the control inputs.
- A successful change to the function state "Positioning active" is displayed by a TRUE signal at the *POS_bEnabled* status output.

Deactivation

When the *POS_bEnable* enable input is reset to FALSE, an active positioning is stopped, i.e. the control inputs for positioning are inhibited and the drive is braked to standstill within the deceleration time for stop.

- The status output POS_*bEnabled* is reset to FALSE and a change-over from the active "Positioning active" function state via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.

11.7.3.1 Starting positioning

By setting the control input *POS_bStart* to TRUE, the positioning process is started.

- The (travel) profile that has been transferred to the basic function via the input *POS_ProfileData* is traversed.

11.7.3.2 Aborting/interrupting positioning

By setting the control input *POS_bAbort* to TRUE, the active positioning can be aborted or interrupted.

- The current profile is not completed but braked to standstill with the deceleration defined in the profile data.
- If the control input *POS_bAbort* remains on TRUE, a restart or the continuation of an interrupted positioning is inhibited.
- After resetting the control input *POS_bAbort* to FALSE, a restart or the continuation of an interrupted positioning process is possible again.

11.7.3.3 Continue interrupted positioning

By setting the control input *POS_bRestart* to TRUE, an interrupted positioning process can be continued if the control input *POS_bAbort* has been reset to FALSE before.

- Distances of a relative positioning that have already been covered are taken into consideration.
- If the continuation of a positioning process via the input *POS_bRestart* is not possible, this is displayed via bit 11 of the status output *POS_dnState*.

The following applies to software versions lower than V3.0:

- If during an active or cancelled positioning process a change-over to the states "Quick stop active", "Drive is stopped", or "Drive at standstill" is effected, it is also possible to continue a positioning process via *POS_bRestart*, taking the distance that has already been covered into consideration.

The following applies from software version V3.0:

- If during an active or cancelled positioning process a change-over is effected (e. g. by activating a quick stop or inhibiting the controller), it is also possible to continue a positioning process via *POS_bRestart*, taking the distance that has already been covered into consideration.
- However, after a homing has been carried out again, or after the following machine parameters have been changed, a continuation of an interrupted positioning process via the control input *POS_bRestart* is no longer possible:
  - Encoder resolution (C00100)
  - Position encoder selection (C00490), motor encoder selection (C00495)
  - Gearbox factors (C02520, C02521, C02522, C02523)
  - Feed constant (C02524)
  - Motor mounting direction (C02527), position encoder mounting direction (C02529)
  - Traversing range (C02528)
  - Cycle (C02536) for modulo traversing range
  - Position control structure (C02570)

11.7.3.4 Activating override

"Override" is the change of profile parameters and their acceptance during the positioning process.

- When the input *POS_bEnableOverride* is set to TRUE, a speed and acceleration override occurs according to the override values applied to the inputs *POS_dnSpeedOverride_n* and *POS_dnAccOverride_n*.
  - The override values represent percentage multipliers with regard to the current profile parameters for speed and acceleration.
  - For override values ≤ 1 % a status bit is set.
  - Override values ≤ 0 % are internally set to 0 %.
  - Changes of the override values are accepted in each cycle.

Note!

The online change of speed and acceleration is effective from the profile start to the beginning of the deceleration phase. Thus, the deceleration phase cannot be changed via an override!

- If the override value for speed is 0 %, the drive is braked to standstill.
- If the override for the acceleration is 0 %, there is no acceleration.

- From software version V5.0 also a deceleration override via the input *POS_dnDecOverride_n* can be carried out if the input *POS_bEnableOverride* is set to TRUE.
  The deceleration override is effective:
  - During the deceleration phase of a profile
  - During an abort process
  - In the case of a speed change-over from a high to a low speed within a profile (e. g. if the speed override is used)

Note!

If the override value for deceleration is 0 %, there is no deceleration, i. e. the drive does not come to a standstill!

- If the input *POS_bEnableOverride* is reset to FALSE again, the speeds, accelerations, and decelerations are run again, which have been defined via the profile parameters. There is an immediate acceleration from the override speed to the speed set in the profile.

11.8 Position follower

This basic function is used as setpoint interface for position-controlled drives.

- The specified position setpoint can either refer to the encoder on the motor side or to the (position) encoder used additionally to detect the machine position. The selection of the encoder configuration serves to adapt the internal control structure accordingly.
- Instead of a position setpoint alternatively also a speed setpoint can be specified by an according selection in C02680; the set position is then calculated by the integration of the speed setpoint on the basis of the current actual position (relative positioning).
- If the direction of rotation of the motor has to be inverted due to the mounting position of the motor or the gearbox ratio available, the use of the control signals can be accordingly changed over by means of parameterisation.
- The speed feedforward control can also be executed with the position setpoint by a corresponding selection in C02681. Then, the speed is calculated by differentiation of the position setpoint.

STOP

Stop!

If a limit switch is approached by means of the basic function "Position follower" and by this a fault with the "Quick stop by trouble" response is activated, always a set/actual adjustment of the position has to be carried out before the fault is acknowledged, as otherwise an uncontrolled motor movement may result after the fault is acknowledged!

▸ Hardware limit positions (limit switch) (📖 522)

Note!

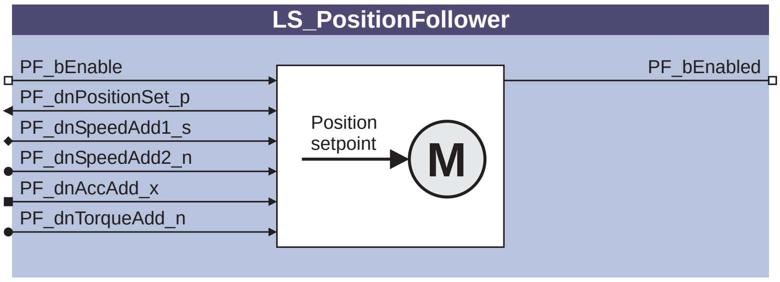
When the basic function is activated, a start acceleration is considered. ▸ Start acceleration/acceleration reduction when the basic function changes (📖 383)

For the encoderless motor control types (from software version V3.0) **the following applies:**

The basic function "Position follower" can only be activated for V/f control or sensorless vector control if the position controller has been selected for the position control (C02570 = "2: position controller active").

11.8.1 Internal interfaces | "LS_PositionFollower" system block

The **LS_PositionFollower** system block provides the internal interfaces for the basic function "Position follower" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

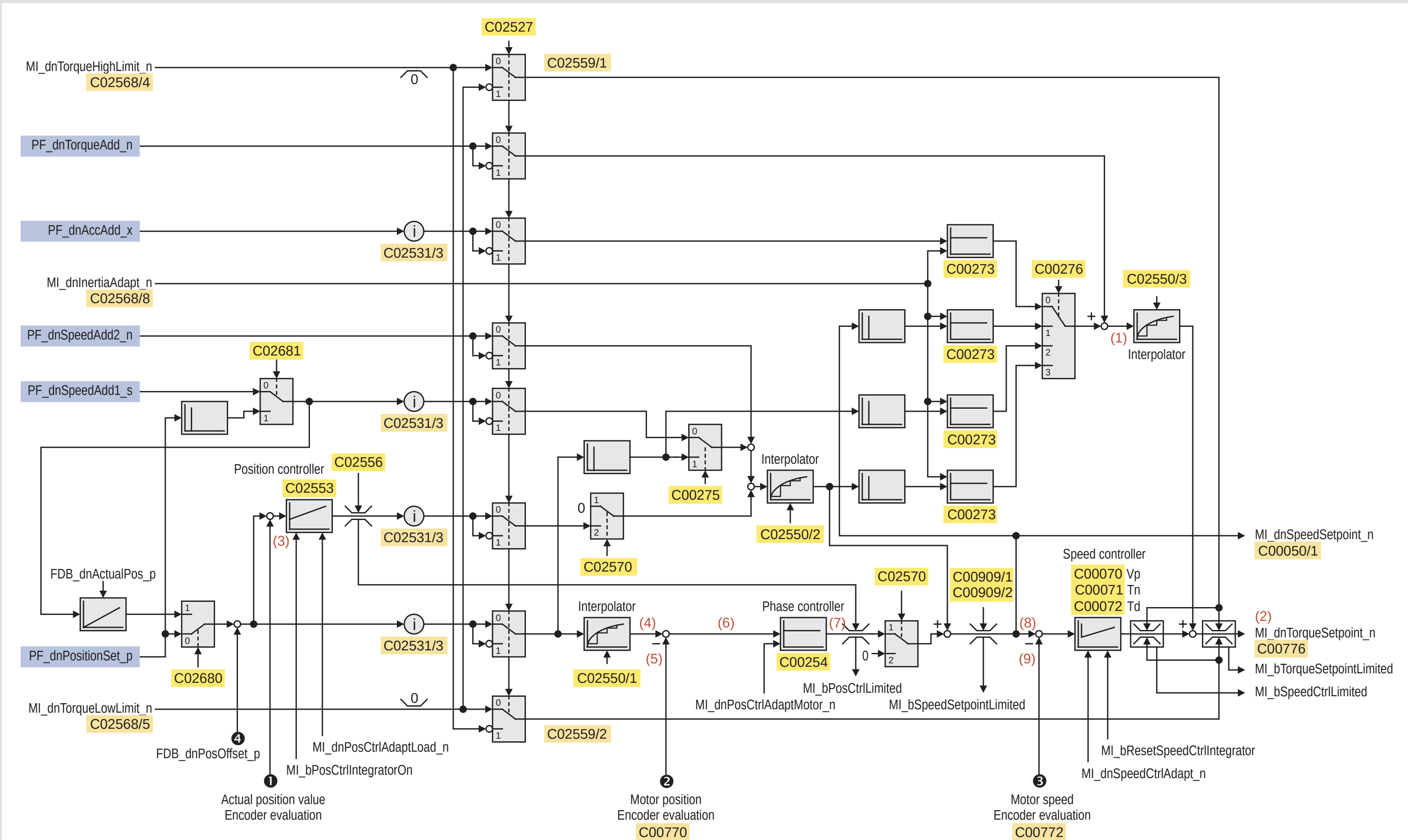
Inputs

Identifier DIS code | data type	Information/possible settings	
PF_bEnable C02689/1 | BOOL	Requesting control via the basic function	
	TRUE	If no other basic function is active, a change-over to the "Position follower active" function state is carried out, and the setpoints defined are accepted.
	TRUE↘FALSE	If no other basic function takes over the control of the drive, the drive is brought to standstill, i. e. a change-over from the active function state "Position follower active" via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.
PF_dnPositionSet_p C02688/1 | DINT	Position setpoint in [increments]	
PF_dnSpeedAdd1_s C02686 | DINT	Speed feedforward control value in [rpm]	
PF_dnSpeedAdd2_n C02687/1 | DINT	Additional speed setpoint in [%] • 100 % ≡ Motor reference speed (C00011)	
PF_dnAccAdd_x C02685 | DINT	Motor acceleration • For calculating the acceleration torque (for setting C00276 = "0"). • Selection as speed variation/time in [rpm/s]	
PF_dnTorqueAdd_n C02687/2 | DINT	Additive torque feedforward control value in [%] • 100 % ≡ motor reference torque (display in C00057/2).	

Outputs

Identifier DIS code | data type	Value/meaning	
PF_bEnabled C02689/2 | BOOL	Status signal "Basic function is enabled"	
	TRUE	The defined setpoints are accepted.

11.8.2 Signal flow



❶...❹ Signals from Signal flow - encoder evaluation (🕮 245)

FDB_... = input/output signals of the Encoder evaluation (🕮 241)
MI_... = input/output signals of the Motor interface (🕮 123)

[11-1] Signal flow - position follower

Internal variables of the motor control (oscilloscope signals)

- The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (590)

No.	Variable of the motor control	Meaning
(1)	Torque.dnTotalTorqueAdd	Additive torque feedforward control value
(2)	Torque.dnTorqueSetpoint	Torque setpoint
(3)	Position.dnActualLoadPos	Actual position
(4)	Position.dnPositionSetpoint	Position setpoint
(5)	Position.dnActualMotorPos	Current motor position
(6)	Position.dnContouringError	Following error
(7)	Speed.dnOutputPosCtrl	Output signal - phase controller
(8)	Speed.dnSpeedSetpoint	Speed setpoint
(9)	Speed.dnActualMotorSpeed	Current motor speed

11.8.3 Parameter setting

- Parameterisation dialog in the »Engineer«: Tab **Application parameters** → Dialog level *Overview* → *All basic functions* → *Position follower*
- Short overview of the parameters for the position follower:

Parameter	Information
C00050/1	Speed setpoint 1 [rpm]
C00070	Speed controller gain
C00071	Speed controller reset time
C00072	Speed controller rate time
C00273/1	Motor moment of inertia
C00273/2	Load moment of inertia
C00275	Signal source - speed setpoint
C00276	Signal source - torque setpoint
C00909/1	Upper speed limit value
C00909/2	Lower speed limit value
C02520	Gearbox factor numerator: Motor
C02521	Gearbox factor denom.: Motor
C02522	Gearbox factor num.: Pos. enc.
C02523	Gearbox fac. denom.: Pos. enc.
C02527	Motor mounting direction
C02550/1	Position setpoint interpolat.
C02550/2	Speed setpoint interpolat.
C02550/3	Torque setpoint interpolat.
C02553	Position controller gain
C02554	Position controller reset time
C02555	D component position controller
C02559	Internal torque limit
C02680	Source position setpoint
C02681	Source add. speed
Highlighted in grey = display parameter	

11.8.3.1 Setpoint interpolation

When the setpoint interpolation is activated, the motor control creates intermediate values to "smoothly" follow the setpoints which may be transferred from a slower task.

- C02550/1 = "1": The motor control follows the position setpoint in interpolated steps.
- C02550/2 = "1": The motor control follows the speed setpoint in interpolated steps.
- C02550/3 = "1": The motor control follows the torque setpoint in interpolated steps.

11.8.3.2 Inversion of the direction of rotation

Depending on the motor mounting position, if required, the direction of rotation can be inverted:

- C02527 = "0": Clockwise rotating motor ≡ positive machine direction.
- C02527 = "1": Counter-clockwise rotating motor ≡ positive machine direction.

11.8.4 Activating setpoint interface

Requirements

- The controller is in the "Operation" device state.
- The basic function "Position follower" is part of the active application.
- No other basic function is active.

Activation

To request the control via the basic function, the *PF_bEnable* enable input in the application must be set to TRUE.

- If no other basic function is active, a change-over to the "Position follower active" function state is carried out. Setpoints can now be defined via the corresponding inputs. ▶ Signal flow
- A successful change to the "Position follower active" function state is displayed by a TRUE signal at the status output *PF_bEnabled*.

Stop!

The basic function does not take over the control of the drive from the current speed, but immediately with the setpoint defined, which may cause a jerk!

Deactivation

When the *PF_bEnable* enable input is reset to FALSE, the setpoint inputs are inhibited. If the drive is not at standstill, it is braked to standstill within the deceleration time set for stop unless another basic function takes over the control of the drive.

- The status output *PF_bEnabled* is reset to FALSE and a change-over from the active "Position follower active" function state via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.

11.9 Speed follower

This basic function is used as setpoint interface for speed-controlled drives.

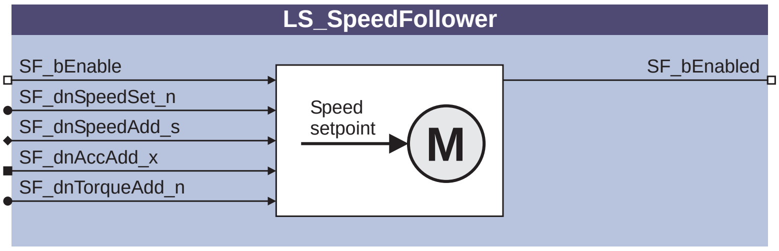
- The motor control is switched over automatically to speed control with torque limitation.
- If the direction of rotation of the motor has to be inverted due to the mounting position of the motor or the gearbox ratio available, the use of the control signals can be accordingly changed over by means of parameterisation.

Note!

When the basic function is activated, a start acceleration is considered. ▶ Start acceleration/acceleration reduction when the basic function changes (📖 383)

11.9.1 Internal interfaces | "LS_SpeedFollower" system block

The **LS_SpeedFollower** system block provides the internal interfaces for the basic function "Speed follower" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Inputs

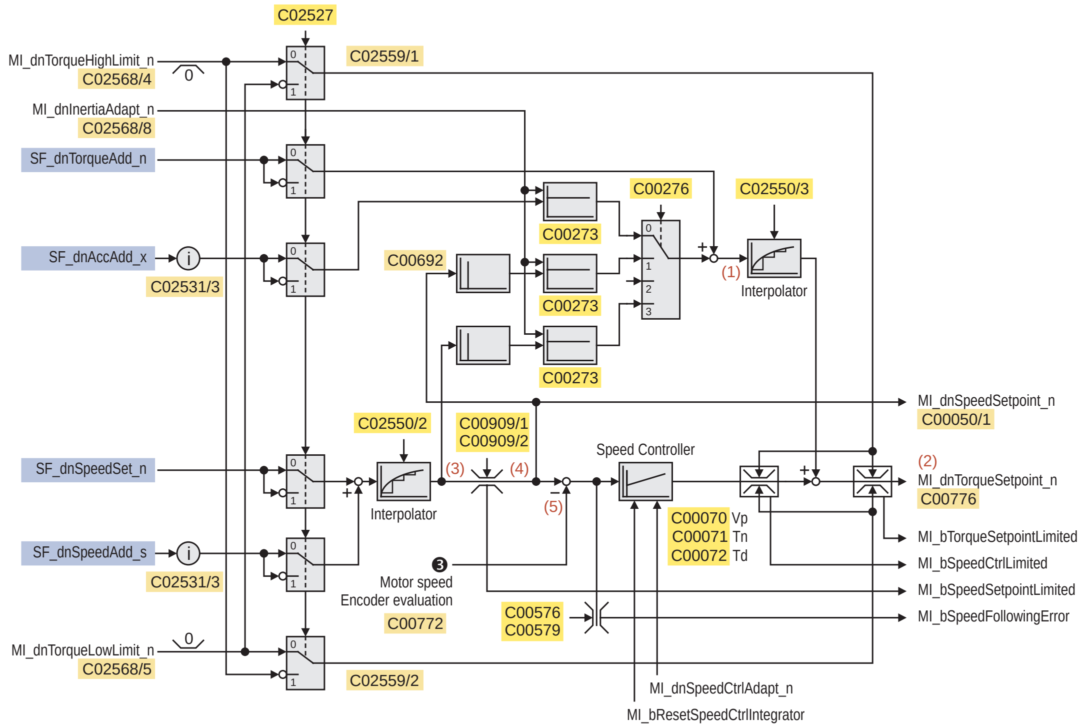
Identifier DIS code | data type	Information/possible settings	
SF_bEnable C02695/1 | BOOL	Requesting control via the basic function	
	TRUE	If no other basic function is active, a change-over to the "Speed follower active" function state is carried out, and the setpoints defined are accepted.
	TRUE↘FALSE	If no other basic function takes over the control of the drive, the drive is brought to standstill, i. e. a change-over from the active function state "Speed follower active" via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.

Identifier DIS code | data type	Information/possible settings
SF_dnSpeedSet_n C02694/1 | DINT	Speed setpoint in [%] • 100 % ≡ Motor reference speed (C00011)
SF_dnSpeedAdd_s C02693 | DINT	Additive speed setpoint in [rpm] • Without position control function.
SF_dnAccAdd_x C02692 | DINT	Motor acceleration • For calculating the acceleration torque (for setting C00276 = "0"). • Selection as speed variation/time in [rpm/s]
SF_dnTorqueAdd_n C02694/2 | DINT	Additive torque feedforward control value in [%] • 100 % ≡ motor reference torque (display in C00057/2).

Outputs

Identifier DIS code | data type	Value/meaning	
SF_bEnabled C02695/2 | BOOL	Status signal "Basic function is enabled"	
	TRUE	The defined setpoints are accepted.

11.9.2 Signal flow



❸ Signal from Signal flow - encoder evaluation (🕮 245)
MI_... = input/output signals of the Motor interface (🕮 123)

[11-2] Signal flow - speed follower

Internal variables of the motor control (oscilloscope signals)

- The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (🕮 590)

No.	Variable of the motor control	Meaning
(1)	Torque.dnTotalTorqueAdd	Additive torque feedforward control value
(2)	Torque.dnTorqueSetpoint	Torque setpoint
(3)	Speed.dnTotalSpeedAdd	Additive speed setpoint
(4)	Speed.dnSpeedSetpoint	Speed setpoint
(5)	Speed.dnActualMotorSpeed	Current motor speed

11.9.3 Parameter setting

- Parameterisation dialog in the »Engineer«: Tab **Application parameters** → Dialog level *Overview* → *All basic functions* → *Speed follower*
- Short overview of the parameters for the speed follower:

Parameter	Information
C00050/1	Speed setpoint 1
C00070	Speed controller gain
C00071	Speed controller reset time
C00072	Speed controller rate time
C00273/1	Motor moment of inertia
C00273/2	Load moment of inertia
C00276	Signal source - torque setpoint
C00576	Speed monitoring window
C00579	Resp. to speed monitoring
C00909/1	Upper speed limit value
C00909/2	Lower speed limit value
C02520	Gearbox factor numerator: Motor
C02521	Gearbox factor denom.: Motor
C02522	Gearbox factor num.: Pos. enc.
C02523	Gearbox fac. denom.: Pos. enc.
C02527	Motor mounting direction
C02531/3	Effective gearbox factor (dec.)
C02550/2	Speed setpoint interpolat.
C02550/3	Torque setpoint interpolat.
C02570	Position control structure
C02559	Internal torque limit
Highlighted in grey = display parameter	

11.9.3.1 Setpoint interpolation

When the setpoint interpolation is activated, the motor control creates intermediate values to "smoothly" follow the speed and/or torque setpoints which may be transferred from a slower task.

- C02550/2 = "1": The motor control follows the speed setpoint in interpolated steps.
- C02550/3 = "1": The motor control follows the torque setpoint in interpolated steps.

11.9.3.2 Inversion of the direction of rotation

Depending on the motor mounting position, if required, the direction of rotation can be inverted:

- C02527 = "0": Clockwise rotating motor ≡ positive machine direction.
- C02527 = "1": Counter-clockwise rotating motor ≡ positive machine direction.

11.9.4 Activating setpoint interface

Requirements

- The controller is in the "Operation" device state.
- The basic function "Speed follower" is part of the active application.
- No other basic function is active.

Activation

To request the control via the basic function, the *SF_bEnable* enable input in the application must be set to TRUE.

- If no other basic function is active, a change-over to the "Speed follower active" function state is carried out, and the motor control is automatically switched over to speed control with torque limitation. Setpoints can now be defined via the corresponding inputs. Signal flow
- A successful change to the function state "Speed follower active" is displayed by a TRUE signal at the *SF_bEnabled* status output.

STOP **Stop!**

The basic function does not take over the control of the drive from the current speed, but immediately with the setpoint defined, which may cause a jerk!

Deactivation

When the *SF_bEnable* enable input is reset to FALSE, the setpoint inputs are inhibited. If the drive is not at standstill, it is braked to standstill within the deceleration time set for stop unless another basic function takes over the control of the drive.

- The status output *SF_bEnabled* is reset to FALSE and a change-over from the active "Speed follower active" function state via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.

11.10 Torque follower

This basic function is used as setpoint interface for torque-controlled drives.

- The motor control is switched over automatically to torque control with speed limitation.
- If the direction of rotation of the motor has to be inverted due to the mounting position of the motor or the gearbox ratio available, the use of the control signals can be accordingly changed over by means of parameterisation.
- A stable speed limitation requires a minimum difference of the speed limit values of 50 rpm. If the defined speed limit values fall below this minimum difference, the internal lower speed limit value is lowered accordingly. The upper speed limit value remains unchanged. Signal flow - torque follower (510)

Note!

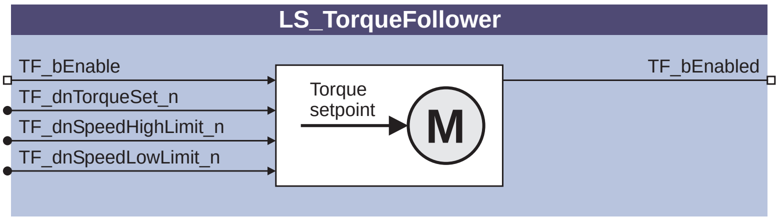
When the basic function is activated, a start acceleration is considered. Start acceleration/acceleration reduction when the basic function changes (383)

For the encoderless motor control types (from software version V3.0) **the following applies:**

The basic function "Torque follower" cannot be activated when the V/f control has been selected.

11.10.1 Internal interfaces | "LS_TorqueFollower" system block

The **LS_TorqueFollower** system block provides the internal interfaces for the basic function "Torque follower" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

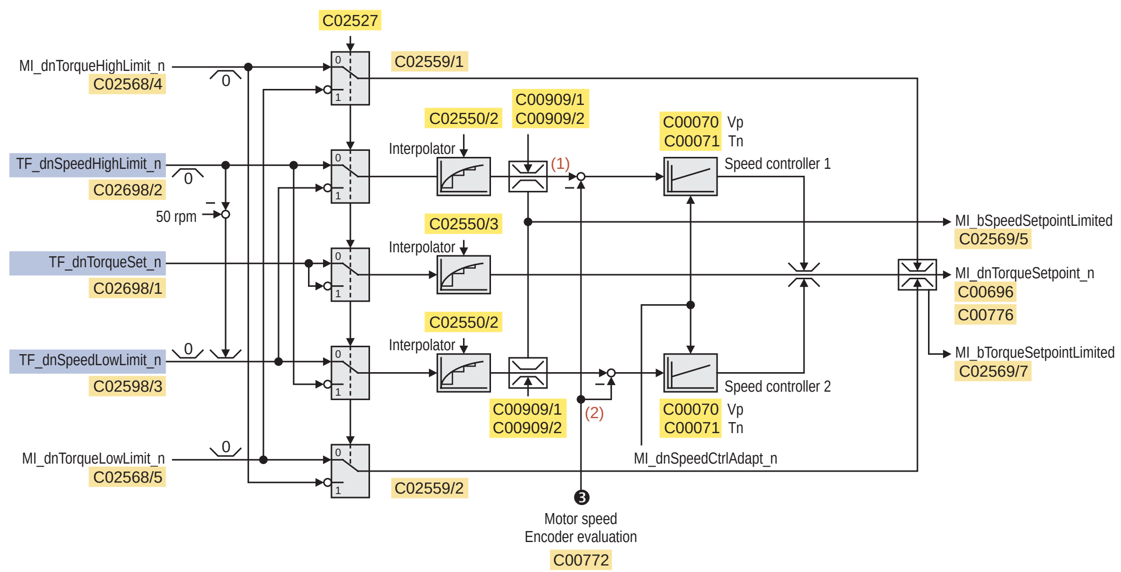
Inputs

Identifier DIS code | data type	Information/possible settings	
TF_bEnable C02699/1 | BOOL	Requesting control via the basic function	
	TRUE	If no other basic function is active, a change-over to the "Torque follower active" function state is carried out, and the setpoints defined are accepted.
	TRUE↘FALSE	If no other basic function takes over the control of the drive, the drive is brought to standstill, i. e. a change-over from the active function state "Torque follower active" via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.
TF_dnTorqueSet_n C02698/1 | DINT	Torque setpoint in [%] • 100 % ≡ motor reference torque (display in C00057/2).	
TF_dnSpeedHighLimit_n C02698/2 | DINT	Upper speed limit value in [%] for speed limitation • For positive direction of motion. • 100 % ≡ Motor reference speed (C00011). • Negative values are limited internally to the value "0".	
TF_dnSpeedLowLimit_n C02698/3 | DINT	Lower speed limit value in [%] for speed limitation • For negative direction of motion. • 100 % ≡ Motor reference speed (C00011). • Positive values are limited internally to the value "0".	

Outputs

Identifier DIS code | data type	Value/meaning	
TF_bEnabled C02699/2 | BOOL	Status signal "Basic function is enabled"	
	TRUE	The defined setpoints are accepted.

11.10.2 Signal flow



❸ Signal from Signal flow - encoder evaluation (📖 245)
MI_... = input/output signals of the Motor interface (📖 123)

[11-3] Signal flow - torque follower

Internal variables of the motor control (oscilloscope signals)

▶ The red numbers in brackets listed in the signal flow stand for internal variables of the motor control, which you can record by means of the Oscilloscope for purposes of diagnostics and documentation. (📖 590)

No.	Variable of the motor control	Meaning
(1)	Speed.dnSpeedSetpoint	Speed setpoint
(2)	Speed.dnActualMotorSpeed	Current motor speed

11.10.3 Parameter setting

- Parameterisation dialog in the »Engineer«: Tab **Application parameters** ➞ Dialog level *Overview* ➞ *All basic functions* ➞ *Torque follower*
- Short overview of the parameters for the torque follower:

Parameter	Information
C00050/1	Speed setpoint 1
C00050/2	Speed setpoint 2
C00070	Speed controller gain
C00071	Speed controller reset time
C00909/1	Upper speed limit value
C00909/2	Lower speed limit value
C02527	Motor mounting direction
C02550/2	Speed setpoint interpolat.
C02550/3	Torque setpoint interpolat.
C02559	Internal torque limit
Highlighted in grey = display parameter	

11.10.3.1 Setpoint interpolation

When the setpoint interpolation is activated, the motor control creates intermediate values to "smoothly" follow the speed and/or torque setpoints which may be transferred from a slower task.

- C02550/2 = "1": The motor control follows the speed setpoint in interpolated steps.
- C02550/3 = "1": The motor control follows the torque setpoint in interpolated steps.

11.10.3.2 Inversion of the direction of rotation

Depending on the motor mounting position, if required, the direction of rotation can be inverted:

- C02527 = "0": Clockwise rotating motor ≡ positive machine direction.
- C02527 = "1": Counter-clockwise rotating motor ≡ positive machine direction.

11.10.4 Activating setpoint interface

Requirements

- The controller is in the "Operation" device state.
- The basic function "Torque follower" is part of the active application.
- No other basic function is active.

Activation

To request the control via the basic function, the *TF_bEnable* enable input in the application must be set to TRUE.

- If no other basic function is active, a change-over to the "Torque follower active" function state is carried out, and the motor control is automatically switched over to torque control with speed limitation. Setpoints can now be defined via the corresponding inputs. ▶ Signal flow
- A successful change to the function state "Torque follower active" is displayed by a TRUE signal at the *TF_bEnabled* status output.

STOP **Stop!**

The basic function does not take over the control of the drive from the current speed, but immediately with the setpoint defined, which may cause a jerk!

Deactivation

When the *TF_bEnable* enable input is reset to FALSE, the setpoint inputs are inhibited. If the drive is not at standstill, it is braked to standstill within the deceleration time set for stop unless another basic function takes over the control of the drive.

- The status output *TF_bEnabled* is reset to FALSE and a change-over from the active "Torque follower active" function state via the "Drive is stopped" function state back to the basic state "Drive at standstill" is carried out.

11.11 Limiter

The basic function "Limiter" monitors the travel range limits via limit switches and parameterised software limit positions and can lead the drive to defined limit ranges when being requested accordingly by the safety module.

Danger!

The safety is exclusively ensured by the safety module!

When the request for the safety function is cancelled, the drive restarts automatically.

Ensure by external measures that the drive only starts after a confirmation (EN 60204).

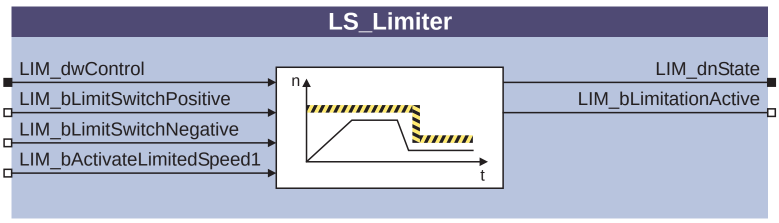
Note!

In order to make it possible for the basic function "Limiter" to lead the drive to the limit ranges defined **after a corresponding request by the safety module**, before the limits set for the safety module have been reached and it shuts down the drive, the limits for the basic function "Limiter" have to be set lower than the limits of the safety module!

See also: Safety engineering (📖 367)

11.11.1 Internal interfaces | "LS_Limiter" system block

The **LS_Limiter** system block provides the internal interfaces for the basic function "Limiter" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Inputs

Identifier DIS code | data type	Information/possible settings	
LIM_dwControl C02717 | DWORD	Interface to the safety module (bit coded) • For a simple connection of the safety module to the application, the transmission of the currently valid safety requirement(s) is effected via this bit coded control signal. • It is also possible to make several requirements at the same time via the control signal, e.g. manual jog with limited increment and limited speed 2. • Bits not listed are reserved for future extensions!	
	Bit 0	Switched-off torque: request controller inhibit. • This bit is no longer supported by the control signal of the **LS_SafetyModuleInterface** system block.
	Bit 1	Stop 1: request quick stop with subsequent controller inhibit.
	Bit 2	Stop 2: request quick stop. • If the automatic brake operation is activated, the brake remains open at standstill.
	Bit 3	Request limited speed 1. • Change of the traversing profile according to the parameters set for the limited speed 1 (C02708/1, C02710/1, C02711/1). • Only effective for the basic functions "Manual jog", "Homing" and "Positioning".
	Bit 4	Request limited speed 2. • Change of the traversing profile according to the parameters set for the limited speed 2 (C02708/2, C02710/2, C02711/2). • Only effective for the basic functions "Manual jog", "Homing" and "Positioning".
	Bit 5	Request limited speed 3. • Change of the traversing profile according to the parameters set for the limited speed 3 (C02708/3, C02710/3, C02711/3). • Only effective for the basic functions "Manual jog", "Homing" and "Positioning".
	Bit 6	Request limited speed 4. • Change of the traversing profile according to the parameters set for the limited speed 4 (C02708/4, C02710/4, C02711/4). • Only effective for the basic functions "Manual jog", "Homing" and "Positioning".
	Bit 7	Only permit positive direction of rotation. • Only effective for the basic functions "Manual jog", "Homing" and "Positioning".
	Bit 8	Only permit negative direction of rotation. • Only effective for the basic functions "Manual jog", "Homing" and "Positioning".
	Bit 10	Limited increment • Activate maximum distance set in C02713 as limited increment for the basic function "Manual jog".
	Bit 12	Defined limit positions • Activate software limit positions set in C02701/1 and C02701/2.

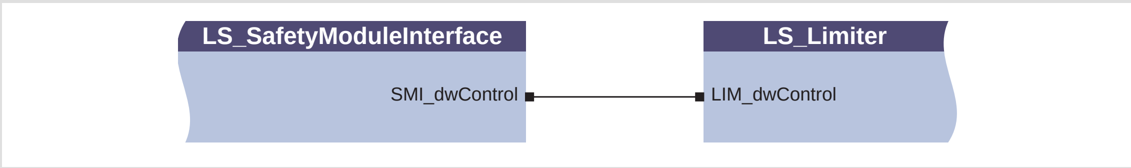
Identifier DIS code | data type	Information/possible settings	
LIM_bLimitSwitchPositive C02719/1 | BOOL	Input for positive travel range limit switch	
	TRUE	Limit switch is activated.
LIM_bLimitSwitchNegative C02719/2 | BOOL	Input for negative travel range limit switch	
	TRUE	Limit switch is activated.
LIM_bActivateLimitedSpeed1 C02719/3 | BOOL	Request limited speed 1 • If a setpoint follower is active, no limitation takes place, but an exceeding of the limit values is displayed via the output *LIM_dnState*.	
	TRUE	Request limited speed 1.

Outputs

Identifier DIS code | data type	Value/meaning	
LIM_dnState C02718 | DINT	Status word (bit coded) • Bits which are not listed are not assigned with a status (always "0").	
	Bit 0	Controller inhibit is initiated. (Safe torque off is requested; bit 0 of the *LIM_dwControl* control signal is set to "1".)
	Bit 1	Quick stop is initiated. (Safe stop 1 is requested; bit 1 of the *LIM_dwControl* control signal is set to "1".)
	Bit 2	Quick stop is initiated. (Safe stop 2 is requested; bit 2 of the *LIM_dwControl* control signal is set to "1".)
	Bit 3	Profile change due to speed limitation. (Limited speed 1 is requested; bit 3 of the *LIM_dwControl* control signal is set to "1".)
	Bit 4	Profile change due to speed limitation. (Limited speed 2 is requested; bit 4 of the *LIM_dwControl* control signal is set to "1".)
	Bit 5	Profile change due to speed limitation. (Limited speed 3 is requested; bit 5 of the *LIM_dwControl* control signal is set to "1".)
	Bit 6	Profile change due to speed limitation. (Limited speed 4 is requested; bit 6 of the *LIM_dwControl* control signal is set to "1".)
	Bit 7	Only positive direction of rotation is permissible. • When the direction of rotation is negative while requesting "Only positive direction of rotation", the drive is braked to standstill.
	Bit 8	Only negative direction of rotation is permissible. • When the direction of rotation is positive while requesting "Only negative direction of rotation", the drive is braked to standstill.
	Bit 10	Increment in manual jog mode is limited.
	Bit 12	Limitation of the set position is active.
	Bit 16	Positive limit switch inhibits travel in positive direction.
	Bit 17	Negative limit switch inhibits travel in negative direction.
	Bit 18	Positive software limit position inhibits travel in positive direction.
	Bit 19	Negative software limit position inhibits travel in negative direction.
	Bit 20	Limitation of speed is active.
	Bit 21	Limitation of acceleration is active.
	Bit 22	Limitation of deceleration is active.
	Bit 23	Limitation of jerk is active (S-ramp time is increased).
LIM_bLimitationActive C02715 | BOOL	Status signal "Limitation is active" (group signal)	
	TRUE	A limitation is active.

11.11.1.1 Interface to the safety module

For the simple connection of the safety module to the application, the transmission of the currently valid safety requirement(s) is effected in the form of a bit coded control signal via the following interface:



[11-4] Interface to connect the safety module to the basic function "Limiter"

- It is also possible to make several requirements at the same time via the control signal, e.g. manual jog with limited increment and limited speed 2.
- If no safety module is connected, the control signal can also be generated by means of a converter block (FB **L_DevSMControlEncoder**).

11.11.2 Parameter setting

- Parameterisation dialog in the »Engineer«: Tab **Application parameters** → Dialog level *Overview* → *All basic functions* → *Limiter*
- Short overview of the parameters for the limiter:

Parameter	Information
C02700	Software limits pos. effective
C02701/1	Positive software limit position
C02701/2	Negative software limit position
C02702	Limitations effective
C02703	Max. speed
C02704	Max. speed [rpm]
C02705	Max. acceleration
C02706	Min. S-ramp time
C02707	Permissible direction of rotation
C02708/1...4	Limited speed 1 4
C02709/1...4	Limited speed 1 .... 4 (display in [rpm])
C02710/1...4	Dec. limited speed 1 4
C02711/1...4	S-ramp time limited speed 1 4
C02712/1...4	Dec. time limited speed 1 4
C02713	Max. distance manual jog
C02714	Max. dist. manual control (display in [increments])
C02715	Limitation active (status display)
C02716/1	Resp. to rotation limitation
C02716/2	Resp. to SW lim. pos. exceeded
C02716/3	Resp. to max. value exceeded
C02720	Observation software limit positions
Highlighted in grey = display parameter	

Note!

The safety module has its own parameters.

Relevant to the basic function "Limiter" are the parameters of the safety modules for setting "Limited direction of rotation", "Speed with time limit" and "Limited increment (position)".

However, several other parameters of the safety module have no significance for the basic function "Limiter", e. g. the parameters for the configuration of the inputs of the safety module.

11.11.2.1 Software limit positions

The parameterisable software limit positions serve to limit the travel range.

Note!

Software limit positions are only evaluated and monitored if the drive knows the home position and the software limit positions are active (C02700 = "1").

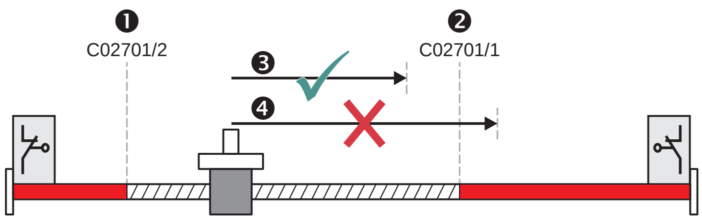
- When the traversing range is limited (C02528 = "1") and the software limit positions are not active, the range is limited by the software to the internal value range that can be maximally displayed ($\pm 2^{31}$ increments).
- For the "Modulo" traversing range (C02528 = "2") the software limit positions are generally not effective.
- If the error response that can be set in C02716/2 is deactivated or is only set to "Warning" or "Information", the software limit positions are not effective in an active manner for the basic functions "Speed follower", "Torque follower", and "Position follower"!

From software version V4.0 onwards, the triggering behaviour of the software limit position monitoring can be parameterised in C02720.

- If you want to maintain the device behaviour known from the previous versions, select "1: Based on set and actual value" in C02720.

▸ Triggering behaviour of software limit position monitoring (📖 521)

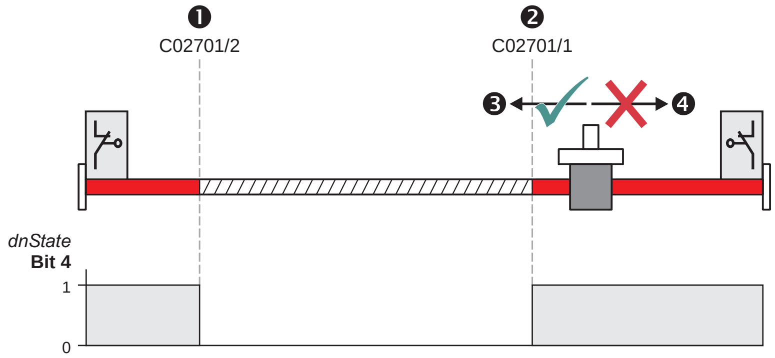
- ▶ The positive software limit position is set in C02701/1, and the negative software limit position is set in C02701/2.
- ▶ If the software limit positions are effective, travel commands the result of which would be that the permissible travel range would be exited can no longer be executed:



❶ C02701/2: Negative software limit position
❷ C02701/1: Positive software limit position
❸ Travel command possible
❹ Travel command not possible since the target position is outside the permissible traversing range.

[11-5] Example: Traversing range limitation by software limit position

- If the drive is already beyond the permissible travel range and the software limit positions are switched effectively, only travel commands resulting in the traversing of the drive back to the permissible travel range can be executed:



❶ C02701/2: Negative software limit position
❷ C02701/1: Positive software limit position
❸ Traversing in "retracting direction" is possible
❹ Traversing opposite to the "Retracting direction" is not possible

[11-6] Example: Permissible traversing direction for effective software limit positions

- If the software limit positions are active and a software limit position is "overtravelled":
  - The error response "quick stop by trouble" is carried out in the Lenze setting, i.e. the drive is braked to standstill within the deceleration time set for the quick stop function irrespective of the setpoint selection. The error response can be parameterised in C02716/2.
  - The fault message "Pos. SW limit switch overtravelled" or "Neg. SW limit overtravelled" is entered in the logbook of the controller.
  - A corresponding status is output via the *LIM_dnState* output.
  - Depending on the error response parameterised, the drive can only be traversed after the fault has been acknowledged.

See also: Manual jog to limit position (408)

11.11.2.2 Triggering behaviour of software limit position monitoring

This function extension is available from software version V4.0!

C02720 can be used to select the triggering behaviour of the software limit position monitoring of non-positon-controlled basic functions:

Selection "0: Based on set value"

From software version V4.0 onwards, this is the Lenze setting:

- If the basic functions "Speed follower" and "Torque follower" are used, the monitoring responds if the drive is outside the software limit positions and a command to travel in the "forbidden" direction is given (depending on the speed setpoint of the application).
- For all other non-position-controlled basic functions, the monitoring does not respond if a software limit position is exceeded.

Selection "1: Based on set and actual value"

This selection corresponds to the behaviour known from the previous versions (software versions < V4.0):

- For all non-position-controlled basic functions, the monitoring responds if the actual position exceeds a software limit position.
  - The monitoring also responds in the function states "Controller not ready" and "Error". This may cause the monitoring to trigger permanently if the drive traverses to a software limit position and controller inhibit is set subsequently because the actual position slightly changes around the software limit position.
- If the basic functions "Speed follower" and "Torque follower" are used, the monitoring also responds if the drive is outside the software limit positions and a command to travel in the "forbidden" direction is given (depending on the speed setpoint of the application).

Note!

If the position-controlled basic functions "Manual jog", "Positioning" and "Position follower" are used, the monitoring responds independently of the triggering behaviour parameterised in C02720 if one of the following cases occurs:

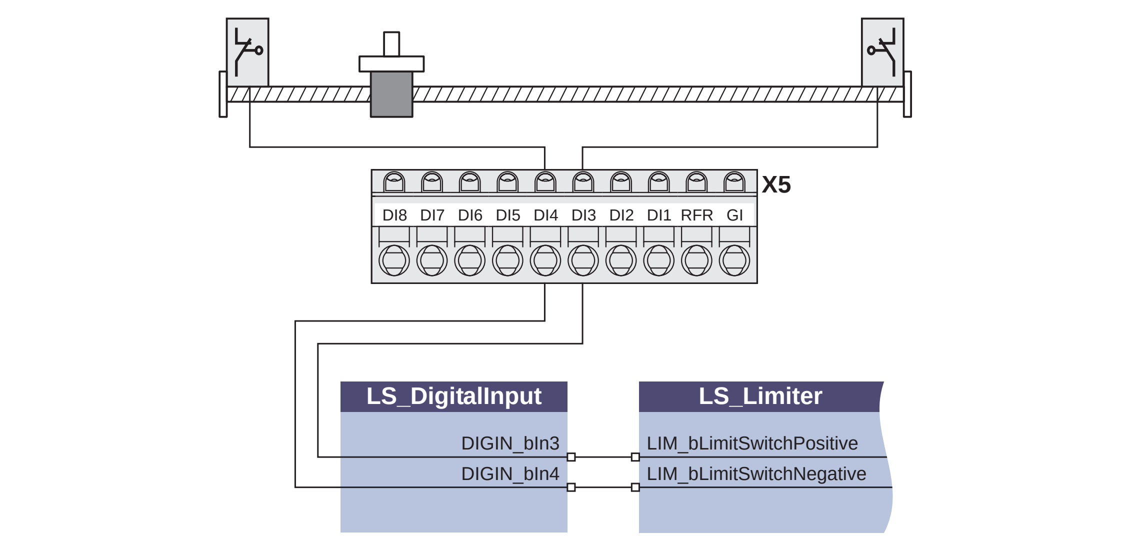
- The setpoint position exceeds a software limit position.
- A travel command is given which would cause the drive to leave the permissible travel range.
- The drive is outside the software limit positions and a command to travel in the "forbidden" direction is given.

If the software limit position monitoring is triggered, the error response parameterised in C02716/2 is carried out.

11.11.2.3 Hardware limit positions (limit switch)

Monitoring of the travel range limit by means of limit switches is effected via the inputs *LIM_bLimitSwitchPositive* and *LIM_bLimitSwitchNegative* of the **LS_Limiter** SB.

- The two inputs respond to the TRUE state and are to be connected to the corresponding digital inputs to which the limit switches are connected:



[11-7] Example: Connection of the travel range limit switches to the digital inputs DI3 & DI4

- If the limit switches are connected to decentralised terminals, the two inputs *LIM_bLimitSwitchPositive* and *LIM_bLimitSwitchNegative* can be connected to the decentralised terminal via a bus system (e. g. system bus).

Note!

If the digital inputs used for the connection of the limit switches are to be designed in a fail-safe manner (activation at LOW level), you simply change the terminal polarity of the corresponding digital inputs in C00114.

- If one of the two monitoring inputs is set to TRUE:
  - The error response "quick stop by trouble" is carried out, i.e. the drive is braked to standstill within the deceleration time set for the quick stop function irrespective of the setpoint selection.
  - The fault message "Pos. SW limit switch has tripped" or "Neg. SW limit has tripped" is entered in the logbook of the controller.
  - A corresponding status is output via the *LIM_dnState* output.
  - The drive can only be traversed again after the error has been acknowledged.

Stop!

If a limit switch is approached by means of the basic function Position follower and by this a fault with the "Quick stop by trouble" response is activated, always a set/actual adjustment of the position has to be carried out before the fault is acknowledged, as otherwise an uncontrolled motor movement may result after the fault is acknowledged!

Tip!

An activated limit switch can be retracted using the function "Retracting the limit switch". ▸ Retracting of an activated limit switch (📖 409)

See also: ▸ Manual jog to limit position (📖 408)

11.11.2.4 Limitations

Limit values for the basic functions "Manual jog", "Homing" and "Positioning" can be set via the following parameters:

Parameter	Information
C02703	Max. speed • Max. permissible speed that can be driven by the system. • This parameter depends, among other things, on the max. motor speed.
C02705	Max. acceleration • Max. permissible acceleration or deceleration for positioning processes. • This parameter depends, among other things, on the motor torque and moment of inertia of the entire mechanics which is driven during the positioning process.
C02706	Min. S-ramp time

- The parameters depend on the mechanics (e.g. the tool used).
- Usually the parameters must be changed when a tool is exchanged, e.g. by means of a recipe management of a superimposed control or via an HMI ("*Human Machine Interface*").

Note!

In order that the set limit values are effective, "1" must be selected in C02702.

- Irrespective of this setting, basically, the speed setpoint is limited to the motor reference speed (C00011)!

The limitations are not effective for the basic functions "Speed follower", "Torque follower" and "Position follower"!

- In case of these basic functions only speed and acceleration are monitored.
- If the limit values for speed and acceleration are exceeded, the response parameterised in C02716/3 is activated (Lenze setting: no response).
- **Background:** In the case of technology applications which are synchronised via an electrical shaft, the setpoint followers may not be limited, since synchronism would be lost by this. A possible consequence would be a collision of tools.

- If the limit values are switched effectively and a limit value that is set is exceeded:
  - The setpoints of the active basic function ("Manual jog", "Homing" or "Positioning") are changed (limited).
  - The response parameterised in C02716/3 (Lenze setting: "No response") is activated.
  - A corresponding error message is entered into the logbook of the controller.
  - A corresponding status is output via the *LIM_dnState* output.
  - The display parameter "Limitation active" (C02715) is set to "1: Activated".

11.11.2.5 Permissible direction of rotation

Via C02707, or alternatively via the input *LIM_dwControl* (generally by the control word of the safety module) the permissible direction of rotation for the basic functions "Manual jog", "Homing", and "Positioning" can be restricted.

Note!

The restriction of the permissible direction of rotation is not actively effective for the basic functions "Speed follower", "Torque follower" and "Position follower"!

- Only the response parameterised in C02716/1 is executed. (Lenze setting: "No response")

▶ If the permissible direction of rotation is restricted and a travel command in the inhibited direction of rotation is requested:

- The movement of the active basic function ("Manual jog", "Homing" or "Positioning") is cancelled.
- The response parameterised in C02716/1 (Lenze setting: "No response") is activated.
- The fault message "Pos. direction of rotation was limited" or "Neg. direction of rotation was limited" is entered in the logbook of the controller.
- A corresponding status is output via the *LIM_dnState* output.

11.11.2.6 Limited speed

"Limited speeds" for the basic functions "Manual jog", "Homing" and "Positioning" can be set via the following parameters:

Parameter	Information
C02708/1...4	Limited speed 1 4
C02710/1...4	Dec. limited speed 1 4
C02711/1...4	S-ramp time limited speed 1 4
C02712/1...4	Dec. time limited speed 1 4

Note!

The limited speeds are not effective for the basic functions "Speed follower", "Torque follower" and Position follower"!

- The request "Limited speed 1 ... 4" is effected via the input *LIM_dwControl*, generally by the control word of the safety module. If no safety module is available, the control word for the input *LIM_dwControl* can also be generated by means of an inverter.
- By means of the input *LIM_bActivateLimitedSpeed1* additionally the request of "Limited speed 1" can be effected, e. g. via a digital input that is connected to this input.
- If a limited speed is requested and the current speed exceeds the limited speed:
  - The setpoints of the active basic function ("Manual jog", "Homing" or "Positioning") are changed (limited).
  - The response parameterised in C02716/3 (Lenze setting: "No response") is activated.
  - The error message "Speed has been limited" is entered into the logbook of the controller.
  - A corresponding status is output via the *LIM_dnState* output.
  - The display parameter "Limitation active" (C02715) is set to "1: Activated".

Process example: "Manual jog"

1. Manual jog in positive direction is active and the manual speed is greater than the "Limited speed 1" set.
2. Via the control word of the safety module the "Limited speed 1" is requested.
3. The drive is decelerated to the "Limited speed 1" with the deceleration and S-ramp time set for the "Limited speed 1".
4. At the same time, a corresponding status is output via the *LIM_dnState* output.

Priorisation of the limited speeds

The following applies to software versions lower than V3.0:

- If several limited speeds are requested at the same time, the parameters of the limited speed with the lowest number are used, i.e. the "Limited speed 1" has the highest priority.

The following applies from software version V3.0:

- If several limited speeds are requested at the same time, the lowest speed with the greatest deceleration and the lowest S-ramp time is approached from the parameters of the requested limited speeds.

11.11.2.7 Limited increment for manual jog

Via C02713 the maximum permissible distance (limited increment) for the basic function "Manual jog" can be set.

- The request "Limited increment" is effected via the input *LIM_dwControl*, generally by the control word of the safety module. If no safety module is available, the control word for the input *LIM_dwControl* can also be generated by means of an inverter.
- In C02714 the maximum permissible distance in [increments] is displayed.

11.12 Brake control

This basic function is used for a wear free control and monitoring of a motor holding brake which in the simplest case is connected to the optionally available motor holding brake control module. Alternatively, the motor holding brake can be controlled via a digital output and monitored via a digital input.

Intended use

Motor holding brakes are used to hold axes in the case of controller inhibit or pulse inhibit and in the "Mains OFF" system state. This is not only important for vertical axes, but for instance also for horizontal axes for which an uncontrolled movement can bring about diverse problems.

Examples:

- Loss of the reference information after mains OFF and coasting of the drive.
- Collision with other moving machine parts.

Danger!

Please bear in mind that the motor holding brake is an important element of the safety concept of the entire machine. Thus maintain this system component with special care!

Stop!

Motor holding brakes at Lenze motors are not designed for braking during operation. The increased wear resulting from braking during operation may lead to an early destruction of the motor holding brake!

Please observe the notes in the hardware manual for mounting and electrical installation of the motor holding brake!

Note!

For the operation with motor holding brake control module:
- For single-axis controllers (Single Drive) the control (release) of the motor holding brake is only possible if both the DC-bus voltage and a 24-V supply voltage are available for the motor brake control!
- For multi-axis controllers (Multi Drive) the motor holding brake can also be released without a DC-bus voltage.

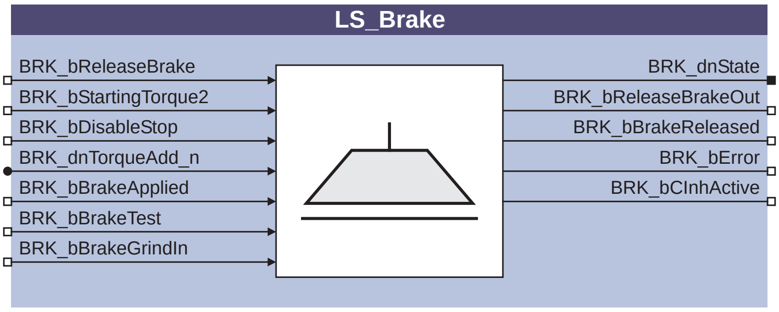
For the encoderless motor control types (from software version V3.0) **the following applies:**

The operation of vertical drives/hoists is
- only supported up to 55 kW by the V/f control!
- not supported by the sensorless vector control!

When the V/f control or sensorless vector control are selected, standstill monitoring is always switched off.

11.12.1 Internal interfaces | "LS_Brake" system block

The **LS_Brake** system block provides the internal interfaces for the basic function "Brake control" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Inputs

Identifier DIS code | data type	Information/possible settings	
BRK_bReleaseBrake C02609/1 | BOOL	Opening/closing the brake in connection with the selected operating mode	
	FALSE	Close the brake. • In automatic operation the internal brake logic takes over the control of the brake.
	TRUE	Release brake. • In automatic operation the internal brake logic is deactivated and the brake is released. This controller inhibit that may be set by the brake control is deactivated again.
BRK_bStartingTorque2 C02609/2 | BOOL	Selection of the torque feedforward control value • For the general use of the parameterisable starting torque as a feedforward control value, the setting C02588 = 0 is required. ▸ Torque feedforward control (📖 543)	
	FALSE	Starting torque 1 (C02586) is active.
	TRUE	Starting torque 2 (C02587) is active.
BRK_bDisableStop C02609/10 | BOOL	Prevent the brake from being applied in automatic operation • By this the drive remains position-controlled in the function states "Quick stop active", "Drive is stopped", and "Drive at standstill". • The input has no effect when the controller is inhibited.	
	TRUE	The application of the brake in automatic operation is prevented.
BRK_dnTorqueAdd_n C02608 | DINT	Additive torque value in [%] for torque feedforward control during start • 100 % ≡ C00057/2 ▸ Torque feedforward control (📖 543)	

Identifier DIS code | data type	Information/possible settings	
BRK_bBrakeApplied C02609/3 | BOOL	Input for status detection via switching contact at the brake • Activation of the input by setting C02583 = 1. ▶ Signal configuration (📖 534)	
	FALSE	Status "Brake is released".
	TRUE	Status "Brake is applied".
BRK_bBrakeTest C02609/4 | BOOL	Start/abort of the brake test ▶ Carrying out brake test (📖 560)	
	TRUE	Carry out brake test
	TRUE↘FALSE	Abort brake test (deactivate mode).
BRK_bBrakeGrindIn C02609/5 | BOOL	Start/abort of the brake grinding process ▶ Grinding the brake (📖 558)	
	TRUE	Brake grinding.
	TRUE↘FALSE	Abort grinding process (deactivate mode).

Outputs

Identifier DIS code | data type	Value/meaning	
BRK_dnState C02607 | DINT	Status (bit coded) • Bits which are not listed are not assigned with a status (always "0").	
	Bit 1	Brake control is active.
	Bit 4	Motor holding brake control module is used.
	Bit 8	Brake status (internal status signal).
	Bit 9	Torque feedforward control is active.
	Bit 10	Controller inhibit by brake is active or set.
	Bit 15	Fault is active (collective message).
	Bit 16	State "Grinding the brake".
	Bit 17	State "Brake test".
	Bit 18	State "Direct control".
	Bit 19	State "Automatic control".
	Bit 20	Error: External feedback.
	Bit 21	Error: Position drift when brake is applied/checked.
	Bit 22	Error: Monitoring of motor holding brake control module
	Bit 23	Information: Brake activation via waiting time.
	Bit 24	Information: Brake grinding process completed.
	Bit 25	Information: Brake test completed.
	Bit 26	Fault: Feedforward control torque could not be established within one second.
	Bit 27	Information: Current speed has fallen below the threshold for brake activation set in C02581.
BRK_bReleaseBrakeOut C02609/6 | BOOL	Control signal for triggering an external brake/status signal for control state	
	FALSE	Close the brake.
	TRUE	Release brake.
BRK_bBrakeReleased C02609/7 | BOOL	Status signal of the brake control considering the closing and opening time of the brake	
	FALSE	Brake is applied (after the brake closing time has elapsed).
	TRUE	Brake is released (after the brake opening time has elapsed).

Identifier DIS code | data type	Value/meaning	
BRK_bError C02609/8 | BOOL	Status signal "Brake error"	
	TRUE	An error has been detected.
BRK_bCInhActive C02609/9 | BOOL	Status signal "Controller inhibit"	
	TRUE	Controller inhibit has been set by brake control.

11.12.2 Parameter setting

Danger!

For the correct function of the brake control the different delay times in the following parameters have to be set correctly!

If the delay times are set incorrectly, a faulty control of the motor holding brake may be caused!

- Parameterisation dialog in the »Engineer«: Tab **Application parameters** ➞ Dialog level *Overview* ➞ *All basic functions* ➞ *Brake control*
- Short overview of parameters for brake control:

Parameter	Information
C02580	Operating mode brake
C02581	Threshold - brake activation
C02582	Brake resp. to pulse inhibit
C02583	Status input monitoring
C02585	Brake control polarity
C02586	Starting torque 1
C02587	Starting torque 2
C02588	Source of starting torque
C02589	Brake closing time
C02590	Brake opening time
C02591	Waiting time - status monitoring
C02593	Waiting time - brake activation
C02594	Test torque
C02595	Permissible angle of rotation
C02596	Grinding speed
C02597	Accel./decel. time - grinding
C02598	Grinding ON time
C02599	Grinding OFF time
C02600	Acceleration time feedf. control
C02601	Ref. for accel. time of brake
C02602	Source for feedf. control brake

Parameter	Information
C02603	Threshold 1 for opening brake
C02604	Threshold 2 for opening brake

11.12.2.1 Operating mode

Various operating modes are available in C02580 for different applications and tasks:

- Mode 0: Brake control is switched off (548)
- Mode 1/11: Direct control of the brake (549)
  - Without a specific logic or automatic system, can for instance be used to carry out a simple check on whether the brake switches correctly.
- Mode 2/12: Automatic control of the brake (550)
  - The normal mode for the control of mech. holding brakes with and without holding torque precontrol.

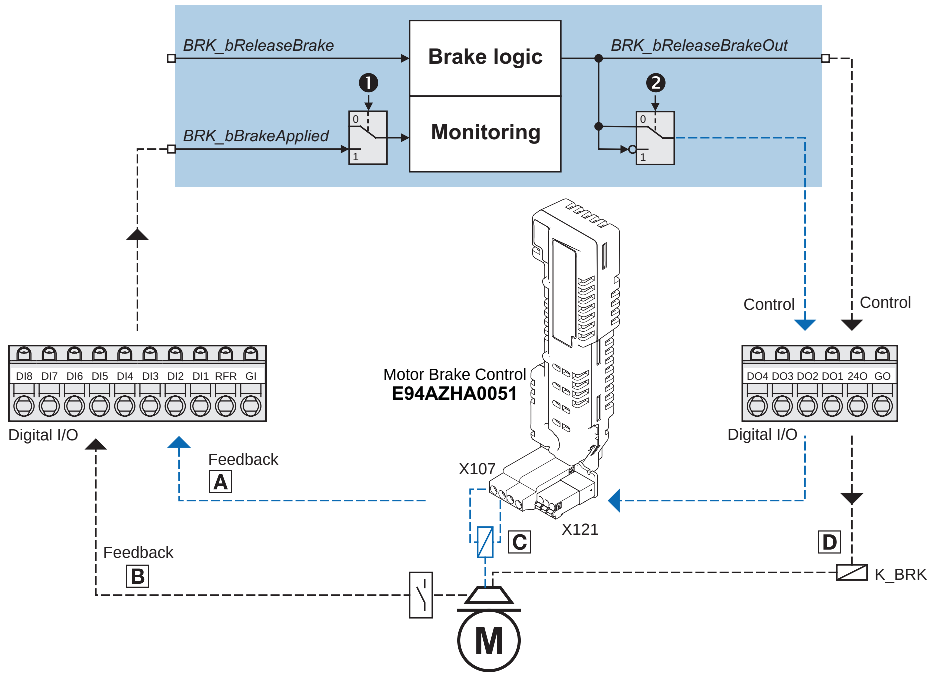
Function extension from software version V3.0:

- Mode 22: Automatic DC-injection braking (555)
  - DC-injection braking for V/f control and sensorless vector control.

11.12.2.2 Signal configuration

The signal configuration of the control and status signals for the brake logic and monitoring function is executed via the parameters shown in the following signal flow.

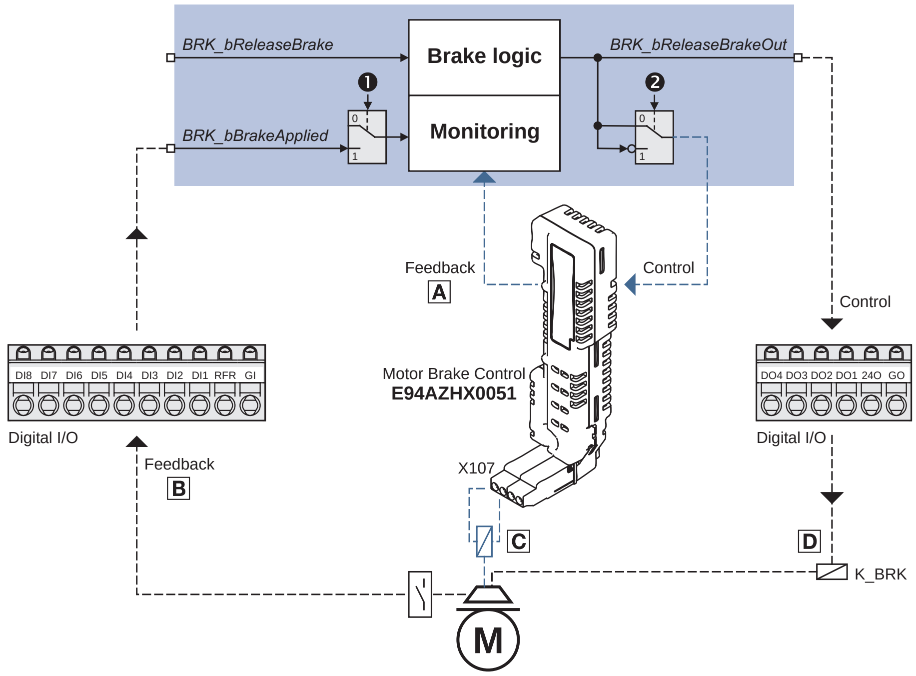
The design of the motor holding brake control module (here: E94AZHA0051) enables the external control of the holding brake via an additional 3-pole terminal (X121).



❶ C02583: Status input monitoring
❷ C02585: Brake control polarity (for motor holding brake control module)
A Feedback by "Motor holding brake control status""
B Feedback by "Brake function test"
C Brake control via motor holding brake control module (here: E94AZHA0051)
D Brake control via contactor at digital output

[11-8] Signal configuration of the control and status signal with the motor holding brake control module E94AZHA0051

The motor holding brake control module E94AZHX0051 does **not** include an additional 3-pole terminal for the external control of the holding brake.



❶ C02583: Status input monitoring
❷ C02585: Brake control polarity (for motor holding brake control module)
A Feedback by "Motor holding brake control status""
B Feedback by "Brake function test"
C Control of the brake via motor holding brake control module (here: E94AZHX0051)
D Brake control via contactor at digital output

[11-9] Signal configuration of the control and status signal with the motor holding brake control module E84AZHX0051

Note!

If an electrically holding (self-releasing) motor holding brake is to be controlled instead of an electrically releasing (self-holding) motor holding brake, the corresponding control and status signals must be inverted!

Please observe the notes in the hardware manual for mounting and electrical installation of the motor holding brake!

Status monitoring by "Motor holding brake control module status"

(See signal path A in fig. [11-8])

- Indirect status detection of the brake function.
- Monitoring of the motor holding brake control module and the electrical brake circuit.

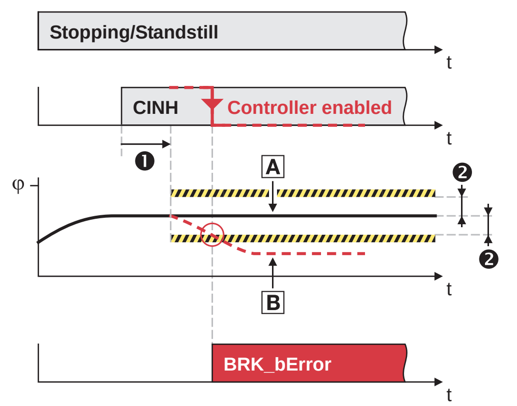
Status monitoring by "Brake function test"

(See signal path B in fig. [11-8])

- Direct function test of the complete brake circuit by microswitches at the brake.
- Wear control of the brake rotor.

11.12.2.3 Standstill monitoring

After the brake closing time and the waiting time for the status monitoring have elapsed, the standstill monitoring becomes active, i. e. the holding position is noted and compared to the permissible angle of rotation set in C02595 (Lenze setting: 5°) when the brake is applied.



A Normal behaviour
B Error behaviour
❶ C02591: Waiting time - status monit.
❷ C02595: Permissible angle of rotation
CINH = controller inhibit

[11-10] Automatic monitoring of the holding position

- If the stop position of the motor axis has changed by more than the permissible angle of rotation set in C02595, although the brake is engaged:
  - The error message "Motor brake: Angular drift with closed brake too high" is entered into the logbook.
  - "Quick stop by trouble" is activated as error response to avoid a further rotation/acceleration of the drive.
  - The error output *BRK_bError* is set to TRUE for one task cycle.
  - The status "position drift when brake is applied" is displayed at the *BRK_dnState* status output via bit 21 for one task cycle.

Note!

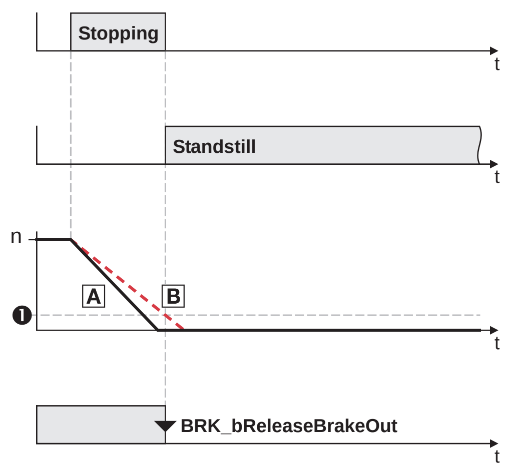
The standstill monitoring can be switched off by the setting C02595 = "0°".

For the encoderless motor control types (from software version V3.0) **the following applies:**

If the V/f control or the sensorless vector control is selected, the standstill monitoring is generally switched off, irrespective of the setting in C02595.

11.12.2.4 Brake activation in automatic operation

Brake activation through N < N_{min}



A Speed setpoint
B Actual speed value
❶ C02581: Brake activation threshold

[11-11] Process of brake activation through N < Nmin

- If the motor speed falls below the threshold for brake activation set in C02581, the function "Close brake" is activated in the automatic operation (mode 2/12).
- Here only the absolute value of the motor speed is considered, the direction of rotation remains unconsidered.
- In manual operation (mode 1/11) C02581 has no function.

Tip!

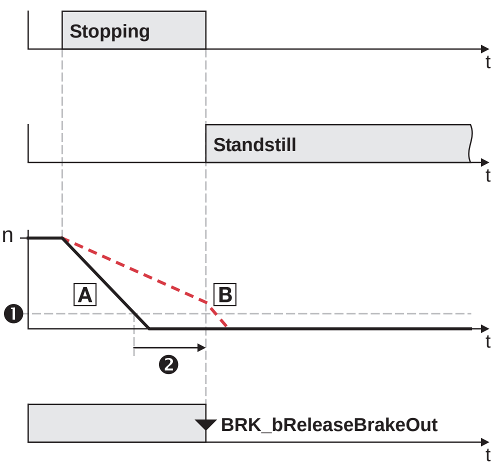
The value in C02581 should be set to approx. 5 ... 20 % of the maximum speed to minimise the wear of the brake and also provide for an optimum braking behaviour by a low grinding of the brake.

Brake activation through time-out

If a waiting time for the brake activation > 0 s is set in C02593, the time monitoring is active, i. e. the brake at the latest is activated for application after the waiting time has elapsed, even if the actual speed value is still above the threshold for the brake activation set in C02581.

Note!

In the Lenze setting the time monitoring is not active (C02593 = "0 s").



A Speed setpoint
B Actual speed value
❶ C02581: Brake activation threshold
❷ C02593: Waiting time for brake activation

[11-12] Process of brake activation through time-out

- The waiting time starts to elapse if the speed setpoint has reached the threshold for the brake activation.
- If the speed setpoint is still above the threshold after the waiting time has elapsed:
  - The brake is automatically triggered to close in automatic operation (mode 2/12).
  - The "brake activation via waiting time" status is displayed at the *BRK_dnState* status output via bit 23.
  - The information "Motor brake: Automatically activated after waiting time has elapsed" is entered in the logbook.

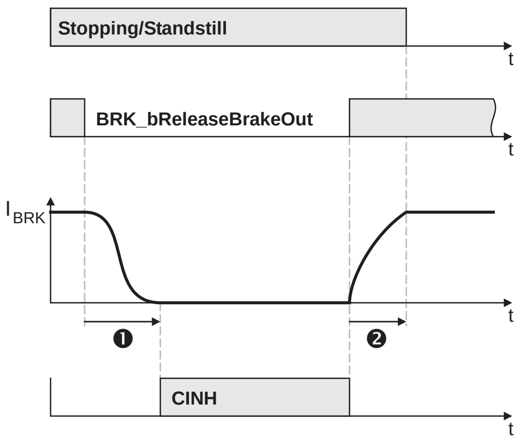
11.12.2.5 Brake time response

Application and opening time

Danger!

A wrong setting of the closing and opening time can cause a wrong activation of the motor holding brake!

- When the closing time is set too low, the controller is inhibited and the drive gets torqueless before the motor holding brake is closed completely.



❶ C02589: Brake closing time
❷ C02590: Brake opening time
CINH = controller inhibit

[11-13] Example: Definition of the closing and opening time of the PM brake

- ▶ Every mechanical motor holding brake has a construction-conditioned application and opening time which has to be taken into consideration by the brake control and which for this purpose has to be set in C02589 and in C02590.
- ▶ The information on the application and opening time of a Lenze-motor holding brake can be found in the corresponding Operating Instructions in the chapter "Technical data".
- ▶ If the application and opening times are set too great, this may be not critical with regard to safety, but causes unnecessary long delays in the case of cyclic braking processes.

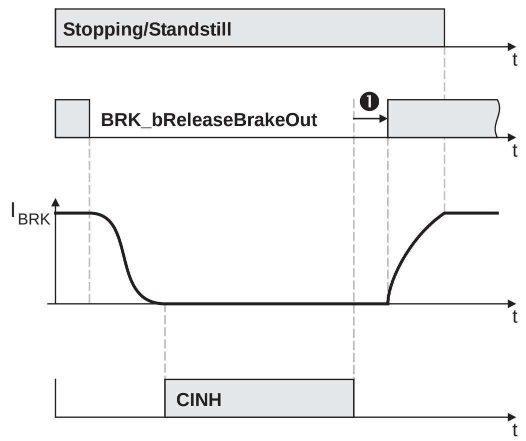
Tip!

The application and opening times do not only vary between the brake types, but they also depend on the basic conditions in the system, like for instance:

- Parameters of the hardware (cable length, temperature, supply voltage etc.)
- Contact elements used (motor holding brake control module or contactor at the digital output)
- Type of overvoltage limitation/suppressor circuit

For purposes of optimisation, the response times should be determined by measurement for each case.

Motor magnetising time (only for asynchronous motor)



❶ Motor magnetising time
CINH = controller inhibit

[11-14] Example: Consideration of the motor magnetising time in case of a PM brake

- For an asynchronous motor first the magnetic field required for the holding torque is generated after controller inhibit has been deactivated (in the case of a synchronous motor it is already available).
- The brake is only released if the actual torque has reached 90 % of the feedforward control torque.

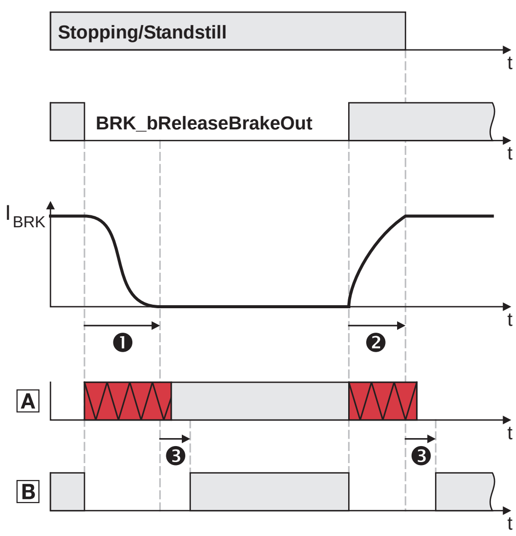
Waiting time for status monitoring

Every time the brake status changes, the waiting time set in C02591 is awaited after the brake opening or brake closing time has elapsed, before the monitoring of the motor holding brake control module and the status input *BRK_bBrakeApplied* (if activated via C02583) and the standstill monitoring function are switched active again.

- During the "Closing the brake" process, a mechanical contact must signal the "brake closed" state after the waiting time has elapsed.
- During the "Releasing the brake" process, a mechanical contact must signal the "brake released" state after the waiting time has elapsed.

Tip!

The additional waiting time is based on the fact that during the state change of the brake also state changes with regard to the monitored signals within the brake logic can occur, e. g. by bouncing the microswitch on the brake, or activation of the short circuit threshold within the motor holding brake control module due to discharge current peaks when the brake voltage is switched on. These state changes result in the activation of the monitoring function, although no stationary fault is pending.



A Brake status (with bouncing or transient response)
B Active status monitoring
❶ C02589: Brake closing time
❷ C02590: Brake opening time
❸ C02591: Waiting time - status monit.

[11-15] Definition of the waiting time for status monitoring

- The waiting time in C02591 must be set so that bouncing of a feedback contact and the transient response of the brake current monitoring will be suppressed completely.
- If no corresponding feedback takes place after the waiting time has elapsed:
  - The error output *BRK_bError* is set to TRUE until the next trigger attempt starts.
  - The error response "Quick stop by trouble" is activated.
  - The error message "Brake status error" is entered into the logbook.

11.12.2.6 Torque feedforward control

In automatic operation (mode 2/12) the brake control offers the possibility of precontrolling the required torque of the drive when the brake is released.

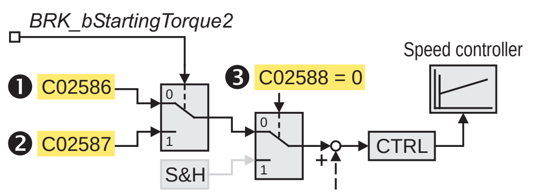
Note!

The torque is precontrolled for one second. During this time, the actual torque must have reached 90 % of the set torque, otherwise a fault is tripped!

The torque feedforward control is also supported for the V/f control (from software version V3.0) .

Via C02588 first the basic selection on whether a parameterised starting torque or the torque memorised during the last application process is to be used for the feedforward control.

Feedforward control with parameterised starting torque



❶ C02586: Starting torque 1
❷ C02587: Starting torque 2
❸ C02588: Source of starting torque

[11-16] Feedforward control with parameterised starting torque

- When C02588 = 0, a change-over between two starting torques is possible via input *BRK_bStartingTorque2*:
  - *BRK_bStartingTorque2* = FALSE: Starting torque 1 (C02586) is used.
  - *BRK_bStartingTorque2* = TRUE: Starting torque 2 (C02587) is used.

Application example:

A hoist drive is to be operated with different loads. Unfortunately we do not know when the load is available, but the starting direction (lifting or lowering) is known.

- In a no-load condition, the hoist drive needs a torque of 10 Nm. For holding the maximum load it needs a torque of 50 Nm.
- The change-over between lifting and lowering at start-up is done via the input *BRK_bStartingTorque2*.

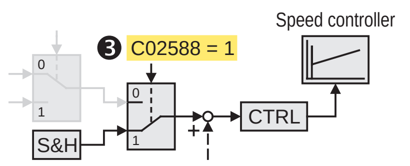
- To ensure the correct direction at start-up, the speed controller is loaded with the following starting torques:

	Lifting	Lowering
Starting torque:	C02586 = 50 Nm	C02587 = 10 Nm

- This results in the following behaviour depending on load and direction:

	Lifting	Lowering
Behaviour at max. load:	Optimum behaviour	Start-up a bit fast, but correct direction (non-critical).
Behaviour without load:	Start-up a bit fast, but correct direction (non-critical).	Optimum behaviour

Feedforward control with memorised torque



❸ C02588: Source of starting torque

[11-17] Feedforward control with parameterised starting torque

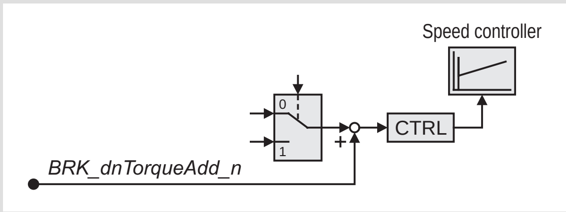
- When C02588 = 1, the starting torque is the setpoint which has been automatically memorised during the last closing process (falling below the speed threshold set in C02581).

Note!

The greater the threshold for the brake activation set in C02581, the greater the dynamic portion (e. g. the speed-dependent friction torque) in the memorised torque.

For the specific case that the load is altered while the motor holding brake is closed, a correction value for the torque feedforward control can be defined via the input *BRK_dnTorqueAdd_n* that is added to the memorised torque.

Further torque feedforward control options



[11-18] Feedforward control with parameterised starting torque

- Via the input *BRK_dnTorqueAdd_n* an additional feedforward control value can be defined.

Application example:

In the case of a hoist drive, the load is always known. For an optimum behaviour a torque proportional to the load and additionally 10 Nm as a constant feedforward control value should be loaded into the speed controller.

- As a constant feedforward control value the starting torque 1 is used (C02586 ="10 Nm", C02588 = "0", and *BRK_bStartingTorque2* = FALSE).
- Via the input *BRK_dnTorqueAdd_n* the torque is specified proportional to the load.

11.12.2.7 Torque feedforward control via ramp function

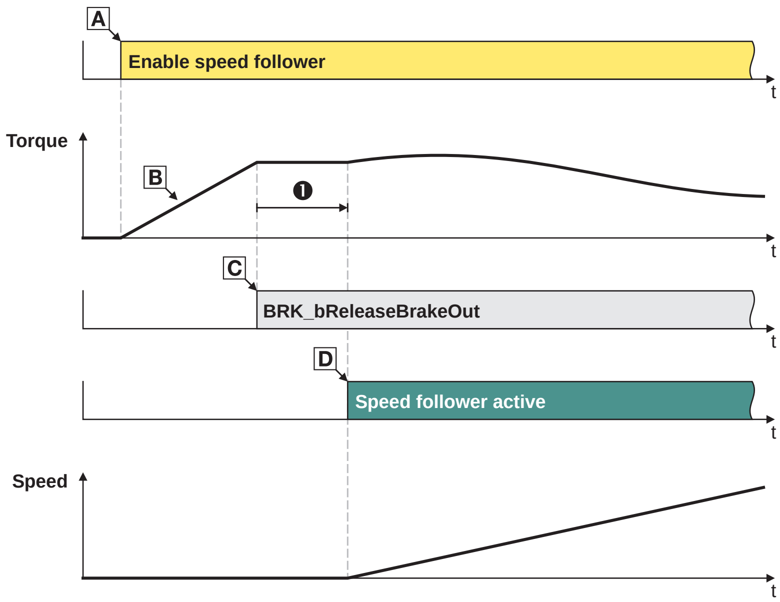
This function extension is available from software version V3.0!

The brake control additionally offers the possibility of establishing the required torque of the drive when the brake is released via a parameterisable ramp function.

Settings

1. Go to C02600 and set the acceleration time for the feedforward control.
2. Go to C02601 and select the reference for the acceleration time:
   – Selection "0: Motor reference value":
   The acceleration time refers to the generation of the motor reference torque (C00057/2), i.e. the acceleration is constant.
   – Selection "1: Starting current value":
   The acceleration time refers to the torque requested, i. e. the acceleration time is constant.

Sequence



A Request of basic function "Speed follower" ➔ open brake
B Torque is generated up to the setpoint via ramp function
C Brake is triggered for being opened
D After brake opening time has elapsed, the basic function "Speed follower" is activated
❶ C02590: Brake opening time

[11-19] Sequence of torque feedforward control via ramp function

11.12.2.8 Speed feedforward control via ramp function for V/f control

This function extension is available from software version V3.0!

For the V/f control there is the possibility of carrying out a feedforward control by means of a speed which is generated via a parameterisable ramp.

Note!

The operation of vertical drives/hoists is only supported up to 55 kW by the V/f control!

Settings:

1. Go to C02602 and set the selection "1: Speed" as source for the feedforward control.
2. Go to C02603 to set the speed threshold from which on the brake is to be opened.
   – C02604 can be used to parameterise a second speed threshold which can be activated by setting *BRK_bStartingTorque2* to TRUE.
3. Go to C02600 and set the acceleration time for the feedforward control.
4. Go to C02601 and set the reference for the acceleration time (0: Motor reference value, 1: Starting current value).

11.12.3 Mode 0: Brake control is switched off

If the mode 0 is selected in C02580, the brake control is switched off.

- If a motor holding brake control module is available, it will not be triggered.
- The brake monitoring function is not active.
- A potential fault reported by the brake control is reset automatically.
- The output signals of the system block LS_Brake are reset:
  - *BRK_dnState* = 0
  - *BRK_bReleaseBrakeOut* = FALSE
  - *BRK_bBrakeReleased* = FALSE
  - *BRK_bError* = FALSE

Note!

In the Lenze setting the mode 0 is preset to reach a safe state after mains connection.

11.12.4 Mode 1/11: Direct control of the brake

If the mode 1 or 11 has been selected in C02580, the brake is directly controlled via the input *BRK_bReleaseBrake*.

Tip!

Mode 1/11 can be used to easily check if the brake switches correctly.

- By the selection of the mode it is also defined in which way the brake is to be controlled:
  - Mode 1: Direct brake control via a motor holding brake control module.
  - Mode 11: Direct brake control via a digital output.

Note!

The digital outputs are not suitable for the "direct" control of a motor holding brake!

- The digital output used must be connected to a relay or power contactor which switches the brake supply.
- If a power contactor is used, the response and release time of the contactor contact is also added to the response and release time of the brake.

For the operation with motor holding brake control module:

- For single-axis controllers (Single Drive) the control (release) of the motor holding brake is only possible if both the DC-bus voltage and a 24-V supply voltage are available for the motor brake control!
- For multi-axis controllers (Multi Drive) the motor holding brake can also be released without a DC-bus voltage.

- Setting the pulse inhibit or controller inhibit does not influence the output signal.
- After the brake has been activated and the brake closing time has elapsed, the controller inhibit is set automatically by the basic function "brake control".
- For the operation with a motor holding brake control module (mode 1) the desired polarity for controlling the brake can be set in C02585.

11.12.5 Mode 2/12: Automatic control of the brake

If mode 2 or mode 12 is selected in C02580, the brake is controlled automatically, i. e. if another basic function is activated, which results in a traversing of the drive, the brake is automatically opened and operation is enabled. If the corresponding basic function is deactivated again, the drive is stopped via the basic function "Stop" and the brake is automatically closed again if the speed setpoint and the actual speed value are below the speed threshold set in C02581.

Tip!

The mode 2/12 is the common mode for brake control.

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

If *BRK_bReleaseBrake* = TRUE, the brake is released permanently and the automatic control cannot close the brake.

- By the selection of the mode it is also defined in which way the brake is to be controlled:
  - Mode 2: Current monitoring active, brake is automatically controlled via motor holding brake control module.
  - Mode 12: Current monitoring deactivated, brake is controlled via digital output. If a motor holding brake control module is installed, this will also be controlled.

Note!

The digital outputs are not suitable for the "direct" control of a motor holding brake!

- The digital output used must be connected to a relay or power contactor which switches the brake supply.
- If a power contactor is used, the response and release time of the contactor contact is also added to the response and release time of the brake.

For the operation with motor holding brake control module:

- For single-axis controllers (Single Drive) the control (release) of the motor holding brake is only possible if both the DC-bus voltage and a 24-V supply voltage are available for the motor brake control!
- For multi-axis controllers (Multi Drive) the motor holding brake can also be released without a DC-bus voltage.

- The brake is also activated automatically if a quick stop is activated in the drive, e.g. via the basic function "Quick stop" or as a response to a fault and also in case of a controller inhibit and pulse inhibit. ▸ Behaviour at pulse inhibit (📖 551)
- By setting the input *BRK_bDisableStop* to TRUE, an application of the brake at standstill or during quick stop can be avoided; by this the drive remains position-controlled.
- After the brake has been activated automatically and the brake closing time has elapsed, the controller inhibit is set automatically by the basic function "brake control".
- For the operation with a motor holding brake control module (in mode 2) the desired polarity for controlling the brake can be set in C02585.

11.12.5.1 Behaviour at pulse inhibit

In case of pulse inhibit the brake is applied. This occurs according to the parameter setting in C02582 either immediately (default setting) or delayed when the threshold set for brake activation is fallen below, which can be selected to protect the brake if high centrifugal masses occur.

Note!

Setting the pulse inhibit results causes the motor to coast down in a load-controlled manner until pulse enable is carried out again.

Pulse inhibit can be set in the enabled controller, e.g. due to a DC overvoltage, DC undervoltage or the "Safe torque off" request.

Stop!

Before setting the parameters of C02582 it is important to assess the energy conditions of the machine.

The amount of energy stored in the machine can exceed the permissible switching energy of a motor holding brake at the time of pulse inhibit and can thus destroy the brake when being applied!

Activate the brake in any case

When C02582 = "0", the brake is immediately triggered to close to prevent the mechanics from being damaged.

Only activate the brake below the threshold set for brake activation

When C02582 = "1", the brake remains released until the threshold set in C02581 for brake activation has been reached to protect the brake from excessive wear.

- Braking is exclusively executed by the friction of the load mechanics.
- Only when the motor speed has reached the threshold for brake activation, the brake will be closed.

Stop!

Do not set the threshold for brake activation in C02581 too high to protect the motor holding brake from wear!

Note!

For the encoderless motor control types (from software version V3.0) **the following applies:**

If V/f control without encoder or sensorless vector control is selected, there is no speed information for the controller in the case of pulse inhibit, therefore the threshold set in C02581 for the brake activation is not effective in this case.

In order to avoid that the motor holding brake is closed in case of pulse inhibit, a waiting time for the brake activation can be parameterised in C02593. In case of pulse inhibit, the motor holding brake is then only triggered to close after this application time has elapsed. ▸ Brake activation in automatic operation (538)

11.12.5.2 Process when brake is released

The following process occurs when a basic function is requested which causes the drive to traverse:

1. The controller inhibit is deactivated.
2. A magnetic field is created in the motor required for the holding torque (is already available in synchronous machines).
3. The feedforward control torque is loaded into the speed controller.
4. If the actual torque has reached 90 % of the feedforward control torque:
   - The output *BRK_bReleaseBrakeOut* is set to TRUE for releasing the brake.
   - Monitoring of the motor holding brake control module is deactivated temporarily.
   - Monitoring of the status input is deactivated temporarily (if switched active via C02583).
   - The brake opening time starts to elapse.
5. After the brake opening time has elapsed:
   - The output *BRK_bBrakeReleased* is set to TRUE.
   - The requested basic function is enabled.
6. After the additional waiting time set for the status monitoring in C02591 has elapsed:
   - Monitoring of the motor holding brake control module is active again.
   - Monitoring of the status input is active again (if switched active via C02583).

11.12.5.3 Process when brake is closed

The following process occurs if the enable of the requested basic function for traversing the drive is deactivated again:

1. The drive is brought to standstill via the basic function "Stop", or, where required, also via the basic function "Quick stop".
2. When speed setpoint and actual speed value have fallen below the speed threshold set in C02581:
   – The output *BRK_bReleaseBrakeOut* is set to FALSE for closing the brake.
   – The current torque is saved, so that, if necessary, it can be used for feedforward control during the next start.
   – Monitoring of the motor holding brake control module is deactivated temporarily.
   – Monitoring of the status input is deactivated temporarily (if switched active via C02583).
   – The brake closing time starts to elapse.
3. After the brake closing time has elapsed and the corresponding state change of the status signal:
   – The output *BRK_bBrakeReleased* is reset to FALSE.
   – The controller inhibit is activated.
4. After the additional waiting time set for the status monitoring in C02591 has elapsed:
   – Monitoring of the motor holding brake control module is active again.
   – Monitoring of the status input is active again (if switched active via C02583).
   – Standstill monitoring is activated. ▶ Standstill monitoring (📖 537)

11.12.6 Mode 22: Automatic DC-injection braking

This function extension is available from software version V3.0!

Note!

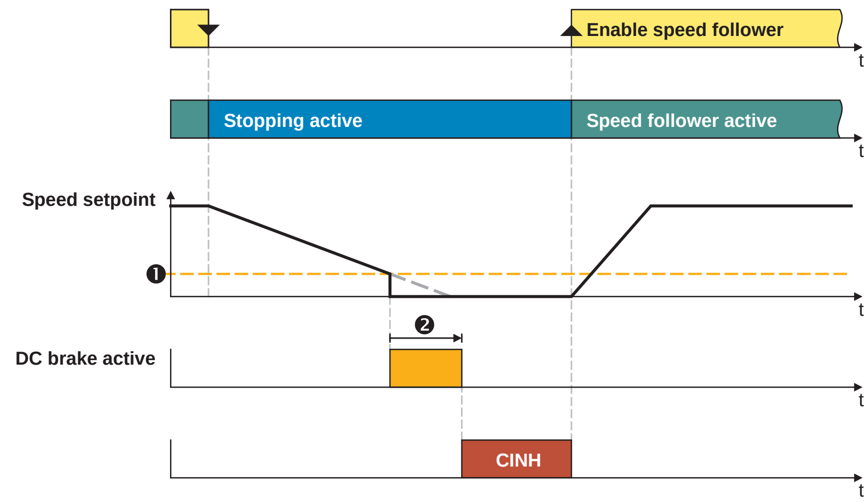
Automatic DC-injection braking is only possible if V/f control or sensorless vector control is selected as motor control type in C00006!

If mode 22 has been selected in C02580, DC-injection braking is executed automatically if the current speed setpoint falls below the speed threshold set in C02581.

- The automatic is only effective in the function states "Drive is stopped", "Drive at standstill", "Quick stop active", and "Fault".
- DC-injection braking is executed for the brake closing time set in C02589 with the braking current set in C00974.
- After the brake closing time has elapsed, the controller inhibit is set automatically by the basic function "brake control".

Danger!

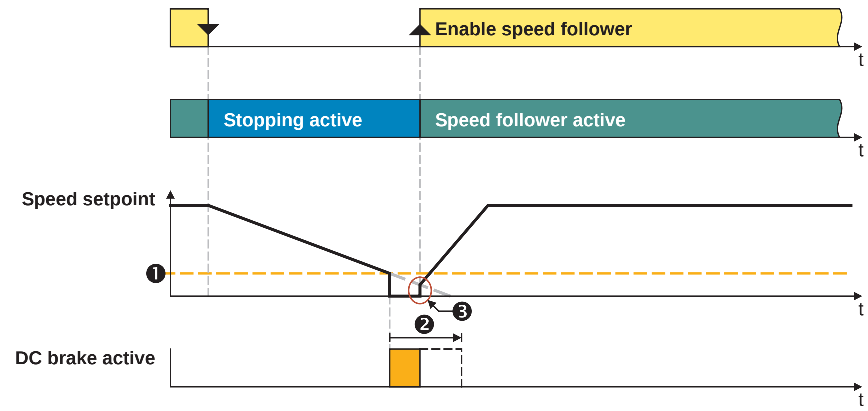
If the braking current is set too low, or the application time is too short, controller inhibit is set and the drive becomes torqueless before being completely braked to standstill by means of DC-injection braking!



❶ C02581: Brake activation threshold
❷ C02589: Brake closing time

[11-20] Example 1: Speed follower active ➞ stopping active (stopping time > brake closing time) ➞ speed follower active

- If a basic function is requested again before the brake closing time has elapsed, DC-injection braking is interrupted and - if activated in C00990 – the flying restart process is started and the basic function is activated:



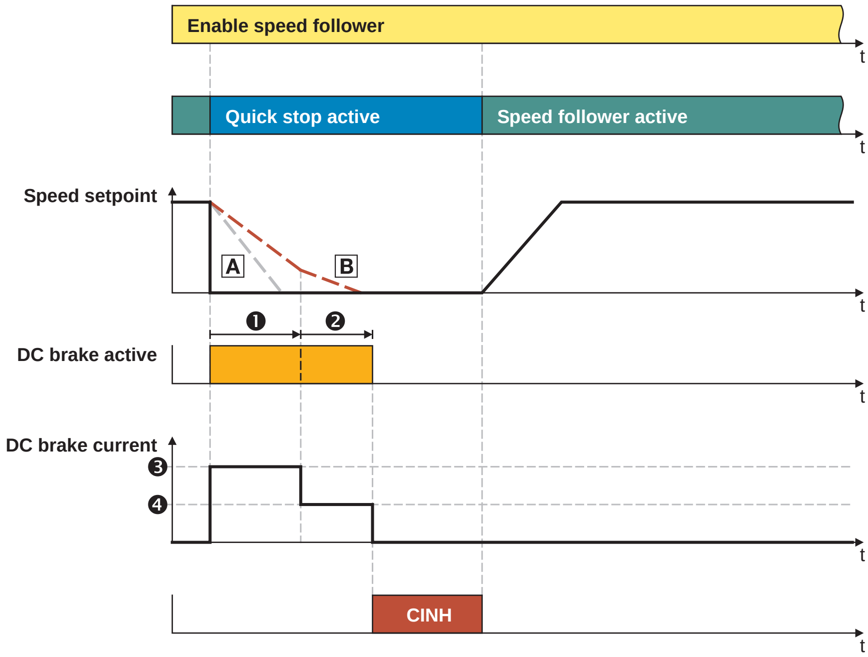
❶ C02581: Brake activation threshold
❷ C02589: Brake closing time
❸ C00990: Flying restart (activated here)

[11-21] Example 2: Speed follower active ➞ stopping active (stopping time > brake closing time) ➞ speed follower active

Automatic DC-injection braking when quick stop is activated

DC-injection braking is activated automatically if a quick stop is triggered in the drive, e.g. via the basic function "Quick stop" or as a response to an error.

- A change-over to the "Quick stop active" function state is effected, and for the quick stop deceleration time set in C00105 a DC-injection braking process with the braking current set in C00975 is carried out.
- After this time has elapsed, a change-over to the braking current parameterised in C00974 is carried out and DC-injection braking is continued with this braking current.
- After the brake closing time set in C02589 has also elapsed, the basic function "Brake control" automatically sets controller inhibit.
- The DC-injection braking in this case is also carried out when the "Quick stop by trouble" error response is actuated; however, instead of the "Quick stop active" function state, the "Fault" function state is active, and the controller is in the "Quick stop by trouble active" device state.



🄰 Actual speed value
🄱 Actual speed value if the quick stop braking current is parameterised too low (C00975)
❶ C00105: Quick stop decel. time
❷ C02589: Brake closing time
❸ C00975: Quick stop braking current
❹ C00974: Braking current

[11-22] Process example: speed follower is active ➞ quick stop activation ➞ speed follower is active

Note!

The quick stop braking current in C00975 has to be set so that the drive can be decelerated from the maximum operating speed to standstill within the deceleration time for quick stop set in C00105!

11.12.7 Grinding the brake

This function may be required after the brake has been replaced. The holding torque specified in the data sheet is only reached if the friction partners are ground in after mounting.

Stop!

If this function is activated, the drive is automatically accelerated to the grinding speed parameterised in C02596.

- The axis must move freely without driving against the travel range limitations.
- The maximally permissible friction energy of the brake must not be exceeded (observe the specifications of the manufacturer)!

$$W_{total}[J] \sim M_K[Nm] \cdot \frac{2\pi}{60} \cdot N[min^{-1}] \cdot t_{total}[s]$$

[11-23] Formula for estimating the friction energy during grinding process

Requirements

In order to be able to activate the grinding of the brake, the following conditions must be fulfilled:

- The grinding speed in C02596 is set higher than 0 rpm.
- The brake is activated, i.e. the "brake closing time" (C02589) and the "waiting time for status monitoring" (C02591) are elapsed.
- No other source for controller inhibit is active so that the controller inhibit can be deactivated by the brake control.

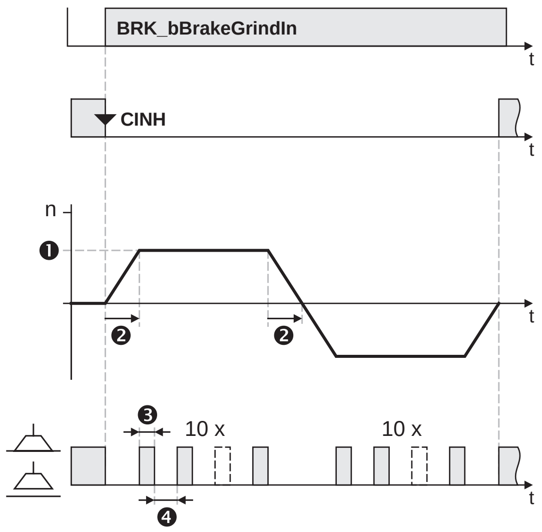
Note!

When grinding the brake, ensure that the motor shaft can be kept at speed against the closed holding brake.

- For this purpose, make sure that the maximum torque of the motor control (C00057/2) is higher than the holding torque of the brake.

Sequence

If all requirements mentioned before have been met, the grinding process can be started by setting the input *BRK_bBrakeGrindIn* to TRUE.



❶ C02596: Grinding speed
❷ C02597: Acc./dec. time - grinding
❸ C02598: Grinding ON time
❹ C02599: Grinding OFF time
CINH = controller inhibit

[11-24] Sequence of the grinding operation

- After the grinding speed has been reached, the friction partners in the brake are ground by a pulse-type control.
- After the brake has been closed and opened ten times, the direction of rotation changes and grinding in the opposite direction is carried out.
- By resetting the input *BRK_bBrakeGrindIn* to FALSE the grinding process can be aborted.

11.12.8 Carrying out brake test

This function can be used to check the holding torque of the brake.

Tip!

You can carry out this test in regular intervals, e. g. to detect a defect or wear of the brake at an early stage.

Note!

Due to possible deviations in the torque generation, the test of the holding torque cannot determine the holding torque exactly!

- The generated motor torque can deviate up to ±15 % from the default value depending on temperature.

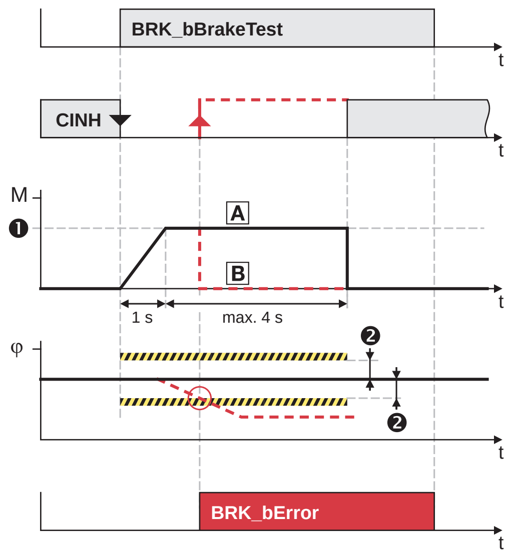
Requirements

In order to be able to activate the brake test, the following requirements have to be met:

- The test torque in C02594 is set higher than 0 Nm.
- The permissible angle of rotation is set greater 0° in C02595, thus standstill monitoring is active. ▶ Standstill monitoring (📖 537)
- The brake is activated, i.e. the "brake closing time" (C02589) and the "waiting time for status monitoring" (C02591) are elapsed.
- No other source for controller inhibit is active so that the controller inhibit can be deactivated by the brake control.

Sequence

If all requirements mentioned before have been met, the brake test can be started by setting the input *BRK_bBrakeTest* to TRUE.



A Normal behaviour
B Error behaviour
❶ C02594: Test torque
❷ C02595: Permissible angle of rotation
CINH = controller inhibit

[11-25] Sequence of the brake test

- The specified test torque is created via a ramp generator with an acceleration time of 1 s and held max. 4 s.
  - By this the motor shaft tries to rotate while the brake is applied.
- By resetting the input *BRK_bBrakeTest* to FALSE the brake test can be aborted.

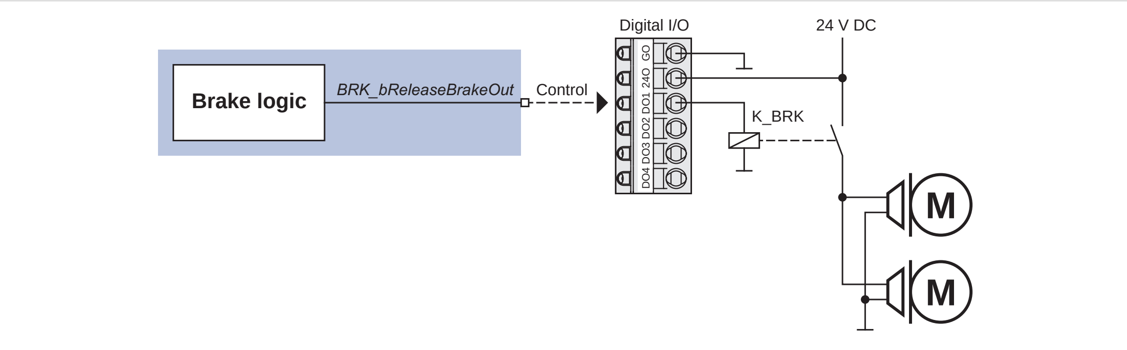
Error behaviour

If during the brake test the stop position of the motor axis has changed by more than the permissible angle of rotation set in C02595, although the brake is engaged:

- The brake test is cancelled immediately and "Quick stop by trouble" is activated as error response to avoid a further rotation/acceleration of the drive.
- The error message "Motor brake: Angular drift with closed brake too high" is entered into the logbook.
- The status "position drift when brake is applied" is displayed at the *BRK_dnState* status output via bit 21 and the status "brake error" is displayed via bit 15.
- The *BRK_bError* output is set to TRUE for one task cycle.

11.12.9 Control of two motor holding brakes

The technical implementation is based on the control of an external relay by a digital output. The relay contact then switches an external 24-V supply for both motor holding brakes:



[11-26] Interconnection example for controlling two motor holding brakes

Tip!

From software version V7.0 onwards, two motor temperature sensors can be evaluated simultaneously via the two encoder inputs X7 and X8 when two motor are used (e.g. double motor for a storage and retrieval unit).

▸ Temperature monitoring of a second motor (📖 232)

11.13 Cam data management

This function extension is available from software version V3.0!

The basic function "Cam data management" provides different functions for the systemwide management of the cam data available in the memory module for a cam application.

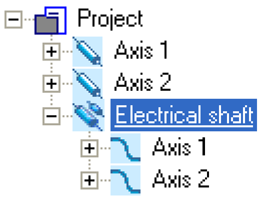
- Cam data are motion profiles/characteristics, cam tracks, and position marks.
- The cam data required can either be created using the »Cam Editor« and transmitted to the controller by means of the »Engineer«, or they can be directly entered via the parameters of this basic drive function if cam data have already been downloaded.

Note!

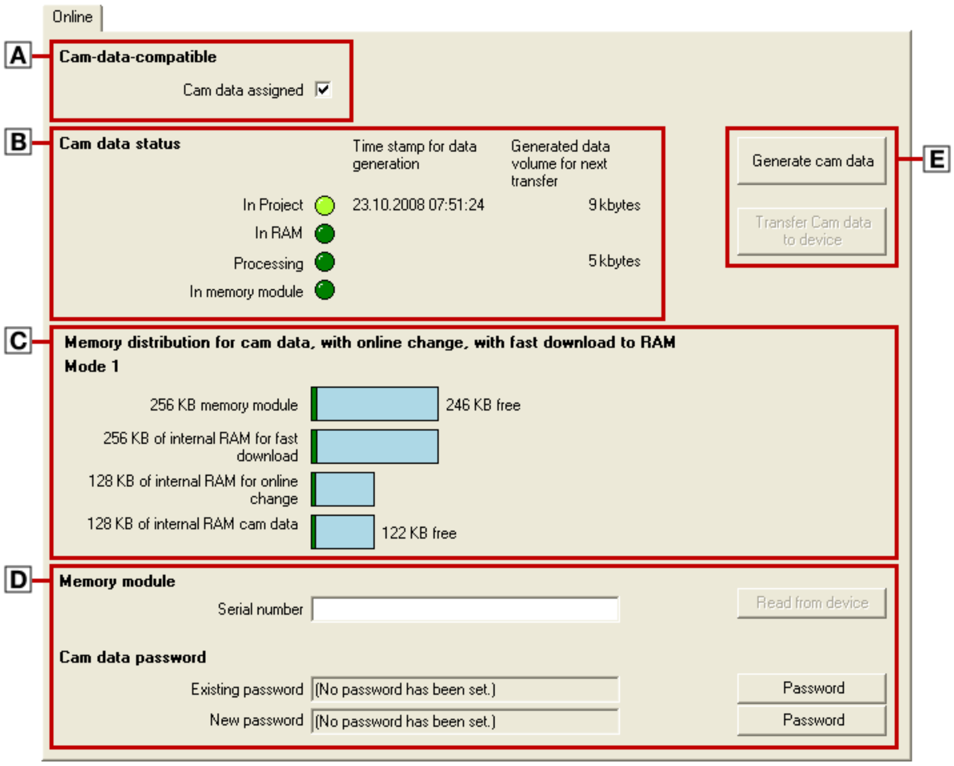
For the use of cam data in the controller, the licence level Motion Control TopLevel is required!

11.13.1 "Online" tab for cam data management

After an interconnection has been created via the electrical shaft, it is shown in the *Project view* with the axes assigned:



If you select the axis representing the 9400 HighLine controller under the electrical shaft, the **Online** tab for cam data management is provided in the *Workspace*:



Information	
A	Cam data-able • This property is set automatically for the controller if the controller supports cam data (licence level Motion Control TopLevel required). • **Note: The checkmark must not be removed, as otherwise the assignment to the cam data is lost!**
B	Cam data status • The first green LED for instance is lit if the cam data file in the project is active. • When an online connection to the controller has been established, also the status of the cam data available in the controller is shown.
C	Display of the current memory distribution for the cam data. ▸ Memory distribution (📖 565)
D	Configuration of the access protection for the cam data. ▸ Access protection (📖 566)
E	Possibility for quickly updating the cam data in the controller (without having to transfer the complete application). ▸ Regenerating the cam data file and transferring it to the controller (📖 568)

11.13.1.1 Memory distribution

From software version V5.0 there are three different storage modes for the distribution of the cam data memory in the controller. The specification of the storage mode is effected automatically by the »Engineer« when the cam data file is generated.

Note!

For controllers with a software version lower than V5.0 the memory distribution always corresponds to storage mode 1 (see following table), i. e. the max. size of the cam data file is limited to 256 kBytes.

Storage modes 2 and 3 automatically provide a greater memory for an extensive amount of cam data, however, in return certain functions are no longer supported (e. g. changing cam data via parameterisation).

Memory distribution of the cam data	Display in [bytes]	Storage mode		
		1	2*	3*
Memory module	-	262144 bytes (256 kBytes)	524288 bytes (512 kBytes)	1048576 bytes (1024 kBytes)
Internal RAM for quick download	C02901/1	262144 bytes (256 kBytes)	0 bytes	0 bytes
Internal RAM for online change	C02901/2	131072 bytes (128 kBytes)	262144 bytes (256 kBytes)	0 bytes
Internal RAM for cam data	C02901/3	131072 bytes (128 kBytes)	262144 bytes (256 kBytes)	524288 bytes (512 kBytes)

* Only from software version V5.0

Functions supported	Storage mode		
	1	2*	3*
Changing cam data via parameterisation	●		
Quick download to the RAM	●		
Online change	●	●	
Device command "Load cam data"	●	●	●
Device command "Save cam data"	●		
Device command "Calculate cam data"	●	●	●
Device command "Calculate cam data checksum"	●		

* Only from software version V5.0

11.13.1.2 Access protection

If required, the cam data can be protected against unauthorised or unintentional change by means of a three-stage access protection concept:

Step 1: Access protection deactivated

- There is no access protection for the upload/download of new cam data and the change of cam data via parameters.

Step 2: Access protection via user password

- The user password must be entered for the upload/download of new cam data and the change of password-protected cam data via parameters.

Step 3: Linking the cam data to the serial number of the memory module

- The user password must be entered for the upload/download of new cam data and the change of password-protected cam data via parameters.
- In addition, the serial number of the memory module must comply with the serial number given in the »Engineer« for the cam data.

Note!

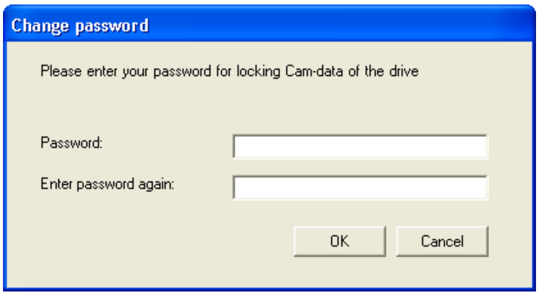
The settings for the access protection are firmly defined for the existing cam data and cannot be changed.

For a change of the settings the cam data have to be updated in the »Engineer« and then transferred to the controller. These two actions can be carried out on the **Online** tab via the buttons **Generate cam data file** and **Transfer cam data to the device**.

Regenerating the cam data file and transferring it to the controller (568)

How to define a password for the cam data:

1. Click the lower **Password** button (for new password).
   - The *Change password* dialog box appears:



2. Enter desired user password.
3. Click **OK** to accept the entry and close the dialog box.

How to change an existing password:

1. Click upper **Password** button (for existing password).
2. Enter the existing user password in the *Change password* dialog box.
3. Click **OK** to accept the entry and close the dialog box.
4. Click the lower **Password** button (for new password).
5. Enter the new user password in the *Change password* dialog box.
6. Click **OK** to accept the entry and close the dialog box.

Tip!

An existing access protection via user password can be cancelled again by carrying out the steps described before for changing the password and simply leaving blank the input field for the new password.

How to link the cam data to the serial number of the memory module:

Go to the **Serial number** input field and enter the serial number of the memory module.

- When an online connection to the controller has been established, you can read out the serial number of the memory module in the controller by clicking the **Read from device** button.

Tip!

The linkage of the cam data to the serial number of the memory module can be cancelled again by carrying out the steps described before and simply leaving blank the **Serial number** input field.

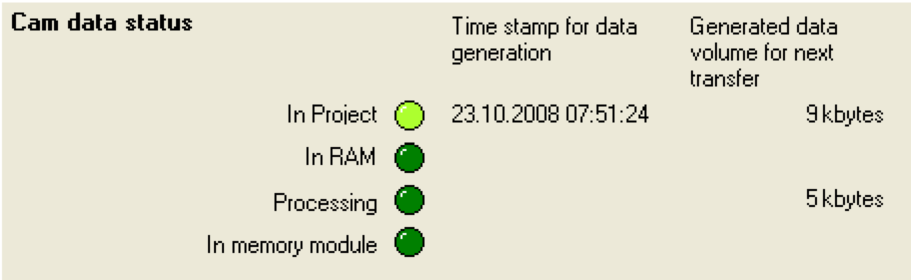
11.13.1.3 Regenerating the cam data file and transferring it to the controller

If you transfer the parameter set or the application from the »Engineer« to the controller, the cam data are also transferred automatically to the controller.

To only regenerate the cam data file and transferring it to the controller, after the cam data have been changed in the »Cam Manager« or the settings have been changed for access protection, carry out the following steps:

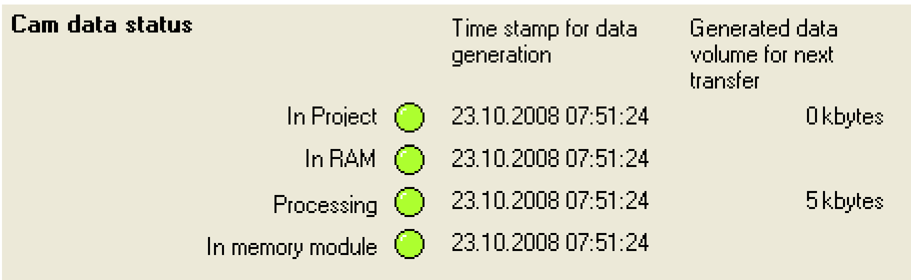
How to update the cam data:

1. Click on the **Generate cam data file** button on the **Online** tab to regenerate the cam data file for the controller.
   - The cam data status shown and the information with regard to the memory distribution on the **Online** tab are updated. The green LED behind "In the project" is lit now, which means that the cam data file in the project is active:



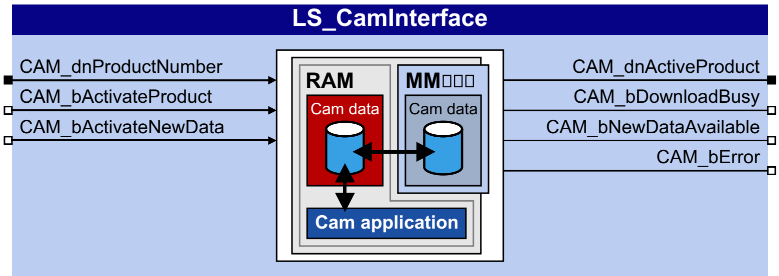
   When an online connection has been established to the controller:

2. To transmit the cam data to the controller, click the **Download cam data** button.
   - The new/altered cam data are accepted in the controller according to the online change mode set. ▸ Online change mode (573)
   - The green LED behind "In the memory module" is now lit as well, which means that the cam data file in the memory module is also active:



11.13.2 Internal interfaces | "LS_CamInterface" system block

The **LS_CamInterface** system block provides the internal interfaces for cam data management in the function block editor.



Inputs

Identifier DIS code | data type	Information/possible settings	
CAM_dnProductNumber DINT	Product number • The basic function manages the product number for all cam FBs within the application. • The product number is displayed in the »Cam Manager« in squared brackets after the product name. • If the product number is to be defined via parameter instead, a corresponding user code must be created in the application and connected with this input. • The highest product number to be created is shown in C02908.	
CAM_bActivateProduct BOOL	Activate product • The change-over to another product is caused by an event which is generated from the application.	
	TRUE	The product with the product number at input *CAM_dnProductNumber* is activated.
CAM_bActivateNewData BOOL	Reload cam data from the backup memory (controlled acceptance) • Only possible if the online change mode "10: Manual Activation" is set in C02905. • If the online change mode "16: Automatic activation with CINH" (Lenze setting) or "15: Automatic activation" is set in C02905, the new cam data are accepted immediately after a download and this input has no function. • The current status of the data acceptance is shown in C02906.	
	TRUE	Reload cam data from the backup memory.

Outputs

Identifier DIS code | data type	Value/meaning	
CAM_dnActiveProduct C02909 | DINT	Product number of the currently active product	
CAM_bDownloadBusy BOOL	Status signal "Download/data change active" • The current status of the data acceptance is shown in C02906.	
	TRUE	Currently the cam data in the RAM of the controller are changed. • For instance, due to parameter set transfer, device command C00002 = "501: Load cam data" or change of the cam data via parameters.
	TRUE↘FALSE	Download/data change completed. • In order to detect if the download/ the data change has been completed correctly, the *CAM_bError* error output should also be evaluated.

<table>
<tr><th>Identifier
DIS code | data type</th><th colspan="2">Value/meaning</th></tr>
<tr><td rowspan="3">CAM_bNewDataAvailable
BOOL</td><td colspan="2">Status signal "New cam data available"</td></tr>
<tr><td>TRUE</td><td>The internal recalculation of the new/changed cam data is completed and the cam data are ready for acceptance.
• The time when the new/altered cam data are accepted depends on the online change mode set in C02905. ▶ Online change mode (📖 573)
• In the Lenze setting the new/altered cam data are accepted automatically as soon as the controller inhibit is set within the controller.
• If the online change mode "15: Automatic activation" is set in C02905, the new/changed cam data are accepted immediately after the internal recalculation and the TRUE signal is only pending for one task cycle.</td></tr>
<tr><td>TRUE↘FALSE</td><td>The new/changed cam data have been accepted.</td></tr>
<tr><td rowspan="2">CAM_bError
BOOL</td><td colspan="2">"Fault" status signal"</td></tr>
<tr><td>TRUE</td><td>An error has occurred (group signal).</td></tr>
</table>

11.13.3 Parameter setting

Short overview of the parameters for cam data management:

Parameter	Information	Effective in storage mode		
		1	2*	3*
C00198	Axis number	●	●	●
C02900	User Password	●	●	●
C02901	Cam storage capacity	●	●	●
C02902	Time stamp of cam data	●	●	●
C02903	GUID cam data	●	●	●
C02905	Online change mode	●	●	
C02906	Online change status	●	●	●
C02908	Number of products	●	●	●
C02909	Active Product	●	●	●
C02910	Product Name	●	●	●
C02911	Product Choice	●	●	●
C02912	Number of products	●		
C02919	Number of curve tracks	●		
C02920	Cam Track Choice	●	●	●
C02921	Cam Track Type	●		
C02922	Number of Cam Data Points	●		
C02923	Cam Data Point Choice	●	●	●
C02924	Change Cam Data Point X	●		
C02925	Change Cam Data Point Y	●		
C02926	Change Cam Data Point M	●		
C02927	Auto Inc Cam Data Points	●	●	●
C02939	Number of Cont Tracks	●		
C02940	Cont Track Choice	●	●	●
C02941	Cont Type	●	●	●
C02942	Number of Cont Data Points	●		
C02943	Cont Data Point Choice	●	●	●
C02944	Cont Pos X0	●		
C02945	Cont Pos X1	●		
C02946	Cont Time	●		
C02959	Number of Position Tracks	●		
C02960	Pos Track Choice	●	●	●
C02962	Number of Pos Data Points	●		
C02963	Pos Data Point Choice	●	●	●
C02964	Change Pos Data Point X	●		
C02965	Change Pos Data Point Y	●		

Highlighted in grey = display parameter
* Storage modes 2 and 3 are only available from software version V5.0. Parameters that are not effective are set to zero.

11.13.3.1 Password entry

If a password has been defined for the cam data in the »Engineer«, the defined user password must be entered once to execute the following actions:

- Download of new cam data during operation
  → Entry of the existing password in the »Engineer«.
- Change of the cam data via parameter setting
  → Entry of the existing password in C02900.
- Loading/saving of the cam data
  → entry of the existing password in C02900.

Note!

From software version V4.0 onwards, you do not need to enter a possibly existing user password (C02900) if you want to save the cam data.

Tip!

The access protection for the cam data can be configured on the **Online** tab.
"Online" tab for cam data management (564)

Validity

The user password entered in C02900 is maintained until the next download, mains switching, or until reset by the user (logout).

- You can "log out" deliberately by entering an invalid password into C02900.

Behaviour in case of invalid entry

If the user password is entered incorrectly three times, the cam data are blocked. A correct entry resets the number of the failed attempts.

To remove the blocking of the cam data, there are two possibilities:

A. Resetting the parameters to the Lenze setting via device command C00002 = "0: Load Lenze setting".
   – When the Lenze setting is loaded, the cam data are deleted.
   – Afterwards the cam data can be transferred to the device again.

B. Transfer complete application to the device again.
   – The application available and the cam data are deleted and all data are transferred to the device again.

11.13.3.2 Online change mode

During running operation, new cam data can be downloaded from the »Engineer« to the controller if the controller is in storage mode 1 or 2.

- If the cam data are provided with an access protection, the user password has to be entered first. Access protection (566)
- The time when the new/altered cam data are accepted depends on the online change mode set in C02905.

Note!

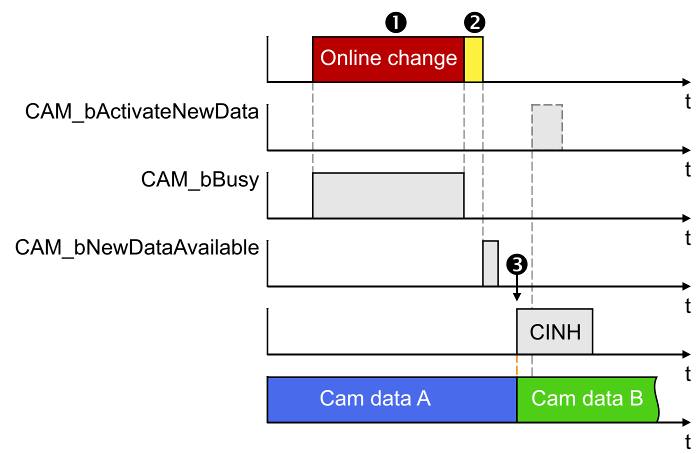
If the controller is in storage mode 3, the "Online change" function is deactivated:

- The online change mode set in C02905 is ineffective.
- In C02906 the status "999: Online change deactivated" is displayed.
- For the download of new cam data, controller inhibit is required.
- The cam data are accepted immediately after download.

Memory distribution (565)

Mode 16: Automatic activation with CINH

In the Lenze setting, the online change mode "Automatic activation with CINH" is set in C02905, i.e. the new cam data are accepted automatically as soon as the controller is inhibited.



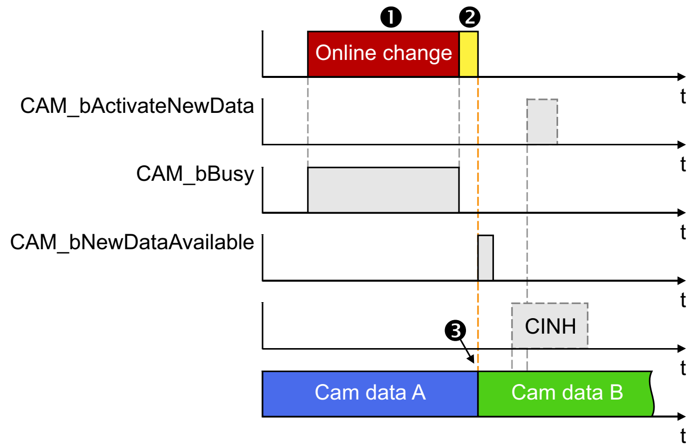
❶ Download/change of cam data active
❷ Internal recalculation of cam data active
❸ Automatic activation with controller inhibit

[11-27] Online change mode "Automatic activation with CINH"

Mode 15: Automatic activation

In the online change mode "Automatic activation", the new cam data are accepted directly after the internal recalculation of the data from the application unit [unit] into the internal unit [increments] has been completed.

▶ The controller does not need to be inhibited for acceptance.

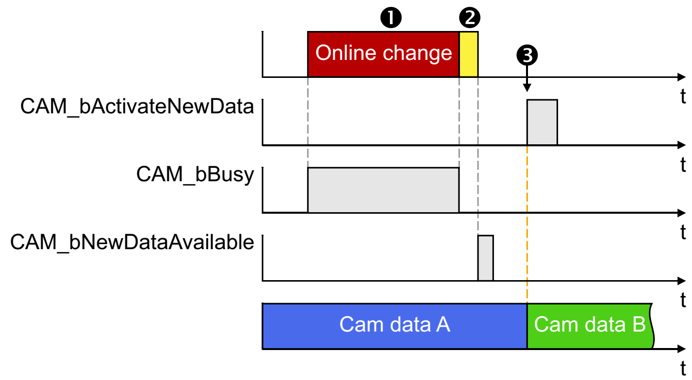


❶ Download/change of cam data active
❷ Internal recalculation of cam data active
❸ Automatic activation

[11-28] Online change mode "Automatic activation"

Mode 10: Manual activation

In the online change mode "Manual activation", the new cam data are accepted when the *CAM_bActivateNewData* control input is set to TRUE.



❶ Download/change of cam data active
❷ Internal recalculation of cam data active
❸ Manual activation

[11-29] Online change mode "Manual activation"

11.13.3.3 Changing cam data via parameterisation

If required, the cam data (motion profiles/characteristics, cams, and position markers) can be changed via corresponding parameters if the controller is in storage mode 1.

- If the cam data are provided with an access protection, the user password has to be entered in C02900 first. Access protection (566)
- From software version V4.0 onwards, the time stamp of the cam data is updated if the cam data are changed by parameter setting. This enables, for instance, the »Engineer» to recognise that the cam data of the »Engineer« project and those of the controller differ.

Note!

If the controller is in storage mode 2 or 3, the cam data cannot be changed via parameterisation. All parameters for change are ineffective and set to zero. Memory distribution (565)

If the cam data of the controller have been changed by parameter setting, the C00002 = "504: Calculate Cam Checksum" device command has to be executed afterwards. Calculate cam data checksum (100)

Then the cam data can be converted into the internal format with the "503: Calculate Cam Data" device command or saved on the memory module in a powerfail-proof manner with the "502: Save Cam Data" device command. Calculate cam data (99) / Save cam data (97)

From software version V4.0 onwards, the changed cam data and the parameters can be saved together on the memory module in a powerfail-proof manner with the C00002 = "11: Save start parameters" device command. Save start parameters (56)

How to change an interpolation point in a curve (motion profile or characteristic):

1. Go to C02911 and set the product number of the product to be edited.
2. Go to C02920 and set the track number of the curve track to be edited.

   Tip!

   In C02921 the curve type, and in C02922 the number of interpolation points of the curve selected is shown.
3. Set the interpolation point to be edited in C02923.
4. Change the desired parameters of the selected grid point:
   - C02924: x position
   - C02925: y position
   - C02926: Torque feedforward control value
     (only in case of a motion profile with feedforward control.)

Tip!

C02927 serves to activate a grid point auto increment if several successive grid points are to be changed.

- When the grid point auto increment is activated, it is automatically incremented to the next grid point every time the y position is written into C02925 so that the specification of the grid point to be changed in C02923 is only required once.

How to change several successive grid points (auto increment):

1. Go to C02911 and set the product number of the product to be edited.
2. Go to C02920 and set the track number of the curve track to be edited.

   Tip!

   In C02921 the curve type, and in C02922 the number of interpolation points of the curve selected is shown.
3. Set selection "1: Activate" in C02927 to activate the grid point auto increment.
4. Set the grid point from which on the grid point auto increment is to be started in C02923.
5. Set the following parameters for the grid point set in C02923 in the given order:
   - C02924: x position
   - C02926: Torque feedforward control value
     (only in case of a motion profile with feedforward control.)
   - C02925: y position

   After the y position is written into C02925 it is automatically incremented to the next grid point.
6. Set the parameters for the next grid point in the same order:
   - C02924: x position
   - C02926: Torque feedforward control value
     (only in case of a motion profile with feedforward control.)
   - C02925: y position
7. Repeat step 4 until all grid points are changed.

   Note: Do not change more grid points then available (depending on the start grid point). Changing a non-available grid point causes an error message!

How to change a cam:

1. Go to C02911 and set the product number of the product to be edited.
2. Go to C02940 and set the track number of the cam track to be edited.

   Tip!

   The cam type is displayed in C02941 and the number of cams of the selected cam data is displayed in C02942.
3. Go to C02943 and set the cam to be edited.
4. Change the desired parameters of the selected cam:
   - C02944: Cam position X0
   - C02945: Cam position X1
   - C02946: Cont Time (for position/time cams)

How to change a position mark:

1. Go to C02911 and set the product number of the product to be edited.
2. Go to C02960 and set the track number of the position track to be edited.

   Tip!

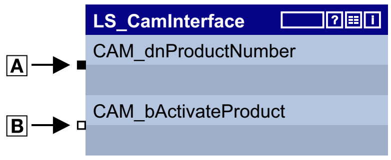
   In C02962 the number of the position marks of the position data selected is shown.
3. Set the position mark to be edited in C02963.
4. Change the desired parameters of the selected position mark:
   - C02964: x position
   - C02965: y position

11.13.4 Product/track change-over

Product change-over

The change-over to another product number is effected for all cam FBs within the application via the basic function "Cam data management".

- The change-over is effected via the input *CAM_bActivateProduct* on the basis of an event that is generated from the application.
- By setting the input *CAM_bActivateProduct* to TRUE, the product is activated with the product number at input *CAM_dnProductNumber*.



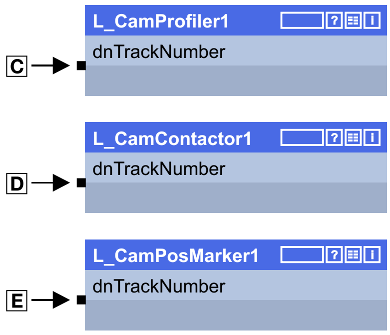
A Product number
B Activate product change-over

[11-30] Principle: Product change-over

Track change-over

The change-over to another curve track, cam track, or position track, however, is individually effected via the input *dnTrackNumber* at the respective cam function block.

- For the two FBs **L_CamProfiler** and **L_CamContactor** it can be parameterised whether the track change-over is to be effected in the next zero crossing of the x axis or immediately (Lenze setting: in the next zero crossing).



C Track number of the curve track
D Track number of the cam track
E Track number of the position track

[11-31] Principle: Track change-over

11.13.5 Invalid cam data due to changed machine parameters

This function extension is available from software version V4.0!

If one or more machine parameters affecting the internal scaling of the cam data are changed, the message "Cam Data: Invalidated due to change of mechanical data" (error number 0x00b80034) is entered into the logbook and the error response "Warning" occurs.

- The cam data are no longer valid and have to be recalculated.
- The warning is automatically reset if the C00002 = "503: Calculate Cam Data" device command is executed. Calculate cam data (99)

Machine parameters affecting the internal scaling of the cam data:

Parameter	Information
C00006	Selection of the motor control
C00100	Resolution of an encoder revolution
C02520 / C02521	Gearbox factor - motor (if motor = reference source)
C02522 / C02523	Gearbox factor - position encoder (if position encoder = reference source)
C02524	Feed constant
C02570	Position control structure

11.13.6 Behaviour after mains switching

After mains switching, the cam data are loaded from the memory module into the controller between loading and start of the application.

Note!

During the initialisation no check of the user password takes place, but a check of the serial number of the memory module is carried out, if this access protection has been activated by the user in the »Cam Designer«. If the serial number specified and the serial number of the memory module do not match, the cam data are not loaded.

- The "Cam data: serial number" error message is entered in the logbook.
- The "Warning locked" error response occurs.

If a download of cam data that was carried out before mains switching was not completed correctly, the previous cam data – if available – are loaded by the memory module.

- The "Cam data restored" error message is entered into the logbook.
- The "Fault" error response occurs.
- After the error is reset (acknowledged), operation with the previous cam data is possible.

11.14 Pole position identification

This function extension is available from software version V7.0!

The device commands "Identify pole position (360°)" and "Identify pole position (min. motion)" serve to execute an identification of pole position to detect the pole position to the motor encoder currently activated in C00495.

From software version V7.0 onwards, the identification of pole position is additionally available as a basic function in the form of the LS_PolePositionIdentification system block.

Note!

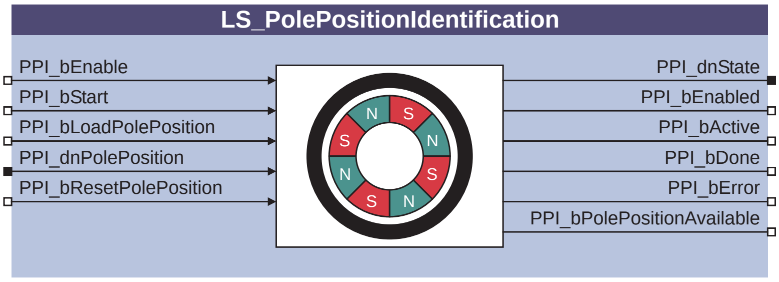
An identification of pole position is only required:

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

Detailed information on the identification of pole position can be found in the subchapter "Pole position identification" for the motor interface. (📖 136)

11.14.1 Internal interfaces | System block "LS_PolePositionIdentification"

The **LS_PolePositionIdentification** system block provides the internal interfaces for the basic function "identification of pole position" in the function block editor.



Note!

Ensure that the system block is called in a cyclic application task.

Basically, projects which only contain an unsolicited task and no cyclic task are not permissible!

Inputs

Identifier DIS code | data type	Information/possible settings	
PPI_bEnable C02789/1 | BOOL	Requesting control via the basic function.	
	TRUE	If no other basic function is active, a change-over to the "Identification of pole position active" function state is effected and an identification of pole position can be carried out via the control inputs.
	TRUE↘FALSE	An active identification of pole position is stopped, i.e. a change-over from the active "Identification of pole position active" function state back to the "Controller not ready" basic state is effected.
PPI_bStart C02789/2 | BOOL	Start pole position identification	
	FALSE↗TRUE	Pole position identification is started in the mode selected in C02786.
PPI_bLoadPolePosition C02789/3 | BOOL	Start pole position identification	
	FALSE↗TRUE	The pole position angle applied at *PPI_dnPolePosition* is accepted in C00058/x. • The subcode to be described of C00058 depends on the motor encoder selected in C00495.
PPI_dnPolePosition C02788 | DINT	Pole position angle in [°] with one decimal position • Value range: -179.9 ... +179.9 °	
PPI_bResetPolePosition C02789/4 | BOOL	Reset "Pole position known" status	
	FALSE↗TRUE	The status outputs *PPI_bDone* and *PPI_bPolePositionAvailable* are reset to FALSE.

Outputs

Identifier DIS code | data type	Value/meaning	
Ppi_dnState C02787 | DINT	Status (bit coded) • When the basic function is not enabled, all bits are set to "0". • Bits which are not listed are not assigned with a status (always "0").	
	Bit 1	Pole position identification active.
	Bit 2	Pole position identification completed.
	Bit 14	Pole position known.
	Bit 15	An error has occurred (group signal).
PPI_bEnabled C02789/5 | BOOL	Status signal "Basic function is enabled"	
	TRUE	Pole position identification vie the control inputs is possible. • The *PPI_bEnable* enable input is set to TRUE and the controller is in the "Pole position identification active" function state.
PPI_bActive C02789/6 | BOOL	Status signal "Basic function is active"	
	TRUE	Pole position identification is active. • Output is reset to FALSE if the *PPI_bStart* is reset to FALSE, the controller enable is deactivated or an error has occurred.
PPI_bDone C02789/7 | BOOL	Status signal "Basic function is ready"	
	TRUE	Pole position identification is completed. • Output is reset to FALSE when input *PPI_bStart* is reset to FALSE.
PPI_bError C02789/8 | BOOL	"Fault" status signal	
	TRUE	An error has occurred (group signal).
PPI_bPolePositionAvailable BOOL	Status signal "Pole position is known"	
	TRUE	The drive knows the pole position. • The value written into C00058/x corresponds to the pole position (x depends on the motor encoder selected in C00422).

11.14.2 Parameter setting

- Parameterisation dialog in the »Engineer«: Tab **Application parameters** → Dialog level *Overview* → *All basic functions* → *Pole position identification*
- Short overview of parameters for pole position identification:

Parameter	Information
C02785	Activation of PPI
C02786	Mode of PPI
C02787	Ppi_dnState
C02788	PolePosition Setpoint
C02789	PolePositionIdentification: Dig. signals
Highlighted in grey = display parameter	

11.14.3 Execute pole position identification

Requirements

- The controller inhibit is active.
- The controller has the "Controller not ready" function state.
- The basic function "Pole position identification" is part of the active application.
- No other basic function is active.

Activate basic function

To request the control via the basic function, the *PPI_bEnable* enable input in the application must be set to TRUE.

- If no other basic function is active, a change-over to the "Identification of pole position active" function state is effected and an identification of pole position can be carried out via the control inputs.
- A successful change to the function state "Pole position identification" is displayed by a TRUE signal at the *PPI_bEnabled* status output.

Start pole position identification

Danger!

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

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

Stop!

Check the correct parameterisation of the max. motor current monitoring (C00619 and C00620) before carrying out the pole position identification to prevent the motor from being permanently damaged.

By setting the *PPI_bStart* control input to TRUE, the pole position identification is started in the mode selected in C02786.

- The procedure starts with controller enable, if
  - a synchronous machine is selected,
  - no other identification is active,
  - no error has occurred, and
  - no test mode is activated.
- If one of the above conditions is not met, the procedure is cancelled and the corresponding device command status is indicated via *PPI_dnState*.

Note!

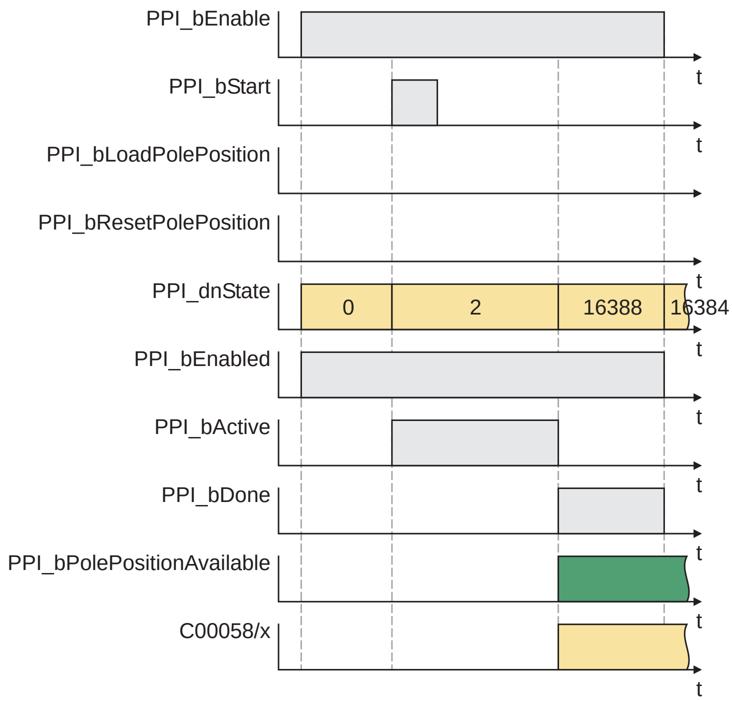
Detailed information on the identification of pole position can be found in the subchapter "Pole position identification" for the motor interface. (📖 136)

Deactivation

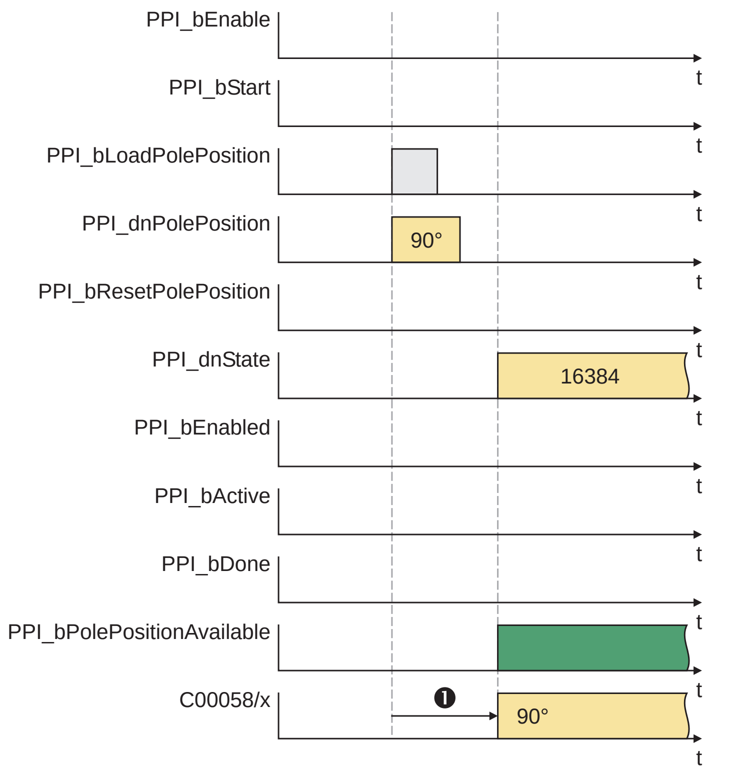
When the *PPI_bEnable* enable input is reset to FALSE, an active pole position identification is stopped.

- If the pole position identification is aborted, no change is made in C00058/x.
- The *PPI_bEnabled* status output is reset to FALSE and a change-over from the active "Pole position identification active" function state back to the basic "Controller not ready" state is carried out.

11.14.4 Signal characteristics

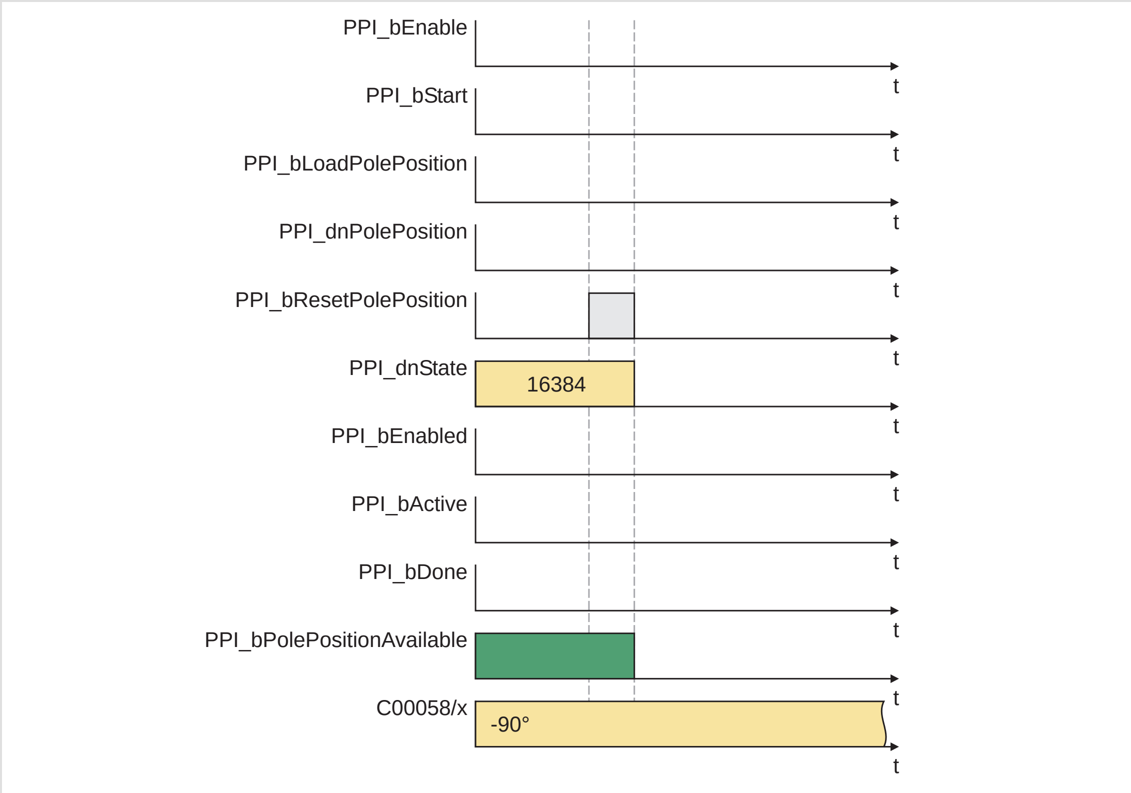


[11-32] Signal characteristic 1: Normal procedure of the pole position identification



➊ Time - writing parameters

[11-33] Signal characteristic 2: Load pole position



[11-34] Signal characteristic 3: Reset pole position

11.14.5 Impacts of parameter changes on the signal PPI_bPolePositionAvailable

Changing the type of motor control (C00006)

Initial situation 1	Parameter change	Impact
• *PPI_bPolePositionAvailable* = TRUE • Motor control (C00006) = "1: SC: Servo control sync. motor"	In C00006 another motor control is set.	*PPI_bPolePositionAvailable* remains on TRUE since the signal is irrelevant for an asynchronous motor.

Initial situation 2	Parameter change	Impact
• *PPI_bPolePositionAvailable* = TRUE • Motor control (C00006) = – "2: SC: Servo control async. motor" or – "4: SLVC: Sensorless vector control" or – "6: VFCplus: V/f control" or – "7: VFCplus: V/f control"	In C00006, the motor control "1: SC: Servo control sync. motor" is set.	*PPI_bPolePositionAvailable* is reset to FALSE since the signal is relevant for a synchronous motor and apparently a new motor has been selected.

Changing relevant motor data

Initial situation	Parameter change	Impact
PPI_bPolePositionAvailable = TRUE	One of the following parameters is changed: • Rated motor speed (C00087) • Rated motor frequency (C00089) • Rated motor voltage (C00090)	*PPI_bPolePositionAvailable* is reset to FALSE since it is assumed that the connected motor has been changed.

Changing relevant encoder data

Initial situation	Parameter change	Impact
PPI_bPolePositionAvailable = TRUE	One of the following parameters is changed: • Resolver - number of pole pairs (C00080) • Encoder - number of increments (C00420) • Encoder type (C00422) • TTL encoder signal evaluation (C00427) • Motor encoder selection (C00495)	*PPI_bPolePositionAvailable* is reset to FALSE.

Behaviour after mains ON

Initial situation 1	Behaviour after mains ON
• *PPI_bPolePositionAvailable* = TRUE • Motor control (C00006) = "1: SC: Servo control sync. motor" • Resolver - number of pole pairs (C00080) > 1	*PPI_bPolePositionAvailable* is only reset to FALSE if: • Motor encoder selection (C00495) = "0: Resolver to X7" **AND** • number of motor pole pairs are (C00059) no integer multiple of the number of resolver pole pairs (C00080).
Examples:	
MSC motor Motor - number of pole pairs = 4 Resolver - number of pole pairs = 1	The *PPI_bPolePositionAvailable* output is not reset to FALSE.
Kuka motor Motor - number of pole pairs = 4 Resolver - number of pole pairs = 4	
Torque motor Motor - number of pole pairs = 12 Resolver - number of pole pairs = 6	

Initial situation 2	Behaviour after mains ON
• *PPI_bPolePositionAvailable* = TRUE • Motor control (C00006) = "1: SC: Servo control sync. motor" • Encoder type (C00422) = – "0: Incremental encoder (TTL signal)" or – "1: Sine/Cosine encoder" • Motor encoder selection (C00495) = "1: Encoder to X8"	*PPI_bPolePositionAvailable* is reset to FALSE.

Behaviour after encoder error

Initial situation	Behaviour after encoder error
• *PPI_bPolePositionAvailable* = TRUE • Motor control (C00006) = "1: SC: Servo control sync. motor" • Encoder type (C00422) = – "0: Incremental encoder (TTL signal)" or – "1: Sine/Cosine encoder" or • Motor encoder selection (C00495) = "1: Encoder to X8"	*PPI_bPolePositionAvailable* is reset to FALSE.

Behaviour after resolver error

Initial situation	Behaviour after resolver error
• *PPI_bPolePositionAvailable* = TRUE • Motor control (C00006) = "1: SC: Servo control sync. motor" • The number of motor pole pairs are (C00059no integer multiple of the number of resolver pole pairs (C00080). • Motor encoder selection (C00495) = "0: Resolver to X7"	*PPI_bPolePositionAvailable* is reset to FALSE.

12 Oscilloscope

The oscilloscope function integrated in the controller can be used as support for commissioning, maintenance, and troubleshooting.

Typical applications

- Graphical representation of any measured values (e.g. speed setpoint, actual speed, and torque)
- Detection of process values without additional measuring instruments (e.g. oscilloscope, voltmeter, and ammeter)
- Convenient documentation for fine tuning of control circuits or parameter changes of the controller
- Documentation of production quality in the context of product liability and quality assurance

Special features

- Recording and saving measured values in the controller
- Measuring on eight independent channels at the same time
- Measuring fast and slow signals by adjustable sample rate
- Triggering on channel, variable, or error message
- Detecting measured values before and after a trigger event (pre-/post-trigger)
- Graphical representation and evaluation of measured values on a PC
- Cursor and zoom function for the measurement analysis
- Saving & loading of oscilloscope configurations
- Exporting measured values via the clipboard for further processing

12.1 Technical data

Servo Drives 9400	
Memory depth	Max. 16384 measured values, depending on the number of channels and the size of the variables to be recorded.
Memory capacity	32768 bytes
Data width of a channel	1 ... 4 bytes, corresponding to the size of the variables to be recorded
Number of channels	1 ... 8
Trigger level	Corresponding to the value range of the variable to be triggered
Trigger selection	Immediate triggering, rising/falling edge, signal change
Trigger delay	-100 % ... +400 %
Trigger source	Channels 1 ... 8, any variable or error message
Max. time base	8 channels 32 bits each ≡ 26 hours
Max. recording time	8 channels 32 bits each ≡ 10 days

12.2 Functional description

With an online connection to the controller, use the oscilloscope user interface of the »Engineer« to set the trigger condition and the sample rate and select the variables to be recorded.

When the measurement is started, the set parameters are transferred to the controller and checked. If invalid settings are found, the oscilloscope sends an error. Otherwise, the measurement is started.

With an online connection, the measured controller data are transferred to the »Engineer« and graphically represented on the oscilloscope user interface as soon as the measurement has been completed.

Recording variable values

The operating system scans the oscilloscope in fixed 1 ms-cycles, i.e. the oscilloscope can record variable values with a sample rate of max. 1 kHz.

Recording system variable values of the internal motor control

Unlike the variables declared in the application, the system variables of the internal motor control (MCTRL) can also be recorded with a sample rate higher than 1 kHz.

Note!

In the servo operating modes, recordings with a time-based resolution of 31.25 µs and 125 µs are not possible!

In comparison to all other variable values, the system variables of the internal motor control have a cycle offset of 2 ... 3 ms!

For recording with sample rates of 1 kHz, up to three measured values are missing at the left and right edges of the oscilloscope screen. This can only be noticed for very small time-based deflections ≤ 1 ms / div. The missing measured values are system-dependent and caused by oversampling.

Recording I/O variable values referring to a task

When selecting the variables to be measured, it is possible to indicate a task reference for the variables. If the behaviour of a certain task is to be examined, you can indicate the task reference to exactly record the values valid for the task run-time.

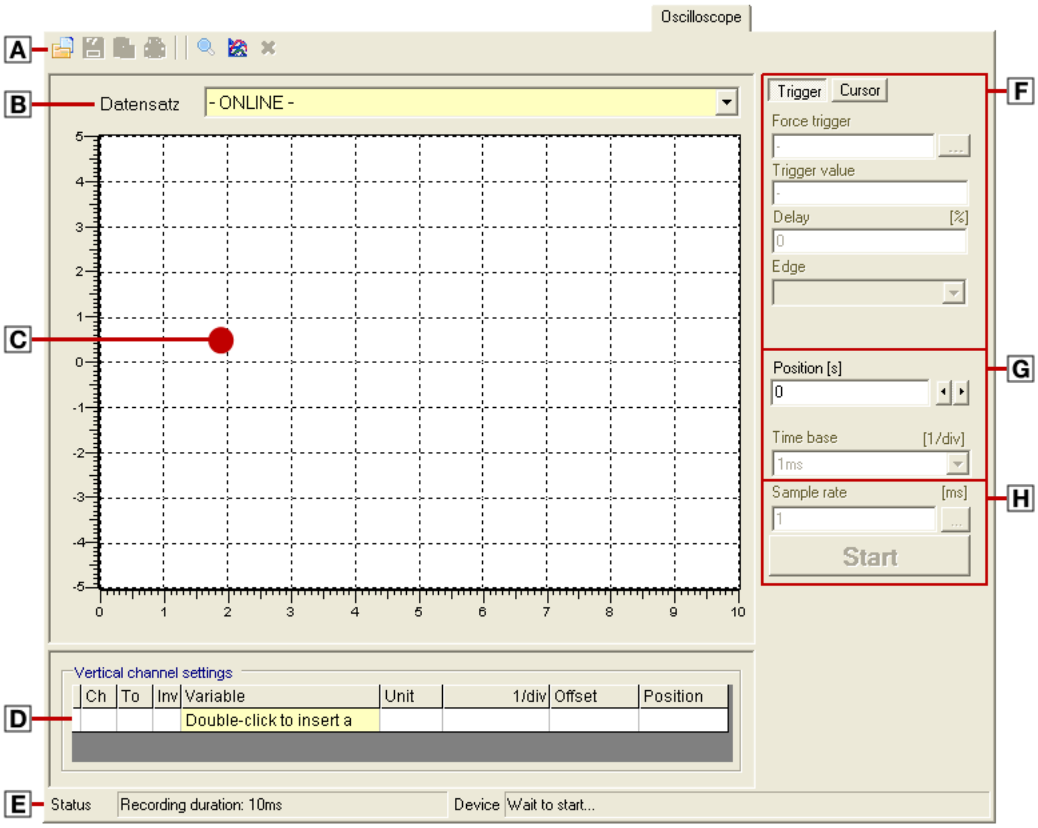
- The variable values are recorded during the process output image generation of the task.

12.3 User interface

How to go to the oscilloscope user interface:

1. Select the controller in the *Project view*.
2. Select the **Oscilloscope** tab from the *Workspace*.

The oscilloscope user interface contains the following control and function elements:



A Oscilloscope toolbar	F Trigger/cursor settings
B Data record selection	G Horizontal settings
C Oscillograph	H Recording settings
D Vertical settings	I Comment
E Status bar	

12.3.1 Oscilloscope toolbar

Icon	Function
	Loading/importing a data record (🕮 605)
	Close data record (🕮 606)
	Saving/exporting a data record (🕮 604)
	Copying a data record to the clipboard (🕮 607)
	Deleting a data record in the project (🕮 606)
	Print oscillogram
	Activate zoom function ▸ Adjusting the representation (🕮 600)
	Activate automatic scaling function ▸ Adjusting the representation (🕮 600)
	Comment data record (🕮 603)

12.3.2 Oscillograph

The oscillograph is used to visualise data records.

- Use the zoom and the automatic scaling function to adjust the representation.
- The measured data represented in the form of interpolated curves can be optionally shown and hidden, represented in any colour or overlaid with the signal characteristics of other variables recorded.

12.3.3 Vertical settings

Use the **Vertical** list field to configure the variables to be recorded.

▶ Simply click into a field to alter the corresponding setting.

Column	Name	Meaning
1	-	Curve colour for representation in the oscillograph
2	Ch	Channel number
3	On	On/off
4	Inv	Inversion on/off
5	Variable	Selection of variable to be recorded
6	Unit	Scaling
7	1/div	Vertical scale factor
8	Offset	Offset value • The offset value depends on the scale factor and is marked by a dotted line in curve colour at the left edge of the oscillograph.
9	Position	Position value • The position value is independent of the scale factor and is marked by a line at the left edge of the oscillograph.

12.3.4 Status bar

The status messages are displayed in the status bar.

12.3.5 Trigger/cursor settings

Trigger

If the **Trigger** button has been pressed, the input fields for configuring the trigger condition are shown. ▸ Selecting the trigger condition (📖 598)

Cursor

If the **Cursor** tab has been pressed instead, you can use a vertical measuring line to read individual measured values of a selectable channel in the oscilloscope. Using a second vertical measuring line, it is possible to indicate the difference between two measured values. ▸ Reading individual measured values (📖 601)

Group box	Meaning
Channel	Channel selection
Value	Display of the value measured at the position of the active measuring line
Difference	Display of the difference between the values measured at the two measuring lines

12.3.6 Horizontal settings

Use the **Horizontal** group box to select the time base and the horizontal position.
▸ Selecting the recording time/sample rate (📖 597)

Input field	Meaning
Time base [1/Div]	Selection of time base • The current time base setting multiplied by ten results in the recording time. • Change the time base to stretch or compress measurements that have already been completed.
Position	Selection of the horizontal display position • The position value can be directly entered into the input field or selected using the arrow buttons. • When the arrow buttons are used and the **<Ctrl>** key is pressed, you can increase the step size to accelerate the shift.

12.3.7 Recording settings

Use the **Recording** group box to select the sample rate and start recording.

▶ Since the measured data memory has a limited capacity, a compromise must be found between the sample rate and the recording time that results from the time base setting. ▸ Selecting the recording time/sample rate (📖 597)

12.4 Operation

This chapter describes step-by-step how to record the signal characteristics of controller variables and represent, analyse, document and process them in the oscilloscope.

Note!

In the oscilloscope, settings can only be selected and recording can only be started, when the controller is connected online.

12.4.1 Selecting the variables to be recorded

The oscilloscope supports up to eight channels, i.e. max. eight variables can be recorded in a data record. Use the **Vertical** group box to select the variables.

How to select a variable for recording:

1. Go to the **Variable** column in the **Vertical** group box and double-click the yellow field to open the *Select variable* dialog box.
2. Select the variable to be recorded from the list field.
3. Click **OK**.
   - The dialog box is closed and the selection is accepted.
4. Repeat steps 1 ... 3 to select up to seven more variables to be recorded.

Of course, it is possible at any time to change or delete a selection later.

How to change a selection:

1. Go to the **Vertical** group box and double-click the variable to be changed in the **Variable** column.
2. Select a new variable in the *Select variable* dialog box.
3. Click **OK**.
   - The dialog box is closed and the selection is accepted.

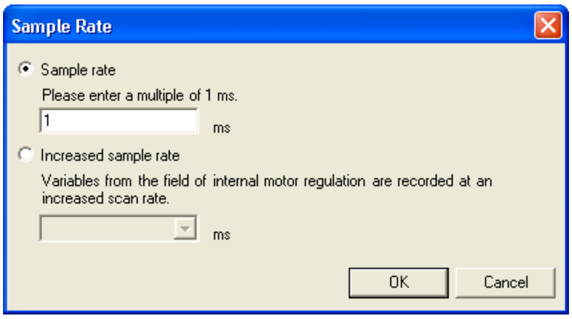
How to delete a selection:

1. Go to the **Vertical** group box and double-click the variable to be removed in the **Variable** column.
2. Click **Delete channel** in the *Select variable* dialog box.
   - The dialog box is closed and the selection is deleted.

12.4.2 Selecting the recording time/sample rate

How to select the recording time and the sample rate:

1. Go to the **Horizontal** group box and select the desired time base from the **Time base** list field.
   - The current time base setting multiplied by ten results in the recording time.
   - Since the measured data memory of the controller has a limited capacity, usually a compromise is made between sample rate and recording time.
2. Enter the desired sampling rate in [ms] in the **Sampling rate** input field of the **Recording** group field.
   - By pressing the ... button behind the **Sampling rate** input field, you open the *Sampling rate* dialog box where you can also select the option **Increased sampling rate**:



Note!

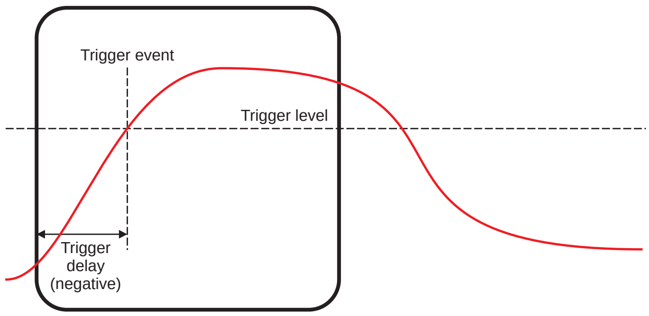
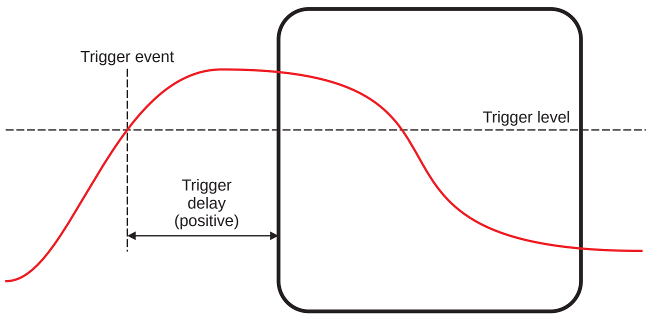
When the option **Increased sampling rate** is selected, only integer multiples of 1/(sampling rate in ms) are detected by the system.

- Since a complete representation in the oscillograph requires (10 * 1/(sample rate in ms) * (horizontal resolution in ms/div)) + 1 measured values, but due to system-dependent reasons only integer multiples of (1 / (sample rate in ms)) can be recorded, 1 to 3 measured values may be missing at the left or right edges of the oscillograph. The displayed curve then ends before the end or starts after the start of the oscillograph. The curve is not compressed or extended.
- The curve that is recorded is not extended or compressed.

12.4.3 Selecting the trigger condition

The trigger condition serves to define the starting time of recording in the controller. The oscilloscope provides various trigger conditions by means of which recording of the measured values can be controlled.

If the cursor display is activated, press the **Trigger** button to show the input fields for configuring the trigger condition.

Setting	Function
Source	Selection of trigger source:
Variable	The oscilloscope triggers on any variable of the PLC program. • Unlike triggering on a channel, triggering on a variable requires no recording channel.
Channel	The oscilloscope triggers on a channel configured in the **Vertical** table.
System event	Triggering is started on occurrence of a selectable controller event (e.g. TRIP, trouble or warning). • Select a negative trigger delay to record signals prior to occurrence of the event.
Force trigger	No trigger condition, recording starts immediately after the start.
Trigger value	Value from which on triggering is activated. • The trigger level is not effective for triggering on Boolean variables.
Delay	Delay between recording and trigger event.
Pre-trigger	Select a negative delay time to detect signals **prior to** the trigger event.  • In the oscillograph, the trigger time is marked by a dotted line. • When triggering on occurrence of an event, it is thus possible to detect the values that have caused the event.
Post-trigger	Select a positive delay time to detect signals occurring a certain time **after** the trigger event. 
Edge	If the trigger source is a channel or a variable, you can choose between the following trigger modes:
Positive edge	For triggering on a BOOL variable: • Trigger activation requires a FALSE-TRUE transition. For triggering on a different variable: • For trigger activation, the selected trigger value must be exceeded.

Setting	Function
Negative edge	For triggering on a BOOL variable: • Trigger activation requires a TRUE-FALSE transition. For triggering on a different variable: • For trigger activation, the selected trigger value must be fallen below.
Change	For triggering on a BOOL variable: • Trigger activation requires a state change. For triggering on a different variable: • For trigger activation, the current value must be different than the last value.

12.4.4 Starting recording

Click **Start** to start recording.

To obtain a maximum sampling rate when recording the variable values, the data are first stored in the measured data memory of the controller and then transferred as a data record to the PC.

▶ The current recording status is displayed in the status bar.

12.4.5 Adjusting the representation

After the variable values have been recorded and the online data record has been transferred to the PC, the data record is visualised in the oscillograph. If required, the representation can now be adjusted by using the zoom or the automatic scaling function.

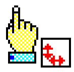
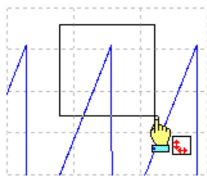
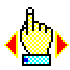
Tip!

From the »Engineer« V2.10 onwards, a scrollbar will appear below the time axis as soon as the complete measurement does is not displayed anymore in the oscillograph.

The scrollbar serves to move the visible cutout horizontally. The labelling of the time axis and the position display are automatically updated when the cutout is moved.

Zoom function

Go to the *oscilloscope toolbar* and click the icon to activate the zoom function.

Zoom function	Procedure	
Zoom selection		Hold down the left mouse button and draw the oscilloscope section to be zoomed:  • The selection is shown with a frame. • When the left mouse button is released, the selection is zoomed in the oscilloscope.
Horizontal/vertical shift of selection		Hold down the left and right mouse button and move the mouse pointer on the horizontal scale to the left or right or on the vertical scale to the top or bottom to shift the selection accordingly. • With a three-button mouse, you can use the mouse button in the middle.
Horizontal stretching		Hold down the left mouse button and move the mouse pointer on the horizontal scale to the left to stretch the shown selection from the right edge. • Moving the mouse pointer in opposite direction continuously reduces the stretching.
		Hold down the right mouse button and move the mouse pointer on the horizontal scale to the right to stretch the shown selection from the left edge. • Moving the mouse pointer in opposite direction continuously reduces the stretching.

Zoom function	Procedure	
Vertical stretching		Hold down the left mouse button and move the mouse pointer on the vertical scale to the bottom to stretch the shown selection from the top. • Moving the mouse pointer in opposite direction continuously reduces the stretching.
		Hold down the right mouse button and move the mouse pointer on the vertical scale to the top to stretch the shown selection from the bottom. • Moving the mouse pointer in opposite direction continuously reduces the stretching.
Return to original representation		Click the right mouse button in the oscillograph to return step by step to the original representation.

Automatic scaling function

Use the automatic scaling function to automatically scale and reposition the representation of selectable signal characteristics in the oscillograph and reset the offset to "0".

How to carry out automatic scaling:

1. Go to the *oscilloscope toolbar* and click the icon to activate the automatic scaling function.
2. Select the channels/variables for automatic scaling in the *Select variable* dialog box.
3. Click **OK**.
   - The dialog box is closed and the selected channels/variables are scaled automatically.

12.4.6 Reading individual measured values

In addition to the zoom and scaling function, the oscilloscope offers a "cursor function" that can be used to display individual measured values of a selectable channel or the difference between two measured values.

▶ If the **Cursor** button has been pressed, the cursor function is active and two movable vertical measuring lines are shown in the oscillograph.

How to use the cursor function:

1. Press the **Cursor** button.
2. Select the channel for which individual measured values are to be indicated from the **Channel** list field.

3. Hold down the left mouse button and drag the red vertical measuring line to the desired position.
    - The active measuring line is represented by a continuous line, the inactive measuring line is represented by a dotted line.
    - If you position the mouse pointer over the inactive measuring line, the measuring line automatically becomes active.
    - The value measured at the position of the active measuring line is indicated in the **Value** group box.
    - The difference between the values measured at the two measuring lines is indicated in the **Difference** group box.

12.4.7 Compare peak values

This function is available from the »Engineer« V2.10 onwards!

Several values displayed in the oscillograph can be compared by means of a horizontal measurement line.

- The horizontal measurement line is displayed if the **Cursor** button is pressed.
- The measurement line is automatically generated based on the current cursor position and thus cannot be moved separately.

12.5 Data records

If several data records are loaded in the oscilloscope at the same time, the data record to be displayed is selected via the **Data record** list field. There are three types of data records:

Online data record

The online data record is the only data record which serves to establish a connection to the target system. When the online data record has established the connection completely, it can communicate with the target system.

- In the **Data record** list field the online data record is marked with a prefixed asterisk (*).

Offline data record

The offline data record is a data record already stored in the project and loaded in the oscilloscope again, or a data record imported from a file.

Merge data record

The merge data record is automatically available in the **Data record** list field if two or more data records are loaded in the oscilloscope at the same time.

- In the merge data record, several characteristics from the currently loaded data records can overlaid, e.g. to compare signal characteristics from different recordings. ▶ Overlay function (📖 607)

12.5.1 Comment data record

This function is available from the »Engineer« V2.10 onwards!

You can add a comment to the selected data record.

- The comment is saved together with the data record, both when saving in a file and in a project.
- If you select a data record for loading in the *Load data record* dialog box that contains a comment, this comment is shown in the dialog box.

How to add a comment to the selected data record:

Option 1: Enter the comment directly into the **Comment** text field.

Option 2: Go to the *Data logger toolbar* and click the symbol to open the *Comment* dialog box for entering the comment.

12.5.2 Saving/exporting a data record

After the variables to be recorded have been selected and the required settings have been entered, you can save the configuration and recording, if already executed, for future use in the project or export them to a file.

Note!

The reuse of a saved configuration is only reasonable for controllers of the same type, as otherwise due to a scaling of the oscilloscope channels that is not adapted, incorrect values are displayed!

How to save the data record in the project:

1. Click on the icon in the *Oscilloscope toolbar*.
   - The *Save data record* dialog box appears.
2. Enter a name in the **Name of the data record to be stored** input field.
3. Click **Filing in the project**.
   - The dialog box is closed and the current data record is filed in the project.

Note!

The data record is only saved if the entire project is saved!

How to export the data record to a file:

1. Click on the icon in the *Oscilloscope toolbar*.
   - The *Save data record* dialog box appears.
2. Enter a name in the **Name of the data record to be stored** input field.
3. Click the **Export to file** button.
4. Specify the data record to be stored and the filing folder in the *Save as* dialog box.
5. Click **Save**.
   - The dialog box is closed and the current data record is saved.

12.5.3 Loading/importing a data record

Configurations/data records already stored can be reloaded into the oscilloscope any time, e.g. for the overlay function.

Note!

The reuse of a saved configuration is only reasonable for controllers of the same type, as otherwise due to a scaling of the oscilloscope channels that is not adapted, incorrect values are displayed!

How to load a data record from the project:

1. Click on the icon in the *Oscilloscope toolbar*.
   - The *Load data record* dialog box appears.
2. Select the data record to be loaded from the **Data record** list field.
3. If the data record is to be used as configuration, select the option **as configuration...**.
4. Click **Open**.
   - The dialog box is closed and the selected data record or the configuration is loaded.
   - If the configuration to be loaded contains variables that are no longer available in the controller, these variables are automatically removed from the configuration.

How to import a data record from a file:

1. Click on the icon in the *Oscilloscope toolbar*.
   - The *Load data record* dialog box appears.
2. Select the option **from a file**.
3. Click the **Search...** button.
4. Select the file to be imported within the desktop environment from the *Open* dialog box.
5. If the data record is to be used as configuration, select the option **as configuration...**.
6. Click **Open**.
   - The dialog box is closed and the selected data record or the configuration is imported.
   - If the configuration to be loaded contains variables that are no longer available in the controller, these variables are automatically removed from the configuration.

12.5.4 Close data record

This function is available from the »Engineer« V2.10 onwards!

You can close an open offline data record at any time.

- After a data record is closed, it is not available anymore in the **Data record** list field. The oscilloscope changes automatically to the display of the next offline data record in the list field.
- If the closed data record was included in the MERGE data record, its channels will be removed from the MERGE data record.
- If all offline data records have been closed, the oscilloscope changes to the online data record if the device is currently online. Otherwise no data record will be displayed.

Go to the *Oscilloscope toolbar* and click the symbol to close the currently displayed offline data record.

12.5.5 Deleting a data record in the project

How to delete the currently displayed offline data record:

1. Click on the icon in the *Oscilloscope toolbar*.
   - You will be prompted to confirm if the data record should really be deleted.
2. Confirm the prompt with **Yes** to delete the data record.
   - The data record is closed and then deleted in the project.
   - After a data record is closed, it is not available anymore in the **Data record** list field. The oscilloscope changes automatically to the display of the next offline data record in the list field.
   - If the closed data record was included in the MERGE data record, its channels will be removed from the MERGE data record.
   - If all offline data records have been deleted, the oscilloscope changes to the online data record if the device is currently online. Otherwise no data record will be displayed.

12.5.6 Overlay function

The overlay function serves to lay several characteristics from the currently loaded data records on top of each other, e.g. to compare signal characteristics from different recordings.

- If two or more data records are loaded in the oscilloscope, e.g. an online data record and a data record saved before in the project, a "merge" data record is automatically provided in the **Data record** list field.
- If the merge data record is selected, the desired characteristics to be overlaid or compared can be selected from the loaded data records in the **Vertical channel settings** group field.
- If an online data record is used in the merge data record, an update is carried out in the merge data record in case of a renewed recording.
- Removing variables from an offline or online data record causes the characteristics in the merge data record to be deleted.

12.5.7 Copying a data record to the clipboard

For documentation purposes, it is possible to copy the measured data of a data record as a table or, alternatively, the oscilloscope user interface as a picture, to the clipboard for use in other programs.

How to copy measured data or a picture of the user interface to the clipboard:

1. Click on the icon in the *Oscilloscope toolbar*.
   - The *Clipboard* dialog box appears.
2. Select **Curve points** if the measured data are to be copied to the clipboard as a table, or select **Screenshot** if the oscilloscope user interface is to be copied to the clipboard as a picture.
3. Click **OK**.
   - The dialog box is closed and the selected option is copied to the clipboard.

12.6 Variables of the motor control (oscilloscope signals)

The system variables of the internal motor control listed in the following table can be recorded with the oscilloscope for diagnostic and documentation purposes.

Note!

In comparison to all other variable values, the system variables of the internal motor control have a cycle offset of 2 ... 3 ms!

Tip!

The exact position of a variable in the motor control can be obtained from the corresponding signal flow.

Variable of the motor control	Meaning
▶ Signal flow - servo control for synchronous motor (📖 168) ▶ Signal flow - servo control for asynchronous motor (📖 170)	
Common.dnActualFlux	Actual flux value
Common.dnFluxSet	Flux setpoint
Current.dnActualCurrentPhaseU	Actual motor current (phase U)
Current.dnActualCurrentPhaseV	Actual motor current (phase V)
Current.dnActualCurrentPhaseW	Actual motor current (phase W)
Current.dnActualDirectCurrent	Actual D current
Current.dnActualQuadratureCurrent	Actual Q current
Current.dnDirectCurrentSet	D current setpoint
Current.dnQuadratureCurrentSet	Q current setpoint
Torque.dnActualMotorTorque	Actual torque
Torque.dnFilteredTorqueSetpoint	Filtered torque setpoint
Torque.dnInputNotchFilter1	Torque setpoint at the band-stop filter input 1
Torque.dnInputNotchFilter2	Torque setpoint at the band-stop filter 2 input
Voltage.dnActualDCBusVoltage	Actual DC-bus voltage
Voltage.dnActualMotorVoltage	Current motor voltage
Voltage.dnOutputQuadratureCurrentCtrl	Q-output voltage of the current controller
Voltage.dnOutputDirectCurrentCtrl	D-output voltage of the current controller
Voltage.dnDirectVoltage	D voltage
Voltage.dnQuadratureVoltage	Q voltage

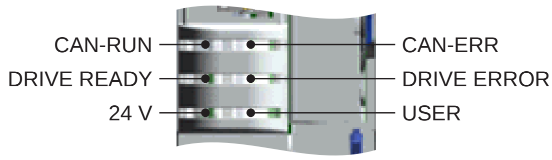
Variable of the motor control	Meaning
Signal flow - sensorless vector control (📖 189)	
Common.dnActualFlux	Actual flux value
Current.dnActualDirectCurrent	Actual D current
Current.dnActualQuadratureCurrent	Actual Q current
Current.dnDirectCurrentSet	D current setpoint
Current.dnQuadratureCurrentSet	Q current setpoint
Frequency.dnActualRotatingFieldFrequency	Actual field frequency
Frequency.dnActualSlipFrequency	Actual slip frequency
Speed.dnActualMotorSpeed	Actual speed
Torque.dnTorqueSetpoint	Torque setpoint
Voltage.dnActualMotorVoltage	Current motor voltage
Voltage.dnDirectVoltage	D voltage
Voltage.dnQuadratureVoltage	Q voltage
Signal flow - V/f control (📖 205) Signal flow for closed loop V/f control (📖 207)	
Current.Current.dnActualMotorCurrent	Actual motor current
Current.dnActualQuadratureCurrent	Actual Q current
Frequency.dnActualRotatingFieldFrequency	Actual field frequency
Frequency.dnActualSlipFrequency	Actual slip frequency
Speed.dnActualMotorSpeed	Current motor speed
Speed.dnSpeedSetpoint	Speed setpoint
Voltage.dnActualMotorVoltage	Current motor voltage
Voltage.dnOutputDirectCurrentCtrl	D voltage
Voltage.dnOutputQuadratureVoltage	Q voltage
Signal flow - encoder evaluation (📖 245)	
Position.dnActualLoadPos	Actual position
Position.dnActualMotorPos	Current motor position
Speed.dnActualEncoderSpeed	Current encoder speed
Speed.dnActualMotorSpeed	Current motor speed
Speed.dnActualResolverSpeed	Current resolver speed
Signal flow - position follower (📖 497)	
Position.dnActualLoadPos	Actual position
Position.dnActualMotorPos	Current motor position
Position.dnContouringError	Following error
Position.dnPositionSetpoint	Position setpoint
Speed.dnActualMotorSpeed	Current motor speed
Speed.dnOutputPosCtrl	Output signal - phase controller
Speed.dnSpeedSetpoint	Speed setpoint
Torque.dnTorqueSetpoint	Torque setpoint
Torque.dnTotalTorqueAdd	Additive torque feedforward control value

Variable of the motor control		Meaning
▶ Signal flow - speed follower (📖 504)		
	Speed.dnActualMotorSpeed	Current motor speed
	Speed.dnSpeedSetpoint	Speed setpoint
	Speed.dnTotalSpeedAdd	Additive speed setpoint
	Torque.dnTorqueSetpoint	Torque setpoint
	Torque.dnTotalTorqueAdd	Additive torque feedforward control value
▶ Signal flow - torque follower (📖 510)		
	Speed.dnActualMotorSpeed	Current motor speed
	Speed.dnSpeedSetpoint	Speed setpoint

13 Diagnostics & fault analysis

13.1 LED status displays

You can quickly receive notes on some operating states via the LED display:

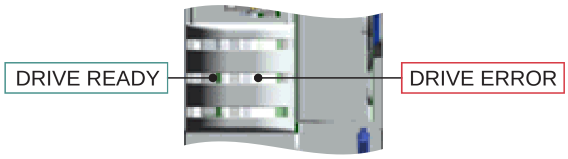


[13-1] LED display on the front panel of the controller

Labelling	Colour	Description	
CAN-RUN	Green	CAN bus is OK	▶ LED status displays for the system bus (📖 298)
CAN-ERR	Red	CAN bus error	
DRIVE READY	Green	Standard device is ready for operation	▶ LED status displays for the device state (📖 612)
DRIVE ERROR	Red	Warning/Trouble/Fault	
24 V	Green	24 V supply voltage is OK	
USER	Yellow	Message parameterised by the application	

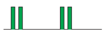
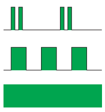
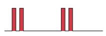
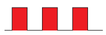
13.1.1 LED status displays for the device state

The control of the two LEDs "DRIVE READY" and "DRIVE ERROR" in the middle of the controller's front panel depends on the device state. ▶ Device states (📖 105)

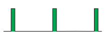


[13-2] LED status displays DRIVE READY and DRIVE ERROR

The meaning can be obtained from the following table:

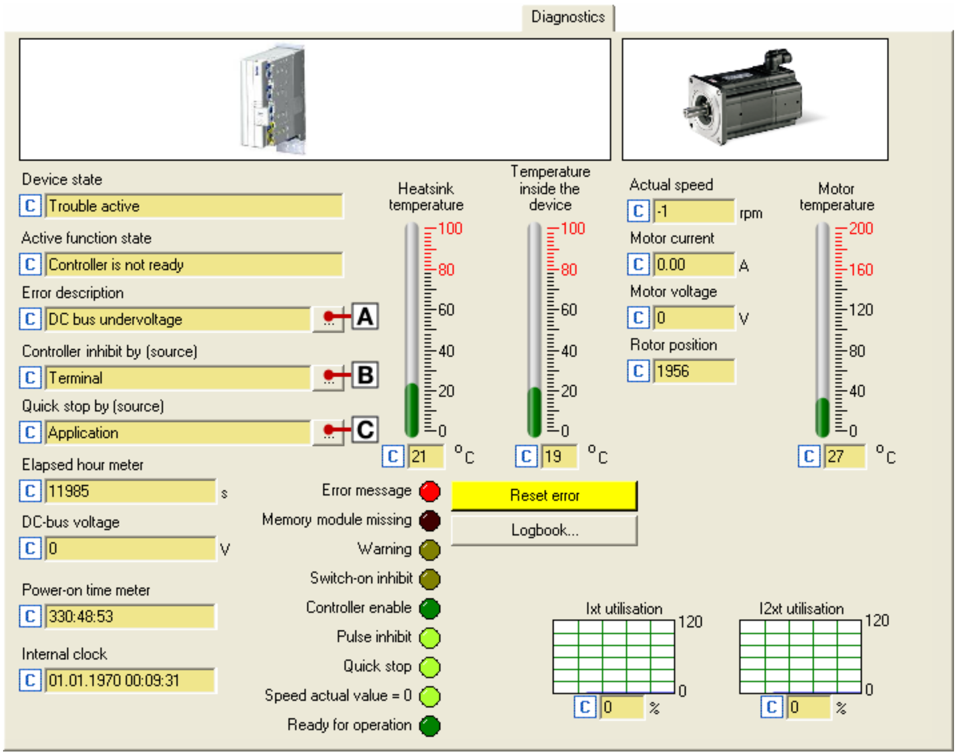
DRIVE READY	DRIVE ERROR	Meaning
OFF	OFF	"Initialisation active" state
	OFF	"Safe torque off active" state Observe LED of the safety module!
	OFF	"Device is ready to switch on" state
	OFF	"Device is switched on" state
	OFF	"Operation" state
		"Warning active" state or "Warning locked active" state The controller is ready to switch on, switched on or operation is enabled, and a warning is present.
		"Quick stop by trouble active" state
OFF		"Trouble active" state
OFF		"Fault active" state
OFF		"System fault active" state

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

	LED flashes once approx. every three seconds (*slow flash*)
	LED flashes once approx. every 1.25 seconds (*flash*)
	LED flashes twice approx. every 1.25 seconds (*double flash*)
	LED blinks every second
	LED is permanently on

13.2 Drive diagnostics with the »Engineer«

When an online connection to the controller has been established, you can carry out a diagnostics of the connected controller by means of the »Engineer« and have important actual states of the controller displayed in a concise visualisation:



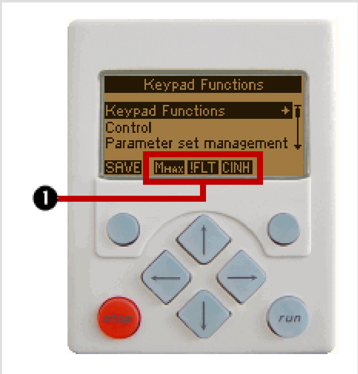
Button		Function
...	A	Show details on the current error.
	B	Show all active sources for controller inhibit.
	C	Show all active sources for quick stop.
Reset error		Acknowledge error message (if the error cause is eliminated).
Logbook...		Show Logbook of the controller. (616)

How to carry out a drive diagnostics by means of the »Engineer«:

1. Select the 9400 HighLine controller to be diagnosed in the *Project view*.
2. Click the icon or select the command **Online→Go online** 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.

13.3 Drive diagnostics via keypad/bus system

Keypad display of the controller status



- If the keypad on the front of the controller is connected to the diagnostic interface X6, the status of the controller is shown via different icons on the LCD display in the area ❶.

Icon	Meaning	Note
RDY	Controller is ready for operation.	
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 switch on.	
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 active	
!TRB	Trouble active	
!TQSP	Quick stop by trouble active	
WRN	Warning active	

Display parameters

Via the parameters listed in the following tables current states and actual values of the controller can be queried for diagnostic purposes, e.g. by using the keypad, a bus system, or the »Engineer« (when an online connection to the controller has been established).

- 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". (📖 705)

Parameter	Display
C00183	Device state
C00166	Error description
C00168	Error number
C00051	Actual speed [rpm]

Parameter	Display
C00052	Motor voltage
C00054	Motor current
C00057/1	Maximum torque
C00057/2	Motor reference torque
C00059	Motor - number of pole pairs
C00060	Rotor position
C00061	Heatsink temperature
C00062	Temperature inside the controller
C00063	Motor temperature
C00064	Device utilisation (Ixt) during the last 180 seconds
C00065	Ext. 24-V voltage
C00066	Thermal motor load (I^2xt)
C00068	Capacitor temperature
C00069	CPU temperature
C00178	Time during which the controller was enabled (elapsed-hour meter)
C00179	Time during which the mains was switched on (power-on time meter)
C00186	ENP: Identified motor type

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
C00200	Firmware product type
C00201	Firmware compilation date
C00203/1...9	HW product types
C00204/1...9	HW serial numbers
C00205/1...6	HW descriptions
C00206/1...6	HW manufacturing data
C00208/1...6	HW manufacturer
C00209/1...6	HW countries of origin
C00210/1...6	HW versions
C02113	Program name

13.4 Logbook

The integrated logbook function of the controller chronologically logs important events within the system and thus plays an important role for troubleshooting or controller diagnostics.

Events that can be logged

The following events can be logged in the logbook:

- Error messages of the operating system (623)
- Error messages generated by the application
- Controller enable
- Starting / stopping the application
- Loading/saving of parameter sets, loading of the Lenze setting
- Transmitting an application or firmware to the controller
- Switching the controller on/off
- Formatting the file system

Tip!

Use a parameterisable filter to exclude certain events from logbook entry. Filtering logbook entries (617)

Information saved

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

- Type of response (e.g. trouble, warning, or information) to the event
- Event
- Value of power-on time meter
- Date/time (for memory module with real-time clock)
- Module that activated the event (A = application; S = system).

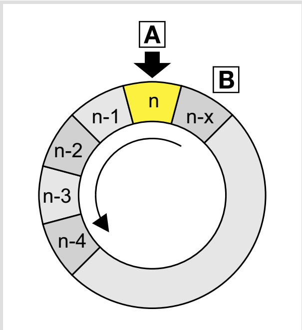
Memory depth

The number of possible logbook entries depends on the memory module used:

- MM1xx, MM2xx: 7 entries
- MM3xx, MM4xx: 439 entries

13.4.1 Functional description

The structure of the logbook corresponds to a ring buffer structure:



- 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 latest entries always remain available.

[13-3] Ring buffer structure

13.4.2 Filtering logbook entries

The logbook enters new entries in the ring buffer after they have passed through a parameterisable filter. By means of this filter you can exclude events with a specific error response (trouble, warning, information, etc.) from being entered in the logbook.

Note!

Basically, events with the set response "None" are not entered in the logbook.

The filter is parameterised in C00169 by means of a bit mask. A set bit inhibits the entry of the corresponding event in the logbook.

- From software version V5.0 the option that identical consecutive entries ("Multiple entries") into the logbook are suppressed can be additionally activated via bit 0. Then only the time stamp of the last (latest) entry and the number of times the same event has occurred successively are saved.

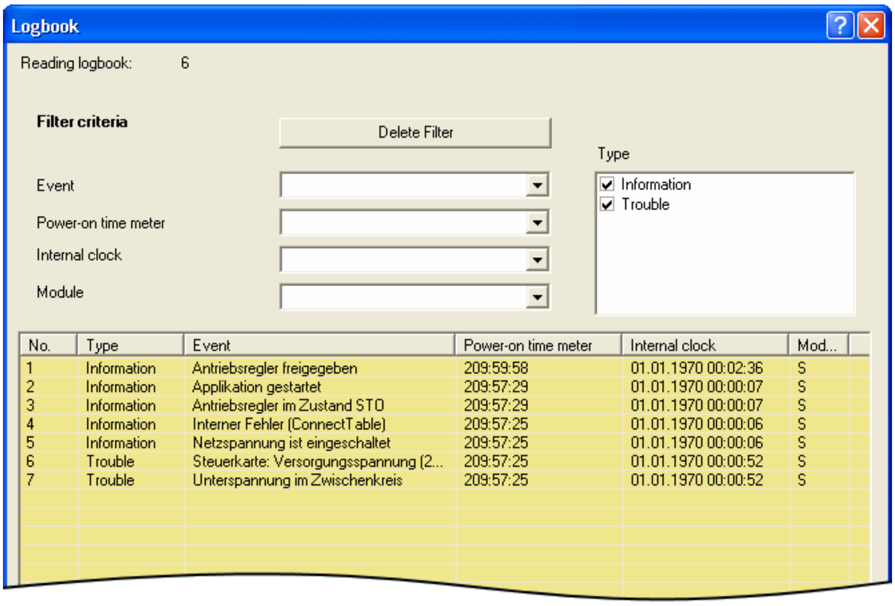
Bit	Filter	Lenze setting
0	No multiple entries	0 ≡ filter inactive
1	Fault	0 ≡ filter inactive
2	Trouble	0 ≡ filter inactive
3	Quick stop by trouble	0 ≡ filter inactive
4	Warning locked	0 ≡ filter inactive
5	Warning	0 ≡ filter inactive
6	Information	0 ≡ filter inactive

13.4.3 Reading out logbook entries

When an online connection has been established, the existing logbook entries can simply 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. Go to the *Project view* and select the 9400 HighLine controller the logbook entries of which are to be read out.
2. Click the icon or select the command **Online→Go online** to build up an online connection with the controller.
3. Select the **Diagnostics** tab from the *Workspace*.
4. Click **Logbook**.
   - The *Logbook* dialog box appears:



   - You can filter the entries displayed systematically by selecting or defining filter criteria.
   - Click **Delete** to delete an entry from the logbook.
5. Click **Previous** to close the *Logbook* dialog box.

Tip!

- The **Export...** button serves to export the logbook entries into a file. ▶ Export logbook entries into a file (📖 619)

13.4.4 Export logbook entries into a file

How to export logbook entries into 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 **Save** to export the logbook entries into the given file.
   - Hidden logbook entries are not exported, i.e., the specified filter criteria will also be considered during the export.
   - The logbook entries are written into the file in the form of a list separated by semicolons.

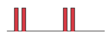
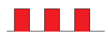
Example

```
Type;event;error number;number;power-on time meter;internal clock;module
Fault;motor:overtemperature;611778563;1;16243:36:56;01.01.1970 00:00;temperature monitoring
Fault;motor:thermal detector is defective;611778572;1;16243:36:56;01.01.1970 00:00;temperature
monitoring
Fault;resolver: open circuit;612040728;1;16243:36:55;01.01.1970 00:00;motor control
```

13.5 Monitoring

The controller is provided with different monitoring functions that protect the drive against impermissible operating conditions.

- If a monitoring function responds,
  - an entry is made into the Logbook of the controller,
  - the response (quick stop by trouble, warning, fault, etc.) selected for this monitoring function is activated,
  - the state of the internal device control changes according to the response selected, controller inhibit is set, and the "DRIVE ERROR" LED on the front of the controller goes on:

Response	Logbook entry	Display under C00168	Pulse inhibit	Controller inhibit	Acknowledgement required	LED "DRIVE ERROR"
None						OFF
Information	☑					OFF
Warning	☑	☑				
Warning locked	☑	☑			☑	
Quick stop by trouble	☑	☑			☑	
Trouble	☑	☑	☑	☑ (after 0.5 s)		
Fault	☑	☑	☑	☑	☑	
System fault	☑	☑	☑	☑	Mains switching is required!	

Danger!

If the automatic restart is enabled (C00142 = "1: Enabled"), the drive can restart automatically from the "Trouble" and "Safe torque off" device states when the trouble or requirement for "Safe torque off active" is no longer available!

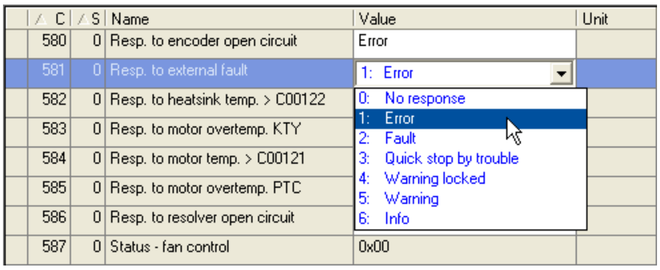
▸ Automatic restart after mains connection/trouble... (📖 112)

See also:
▸ Device states (📖 105)
▸ LED status displays for the device state (📖 612)

13.5.1 Setting the error response

If a monitoring function responds, the response set for this monitoring function (quick stop by trouble, warning, fault, etc.) is triggered.

- ▶ For many monitoring functions the response can be individually configured via parameters:



C	S	Name	Value	Unit
580	0	Resp. to encoder open circuit	Error	
581	0	Resp. to external fault	1: Error	
582	0	Resp. to heatsink temp. > C00122		
583	0	Resp. to motor overtemp. KTY		
584	0	Resp. to motor temp. > C00121		
585	0	Resp. to motor overtemp. PTC		
586	0	Resp. to resolver open circuit		
587	0	Status - fan control	0x00	

Tip!

The fault messages for which the response can be set can be gathered from the table in the chapter "Short overview (A-Z)". (📖 628)

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	Information
C00120	Motor overload protection (I²xt)
C00121	Motor temp. warning threshold
C00122	Heatsink temp. warn. threshold
C00123	Device utilisation warning threshold
C00126	CPU temp. warning threshold
C00127	Mot. overload warning threshold
C00128	Thermal time constant of motor
C00174	Undervoltage (LU) threshold
C00570	Warning thres. brake transistor
C00572	Warning thres. brake resistor
C00576	Speed monitoring tolerance
C00596	Threshold max. speed reached
C00599	Motor phase failure threshold
C00620	Max. motor current threshold

13.6 Maloperation of the drive

The motor does not rotate.

Cause	Remedy
DC-bus voltage is too low.	Check mains voltage.
Controller is inhibited.	Deactivate controller inhibit (can be set by several sources).
Motor holding brake is not released.	Release motor holding brake.
Quick stop active.	Deactivate quick stop.
Setpoint = 0	Select setpoint.

With a positive speed setpoint selection, the motor rotates counter-clockwise instead of clockwise (when looking at the motor shaft).

Cause	Remedy
Feedback system is not connected in correct phase relation.	Connect feedback system in correct phase relation.

The maximum current (C00022) flows and the motor does not rotate according to the defined speed setpoint.

Cause	Remedy
Two motor phases are interchanged, i.e. an anti-clockwise rotating field is applied to the motor.	Carry out the following steps for verification: 1. Ensure that the motor shaft is not blocked and can rotate freely without damaging the system. 2. Activate the "U-rotation test mode" for the motor control (C00398 = "1"). – In this test mode a voltage phasor with the frequency set in C00399/1 and the amplitude from the linear characteristic of rated voltage and rated frequency is applied to the machine, which corresponds to a clockwise rotating field. – ⚠ **Danger!** **When the test mode is active, the parameterisable error response "Quick stop by trouble" has no effect!** If the test mode is active and a monitoring function responds with this error response, no quick stop is executed but the motor continues to rotate with the frequency set for the test mode! 3. Increase the frequency step by step for the test mode in C00399/1 until the motor shaft starts to rotate. – If the motor shaft does not rotate, check the electrical connection. 4. While the motor shaft is rotating, check whether it rotates clockwise when looking at the A end shield. If not, two motor phases are interchanged. 5. Additionally check whether the actual speed value shown in C00051 is positive and whether it corresponds to the defined frequency, taking the number of pole pairs of the machine into consideration (C00059). If this is not the case, the connection and the parameter setting of the feedback system are to be checked. 6. Deactivate the test mode for the motor control again (C00398 = "0").

13.7 Error messages of the operating system

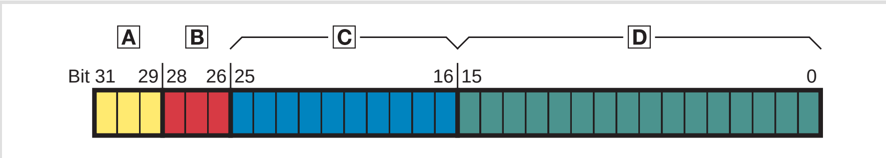
This chapter describes all error messages of the controller operating system and possible causes & remedies.

Tip!

Each error message is also saved in the logbook in chronological order. ▸ Logbook (📖 616)

13.7.1 Structure of the error number (bit coding)

If an error occurs in the controller, a 32-bit value will be saved in decimal format in the internal history buffer (C00168), which contains the following information:

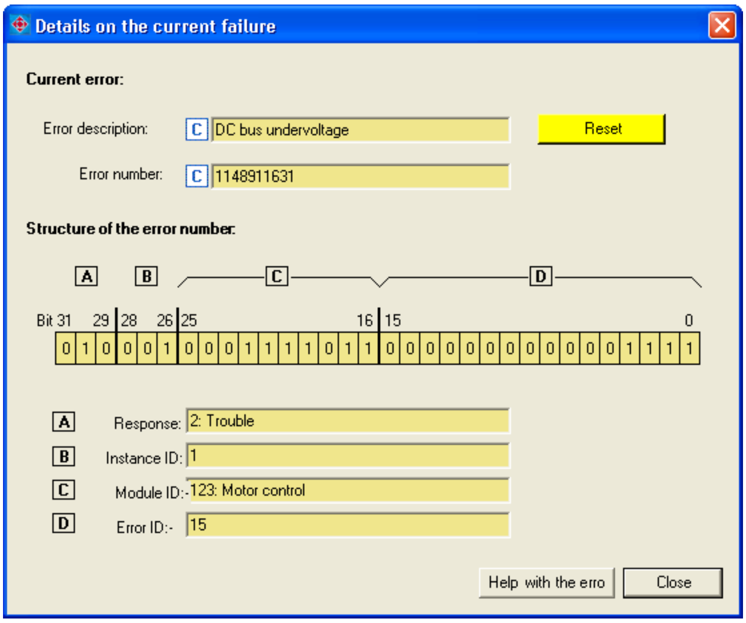


- A Response
- B Instance ID
- C Module ID
- D Error ID

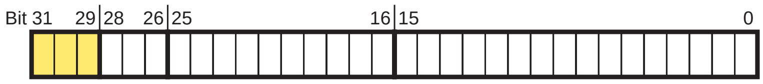
[13-4] Structure of the error number

Tip!

If you go to the **Diagnostics** tab and click the button ... to the right of the **Error description** display parameter, you will be shown all details on the current error in a separate dialog box.

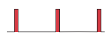
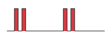
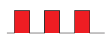


13.7.1.1 Response

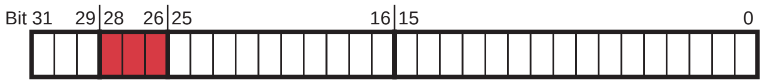


Bit 31	Bit 30	Bit 29	Response
0	0	0	0: No response
0	0	1	1: Fault
0	1	0	2: Trouble
0	1	1	3: Quick stop by trouble
1	0	0	4: Warning locked
1	0	1	5: Warning
1	1	0	6: Information
1	1	1	7: System fault

The state of the internal device control changes according to the selected response to an error, controller inhibit is set, and the "DRIVE ERROR" LED on the front of the controller goes on:

Response	Logbook entry	Display under C00168	Pulse inhibit	Controller inhibit	Acknowledgement required	LED "DRIVE ERROR"
None						OFF
Information	☑					OFF
Warning	☑	☑				
Warning locked	☑	☑			☑	
Quick stop by trouble	☑	☑			☑	
Trouble	☑	☑	☑	☑ (after 0.5 s)		
Fault	☑	☑	☑	☑	☑	
System fault	☑	☑	☑	☑	Mains switching is required!	

13.7.1.2 Instance ID

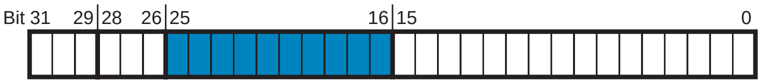


The instance ID is dynamically assigned by the operating system.

Bit 28	Bit 27	Bit 26	Meaning
0	0	0	Instance ID 0
0	0	1	Instance ID 1
0	1	0	Instance ID 2
0	1	1	Instance ID 3
1	0	0	Instance ID 4
1	0	1	Instance ID 5

Bit 28	Bit 27	Bit 26	Meaning
1	1	0	Instance ID 6
1	1	1	Instance ID 7

13.7.1.3 Module ID

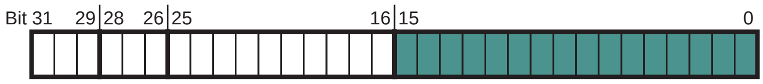


Use the module ID to identify the module in which the error has occurred.

Module ID		Module
hex	decimal	
0x0065	**101**	Logbook module
0x0068	**104**	Module identification
0x0069	**105**	Error check of the program memory during runtime
0x006a	**106**	Runtime environment for IEC 61131-3 programs
0x006e	**110**	Supply voltage monitoring
0x006f	**111**	24-V supply voltage monitoring
0x0072	**114**	Service register
0x0075	**117**	Device control
0x0077	**119**	Temperature monitoring
0x0078	**120**	Analog signal monitoring
0x0079	**121**	Motor data interface
0x007a	**122**	Processing of digital inputs/outputs
0x007b	**123**	Motor control
0x007c	**124**	Device command module (C00002)
0x007d	**125**	Processing of analog inputs/outputs
0x007f	**127**	Interface to the intelligent communication module
0x0083	**131**	"CAN on board": CAN-Dispatcher
0x0084	**132**	"CAN on board": CAN-NMT-Handler
0x0085	**133**	"CAN on board": CAN-Emergency-Handler
0x0086	**134**	"CAN on board": CAN-NMT-Master
0x0087	**135**	"CAN on board": CAN-PDO-Handler
0x0088	**136**	"CAN on board": CAN-SDO-Server
0x0089	**137**	"CAN on board": CAN-SDO-Client
0x008c	**140**	Application project manager
0x008e	**142**	Communication interface for internal communication
0x0090	**144**	Parameter manager
0x0091	**145**	Lenze runtime system
0x0092	**146**	Interface for safety module
0x0093	**147**	Sync signal generation
0x0099	**153**	Extension module - digital frequency in MXI1
0x009d	**157**	CAN module in MXI1: CAN-Dispatcher
0x009e	**158**	CAN module in MXI1: CAN-NMT-Handler
0x009f	**159**	CAN module in MXI1: CAN-Emergency-Handler
0x00a0	**160**	CAN module in MXI1: CAN-NMT-Master

Module ID		Module
hex	decimal	
0x00a1	**161**	CAN module in MXI1: CAN-PDO-Handler
0x00a2	**162**	CAN module in MXI1: CAN-SDO-Server
0x00a3	**163**	CAN module in MXI1: CAN-SDO-Client
0x00aa	**170**	Extension module - digital frequency in MXI2
0x00ac	**172**	CAN module in MXI2: CAN-Dispatcher
0x00ad	**173**	CAN module in MXI2: CAN-NMT-Handler
0x00ae	**174**	CAN module in MXI2: CAN-Emergency-Handler
0x00af	**175**	CAN module in MXI2: CAN-NMT-Master
0x00b0	**176**	CAN module in MXI2: CAN-PDO-Handler
0x00b1	**177**	CAN module in MXI2: CAN-SDO-Server
0x00b2	**178**	CAN module in MXI2: CAN-SDO-Client
0x00b8	**184**	Basic drive functions
0x00c8	**200**	Intelligent communication module
0x012f	**303**	Safety module SM300/SM301

13.7.1.4 Error ID

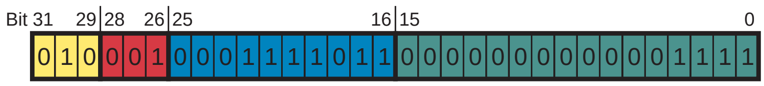


16-bit value (0 ... 65535$_{dec}$) for error identification.

13.7.1.5 Example for bit coding of the error number

C00168 displays the error number "1148911631".

- This decimal value corresponds to the following bit sequence:



Assignment	Information	Meaning in the example
0 1 0	Response	2: Trouble
0 0 1	Instance ID	1: Instance ID 1
0 0 0 1 1 1 1 0 1 1	Module ID	Module ID 123 (0x007b): Motor control
0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	Error ID	Error ID 15 (0x000f) for motor control: Undervoltage in the DC bus

- Error number "1148911631" thus means:
  The "DC-bus undervoltage" error with the response "Trouble" occurred in the "motor control" module with the instance ID 1.

13.7.2 Reset of error message

An error message with the response "Fault", "Quick stop by trouble", or "Warning locked" must be reset (acknowledged) explicitly after the cause of error has been eliminated.

To reset (acknowledge) a pending error message, execute the device command C00002 = "43: Reset error".

Tip!

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

13.7.3 Short overview (A-Z)

The following table contains all error messages of the controller operating system in alphabetical order with the preset error response and - if available – the parameter for setting the error response.

Tip!

If you click on the cross-reference in the first column, you get to the detailed description of the corresponding error response in the following chapter "Cause & possible remedies". (636)

Note!

Error message "Unknown error"

If the "Unknown error xxxx" error message is indicated in the logbook or in C00166, the reason for the missing plain text is that the error texts required have not been downloaded to the controller during the application download.

- This, for instance, is the case if a device module plugged into the controller has not been included in the Engineer project.
- Remedy: Include the device module, recompile and download the project.

Error number		Error message	Response	Adjustable in
hex	dec		(Lenze setting)	
0x0090000c	9437196	Disconnection in the case of par. storage		-
0x007b001a	8060954	Absolute value encoder: Communication error	Fault	-
0x006a0000	6946816	General error in the application	Fault	-
0x007d0000	8192000	Analog input 1: master current < 4 mA	Fault	C00598
0x00750001	7667713	Controller enabled	Information	-
0x00750003	7667715	Controller in STO state	Information	-
0x007b0047	8060999	Controller: Clamp operation	Information	-
0x00750005	7667717	Controller: Pulse inhibit is active	Information	-
0x007b0035	8060981	Controller: Overload during acceleration phases	Fault	-
0x006a0004	6946820	ApplicationTask: Overflow	Fault	C02111
0x006a0013	6946835	Application has started	Information	-
0x006a000e	6946830	Application has stopped	Information	-
0x006a0014	6946836	Application has stopped	Information	-
0x008c000c	9175052	Application and device are incompatible	Fault	-
0x006a0010	6946832	Faulty application parameter	Fault	-
0x007b003f	8060991	A mains phase has failed	Fault	-
0x007b002d	8060973	Failure of motor phase U	No response	C00597
0x007b002e	8060974	Failure of motor phase V	No response	C00597
0x007b002f	8060975	Failure of motor phase W	No response	C00597
0x00b8000c	12058636	Acceleration has been limited	Information	C02716/3
0x00910012	9502738	Block function in wrong MEC task	Fault	-
0x006a000f	6946831	Breakpoint reached	Information	-
0x007b0040	8060992	Brake chopper: Ixt > C00570	Warning	C00569
0x007b001c	8060956	Brake transistor: Ixt overload	No response	C00573
0x007b0021	8060961	Brake transistor: Overcurrent	Fault	-

Error number		Error message	Response	Adjustable in
hex	dec		(Lenze setting)	
0x007b0041	8060993	Brake resistor: I2t > C00572	Warning	C00571
0x007b001d	8060957	Brake resistor: I²xt overload	Information	-
0x00b80014	12058644	Cam Data is corrupted	Warning locked	-
0x00b80016	12058646	Cam Data locked due to wrong password	Warning locked	-
0x00b80017	12058647	Cam Data locked due to wrong security key	Warning locked	-
0x00b80015	12058645	Cam Data restored	Fault	-
0x00b80013	12058643	Cam data: Serial number MM does not match	Warning locked	-
0x00b80034	12058676	Cam Data: Invalidated (due to change of mechanical data)	Warning	-
0x00b80035	12058677	Cam Data: Invalid product number	Information	-
0x00870008	8847368	CAN on board PDO manager: Faulty configuration	Warning locked	-
0x00870000	8847360	CAN on board RPDO1: Telegram not received or faulty	No response	C00591/1
0x00870001	8847361	CAN on board RPDO2: Telegram not received or faulty	No response	C00591/2
0x00870002	8847362	CAN on board RPDO3: Telegram not received or faulty	No response	C00591/3
0x00870003	8847363	CAN on board RPDO4: Telegram not received or faulty	No response	C00591/4
0x00890000	8978432	CAN on board SDO client: Faulty configuration	Warning locked	-
0x00880000	8912896	CAN on board SDO server: Faulty configuration	Warning locked	-
0x00830002	8585218	CAN on board: Basic configuration invalid	Warning locked	-
0x00830000	8585216	CAN on board: Bus off	No response	C00595
0x00850000	8716288	CAN on board: Faulty emergency configuration	Warning locked	-
0x00840000	8650752	CAN on board: Heartbeat error index 1 ... 32	No response	C00613/1...32
0x00840020	8650784	CAN on board: Lifeguarding error	No response	C00614
0x00860020	8781856	CAN on board: Faulty NMT master configuration	Warning locked	-
0x00840021	8650785	CAN on board: Faulty NMT slave configuration	Warning locked	-
0x00860000	8781824	CAN on board: Node guarding error 1 ... 32	No response	C00612/1...32
0x00830001	8585217	CAN on board: Invalid node address 0	Warning	-
0x00a10008	10551304	CAN module (MXI1) PDO manager: Faulty configuration	Warning locked	-
0x00a10000	10551296	CAN module (MXI1) RPDO1: Telegram not received or faulty	No response	C13591/1
0x00a10001	10551297	CAN module (MXI1) RPDO2: Telegram not received or faulty	No response	C13591/2
0x00a10002	10551298	CAN module (MXI1) RPDO3: Telegram not received or faulty	No response	C13591/3
0x00a10003	10551299	CAN module (MXI1) RPDO4: Telegram not received or faulty	No response	C13591/4
0x00a10004	10551300	CAN module (MXI1) RPDO5: Telegram not received or faulty	No response	C13591/5
0x00a10005	10551301	CAN module (MXI1) RPDO6: Telegram not received or faulty	No response	C13591/6
0x00a10006	10551302	CAN module (MXI1) RPDO7: Telegram not received or faulty	No response	C13591/7
0x00a10007	10551303	CAN module (MXI1) RPDO8: Telegram not received or faulty	No response	C13591/8
0x00a30000	10682368	CAN module (MXI1) SDO client: Faulty configuration	Warning locked	-
0x00a20000	10616832	CAN module (MXI1) SDO server: Faulty configuration	Warning locked	-
0x009d0000	10289152	CAN module (MXI1): Bus off	Information	C13595
0x009f0000	10420224	Can module (MXI1): Faulty emergency configuration	Warning locked	-
0x009e0021	10354721	CAN module (MXI1): Faulty NMT slave configuration	Warning locked	-
0x009d0002	10289154	CAN module (MXI1): Basic configuration invalid	Warning locked	-
0x009e0000	10354688	CAN module (MXI1): Heartbeat error index 1 ... 32	No response	C13613/1...32
0x009e0020	10354720	CAN module (MXI1): Lifeguarding error	No response	C13614
0x00a00020	10485792	CAN module (MXI1): Faulty NMT master configuration	Warning locked	-
0x00a00000	10485760	CAN module (MXI1): Node guarding error 1 ... 32	No response	C13612/1...32
0x009d0001	10289153	CAN module (MXI1): Invalid node address 0	Warning	-
0x00b00008	11534344	CAN module (MXI2) PDO manager: Faulty configuration	Warning locked	-
0x00b00000	11534336	CAN module (MXI2) RPDO1: Telegram not received or faulty	No response	C14591/1
0x00b00001	11534337	CAN module (MXI2) RPDO2: Telegram not received or faulty	No response	C14591/2
0x00b00002	11534338	CAN module (MXI2) RPDO3: Telegram not received or faulty	No response	C14591/3
0x00b00003	11534339	CAN module (MXI2) RPDO4: Telegram not received or faulty	No response	C14591/4
0x00b00004	11534340	CAN module (MXI2) RPDO5: Telegram not received or faulty	No response	C14591/5

Error number		Error message	Response	Adjustable in
hex	dec		(Lenze setting)	
0x00b00005	11534341	CAN module (MXI2) RPDO6: Telegram not received or faulty	No response	C14591/6
0x00b00006	11534342	CAN module (MXI2) RPDO7: Telegram not received or faulty	No response	C14591/7
0x00b00007	11534343	CAN module (MXI2) RPDO8: Telegram not received or faulty	No response	C14591/8
0x00b20000	11665408	CAN module (MXI2) SDO client: Faulty configuration	Warning locked	-
0x00b10000	11599872	CAN module (MXI2) SDO server: Faulty configuration	Warning locked	-
0x00ac0000	11272192	CAN module (MXI2): Bus off	Information	C14595
0x00ae0000	11403264	Can module (MXI2): Faulty emergency configuration	Warning locked	-
0x00ac0002	11272194	CAN module (MXI2): Basic configuration invalid	Warning locked	-
0x00ad0000	11337728	CAN module (MXI2): Heartbeat error index 1 ... 32	No response	C14613/1...32
0x00ad0020	11337760	CAN module (MXI2): Lifeguarding error	No response	C14614
0x00af0020	11468832	CAN module (MXI2): Faulty NMT master configuration	Warning locked	-
0x00ad0021	11337761	CAN module (MXI2): Faulty NMT slave configuration	Warning locked	-
0x00af0000	11468800	CAN module (MXI2): Node guarding error 1 ... 32	No response	C14612/1...32
0x00ac0001	11272193	CAN module (MXI2): Invalid node address 0	Warning	-
0x00900008	9437192	Code number duplicated	Warning locked	-
0x00690000	6881280	Code refresh	System fault	-
0x008c001a	9175066	ConnectTable active	Information	-
0x00770008	7798792	CPU: Temperature > C00126	No response	C00589
0x0077000e	7798798	CPU: Thermal detector is defective	Fault	C00588
0x00770009	7798793	CPU: Overtemperature	Warning	-
0x008c0002	9175042	File DeviceCFG.dat is defective	Fault	-
0x008c0005	9175045	File DeviceCFG.dat is missing	Fault	-
0x008c0008	9175048	File DeviceCFG.dat is invalid	Fault	-
0x008c0001	9175041	File ProjectList.dat is defective	Fault	-
0x008c0004	9175044	File ProjectList.dat is missing	Fault	-
0x008c0007	9175047	File ProjectList.dat is invalid	Fault	-
0x008c0000	9175040	File ProjectSelection.dat is defective	Fault	-
0x008c0003	9175043	File ProjectSelection.dat is missing	Fault	-
0x008c0006	9175046	File ProjectSelection.dat is invalid	Fault	-
0x00990003	10027011	DFIN (MXI1): Signal error enable/lamp control	Warning	C13041
0x00990000	10027008	DFIN (MXI1): Track error A-/A	Fault	C13040
0x00990001	10027009	DFIN (MXI1): Track error B-/B	Fault	C13040
0x00990002	10027010	DFIN (MXI1): Track error Z-/Z	Fault	C13040
0x00990004	10027012	DFIN (MXI1): Supply cannot be corrected anymore	Warning	C13042
0x00aa0003	11141123	DFIN (MXI2): Signal error enable/lamp control	Warning	C14041
0x00aa0000	11141120	DFIN (MXI2): Track error A-/A	Fault	C14040
0x00aa0001	11141121	DFIN (MXI2): Track error B-/B	Fault	C14040
0x00aa0002	11141122	DFIN (MXI2): Track error Z-/Z	Fault	C14040
0x00aa0004	11141124	DFIN (MXI2): Supply cannot be corrected anymore	Warning	C14042
0x00990005	10027013	DFOUT (MXI1): Maximum frequency reached	Warning	C13080
0x00aa0005	11141125	DFOUT (MXI2): Maximum frequency reached	Warning	C14080
0x006a0011	6946833	Division by zero	Fault	-
0x006a0001	6946817	Faulty program download	Fault	-
0x007b0012	8060946	Actual speed value outside tolerance (C00576)	No response	C00579
0x00680022	6815778	Real-time clock is defective	Warning locked	-
0x00680024	6815780	Real-time clock: No battery, time lost	Warning locked	-
0x00680023	6815779	Real-time clock: Change battery	Warning locked	-
0x007b0039	8060985	Electronic nameplate: Data outside parameter limits	Information	-
0x007b0030	8060976	Electronic nameplate: Data loaded	Information	-
0x0078000a	7864330	Electronic nameplate: Data are incompatible	Information	-
0x007b0032	8060978	Electronic nameplate: Encoder protocol unknown	Information	-

Error number		Error message	Response	Adjustable in
hex	dec		(Lenze setting)	
0x007b0033	8060979	Electronic nameplate: Encoder signal unknown	Information	-
0x0068001b	6815771	Electronic nameplate: Communication error	Warning	-
0x007b0031	8060977	Electronic nameplate: Not found	Information	-
0x0068001d	6815773	Electronic nameplate: Checksum error	Warning	-
0x007b001b	8060955	Encoder: Open circuit	Fault	C00580
0x007b002c	8060972	EnDat encoder: Battery empty	Information	-
0x007b004f	8061007	EnDat encoder: Command error	Information	-
0x007b0026	8060966	EnDat encoder: Lamp error	Information	-
0x007b0050	8061008	EnDat encoder: Initial position error	Information	-
0x007b0028	8060968	EnDat encoder: Position error	Information	-
0x007b0027	8060967	EnDat encoder: Signal error	Information	-
0x007b0029	8060969	EnDat encoder: Overvoltage	Information	-
0x007b002b	8060971	EnDat encoder: Overcurrent	Information	-
0x007b004e	8061006	EnDat encoder: Transmission error	Information	-
0x007b002a	8060970	EnDat encoder: Undervoltage	Information	-
0x007b0011	8060945	Earth fault detected	Fault	-
0x008c000b	9175051	Required license missing	Fault	-
0x00750000	7667712	External error	Fault	C00581
0x00680013	6815763	Incorrect safety module	System fault	-
0x00680012	6815762	Incorrect memory module	System fault	-
0x006a0002	6946818	Error during the update of inputs and outputs	Fault	-
0x006a0017	6946839	Error in control configuration	System fault	-
0x00910011	9502737	Error during initialisation	System fault	-
0x0068001e	6815774	Firmware incompatible to control card	System fault	-
0x0068001a	6815770	Firmware has been changed	Information	-
0x007b004d	8061005	Encoder: Timeout error		-
0x007b003e	8060990	Encoder monitoring: Pulse deviation detected	No response	C00621
0x00780001	7864321	Device utilisation Ixt > 100 %	Fault	-
0x00780000	7864320	Device utilisation Ixt > C00123	Warning	C00604
0x00790000	7929856	Incorrect device command transfer	System fault	-
0x00770011	7798801	Inside the device: Fan is defective	Fault	C00611
0x0077000b	7798795	Inside the device: Thermal detector is defective	Fault	C00588
0x00b8000b	12058635	Speed has been limited	Information	C02716/3
0x00910003	9502723	Heartbeat not periodic	System fault	-
0x007b003b	8060987	Hiperface encoder: Command error	Information	-
0x007b003c	8060988	Hiperface encoder: Unknown encoder	Information	-
0x007b003d	8060989	Hiperface encoder: Initial position error	Information	-
0x007b003a	8060986	Hiperface encoder: Transmission error	Information	-
0x006a0006	6946822	IdleTask: Overflow	Fault	-
0x00b8001a	12058650	Int. overflow C02620 (manual jog: Speed 1)	Fault	-
0x00b8001b	12058651	Int. overflow C02621 (manual jog: Speed 2)	Fault	-
0x00b8001c	12058652	Int. overflow C02622 (manual jog: Acc.)	Fault	-
0x00b8001d	12058653	Int. overflow C02623 (manual jog: Dec.)	Fault	-
0x00b8002d	12058669	Int. overflow C02642 (home position)	Fault	-
0x00b8002e	12058670	Int. overflow C02643 (homing: Target position)	Fault	-
0x00b8002f	12058671	Int. overflow C02644 (homing: Speed 1)	Fault	-
0x00b80030	12058672	Int. overflow C02645 (homing: Acceleration 1)	Fault	-
0x00b80031	12058673	Int. overflow C02646 (homing: Speed 2)	Fault	-
0x00b80032	12058674	Int. overflow C02647 (homing: Acceleration 2)	Fault	-
0x00b80033	12058675	Int. overflow C02670 (positioner: Tol. for target position)	Fault	-
0x00b80020	12058656	Int. overflow C02701/1 (positive software limit pos.)	Fault	-

Error number		Error message	Response	Adjustable in
hex	dec		(Lenze setting)	
0x00b80021	12058657	Int. overflow C02701/2 (negative software limit pos.)	Fault	-
0x00b80022	12058658	Int. overflow C02703 (maximum speed)	Fault	-
0x00b80023	12058659	Int. overflow C02705 (maximum acceleration)	Fault	-
0x00b80028	12058664	Int. overflow C02708/1 (decel. limited speed 1)	Fault	-
0x00b80024	12058660	Int. overflow C02708/1 (limited speed 1)	Fault	-
0x00b80029	12058665	Int. overflow C02708/2 (decel. limited speed 2)	Fault	-
0x00b80025	12058661	Int. overflow C02708/2 (limited speed 2)	Fault	-
0x00b8002a	12058666	Int. overflow C02708/3 (decel. limited speed 3)	Fault	-
0x00b80026	12058662	Int. overflow C02708/3 (limited speed 3)	Fault	-
0x00b8002b	12058667	Int. overflow C02708/4 (decel. limited speed 4)	Fault	-
0x00b80027	12058663	Int. overflow C02708/4 (limited speed 4)	Fault	-
0x00b8002c	12058668	Int. overflow C02713 (max. dist. manual control)	Fault	-
0x00690009	6881289	Internal error (event mechanism)	System fault	-
0x0069000a	6881290	Internal error (event mechanism)	System fault	-
0x00690002	6881282	Internal error (LDS instance data)	System fault	-
0x00690003	6881283	Internal error (LDS tasks)	System fault	-
0x0069000d	6881293	Internal error (file system lifetime)	Warning	-
0x00690007	6881287	Internal error (message queue)	System fault	-
0x00690006	6881286	Internal error (message memory)	System fault	-
0x00690008	6881288	Internal error (name database)	System fault	-
0x0069000b	6881291	Internal error (semaphores)	System fault	-
0x0069000c	6881292	Internal error (semaphores)	System fault	-
0x00690001	6881281	Internal error (memory area - logbook)	System fault	-
0x00690004	6881284	Internal error (memory blocks)	System fault	-
0x00690005	6881285	Internal error (task queue)	System fault	-
0x00910004	9502724	Internal error: See C00180	System fault	-
0x00910005	9502725	Internal error: See C00180	System fault	-
0x00910006	9502726	Internal error: See C00180	System fault	-
0x00910008	9502728	Internal error: See C00180	Fault	-
0x00910009	9502729	Internal error: See C00180	Fault	-
0x007b0034	8060980	Internal communication error (DMA)	System fault	-
0x007b0014	8060948	Internal communication error (host MCTRL)	System fault	-
0x007b0036	8060982	Internal communication error (MCTRL host)	System fault	-
0x00910002	9502722	No heartbeat signal detected	System fault	-
0x0090000a	9437194	No parameters for module in MXI1	Fault	C00615/2
0x0090000b	9437195	No parameters for module in MXI2	Fault	C00615/3
0x00680019	6815769	Combination MXI1/MXI2 not possible	System fault	-
0x0068001f	6815775	Combination of memory module/device not possible	System fault	-
0x00680020	6815776	Combination of module in MXI1/device not possible	System fault	-
0x00680021	6815777	Combination of module in MXI2/device not possible	System fault	-
0x007f0003	8323075	Communication with module in MXI1 interrupted	Information	-
0x007f0004	8323076	Communication with module in MXI2 interrupted	Information	-
0x00920001	9568257	Communication with safety module interrupted	Information	-
0x007f0002	8323074	Communication error between device and device module	No response	C01501
0x0091000e	9502734	Communication task: Standstill > 3 s	Fault	-
0x00770010	7798800	Heatsink: Fan is defective	Fault	C00610
0x00770000	7798784	Heatsink: Temperature > C00122	Warning	C00582
0x0077000a	7798794	Heatsink: Thermal detector is defective	Fault	C00588
0x00770001	7798785	Heatsink: Overtemperature	Fault	-
0x007b0023	8060963	Load encoder: Module selected in C00490 is not available	Fault	-
0x006a000d	6946829	Run-time error	Fault	-

Error number		Error message	Response	Adjustable in
hex	dec		(Lenze setting)	
0x0068000f	6815759	Power section incompatible	System fault	-
0x00680001	6815745	Power section is defective	System fault	-
0x00680009	6815753	Power section is defective	System fault	-
0x007b0042	8060994	Power section is defective	System fault	-
0x00680014	6815764	Power section has been changed	Information or warning locked if the hardware type has also changed.	-
0x00900001	9437185	Lenze setting loaded	Information	-
0x00900004	9437188	Loading of Lenze setting failed	Fault	-
0x00720000	7471104	Read error service register	Fault	-
0x00650001	6619137	Logbook: Reset (read error)	Information	-
0x00650002	6619138	Logbook: Reset (version error)	Information	-
0x00650000	6619136	Logbook: Overflow	Information	-
0x00b80010	12058640	Maximum speed exceeded	Information	C02716/3
0x00b80011	12058641	Maximum acceleration exceeded	Information	C02716/3
0x007b004c	8061004	Motor disconnected	No response	C00597
0x007b0007	8060935	Motor: rated current < rated magnetising current	Information	-
0x007b0002	8060930	Motor: Calculated mutual inductance unrealistic	Information	-
0x007b000a	8060938	Motor: Calculated mutual inductance unrealistic	Information	-
0x007b0001	8060929	Motor: Calculated motor impedance unrealistic	Information	-
0x007b000c	8060940	Motor: Calculated rotor time constant unrealistic	Information	-
0x007b0019	8060953	Motor: Calculated leakage inductance unrealistic	Information	-
0x007b000b	8060939	Motor: Calculated e.m.f. factor unrealistic	Information	-
0x007b000d	8060941	Motor: Calculated flux factor unrealistic	Information	-
0x007b0009	8060937	Motor: Calculated rotor resistance unrealistic	Information	-
0x007b0020	8060960	Motor: Actual speed value > C00596	Fault	C00607
0x007b0006	8060934	Motor: Device current too low for rated magnetisation	Information	-
0x007b0004	8060932	Motor: Phase resistance too high	Information	-
0x0077000f	7798799	Motor: PTC has triggered	No response	C00585
0x007b001e	8060958	Motor: Actual current value > C00620	Fault	C00619
0x00770002	7798786	Motor: Temperature > C00121	Warning	C00584
0x0077000c	7798796	Motor: Thermal detector is defective	Fault	C00594
0x00770003	7798787	Motor: Overtemperature	Fault	C00583
0x00780003	7864323	Motor load I^2xt > C00120	Fault	-
0x00780002	7864322	Motor load I^2xt > C00127	Warning	C00606
0x00b80004	12058628	Motor brake: Autom. activated after waiting time has elapsed	Information	-
0x00b80005	12058629	Motor brake: Status monitoring error	Quick stop by trouble	-
0x00b80003	12058627	Motor brake: Angular drift with closed brake too high	Quick stop by trouble	-
0x007b0003	8060931	Motor data are inconsistent	Information	-
0x007b0017	8060951	Motor data are inconsistent	Information	-
0x007b0024	8060964	Motor encoder: Module selected in C00495 is not available	Fault	-
0x007b0038	8060984	Motor parameter identification cancelled	Fault	-
0x007b0013	8060947	Motor control: Task overflow	System fault	-
0x007b0025	8060965	Motor temperature: Module selected in C01193 is not available	Fault	-
0x008c0017	9175063	MXI1: CAN module is missing or incompatible	Fault	C00615/2
0x008c0011	9175057	MXI1: Ethernet module is missing or incompatible	Fault	C00615/2
0x00680010	6815760	MXI1: Wrong module	System fault	-
0x008c0015	9175061	MXI1: ICM module is missing or incompatible	Fault	C00615/2
0x008c0013	9175059	MXI1: Digital frequency module is missing or incompatible	Fault	C00615/2
0x008c000d	9175053	MXI1: Module is missing or incompatible	Fault	-
0x0068000a	6815754	MXI1: Module is defective or missing	Fault	-
0x00680004	6815748	MXI1: Module changed during operation	Warning locked	-

Error number		Error message	Response	Adjustable in
hex	dec		(Lenze setting)	
0x00680015	6815765	MXI1: Module has been changed	Information or warning locked if the hardware type has also changed.	-
0x008c000f	9175055	MXI1: PROFIBUS module is missing or incompatible	Fault	C00615/2
0x008c0018	9175064	MXI2: CAN module is missing or incompatible	Fault	C00615/3
0x008c0012	9175058	MXI2: Ethernet module is missing or incompatible	Fault	C00615/3
0x00680011	6815761	MXI2: Wrong module	System fault	-
0x008c0016	9175062	MXI2: ICM module is missing or incompatible	Fault	C00615/3
0x008c0014	9175060	MXI2: Digital frequency module is missing or incompatible	Fault	C00615/3
0x008c000e	9175054	MXI2: Module is missing or incompatible	Fault	-
0x0068000b	6815755	MXI2: Module is defective or missing	System fault	-
0x00680005	6815749	MXI2: Module changed during operation	Warning locked	-
0x00680016	6815766	MXI2: Module has been changed	Information or warning locked if the hardware type has also changed.	-
0x008c0010	9175056	MXI2: PROFIBUS module is missing or incompatible	Fault	C00615/3
0x00b8000a	12058634	Negative direction of rotation limited	Information	C02716/1
0x00b80002	12058626	Negative limit switch has triggered	Quick stop by trouble	-
0x00b80008	12058632	Negative software limit switch overtravelled	Quick stop by trouble	C02716/2
0x00910001	9502721	Mains voltage is switched off	Information	-
0x00910000	9502720	Mains voltage is switched on	Information	-
0x006a0003	6946819	New applications loaded	Information	-
0x00900006	9437190	Saving of parameters failed	Fault	-
0x00900000	9437184	Parameter set faulty	Fault	-
0x00900003	9437187	Parameter set loaded	Information	-
0x00900002	9437186	Parameter set saved	Information	-
0x00900005	9437189	Parameter set restored	Fault	-
0x00900009	9437193	Parameter set: Type of standard device has been changed	Information	-
0x00900007	9437191	Parameter set: Version conflict	Fault	-
0x006a0015	6946837	PDO mapping (MXI1): Faulty configuration	Fault	-
0x006a0016	6946838	PDO mapping (MXI2): Faulty configuration	Fault	-
0x007b004a	8061002	Pole position identification cancelled	Fault	C00640
0x00910010	9502736	Position value faulty	Fault	-
0x00b8000f	12058639	Position target outside the software limit positions	Quick stop by trouble	C02716/2
0x00b80009	12058633	Positive direction of rotation limited	Information	C02716/1
0x00b80001	12058625	Positive limit switch has triggered	Quick stop by trouble	-
0x00b80007	12058631	Positive software limit switch overtravelled	Quick stop by trouble	C02716/2
0x008c0009	9175049	Project not loaded	Fault	-
0x008c000a	9175050	Project not available	Fault	-
0x00b80019	12058649	Homing mode not allowed	Fault	-
0x007b001f	8060959	Resolver: Calculated acceleration unrealistic	Information	-
0x007b0018	8060952	Resolver: Open circuit	Fault	-
0x006a001a	6946842	Retain memory of the application faulty		-
0x00b8000e	12058638	Jerk has been limited	Information	C02716/3
0x00680003	6815747	Safety module is defective or missing	System fault	-
0x0068000d	6815757	Safety module is defective or missing	System fault	-
0x00680007	6815751	Safety module has been removed	System fault	-
0x00680018	6815768	Safety module has been changed	Information or warning locked if the hardware type has also changed.	-
0x00920000	9568256	Safety module: Incompatible with setting in C00214	System fault	-
0x00650003	6619139	Memory module is missing	Information	-
0x00680002	6815746	Memory module is defective or missing	Fault	-

Error number		Error message	Response	Adjustable in
hex	dec		(Lenze setting)	
0x0068000c	6815756	Memory module is defective or missing	System fault	-
0x00680006	6815750	Memory module has been removed	System fault	-
0x00680017	6815767	Memory module has been changed	Information or warning locked if the hardware type has also changed.	-
0x0068001c	6815772	Memory module: Faulty file system	Fault	-
0x007c0000	8126464	Memory module: File system has been formatted	Information	-
0x007c0001	8126465	Memory module: File system has been restored	Information	-
0x00b80000	12058624	PLC configuration invalid	Fault	-
0x007d0001	8192001	PLC configuration invalid	Fault	-
0x007b0037	8060983	PLC configuration invalid	Fault	-
0x00750006	7667718	PLC configuration invalid	Fault	-
0x0068000e	6815758	Control card incompatible	System fault	-
0x00680000	6815744	Control card is defective	System fault	-
0x00680008	6815752	Control card is defective	System fault	-
0x00780008	7864328	Control card is defective (UB18 neg.)	System fault	-
0x00780004	7864324	Control card is defective (UB24)	System fault	-
0x00780006	7864326	Control card is defective (UB8)	System fault	-
0x00780007	7864327	Control card is defective (VCC15 neg.)	System fault	-
0x00780005	7864325	Control card is defective (VCC15)	System fault	-
0x00780009	7864329	Control card is defective (VCC5)	System fault	-
0x006f0000	7274496	Control card: Supply voltage (24 V DC) too low	Trouble	-
0x0091000a	9502730	System task 1: Task overflow	System fault	-
0x0091000b	9502731	System task 2: Task overflow	Information	-
0x0091000c	9502732	System task 3: Task overflow	System fault	-
0x0091000d	9502733	System task: Task overflow	Fault	-
0x00b80012	12058642	Time-out torque feedforward control brake	Quick stop by trouble	-
0x007b0010	8060944	Overcurrent detected	Fault	-
0x006a0005	6946821	UserTask: Overflow	Fault	C02111
0x00790002	7929858	Violation of time slice	Fault	-
0x00b8000d	12058637	Deceleration has been limited	Information	C02716/3
0x006a0012	6946834	Pointer access in impermissible memory area	Fault	-
0x00790001	7929857	Time error - controller interface	System fault	-
0x0077000d	7798797	DC-bus capacitor: Thermal detector is defective	Fault	C00588
0x007b000e	8060942	DC-bus overvoltage	Trouble	C00600
0x007b000f	8060943	DC-bus undervoltage	Trouble	-
0x0091000f	9502735	Cyclic task: Standstill > 60 s	Information	-

13.7.4 Cause & possible remedies

This chapter contains all error messages of the controller operating system in numerical order of the error number with detailed information on the response to the error message and information on the cause & possible remedies.

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)". (628)

Note!

Error message "Unknown error"

If the "Unknown error xxxx" error message is indicated in the logbook or in C00166, the reason for the missing plain text is that the error texts required have not been downloaded to the controller during the application download.

- This, for instance, is the case if a device module plugged into the controller has not been included in the Engineer project.
- Remedy: Include the device module, recompile and download the project.

Logbook: Overflow [0x00650000]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
Too many events/faults have occurred in a very short time. It was therefore not possible to list all of them in the logbook.	Check whether application generates too many error messages.

Logbook: Reset (read error) [0x00650001]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
The logbook has been reset due to a read error..	- (is irreversible)

Logbook: Reset (version error) [0x00650002]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
The logbook has been reset due to a version conflict.	- (is irreversible)

Memory module is missing [0x00650003]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
Memory module is defective or not available.	Use a different memory module.

Control card is defect [0x00680000]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Operating system could not identify the control card.	Mains switching • Please contact Lenze if the error occurs again.

Power section is defect [0x00680001]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Operating system could not identify the power section.	Mains switching • Please contact Lenze if the error occurs again.

Memory module is defect or missing [0x00680002]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Operating system could not identify the memory module.	Mains switching • If the error occurs again: Switch off the controller, remove memory module and plug in again, switch on the controller again. • If the error still occurs: Switch off controller and use a different memory module.

Safety module is defect or missing [0x00680003]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Operating system could not identify the safety module.	Mains switching • If the error occurs again: Switch off the controller, remove safety module and plug in again, switch on the controller again. • If the error still occurs: Switch off controller and use a different safety module.

MXI1: Module changed during operation [0x00680004]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information	

Cause	Remedy
It was tried to plug an extension module into module slot MXI1, which is not "Hot-plug"-able.	Plug in valid module and switch on mains. • Through mains switching, the system accepts modules without "hot plug" capability in the following switch-on phase.

MXI2: Module changed during operation [0x00680005]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information	

Cause	Remedy
It was tried to plug an extension module into module slot MXI2, which is not "Hot-plug"-able.	Plug in valid module and switch on mains. • Through mains switching, the system accepts modules without "hot plug" capability in the following switch-on phase.

Memory module has been removed [0x00680006]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
You have tried to remove or change the memory module during operation.	Switch off the controller, plug in memory module and switch on the controller again. • If the error occurs again, the memory module is defective and must be replaced.

Safety module has been removed [0x00680007]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
You have tried to remove or change the safety module during operation.	Switch off the controller, plug in safety module and switch on the controller again. • If the error occurs again, the safety module is defective and must be replaced.

Control card is defect [0x00680008]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Operating system could not identify the control card.	Please contact Lenze.

Power section is defect [0x00680009]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Operating system could not identify the power section.	Please contact Lenze.

MXI1: Module is defect or missing [0x0068000a]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Operating system could not identify the extension module in module slot MXI1.	• Use a different extension module. • Please contact Lenze.

MXI2: Module is defect or missing [0x0068000b]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Operating system could not identify the extension module in module slot MXI2.	• Use a different extension module. • Please contact Lenze.

Memory module is defect or missing [0x0068000c]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Operating system could not identify the memory module.	• Use a different memory module. • Please contact Lenze.

Safety module is defect or missing [0x0068000d]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Operating system could not identify the safety module.	• Use a different safety module. • Please contact Lenze.

Control card incompatible [0x0068000e]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
The control card is not supported by the operating system.	Please contact Lenze.

Power section incompatible [0x0068000f]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
The power section is not supported by the operating system.	Please contact Lenze.

MXI1: Wrong module [0x00680010]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
The extension module in module slot MXI1 is not supported by the operating system.	• Use a different module. • Please contact Lenze.

MXI2: Wrong module [0x00680011]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
The extension module in module slot MXI2 is not supported by the operating system.	• Use a different module. • Please contact Lenze.

Incorrect memory module [0x00680012]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
The memory module is not supported by the operating system.	• Use a different module. • Please contact Lenze.

Incorrect safety module [0x00680013]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
The safety module is not supported by the operating system.	• Use a different module. • Please contact Lenze.

Power section was changed [0x00680014]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☒ **Information**	
Cause	**Remedy**
The power section has been changed since the last mains switching.	(Only information or warning locked if the hardware type has also changed.)

MXI1: Module has been changed [0x00680015]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☒ **Information**	
Cause	**Remedy**
The extension module in module slot MXI1 has been changed since the last mains switching.	(Only information or warning locked if the hardware type has also changed.)

MXI2: Module has been changed [0x00680016]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☒ **Information**	
Cause	**Remedy**
The extension module in module slot MXI2 has been changed since the last mains switching.	(Only information or warning locked if the hardware type has also changed.)

Memory module has been changed [0x00680017]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☒ **Information**	
Cause	**Remedy**
The memory module has been changed since the last mains switching.	(Only information or warning locked if the hardware type has also changed.)

Safety module has been changed [0x00680018]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☒ **Information**	
Cause	**Remedy**
The safety module has been changed since the last mains switching.	(Only information or warning locked if the hardware type has also changed.)

Combination MXI1/MXI2 not possible [0x00680019]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Extension modules which are not supported in this combination are plugged into module slots MXI1 & MXI2.	Create permitted module combination.

Firmware has been changed [0x0068001a]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
The firmware of the operating system has been updated.	- (Information only)

Electronic nameplate: Communication error [0x0068001b]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information	
Cause	**Remedy**
Communication with the electronic nameplate is interrupted, the data could not be read.	Check correct connection of the encoder cable.

Memory module: Faulty file system [0x0068001c]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Memory module is plugged in incorrectly or is defective.	• Plug in the memory module correctly. • Exchange defective memory module.

Electronic nameplate: Checksum error [0x0068001d]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information

Cause	Remedy
The checksum of the electronic nameplate is defective.	Please contact Lenze.

Firmware is incompatible with control card [0x0068001e]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Firmware is not compatible with the hardware.	Import the compatible firmware.

Combination memory module/device not possible [0x0068001f]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
The memory module used is not supported by the controller according to the license model.	Plug in supported module and switch the mains. • The 9400 HighLine supports the MM220 (licence: Motion Control HighLevel) and MM330 memory modules (licence: Motion Control TopLevel).

Combination of module in MXI1/device not possible [0x00680020]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
The extension module in module slot MXI1 is not supported by the controller.	• Remove the extension module and switch the mains. • Plug in supported extension module and switch the mains.

Combination of module in MXI2/device not possible [0x00680021]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
The extension module in module slot MXI2 is not supported by the controller.	• Remove the extension module and switch the mains. • Plug in supported extension module and switch the mains.

Real-time clock is defective [0x00680022]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
Device error: The clock integrated in the MM440 memory module is defective.	• Replace memory module. • Please contact Lenze.

Real-time clock: Change battery [0x00680023]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
The battery in the clock integrated in the MM440 memory module is low. The clock is expected to fail soon.	Replace memory module.

Real-time clock: Battery empty, time lost [0x00680024]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
The battery integrated in the clock in the MM440 memory module is empty. The clock has been reset to its initial value (01.01.1970 - 00:00:00 o'clock).	• If the memory module is used for the first time, restart the controller to initialise the memory module. • If the problem occurs again, replace the memory module.

Code refresh [0x00690000]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (memory area - logbook) [0x00690001]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (LDS instance data) [0x00690002]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (LDS tasks) [0x00690003]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (storage blocks) [0x00690004]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (task queue) [0x00690005]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (message memory) [0x00690006]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (message queue) [0x00690007]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (name data base) [0x00690008]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (event mechanism) [0x00690009]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (event mechanism) [0x0069000a]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (semaphores) [0x0069000b]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (semaphores) [0x0069000c]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Serious device error or component failure.	Switch the controller off and then on again. • Please contact Lenze if the problem occurs again.

Internal error (file system lifetime) [0x0069000d]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information	
Cause	**Remedy**
The maximum number of permissible writing cycles has been reached for the memory module.	Replace memory module, otherwise data may get lost.

General error in the application [0x006a0000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
General application error.	Mains switching. Transmit the application to the controller again. • Please contact Lenze if the problem occurs again.

Faulty program download [0x006a0001]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Faulty transmission of the application to the controller (checksum error).	Repeat transmission.

Fault during the update of the inputs and outputs [0x006a0002]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	Mains switching. Transmit the application to the controller again. • Please contact Lenze if the problem occurs again.

New application loaded [0x006a0003]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Application has been changed by transmission from the Engineer or loading from the memory module.	- (Information only)

ApplicationTask: Overflow [0x006a0004]

Response (Lenze setting printed in bold)	**Setting:** C02111 (☑ Adjustable response)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Program runtime in Application task is too high.	Reduce program runtime by means of: • Omitting functions (e.g. by reducing the number of active FBs). • Optimisation of functions to the calculating time.

UserTask: Overflow [0x006a0005]

Response (Lenze setting printed in bold)	**Setting:** C02111 (☑ Adjustable response)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Program runtime in user task is too high.	Reduce program runtime by means of: • Omitting functions (e.g. by reducing the number of active FBs). • Optimisation of functions to the calculating time.

IdleTask: Overflow [0x006a0006]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Program runtime in idle task is too high.	Reduce program runtime by means of: • Omitting functions (e.g. by reducing the number of active FBs). • Optimisation of functions to the calculating time.

Runtime error [0x006a000d]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
A runtime error has occurred in the application. The application processing has been interrupted.	Remove runtime error in the application and retransfer application to controller.

Application has stopped [0x006a000e]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
The application has been stopped using the device command C00002="32". All user tasks are stopped.	Restart application with device command C00002="31".

Breakpoint reached [0x006a000f]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
The application has reached a set breakpoint and the user task with the breakpoint has stopped.	Delete breakpoint and restart application.

Faulty application parameter [0x006a0010]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
An invalid parameter description is available.	Transmit application and parameter set to the controller again.

Division by zero [0x006a0011]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
An impermissible division by zero occurred in the application. The division has been intercepted and the divisor was replaced by the value "1".	Replace application.

Pointer access in impermissible memory area [0x006a0012]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
An invalid pointer access to a protected area occurred in the application.	Replace application.

Application has started [0x006a0013]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
The application in the controller has been started.	- (Information only)

Application has stopped [0x006a0014]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
The application in the controller has been stopped.	- (Information only)

PDO mapping (MXI1): Faulty configuration [0x006a0015]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI1: Incorrectly configured process data mapping. • The corresponding PDO channel is not installed, e. g. because no communication module was selected for module slot MXI1 in the Engineer project. • The communication module selected for module slot MXI1 in the Engineer project does not support PDO mapping. • The mapping information downloaded to the controller is faulty.	• Integrate suitable communication module for module slot MXI1 in the Engineer project. • Check the configuration of the network. Then recompile the project and transmit it to the controller.

PDO mapping (MXI2): Faulty configuration [0x006a0016]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI2: Incorrectly configured process data mapping. • The corresponding PDO channel is not installed, e. g. because no communication module was selected for module slot MXI2 in the Engineer project. • The communication module selected for module slot MXI2 in the Engineer project does not support PDO mapping. • The mapping information downloaded to the controller is faulty.	• Integrate suitable communication module for module slot MXI2 in the Engineer project. • Check the configuration of the network. Then recompile the project and transmit it to the controller.

Fault in the control configuration [0x006a0017]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
An invalid control configuration has occurred.	Load a different application.

Retain memory of the application faulty [0x006a001a]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**

Control card: Supply voltage (24 V DC) too low [0x006f0000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☒ **Trouble** ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
External 24 V supply voltage of the control card is lower than 18 V.	Check external supply voltage. • If the external supply voltage is available and the error message does not disappear, please contact Lenze.

Read error service register [0x00720000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
During reading or writing the service register a fault has occurred.	Mains switching • Please contact Lenze if the problem occurs again.

External error [0x00750000]

Response (Lenze setting printed in bold)	**Setting:** C00581 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	
Cause	**Remedy**
The drive interface has activated the error message "External error". • The input *DI_bSetExternError* of the system block **LS_DriveInterface** has been set to TRUE.	• Check external device to be monitored. • Check assignment of the input *DI_bSetExternError* in the application.

Controller enabled [0x00750001]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
The controller is enabled and has the "Operation" state.	- (Information only)

Controller in STO state [0x00750003]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
The controller has received the "Safe torque off (STO)" request by the safety module and is now in the "Safe torque off active" device state.	- (Information only)

Controller: Pulse inhibit is active [0x00750005]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
The pulse inhibit is active in the controller.	- (Information only)

PLC configuration invalid [0x00750006]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
An invalid control configuration has occurred.	Load a different application.

Heatsink: Temperature > C00122 [0x00770000]

Response (Lenze setting printed in bold)	**Setting:** C00582 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information	

Cause	Remedy
Heatsink temperature higher than variable temperature limit (C00122). • Ambient controller temperature too high. • Dirty fan or ventilation slots. • Value set under C00122 is too low.	• Check control cabinet temperature. • Clean filter. • Clean controller. • Set a higher value under C00122.

Heatsink: Overtemperature [0x00770001]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Heatsink temperature higher than fixed temperature limit (90 °C). • Ambient controller temperature too high. • Dirty fan or ventilation slots.	• Check control cabinet temperature. • Clean filter. • Clean controller.

Motor: Temperature > C00121 [0x00770002]

Response (Lenze setting printed in bold)	**Setting:** C00584 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information	

Cause	**Remedy**
Motor temperature higher than variable temperature limit (C00121). • Motor too hot due to impermissibly high currents or frequent and too long acceleration processes. • No PTC connected. • Value set under C00121 is too low.	• Check drive dimensioning. • Connect PTC or switch off monitoring (C00584="0"). • Set a higher value under C00121.

Motor: Overtemperature [0x00770003]

Response (Lenze setting printed in bold)	**Setting:** C00583 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
Motor temperature higher than fixed temperature limit (150 °C). • Motor too hot due to impermissibly high currents or frequent and too long acceleration processes. • No PTC connected.	• Check drive dimensioning. • Connect PTC or switch off monitoring (C00583="0").

CPU: Temperature > C00126 [0x00770008]

Response (Lenze setting printed in bold)	**Setting:** C00589 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
CPU temperature higher than variable temperature limit (C00126). • Ambient controller temperature too high. • Dirty fan or ventilation slots. • Value set under C00126 is too low.	• Check control cabinet temperature. • Clean filter. • Clean controller. • Set a higher value under C00126.

CPU: Overtemperature [0x00770009]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information

Cause	**Remedy**
CPU temperature higher than fixed limit temperature (85 °C). • Ambient controller temperature too high. • Dirty fan or ventilation slots.	• Check control cabinet temperature. • Clean filter. • Clean controller.

Heatsink: Thermal detector is defect [0x0077000a]

Response (Lenze setting printed in bold)	**Setting:** C00588 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
Encoder for heatsink temperature supplies undefined values.	Check control cabinet temperature, maybe it is too low.

Inside the device: Thermal detector is defective [0x0077000b]

Response (Lenze setting printed in bold)	**Setting:** C00588 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
Encoder for interior temperature supplies undefined values.	Check control cabinet temperature, maybe it is too low.

Motor: Thermal detector is defect [0x0077000c]

Response (Lenze setting printed in bold)	**Setting:** C00594 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
The signals of the connected encoder for the motor temperature detection (resolver at X7 or encoder at X8) are outside the defined operating range of the detection.	• Check contacts of the encoder cable at the motor and controller. • Check selection of the motor temperature sensor in C01190 and make sure that it complies with the assembly in the motor. • Possibly switch off temperature sensor monitoring (C00594="0"). • If a PTC is in the motor, activate the monitoring of the PTC temperature in C00585 instead.

DC-bus capacitor: Thermal detector is defect [0x0077000d]

Response (Lenze setting printed in bold)	**Setting:** C00588 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
Encoder for capacitor temperature supplies undefined values.	Check control cabinet temperature, maybe it is too low.

CPU: Thermal detector is defect [0x0077000e]

Response (Lenze setting printed in bold)	**Setting:** C00588 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
Encoder for CPU temperature supplies undefined values.	Check control cabinet temperature, maybe it is too low.

Motor: PTC has triggered [0x0077000f]

Response (Lenze setting printed in bold)	**Setting:** C00585 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
The motor temperature detected via the terminals T1/T2 is too high. • Motor is too hot due to an increased effective current that results from operation with too high/too frequent acceleration processes. • Motor too hot due to increased ambient conditions. • Motor too hot due to lacking cooling in the case of self-ventilation and continuous operation with speeds lower than the rated speed. • Terminals T1/T2 are not assigned. • Open circuit of the supply cables for terminals T1/T2.	• Check drive dimensioning. • Connect PTC or thermal contact to terminals T1/T2. • If a motor without integrated temperature monitoring is used, switch off the monitoring function (C00585="0").

Heatsink: Fan is defect [0x00770010]

Response (Lenze setting printed in bold)	**Setting:** C00610 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	
Cause	**Remedy**
Speed of heatsink fan too low, e.g. due to dirt.	Check/clean fan.

Inside the device: Fan is defective [0x00770011]

Response (Lenze setting printed in bold)	**Setting:** C00611 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☑ Warning ☐ Information	
Cause	**Remedy**
Speed of internal fan is too low, e.g. due to dirt.	Check/clean fan.

Device utilisation > C00123 [0x00780000]

Response (Lenze setting printed in bold)	**Setting:** C00604 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information	
Cause	**Remedy**
Frequent and too long acceleration processes with overcurrent > C00123.	Check drive dimensioning.

Device utilisation Ixt > 100 % [0x00780001]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Frequent and too long acceleration processes with overcurrent.	Check drive dimensioning.

Motor load I²xt > C00127 [0x00780002]

Response (Lenze setting printed in bold)	**Setting:** C00606 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information	
Cause	**Remedy**
Motor is thermally overloaded, e.g. due to: • impermissible continuous current • frequent or too long acceleration processes	• Check drive dimensioning. • Check setting under C00127.

Motor load I²xt > C00120 [0x00780003]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Motor is thermally overloaded, e.g. due to: • impermissible continuous current • frequent or too long acceleration processes	• Check drive dimensioning. • Check setting under C00120.

Control card is defect (UB24) [0x00780004]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Device error	Please contact Lenze.

Control card is defect (VCC15) [0x00780005]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Device error	Please contact Lenze.

Control card is defect (UB8) [0x00780006]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Device error	Please contact Lenze.

Control card is defect (VCC15 neg.) [0x00780007]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Device error	Please contact Lenze.

Control card is defect (UB18 neg.) [0x00780008]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Device error	Please contact Lenze.

Control card is defect (VCC5) [0x00780009]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Device error	Please contact Lenze.

Electronic nameplate: Data are incompatible [0x0078000a]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
The connected motor with feedback is not supported by the controller firmware.	Check drive dimensioning.

Device command transferred incorrectly [0x00790000]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	If the error occurs frequently, please contact Lenze.

Time error - controller interface [0x00790001]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	If the error occurs frequently, please contact Lenze.

Violation of time slice [0x00790002]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	If the error occurs frequently, please contact Lenze.

Motor: Calculated motor impedance unrealistic [0x007b0001]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters.

Motor: Calculated mutual inductance unrealistic [0x007b0002]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters.

Motor data are inconsistent [0x007b0003]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters.

Motor: Phase resistance too high [0x007b0004]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters.

Motor: Device current too low for rated magnetisation [0x007b0006]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Controller current is too low for rated magnetisation, i.e. the controller cannot energise the motor sufficiently.	Check drive dimensioning.

Motor: Rated current &lt rated magnetising current [0x007b0007]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters and setting of C00022.

Motor: Calculated rotor resistance unrealistic [0x007b0009]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters.

Motor: Calculated mutual inductance unrealistic [0x007b000a]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters.

Motor: Calculated e. m. f. factor unrealistic [0x007b000b]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters.

Motor: Calculated rotor time constant unrealistic [0x007b000c]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters.

Motor: Calculated flux factor unrealistic [0x007b000d]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Faulty motor parameterisation.	Check motor parameters.

DC bus overvoltage [0x007b000e]

Response (Lenze setting printed in bold)	**Setting:** C00600 (☑ Adjustable response)
☐ None ☐ System fault ☑ Fault ☒ **Trouble** ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	**Remedy**
Due to a too high braking energy, the DC-bus voltage exceeds the overvoltage threshold which results from the mains voltage setting in C00173.	• Use brake resistor or regenerative module. • Check setting under C00173.

DC bus undervoltage [0x007b000f]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☒ **Trouble** ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
DC bus voltage is lower than the undervoltage threshold resulting from the mains setting under C00173.	• Check mains voltage. • Check setting under C00173.

Overcurrent detected [0x007b0010]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
• Short circuit/earth fault in motor cable. • Excessive capacitive charging current in the motor cable.	• Check motor cable. • Use motor cable that is shorter or has a lower capacitance.

Earth fault detected [0x007b0011]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
• Earth fault in motor cable. • Excessive capacitive charging current in the motor cable.	• Check motor cable. • Use motor cable that is shorter or has a lower capacitance.

Actual speed value outside the tolerance (C00576) [0x007b0012]

Response (Lenze setting printed in bold)	**Setting:** C00579 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
Difference between actual and setpoint speed is too big.	• Increase speed tolerance margin under C00576. • Check drive dimensioning.

Motor control: Task overflow [0x007b0013]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
Internal error (motor control).	Please contact Lenze.

Internal communication error (host MCTRL) [0x007b0014]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Internal error (motor control).	Please contact Lenze.

Motor data are inconsistent [0x007b0017]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	

Cause	Remedy
Faulty motor parameterisation.	Check motor parameters.

Resolver: Open circuit [0x007b0018]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	

Cause	Remedy
• Resolver cable interrupted. • Resolver defective.	• Check resolver cable. • Check resolver. • Switch off the monitoring function (C00586="3") if no resolver is used.

Motor: Calculated leakage inductance unrealistic [0x007b0019]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	

Cause	Remedy
Faulty motor parameterisation.	Check motor parameters.

Absolute value encoder: Communication error [0x007b001a]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Absolute value encoder does not send any data or a digital readout has been activated while the machine is coasting down.	• Check supply cable. • Check encoder. • Check voltage supply (C00421).

Encoder: Open circuit [0x007b001b]

Response (Lenze setting printed in bold)	**Setting:** C00580 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
• Encoder cable interrupted. • Encoder defective. • Faulty parameter setting of the encoder. • Interruption of the light path of laser measuring systems. • Insufficient reflection of the light of laser measuring systems. **Note:** The encoder open-circuit monitoring for incremental encoders (C00422 = "0: Incremental encoder (TTL signal)") requires a signal amplitude > 3.5 V! If the signal amplitude is lower than 3.0 V, the error response parameterised in C00580 is triggered.	• Check encoder cable. • Check encoder. • Check parameter setting (C00422). • Switch off the monitoring function (C00580="3") if no encoder is used. • Correct the light path (for laser measuring systems). • Improve the reflection (for laser measuring systems).

Brake transistor: Ixt overload [0x007b001c]

Response (Lenze setting printed in bold)	**Setting:** C00573 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
Too frequent and too long braking operations.	Check drive dimensioning.

Brake resistor: I²xt overload [0x007b001d]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	**Remedy**
Too frequent and too long braking operations.	• Check drive dimensioning. • Check parameter setting (C00129, C00130, C00131).

Motor: Actual current value > C00620 [0x007b001e]

Response (Lenze setting printed in bold)	**Setting:** C00619 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
The instantaneous value of the motor current has exceeded the value set in C00620.	• Set a higher value in C00620. • Reduce maximum current (C00022). • Change response (C00619).

Resolver: Calculated acceleration unrealistic [0x007b001f]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	**Remedy**
Resolver evaluation faulty (implausible acceleration at the resolver).	Check structure.

Motor: Actual speed value > C00596 [0x007b0020]

Response (Lenze setting printed in bold)	Setting: C00607 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	
Cause	**Remedy**
Threshold for speed monitoring set in C00596 has been exceeded.	Check drive dimensioning.

Brake transistor: Overcurrent [0x007b0021]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Brake chopper short circuit/earth fault detected.	Check brake chopper cable and brake resistor.

Position encoder: Module selected in C00490 not available [0x007b0023]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
The position encode selected under C00490 has not been recognised.	• Check position encoder. • Check parameter setting (C00490).

Motor encoder: Module selected in C00495 not available [0x007b0024]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
The motor encoder selected under C00495 has not been recognised.	• Check motor encoder. • Check parameter setting (C00495).

Motor temperature: Module selected in C01193 not available [0x007b0025]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
The module for temperature feedback selected in C01193 has not been recognised.	• Check feedback module. • Check parameter setting (C01193).

EnDat encoder: Lamp error [0x007b0026]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
EnDat encoder defective.	Check EnDat encoder.

EnDat encoder: Signal error [0x007b0027]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
EnDat encoder defective.	Check EnDat encoder.

EnDat encoder: Position error [0x007b0028]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
EnDat encoder defective.	Check EnDat encoder.

EnDat encoder: Overvoltage [0x007b0029]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
EnDat encoder defective.	Check EnDat encoder.

EnDat encoder: Undervoltage [0x007b002a]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
EnDat encoder defective.	Check EnDat encoder.

EnDat encoder: Overcurrent [0x007b002b]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
EnDat encoder defective.	Check EnDat encoder.

EnDat encoder: Battery empty [0x007b002c]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
EnDat encoder defective.	Check EnDat encoder.

Failure of motor phase U [0x007b002d]

Response (Lenze setting printed in bold)	**Setting:** C00597 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	
Cause	**Remedy**
U phase interruption of motor cable.	• Check cabling between the controller and motor. • Check parameter setting (C00599).

Failure of motor phase V [0x007b002e]

Response (Lenze setting printed in bold)	**Setting:** C00597 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	
Cause	**Remedy**
V phase interruption of the motor cable.	• Check cabling between the controller and motor. • Check parameter setting (C00599).

Failure of motor phase W [0x007b002f]

Response (Lenze setting printed in bold)	**Setting:** C00597 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	
Cause	**Remedy**
W phase interruption of the motor cable.	• Check cabling between the controller and motor. • Check parameter setting (C00599).

Electronic nameplate: Data loaded [0x007b0030]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
New electronic nameplate (ENP) has been found.	- (Information only)

Electronic nameplate: Not found [0x007b0031]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Electronic nameplate (ENP) is not available.	- (Information only)

Electronic nameplate: Encoder protocol unknown [0x007b0032]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
The connected motor with feedback is not supported by the controller firmware.	Check drive dimensioning.

Electronic nameplate: Encoder signal unknown [0x007b0033]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
The connected motor with feedback is not supported by the controller firmware.	Check drive dimensioning.

Internal communication error (DMA) [0x007b0034]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Internal error.	Please contact Lenze.

Controller: Overload during acceleration phases [0x007b0035]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Frequent or too long acceleration processes.	• Check drive dimensioning. • Reduce steepness of acceleration ramps.

Internal communication error (MCTRL host) [0x007b0036]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Internal error.	Please contact Lenze.

PLC configuration invalid [0x007b0037]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
An invalid control configuration has occurred.	Load a different application.

Motor parameter identification was cancelled [0x007b0038]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
The motor current during identification was too high.	• The motor must not move during identification. • Check motor parameters.

Electronic nameplate: Data outside the parameter limits [0x007b0039]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
The motor parameters of the electronic nameplate are outside of the limit values of the controller and therefore cannot be accepted.	Please contact Lenze.

Hiperface encoder: Transmission error [0x007b003a]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	**Remedy**
• Encoder signal interferences (EMC). • Encoder cable interrupted. • Encoder defective. • Faulty parameter setting of the encoder. **Note:** This information is always supplied together with the "I/O error - encoder communication" error message and is entered in the logbook even if the encoder monitoring is deactivated (C00580).	• Check encoder cable, if required, use shorter encoder cable. • Check encoder. • Check parameter setting (C00420, C00421, C00422).

Hiperface encoder: Command error [0x007b003b]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	**Remedy**
• Encoder signal interferences (EMC). • Encoder cable interrupted. • Encoder defective. • Faulty parameter setting of the encoder. **Note:** This information is always supplied together with the "I/O error - encoder communication" error message and is entered in the logbook even if the encoder monitoring is deactivated (C00580).	• Check encoder cable, if required, use shorter encoder cable. • Check encoder. • Check parameter setting (C00420, C00421, C00422).

Hiperface encoder: Unknown encoder [0x007b003c]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	**Remedy**
Encoder is not supported by the controller firmware.	• Exchange encoder for known type. • From software version V4.0: Parameterise the unknown encoder with the codes C00414 (type code) and C00415 (number of revolutions).

Hiperface encoder: Position initialisation error [0x007b003d]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	**Remedy**
A digital readout of the absolute encoder is only possible at standstill. The readout is activated by the following actions: • Mains switching • Change of C00420, C00422 and C00427 • To "Absolute value encoder: Communication error" • To "Encoder: Open circuit" **Note:** This information is always supplied together with the "I/O error - encoder communication" error message and is entered in the logbook even if the encoder monitoring is deactivated (C00580).	Prevent coasting of the machine while the absolute encoder is read out.

Encoder monitoring: Pulse deviation detected [0x007b003e]

Response (Lenze setting printed in bold)	**Setting:** C00621 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	

Cause	**Remedy**
• Encoder signal interferences (EMC). • Encoder cable interrupted. • Encoder defective. • Faulty parameter setting of the encoder.	• Check encoder cable, if required, use shorter encoder cable. • Check encoder. • Check parameter setting (C00422). • Possibly switch off monitoring (C00621).

Failure of a mains phase [0x007b003f]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
Missing mains phase. **Note:** This monitoring is only available for devices ≥ 75 kW (type 8S and bigger).	Check mains connection.

Brake chopper: Ixt > C00570 [0x007b0040]

Response (Lenze setting printed in bold)	**Setting:** C00569 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☒ **Warning** ☐ Information	

Cause	**Remedy**
Frequent and too long braking.	• Check drive dimensioning. • Check setting in C00570.

Brake resistor: I2t > C00572 [0x007b0041]

Response (Lenze setting printed in bold)	**Setting:** C00571 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ **Warning locked** ☒ Warning ☐ Information	

Cause	**Remedy**
Frequent and too long braking.	• Check drive dimensioning. • Check setting in C00572. • Check settings for brake resistor (C00129, C00130, C00131).

Power section is defect [0x007b0042]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
Device error	Mains switching • Please contact Lenze if the problem occurs again.

Controller: Clamp operation [0x007b0047]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Actual motor current is higher than the maximum device current (C00789).	• Increase speed setpoint ramps. • Optimise Imax controller.

Pole position recognition cancelled [0x007b004a]

Response (Lenze setting printed in bold)	**Setting:** C00640 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
From software version V4.0 An error occurred during the pole position identification. The pole position identification could not be completed successfully.	• Check whether all requirements for an identification of the pole position are fulfilled. • Ensure that the machine is not braked or blocked during the pole position identification. • Repeat the pole position identification.

Motor switched off [0x007b004c]

Response (Lenze setting printed in bold)	**Setting:** C00597 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information	
Cause	**Remedy**
From software version V5.0 Interruption of several motor cable phases.	Check cabling between the controller and motor.

Encoder: Time-out error [0x007b004d]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**

EnDat encoder: Transmission error [0x007b004e]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
• Encoder signal interferences (EMC). • Encoder cable interrupted. • Encoder defective. • Faulty parameter setting of the encoder. **Note:** This information is always supplied together with the "I/O error - encoder communication" error message and is entered in the logbook even if the encoder monitoring is deactivated (C00580).	• Check encoder cable, if required, use shorter encoder cable. • Check encoder. • Check parameter setting (C00420, C00421, C00422).

EnDat encoder: Command error [0x007b004f]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
• Encoder signal interferences (EMC). • Encoder cable interrupted. • Encoder defective. • Faulty parameter setting of the encoder. **Note:** This information is always supplied together with the "I/O error - encoder communication" error message and is entered in the logbook even if the encoder monitoring is deactivated (C00580).	• Check encoder cable, if required, use shorter encoder cable. • Check encoder. • Check parameter setting (C00420, C00421, C00422).

EnDat encoder: Position initialisation error [0x007b0050]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
A digital readout of the absolute encoder is only possible at standstill. The readout is activated by the following actions: • Mains switching • Change of C00420, C00422 and C00427 • To "Absolute value encoder: Communication error" • To "Encoder: Open circuit" **Note:** This information is always supplied together with the "I/O error - encoder communication" error message and is entered in the logbook even if the encoder monitoring is deactivated (C00580).	Prevent coasting of the machine while the absolute encoder is read out.

Memory module: File system has been formatted [0x007c0000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
File system of the memory module has been formatted.	- (Information only)

Memory module: File system has been restored [0x007c0001]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
File system of the memory module has been restored.	- (Information only)

Analog input 1: Master current < 4 mA [0x007d0000]

Response (Lenze setting printed in bold)	Setting: C00598 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	
Cause	**Remedy**
Master current is in the impermissible range -4 ... +4 mA, e. g. due to a cable break or a defective master current value encoder. • Only for parameterisation as master current input (see C00034).	Remove cable break.

PLC configuration invalid [0x007d0001]

Response (Lenze setting printed in bold)

☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
An invalid control configuration has occurred.	Load a different application.

Communication error between device and device module [0x007f0002]

Response (Lenze setting printed in bold) **Setting:** C01501 (☑ Adjustable response)

☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☐ Information

Cause	Remedy
Communication between the controller and extension module is interrupted, e.g. due to disturbances in the ambience (EMC), defective hardware, or loose contact. • This monitoring is designed for safe process data communication.	• Remove EMC fault. • Switch off the controller, plug in module correctly and switch on the controller again. • Switch the mains or restart controller. • Exchange module/controller. • Please contact Lenze if the problem occurs again.

Communication with module in MXI1 interrupted [0x007f0003]

Response (Lenze setting printed in bold)

☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
Communication between the controller and the extension module in module slot MXI1 cannot be established.	• Switch off controller, plug module correctly in module slot MXI1, switch on controller again. • If the problem occurs again, replace the module.

Communication with module in MXI2 interrupted [0x007f0004]

Response (Lenze setting printed in bold)

☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
Communication between the controller and the extension module in module slot MXI2 cannot be established.	• Switch off controller, plug module correctly in module slot MXI2, switch on controller again. • If the problem occurs again, replace the module.

CAN on board: Bus off [0x00830000]

Response (Lenze setting printed in bold) **Setting:** C00595 (☑ Adjustable response)

☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information

Cause	Remedy
CAN on board: "Bus-Off" state • Too many faulty telegrams received. • Defective cable (e.g. loose contact). • Two nodes with the same ID.	• Remove fault (e.g. EMC). • Remove loose contact, screw down adapter. • Assign different node IDs.

CAN on board: Invalid node address 0 [0x00830001]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information

Cause	Remedy
CAN on board: initialisation error • The hardware allocation of the node address was selected via DIP switches, and the DIP switches of the node address are all on zero. **Note:** Instead of the impermissible node address 0, node address 1 is used.	• Set a non-zero node address by means of the DIP switches and then switch mains. • Activation of the software allocation of the node number by switching over DIP switch 2, then switch mains.

CAN on board: Basic configuration invalid [0x00830002]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CAN on board: configuration error • Faulty download of an Engineer or PLC Designer project • Invalid CAN settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN on board: Heartbeat error index 1 ... 32 [0x00840000 ... 0x0084001f]

Response (Lenze setting printed in bold)	**Setting:** C00613/1...32 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CAN on board: Cyclic node monitoring • Node station has not received a heartbeat telegram from node 1 ... 32 within the defined time.	• Reactivate CAN node by mains switching, restart of the controller (C00002="11000") or CAN reset node. • Select a different heartbeat producer monitoring time or switch off monitoring and reset locked error status, if necessary. Tip: Save the current parameter set before mains switching and restart of the controller (C00002="11").

CAN on board: Lifeguarding error [0x00840020]

Response (Lenze setting printed in bold)	**Setting:** C00614 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CAN on board: Cyclic node monitoring • Slave response: Maximum time between two node guarding telegrams (remote transmission request telegram) from the master has been exceeded.	Select a different Lifeguarding monitoring time or switch off monitoring.

CAN on board: Faulty NMT slave configuration [0x00840021]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CAN on board: A configuration error has occurred in the network management of the CAN slave. • Faulty download of an Engineer or PLC Designer project • Invalid CAN settings according to DS301V402 in the Engineer or PLC Designer. • Incorrect parameterisation of node guarding or heartbeat.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN on board: Faulty emergency configuration [0x00850000]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CAN on board: CAN emergency configuration is faulty. • Faulty download of an Engineer or PLC Designer project • Invalid CAN emergency settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN on board: Node guarding error 1 ... 32 [0x00860000 ... 0x0086001f]

Response (Lenze setting printed in bold)	**Setting:** C00612/1...32 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CAN on board: Cyclic node monitoring • CAN master has not received a response to a node guarding telegram (remote transmission request telegram) from node 1 ... 32 within the defined time.	• Reactivate CAN node by mains switching, restart of the controller (C00002="11000") or CAN reset node. • Select a different node guarding monitoring time or switch off monitoring. • Reset potentially caught error status. Tip: Save the current parameter set before mains switching and restart of the controller (C00002="11").

CAN on board: Faulty NMT master configuration [0x00860020]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CAN on board: A configuration error has occurred in the network management of the CAN master. • Faulty download of an Engineer or PLC Designer project. • Invalid CAN master settings according to DS301V402 and DS405 in the Engineer or PLC Designer. • Incorrect parameterisation of node guarding or heartbeat.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN on board RPDO1: Telegram not received or faulty [0x00870000]

Response (Lenze setting printed in bold)	**Setting:** C00591/1 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CAN on board: CAN-IN 1 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN on board RPDO2: Telegram not received or faulty [0x00870001]

Response (Lenze setting printed in bold)	**Setting:** C00591/2 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CAN on board: CAN-IN 2 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN on board RPDO3: Telegram not received or faulty [0x00870002]

Response (Lenze setting printed in bold)	**Setting:** C00591/3 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CAN on board: CAN-IN 3 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN on board RPDO4: Telegram not received or faulty [0x00870003]

Response (Lenze setting printed in bold)	**Setting:** C00591/4 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CAN on board: CAN-IN 4 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN on board PDO manager: Faulty configuration [0x00870008]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CAN on board: CAN-PDO configuration error • Faulty project download. • Invalid CAN settings according to DS301V402 in the Engineer or PLC Designer. • Mapping variables have incorrect CANopen indices according to DS405.	• Repeat download. • Correct CAN settings in the project and regenerate project.

CAN on board SDO server: Faulty configuration [0x00880000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information	
Cause	**Remedy**
CAN on board: A configuration error has occurred in the CAN SDO server. • Faulty project download. • Invalid SDO server settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download. • Correct CAN settings in the project and regenerate project.

CAN on board SDO client: Faulty configuration [0x00890000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information	
Cause	**Remedy**
CAN on board: A configuration error has occurred in the CAN SDO client. • Faulty project download. • Invalid SDO client settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download • Correct CAN settings in the project and regenerate project.

File ProjectSelection.dat defect [0x008c0000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	Reformat memory module (C00002="1030") and repeat project download.

File ProjectList.dat defect [0x008c0001]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	Reformat memory module (C00002="1030") and repeat project download.

File DeviceCFG.dat defect [0x008c0002]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	Reformat memory module (C00002="1030") and repeat project download.

File ProjectSelection.dat is missing [0x008c0003]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	Reformat memory module (C00002="1030") and repeat project download.

File ProjectList.dat is missing [0x008c0004]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	**Remedy**
Internal error	Reformat memory module (C00002="1030") and repeat project download.

File DeviceCFG.dat is missing [0x008c0005]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	**Remedy**
Internal error	Reformat memory module (C00002="1030") and repeat project download.

File ProjectSelection.dat invalid [0x008c0006]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	**Remedy**
Internal error	Reformat memory module (C00002="1030") and repeat project download.

File ProjectList.dat invalid [0x008c0007]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	**Remedy**
Internal error	Reformat memory module (C00002="1030") and repeat project download.

File DeviceCFG.dat invalid [0x008c0008]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	**Remedy**
Internal error	Reformat memory module (C00002="1030") and repeat project download.

Project is not loaded [0x008c0009]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	**Remedy**
Application could not be loaded because of a file error.	Load new or different application.

Project is not available [0x008c000a]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Application not available.	• Download application with the Engineer • Switch off controller and use a different memory module with an existing application.

Required licence missing [0x008c000b]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Memory module could not be initialised.	Two possibilities: • Use the Engineer to download and activate an application suitable for the memory module. • Switch off controller and use memory module suitable for the application.

Application and device are incompatible [0x008c000c]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Application is incompatible with the controller.	• Download of an application suitable for the controller using the Engineer. • Switch off controller and use a different memory module with suitable application.

MXI1: Module is missing or incompatible [0x008c000d]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Extension module in module slot MXI1 is incompatible with the application.	Use extension module supported by the application.

MXI2: Module is missing or incompatible [0x008c000e]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Extension module in module slot MXI2 is incompatible with the application.	Use extension module supported by the application.

MXI1: PROFIBUS module is missing or incompatible [0x008c000f]

Response (Lenze setting printed in bold)	**Setting:** C00615/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
E94AYCPM communication module (PROFIBUS) in module slot MXI1 is incompatible with the application.	Use communication module supported by the application.

MXI2: PROFIBUS module is missing or incompatible [0x008c0010]

Response (Lenze setting printed in bold)	**Setting:** C00615/3 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	**Remedy**
E94AYCPM communication module (PROFIBUS) in module slot MXI2 is incompatible with the application.	Use communication module supported by the application.

MXI1: Ethernet module is missing or incompatible [0x008c0011]

Response (Lenze setting printed in bold)	**Setting:** C00615/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	**Remedy**
E94AYCEN communication module (Ethernet) in module slot MXI1 is incompatible with the application.	Use communication module supported by the application.

MXI2: Ethernet module is missing or incompatible [0x008c0012]

Response (Lenze setting printed in bold)	**Setting:** C00615/3 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	**Remedy**
E94AYCEN communication module (Ethernet) in module slot MXI2 is incompatible with the application.	Use communication module supported by the application.

MXI1: Digital frequency module is missing or incompatible [0x008c0013]

Response (Lenze setting printed in bold)	**Setting:** C00615/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	**Remedy**
Digital frequency extension module in module slot MXI1 is incompatible with the application.	Use extension module supported by the application.

MXI2: Digital frequency module is missing or incompatible [0x008c0014]

Response (Lenze setting printed in bold)	**Setting:** C00615/3 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	**Remedy**
Digital frequency extension module in module slot MXI2 is incompatible with the application.	Use extension module supported by the application.

MXI1: ICM module is missing or incompatible [0x008c0015]

Response (Lenze setting printed in bold)	**Setting:** C00615/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	**Remedy**
ICM extension module in module slot MXI1 is incompatible with the application.	Use extension module supported by the application.

MXI2: ICM module is missing or incompatible [0x008c0016]

Response (Lenze setting printed in bold)	**Setting:** C00615/3 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	
Cause	**Remedy**
ICM extension module in module slot MXI2 is incompatible with the application.	Use extension module supported by the application.

MXI1: CAN module is missing or incompatible [0x008c0017]

Response (Lenze setting printed in bold)	**Setting:** C00615/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	
Cause	**Remedy**
CANopen communication module in module slot MXI1 is incompatible with the application.	Use communication module supported by the application.

MXI2: CAN module is missing or incompatible [0x008c0018]

Response (Lenze setting printed in bold)	**Setting:** C00615/3 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	
Cause	**Remedy**
CANopen communication module in module slot MXI2 is incompatible with the application.	Use communication module supported by the application.

ConnectTable active [0x008c001a]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
The application is provided with a so-called connection table, i. e. connections can be altered "online" in the function block editor without having to carry out a new complete download.	- (Information only)

Parameter set faulty [0x00900000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Parameter set is invalid.	Transfer parameter set from Engineer to the controller and save with C00002="11".

Lenze setting loaded [0x00900001]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Lenze setting has been loaded.	- (Information only)

Parameter set saved [0x00900002]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
Parameter set has been saved.	- (Information only)

Parameter set loaded [0x00900003]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
Parameter set has been loaded.	- (Information only)

Loading of Lenze setting failed [0x00900004]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Lenze setting of a parameter is not within the valid limits.	Eliminate error in the application and retransfer application to controller.

Parameter set restored [0x00900005]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
An error has occurred while loading the selected parameter set.	Transfer parameter set from Engineer to the controller and save with C00002="11".

Saving of parameters failed [0x00900006]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
An error has occurred while saving the current parameter set.	Use a different memory module.

Parameter set: Version conflict [0x00900007]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
The parameter set version on the memory module is not compatible with the firmware of the controller.	Transfer parameter set from Engineer to the controller and save with C00002="11".

Code number duplicated [0x00900008]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information	
Cause	**Remedy**
Code number of the operating system has been assigned to the application.	Eliminate error in the application and retransfer application to controller.

Parameter set: Type of standard device has been changed [0x00900009]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
The firmware has loaded a parameter set the type code of which does not correspond to the type code of the controller.	Load a suitable parameter set.

No parameters for module in MXI1 [0x0090000a]

Response (Lenze setting printed in bold)	**Setting:** C00615/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	
Cause	**Remedy**
The parameter set contains no parameters for the module inserted in MXI1.	Integrate the module inserted in MXI1 into the Engineer project and then retransmit the parameter set to the controller.

No parameters for module in MXI2 [0x0090000b]

Response (Lenze setting printed in bold)	**Setting:** C00615/3 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	
Cause	**Remedy**
The parameter set contains no parameters for the module inserted in MXI2.	Integrate the module inserted in MXI2 into the Engineer project and then retransmit the parameter set to the controller.

Disconnection in the case of par. storage [0x0090000c]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**

Mains voltage is switched on [0x00910000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Mains voltage has been switched on.	- (Information only)

Mains voltage is switched off [0x00910001]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	
Cause	**Remedy**
Mains voltage has been switched off.	- (Information only)

No heartbeat signal detected [0x00910002]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	If the error occurs frequently, please contact Lenze.

Heartbeat not periodic [0x00910003]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	If the error occurs frequently, please contact Lenze.

Internal error: See C00180 [0x00910004]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	If the error occurs frequently, please contact Lenze.

Internal error: See C00180 [0x00910005]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	Please contact Lenze.

Internal error: See C00180 [0x00910006]

Response (Lenze setting printed in bold)	
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Divisor of division was "0".	Replace application.

Internal error: See C00180 [0x00910008]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
Internal error	If the error occurs frequently, please contact Lenze.

Internal error: See C00180 [0x00910009]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
Internal error	If the error occurs frequently, please contact Lenze.

System task 1: Task overflow [0x0091000a]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
System overload.	Please contact Lenze.

System task 2: Task overflow [0x0091000b]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	**Remedy**
System overload.	Please contact Lenze.

System task 3: Task overflow [0x0091000c]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
System overload.	Please contact Lenze.

System task: Task overflow [0x0091000d]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
System overload.	Please contact Lenze.

Communication task: Standstill > 3 s [0x0091000e]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
System overload or communication task crash.	Reduce system load. • This is possible in the application or data transfer of the communication interfaces.

Cyclic task: Standstill > 60 s [0x0091000f]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
System overload or CRC check task crash.	Reduce system load. • This is possible in the application or data transfer of the communication interfaces.

Position value faulty [0x00910010]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Internal error	If the error occurs frequently, please contact Lenze.

Error during initialisation [0x00910011]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Internal error	If the error occurs frequently, please contact Lenze.

Block function in wrong MEC task [0x00910012]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
Internal error	If the error occurs frequently, please contact Lenze.

Safety module: Incompatible with setting in C00214 [0x00920000]

Response (Lenze setting printed in bold)
☐ None ☒ **System fault** ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
The controller has detected a safety module which does not match the setting under C00214.	Change setting underC00214 or use a suitable safety module. • Afterwards mains switching is required.

Communication with safety module interrupted [0x00920001]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
It is not possible to establish communication between the controller and safety module.	• Switch off the controller, plug in safety module correctly and switch on the controller again. • If the problem occurs again, replace the safety module.

DFIN (MXI1): Track error A-/A [0x00990000]

Response (Lenze setting printed in bold)	**Setting:** C13040 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	**Remedy**
Digital frequency extension module in MXI1: Interruption (open circuit) of the signal cable for track A.	• Check signal cable for track A. • Check encoder.

DFIN (MXI1): Track error B-/B [0x00990001]

Response (Lenze setting printed in bold)	**Setting:** C13040 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	**Remedy**
Digital frequency extension module in MXI1: Interruption (open circuit) of the signal cable for track B.	• Check signal cable for track B. • Check encoder.

DFIN (MXI1): Track error Z-/Z [0x00990002]

Response (Lenze setting printed in bold)	**Setting:** C13040 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	**Remedy**
Digital frequency extension module in MXI1: Interruption (open circuit) of the signal cable for track Z.	• Check signal cable for track Z. • Check encoder.

DFIN (MXI1): Signal error enable/lamp control [0x00990003]

Response (Lenze setting printed in bold)	**Setting:** C13041 (☑ Adjustable response)
☑ None ☐ System fault ☑☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☒ **Warning** Information	

Cause	**Remedy**
Digital frequency extension module in MXI1: Interruption (open circuit) of the signal cable for the "Enable" signal or no "Enable" signal available.	• Check signal cable for "Enable" signal. • Check encoder.

DFIN (MXI1): Supply cannot be corrected anymore [0x00990004]

Response (Lenze setting printed in bold)	**Setting:** C13042 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☒ **Warning** ☑ Information	

Cause	**Remedy**
Digital frequency extension module in MXI1: The encoder voltage controlled by the digital frequency input has reached the voltage limit.	Check encoder.

DFOUT (MXI1): Maximum frequency reached [0x00990005]

Response (Lenze setting printed in bold)	**Setting:** C13080 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☒ **Warning** ☑ Information	

Cause	**Remedy**
Digital frequency extension module in MXI1: Limit frequency at the digital frequency output reached. • The digital frequency has reached the limit value set in C013053.	Check limit value set.

CAN module (MXI1): Bus off [0x009d0000]

Response (Lenze setting printed in bold)	**Setting:** C13595 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	

Cause	Remedy
CANopen communication module in MXI1: "Bus-off" state • Too many faulty telegrams received. • Defective cable (e.g. loose contact). • Two nodes with the same ID.	• Remove fault (e.g. EMC). • Remove loose contact, screw down adapter. • Assign different node IDs.

CAN module (MXI1): Invalid node address 0 [0x009d0001]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information

Cause	Remedy
CANopen communication module in MXI1: Initialisation error • The hardware allocation of the node address was selected via DIP switches, and the DIP switches of the node address are all on zero. **Note:** Instead of the impermissible node address 0, node address 1 is used.	• Set a non-zero node address by means of the DIP switches and then switch mains. • Activation of the software allocation of the node number by switching over DIP switch 2, then switch mains.

CAN module (MXI1): Basic configuration invalid [0x009d0002]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI1: Configuration error • Faulty download of an Engineer or PLC Designer project • Invalid CAN settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN module (MXI1): Heartbeat error index 1 ... 32 [0x009e0000 ... 0x009e001f]

Response (Lenze setting printed in bold)	**Setting:** C13613/1...32 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: Cyclic node monitoring • Node station has not received a heartbeat telegram from node 1 ... 32 within the defined time.	• Reactivate CAN node by mains switching, restart of the controller (C00002="11000") or CAN reset node. • Select a different heartbeat producer monitoring time or switch off monitoring and reset locked error status, if necessary. Tip: Save the current parameter set before mains switching and restart of the controller (C00002="11").

CAN module (MXI1): Lifeguarding error [0x009e0020]

Response (Lenze setting printed in bold)	**Setting:** C13614 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: Cyclic node monitoring • Slave response: Maximum time between two node guarding telegrams (remote transmission request telegram) from the master has been exceeded.	Select a different Lifeguarding monitoring time or switch off monitoring.

CAN module (MXI1): Faulty NMT slave configuration [0x009e0021]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI1: A configuration error has occurred in the network management of the CAN slave. • Faulty download of an Engineer or PLC Designer project • Invalid CAN settings according to DS301V402 in the Engineer or PLC Designer. • Incorrect parameterisation of node guarding or heartbeat.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN module (MXI1): Faulty emergency configuration [0x009f0000]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI1: CAN emergency configuration is faulty. • Faulty download of an Engineer or PLC Designer project • Invalid CAN emergency settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN module (MXI1): Node guarding error 1 ... 32 [0x00a00000 ... 0x00a0001f]

Response (Lenze setting printed in bold)	**Setting:** C13612/1...32 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: Cyclic node monitoring • CAN master has not received a response to a node guarding telegram (remote transmission request telegram) from node 1 ... 32 within the defined time.	• Reactivate CAN node by mains switching, restart of the controller (C00002="11000") or CAN reset node. • Select a different node guarding monitoring time or switch off monitoring. • Reset potentially caught error status. Tip: Save the current parameter set before mains switching and restart of the controller (C00002="11").

CAN module (MXI1): Faulty NMT master configuration [0x00a00020]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI1: A configuration error has occurred in the network management of the CAN master. • Faulty download of an Engineer or PLC Designer project. • Invalid CAN master settings according to DS301V402 and DS405 in the Engineer or PLC Designer. • Incorrect parameterisation of node guarding or heartbeat.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN module (MXI1) RPDO1: Telegram not received or faulty [0x00a10000]

Response (Lenze setting printed in bold)	**Setting:** C13591/1 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: CAN-IN 1 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI1) RPDO2: Telegram not received or faulty [0x00a10001]

Response (Lenze setting printed in bold)	**Setting:** C13591/2 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: CAN-IN 2 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI1) RPDO3: Telegram not received or faulty [0x00a10002]

Response (Lenze setting printed in bold)	**Setting:** C13591/3 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: CAN-IN 3 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI1) RPDO4: Telegram not received or faulty [0x00a10003]

Response (Lenze setting printed in bold)	**Setting:** C13591/4 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: CAN-IN 4 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI1) RPDO5: Telegram not received or faulty [0x00a10004]

Response (Lenze setting printed in bold)	**Setting:** C13591/5 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: CAN-IN 5 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI1) RPDO6: Telegram not received or faulty [0x00a10005]

Response (Lenze setting printed in bold)	**Setting:** C13591/6 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: CAN-IN 6 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI1) RPDO7: Telegram not received or faulty [0x00a10006]

Response (Lenze setting printed in bold)	**Setting:** C13591/7 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: CAN-IN 7 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI1) RPDO8: Telegram not received or faulty [0x00a10007]

Response (Lenze setting printed in bold)	**Setting:** C13591/8 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI1: CAN-IN 8 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI1) PDO manager: Faulty configuration [0x00a10008]

Response (Lenze setting printed in bold)

☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI1: CAN-PDO configuration error • Faulty project download. • Invalid CAN settings according to DS301V402 in the Engineer or PLC Designer. • Mapping variables have incorrect CANopen indices according to DS405.	• Repeat download. • Correct CAN settings in the project and regenerate project.

CAN module (MXI1) SDO server: Faulty configuration [0x00a20000]

Response (Lenze setting printed in bold)

☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI1: In the CAN SDO server a configuration error has occurred. • Faulty project download. • Invalid SDO server settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download. • Correct CAN settings in the project and regenerate project.

CAN module (MXI1) SDO client: Faulty configuration [0x00a30000]

Response (Lenze setting printed in bold)

☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI1: In the CAN SDO client a configuration error has occurred. • Faulty project download. • Invalid SDO client settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download • Correct CAN settings in the project and regenerate project.

DFIN (MXI2): Track error A-/A [0x00aa0000]

Response (Lenze setting printed in bold) **Setting:** C14040 (☑ Adjustable response)

☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information

Cause	Remedy
Digital frequency extension module in MXI2: Interruption (open circuit) of the signal cable for track A.	• Check signal cable for track A. • Check encoder.

DFIN (MXI2): Track error B-/B [0x00aa0001]

Response (Lenze setting printed in bold) **Setting:** C14040 (☑ Adjustable response)

☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information

Cause	Remedy
Digital frequency extension module in MXI2: Interruption (open circuit) of the signal cable for track B.	• Check signal cable for track B. • Check encoder.

DFIN (MXI2): Track error Z-/Z [0x00aa0002]

Response (Lenze setting printed in bold)	**Setting:** C14040 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	
Cause	**Remedy**
Digital frequency extension module in MXI2: Interruption (open circuit) of the signal cable for track Z.	• Check signal cable for track Z. • Check encoder.

DFIN (MXI2): Signal error enable/lamp control [0x00aa0003]

Response (Lenze setting printed in bold)	**Setting:** C14041 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☒ **Warning** ☑ Information	
Cause	**Remedy**
Digital frequency extension module in MXI2: Interruption (open circuit) of the signal cable for the "Enable" signal or no "Enable" signal available.	• Check signal cable for "Enable" signal. • Check encoder.

DFIN (MXI2): Supply cannot be corrected anymore [0x00aa0004]

Response (Lenze setting printed in bold)	**Setting:** C14042 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☒ **Warning** ☑ Information	
Cause	**Remedy**
Digital frequency extension module in MXI2: The encoder voltage controlled by the digital frequency input has reached the voltage limit.	Check encoder.

DFOUT (MXI2): Maximum frequency reached [0x00aa0005]

Response (Lenze setting printed in bold)	**Setting:** C14080 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☒ **Warning** ☑ Information	
Cause	**Remedy**
Digital frequency extension module in MXI2: Limit frequency at the digital frequency output reached. • The digital frequency has reached the limit value set in C014053.	Check limit value set.

CAN module (MXI2): Bus off [0x00ac0000]

Response (Lenze setting printed in bold)	**Setting:** C14595 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	
Cause	**Remedy**
CANopen communication module in MXI2: "Bus-off" state • Too many faulty telegrams received. • Defective cable (e.g. loose contact). • Two nodes with the same ID.	• Remove fault (e.g. EMC). • Remove loose contact, screw down adapter. • Assign different node IDs.

CAN module (MXI2): Invalid node address 0 [0x00ac0001]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information

Cause	Remedy
CANopen communication module in MXI2: Initialisation error • The hardware allocation of the node address was selected via DIP switches, and the DIP switches of the node address are all on zero. **Note:** Instead of the impermissible node address 0, node address 1 is used.	• Set a non-zero node address by means of the DIP switches and then switch mains. • Activation of the software allocation of the node number by switching over DIP switch 2, then switch mains.

CAN module (MXI2): Basic configuration invalid [0x00ac0002]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI2: Configuration error • Faulty download of an Engineer or PLC Designer project • Invalid CAN settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN module (MXI2): Heartbeat error index 1 ... 32 [0x00ad0000 ... 0x00ad001f]

Response (Lenze setting printed in bold)	Setting: C14613/1...32 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: Cyclic node monitoring • Node station has not received a heartbeat telegram from node 1 ... 32 within the defined time.	• Reactivate CAN node by mains switching, restart of the controller (C00002="11000") or CAN reset node. • Select a different heartbeat producer monitoring time or switch off monitoring and reset locked error status, if necessary. Tip: Save the current parameter set before mains switching and restart of the controller (C00002="11").

CAN module (MXI2): Lifeguarding error [0x00ad0020]

Response (Lenze setting printed in bold)	Setting: C14614 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: Cyclic node monitoring • Slave response: Maximum time between two node guarding telegrams (remote transmission request telegram) from the master has been exceeded.	Select a different Lifeguarding monitoring time or switch off monitoring.

CAN module (MXI2): Faulty NMT slave configuration [0x00ad0021]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information	
Cause	**Remedy**
CANopen communication module in MXI2: A configuration error has occurred in the network management of the CAN slave. • Faulty download of an Engineer or PLC Designer project • Invalid CAN settings according to DS301V402 in the Engineer or PLC Designer. • Incorrect parameterisation of node guarding or heartbeat.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN module (MXI2): Faulty emergency configuration [0x00ae0000]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information	
Cause	**Remedy**
CANopen communication module in MXI2: CAN emergency configuration is faulty. • Faulty download of an Engineer or PLC Designer project • Invalid CAN emergency settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN module (MXI2): Node guarding error 1 ... 32 [0x00af0000 ... 0x00a0001f]

Response (Lenze setting printed in bold)	**Setting:** C14612/1...32 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	
Cause	**Remedy**
CANopen communication module in MXI2: Cyclic node monitoring • CAN master has not received a response to a node guarding telegram (remote transmission request telegram) from node 1 ... 32 within the defined time.	• Reactivate CAN node by mains switching, restart of the controller (C00002="11000") or CAN reset node. • Select a different node guarding monitoring time or switch off monitoring. • Reset potentially caught error status. Tip: Save the current parameter set before mains switching and restart of the controller (C00002="11").

CAN module (MXI2): Faulty NMT master configuration [0x00af0020]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information	
Cause	**Remedy**
CANopen communication module in MXI2: A configuration error has occurred in the network management of the CAN master. • Faulty download of an Engineer or PLC Designer project. • Invalid CAN master settings according to DS301V402 and DS405 in the Engineer or PLC Designer. • Incorrect parameterisation of node guarding or heartbeat.	• Repeat download • Correct CAN settings in the project and regenerate project.

CAN module (MXI2) RPDO1: Telegram not received or faulty [0x00b00000]

Response (Lenze setting printed in bold)	Setting: C14591/1 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: CAN-IN 1 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI2) RPDO2: Telegram not received or faulty [0x00b00001]

Response (Lenze setting printed in bold)	Setting: C14591/2 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: CAN-IN 2 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI2) RPDO3: Telegram not received or faulty [0x00b00002]

Response (Lenze setting printed in bold)	Setting: C14591/3 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: CAN-IN 3 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI2) RPDO4: Telegram not received or faulty [0x00b00003]

Response (Lenze setting printed in bold)	Setting: C14591/4 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: CAN-IN 4 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI2) RPDO5: Telegram not received or faulty [0x00b00004]

Response (Lenze setting printed in bold)	Setting: C14591/5 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: CAN-IN 5 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI2) RPDO6: Telegram not received or faulty [0x00b00005]

Response (Lenze setting printed in bold)	Setting: C14591/6 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: CAN-IN 6 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI2) RPDO7: Telegram not received or faulty [0x00b00006]

Response (Lenze setting printed in bold)	Setting: C14591/7 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: CAN-IN 7 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI2) RPDO8: Telegram not received or faulty [0x00b00007]

Response (Lenze setting printed in bold)	Setting: C14591/8 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
CANopen communication module in MXI2: CAN-IN 8 error • Incorrect PDO telegram length. • Transmission error. • Time monitoring of the PDOs has tripped.	• Set correct telegram length for CAN master (transmitter). • Eliminate trouble in the environment (e. g. EMC). • Select a different time monitoring or switch off time monitoring.

CAN module (MXI2) PDO manager: Faulty configuration [0x00b00008]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI2: CAN-PDO configuration error • Faulty project download. • Invalid CAN settings according to DS301V402 in the Engineer or PLC Designer. • Mapping variables have incorrect CANopen indices according to DS405.	• Repeat download. • Correct CAN settings in the project and regenerate project.

CAN module (MXI2) SDO server: Faulty configuration [0x00b10000]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI2: In the CAN SDO server a configuration error has occurred. • Faulty project download. • Invalid SDO server settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download. • Correct CAN settings in the project and regenerate project.

CAN module (MXI2) SDO client: Faulty configuration [0x00b20000]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
CANopen communication module in MXI2: In the CAN SDO client a configuration error has occurred. • Faulty project download. • Invalid SDO client settings according to DS301V402 in the Engineer or PLC Designer.	• Repeat download • Correct CAN settings in the project and regenerate project.

PLC configuration invalid [0x00b80000]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
An invalid control configuration has occurred.	Load a different application.

Positive limit switch has triggered [0x00b80001]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☒ **Quick stop by trouble** ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
The travel range limit switch in positive traversing direction has tripped.	Reset error message and retract limit switch.

Negative limit switch has triggered [0x00b80002]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☒ **Quick stop by trouble** ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
The travel range limit switch in negative traversing direction has tripped.	Reset error message and retract limit switch.

Motor brake: Angular drift with closed brake too high [0x00b80003]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☒ **Quick stop by trouble** ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
The stop position of the motor axis has changed by more than the permissible angle of rotation set in C02595, although the brake is engaged.	• Deactivate standstill monitoring (C02595 = 0). • Increase waiting time for status monitoring (C02591). The standstill monitoring only starts after this waiting time has elapsed. • Increase brake closing time (C02589) since during the brake closing time the standstill monitoring is not active. • Reduce threshold for brake activation (C02581).

Motor brake: Automatically activated after waiting time has elapsed [0x00b80004]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	

Cause	Remedy
This time monitoring will only be active if the speed setpoint has reached the threshold for brake activation (C02581). If the actual speed value does not reach/fall below the threshold set in C02581within the parameterised waiting time for brake activation (C02593), the brake will be closed due to time-out.	• Increase waiting time for brake activation (C02593). • Reduce threshold for brake activation (C02581).

Motor brake: Status monitoring error [0x00b80005]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☒ **Quick stop by trouble** ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Faulty external feedback of the brake status to the brake control.	• Check brake configuration with regard to the control selection in C02580. • Check setting for status input monitoring in C02583. When monitoring is active, the input *bBrakeApplied* must be triggered correctly (*bBrakeApplied* = *bBrakeReleased*). • Check voltage supply of the brake module.

Positive software limit switch overtravelled [0x00b80007]

Response (Lenze setting printed in bold)	**Setting:** C02716/2 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☒ **Quick stop by trouble** ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
The positive software limit position parameterised in C02702/2 has been overtravelled.	• Position within the software limit positions. • Increase permissible traversing range (change setting of the software limit positions). • Deactivate monitoring of the software limit positions by the basic function "Limiter".

Negative software limit switch overtravelled [0x00b80008]

Response (Lenze setting printed in bold)	Setting: C02716/2 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☒ **Quick stop by trouble** ☑ Warning locked ☑ Warning ☑ Information	

Cause	Remedy
The negative software limit position parameterised in C02702/1 has been overtravelled.	• Position within the software limit positions. • Increase permissible traversing range (change setting of the software limit positions). • Deactivate monitoring of the software limit positions by the basic function "Limiter".

Positive direction of rotation limited [0x00b80009]

Response (Lenze setting printed in bold)	Setting: C02716/1 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	

Cause	Remedy
Due to the setting of C02707 it was tried to traverse in the impermissible positive direction of rotation.	• Only traverse in permissible (negative) direction of rotation. • Change setting of the permissible direction of rotation (C02707).

Negative direction of rotation limited [0x00b8000a]

Response (Lenze setting printed in bold)	Setting: C02716/1 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	

Cause	Remedy
Due to the setting of C02707 it was tried to traverse in the impermissible negative direction of rotation.	• Only traverse in permissible (positive) direction of rotation. • Change setting of the permissible direction of rotation (C02707).

Speed has been limited [0x00b8000b]

Response (Lenze setting printed in bold)	Setting: C02716/3 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	

Cause	Remedy
• The requested profile speed is higher than the maximum speed set in C02703 and has been limited to this speed. • The required profile speed cannot be achieved with the motor reference speed set in C00011.	• Reduce speed of the traversing profile of the basic function (manual jog, homing, or positioning). • Increase maximum speed (C02703). • Deactivate monitoring of the limit values by the basic function "Limiter". • Set motor reference speed correctly (C00011).

Acceleration has been limited [0x00b8000c]

Response (Lenze setting printed in bold)	Setting: C02716/3 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	

Cause	Remedy
The requested profile acceleration is higher than the maximum acceleration set in C02705 and has been limited to this acceleration.	• Reduce acceleration of the traversing profile of the basic function (manual jog, homing, or positioning). • Increase maximum acceleration (C02705). • Deactivate monitoring of the limit values by the basic function "Limiter".

Deceleration has been limited [0x00b8000d]

Response (Lenze setting printed in bold)	**Setting:** C02716/3 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	
Cause	**Remedy**
The requested profile deceleration is higher than the maximum acceleration set in C02705 and has been limited to this acceleration.	• Reduce acceleration of the traversing profile of the basic function (manual jog, homing, or positioning). • Increase maximum acceleration (C02705). • Deactivate monitoring of the limit values by the basic function "Limiter".

Jerk has been limited [0x00b8000e]

Response (Lenze setting printed in bold)	**Setting:** C02716/3 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	
Cause	**Remedy**
The requested S-ramp time is lower than the minimum S-ramp time set in C02706 and has been limited to this S-ramp time.	• Increase S-ramp time of the traversing profile of the basic function (manual jog, homing, or positioning). • Reduce minimum S-ramp time (C02706). • Deactivate monitoring of the limit values by the basic function "Limiter".

Position target outside the software limit positions [0x00b8000f]

Response (Lenze setting printed in bold)	**Setting:** C02716/2 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☒ **Quick stop by trouble** ☑ Warning locked ☑ Warning ☑ Information	
Cause	**Remedy**
It was tried to position to a target outside the software limit positions.	• Select position target inside the software limit positions. • Increase permissible traversing range (change setting of the software limit positions). • Deactivate monitoring of the software limit positions by the basic function "Limiter".

Maximum speed exceeded [0x00b80010]

Response (Lenze setting printed in bold)	**Setting:** C02716/3 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	
Cause	**Remedy**
The max. speed parameterised in C02703 has been exceeded.	• Reduce speed. • Increase maximum speed (C02703). • Deactivate monitoring of the limit values by the basic function "Limiter".

Maximum acceleration exceeded [0x00b80011]

Response (Lenze setting printed in bold)	**Setting:** C02716/3 (☑ Adjustable response)
☑ None ☐ System fault ☑ Fault ☑ Trouble ☑ Quick stop by trouble ☑ Warning locked ☑ Warning ☒ **Information**	
Cause	**Remedy**
The max. acceleration parameterised in C02705 has been exceeded.	• Reduce acceleration. • Increase maximum acceleration (C02705). • Deactivate monitoring of the limit values by the basic function "Limiter".

Time-out torque feedforward control - brake [0x00b80012]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☒ **Quick stop by trouble** ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
After 1 second the actual torque has not yet reached 90 % of the precontrolled setpoint torque for releasing the brake.	Check settings for torque feedforward control. • Possibly the maximum current (C00022) is not sufficient for the required torque feedforward control.

Cam data: Serial number MM does not match [0x00b80013]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
The cam data are provided with an access protection of level 3 (linkage to the memory module), and the serial number contained in the cam data does not correspond to the serial number of the memory module.	Check and, if required, correct the serial number entered in the »Cam Designer«. Afterwards redownload the cam data.

Cam data are corrupted [0x00b80014]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
Checksum error during reading the file, or the password was manipulated.	Redownload cam data.

Cam data restored [0x00b80015]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
The last download of the cam data was faulty or has not been completed successfully. The previous cam data – if available – have been downloaded from the backup file.	• Redownload cam data. • Save cam data within the controller in the memory module via device command C00002 = "502: Save cam data".

Cam data locked due to incorrect password [0x00b80016]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	Remedy
The cam data were locked since the user password was entered incorrectly for three times. The cam data were not loaded.	Reset parameters to the Lenze setting via device command C00002 = "0: Load Lenze setting". Then download cam data again.

Cam data locked due to incorrect safety key [0x00b80017]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information

Cause	**Remedy**
The cam data were not loaded because the safety key is damaged. The password in the cam data or the serial number of the memory module has been manipulated.	Reset parameters to the Lenze setting via device command C00002 = "0: Load Lenze setting". Then download cam data again.

Homing mode not allowed [0x00b80019]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
From software version V3.0 The homing mode selected in C02640 is not supported in the motor control type selected in C00006.	Select another homing mode in C02640.

Int. overflow C02620 (manual jog: Speed 1) [0x00b8001a]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02620.

Int. overflow C02621 (manual jog: Speed 2) [0x00b8001b]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	**Remedy**
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02621.

Int. overflow C02622 (manual jog: Acceleration) [0x00b8001c]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02622.

Int. overflow C02623 (manual jog: Deceleration) [0x00b8001d]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02623.

Int. overflow C02701/1 (positive software limit position) [0x00b80020]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02701/1.

Int. overflow C02701/2 (negative software limit position) [0x00b80021]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02701/2.

Int. overflow C02703 (maximum speed) [0x00b80022]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02703.

Int. overflow C02705 (maximum acceleration) [0x00b80023]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02705.

Int. overflow C02708/1 (limited speed 1) [0x00b80024]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02708/1.

Int. overflow C02708/2 (limited speed 2) [0x00b80025]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	**Remedy**
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02708/2.

Int. overflow C02708/3 (limited speed 3) [0x00b80026]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02708/3.

Int. overflow C02708/4 (limited speed 4) [0x00b80027]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02708/4.

Int. overflow C02708/1 (decel. limited speed 1) [0x00b80028]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02710/1.

Int. overflow C02708/2 (decel. limited speed 2) [0x00b80029]

Response (Lenze setting printed in bold)	
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02710/2.

Int. overflow C02708/3 (decel.limited speed 3) [0x00b8002a]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02710/3.

Int. overflow C02708/4 (decel. limited speed 4) [0x00b8002b]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02710/4.

Int. overflow C02713 (max. dist. manual control) [0x00b8002c]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02713.

Int. overflow C02642 (home position) [0x00b8002d]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02642.

Int. overflow C02643 (homing: Target position) [0x00b8002e]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02643.

Int. overflow C02644 (homing: Speed 1) [0x00b8002f]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02644.

Int. overflow C02645 (homing: Acceleration 1) [0x00b80030]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02645.

Int. overflow C02646 (homing: Speed 2) [0x00b80031]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02646.

Int. overflow C02647 (homing: Acceleration 2) [0x00b80032]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02647.

Int. overflow C02670 (positioner: Tol. for target position) [0x00b80033]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information

Cause	Remedy
From software version V3.0 Due to a subsequent change of the machine parameters for gearbox ratio, feed constant, or resolution of an encoder revolution, the value set in the parameter cannot be displayed in the internal unit. See chapter "Max. position, speed, and acceleration that can be displayed internally".	• Check machine parameters and adapt them if required. • Change setting in C02670.

Cam data: Invalidated due to change of mechanical data [0x00b80034]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ **Warning** ☐ Information

Cause	Remedy
From software version V4.0 One or several machine parameters have been changed that have an influence on the internal scaling of the Cam data. The Cam data has to be recalculated. See chapter "Invalid Cam data due to changed machine parameters".	Execute device command C00002 = "503: Calculate Cam Data". This automatically resets the warning.

Cam data: invalid product number [0x00b80035]

Response (Lenze setting printed in bold)
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**

Cause	Remedy
From software version V5.0 A product number has been created in the system block LS_CamInterface which is not in the range of product number of the downloaded cam data.	Check number of products. • The product number must be higher than 0 and lower than the value displayed in C02908. • C02908 displays the highest product number +1 of the cam data currently being processed.

14 Parameter reference

All parameters for controller parameter setting or monitoring are stored under "codes".

- The codes are numbered and designated by a "C" in front of the code, e.g. "C00002".
- 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".

Parameters available in the controller only from a certain software version are marked accordingly ("From software version Vx.x").

The parameter descriptions are based on the software version V07.00.12

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.

For general information on how to read and change parameters please see the online documentation for the »Engineer«.

14.1 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 notes:

- Parameter number (Cxxxxx)
- Parameter name (display text in the »Engineer» and keypad)
- Data type
- Decimal and hexadecimal parameter index for access via bus systems

Table contents

The table contains further general explanations & notes on the parameter and the possible settings the representation of which depends on the parameter type:

- Parameters with read-only access
- Parameters with write access

Table footer

The table footer contains the Parameter attributes.

14.1.1 Data type

The following data types are available for parameters:

Data type	Meaning
INTEGER_8	8-bit value with sign
INTEGER_16	16-bit value with sign
INTEGER_32	32-bit value with sign
INTEGER_64	64-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
UNSIGNED_64	64-bit value without sign
FLOATING_POINT	32-bit floating point number
VISIBLE_STRING	String of characters of printable characters
OCTET_STRING	String of characters of any characters
BITFIELD_8	8-bit value bit-coded
BITFIELD_16	16-bit value bit-coded
BITFIELD_32	32-bit value bit-coded

14.1.2 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			

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	Wert	Einheit
	61	0	Kühlkörpertemperatur	30	°C

14.1.3 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 are either selected by means of a selection list or through direct value entry.
- Values outside the valid setting range are represented in red in the »Engineer«.

14.1.3.1 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				

Parameter setting in the »Engineer«

In the »Engineer«, parameters are set by entering the desired value into the input field:

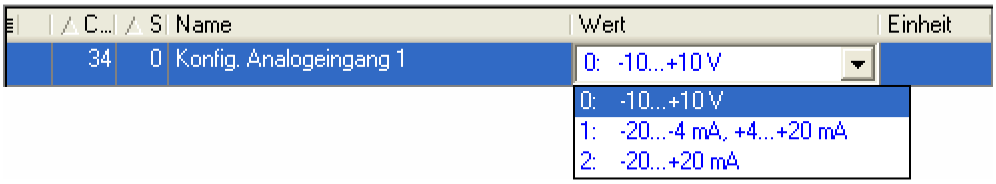
14.1.3.2 Parameters with selection list

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		

Parameter setting in the »Engineer«

In the »Engineer«, a list field is used for parameter setting:



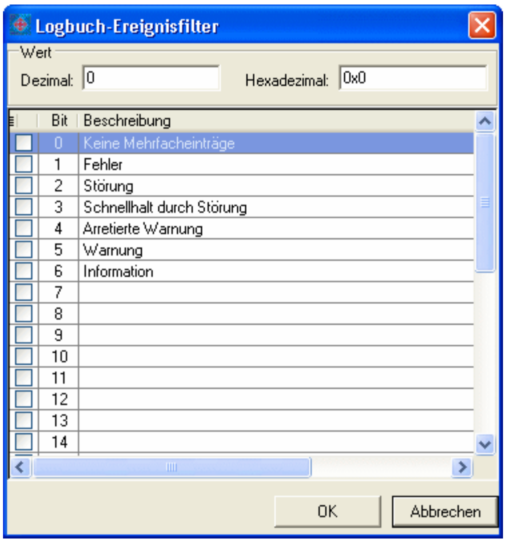
14.1.3.3 Parameters with a bit coded setting

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		

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:



14.1.3.4 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			

Parameter setting in the »Engineer«

The »Engineer« parameter list itemises all subcodes separately. The parameters are set as described in the previous chapters.

	C...	S	Name	Wert	Einheit
	114	1	Digitaleing. 1: Klemmenpol.	0	
	114	2	Digitaleing. 2: Klemmenpol.	0	
	114	3	Digitaleing. 3: Klemmenpol.	0	
	114	4	Digitaleing. 4: Klemmenpol.	0	

14.1.4 Parameter attributes

Description structure

The table footer contains the parameter attributes:

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Meaning of the attributes

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	The parameter is **not** transferred to the regenerative power supply module when the command Download parameter set is executed.	

Scaling factor

The "scaling factor" is important for the parameter access via a bus system:

$$\text{Read value (via bus system)} = \text{Scaling factor} \cdot \text{Indicated value (Engineer)}$$

14.1.5 Abbreviations used in parameter & selection texts

Since the character length of the parameter and selection texts is limited, the following abbreviations are used:

Abbreviation	Meaning
CAN module	CANopen communication module (type E94AYCCA)
ENP	Electronic nameplate
Ethernet module	Ethernet communication module (type E94AYCEN)
MXI1	*Module eXtension Interface* 1 - module slot for extension 1
MXI2	*Module eXtension Interface* 2 - module slot for extension 2

14.2 Parameter list

This chapter describes all parameters of the operating system in numerically ascending order.

Note!

The parameter descriptions are based on the software version V05.

C00002

Parameter | Name: **C00002 | Device commands**	Data type: UNSIGNED_32 Index: 24573_d = $5FFD_h$

Device commands
- C00003 shows the status of the last executed device command.
- Under C00150 you can query the current status of the device control.

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!
The meaning of the status display in C00003 can be obtained from the subchapter for the corresponding device command in chapter "Device commands".

Selection list (Lenze setting printed in bold)		Information
0	**Load Lenze setting**	Resets parameters to Lenze setting. • Only possible when the application has stopped and the controller is inhibited.
1	Load start parameters	Loads parameters from the memory module. • Only possible when the application has stopped and the controller is inhibited.
2	ENP: Load plant data	Reads plant data from the electronic motor nameplate. • Only possible when the application has stopped and the controller is inhibited.
5	Activate application	Activates the application selected under C00005. • Whether the application is also started, depends on the auto-start setting selected. • Only possible when the application has stopped.
7	Save selected application	Selects the active application as start application.
11	Save start parameters	Saves parameters fail-safe in the memory module.
20	Delete logbook	Deletes all logbook entries.
21	Archive log file	Exports logbook entries to file.
31	Start application	
32	Stop application	
33	Reset program	Carries out a reset. • All variables are reset to their initialisation value. • The situation corresponds to the start of a new program loaded into the control (cold start).
34	Delete program	Carries out a reset (source). • All variables are reset to their initialisation value. • The application program is deleted and the controller is reset to its original state.

Parameter | Name: **C00002 | Device commands**		Data type: UNSIGNED_32 Index: 24573_d = $5FFD_h$
35	Restart program	Carries out a reset (warm start). • All variables except the RETAIN variables are reset to their initialisation value. • The situation corresponds to a power failure or switching the controller off/on (warm start) while the program is running.
36	Reset runtime measurement	▶ Runtime measurement
41	Inhibit controller	
42	Enable controller	
43	Reset error	
45	Activate quick stop	▶ Basic function "Quick stop"
46	Reset quick stop	▶ Basic function "Quick stop"
47	Internal command 47	For Lenze service only
48	Internal command 48	For Lenze service only
51	Identify pole position (360°)	Executes identification of pole position. • The function can only be activated when the controller is inhibited. • After this, the execution of the command starts automatically when the controller is enabled. • During the pole position identification, the motor makes one electrical revolution. This leads to a mechanical rotation of the motor shaft. • The determined pole position is indicated under code C00058. ▶ Pole position identification
52	Identify pole position (min. motion)	Executes identification of pole position. • The function can only be activated when the controller is inhibited. • After this, the execution of the command starts automatically when the controller is enabled. • During the pole position identification, the rotor aligns itself. This is compensated by a position control. • The determined pole position is indicated under code C00058. ▶ Pole position identification
59	Resolver error identification	From software version V7.0 Execute resolver error identification. ▶ Resolver error compensation
70	Load Lenze inverter charac.	From software version V4.0 Load type-dependent inverter error characteristic. • For the case that the determination of the inverter error characteristic with the device command "71: Determine inverter characteristic" was not possible or has supplied incorrect results. • The function can only be activated when the controller is inhibited.
71	Calculate inv. characteristic	Determines inverter error characteristic. • The function can only be activated when the controller is inhibited. • After this, the execution of the command starts automatically when the controller is enabled. ▶ Optimise the switching performance of the inverter

Parameter | Name: **C00002 | Device commands**		Data type: UNSIGNED_32 Index: 24573_d = $5FFD_h$
72	Set motor parameters	Determines motor parameters automatically. • The function can only be activated when the controller is inhibited. • After this, the execution of the command starts automatically when the controller is enabled. ▶ Determine the motor parameters
77	Calculate current controller parameters	From software version V5.0 Calculates the gain and reset time of the current controller. • Usually not required for a Lenze motor. • The device command is no identification procedure for determining the current controller parameters! ▶ Calculate current controller parameters
78	Calculate speed controller parameters	From software version V5.0 Calculates the gain, reset, and rate time of the speed controller. • The device command is no identification procedure for determining the speed controller parameters! ▶ Calculate speed controller parameters
91	CAN on-board: Reset node	Reinitialises the "CAN on-board" interface. • Required when the baud rate, node address or identifiers are changed. ▶ "CAN on board" system bus
92	CAN module: Reset node	Reinitialises CANopen interface of the CANopen communication module. • Required when the baud rate, node address or identifiers are changed.
93	CAN on-board: Pred.Connect.Set	Sets basic identifier for the "CAN on board" interface according to the "Predefined Connection Set" (DS301 V4.02). ▶ "CAN on board" system bus
94	CAN module: Pred.Connect.Set	Sets basic identifier for the CANopen interface of the CANopen communication module according to the Predefined Connection Set" (DS301 V4.02).
95	CAN on-board: Identify node	Detects nodes connected to the "CAN on board" interface. • The result of the CAN bus scan is displayed in C00393. ▶ "CAN on board" system bus
96	CAN module: Identify node	Detects the nodes connected to the CANopen interface of the CANopen communication module. • The result of the CAN bus scan is displayed in C13393 (for MXI1) or C14393 (for MXI2).
101	Unbind/bind Ethernet module MXI1	Reinitialises the Ethernet interface of the Ethernet communication module in module slot MXI1. • Required when a new setting for an IP or gateway address is to be accepted without mains switching.
102	Unbind/bind Ethernet module MXI2	Reinitialises the Ethernet interface of the Ethernet communication module in module slot MXI2 • Required when a new setting for an IP or gateway address is to be accepted without mains switching.
201	Activate parameter set 1	Loads parameter set 1 from the memory module. • Only possible when the application has stopped and the controller is inhibited.
202	Activate parameter set 2	Loads parameter set 2 from the memory module. • Only possible when the application has stopped and the controller is inhibited.

Parameter | Name: **C00002 | Device commands**		Data type: UNSIGNED_32 Index: 24573_d = $5FFD_h$
203	Activate parameter set 3	Loads parameter set 3 from the memory module. • Only possible when the application has stopped and the controller is inhibited.
204	Activate parameter set 4	Loads parameter set 4 from the memory module. • Only possible when the application has stopped and the controller is inhibited.
301	Archive parameter set 1	Saves the current parameter set as parameter set 1 in the memory module.
302	Archive parameter set 2	Saves the current parameter set as parameter set 2 in the memory module.
303	Archive parameter set 3	Saves the current parameter set as parameter set 3 in the memory module.
304	Archive parameter set 4	Saves the current parameter set as parameter set 4 in the memory module.
401	Internal command 401	For Lenze service only
501	Load Cam Data	From software version V3.0 Reloads cam data from the memory module into the controller. • Only possible when the application has stopped and the controller is inhibited. • If the cam data are provided with an access protection, the user password has to be entered in C02900 first. ▸ Basic function "Cam data management"
502	Save Cam Data	From software version V3.0 Saves the cam data available in the main memory of the controller in the memory module with mains failure protection. • If the cam data are provided with an access protection, the user password has to be entered in C02900 first. ▸ Basic function "Cam data management"
503	Calculate Cam Data	From software version V3.0 Converts the cam data available in the main memory of the controller into the internal format and provides them to the application. ▸ Basic function "Cam data management"
504	Calculate Cam Checksum	From software version V3.0 Recalculates the checksum of the cam data available in the main memory of the controller. • Required if the cam data in the main memory of the controller have been changed via parameters. ▸ Basic function "Cam data management"
730	Internal command 730	For Lenze service only
731	Internal command 731	For Lenze service only
732	Internal command 732	For Lenze service only
733	Internal command 733	For Lenze service only
800	Internal command 800	For Lenze service only
810	Internal command 810	
811	Internal command 811	
812	Internal command 812	
1001	Internal command 1001	For Lenze service only
1020	Internal command 1020	For Lenze service only
1021	Export parameters to file	For Lenze service only

Parameter | Name: **C00002 | Device commands**		Data type: UNSIGNED_32 Index: 24573$_d$ = 5FFD$_h$
1030	Format file system	Formats file system of the memory module.
1040	Restore file system	Restores file system of the memory module (low level formatting). • The low level formatting of the file system by the user is only intended for the exceptional case when the standard formatting is not possible anymore, e.g. due to damaged internal management information.
10000	Prepare firmware update	Sets the controller to the firmware update mode.
11000	Restart controller	Restarts controller via parameter setting.
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer		Scaling factor: 1

C00003

Parameter | Name: **C00003 | Device command status**			Data type: UNSIGNED_32 Index: 24572$_d$ = 5FFC$_h$

Display of the number/status of the device command last executed (C00002).
- The number of the command is situated in the upper 16 bits (for the meaning of the number see code C00002).
- The result of the command stands in the lower 16 bits.

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!
The meaning of the status display in C00003 can be obtained from the subchapter for the corresponding device command in chapter "Device commands".

Display range (min. value | unit | max. value)		
0		4294967295

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer   Scaling factor: 1

Status	Meaning	Controller command
0	Device command executed successfully	0: Load Lenze setting
1	General fault	
34050	Device command in process	
39424	CAN fault	
...	...	
39679	CAN fault	
65536	Device command executed successfully	1: Load start parameters
65537	General fault	
99371	Fault while reading the parameter set partition	
99374	No memory module available	
99586	Device command in process	
104960	CAN fault	
...	...	
105215	CAN fault	
131072	Device command executed successfully	2: ENP: Load plant data
131073	General fault	
165122	Device command in process	
327680	Device command executed successfully	5: Activate application
327681	General fault	
361730	Device command in process	

Status	Meaning	Controller command
458752	Device command executed successfully	7: Save selected application
458753	General fault	
492802	Device command in process	
720896	Device command executed successfully	11: Save start parameters
720897	General fault	
754718	Fault while writing into a file	
754734	No memory module available	
754946	Device command in process	
761857	Access to file has been denied since the file is already accessed from another position	
761861	I/O fault when accessing the file system	
761868	RAM is full	
761869	Access authorisation denied	
761884	No free memory on the memory module	
1310720	Device command executed successfully	20: Delete logbook
1310721	General fault	
1344770	Device command in process	
1376256	Device command executed successfully	21: Archive logbook
1376257	General fault	
1410306	Device command in process	
2031616	Device command executed successfully	31: Start application
2031617	General fault	
2065666	Device command in process	
2097152	Device command executed successfully	32: Stop application
2097153	General fault	
2131202	Device command in process	
2162688	Device command executed successfully	33: Reset program
2162689	General fault	
2196738	Device command in process	
2228224	Device command executed successfully	34: Delete program
2228225	General fault	
2262274	Device command in process	
2293760	Device command executed successfully	35: Restart program
2293761	General fault	
2327810	Device command in process	
2359296	Device command executed successfully	36: Reset runtime measurement
2359297	General fault	
2393346	Device command in process	

Status	Meaning	Controller command
3342336	Device command executed successfully	51: Identify pole position (360°)
3342337	General fault	
3376386	Device command in process	
3382023	Pole position identification cannot be executed because of wrong motor type (asynchronous motor).	
3382024	Pole position identification has been aborted	
3382025	Pole position identification cannot be executed because another identification is already active.	
3382026	Identification of pole position cannot be executed because U-rotation or I-rotation test mode is active.	
3382027	Identification of pole position cannot be executed because current controller optimisation mode is active.	
3382033	Pole position identification cannot be executed because the motor is blocked (e.g. by a mechanical brake), a motor phase is not connected, or a phase shifter is in the motor cable.	
3382047	Pole position identification cannot be executed because an error or trouble is active.	
3382065 From software version V3.0	Pole position identification cannot be executed because either the entire motor or a motor phase is not connected.	
3407872	Device command executed successfully	52: Identify pole position (min. motion)
3407873	General fault	
3441922	Device command in process	
3447559	Pole position identification cannot be executed because of wrong motor type (asynchronous motor).	
3447560	Pole position identification has been aborted	
3447561	Pole position identification cannot be executed because another identification is already active.	
3447562	Identification of pole position cannot be executed because U-rotation or I-rotation test mode is active.	
3447563	Identification of pole position cannot be executed because current controller optimisation mode is active.	
3447569 From software version V4.0	Pole position identification cannot be executed because the motor is blocked (e.g. by a mechanical brake), a motor phase is not connected, or a phase shifter is in the motor cable.	
3447583	Pole position identification cannot be executed because an error or trouble is active.	
3447597	Identification of pole position cannot be executed because the rotor has moved too strongly.	
3447601 From software version V3.0	Pole position identification cannot be executed because either the entire motor or a motor phase is not connected.	

Status	Meaning	Controller command
3900674	Device command in process	59: Resolver error identification
3866624	Device command executed successfully	
3866625	General fault	
3906358	Resolver error identification cannot be executed since the wrong control type is active (no servo control).	
3906359	Resolver error identification cannot be executed since an error or trouble is active.	
3906360	Resolver error identification cannot be executed because another identification is already active.	
3906361	Resolver error identification cannot be executed because of too small speed (< 500 rpm).	
4587520	Device command executed successfully	70: Load Lenze inverter characteristic
4587521	General fault	
4621570	Device command in process	
4653056	Device command executed successfully	71: Calculate inv. characteristic
4653057	General fault	
4687106	Device command in process	
4692754	The calculation of the inverter characteristic cannot be started since the current controller test mode is active.	
4692755	The calculation of the inverter characteristic cannot be started since the V/f test mode is active.	
4692756	The calculation of the inverter characteristic cannot be started since the pole position identification is active.	
4692757	Calculation of the inverter characteristic has been aborted.	
4692758	Calculation of the inverter characteristic has been interrupted by error.	
4692789 From software version V5.0	The detected inverter error characteristic exceeds internal limits. This situation can for instance occur if the motor power is very much lower than the device power.	
4718592	Device command executed successfully	72: Determine motor parameters
4718593	General fault	
4752642	Device command in process	
4758290	Motor identification cannot be started since the current controller test mode is active.	
4758291	Motor identification cannot be started since the V/f test mode is active.	
4758292	Motor identification cannot be started because pole position identification is active.	
4758293	Motor identification has been aborted.	
4758294	Motor identification has been aborted by fault.	
4758332 From software version V7.0	Motor identification aborted due to inconsistent motor parameters.	
5046272	Device command executed successfully	77: Calculate current controller parameters
5046273	General fault	
5080322	Device command in process	
5086002	At least one calculated value is beyond the valid setting range.	
5086003	Stator resistance (C00084) too small (zero).	

Status	Meaning	Controller command
5111808	Device command executed successfully	78: Calculate speed controller parameters
5111809	General fault	
5145858	Device command in process	
5151540	At least one calculated value is beyond the valid setting range.	
5963776	Device command executed successfully	91: CAN on board: Reset node
5963777	General fault	
5997826	Device command in process	
6003200	CAN fault	
...	...	
6003455	CAN fault	
6029312	Device command executed successfully	92: CAN module: Reset node
6029313	General fault	
6063362	Device command in process	
6068736	CAN fault	
...	...	
6068991	CAN fault	
6094848	Device command executed successfully	93: CAN on board: Pred.Connect.Set
6094849	General fault	
6128898	Device command in process	
6160384	Device command executed successfully	94: CAN module: Pred.Connect.Set
6160385	General fault	
6194434	Device command in process	
6225920	Device command executed successfully	95: CAN on board: Identify node
6225921	General fault	
6259970	Device command in process	
6291456	Device command executed successfully	96: CAN module: Identify node
6291457	General fault	
6325506	Device command in process	
6619136	Device command executed successfully	101: Ethernet module MXI2 unbind/bind
6619137	General fault	
6653186	Device command in process	
6684672	Device command executed successfully	102: Ethernet module MXI2 unbind/bind
6684673	General fault	
6718722	Device command in process	

Status	Meaning	Controller command
13172731	General fault	201: Activate parameter set 1
13172736	Device command executed successfully	
13206532	File could not be opened.	
13206557	Fault while reading out of a file.	
13206558	Fault while writing into a file.	
13206559	Invalid file type.	
13206560	Unexpected file end.	
13206562	Checksum error	
13206786	Device command in process	
13212160	CAN fault	
...	...	
13212415	CAN fault	
13213697	Access to file has been denied since the file is already accessed from another position	
13213701	I/O fault when accessing the file system	
13213708	RAM is full	
13213709	Access authorisation denied	
13213724	No free memory on the memory module	
13238272	Device command executed successfully	202: Activate parameter set 2
13238273	General fault	
13272068	File could not be opened.	
13272093	Fault while reading out of a file.	
13272094	Fault while writing into a file.	
13272095	Invalid file type.	
13272096	Unexpected file end.	
13272098	Checksum error	
13272322	Device command in process	
13277696	CAN fault	
...	...	
13277951	CAN fault	
13279233	Access to file has been denied since the file is already accessed from another position	
13279237	I/O fault when accessing the file system	
13279244	RAM is full	
13279245	Access authorisation denied	
13279260	No free memory on the memory module	

Status	Meaning	Controller command
13303808	Device command executed successfully	203: Activate parameter set 3
13303809	General fault	
13337604	File could not be opened.	
13337629	Fault while reading out of a file.	
13337630	Fault while writing into a file.	
13337631	Invalid file type.	
13337632	Unexpected file end.	
13337634	Checksum error	
13337858	Device command in process	
13343232	CAN fault	
...	...	
13343487	CAN fault	
13344769	Access to file has been denied since the file is already accessed from another position	
13344773	I/O fault when accessing the file system	
13344780	RAM is full	
13344781	Access authorisation denied	
13344796	No free memory on the memory module	
13369344	Device command executed successfully	204: Activate parameter set 4
13369345	General fault	
13403140	File could not be opened.	
13403165	Fault while reading out of a file.	
13403166	Fault while writing into a file.	
13403167	Invalid file type.	
13403168	Unexpected file end.	
13403170	Checksum error	
13403394	Device command in process	
13408768	CAN fault	
...	...	
13409023	CAN fault	
13410305	Access to file has been denied since the file is already accessed from another position	
13410309	I/O fault when accessing the file system	
13410316	RAM is full	
13410317	Access authorisation denied	
13410332	No free memory on the memory module	

Status	Meaning	Controller command
19726336	Device command executed successfully	301: Archive parameter set 1
19726337	General fault	
19760132	File could not be opened.	
19760157	Fault while reading out of a file.	
19760158	Fault while writing into a file.	
19760160	Unexpected file end.	
19760386	Device command in process	
19767297	Access to file has been denied since the file is already accessed from another position	
19767301	I/O fault when accessing the file system	
19767308	RAM is full	
19767309	Access authorisation denied	
19767324	No free memory on the memory module	
19791872	Device command executed successfully	302: Archive parameter set 2
19791873	General fault	
19825668	File could not be opened.	
19825693	Fault while reading out of a file.	
19825694	Fault while writing into a file.	
19825696	Unexpected file end.	
19825922	Device command in process	
19832833	Access to file has been denied since the file is already accessed from another position	
19832837	I/O fault when accessing the file system	
19832844	RAM is full	
19832845	Access authorisation denied	
19832860	No free memory on the memory module	
19857408	Device command executed successfully	303: Archive parameter set 3
19857409	General fault	
19891204	File could not be opened.	
19891229	Fault while reading out of a file.	
19891230	Fault while writing into a file.	
19891232	Unexpected file end.	
19891458	Device command in process	
19898369	Access to file has been denied since the file is already accessed from another position	
19898373	I/O fault when accessing the file system	
19898380	RAM is full	
19898381	Access authorisation denied	
19898396	No free memory on the memory module	

Status	Meaning	Controller command
19922944	Device command executed successfully	304: Archive parameter set 4
19922945	General fault	
19956740	File could not be opened.	
19956765	Fault while reading out of a file.	
19956766	Fault while writing into a file.	
19956768	Unexpected file end.	
19956994	Device command in process	
19963905	Access to file has been denied since the file is already accessed from another position	
19963909	I/O fault when accessing the file system	
19963916	RAM is full	
19963917	Access authorisation denied	
19963932	No free memory on the memory module	
32833536	Device command executed successfully	501: Load cam data
32833537	General fault	
32867586	Device command in process	
32875521	No cam data available on the memory module	
32875523	Loading of the cam data failed	
32875525	Checksum error	
32875542	Wrong password entered	
32875545	The cam functionality is deactivated	
32899072	Device command executed successfully	502: Save cam data
32899073	General fault	
32933122	Device command in process	
32941057	No cam data to be saved are available in the RAM of the controller	
32941060	Saving of the cam data failed	
32941078	Wrong password entered	
32941081	The cam functionality is deactivated	
32964608	Device command executed successfully	503: Calculate cam data
32964609	General fault	
32998658	Device command in process	
33006617	The cam functionality is deactivated	
33030144	Device command executed successfully	504: Calculate cam data checksum
33030145	General fault	
33064194	Device command in process	
33072153	The cam functionality is deactivated	
67502080	Device command executed successfully	1030: Format file system
67502081	General fault	
67536130	Device command in process	
655360000	Device command executed successfully	10000: Prepare firmware update
655360001	General fault	
655394050	Device command in process	
720896001	General fault	11000: Restart controller
720930050	Device command in process	

C00004

Parameter | Name: **C00004 | Service password**			Data type: UNSIGNED_32 Index: 24571_d = $5FFB_h$
Service code to unlock protected device commands (C00002).			
Setting range (min. value | unit | max. value)			**Lenze setting**
0		4294967295	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 1

C00005

Parameter | Name: **C00005 | Application selection**			Data type: INTEGER_32 Index: 24570_d = $5FFA_h$
Application selection • Use the device command C00002="5" to activate the selected application.			
Setting range (min. value | unit | max. value)			**Lenze setting**
-1		16	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 1

C00006

Parameter | Name: **C00006 | Select motor control**		Data type: UNSIGNED_32 Index: 24569_d = $5FF9_h$
		▶ Motor interface
Selection list (Lenze setting printed in bold)		**Information**
1	**SC: Servo control sync. motor**	For synchronous motors with speed sensor ▶ Servo control
2	SC: Servo control async. motor	For asynchronous motors with speed sensor ▶ Servo control
4	SLVC: Sensorless vector control	From software version V3.0 ▶ Sensorless vector control
6	VFCplus: V/f control open loop	From software version V3.0 ▶ V/f control
7	VFCplus: V/f control closed loop	From software version V3.0 ▶ V/f control closed loop
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00007

Parameter | Name: **C00007 | Active application**			Data type: INTEGER_32 Index: 24568_d = $5FF8_h$
Display range (min. value | unit | max. value)			
-2147483648		2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00008

Parameter | Name: **C00008 | Progress of device command**			Data type: UNSIGNED_32 Index: 24567_d = $5FF7_h$
From software version V7.0			
Display range (min. value | unit | max. value)			
0		4294967295	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00011

Parameter | Name: **C00011 | Motor reference speed**			Data type: UNSIGNED_32 Index: 24564_d = $5FF4_h$
For parameter setting via interface: In case of bigger changes, only change the setting in one step when the controller is inhibited!			
Setting range (min. value | unit | max. value)			**Lenze setting**
50	rpm	50000	**3000 rpm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00018

Parameter | Name: **C00018 | Chopper frequency**		Data type: UNSIGNED_32 Index: 24557_d = $5FED_h$
Selection list (Lenze setting printed in bold)		**Information**
2	1 kHz fixed/drive-optimised	**Note:** • The maximum output frequency of the controller is limited to 1/8 of the switching frequency selected here! • The switching frequencies that can be selected depend on the device type (see Hardware Manual, chapter "Rated data"). • In the case of an offline parameter setting or when exchanging the memory module between different Servo Drives 9400 HighLine device types, always check the setting of this parameter and adapt it, if required, to prevent a parameter error after the parameter set download or module change!
3	2 kHz fixed/drive-optimised	
4	4 kHz fixed/drive-optimised	
5	8 kHz fixed/drive-optimised	
8	2 kHz var./drive-optimised	
9	4 kHz var./drive-optimised	
10	**8 kHz var./drive-optimised**	
11	16 kHz var./drive-optimised	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00019

Parameter | Name: **C00019 | Threshold - standstill recognition**			Data type: UNSIGNED_32 Index: 24556_d = $5FEC_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0	rpm	450	**5 rpm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00022

Parameter | Name: **C00022 | Maximum current**			Data type: UNSIGNED_32 Index: 24553_d = $5FE9_h$
Note: • To avoid that the motor starts unintentionally without adjusting the plant data, the maximum current in the Lenze setting is set to "0 A"! • The upper limit value is the maximum device current (see display in C00789). • In the case of an offline parameter setting or when exchanging the memory module between different 9400 HighLine device types, always check the setting of this parameter and adapt it, if required, to prevent a parameter error after the parameter set download or module change! • Also check the threshold set for the maximum motor current monitoring (C00620).			
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	A	21474836.47	**0.00 A**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00034

Parameter | Name: **C00034 | Config. analog input 1**		Data type: UNSIGNED_32 Index: 24541_d = $5FDD_h$
Selection list (Lenze setting printed in bold)		
0	**-10 ... +10 V**	
1	-20...-4 mA, +4...+20 mA	
2	-20 ... +20 mA	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00050

Parameter | Name: **C00050 | Speed setpoint [rpm]**			Data type: INTEGER_32 Index: 24525_d = $5FCD_h$
Display range (min. value | unit | max. value)			
-480000	rpm	480000	
Subcodes			**Information**
C00050/1			Speed setpoint 1 [rpm]
C00050/2			Speed setpoint 2 [rpm]
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00051

Parameter | Name: **C00051 | Actual speed [rpm]**			Data type: INTEGER_32 Index: 24524_d = $5FCC_h$
Display range (min. value | unit | max. value)			
-480000	rpm	480000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00052

Parameter | Name: **C00052 | Motor voltage**			Data type: UNSIGNED_32 Index: 24523_d = $5FCB_h$
Display range (min. value | unit | max. value)			
0	V	2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00053

Parameter | Name: **C00053 | DC-bus voltage**			Data type: UNSIGNED_32 Index: 24522_d = $5FCA_h$
Display range (min. value | unit | max. value)			
0	V	2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00054

Parameter | Name: **C00054 | Motor current**			Data type: UNSIGNED_32 Index: 24521_d = $5FC9_h$
Display range (min. value | unit | max. value)			
0.00	A	500.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00055

Parameter | Name: **C00055 | Phase currents**			Data type: INTEGER_32 Index: 24520_d = $5FC8_h$
Display range (min. value | unit | max. value)			
-500.00	A	500.00	
Subcodes			**Information**
C00055/1			Phase zero system
C00055/2			Phase U
C00055/3			Phase V
C00055/4			Phase W
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00056

Parameter | Name: **C00056 | Torque setpoint**			Data type: INTEGER_32 Index: 24519_d = $5FC7_h$
Display range (min. value | unit | max. value)			
-21474836.47	Nm	21474836.47	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00057

Parameter | Name: **C00057 | Torque**			Data type: UNSIGNED_32 Index: 24518_d = $5FC6_h$
Display range (min. value | unit | max. value)			
0.000	Nm	2147483.647	
Subcodes			**Information**
C00057/1			Maximum torque • With regard to the selected motor and the max. short-time output current of the device.
C00057/2			Motor reference torque • Torque at maximum current (C00022).
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C00058

Parameter | Name: **C00058 | Pole position**			Data type: INTEGER_32 Index: 24517_d = $5FC5_h$
Setting range (min. value | unit | max. value)			
-179.9	°	179.9	
Subcodes	**Lenze setting**		**Information**
C00058/1	-90.0 °		Pole position resolver
C00058/2	0.0 °		Pole position encoder
C00058/3	0.0 °		Pole position module
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00059

Parameter | Name: **C00059 | Motor - number of pole pairs**			Data type: UNSIGNED_32 Index: 24516_d = $5FC4_h$
Display range (min. value | unit | max. value)			
0		200	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00060

Parameter | Name: **C00060 | Rotor position**			Data type: INTEGER_32 Index: 24515_d = $5FC3_h$
Display range (min. value | unit | max. value)			
0		2047	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00061

Parameter | Name: **C00061 | Heatsink temperature**			Data type: INTEGER_32 Index: 24514_d = $5FC2_h$
Display range (min. value | unit | max. value)			
-200	°C	200	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00062

Parameter | Name: C00062 | Temperature inside the controller			Data type: INTEGER_32 Index: 24513_d = $5FC1_h$
Display range (min. value | unit | max. value)			
-200	°C	200	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00063

Parameter | Name: C00063 | Motor temperature			Data type: INTEGER_32 Index: 24512_d = $5FC0_h$
			▸ Motor temperature monitoring
Display range (min. value | unit | max. value)			
-200	°C	200	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00064

Parameter | Name: C00064 | Device utilisation (Ixt)			Data type: UNSIGNED_32 Index: 24511_d = $5FBF_h$
Device utilisation during the last 180 seconds • C00064 > 100 % activates error (OC5). • Error reset only possible if C00064 < 95 %.			
Display range (min. value | unit | max. value)			
0	%	250	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00065

Parameter | Name: C00065 | Ext. 24-V voltage			Data type: INTEGER_32 Index: 24510_d = $5FBE_h$
Display range (min. value | unit | max. value)			
0.0	V	1000.0	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00066

Parameter | Name: C00066 | Thermal motor load (I^2xt)			Data type: UNSIGNED_32 Index: 24509_d = $5FBD_h$
A 100 % load corresponds to a permanently flowing rated motor current			
Display range (min. value | unit | max. value)			
0	%	250	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00068

Parameter | Name: C00068 | Capacitor temperature			Data type: INTEGER_32 Index: 24507_d = $5FBB_h$
Display range (min. value | unit | max. value)			
-200	°C	200	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00069

Parameter | Name: C00069 | CPU temperature			Data type: INTEGER_32 Index: 24506_d = $5FBA_h$
Display range (min. value | unit | max. value)			
-200	°C	200	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00070

Parameter | Name: **C00070 | Speed controller gain**			Data type: UNSIGNED_32 Index: 24505_d = $5FB9_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00000	Nm/rpm	20000.00000	**0.00044 Nm/rpm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100000

C00071

Parameter | Name: **C00071 | Speed contr. reset time**			Data type: UNSIGNED_32 Index: 24504_d = $5FB8_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
1.0	ms	6000.0	**14.4 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00072

Parameter | Name: **C00072 | Speed contr.D component**			Data type: UNSIGNED_32 Index: 24503_d = $5FB7_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	ms	3.00	**0.00 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00074

Parameter | Name: **C00074 | Feedfwd. ctrl. - current contr.**		Data type: UNSIGNED_8 Index: 24501_d = $5FB5_h$
Selection list (Lenze setting printed in bold)		
0	**Deactivate feedfwd. ctrl**	
1	Activate feedfwd. ctrl	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00075

Parameter | Name: **C00075 | Current controller gain**			Data type: UNSIGNED_32 Index: 24500_d = $5FB4_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	V/A	750.00	**105.00 V/A**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00076

Parameter | Name: **C00076 | Current contr. reset time**			Data type: UNSIGNED_32 Index: 24499_d = $5FB3_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.01	ms	2000.00	**2.00 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00077

Parameter | Name: **C00077 | Field controller gain**			Data type: UNSIGNED_32 Index: 24498_d = $5FB2_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	A/Vs	50000.00	**165.84 A/Vs**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00078

Parameter | Name: **C00078 | Field contr. reset time**			Data type: UNSIGNED_32 Index: 24497_d = $5FB1_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
1.0	ms	6000.0	**15.1 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00079

Parameter | Name: C00079 | Motor - mutual inductance			Data type: UNSIGNED_32 Index: 24496d = 5FB0h
Display range (min. value | unit | max. value)			
0.0	mH	214748364.7	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00080

Parameter | Name: C00080 | Resolver - number of pole pairs			Data type: UNSIGNED_32 Index: 24495d = 5FAFh
Setting range (min. value | unit | max. value)			**Lenze setting**
1		10	**1**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00081

Parameter | Name: C00081 | Rated motor power			Data type: UNSIGNED_32 Index: 24494d = 5FAEh
Setting range (min. value | unit | max. value)			**Lenze setting**
0.01	kW	500.00	**0.25 kW**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00082

Parameter | Name: C00082 | Motor - rotor resistance			Data type: UNSIGNED_32 Index: 24493d = 5FADh
Display range (min. value | unit | max. value)			
0.0000	Ohm	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C00083

Parameter | Name: C00083 | Motor - rotor time constant			Data type: UNSIGNED_32 Index: 24492d = 5FACh
Display range (min. value | unit | max. value)			
0.00	ms	21474836.47	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00084

Parameter | Name: C00084 | Motor stator resistance			Data type: UNSIGNED_32 Index: 24491d = 5FABh
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Ohm	125.0000	**18.2200 Ohm**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C00085

Parameter | Name: C00085 | Motor stator leakage induct.			Data type: UNSIGNED_32 Index: 24490d = 5FAAh
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	mH	500.000	**51.000 mH**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C00087

Parameter | Name: C00087 | Rated motor speed			Data type: UNSIGNED_32 Index: 24488d = 5FA8h
Setting range (min. value | unit | max. value)			**Lenze setting**
50	rpm	50000	**4050 rpm**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00088

Parameter | Name: **C00088 | Rated motor current**			Data type: UNSIGNED_32 Index: 24487_d = $5FA7_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.01	A	1500.00	**1.30 A**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00089

Parameter | Name: **C00089 | Rated motor frequency**			Data type: UNSIGNED_32 Index: 24486_d = $5FA6_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.1	Hz	1000.0	**270.0 Hz**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00090

Parameter | Name: **C00090 | Rated motor voltage**			Data type: UNSIGNED_32 Index: 24485_d = $5FA5_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
50	V	15000	**225 V**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00091

Parameter | Name: **C00091 | Motor - cosine phi**			Data type: UNSIGNED_32 Index: 24484_d = $5FA4_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.50		1.00	**0.80**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00092

Parameter | Name: **C00092 | Motor - magnetising current**			Data type: UNSIGNED_32 Index: 24483_d = $5FA3_h$
Display range (min. value | unit | max. value)			
0.00	A	500.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00093

Parameter | Name: **C00093 | Field weakening for SM**		Data type: UNSIGNED_32 Index: 24482_d = $5FA2_h$
From software version V2.0		▸ Field weakening for synchronous machines
Selection list (Lenze setting printed in bold)		
0	**Field weakening for SM off**	
1	Field weakening for SM on	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00099

Parameter | Name: **C00099 | Firmware version**	Data type: VISIBLE_STRING Index: 24476_d = $5F9C_h$
Format: "xx.xx.xx.xx" (main version, subversion, release version, build number)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00100

Parameter | Name: **C00100 | Resol. of an encoder revolution**			Data type: UNSIGNED_32 Index: 24475_d = $5F9B_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
10		24	**16**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00105

Parameter | Name: **C00105 | Quick stop decel. time**			Data type: UNSIGNED_32 Index: 24470_d = $5F96_h$
Time between quick stop activation and standstill plus relative S-ramp time (C00106). ▸ Basic function "Quick stop"			
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			Scaling factor: 1000

C00106

Parameter | Name: **C00106 | Quick stop S-ramp time**			Data type: UNSIGNED_32 Index: 24469_d = $5F95_h$
S-ramp time in [%] relating to the deceleration time set under C00105. ▸ Basic function "Quick stop"			
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	%	100.00	**0.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00107

Parameter | Name: **C00107 | Ref. for quick stop dec. time**		Data type: UNSIGNED_8 Index: 24468_d = $5F94_h$
Reference for the deceleration time set in C00105. ▸ Basic function "Quick stop"		
Selection list (Lenze setting printed in bold)		
0	**Motor reference speed (C00011)**	
1	Current speed	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00114

Parameter | Name: **C00114 | Dig. input x: Terminal polarity**			Data type: UNSIGNED_8 Index: 24461_d = $5F8D_h$
"0" = positive logic (HIGH level = TRUE, LOW level = FALSE) "1" = negative logic (HIGH level = FALSE, LOW level = TRUE)			
Setting range (min. value | unit | max. value)			
0		1	
Subcodes	**Lenze setting**		**Information**
C00114/1	0		Terminal polarity - digital input 1 ... 8
C00114/...			
C00114/8			
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00118

Parameter | Name: **C00118 | Dig. output. x: Terminal polarity**			Data type: UNSIGNED_8 Index: 24457_d = $5F89_h$
"0" ≡ positive logic (TRUE ≡ HIGH level, FALSE ≡ LOW level) "1" ≡ negative logic (FALSE ≡ HIGH level, TRUE ≡ LOW level)			
Setting range (min. value | unit | max. value)			
0		1	
Subcodes	**Lenze setting**		**Information**
C00118/1	0		Terminal polarity - digital output 1 ... 4
C00118/...			
C00118/4			
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00120

Parameter | Name: **C00120 | Mot. overload protection (I²xt)**			Data type: UNSIGNED_32 Index: 24455_d = $5F87_h$
Threshold for I^2 x t disconnection • Disconnection is carried out if the thermal motor load (C00066) is higher than the set threshold. • A 100 % thermal motor load corresponds to a permanently flowing rated motor current			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	200	**105 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00121

Parameter | Name: **C00121 | Motor temp. warning threshold**			Data type: UNSIGNED_32 Index: 24454_d = $5F86_h$
Temperature threshold for motor temperature advance warning • The response to reaching the threshold can be selected in C00584.			
			▶ Motor temperature monitoring
Setting range (min. value | unit | max. value)			**Lenze setting**
0	°C	150	**120 °C**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00122

Parameter | Name: **C00122 | Heatsink temp. warn. threshold**			Data type: UNSIGNED_32 Index: 24453_d = $5F85_h$
Temperature threshold for heatsink temperature advance warning • The response to reaching the threshold can be selected in C00582.			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	°C	85	**85 °C**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00123

Parameter | Name: **C00123 | Warning threshold device util.**			Data type: UNSIGNED_32 Index: 24452_d = $5F84_h$
Adjustable threshold for I x t advance warning • The advance warning is sent if the device utilisation (C00064) is higher than the set threshold. • The response to reaching the threshold can be selected in C00604.			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	100	**90 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00126

Parameter | Name: C00126 | CPU temp. warning threshold			Data type: UNSIGNED_32 Index: 24449_d = $5F81_h$
Temperature threshold for advance warning of CPU temperature on the control card • The response to reaching the threshold can be selected in C00589.			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	°C	85	**70 °C**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00127

Parameter | Name: C00127 | Mot. overload warning threshold			Data type: UNSIGNED_32 Index: 24448_d = $5F80_h$
Adjustable threshold for I^2 x t advance warning • The advance warning is sent if the thermal motor load (C00066) is higher than the set threshold. • The response to reaching the threshold can be selected in C00606.			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	200	**100 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00128

Parameter | Name: C00128 | Therm. motor time constant			Data type: UNSIGNED_32 Index: 24447_d = $5F7F_h$
Setting range (min. value | unit | max. value)			
0.1	min	600.0	
Subcodes	**Lenze setting**		**Information**
C00128/1	1.0 min		Therm. time constant coil
C00128/2	5.0 min		Therm. time constant plates
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00129

Parameter | Name: C00129 | Brake resistor value			Data type: INTEGER_32 Index: 24446_d = $5F7E_h$
Required for monitoring of the brake resistor temperature. ▸ Braking operation			
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0	Ohm	500.0	**180.0 Ohm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00130

Parameter | Name: C00130 | Rated power of brake resistor			Data type: INTEGER_32 Index: 24445_d = $5F7D_h$
Required for monitoring of the brake resistor temperature. ▸ Braking operation			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	W	800000	**5600 W**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00131

Parameter | Name: C00131 - Rated quantity of heat for brake res.			Data type: INTEGER_32 Index: 24444_d = $5F7C_h$
Required for monitoring of the brake resistor temperature. ▸ Braking operation			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	kWs	100000	**485 kWs**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00132

Parameter | Name: **C00132 | Service code**	Data type: INTEGER_32 Index: 24443_d = $5F7B_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00133

Parameter | Name: **C00133 | Ref.: Brake chopper utilisation**		Data type: UNSIGNED_8 Index: 24442_d = $5F7A_h$
From software version V1.5		
		Braking operation
Selection list (Lenze setting printed in bold)		**Information**
0	**Minimum resistance (C00134)**	C00134
1	Resistance in C00129	C00129
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00134

Parameter | Name: **C00134 | Minimum brake resistance**			Data type: INTEGER_32 Index: 24441_d = $5F79_h$
From software version V1.5			
			Braking operation
Display range (min. value | unit | max. value)			
0.0	Ohm	500.0	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00137

Parameter | Name: **C00137 | Brake transistor utilisation**			Data type: INTEGER_32 Index: 24438_d = $5F76_h$
From software version V1.5			
			Braking operation
Display range (min. value | unit | max. value)			
0	%	250	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00138

Parameter | Name: **C00138 | Brake resistor utilisation**			Data type: INTEGER_32 Index: 24437_d = $5F75_h$
From software version V1.5			
			Braking operation
Display range (min. value | unit | max. value)			
0	%	250	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00142

Parameter | Name: **C00142 | Autom. restart after mains ON** — Data type: UNSIGNED_32, Index: 24433_d = $5F71_h$

Starting performance of the controller after mains connection and reset of "Trouble", "Fault" or "safe torque off active".

⚠ **Danger!**

If the automatic restart is enabled (C00142 = "1: Enabled"), the drive can restart automatically from the "Trouble" and "Safe torque off" device states when the trouble or requirement for "Safe torque off active" is no longer available!

Automatic restart after mains connection

Selection list (Lenze setting printed in bold)		
0	**Inhibited**	
1	Enabled	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00150

Parameter | Name: **C00150 | Status word device control 1**		Data type: BITFIELD_16 Index: 24425_d = $5F69_h$
Status word 1 of the drive interface		
Display area		
0x0000	0xFFFF	
Value is bit-coded:		**Information**
Bit 0	Reserved	For the meaning of bits 8 ... 11 see chapter "Device states".
Bit 1	Pulse inhibit active	
Bit 2	Reserved	
Bit 3	Reserved	
Bit 4	Reserved	
Bit 5	Reserved	
Bit 6	Actual speed value = 0	
Bit 7	Controller inhibit active	
Bit 8	Device state - Bit 1	
Bit 9	Device state - Bit 2	
Bit 10	Device state - Bit 3	
Bit 11	Device state - Bit 4	
Bit 12	Warning active	
Bit 13	Trouble active	
Bit 14	Reserved	
Bit 15	Reserved	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00155

Parameter | Name: **C00155 | Status word device control 2**		Data type: BITFIELD_16 Index: 24420_d = $5F64_h$
Status word 2 of the drive interface		
Display area		
0x0000		0xFFFF
Value is bit-coded:		
Bit 0	Error status active	
Bit 1	Torque limit reached	
Bit 2	Current limit reached	
Bit 3	Reserved	
Bit 4	Drive switched on/in operation	
Bit 5	Drive ready for operation	
Bit 6	Fault active	
Bit 7	Drive initialisation	
Bit 8	Motor CCW rotation active	
Bit 9	Quick stop by trouble active	
Bit 10	Safe torque off active	
Bit 11	Application active	
Bit 12	Reserved	
Bit 13	Reserved	
Bit 14	Quick stop active	
Bit 15	Reserved	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00156

Parameter | Name: **C00156 | Status/Control word MCTRL**			Data type: UNSIGNED_32 Index: 24419_d = $5F63_h$
Status and control word of the motor interface			
Display range (min. value | unit | max. value)			
0		4294967295	
Subcodes			**Information**
C00156/1			Status word motor control
C00156/2			Control word motor control
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00158

Parameter | Name: **C00158 | Controller inhibit by (source)**			Data type: BITFIELD_16 Index: 24417_d = $5F61_h$
Display area			
0x0000		0xFFFF	
Value is bit-coded:			
Bit 0	Terminal		
Bit 1	Reserved		
Bit 2	Reserved		
Bit 3	Reserved		
Bit 4	Application		
Bit 5	Controller command		
Bit 6	Error response		
Bit 7	Internal PLC		
Bit 8	Reserved		
Bit 9	Reserved		
Bit 10	Operating system		
Bit 11	Identification program		
Bit 12	Brake		
Bit 13	Limiter		
Bit 14	PPI		
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00159

Parameter | Name: **C00159 | Quick stop by (source)**			Data type: BITFIELD_16 Index: 24416_d = $5F60_h$
Display area			
0x0000		0xFFFF	
Value is bit-coded:			
Bit 0	Reserved		
Bit 1	Reserved		
Bit 2	Reserved		
Bit 3	Reserved		
Bit 4	Application		
Bit 5	Controller command		
Bit 6	Error response		
Bit 7	Internal PLC		
Bit 8	Reserved		
Bit 9	Reserved		
Bit 10	Reserved		
Bit 11	Reserved		
Bit 12	Reserved		
Bit 13	Reserved		
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00162

Parameter | Name: **C00162 | Masked Error number**			Data type: UNSIGNED_32 Index: 24413_d = $5F5D_h$
From software version V5.0 Display of the individual components of the error number shown in C00168.			
Display range (min. value | unit | max. value)			
0		4294967295	
Subcodes			**Information**
C00162/1			Module ID + error number • As described in the chapter "Error messages".
C00162/2			Instance number
C00162/3			Error response
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00166

Parameter | Name: **C00166 | Error description**	Data type: VISIBLE_STRING Index: 24409_d = $5F59_h$
Error description for error number indicated in C00168	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00167

Parameter | Name: **C00167 | Service code**	Data type: VISIBLE_STRING Index: 24408_d = $5F58_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00168

Parameter | Name: **C00168 | Error number**			Data type: UNSIGNED_32 Index: 24407_d = $5F57_h$
Display of the error number of the first error with highest priority			
Display range (min. value | unit | max. value)			
0		4294967295	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00169

Parameter | Name: **C00169 | Logbook - event filter**			Data type: BITFIELD_32 Index: 24406_d = $5F56_h$

Bit coded word for filtering system events (trouble, warning, information)
- A set filter bit inhibits entry of the corresponding event into the logbook.
- From software version V5.0 the option that identical consecutive entries ("Multiple entries") into the logbook are suppressed can be additionally activated via bit 0. Then only the time stamp of the last (latest) entry and the number of times the same event has occurred successively are saved.

▶ Logbook

Setting range			**Lenze setting**
0x00000000		0xFFFFFFFF	**0x00000001** (decimal: 1)
Value is bit-coded: (☑ = bit set)			**Information**
Bit 0 ☑	No multiple entries		Bits not listed are reserved for future extensions!
Bit 1 ☐	Fault		
Bit 2 ☐	Trouble		
Bit 3 ☐	Quick stop by trouble		
Bit 4 ☐	Warning locked		
Bit 5 ☐	Warning		
Bit 6 ☐	Information		
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00171

Parameter | Name: **C00171 | Service code**	Data type: UNSIGNED_32 Index: 24404_d = $5F54_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00173

Parameter | Name: **C00173 | Mains voltage**	Data type: UNSIGNED_8 Index: 24402_d = $5F52_h$

Adjustment of the DC bus voltage thresholds
- Check during commissioning and adapt, if necessary.
- All drive components connected via the DC bus must have the same thresholds.
- For selection 0 ... 3, the undervoltage threshold is firmly defined depending on the device type (see "Rated data" chapter in the Hardware Manual).

Note: Altering this setting also has an impact on the permissible device utilisation!
In the chapter "Rated data" of the hardware manual the device types and their permissible device utilisation at a certain mains voltage and switching frequency are specified.

Selection list (Lenze setting printed in bold)		Information
0	230 V	Operation on 230 V mains • Undervoltage threshold is firmly defined. • Overvoltage threshold = 400 V • Brake chopper threshold = 390 V
1	**400/415 V**	Operation on 400 V mains/415 V mains • Undervoltage threshold is firmly defined. • Overvoltage threshold = 800 V • Brake chopper threshold = 725 V
2	460/480 V	Operation on 460 V mains/480 V mains • Undervoltage threshold is firmly defined. • Overvoltage threshold = 800 V • Brake chopper threshold = 765 V
3	500 V	Operation on 500 V mains • Undervoltage threshold is firmly defined. • Overvoltage threshold = 800 V • Brake chopper threshold = 790 V
4	230 V, LU configurable	Operation on 230 V mains • Undervoltage threshold is defined in C00174. • Overvoltage threshold = 400 V • Brake chopper threshold = 390 V
5	400/415 V, LU configurable	Operation on 400 V mains/415 V mains • Undervoltage threshold is defined in C00174. • Overvoltage threshold = 800 V • Brake chopper threshold = 725 V
6	460/480 V, LU configurable	Operation on 460 V mains/480 V mains • Undervoltage threshold is defined in C00174. • Overvoltage threshold = 800 V • Brake chopper threshold = 765 V
7	500 V, LU configurable	Operation on 500 V mains • Undervoltage threshold is defined in C00174. • Overvoltage threshold = 800 V • Brake chopper threshold = 790 V
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00174

Parameter | Name: **C00174 | Undervoltage (LU) threshold**			Data type: UNSIGNED_32 Index: 24401_d = $5F51_h$
When C00173 = 4 ... 7, the undervoltage threshold (LU) can be freely selected. **Note:** The minimum adjustable undervoltage threshold depends on the device type: • Single-axis controller (Single Drive) up to and including BF7: 210 V • Single-axis controller (Single Drive) from BF8s: 400 V • Multi-axis controller (Multi Drive): 15 V In the case of an offline parameter setting or when exchanging the memory module between different 9400 HighLine device types, always check the setting of this parameter and adapt it, if required, to prevent a parameter error after the parameter set download or module change!			
Setting range (min. value | unit | max. value)			**Lenze setting**
15	V	400	**285 V**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00175

Parameter | Name: **C00175 | Service code**	Data type: UNSIGNED_32 Index: 24400_d = $5F50_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00176

Parameter | Name: **C00176 | Service code**	Data type: UNSIGNED_32 Index: 24399_d = $5F4F_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00177

Parameter | Name: **C00177 | Service code**	Data type: UNSIGNED_32 Index: 24398_d = $5F4E_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00178

Parameter | Name: **C00178 | Elapsed hour meter**			Data type: UNSIGNED_32 Index: 24397_d = $5F4D_h$
Display range (min. value | unit | max. value)			
0	s	4294967295	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00179

Parameter | Name: **C00179 | Power-on time meter**			Data type: UNSIGNED_32 Index: 24396_d = $5F4C_h$
Display range (min. value | unit | max. value)			
0	s	4294967295	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00180

Parameter | Name: **C00180 | Service code**	Data type: VISIBLE_STRING Index: 24395_d = $5F4B_h$
For Lenze service only	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer	

C00181

Parameter | Name: **C00181 | Red. brake chopper threshold**			Data type: UNSIGNED_32 Index: 24394_d = $5F4A_h$

▸ Braking operation

Setting range (min. value | unit | max. value)			**Lenze setting**
0	V	100	**0 V**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00183

Parameter | Name: **C00183 | Device state**	Data type: UNSIGNED_32 Index: 24392_d = $5F48_h$

Display of the device state for diagnostic purposes

Selection list (display only)	
0	Operation
1	Operation/Warning active
2	Operation/warning locked act.
3	Operation/Quick stop active
4	Operation/Application stopped
10	Initialisation active
20	System fault active
90	Drive switched on
91	Drive is switched on/QSP trouble
101	Safe torque off active
102	Fault active
104	Trouble active
141	Drive ready to start --> C00142
151	Quick stop by trouble active

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00185

Parameter | Name: **C00185 | Mains recov. detect. threshold**			Data type: UNSIGNED_32 Index: 24390_d = $5F46_h$

This code must not be written to by the user!

Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	100	**90 %**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00186

Parameter | Name: **C00186 | ENP: Identified motor type**	Data type: VISIBLE_STRING Index: 24389_d = $5F45_h$

Motor type read from the electronic nameplate (ENP)

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00187

Parameter | Name: **C00187 | ENP: Identified serial number**	Data type: VISIBLE_STRING Index: 24388_d = $5F44_h$

Serial number read from the electronic nameplate (ENP)

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00188

Parameter | Name: **C00188 | ENP: Status**	Data type: UNSIGNED_8 Index: 24387_d = $5F43_h$
Selection list (display only)	
0	No ENP found
1	ENP data loaded
2	Known ENP found
3	ENP found but not read

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00198

Parameter | Name: **C00198 | Service code**	Data type: UNSIGNED_32 Index: 24377_d = $5F39_h$

This code is used internally by the controller and must not be overwritten by the user!

C00199

Parameter | Name: **C00199 | Device name**	Data type: VISIBLE_STRING Index: 24376_d = $5F38_h$

Device name to be defined by the user (e.g. "Cross cutter" or "hoist axis 1") with max. 128 characters

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00200

Parameter | Name: **C00200 | Firmware product type**	Data type: VISIBLE_STRING Index: 24375_d = $5F37_h$

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00201

Parameter | Name: **C00201 | Firmware - compiling date**	Data type: VISIBLE_STRING Index: 24374_d = $5F36_h$

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00202

Parameter | Name: **C00202 | Autom. ENP data transfer**	Data type: UNSIGNED_32 Index: 24373_d = $5F35_h$

From software version V1.5

Selection list (Lenze setting printed in bold)	
0	Off
1	**ON**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00203

Parameter | Name: **C00203 | HW product types**	Data type: VISIBLE_STRING Index: 24372_d = $5F34_h$

Subcodes	Information
C00203/1	Type: Control card
C00203/2	Type: Power section
C00203/3	Type: Module in MXI1
C00203/4	Type: Module in MXI2
C00203/5	Type: Memory module
C00203/6	Type: Safety module
C00203/7	Type: Standard device
C00203/8	Type: Complete device
C00203/9	Type: ENP

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00204

Parameter | Name: **C00204 | HW serial numbers**	Data type: VISIBLE_STRING Index: 24371_d = $5F33_h$
Subcodes	**Information**
C00204/1	Serial no.: Control card
C00204/2	Serial no.: Power section
C00204/3	Serial no.: Module in MXI1
C00204/4	Serial no.: Module in MXI2
C00204/5	Serial no.: Memory module
C00204/6	Serial no.: Safety module
C00204/7	Serial no.: Standard device
C00204/8	Serial no.: Complete device
C00204/9	Serial no.: ENP
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00205

Parameter | Name: **C00205 | HW descriptions**	Data type: VISIBLE_STRING Index: 24370_d = $5F32_h$
Subcodes	**Information**
C00205/1	Info: Control card
C00205/2	Info: Power section
C00205/3	Info: Module in MXI1
C00205/4	Info: Module in MXI2
C00205/5	Info: Memory module
C00205/6	Info: Safety module
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00206

Parameter | Name: **C00206 | HW manufacturing data**	Data type: VISIBLE_STRING Index: 24369_d = $5F31_h$
Subcodes	**Information**
C00206/1	Date: Control card
C00206/2	Date: Power section
C00206/3	Date: Module in MXI1
C00206/4	Date: Module in MXI2
C00206/5	Date: Memory module
C00206/6	Date: Safety module
C00206/7	Date: Standard device
C00206/8	Date: Complete device
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00208

Parameter | Name: **C00208 | HW manufacturer**	Data type: VISIBLE_STRING Index: 24367_d = $5F2F_h$
Subcodes	**Information**
C00208/1	Manufacturer: Control card
C00208/2	Manufacturer: Power section
C00208/3	Manufacturer: Module in MXI1
C00208/4	Manufacturer: Module in MXI2
C00208/5	Manufacturer: Memory module
C00208/6	Manufacturer: Safety module
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00209

Parameter | Name: **C00209 | HW countries of origin**	Data type: VISIBLE_STRING Index: 24366_d = $5F2E_h$
Subcodes	**Information**
C00209/1	Country: Control card
C00209/2	Country: Power section
C00209/3	Country: Module in MXI1
C00209/4	Country: Module in MXI2
C00209/5	Country: Memory module
C00209/6	Country: Safety module
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00210

Parameter | Name: **C00210 | HW versions**	Data type: VISIBLE_STRING Index: 24365_d = $5F2D_h$
Subcodes	**Information**
C00210/1	HW version: Control card
C00210/2	HW version: Power section
C00210/3	HW version: Module in MXI1
C00210/4	HW version: Module in MXI2
C00210/5	HW version: Memory module
C00210/6	HW version: Safety module
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00211

Parameter | Name: **C00211 | Application: Version**	Data type: VISIBLE_STRING Index: 24364_d = $5F2C_h$
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00212

Parameter | Name: **C00212 | Application: Type code**	Data type: VISIBLE_STRING Index: 24363_d = $5F2B_h$
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00213

Parameter | Name: **C00213 | Application: Compiler date**	Data type: VISIBLE_STRING Index: 24362_d = $5F2A_h$
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00214

Parameter | Name: **C00214 | Required safety module**	Data type: UNSIGNED_8 Index: 24361_d = $5F29_h$

Setting of the expected safety module
- If a different safety module is detected, a fault (trouble) will be activated. The fault can only be reset by mains switching.

Selection list (Lenze setting printed in bold)	
1	**SM0**
2	SM100
4	SM300
5	SM301

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00217

Parameter | Name: **C00217 | Parameter error information**	Data type: UNSIGNED_32 Index: 24358_d = $5F26_h$

This code is used internally by the controller and must not be overwritten by the user!

C00218

Parameter | Name: **C00218 | Application: ID number**	Data type: UNSIGNED_32 Index: 24357_d = $5F25_h$

Display range (min. value | unit | max. value)		
0		0

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00219

Parameter | Name: **C00219 | CAN/EPL device type**	Data type: UNSIGNED_32 Index: 24356_d = $5F24_h$

This code is used internally by the controller and must not be overwritten by the user!

C00225

Parameter | Name: **C00225 | Check configuration**	Data type: UNSIGNED_32 Index: 24350_d = $5F1E_h$

This code is used internally by the controller and must not be overwritten by the user!

C00227

Parameter | Name: **C00227 | Behaviour due to change of parameter set**	Data type: UNSIGNED_32 Index: 24348_d = $5F1C_h$

From software version V5.0

By selecting "1" in the corresponding subcode a module plugged into module slot MXI1 or MXI2 can be excepted from the parameter set changeover via the device command "Activate parameter set n".
- By this the parameter set changeover, in particular for active modules, is carried out much more quickly.
- An exception from the parameter set changeover for instance is reasonable if different parameter sets are used (e. g. for different tools within the machine), but if the parameters are always the same for the module (e. g. communication parameters).

Selection list (Lenze setting printed in bold)	
0	Included
1	Excluded

Subcodes	Lenze setting	Information
C00227/1	0: Included	Change of parameter set: MXI1
C00227/2	0: Included	Change of parameter set: MXI2

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00254

Parameter | Name: **C00254 | Phase controller gain**			Data type: UNSIGNED_32 Index: 24321_d = $5F01_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	1/s	1000.00	**20.00 1/s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00270

Parameter | Name: **C00270 | Freq. - current setpoint filter**			Data type: UNSIGNED_32 Index: 24305_d = $5EF1_h$
			▶ Set current setpoint filter (band-stop filter)
Setting range (min. value | unit | max. value)			
1.0	Hz	1000.0	
Subcodes	**Lenze setting**		**Information**
C00270/1	200.0 Hz		Freq. - current setp. filter 1
C00270/2	400.0 Hz		Freq. - current setp. filter 2
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00271

Parameter | Name: **C00271 | Width - current setp. filter**			Data type: UNSIGNED_32 Index: 24304_d = $5EF0_h$
			▶ Set current setpoint filter (band-stop filter)
Setting range (min. value | unit | max. value)			
0.0	Hz	500.0	
Subcodes	**Lenze setting**		**Information**
C00271/1	20.0 Hz		Width current setp. filter 1
C00271/2	40.0 Hz		Width current setp. filter 2
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00272

Parameter | Name: **C00272 | Depth - current setp. filter**			Data type: UNSIGNED_32 Index: 24303_d = $5EEF_h$
The setting "0 dB" deactivates the current setpoint filter.			
			▶ Set current setpoint filter (band-stop filter)
Setting range (min. value | unit | max. value)			
0	db	100	
Subcodes	**Lenze setting**		**Information**
C00272/1	0 db		Depth current setp. filter 1
C00272/2	0 db		Depth current setp. filter 2
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00273

Parameter | Name: **C00273 | Moment of inertia**			Data type: UNSIGNED_32 Index: 24302_d = $5EEE_h$
Note: The load moment of inertia must be set with regard to the motor end (i.e. considering the gearbox factors).			
Setting range (min. value | unit | max. value)			
0.00	kg cm²	20000000.00	
Subcodes	**Lenze setting**		**Information**
C00273/1	0.14 kg cm²		Motor moment of inertia
C00273/2	0.00 kg cm²		Load moment of inertia
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00274

Parameter | Name: **C00274 | Max. acceleration change**			Data type: UNSIGNED_32 Index: 24301_d = $5EED_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0	%/ms	400.0	**400.0 %/ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00275

Parameter | Name: **C00275 | Signal source - speed setpoint**		Data type: UNSIGNED_16 Index: 24300_d = $5EEC_h$
Selection list (Lenze setting printed in bold)		
0	**SpeedAdd signal**	
1	Differentiated PosSet signal	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00276

Parameter | Name: **C00276 | Signal source - torque setpoint**		Data type: UNSIGNED_16 Index: 24299_d = $5EEB_h$
Selection list (Lenze setting printed in bold)		**Information**
0	**TorqueAdd/AccAdd signal**	
1	Differentiated SpeedSet signal	
2	2x diff. PosSet signal	
3	Differentiated SpeedAdd signal	From software version V5.0 This alternative selection to the differentiated SpeedSet signal is recommended if the position controller works with a high gain. By the position controller also troubles within to the actual position value are detected and like this reach the speed setpoint. In the following differentiation of the feedforward control, these troubles in particular in the case of high position controller gains result in a very unsettled torque feedforward control value. By means of this selection the problem can be avoided, because then only the trouble-free speed feedforward control value is differentiated.
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00280

Parameter | Name: **C00280 | Filter time const. DC detection**			Data type: UNSIGNED_32 Index: 24295_d = $5EE7_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
1.0	ms	1000.0	**25.0 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00281

Parameter | Name: **C00281 | Filter for PWM adjustment**	Data type: UNSIGNED_8 Index: 24294_d = $5EE6_h$

From software version V7.0

The output voltage of the inverter is generated from the DC-bus voltage by a pulse width modulation (PWM). The product from the DC-bus voltage and the control factor of the PWM corresponds to the height of the output voltage. If the DC-bus voltage changes due to mains fluctuations or load changes, the control factor of the PWM must be adapted if the output voltage is to remain constant. This correction is carried out in the control software by measuring the DC-bus voltage. In order that no response takes place to faults in the DC-bus voltage measurement, a filter can be activated for the measured signal via this parameter.

- The filter time constant is selected so that a quick correction can be carried out even under bad EMC conditions.
- A disadvantage of this filter is that the responses to real flickers on the DC-bus voltage are too slow. In positioning tasks, this may cause an extreme increase of the following error.
- Under good EMC conditions, this filter is not required (Lenze setting).
- For devices > BF7 the filter is ineffective.

Selection list (Lenze setting printed in bold)		
0	**Deactivated**	
1	Activated	

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00308

Parameter | Name: **C00308 | Service code**	Data type: UNSIGNED_16 Index: 24267_d = $5ECB_h$

This code is used internally by the controller and must not be overwritten by the user!

C00309

Parameter | Name: **C00309 | Service code**	Data type: UNSIGNED_32 Index: 24266_d = $5ECA_h$

This code is used internally by the controller and must not be overwritten by the user!

C00310

Parameter | Name: **C00310 | Service code**	Data type: UNSIGNED_8 Index: 24265_d = $5EC9_h$

This code is used internally by the controller and must not be overwritten by the user!

C00311

Parameter | Name: **C00311 | CAN TPDO1 mask byte x**	Data type: BITFIELD_8 Index: 24264_d = $5EC8_h$

A mask can be parameterised for each byte of the TPDO1 in the assigned subcode.

- In case of an event-controlled PDO transmission, only the masked bits will be considered for event control.
- Mask "0x0" means that no bit of the corresponding byte actuates the transmission.
- Mask "0xff" means that every bit of the corresponding byte can actuate the transmission.

▸ "CAN on board" system bus

Setting range			
0x00		0xFF	
Value is bit-coded:			
Bit 0	Mask bit 0		
...	...		
Bit 7	Mask bit 7		

Subcodes	Lenze setting	Information
C00311/1	0x00	Mask for byte 1 ... byte 8 of TPDO1
C00311/...		
C00311/8		

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00312

Parameter | Name: **C00312 | CAN TPDO2 mask byte x**			Data type: BITFIELD_8 Index: 24263_d = $5EC7_h$

A mask can be parameterised for each byte of the TPDO2 in the assigned subcode.
- In case of an event-controlled PDO transmission, only the masked bits will be considered for event control.
- Mask "0x0" means that no bit of the corresponding byte actuates the transmission.
- Mask "0xff" means that every bit of the corresponding byte can actuate the transmission.

"CAN on board" system bus

Setting range			
0x00		0xFF	
Value is bit-coded:			
Bit 0	Mask bit 0		
...	...		
Bit 7	Mask bit 7		

Subcodes	Lenze setting	Information
C00312/1	0x00	Mask for byte 1 ... byte 8 of TPDO2
C00312/...		
C00312/8		

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00313

Parameter | Name: **C00313 | CAN TPDO3 mask byte x**			Data type: BITFIELD_8 Index: 24262_d = $5EC6_h$

A mask can be parameterised for each byte of the TPDO3 in the assigned subcode.
- In case of an event-controlled PDO transmission, only the masked bits will be considered for event control.
- Mask "0x0" means that no bit of the corresponding byte actuates the transmission.
- Mask "0xff" means that every bit of the corresponding byte can actuate the transmission.

"CAN on board" system bus

Setting range			
0x00		0xFF	
Value is bit-coded:			
Bit 0	Mask bit 0		
...	...		
Bit 7	Mask bit 7		

Subcodes	Lenze setting	Information
C00313/1	0x00	Mask for byte 1 ... byte 8 of TPDO3
C00313/...		
C00313/8		

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00314

Parameter | Name: **C00314 | CAN TPDO4 mask byte x**		Data type: BITFIELD_8 Index: 24261_d = $5EC5_h$

A mask can be parameterised for each byte of the TPDO4 in the assigned subcode.
- In case of an event-controlled PDO transmission, only the masked bits will be considered for event control.
- Mask "0x0" means that no bit of the corresponding byte actuates the transmission.
- Mask "0xff" means that every bit of the corresponding byte can actuate the transmission.

"CAN on board" system bus

Setting range			
0x00		0xFF	
Value is bit-coded:			
Bit 0	Mask bit 0		
...	...		
Bit 7	Mask bit 7		

Subcodes	Lenze setting	Information
C00314/1	0x00	Mask for byte 1 ... byte 8 of TPDO4
C00314/...		
C00314/8		

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00320

Parameter | Name: **C00320 | CAN TPDOx identifier**		Data type: BITFIELD_32 Index: 24255_d = $5EBF_h$

Identifier for TPDO1 ... TPDO4
- If bit 31 is set (0x8nnnnnnn), the TPDO is deactivated.
- The basic setting is according to the "Predefined Connection Set".
- Mapping of the CANopen objects I-1800/1 ... I-1803/1 (see DS301 V4.02).

"CAN on board" system bus

Setting range			
0x00000000		0xFFFFFFFF	
Value is bit-coded:			**Information**
Bit 0	COB-ID bit 0		• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • Bit 31: PDO invalid
...	...		
Bit 31	PDO invalid		

Subcodes	Lenze setting	Information
C00320/1	0x00000181	Identifier TPDO1 • After a node address change and CAN reset node, the value 0x180 + node address will be set by default.
C00320/2	0x00000281	Identifier TPDO2 • After a node address change and CAN reset node, the value 0x280 + node address will be set by default.
C00320/3	0x00000381	Identifier TPDO3 • After a node address change and CAN reset node, the value 0x380 + node address will be set by default.
C00320/4	0x00000481	Identifier TPDO4 • After a node address change and CAN reset node, the value 0x480 + node address will be set by default.

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00321

Parameter | Name: **C00321 | CAN RPDOx identifier**			Data type: BITFIELD_32 Index: 24254_d = $5EBE_h$
Identifier for RPDO1 ... RPDO4 • If bit 31 is set (0x8nnnnnnn), the RPDO is deactivated. • The basic setting is according to the "Predefined Connection Set". • Mapping of the CANopen objects I-1400/1 ... I-1403/1 (see DS301 V4.02). ▸ "CAN on board" system bus			
Setting range			
0x00000000		0xFFFFFFFF	
Value is bit-coded:			**Information**
Bit 0	COB-ID bit 0		• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • Bit 31: PDO invalid
...	...		
Bit 31	PDO invalid		
Subcodes	**Lenze setting**		**Information**
C00321/1	0x00000201		Identifier RPDO1 • After a node address change and CAN reset node, the value 0x200 + node address will be set by default.
C00321/2	0x00000301		Identifier RPDO2 • After a node address change and CAN reset node, the value 0x300 + node address will be set by default.
C00321/3	0x00000401		Identifier RPDO3 • After a node address change and CAN reset node, the value 0x400 + node address will be set by default.
C00321/4	0x00000501		Identifier RPDO4 • After a node address change and CAN reset node, the value 0x500 + node address will be set by default.
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00322

Parameter | Name: **C00322 | CAN TPDOx Tx mode**			Data type: UNSIGNED_8 Index: 24253_d = $5EBD_h$
TPDO transmission type according to DS301 V4.02 • Types 0 (acyclic sync), 1-240 (cyclic sync), 254 (event-controlled manufacturer-specific), 255 (event-controlled device-profile-specific) are supported. • The basic PDO setting is "254" (event-controlled). • Mapping of the CANopen objects I-1800/2 ... I-1803/2 (see DS301 V4.02). ▸ "CAN on board" system bus			
Setting range (min. value | unit | max. value)			
0		255	
Subcodes	**Lenze setting**		**Information**
C00322/1	254		Transmission mode for TPDO1 ... TPDO4
C00322/...			
C00322/4			
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00323

Parameter | Name: **C00323 | CAN RPDOx Rx mode**			Data type: UNSIGNED_8 Index: 24252_d = $5EBC_h$

RPDO Transmission type according to DS301 V4.02
- For the RPDO serves as monitoring setting in the case of sync-controlled PDOs.
- Types 0 (acyclic sync), 1-240 (cyclic sync), 254 (event-controlled manufacturer-specific), 255 (event-controlled device-profile-specific) are supported.
- The basic PDO setting is "254" (event-controlled).
- Mapping of the CANopen objects I-1400/2 ... I-1403/2 (see DS301 V4.02).

▶ "CAN on board" system bus

Setting range (min. value | unit | max. value)			
0		255	
Subcodes	**Lenze setting**		**Information**
C00323/1	254		Transmission mode for RPDO1 ... RPDO4
C00323/...			
C00323/4			

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00324

Parameter | Name: **C00324 | CAN TPDOx delay time**			Data type: UNSIGNED_16 Index: 24251_d = $5EBB_h$

TPDO inhibit time according to DS301 V4.02
- Minimum time between the transmission of two identical TPDOs.
- The delay time is entered in 1/10 ms and automatically rounded to full milliseconds by the code.
- Mapping of the CANopen objects I-1800/3 ... I-1803/3 (see DS301 V4.02).

▶ "CAN on board" system bus

Setting range (min. value | unit | max. value)			
0	1/10 ms	65535	
Subcodes	**Lenze setting**		**Information**
C00324/1	0 1/10 ms		Delay time for TPDO1 ... TPDO4
C00324/...			
C00324/4			

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00325

Parameter | Name: **C00325 | Service code**	Data type: UNSIGNED_8 Index: 24250_d = $5EBA_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00326

Parameter | Name: **C00326 | Service code**	Data type: UNSIGNED_8 Index: 24249_d = $5EB9_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00327

Parameter | Name: **C00327 | Service code**	Data type: BITFIELD_32 Index: 24248_d = $5EB8_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00328

Parameter | Name: **C00328 | Service code**	Data type: BITFIELD_32 Index: 24247_d = $5EB7_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00329

Parameter | Name: **C00329 | Service code**	Data type: BITFIELD_32 Index: 24246_d = $5EB6_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00330

Parameter | Name: **C00330 | Service code**	Data type: BITFIELD_32 Index: 24245_d = $5EB5_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00335

Parameter | Name: **C00335 | Service code**	Data type: BITFIELD_32 Index: 24240_d = $5EB0_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00336

Parameter | Name: **C00336 | Service code**	Data type: BITFIELD_32 Index: 24239_d = $5EAF_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00337

Parameter | Name: **C00337 | Service code**	Data type: BITFIELD_32 Index: 24238_d = $5EAE_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00338

Parameter | Name: **C00338 | Service code**	Data type: BITFIELD_32 Index: 24237_d = $5EAD_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00343

Parameter | Name: **C00343 | CAN TPDO counter**			Data type: UNSIGNED_32 Index: 24232_d = $5EA8_h$
Display range (min. value | unit | max. value)			
0		4294967295	
Subcodes			**Information**
C00343/1			From software version V1.5 Counter for TPDO1 ... TPDO4 ▶ "CAN on board" system bus
C00343/...			
C00343/4			
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00344

Parameter | Name: **C00344 | CAN RPDO counter**			Data type: UNSIGNED_32 Index: 24231_d = $5EA7_h$
Display range (min. value | unit | max. value)			
0		4294967295	
Subcodes			**Information**
C00344/1			From software version V1.5 Counter for RPDO1 ... RPDO4 ▶ "CAN on board" system bus
C00344/...			
C00344/4			
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00345

Parameter | Name: **C00345 | CAN error**	Data type: UNSIGNED_8 Index: 24230_d = $5EA6_h$

▸ "CAN on board" system bus

Selection list (display only)	
0	No error
1	Guard Event
2	Warning
3	Bus off
4	Sync telegram error
6	CAN controller overflow

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00346

Parameter | Name: **C00346 | CAN heartbeat activity**	Data type: BITFIELD_32 Index: 24229_d = $5EA5_h$

▸ "CAN on board" system bus: heartbeat protocol

Display area		
0x00000000		0xFFFFFFFF

Value is bit-coded:	
Bit 0	Heartbeat node 1
...	...
Bit 31	Heartbeat node 32

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00347

Parameter | Name: **C00347 | CAN heartbeat status**	Data type: UNSIGNED_8 Index: 24228_d = $5EA4_h$

▸ "CAN on board" system bus: heartbeat protocol

Selection list (display only)	
0	Unknown
4	Stopped
5	Operational
127	Pre-operational

Subcodes	**Information**
C00347/1	Status node 1 ... 32
C00347/...	
C00347/32	

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00348

Parameter | Name: **C00348 | CAN status DIP switch**	Data type: UNSIGNED_8 Index: 24227_d = $5EA3_h$

- "1" means that the CAN DIP switch has been identified after mains switching and a valid baud rate and node address have been set.
- "0" means that no CAN DIP switch or no valid setting has been identified or the setting has been overwritten by writing to code C00350 or C00351.

▸ "CAN on board" system bus

Selection list (display only)		**Information**
0	Setting not accepted	
1	Setting accepted	

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00349

Parameter | Name: **C00349 | CAN setting of DIP switch**				Data type: UNSIGNED_8 Index: 24226_d = $5EA2_h$
Setting of the CAN DIP switch at the last mains connection				
				▶ "CAN on board" system bus
Display range (min. value | unit | max. value)				
0		255		
Subcodes			**Information**	
C00349/1			Node address	
C00349/2			Baud rate: 0: 500 kbps 1: 250 kbps 2: 125 kbps 3: 50 kbps 4: 1 Mbps 5: 20 kbps 6: 10 kbit/s 14: 800 kbps 255: Automatic recognition	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1	

C00350

Parameter | Name: **C00350 | CAN node address**				Data type: UNSIGNED_8 Index: 24225_d = $5EA1_h$
• A change of the node address only gets active after a CAN reset node. • The basic server channel RX/TX is automatically provided by the node address (C00372 and C00373; subcode 1). • Overwriting the value deactivates a possibly existing node address selection entered by means of hardware.				
				▶ "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			Scaling factor: 1	

C00351

Parameter | Name: **C00351 | CAN baud rate**	Data type: UNSIGNED_8 Index: 24224_d = $5EA0_h$

- A change of the baud rate only gets active after a CAN reset node.
- Overwriting the value deactivates a possibly existing node address selection entered by means of hardware.

▶ "CAN on board" system bus

Selection list (Lenze setting printed in bold)	
0	**500 kbps**
1	250 kbps
2	125 kbps
3	50 kbps
4	1 Mbps
5	20 kbps
6	10 kbps
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	800 kbps
15	Reserved
255	Automatic recognition

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00352

Parameter | Name: **C00352 | CAN slave/master**	Data type: UNSIGNED_8 Index: 24223_d = $5E9F_h$

If "1" is entered and saved, the drive will start as CAN master after mains switching.

▶ "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 Scaling factor: 1

C00356

Parameter | Name: **C00356 | CAN TPDOx cycle time**	Data type: UNSIGNED_16 Index: 24219_d = $5E9B_h$

TPDO event time according to DS301 V4.02
- If the value is entered that is not "0", the TPDO is transmitted without further consideration of the transport type after the set time has elapsed.
- Mapping of the CANopen objects I-1800/5 ... I-1803/5 (see DS301 V4.02).

▶ "CAN on board" system bus

Setting range (min. value | unit | max. value)		
0	ms	65535

Subcodes	Lenze setting	Information
C00356/1	0 ms	Cycle time for TPDO1 ... TPDO4
C00356/...		
C00356/4		

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00357

Parameter | Name: C00357 | CAN RPDOx monitoring time			Data type: UNSIGNED_16 Index: 24218_d = $5E9A_h$

Mapping of the RPDO event time (see DS301 V4.02)
- If a value is entered that is not "0", the RPDO is expected after the time set has elapsed.
- If the RPDO is not received within this time, a parameterisable error message can be activated.

▶ "CAN on board" system bus

Setting range (min. value | unit | max. value)			
0	ms	65535	
Subcodes	**Lenze setting**		**Information**
C00357/1	3000 ms		Monitoring time for RPDO1 ... RPDO4
C00357/...			
C00357/4			

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer 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 (display only)	
0	Operational
1	Pre-operational
4	Boot up
5	Stopped
7	Reset
8	Initialisation
9	Unknown
10	Baud rate autom. detected

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00360

Parameter | Name: **C00360 | CAN telegram and error counter**			Data type: UNSIGNED_16 Index: 24215_d = $5E97_h$

- After mains connection all counters start with "0".
- After the maximum value has been reached, counting restarts with "0".

▶ "CAN on board" system bus

Display range (min. value | unit | max. value)			
0		65535	

Subcodes	Information
C00360/1	Stuffing bit error counter • More than five identical bits have been detected.
C00360/2	Format error counter • CAN frame has not been observed.
C00360/3	Acknowledge error counter • No device has acknowledged the telegram.
C00360/4	Bit1 error counter • "1" should be sent after bus arbitration, but "0" was read.
C00360/5	Bit0 error counter • "0" should be sent after bus arbitration, but "1" was read.
C00360/6	CRC error counter • Checksum check has indicated an error.
C00360/7	Tx telegram counter • Correctly received telegrams.
C00360/8	Rx telegram counters • Correctly transmitted telegrams.
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00361

Parameter | Name: **C00361 | CAN bus load**			Data type: UNSIGNED_32 Index: 24214_d = $5E96_h$

The display of the node peak load (subcodes 4 ... 6) is reset by repeated mains switching or via the "Reset node" device command (C00002).

▶ "CAN on board" system bus

Display range (min. value | unit | max. value)			
0	%	100	

Subcodes	Information
C00361/1	Current node load in Tx direction
C00361/2	Current node load in Rx direction
C00361/3	Current node load through faulty telegrams
C00361/4	Maximum node load in Tx direction
C00361/5	Maximum node load in Rx direction
C00361/6	Maximum node load through faulty telegrams
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00367

Parameter | Name: **C00367 | CAN SYNC Rx identifier**			Data type: UNSIGNED_32 Index: 24208_d = $5E90_h$
Identifier with 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: sync telegram			
Setting range (min. value | unit | max. value)			**Lenze setting**
0		2047	**128**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00368

Parameter | Name: **C00368 | CAN SYNC Tx identifier**			Data type: UNSIGNED_32 Index: 24207_d = $5E8F_h$
Identifier with which the sync master is to send sync telegrams. • Mapping of the CANopen object I-1005 (see DS301 V4.02). ▸ "CAN on board" system bus: sync telegram			
Setting range (min. value | unit | max. value)			**Lenze setting**
0		2047	**128**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00369

Parameter | Name: **C00369 | CAN SYNC transmit cycle time**			Data type: UNSIGNED_16 Index: 24206_d = $5E8E_h$
Cycle within which the sync master is to send sync telegrams. • With "0 ms" (Lenze setting), no sync telegrams are created. • Mapping of the CANopen object I-1006 (see DS301 V4.02). ▸ "CAN on board" system bus: sync telegram			
Setting range (min. value | unit | max. value)			
0	ms	65535	
Subcodes	**Lenze setting**		**Information**
C00369/1	0 ms		Transmission cycle time for CAN on board
C00369/2	0 ms		Transmission cycle time for CAN module in MXI1/MXI2
C00369/3	0 ms		- (no meaning)
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00372

Parameter | Name: **C00372 | CAN SDO server Rx identifier**		Data type: BITFIELD_32 Index: 24203_d = $5E8B_h$

Identifier which serves to reach the assigned SDO server.
- If bit 31 is set (0x8nnnnnnn), the corresponding SDO server is deactivated.
- Mapping of the CANopen objects I-1200/1 ... I-1209/1 (see DS301 V4.02).

▶ "CAN on board" system bus

Setting range			
0x00000000		0xFFFFFFFF	

Value is bit-coded:		Information
Bit 0	COB-ID bit 0	• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • Bit 31: SDO invalid
...	...	
Bit 31	SDO invalid	

Subcodes	Lenze setting	Information
C00372/1	0x00000601	SDO server channel 1 RX • Subcode 1 contains the basic SDO channel which can neither be changed nor deactivated, according to DS301 V4.02. Writing to the subcode has no effect. • The value under subcode 1 results from the node address (C00350) + 0x600.
C00372/2	0x80000000	SDO server channel 2 RX
C00372/3	0x80000000	SDO server channel 3 RX
C00372/4	0x80000000	SDO server channel 4 RX
C00372/5	0x80000000	SDO server channel 5 RX
C00372/6	0x80000000	SDO server channel 6 RX
C00372/7	0x80000000	SDO server channel 7 RX
C00372/8	0x80000000	SDO server channel 8 RX
C00372/9	0x80000000	SDO server channel 9 RX
C00372/10	0x80000000	SDO server channel 10 RX

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00373

Parameter | Name: **C00373 | CAN SDO server Tx identifier**		Data type: BITFIELD_32 Index: 24202_d = $5E8A_h$

Identifier with which the assigned SDO server is able to transmit.
- If bit 31 is set (0x8nnnnnnn), the corresponding SDO server is deactivated.
- Mapping of the CANopen objects I-1200/2 ... I-1209/2 (see DS301 V4.02).

"CAN on board" system bus

Setting range			
0x00000000		0xFFFFFFFF	

Value is bit-coded:		Information
Bit 0	COB-ID bit 0	• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • Bit 31: SDO invalid
...	...	
Bit 31	SDO invalid	

Subcodes	Lenze setting	Information
C00373/1	0x00000581	SDO server channel 1 TX • Subcode 1 contains the basic SDO channel which can neither be changed nor deactivated, according to DS301 V4.02. Writing to the subcode has no effect. • The value under subcode 1 results from the node address (C00350) + 0x580.
C00373/2	0x80000000	SDO server channel 2 TX
C00373/3	0x80000000	SDO server channel 3 TX
C00373/4	0x80000000	SDO server channel 4 TX
C00373/5	0x80000000	SDO server channel 5 TX
C00373/6	0x80000000	SDO server channel 6 TX
C00373/7	0x80000000	SDO server channel 7 TX
C00373/8	0x80000000	SDO server channel 8 TX
C00373/9	0x80000000	SDO server channel 9 TX
C00373/10	0x80000000	SDO server channel 10 TX

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00374

Parameter | Name: **C00374 | CAN SDO client node address**			Data type: UNSIGNED_8 Index: 24201_d = $5E89_h$
Node address of the client assigned to this server (see DS301 V4.02).			"CAN on board" system bus
Setting range (min. value | unit | max. value)			
1		127	
Subcodes	**Lenze setting**		**Information**
C00374/1	1		SDO server channel 1 remote client node address • Subcode 1 contains the basic SDO channel which, according to DS301 V4.02, does not feature this entry. Writing to the subcode has no effect. • The value of subindex 1 results in 0.
C00374/2	1		SDO server channel 2 remote client node address
C00374/3	1		SDO server channel 3 remote client node address
C00374/4	1		SDO server channel 4 remote client node address
C00374/5	1		SDO server channel 5 remote client node address
C00374/6	1		SDO server channel 6 remote client node address
C00374/7	1		SDO server channel 7 remote client node address
C00374/8	1		SDO server channel 8 remote client node address
C00374/9	1		SDO server channel 9 remote client node address
C00374/10	1		SDO server channel 10 remote client node address
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00375

Parameter | Name: **C00375 | CAN SDO client Rx identifier**			Data type: BITFIELD_32 Index: 24200_d = $5E88_h$
Identifier which serves to reach the assigned SDO client. • If bit 31 is set (0x8nnnnnnn), the corresponding SDO client channel is deactivated (see DS301 V4.02). • The client channels need not be parameterised right now. Their functionality will only be required when using the gateway services.			"CAN on board" system bus
Setting range			
0x00000000		0xFFFFFFFF	
Value is bit-coded:			**Information**
Bit 0	COB-ID bit 0		• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • Bit 31: SDO invalid
...	...		
Bit 31	SDO invalid		
Subcodes	**Lenze setting**		**Information**
C00375/1	0x80000000		SDO client channel 1 RX ... 10 RX
C00375/...			
C00375/10			
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00376

Parameter | Name: **C00376 | CAN SDO client Tx identifier**		Data type: BITFIELD_32 Index: 24199_d = $5E87_h$

Identifier with which the assigned SDO client is able to transmit.
- If bit 31 is set (0x8nnnnnnn), the corresponding SDO client channel is deactivated (see DS301 V4.02).
- The client channels need not be parameterised right now. Their functionality will only be required when using the gateway services.

"CAN on board" system bus

Setting range			
0x00000000		0xFFFFFFFF	
Value is bit-coded:			**Information**
Bit 0	COB-ID bit 0		• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • Bit 31: SDO invalid
...	...		
Bit 31	SDO invalid		
Subcodes	**Lenze setting**		**Information**
C00376/1	0x80000000		SDO client channel 1 TX ... 10 TX
C00376/...			
C00376/10			

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00377

Parameter | Name: **C00377 | CAN SDO server node address**		Data type: UNSIGNED_8 Index: 24198_d = $5E86_h$

Node address of the server with which the SDO client communicates via the selected client channel.
- The client functionality need not be activated.
- The entry is required so that the CAN-SDO client channel can be activated (see DS301 V4.02).

"CAN on board" system bus

Setting range (min. value | unit | max. value)			
1		127	
Subcodes	**Lenze setting**		**Information**
C00377/1	1		Remote server node address for SDO client channel 1 ... 10
C00377/...			
C00377/10			

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00378

Parameter | Name: **C00378 | CAN delay boot-up - Operational**		Data type: UNSIGNED_16 Index: 24197_d = $5E85_h$

Time that has to elapse after mains switching before the CAN NMT master places the "Start Remote Node" telegram to the bus.
- This time is only used if the master bit is activated (C00352) and after mains switching.

"CAN on board" system bus

Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	65535	**3000 ms**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00379

Parameter | Name: **C00379 | Service code**	Data type: UNSIGNED_8 Index: 24196_d = $5E84_h$

This code is used internally by the controller and must not be overwritten by the user!

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 one or several consumers.
- The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is transmitted automatically as soon as a time > 0 ms is set. The monitoring function "Node Guarding" is deactivated in this case.
- Mapping of the CANopen object I-1017 (see DS301 V4.02).

▶ "CAN on board" system bus: heartbeat protocol

Setting range (min. value | unit | max. value)			Lenze setting
0	ms	65535	**0 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00382

Parameter | Name: **C00382 | CAN Guard Time**			Data type: UNSIGNED_16 Index: 24193_d = $5E81_h$

After the set guard time multiplied by the life time factor (C00383), a node guarding telegram must have been received.
- Mapping of the CANopen object I-100C (see DS301 V4.02).

▶ "CAN on board" system bus: node guarding protocol

Setting range (min. value | unit | max. value)			Lenze setting
0	ms	65535	**0 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00383

Parameter | Name: **C00383 | CAN Life Time Factor**			Data type: UNSIGNED_8 Index: 24192_d = $5E80_h$

The life time factor multiplied by the guard time (C00382) results in the time in which a node guarding telegram must have been received.
- Mapping of the CANopen object I-100D (see DS301 V4.02).

▶ "CAN on board" system bus: node guarding protocol

Setting range (min. value | unit | max. value)			Lenze setting
0		255	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00385

Parameter | Name: **C00385 | CAN Heartbeat Consumer Time**			Data type: BITFIELD_32 Index: 24190_d = $5E7E_h$

The 32 subcodes represent the nodes to be monitored by means of heartbeat.

- Each subcode entry contains the expected "heartbeat" time and the node ID (node address) from which the heartbeat telegram is expected in the form of a bit code.
- The parameterised time is rounded down to an integer multiple of 5 ms and must have a higher value than the heartbeat producer time of the node to be monitored.
- The response to a missing heartbeat telegram can be parameterised in C00613.
- Mapping of the CANopen object I-1016 (see DS301 V4.02).

"CAN on board" system bus: heartbeat protocol

Setting range			
0x00000000		0xFFFFFFFF	
Value is bit-coded:			**Information**
Bit 0	Heartbeat time bit 0		• Bit 0 ... 15: Heartbeat time • Bit 16 ... 23: Node address • Bit 24 ... 31: Reserved
...	...		
Bit 31	Reserved		
Subcodes	**Lenze setting**		**Information**
C00385/1	0x00000000		Monitoring entry 1 ... 32
C00385/...			
C00385/32			

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00386

Parameter | Name: **C00386 | CAN Node Guarding**			Data type: BITFIELD_32 Index: 24189_d = $5E7D_h$

The 32 subcodes represent the nodes to be monitored by the master by means of node guarding.

- Each subcode entry contains the guard time, the lifetime factor and the node ID (node address) from which the heartbeat telegram is expected in the form of a bit code.
- The response to a missing node guarding response can be parameterised in C00612.

"CAN on board" system bus: node guarding protocol

Setting range			
0x00000000		0xFFFFFFFF	
Value is bit-coded:			**Information**
Bit 0	Guard time bit 0		• Bit 0 ... 15: Guard time • Bit 16 ... 23: Node address • Bit 24 ... 31: Lifetime factor
...	...		
Bit 31	Lifetime factor bit 7		
Subcodes	**Lenze setting**		**Information**
C00386/1	0x00000000		Monitoring entry 1 ... 32
C00386/...			
C00386/32			

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00387

Parameter | Name: **C00387 | CAN Node Guarding Activity**			Data type: BITFIELD_32 Index: 24188_d = $5E7C_h$
			"CAN on board" system bus: node guarding protocol
Display area			
0x00000000		0xFFFFFFFF	
Value is bit-coded:			
Bit 0	Node guarding of node 1		
...	...		
Bit 31	Node guarding of node 32		
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00388

Parameter | Name: **C00388 | CAN node guarding status**		Data type: UNSIGNED_8 Index: 24187_d = $5E7B_h$
		"CAN on board" system bus: node guarding protocol
Selection list (display only)		
0	Unknown	
4	Stopped	
5	Operational	
127	Pre-operational	
Subcodes		**Information**
C00388/1		Status node 1 ... 32
C00388/...		
C00388/32		
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00390

Parameter | Name: **C00390 | CAN Error Register (DS301V402)**			Data type: BITFIELD_8 Index: 24185_d = $5E79_h$
Mapping of the CANopen object I-1001 (see DS301 V4.02).			
			"CAN on board" system bus
Display area			
0x00		0xFF	
Value is bit-coded:			**Information**
Bit 0	Generic error		Currently only bit 0 and bit 4 contain the corresponding information.
Bit 1	Current error (not used)		
Bit 2	Voltage error (not used)		
Bit 3	Temperature error (not used)		
Bit 4	Communication error		
Bit 5	Dev. prof. spec. err (not used)		
Bit 6	Reserved		
Bit 7	Manuf. spec. error (not used)		
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00391

Parameter | Name: **C00391 | CAN Emergency Object**			Data type: BITFIELD_32 Index: 24184_d = $5E78_h$
Identifier of the emergency telegram • Setting bit 31 of this code (0x8nnnnnnn) deactivates the generation of emergency telegrams. • Mapping of the CANopen object I-1014 (see DS301 V4.02). "CAN on board" system bus			
Setting range			**Lenze setting**
0x00000000		0xFFFFFFFF	**0x00000081** (decimal: 129)
Value is bit-coded: (☑ = bit set)			**Information**
Bit 0 ☑	COB-ID bit 0		
Bit 1 ☐	COB-ID bit 1		
Bit 2 ☐	COB-ID bit 2		
Bit 3 ☐	COB-ID bit 3		
Bit 4 ☐	COB-ID bit 4		
Bit 5 ☐	COB-ID bit 5		
Bit 6 ☐	COB-ID bit 6		
Bit 7 ☑	COB-ID bit 7		
Bit 8 ☐	COB-ID bit 8		
Bit 9 ☐	COB-ID bit 9		
Bit 10 ☐	COB-ID bit 10		
Bit 11 ☐	Reserved		
Bit 12 ☐	Reserved		
Bit 13 ☐	Reserved		
Bit 14 ☐	Reserved		
Bit 15 ☐	Reserved		
Bit16	Reserved		
Bit 17 ☐	Reserved		
Bit 18 ☐	Reserved		
Bit 19 ☐	Reserved		
Bit 20 ☐	Reserved		
Bit 21 ☐	Reserved		
Bit 22 ☐	Reserved		
Bit 23 ☐	Reserved		
Bit 24 ☐	Reserved		
Bit 25 ☐	Reserved		
Bit 26 ☐	Reserved		
Bit 27 ☐	Reserved		
Bit 28 ☐	Reserved		
Bit 29 ☐	Reserved		
Bit 30 ☐	Reserved		
Bit 31 ☐	Emergency inactive/active		
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00392

Parameter | Name: **C00392 | CAN emergency delay time**			Data type: UNSIGNED_16 Index: 24183_d = $5E77_h$

Time which at least has to elapse between two subsequent emergency telegrams.
- Setting "0" deactivates the inhibit time check.
- The inhibit time is entered in 1/10 ms. The code automatically rounds the entries down to the preceding full millisecond.
- Mapping of the CANopen object I-1015 (see DS301 V4.02).

▶ "CAN on board" system bus

Setting range (min. value | unit | max. value)			**Lenze setting**
0		65535	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00393

Parameter | Name: **C00393 | CAN result - bus scan**			Data type: UNSIGNED_8 Index: 24182_d = $5E76_h$

Result of CAN bus scanning (see controller commands under C00002).
- Subcode number 1 ... 128 corresponds to CAN node address 1 ... 128.

▶ "CAN on board" system bus

Display range (min. value | unit | max. value)			
0		1	
Subcodes			**Information**
C00393/1			Result of CAN bus scanning for CAN node address 1 ... 128 • "1" means that a device with the corresponding node address has been found.
C00393/...			
C00393/128			
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00394

Parameter | Name: **C00394 | CAN predefined error field (DS301V402)**			Data type: UNSIGNED_32 Index: 24181_d = $5E75_h$
Display range (min. value | unit | max. value)			
0		4294967295	
Subcodes			**Information**
C00394/1			
C00394/2			
C00394/3			
C00394/4			
C00394/5			
C00394/6			
C00394/7			
C00394/8			
C00394/9			
C00394/10			
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00379

Parameter | Name: **C00395 | Service code**	Data type: UNSIGNED_32 Index: 24180_d = $5E74_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00396

Parameter | Name: **C00396 | Service code**	Data type: UNSIGNED_32 Index: 24179_d = $5E73_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00397

Parameter | Name: **C00397 | Service code**	Data type: UNSIGNED_32 Index: 24178_d = $5E72_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00398

Parameter | Name: **C00398 | Test mode motor control**	Data type: UNSIGNED_32 Index: 24177_d = $5E71_h$

⚠ **Danger!**
When the test mode is activated, the parameterisable error response "Quick stop by trouble" has no effect!
- If the test mode is active and a monitoring function responds with this error response, no quick stop is executed but the motor continues to rotate with the frequency set for the test mode!

Selection list (Lenze setting printed in bold)		**Information**
0	**Test mode deactivated**	
1	U rotation test mode	
2	I rotation test mode	
3	Current controller optimisation mode	After controller enable, the motor is supplied with current as long as the controller is enabled.
4	Current controller optimisation mode impulse	From software version V7.0 The motor is supplied with voltage for 50 ms after controller enable. Due to this time limit, the load of the machine is reduced. Afterwards, the controller is inhibited automatically.
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☑ No transfer		Scaling factor: 1

C00399

Parameter | Name: **C00399 | Settings for test mode**			Data type: INTEGER_32 Index: 24176_d = $5E70_h$
Setting range (min. value | unit | max. value)			
-1000.0	Hz/1°	1000.0	

Subcodes	**Lenze setting**	**Information**
C00399/1	0.0 Hz/1°	Frequency [Hz] for test mode
C00399/2	0.0 Hz/1°	Start angle [°] for test mode
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer		Scaling factor: 10

C00413

Parameter | Name: **C00413 | Hiperface: detected TypeCode**	Data type: UNSIGNED_32 Index: 24162_d = $5E62_h$

From software version V4.0
Type code read out of the connected Hiperface encoder

▸ Parameterisation of an unknown Hiperface encoder

Display range (min. value | unit | max. value)			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00414

Parameter | Name: **C00414 | Hiperface: TypeCode**			Data type: UNSIGNED_32 Index: 24161_d = $5E61_h$
From software version V4.0 Setting the type code for a Hiperface encoder unknown to the controller Parameterisation of an unknown Hiperface encoder			
Setting range (min. value | unit | max. value)			**Lenze setting**
0		255	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00415

Parameter | Name: **C00415 | Hiperface: number of rev.**			Data type: UNSIGNED_32 Index: 24160_d = $5E60_h$
From software version V4.0 Number of displayable revolutions for a multi-turn encoder Parameterisation of an unknown Hiperface encoder			
Setting range (min. value | unit | max. value)			**Lenze setting**
0		16384	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00416

Parameter | Name: **C00416 | Service code**	Data type: UNSIGNED_32 Index: 24159_d = $5E5F_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00417

Parameter | Name: **C00417 | Dynamic of resolver evaluation**			Data type: UNSIGNED_32 Index: 24158_d = $5E5E_h$
From software version V5.0 The resolver evaluation of the controller is adapted to the resolver types mounted in Lenze motors and offers a good compromise between the dynamic performance and interference suppression. If the resolver is used as speed feedback system, among other things the dynamic performance of the resolver evaluation determines the maximum speed controller gain by means of which stable operation is possible. In an EMC-compliant system (low interference), you can increase the dynamics of the resolver evaluation via this parameter without a quality loss in the speed signal. By increasing the setting, the evaluation gets more dynamic and thus the speed controller gainVp (C00070) also increases without leaving the stable operating range. The acceleration of the evaluation depends on the cable length, the resolver, and the quality of the electrical shielding. In many cases, a setting of 300 % is possible which can double the speed controller gain. The higher gain in the speed controller may reduce following errors.			
Setting range (min. value | unit | max. value)			**Lenze setting**
100	%	1000	**100 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00418

Parameter | Name: **C00418 | Activate resolver error compensation**		Data type: UNSIGNED_8 Index: 24157_d = $5E5D_h$
From software version V7.0 Resolver error compensation		
Selection list (Lenze setting printed in bold)		
0	**Deactivated**	
1	Activated	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00420

Parameter | Name: **C00420 | Encoder - number of increments**			Data type: UNSIGNED_16 Index: 24155_d = $5E5B_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
1		16384	**512**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00421

Parameter | Name: **C00421 | Encoder voltage**			Data type: UNSIGNED_16 Index: 24154_d = $5E5A_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
5.0	V	12.0	**5.0 V**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00422

Parameter | Name: **C00422 | Encoder type**		Data type: UNSIGNED_16 Index: 24153_d = $5E59_h$
Selection list (Lenze setting printed in bold)		**Information**
0	**Incremental encoder (TTL signal)**	
1	Sin/cos encoder	
2	Absolute value encoder (Hiperface)	
3	Absolute value encoder (EnDat)	
4	SSI encoder	From software version V5.0 ▶ Use of an SSI encoder at X8
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00423

Parameter | Name: **C00423 | SSI encoder: bit rate**			Data type: UNSIGNED_32 Index: 24152_d = $5E58_h$
From software version V5.0			▶ Use of an SSI encoder at X8
Setting range (min. value | unit | max. value)			**Lenze setting**
150	kbps	1000	**400 kbps**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00424

Parameter | Name: **C00424 | SSI encoder: Data word length**			Data type: UNSIGNED_32 Index: 24151_d = $5E57_h$
From software version V5.0			▶ Use of an SSI encoder at X8
Setting range (min. value | unit | max. value)			**Lenze setting**
1	Bit	31	**25 Bit**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00427

Parameter | Name: **C00427 | TTL encoder signal evaluation**		Data type: UNSIGNED_16 Index: 24148_d = $5E54_h$
Selection list (Lenze setting printed in bold)		
0	**4x evaluation (A, B)**	
1	A:Increments B:Sign	
2	Increments A:pos. B:neg.	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00435

Parameter | Name: **C00435 | SSI encoder: partword start position**			Data type: UNSIGNED_8 Index: 24140_d = $5E4C_h$
From software version V5.0			
			▸ Use of an SSI encoder at X8
Setting range (min. value | unit | max. value)			
0		30	
Subcodes	**Lenze setting**		**Information**
C00435/1	0		Ssi-enc.: Partword 1 start
C00435/2	0		Ssi-enc.: Partword 2 start
C00435/3	0		Ssi-enc.: Partword 3 start
C00435/4	0		Ssi-enc.: Partword 4 start
C00435/5	0		Ssi-enc.: Partword 5 start
C00435/6	0		Ssi-enc.: Partword 6 start
C00435/7	0		SSI enc.: Partword 7 start
C00435/8	0		Ssi-enc.: Partword 8 start
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00436

Parameter | Name: **C00436 | SSI encoder: partword length**			Data type: UNSIGNED_8 Index: 24139_d = $5E4B_h$
From software version V5.0			
			▸ Use of an SSI encoder at X8
Setting range (min. value | unit | max. value)			
0		31	
Subcodes	**Lenze setting**		**Information**
C00436/1	31		Ssi-enc.: Partword 1 length
C00436/2	0		Ssi-enc.: Partword 2 length
C00436/3	0		Ssi-enc.: Partword 3 length
C00436/4	0		Ssi-enc.: Partword 4 length
C00436/5	0		Ssi-enc.: Partword 5 length
C00436/6	0		Ssi-enc.: Partword 6 length
C00436/7	0		Ssi-enc.: Partword 7 length
C00436/8	0		SSI enc.: Partword 8 length
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00437

Parameter | Name: **C00437 | SSI encoder: partword data coding**		Data type: UNSIGNED_8 Index: 24138_d = $5E4A_h$
From software version V5.0		
		Use of an SSI encoder at X8
Selection list (Lenze setting printed in bold)		
0	binary coded	
1	gray coded	
Subcodes	**Lenze setting**	**Information**
C00437/1	0: binary coded	Ssi-enc.: Partword 1 coding
C00437/2	0: binary coded	Ssi-enc.: Partword 2 coding
C00437/3	0: binary coded	Ssi-enc.: Partword 3 coding
C00437/4	0: binary coded	Ssi-enc.: Partword 4 coding
C00437/5	0: binary coded	Ssi-enc.: Partword 5 coding
C00437/6	0: binary coded	Ssi-enc.: Partword 6 coding
C00437/7	0: binary coded	SSI enc.: Partword 7 coding
C00437/8	0: binary coded	Ssi-enc.: Partword 8 coding
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00443

Parameter | Name: **C00443 | Status: Digital inputs**			Data type: UNSIGNED_8 Index: 24132_d = $5E44_h$
Display range (min. value | unit | max. value)			
0		1	
Subcodes			**Information**
C00443/1			Digital input 1
C00443/2			Digital input 2
C00443/3			Digital input 3
C00443/4			Digital input 4
C00443/5			Digital input 5
C00443/6			Digital input 6
C00443/7			Digital input 7
C00443/8			Digital input 8
C00443/9			Controller inhibit (inversion of input X5/RFR)
C00443/10			Internal signal
C00443/11			Feedback input of holding brake
C00443/12			State bus input
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00444

Parameter | Name: **C00444 | Status: Digital outputs**	Data type: UNSIGNED_8 Index: 24131_d = $5E43_h$
Display range (min. value | unit | max. value)	
0 | | 1	
Subcodes	**Information**
C00444/1	Digital output 1
C00444/2	Digital output 2
C00444/3	Digital output 3
C00444/4	Digital output 4
C00444/5	Internal signal
C00444/6	Internal signal
C00444/7	Internal signal
C00444/8	Internal signal
C00444/9	User LED
C00444/10	Internal signal
C00444/11	Internal signal
C00444/12	Internal signal
C00444/13	Control output of holding brake
C00444/14	Internal signal
C00444/15	Internal signal
C00444/16	Internal signal
C00444/17	Internal signal
C00444/18	State bus output
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00464

Parameter | Name: **C00464 | Keypad: Mode** — Data type: UNSIGNED_16, Index: 24111_d = $5E2F_h$

From software version V5.0

Definition of the mode in which the keypad attached is to be activated.

- If both
  keypad versions (V1.0 and V2.0) are used, the Lenze setting is to be maintained.
- If only the new keypad V2.0 is used, the initialisation time of the keypad can be reduced by changing over to mode 2. Furthermore mode 2 supports greater future keypad files.

Note: If mode 2 is selected, the keypad V1.0 can no longer be operated on the controller!

Selection list (Lenze setting printed in bold)		**Information**
0	Mode 1	For keypad V1.0 and V2.0
1	**Mode 2**	Only for keypad V2.0
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00465

Parameter | Name: **C00465 | Keypad: Welcome screen time-out** — Data type: UNSIGNED_8, Index: 24110_d = $5E2E_h$

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	Scaling factor: 1

C00466

Parameter | Name: **C00466 | Keypad: Default parameters**			Data type: UNSIGNED_16 Index: 24109_d = $5E2D_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0		65535	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00467

Parameter | Name: **C00467 | Keypad: Default welcome screen**		Data type: UNSIGNED_8 Index: 24108_d = $5E2C_h$
Selection list (Lenze setting printed in bold)		
0	**Main menu**	
1	Parameter list	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00468

Parameter | Name: **C00468 | Service code**	Data type: UNSIGNED_8 Index: 24107_d = $5E2B_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00469

Parameter | Name: **C00469 | Keypad: Fct. STOP key**		Data type: UNSIGNED_8 Index: 24106_d = $5E2A_h$
Selection list (Lenze setting printed in bold)		
0	No function	
1	Inhibit controller	
2	**Activate quick stop**	
3	Stop application	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00490

Parameter | Name: **C00490 | Position encoder selection**		Data type: UNSIGNED_16 Index: 24085_d = $5E15_h$
Chapter "Controller configuration" provides you with more information on parameter setting.		
Selection list (Lenze setting printed in bold)		
0	Resolver on X7	
1	Encoder on X8	
4	Encoder signal on LS_Feedback	
10	**Motor encoder (C00495)**	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00494

Parameter | Name: **C00494 | Motor standstill time constant**			Data type: UNSIGNED_32 Index: 24081_d = $5E11_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	100000	**0 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00495

Parameter | Name: **C00495 | Motor encoder selection**		Data type: UNSIGNED_16 Index: 24080_d = $5E10_h$
Selection list (Lenze setting printed in bold)		
0	**Resolver on X7**	
1	Encoder on X8	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00497

Parameter | Name: **C00497 | Speed act. val. time const.**			Data type: UNSIGNED_32 Index: 24078_d = $5E0E_h$

Time constant for actual speed filter

- In order to maximise the dynamics of the speed control loop, the actual speed filter should be operated with a time constant as low as possible. The lower the time constant the higher the gain of the speed controller. Since actual value filters have the task to dampen measuring errors or interference components, it must be found a compromise between filter task and the resulting delay.
- If a Lenze motor is selected from the motor catalogue, a time constant is automatically preset here which serves to operate the motor even with a faulty detection (e.g. in case of a bad shield connection).

▸ Servo control (SC): Optimise speed controller
▸ Sensorless vector control (SLVC): Optimise speed controller

Setting range (min. value | unit | max. value)			**Lenze setting**
0.0	ms	50.0	**2.0 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00512

Parameter | Name: **C00512 | Service code**	Data type: UNSIGNED_32 Index: 24063_d = $5DFF_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00513

Parameter | Name: **C00513 | Service code**	Data type: VISIBLE_STRING Index: 24062_d = $5DFE_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00514

Parameter | Name: **C00514 | Service code**	Data type: UNSIGNED_32 Index: 24061_d = $5DFD_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00515

Parameter | Name: **C00515 | Service code**	Data type: UNSIGNED_32 Index: 24060_d = $5DFC_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00516

Parameter | Name: **C00516 | Service code**	Data type: UNSIGNED_32 Index: 24059_d = $5DFB_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00569

Parameter | Name: **C00569 | Resp. brake trans. Ixt > C00570**	Data type: UNSIGNED_32 Index: 24006_d = $5DC6_h$

From software version V1.5

Response if adjustable warning threshold (C00570) of brake chopper monitoring is reached.

▸ Braking operation

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
5	**Warning**
4	Warning locked
3	Quick stop by trouble
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00570

Parameter | Name: **C00570 | Warning thres. brake transistor**	Data type: UNSIGNED_32 Index: 24005_d = $5DC5_h$

From software version V1.5

Warning threshold for brake chopper monitoring

- The response to reaching the threshold can be selected in C00569.

▸ Braking operation

Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	100	**90 %**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00571

Parameter | Name: **C00571 | Resp. brake res. i²xt > C00572**	Data type: UNSIGNED_32 Index: 24004_d = $5DC4_h$

From software version V1.5

Response if adjustable warning threshold (C00572) of brake resistor monitoring is reached.

▸ Braking operation

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
5	**Warning**
4	Warning locked
3	Quick stop by trouble
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00572

Parameter | Name: **C00572 | Warning thres. brake resistor**	Data type: UNSIGNED_32 Index: 24003_d = $5DC3_h$

From software version V1.5

Warning threshold for brake resistor monitoring

- The response to reaching the threshold can be selected in C00571.

▸ Braking operation

Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	100	**90 %**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00573

Parameter | Name: **C00573 | Resp. to overload brake trans.**			Data type: UNSIGNED_32 Index: 24002_d = $5DC2_h$
Response to activation of brake chopper monitoring			

▶ Braking operation

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
5	Warning
4	Warning locked
3	Quick stop by trouble
0	**No response**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00574

Parameter | Name: **C00574 | Resp. to overtemp. brake resist.**			Data type: UNSIGNED_32 Index: 24001_d = $5DC1_h$
Response to activation of brake resistor monitoring			

▶ Braking operation

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
5	Warning
4	Warning locked
3	Quick stop by trouble
0	**No response**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00576

Parameter | Name: **C00576 | Speed monitoring tolerance**			Data type: UNSIGNED_32 Index: 23999_d = $5DBF_h$
Monitoring window for speed control error in [%] of nmax			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	100	**100 %**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00577

Parameter | Name: **C00577 | Field weakening controller gain**			Data type: UNSIGNED_32 Index: 23998_d = $5DBE_h$
At "0" the P component is deactivated, a pure I-controller is used.			
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	Vs/V	2147483.647	**0.000 Vs/V**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1000

C00578

Parameter | Name: **C00578 | Field weak. contr. reset time**			Data type: UNSIGNED_32 Index: 23997_d = $5DBD_h$
At "240000.0 ms" the I component of the field weakening controller is deactivated.			
Setting range (min. value | unit | max. value)			**Lenze setting**
1.0	ms	240000.0	**2000.0 ms**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 10

C00579

Parameter | Name: **C00579 | Resp. to speed monitoring**	Data type: UNSIGNED_32 Index: 23996_d = $5DBC_h$

Response to activation of speed monitoring

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
5	Warning
4	Warning locked
3	Quick stop by trouble
0	**No response**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00580

Parameter | Name: **C00580 | Resp. to encoder open circuit**	Data type: UNSIGNED_32 Index: 23995_d = $5DBB_h$

Response to open circuit in encoder
⚠ **Danger!**
If the encoder is used as motor encoder:
In the event of an error, a safe operation of the motor cannot be ensured anymore!
For safety reasons, the "Fault" response should always be set for this case!

Selection list (Lenze setting printed in bold)	
1	**Fault**
2	Trouble
5	Warning
4	Warning locked
3	Quick stop by trouble
13	Quick Stop open loop by trouble
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00581

Parameter | Name: **C00581 | Resp. to external fault**	Data type: UNSIGNED_32 Index: 23994_d = $5DBA_h$

Response to an external error

▶ Drive interface

Selection list (Lenze setting printed in bold)	
1	**Fault**
2	Trouble
3	Quick stop by trouble
4	Warning locked
5	Warning
6	Information
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00582

Parameter | Name: C00582 | Resp. to heatsink temp. > C00122		Data type: UNSIGNED_32 Index: 23993$_d$ = 5DB9$_h$
Response if heatsink temperature > variable limit temperature (C00122).		
Selection list **(Lenze setting printed in bold)**		
1	Fault	
5	**Warning**	
0	No response	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00583

Parameter | Name: C00583 | Resp. to motor KTY overtemp.		Data type: UNSIGNED_32 Index: 23992$_d$ = 5DB8$_h$
Response if motor temperature > fixed limit temperature.		
		▶ Motor temperature monitoring
Selection list **(Lenze setting printed in bold)**		
1	**Fault**	
5	Warning	
0	No response	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00584

Parameter | Name: C00584 | Resp. to motor temp. > C00121		Data type: UNSIGNED_32 Index: 23991$_d$ = 5DB7$_h$
Response if motor temperature > variable limit temperature (C00121).		
		▶ Motor temperature monitoring
Selection list **(Lenze setting printed in bold)**		
1	Fault	
5	**Warning**	
0	No response	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00585

Parameter | Name: C00585 | Resp. to motor PTC overtemp.		Data type: UNSIGNED_32 Index: 23990$_d$ = 5DB6$_h$
Response if motor temperature across PTC input T1/T2 too high.		
Selection list **(Lenze setting printed in bold)**		
1	Fault	
5	Warning	
0	**No response**	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00586

Parameter | Name: **C00586 | Resp. to resolver open circuit**	Data type: UNSIGNED_32 Index: 23989_d = $5DB5_h$

Response to open circuit in resolver
⚠ **Danger!**
If the resolver is used as motor encoder:
In the event of an error, a safe operation of the motor cannot be ensured anymore!
For safety reasons, the "Fault" response should always be set for this case!

Selection list (Lenze setting printed in bold)	
1	**Fault**
2	Trouble
5	Warning
4	Warning locked
3	Quick stop by trouble
13	Quick Stop open loop by trouble
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00587

Parameter | Name: **C00587 | Fan control status**	Data type: BITFIELD_8 Index: 23988_d = $5DB4_h$

Display area		
0x00		0xFF

Value is bit-coded:	
Bit 0	Heatsink fan ON
Bit 1	Integral fan ON
Bit 2	Heatsink fan status 1
Bit 3	Heatsink fan status 2
Bit 4	Integral fan status
Bit 5	Reserved
Bit 6	Reserved
Bit 7	Reserved

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00588

Parameter | Name: **C00588 | Resp. to failure t. sensor drive**	Data type: UNSIGNED_32 Index: 23987_d = $5DB3_h$

Response to error/failure of temperature sensor for heatsink temperature/temperature inside the controller

Selection list (Lenze setting printed in bold)	
1	**Fault**
5	Warning
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00589

Parameter | Name:
C00589 | Resp. to CPU temp. > C00126

Data type: UNSIGNED_32
Index: 23986_d = $5DB2_h$

Response if CPU temperature on the control card > variable limit temperature (C00126).

Selection list (Lenze setting printed in bold)	
1	Fault
5	Warning
0	**No response**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00591

Parameter | Name:
C00591 | Resp. to CAN-RPDOx error

Data type: UNSIGNED_8
Index: 23984_d = $5DB0_h$

Response if the corresponding CAN RPDO has not been received in the configured time or with the configured sync.

▶ "CAN on board" system bus

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
3	Quick stop by trouble
4	Warning locked
5	Warning
6	Information
0	No response

Subcodes	Lenze setting	Information
C00591/1	0: No response	Response to non-received RPDO1 ... RPDO4
C00591/...		
C00591/4		

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00594

Parameter | Name:
C00594 | Resp. temp. sensor motor X7/X8

Data type: UNSIGNED_32
Index: 23981_d = $5DAD_h$

Response to motor temperature sensor error.

- The response to a too high motor temperature via PTC input T1/T2 can be selected in C00585.

▶ Motor temperature monitoring

Selection list (Lenze setting printed in bold)	
1	**Fault**
5	Warning
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00595

Parameter | Name: **C00595 | Resp. to CAN bus OFF**	Data type: UNSIGNED_8 Index: 23980_d = $5DAC_h$

Response if CAN node switches to the bus off state.

▶ "CAN on board" system bus

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
3	Quick stop by trouble
4	Warning locked
5	Warning
6	Information
0	**No response**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00596

Parameter | Name: **C00596 | Threshold max. speed reached**	Data type: UNSIGNED_32 Index: 23979_d = $5DAB_h$

Threshold for speed monitoring
- The response to reaching the threshold can be selected in C00607.

Setting range (min. value | unit | max. value)			**Lenze setting**
50	rpm	50000	**6500 rpm**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00597

Parameter | Name: **C00597 | Resp. to motor phase failure**	Data type: UNSIGNED_32 Index: 23978_d = $5DAA_h$

Response to activation of motor phase failure monitoring

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
5	Warning
4	Warning locked
3	Quick stop by trouble
0	**No response**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00598

Parameter | Name: **C00598 | Resp. to open circuit AIN1**	Data type: UNSIGNED_32 Index: 23977_d = $5DA9_h$

Response if with master current at AIN1 and "LifeZero" mode (±4...±20mA) the current is in the non-permitted range (-4 ... +4 mA).

Selection list (Lenze setting printed in bold)	
1	**Fault**
2	Trouble
3	Quick stop by trouble
4	Warning locked
5	Warning
6	Information
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00599

Parameter | Name: **C00599 | Motor phase failure threshold**			Data type: INTEGER_32 Index: $23976_d = 5DA8_h$
Current value for activating the Motor phase failure monitoring • In [%] relating to the maximum device current (display in C00789). • The response to be triggered by the monitoring can be selected in C00597.			
Setting range (min. value | unit | max. value)			**Lenze setting**
1.0	%	10.0	**5.0 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00600

Parameter | Name: **C00600 | Resp. to DC bus overvoltage**		Data type: UNSIGNED_32 Index: $23975_d = 5DA7_h$
Response to DC bus overvoltage ▸ Braking operation		
Selection list (Lenze setting printed in bold)		
1	Fault	
2	**Trouble**	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00601

Parameter | Name: **C00601 | Resp. to encoder comm. error**		Data type: UNSIGNED_32 Index: $23974_d = 5DA6_h$
Response to activation of encoder monitoring • **For the use of the encoder as motor encoder:** If an error occurs, the safe operation of the motor is no longer ensured, therefore for safety reasons the "Fault" response should always be set!		
Selection list (Lenze setting printed in bold)		
1	**Fault**	
2	Trouble	
5	Warning	
4	Warning locked	
3	Quick stop by trouble	
13	Quick Stop open loop by trouble	
0	No response	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00604

Parameter | Name: **C00604 | Resp. to device overload > C00123**		Data type: UNSIGNED_32 Index: $23971_d = 5DA3_h$
Response if adjustable "I x t" warning threshold (C00123) is reached.		
Selection list (Lenze setting printed in bold)		
1	Fault	
5	**Warning**	
0	No response	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00606

Parameter | Name: **C00606 | Resp. to motor overload > C00127**	Data type: UNSIGNED_32 Index: 23969_d = $5DA1_h$
Response if adjustable "I^2 x t" warning threshold (C00127) is reached.	

Selection list (Lenze setting printed in bold)	
1	Fault
5	**Warning**
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00607

Parameter | Name: **C00607 | Resp. to max. speed reached**	Data type: UNSIGNED_32 Index: 23968_d = $5DA0_h$
Response if adjustable speed threshold (C00596) is reached.	

Selection list (Lenze setting printed in bold)	
1	**Fault**
2	Trouble
5	Warning
4	Warning locked
3	Quick stop by trouble
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00610

Parameter | Name: **C00610 | Resp. to failure heatsink fan**	Data type: UNSIGNED_32 Index: 23965_d = $5D9D_h$
Response if fan speed of heatsink fan is too low.	

Selection list (Lenze setting printed in bold)	
1	**Fault**
5	Warning
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00611

Parameter | Name: **C00611 | Resp. to failure integral fan**	Data type: UNSIGNED_32 Index: 23964_d = $5D9C_h$
Response if fan speed of internal fan is too low.	

Selection list (Lenze setting printed in bold)	
1	**Fault**
5	Warning
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00612

Parameter | Name: **C00612 | Resp. to CAN node guarding error**		Data type: UNSIGNED_8 Index: 23963_d = $5D9B_h$
Response of master if the corresponding node guarding response is not received. "CAN on board" system bus: node guarding protocol		
Selection list (Lenze setting printed in bold)		
1	Fault	
2	Trouble	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
6	Information	
0	No response	
Subcodes	**Lenze setting**	**Information**
C00612/1	0: No response	Response to non-received telegram for monitoring entry 1 ... 32
C00612/...		
C00612/32		
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00613

Parameter | Name: **C00613 | Resp. to CAN heartbeat error**		Data type: UNSIGNED_8 Index: 23962_d = $5D9A_h$
Response if the corresponding heartbeat telegram is not received. "CAN on board" system bus: heartbeat protocol		
Selection list (Lenze setting printed in bold)		
1	Fault	
2	Trouble	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
6	Information	
0	No response	
Subcodes	**Lenze setting**	**Information**
C00613/1	0: No response	Response to non-received telegram for monitoring entry 1 ... 32
C00613/...		
C00613/32		
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00614

Parameter | Name: **C00614 | Resp. to CAN life guarding error**	Data type: UNSIGNED_8 Index: 23961_d = $5D99_h$

Response of slave if node guarding request is not received.

"CAN on board" system bus: node guarding protocol

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
3	Quick stop by trouble
4	Warning locked
5	Warning
6	Information
0	**No response**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00615

Parameter | Name: **C00615 | Resp. to imp. device config.**	Data type: UNSIGNED_32 Index: 23960_d = $5D98_h$

Selection list	
1	Fault
3	Quick stop by trouble
4	Warning locked
6	Information
0	No response

Subcodes	Lenze setting	Information
C00615/1	0: No response	Reserved
C00615/2	1: Fault	Resp. to imp. module in MXI1
C00615/3	1: Fault	Resp. to imp. module in MXI2
C00615/4	0: No response	Reserved
C00615/5	0: No response	Reserved

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00618

Parameter | Name: **C00618 | No. of CRC cycles**	Data type: UNSIGNED_32 Index: 23957_d = $5D95_h$

Display range (min. value | unit | max. value)		
0		4294967295

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00619

Parameter | Name: **C00619 | Resp. to motor current > C00620**	Data type: UNSIGNED_32 Index: 23956_d = $5D94_h$

Response if the ultimate motor current I_ult parameterised in C00620 is reached.

Selection list (Lenze setting printed in bold)	
1	**Fault**
2	Trouble
5	Warning
4	Warning locked
3	Quick stop by trouble
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00620

Parameter | Name: **C00620 | Ultimate motor current I_ult**			Data type: UNSIGNED_32 Index: 23955_d = $5D93_h$

Limit value to protect the motor from destruction or influence of the rated data.
- This limit value must not be travelled cyclically during the drive process.
- The maximum current parameterisable in C00022 should have a sufficient distance from this limit value.
- If the set limit value is exceeded, the error response parameterised in C00619 is carried out for motor protection.

Setting range (min. value | unit | max. value)			**Lenze setting**
0.0	A	3000.0	**3000.0 A**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00621

Parameter | Name: **C00621 | Resp. to encoder pulse deviation**	Data type: UNSIGNED_32 Index: 23954_d = $5D92_h$

From software version V1.5
Response to be triggered by angular drift monitoring

▶ Angular drift monitoring

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
5	Warning
4	Warning locked
3	Quick stop by trouble
13	Quick Stop open loop by trouble
0	**No response**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00625

Parameter | Name: **C00625 | CAN behaviour in case of fault**	Data type: UNSIGNED_8 Index: 23950_d = $5D8E_h$

Mapping of the CANopen object I-1029 (see DS301 V4.02).

▶ "CAN on board" system bus

Selection list (Lenze setting printed in bold)	
0	**Pre-operational state**
1	No state change
2	Stopped state

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00635

Parameter | Name: **C00635 | Resp to new firmw. standard dev.**	Data type: UNSIGNED_32 Index: 23940_d = $5D84_h$

Selection list (Lenze setting printed in bold)	
1	**Fault**
6	Information
5	Warning
4	Warning locked
3	Quick stop by trouble
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00636

Parameter | Name: **C00636 | Resp. to new module in MXI1**	Data type: UNSIGNED_32 Index: 23939_d = $5D83_h$

Selection list (Lenze setting printed in bold)

1	Fault
6	Information
5	Warning
4	**Warning locked**
3	Quick stop by trouble
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00637

Parameter | Name: **C00637 | Resp. to new module in MXI2**	Data type: UNSIGNED_32 Index: 23938_d = $5D82_h$

Selection list (Lenze setting printed in bold)

1	Fault
6	Information
5	Warning
4	**Warning locked**
3	Quick stop by trouble
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00640

Parameter | Name: **C00640 | Resp. to pole pos. id. monit.**	Data type: UNSIGNED_32 Index: 23935_d = $5D7F_h$

From software version V4.0
Error response for abort of the pole position identification

▶ Adjustment of the pole position identification

Selection list (Lenze setting printed in bold)

1	**Fault**
4	Warning locked
0	No response

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00641

Parameter | Name: **C00641 | PLI 360° current amplitude**	Data type: UNSIGNED_32 Index: 23934_d = $5D7E_h$

From software version V4.0
Percentage adjustment of the current amplitude for the pole position identification
Stop!
If there is no temperature monitoring in the motor and/or the I2xt motor monitoring is not parameterised correctly, the motor may be permanently damaged when the current amplitude is set too high (e.g. to the maximum value)!

▶ Adjustment of the pole position identification

Setting range (min. value | unit | max. value)			**Lenze setting**
1	%	1000	**100 %**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00642

Parameter | Name: **C00642 | PolePosId 360° ramp time**			Data type: UNSIGNED_32 Index: 23933_d = $5D7D_h$

From software version V4.0
Percentage adjustment of the ramp time for the pole position identification

▶ Adjustment of the pole position identification

Setting range (min. value | unit | max. value)			**Lenze setting**
5	%	1000	**100 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00643

Parameter | Name: **C00643 | PLI 360° traversing direction**	Data type: UNSIGNED_32 Index: 23932_d = $5D7C_h$

From software version V4.0
Selection of the traversing direction for the pole position identification

▶ Adjustment of the pole position identification

Selection list (Lenze setting printed in bold)	
0	**Right rotating field**
1	Left rotating field
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C00644

Parameter | Name: **C00644 | PolePosId 360° fault tol.**	Data type: INTEGER_32 Index: 23931_d = $5D7B_h$

From software version V4.0
Fault tolerance for the plausibility check of the pole position identification

▶ Adjustment of the pole position identification

Setting range (min. value | unit | max. value)			**Lenze setting**
-6.0	°	30.0	**0.0 °**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00645

Parameter | Name: **C00645 | PLI 360° absolute current amplitude**	Data type: UNSIGNED_32 Index: 23930_d = $5D7A_h$

From software version V7.0

▶ Adjustment of the pole position identification

Display range (min. value | unit | max. value)			
0.00	A	1000.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00646

Parameter | Name: **C00646 | PLI min.mov. curr. amplitude**	Data type: UNSIGNED_32 Index: 23929_d = $5D79_h$

From software version V4.0
Percentage adjustment of the current amplitude for the pole position identification
Stop!
If there is no temperature monitoring in the motor and/or the I2xt motor monitoring is not parameterised correctly, the motor may be permanently damaged when the current amplitude is set too high (e.g. to the maximum value)!

▶ Adjustment of the pole position identification

Setting range (min. value | unit | max. value)			**Lenze setting**
1	%	1000	**100 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00647

Parameter | Name: **C00647 | PolePosId min.mov. cur.rise rate**			Data type: UNSIGNED_32 Index: 23928_d = $5D78_h$

From software version V4.0

Percentage adjustment of the rate of current rise for the pole position identification

▶ Adjustment of the pole position identification

Setting range (min. value | unit | max. value)			**Lenze setting**
5	%	1000	**100 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00648

Parameter | Name: **C00648 | PolePosId min.mov. gain**			Data type: UNSIGNED_32 Index: 23927_d = $5D77_h$

From software version V4.0

P component of the PI controller for the pole position identification

▶ Adjustment of the pole position identification

Setting range (min. value | unit | max. value)			**Lenze setting**
0.00		10.00	**0.00**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00649

Parameter | Name: **C00649 | PLI min. mov. reset time**			Data type: UNSIGNED_32 Index: 23926_d = $5D76_h$

From software version V4.0

I component of the PI controller for the pole position identification

▶ Adjustment of the pole position identification

Setting range (min. value | unit | max. value)			**Lenze setting**
0.01	ms	6000.00	**62.50 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00650

Parameter | Name: **C00650 | PolePosId min.mov. max.perm.mov.**			Data type: INTEGER_32 Index: 23925_d = $5D75_h$

From software version V4.0

Max. movement permitted during the pole position identification

▶ Adjustment of the pole position identification

Setting range (min. value | unit | max. value)			**Lenze setting**
1	°	90	**20 °**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00651

Parameter | Name: **C00651 | PLI min. motion absolute cur. amp.**			Data type: UNSIGNED_32 Index: 23924_d = $5D74_h$

From software version V7.0

▶ Adjustment of the pole position identification

Display range (min. value | unit | max. value)			
0.00	A	1000.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00691

Parameter | Name: **C00691 | Total speed setpoint**			Data type: INTEGER_32 Index: 23884_d = $5D4C_h$
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00692

Parameter | Name: **C00692 | Speed setpoint [%]**			Data type: INTEGER_32 Index: 23883_d = $5D4B_h$
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00693

Parameter | Name: **C00693 | Actual speed [%]**			Data type: INTEGER_32 Index: 23882_d = $5D4A_h$
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00694

Parameter | Name: **C00694 | Speed controller output**			Data type: INTEGER_32 Index: 23881_d = $5D49_h$
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00695

Parameter | Name: **C00695 | Total torque setpoint**			Data type: INTEGER_32 Index: 23880_d = $5D48_h$
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00696

Parameter | Name: **C00696 | Torque setpoint [%]**			Data type: INTEGER_32 Index: 23879_d = $5D47_h$
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00697

Parameter | Name: **C00697 | Filtered torque setpoint**			Data type: INTEGER_32 Index: 23878_d = $5D46_h$
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00698

Parameter | Name: **C00698 | Actual torque [%]**			Data type: INTEGER_32 Index: 23877_d = $5D45_h$
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00730

Parameter | Name: **C00730 | GDO general parameters**	Data type: INTEGER_32 Index: 23845_d = $5D25_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00731

Parameter | Name: **C00731 | GDO channel 1/trigger 1**	Data type: INTEGER_32 Index: 23844_d = $5D24_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00732

Parameter | Name: **C00732 | GDO channel 2/trigger 2**	Data type: INTEGER_32 Index: 23843_d = $5D23_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00733

Parameter | Name: **C00733 | GDO channel 3**	Data type: INTEGER_32 Index: 23842_d = $5D22_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00734

Parameter | Name: **C00734 | GDO channel 4**	Data type: INTEGER_32 Index: 23841_d = $5D21_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00735

Parameter | Name: **C00735 | GDO channel 5**	Data type: INTEGER_32 Index: 23840_d = $5D20_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00736

Parameter | Name: **C00736 | GDO channel 6**	Data type: INTEGER_32 Index: 23839_d = $5D1F_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00737

Parameter | Name: **C00737 | GDO channel 7**	Data type: INTEGER_32 Index: 23838_d = $5D1E_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00738

Parameter | Name: **C00738 | GDO channel 8**	Data type: INTEGER_32 Index: 23837_d = $5D1D_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00739

Parameter | Name: **C00739 | GDO status information**	Data type: INTEGER_32 Index: 23836_d = $5D1C_h$
This code is used internally by the controller and must not be overwritten by the user!	

C00770

Parameter | Name: **C00770 | MCTRL_dnMotorPosAct**			Data type: UNSIGNED_32 Index: 23805_d = $5CFD_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0	Incr.	4294967295	
Subcodes			**Information**
C00770/1			Low word
C00770/2			High word
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00771

Parameter | Name: **C00771 | MCTRL_dnLoadPosAct**			Data type: UNSIGNED_32 Index: 23804_d = $5CFC_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0	Incr.	4294967295	
Subcodes			**Information**
C00771/1			Low word
C00771/2			High word
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00772

Parameter | Name: **C00772 | MCTRL_dnMotorSpeedAct**			Data type: INTEGER_32 Index: 23803_d = $5CFB_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-480000	rpm	480000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00773

Parameter | Name: **C00773 | MCTRL_dnLoadSpeedAct**			Data type: INTEGER_32 Index: 23802_d = $5CFA_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-480000	rpm	480000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00774

Parameter | Name: **C00774 | MCTRL_dnTorqueAct**			Data type: INTEGER_32 Index: 23801_d = $5CF9_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-21474836.47	Nm	21474836.47	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00775

Parameter | Name: **C00775 | MCTRL_dnOutputSpeedCtrl**			Data type: INTEGER_32 Index: 23800_d = $5CF8_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-21474836.47	Nm	21474836.47	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00776

Parameter | Name: **C00776 | MCTRL_dnInputJerkCtrl**			Data type: INTEGER_32 Index: 23799_d = $5CF7_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-21474836.47	Nm	21474836.47	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00777

Parameter | Name: **C00777 | MCTRL_dnInputTorqueCtrl**			Data type: INTEGER_32 Index: 23798_d = $5CF6_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-21474836.47	Nm	21474836.47	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00778

Parameter | Name: **C00778 | MCTRL_dnFluxAct**			Data type: INTEGER_32 Index: 23797_d = $5CF5_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00779

Parameter | Name: **C00779 | MCTRL_dnDCBusVoltage**			Data type: INTEGER_32 Index: 23796_d = $5CF4_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0	V	1000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00780

Parameter | Name: **C00780 | MCTRL_dnImotAct**			Data type: INTEGER_32 Index: 23795_d = $5CF3_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-500.00	A	500.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00781

Parameter | Name: **C00781 | MCTRL_dwMaxMotorSpeed**			Data type: UNSIGNED_32 Index: 23794_d = $5CF2_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0	rpm	480000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00782

Parameter | Name: **C00782 | MCTRL_dwMaxMotorTorque**			Data type: UNSIGNED_32 Index: 23793_d = $5CF1_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0.000	Nm	2147483.647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C00783

Parameter | Name: **C00783 | MCTRL_dwMotorVoltageAct**			Data type: UNSIGNED_32 Index: 23792_d = $5CF0_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0	V	2000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00784

Parameter | Name: **C00784 | MCTRL_dnMotorFreqAct**			Data type: INTEGER_32 Index: 23791_d = $5CEF_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-800.0	Hz	800.0	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00786

Parameter | Name: **C00786 | MCTRL_dnIxtLoad**			Data type: INTEGER_32 Index: 23789_d = $5CED_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00787

Parameter | Name: **C00787 | MCTRL_dnFlyingSpeedAct**			Data type: INTEGER_32 Index: 23788_d = $5CEC_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-480000	rpm	480000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00788

Parameter | Name: **C00788 | MCTRL_dwMaxEffMotorTorque**			Data type: INTEGER_32 Index: 23787_d = $5CEB_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0.000	Nm	2147483.647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C00789

Parameter | Name: **C00789 | MCTRL_dwMaxDeviceCurrent**			Data type: INTEGER_32 Index: 23786_d = $5CEA_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0.00	A	21474836.47	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00790

Parameter | Name: **C00790 | MCTRL_dnI2xtLoad**			Data type: INTEGER_32 Index: 23785_d = $5CE9_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00791

Parameter | Name: **C00791 | MCTRL_dnDeltaMotorPos_p**			Data type: INTEGER_32 Index: 23784_d = $5CE8_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-2147483647	Incr.	2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00792

Parameter | Name: **C00792 | MCTRL_dnOutputPosCtrlMotor_s**			Data type: INTEGER_32 Index: 23783_d = $5CE7_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-200	%	200	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00800

Parameter | Name: **C00800 | MCTRL_dnPosSet**			Data type: UNSIGNED_32 Index: 23775_d = $5CDF_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0	Incr.	4294967295	
Subcodes			**Information**
C00800/1			Low word
C00800/2			High word
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00802

Parameter | Name: **C00802 | MCTRL_dnSpeedAdd**			Data type: INTEGER_32 Index: 23773_d = $5CDD_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-480000	rpm	480000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00803

Parameter | Name: **C00803 | MCTRL_dnTorqueAdd**			Data type: INTEGER_32 Index: 23772_d = $5CDC_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-2147483.647	Nm	2147483.647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C00804

Parameter | Name: **C00804 | MCTRL_dnAccelerationAdd**			Data type: INTEGER_32 Index: 23771_d = $5CDB_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-2147483.647	1/s²	2147483.647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C00805

Parameter | Name: **C00805 | MCTRL_dnSpeedLowLimit**			Data type: INTEGER_32 Index: 23770_d = $5CDA_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-480000	rpm	480000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00806

Parameter | Name: **C00806 | MCTRL_dnTorqueLowLimit**			Data type: INTEGER_32 Index: 23769_d = $5CD9_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-21474836.47	Nm	21474836.47	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00807

Parameter | Name: **C00807 | MCTRL_dnTorqueHighLimit**			Data type: INTEGER_32 Index: 23768_d = $5CD8_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-21474836.47	Nm	21474836.47	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00808

Parameter | Name: **C00808 | MCTRL_dnPosCtrlOutLimit**			Data type: INTEGER_32 Index: 23767_d = $5CD7_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-480000	rpm	480000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00809

Parameter | Name: **C00809 | MCTRL_dnTorqueCtrlAdapt**			Data type: INTEGER_32 Index: 23766_d = $5CD6_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00810

Parameter | Name: **C00810 | MCTRL_dnSpeedCtrlAdapt**			Data type: INTEGER_32 Index: 23765_d = $5CD5_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00811

Parameter | Name: **C00811 | MCTRL_dnPosCtrlAdapt**			Data type: INTEGER_32 Index: 23764_d = $5CD4_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00812

Parameter | Name: **C00812 | MCTRL_dnMotorPosRefValue**			Data type: UNSIGNED_32 Index: 23763_d = $5CD3_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0	Incr.	4294967295	
Subcodes			**Information**
C00812/1			Low word
C00812/2			High word
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00813

Parameter | Name: **C00813 | MCTRL_dnLoadPosRefValue**			Data type: UNSIGNED_32 Index: 23762_d = $5CD2_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
0	Incr.	4294967295	
Subcodes			**Information**
C00813/1			Low word
C00813/2			High word
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00814

Parameter | Name: **C00814 | MCTRL_dnBoost**			Data type: INTEGER_32 Index: 23761_d = $5CD1_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-1000	V	1000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00815

Parameter | Name: **C00815 | MCTRL_dnSpeedCtrlIntegrator**			Data type: INTEGER_32 Index: 23760_d = $5CD0_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-2147483.647	Nm	2147483.647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C00816

Parameter | Name: **C00816 | MCTRL_dnFieldWeak**			Data type: INTEGER_32 Index: 23759_d = $5CCF_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00817

Parameter | Name: **C00817 | MCTRL_dnSpeedSet_s**			Data type: INTEGER_32 Index: 23758_d = $5CCE_h$
Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-480000	rpm	480000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00818

Parameter | Name: **C00818 | MCTRL_dnMvorAdapt**			Data type: INTEGER_32 Index: 23757_d = $5CCD_h$
From software version V1.5 Internal motor control (MCTRL) signal			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00854

Parameter | Name: **C00854 | ID status**			Data type: UNSIGNED_32 Index: 23721_d = $5CA9_h$
From software version V3.0			
Display range (min. value | unit | max. value)			
0		100	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00878

Parameter | Name: **C00878 | Status DCTRL control input**			Data type: UNSIGNED_8 Index: 23697_d = $5C91_h$
Display range (min. value | unit | max. value)			
0		1	
Subcodes			**Information**
C00878/1			Status of control inputs
C00878/...			
C00878/5			
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00909

Parameter | Name: **C00909 | Speed limitation**			Data type: INTEGER_16 Index: 23666_d = $5C72_h$

Speed limitation for speed setpoint
- For the upper speed limit value only positive values are permissible (0.0 % ... 175.0 %).
- For the lower speed limit value only negative values are permissible (-175.0 % ... 0.0 %).

Setting range (min. value | unit | max. value)			
-175.0	%	175.0	
Subcodes	**Lenze setting**		**Information**
C00909/1	175.0 %		Upper speed limit value
C00909/2	-175.0 %		Lower speed limit value

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 10

C00950

Parameter | Name: **C00950 | VFC: V/f characteristic shape**	Data type: UNSIGNED_32 Index: 23625_d = $5C49_h$

From software version V3.0

▶ V/f control

Selection list (Lenze setting printed in bold)	
0	**Linear (V/f)**
1	Quadratic (V/f²)
2	Curve

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00951

Parameter | Name: **C00951 | VFC: V/f base frequency**	Data type: INTEGER_32 Index: 23624_d = $5C48_h$

From software version V3.0

▶ V/f control

Setting range (min. value | unit | max. value)			**Lenze setting**
1	Hz	5000	**50 Hz**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00952

Parameter | Name: **C00952 | VFC: Frequency interpol. point n**			Data type: INTEGER_32 Index: 23623_d = $5C47_h$
From software version V3.0			▸ V/f control
Setting range (min. value | unit | max. value)			
-5000	Hz	5000	
Subcodes	**Lenze setting**		**Information**
C00952/1	-50 Hz		
C00952/2	-40 Hz		
C00952/3	-30 Hz		
C00952/4	-20 Hz		
C00952/5	-10 Hz		
C00952/6	0 Hz		
C00952/7	10 Hz		
C00952/8	20 Hz		
C00952/9	30 Hz		
C00952/10	40 Hz		
C00952/11	50 Hz		
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00953

Parameter | Name: **C00953 | VFC: Voltage interpol. point n**			Data type: INTEGER_32 Index: 23622_d = $5C46_h$
From software version V3.0			▸ V/f control
Setting range (min. value | unit | max. value)			
0.00	V	1000.00	
Subcodes	**Lenze setting**		**Information**
C00953/1	400.00 V		
C00953/2	320.00 V		
C00953/3	240.00 V		
C00953/4	160.00 V		
C00953/5	80.00 V		
C00953/6	0.00 V		
C00953/7	80.00 V		
C00953/8	160.00 V		
C00953/9	240.00 V		
C00953/10	320.00 V		
C00953/11	400.00 V		
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00954

Parameter | Name: **C00954 | VFC: Activat. interpol. point n**		Data type: UNSIGNED_32 Index: 23621_d = $5C45_h$
From software version V3.0		
		▸ V/f control
Selection list		
0	Off	
1	ON	
Subcodes	**Lenze setting**	**Information**
C00954/1	1: ON	
C00954/2	1: ON	
C00954/3	1: ON	
C00954/4	1: ON	
C00954/5	1: ON	
C00954/6	1: ON	
C00954/7	1: ON	
C00954/8	1: ON	
C00954/9	1: ON	
C00954/10	1: ON	
C00954/11	1: ON	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00955

Parameter | Name: **C00955 | VFC: Vmax reduction**			Data type: UNSIGNED_32 Index: 23620_d = $5C44_h$
From software version V3.0			
			▸ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
0	V	500	**0 V**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00957

Parameter | Name: **C00957 | VFC: VVC current setpoint**			Data type: INTEGER_32 Index: 23618_d = $5C42_h$
From software version V3.0			
			▸ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	A	1500.00	**0.00 A**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00958

Parameter | Name: **C00958 | VFC: VVC gain**			Data type: UNSIGNED_32 Index: 23617_d = $5C41_h$
From software version V3.0			
			▸ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	V/A	750.00	**0.00 V/A**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00959

Parameter | Name: C00959 | VFC: VVC reset time			Data type: UNSIGNED_32 Index: 23616_d = $5C40_h$
From software version V3.0			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.01	ms	2000.00	**2000.00 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00960

Parameter | Name: C00960 | VFC: V/f voltage boost			Data type: INTEGER_32 Index: 23615_d = $5C3F_h$
From software version V3.0			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
0	V	1000	**0 V**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00961

Parameter | Name: C00961 | VFC: Load - cw/ccw-operation		Data type: UNSIGNED_32 Index: 23614_d = $5C3E_h$
From software version V3.0		▶ V/f control
Selection list (Lenze setting printed in bold)		
0	**CW: mot. / CCW: mot.**	
1	CW: mot. / CCW: regen.	
2	CW: regen. / CCW: mot.	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00962

Parameter | Name: C00962 | VFC: Load adjustment			Data type: UNSIGNED_32 Index: 23613_d = $5C3D_h$
From software version V3.0			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	%	200.00	**20.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00963

Parameter | Name: C00963 | VFC: Gain - Imax controller			Data type: UNSIGNED_32 Index: 23612_d = $5C3C_h$
From software version V3.0			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	Hz/A	1000.000	**0.001 Hz/A**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C00964

Parameter | Name: C00964 | VFC: Reset time - Imax contr.			Data type: UNSIGNED_32 Index: 23611_d = $5C3B_h$
From software version V3.0			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
1.0	ms	2000.0	**100.0 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00965

Parameter | Name: **C00965 | VFC: Gain - slip compensation**			Data type: INTEGER_32 Index: 23610_d = $5C3A_h$
From software version V3.0			
			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
-200.00	%	200.00	**0.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00966

Parameter | Name: **C00966 | VFC: Time const. - slip compens.**			Data type: UNSIGNED_32 Index: 23609_d = $5C39_h$
From software version V3.0			
			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
1	ms	6000	**2000 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00967

Parameter | Name: **C00967 | VFC: Gain - oscillation damping**			Data type: INTEGER_32 Index: 23608_d = $5C38_h$
From software version V3.0			
			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
-100	%	100	**20 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00968

Parameter | Name: **C00968 | VFC: Time const. - oscill. damp.**			Data type: INTEGER_32 Index: 23607_d = $5C37_h$
From software version V3.0			
			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
1	ms	600	**5 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00969

Parameter | Name: **C00969 | VFC: Limitation - oscill. damp.**			Data type: INTEGER_32 Index: 23606_d = $5C36_h$
From software version V3.0			
			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.1	Hz	20.0	**0.2 Hz**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00970

Parameter | Name: **C00970 | VFC: ramp-end frequ. - oscill. damp.**			Data type: INTEGER_32 Index: 23605_d = $5C35_h$
From software version V5.0			
			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	100	**0 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00971

Parameter | Name:
C00971 | VFC: Influence - speed controller
Data type: UNSIGNED_32
Index: 23604_d = $5C34_h$

From software version V3.0

V/f control

Setting range (min. value | unit | max. value)			Lenze setting
0.00	%	100.00	**10.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00972

Parameter | Name:
C00972 | VFC: Gain - speed controller
Data type: UNSIGNED_32
Index: 23603_d = $5C33_h$

From software version V3.0

V/f control

Setting range (min. value | unit | max. value)			Lenze setting
0.000	Hz/rpm	1000.000	**0.000 Hz/rpm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C00973

Parameter | Name:
C00973 | VFC: Reset time - speed contr.
Data type: UNSIGNED_32
Index: 23602_d = $5C32_h$

From software version V3.0

V/f control

Setting range (min. value | unit | max. value)			Lenze setting
1.0	ms	6000.0	**6000.0 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C00974

Parameter | Name:
C00974 | DC brake: Current
Data type: INTEGER_32
Index: 23601_d = $5C31_h$

From software version V3.0

DC-injection braking

Setting range (min. value | unit | max. value)			Lenze setting
0.00	A	500.00	**0.00 A**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00975

Parameter | Name:
C00975 | DC brake: Current for quick stop
Data type: INTEGER_32
Index: 23600_d = $5C30_h$

From software version V3.0

DC-injection braking

Setting range (min. value | unit | max. value)			Lenze setting
0.00	A	500.00	**0.00 A**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00976

Parameter | Name:
C00976 | DC brake: Activat. by quick stop
Data type: UNSIGNED_32
Index: 23599_d = $5C2F_h$

From software version V3.0

DC-injection braking

Selection list (Lenze setting printed in bold)		
0	**Off**	
1	ON	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C00977

Parameter | Name: **C00977 | Min. inh-time aft. overvolt**			Data type: UNSIGNED_32 Index: 23598_d = $5C2E_h$
From software version V3.0			▶ V/f control
Setting range (min. value | unit | max. value)			**Lenze setting**
1	ms	10000	**500 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00980

Parameter | Name: **C00980 | VFC: Override point of field weakening**			Data type: INTEGER_32 Index: 23595_d = $5C2B_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
-500	Hz	500	**0 Hz**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00985

Parameter | Name: **C00985 | SLVC: Gain - flux controller**			Data type: UNSIGNED_32 Index: 23590_d = $5C26_h$
From software version V3.0			▶ Sensorless vector control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00		21474836.47	**0.00**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00986

Parameter | Name: **C00986 | SLVC: Gain - cross curr. contr.**			Data type: UNSIGNED_32 Index: 23589_d = $5C25_h$
From software version V3.0			▶ Sensorless vector control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00		21474836.47	**0.00**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00987

Parameter | Name: **C00987 | SLVC: Gain - torque controller**			Data type: UNSIGNED_32 Index: 23588_d = $5C24_h$
From software version V3.0			▶ Sensorless vector control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Hz/A	1000.0000	**0.5000 Hz/A**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C00988

Parameter | Name: **C00988 | SLVC: Reset time - torque contr.**			Data type: UNSIGNED_32 Index: 23587_d = $5C23_h$
From software version V3.0			▶ Sensorless vector control
Setting range (min. value | unit | max. value)			**Lenze setting**
0.01	ms	2000.00	**10.00 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00989

Parameter | Name: **C00989 | SLVC: Time const.- Para. adj.**			Data type: UNSIGNED_32 Index: 23586_d = $5C22_h$

From software version V3.0

Sensorless vector control

Setting range (min. value | unit | max. value)			
0	ms	20000	
Subcodes	**Lenze setting**		**Information**
C00989/1	20000 ms		SLVC: Time const.- Para.Rs adj.
C00989/2	20000 ms		SLVC: Time const.- Para.Lh adj.

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00990

Parameter | Name: **C00990 | Flying restart: Activation**	Data type: UNSIGNED_32 Index: 23585_d = $5C21_h$

From software version V8.0

Shifting the override point for field weakening

- In the VFCplus operating mode the pull-out protection function or the maximally permissible current in the field weakening range can be adapted.

Flying restart function

From software version V3.0

Note!

Only deactivate the flying restart for the V/f control or sensorless vector control if it is ensured that the drive is always at standstill in the case of controller enable!

Flying restart function

Selection list (Lenze setting printed in bold)	
0	**Off**
1	ON

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00991

Parameter | Name: **C00991 | Flying restart: Current**	Data type: INTEGER_32 Index: 23584_d = $5C20_h$

From software version V3.0

Flying restart function

Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	100	**15 %**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00992

Parameter | Name: **C00992 | Flying restart: Start frequency**	Data type: INTEGER_32 Index: 23583_d = $5C1F_h$

From software version V3.0

Flying restart function

Setting range (min. value | unit | max. value)			**Lenze setting**
-600.0	Hz	600.0	**20.0 Hz**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 10

C00993

Parameter | Name: **C00993 | Flying restart: Integration time**			Data type: UNSIGNED_32 Index: 23582_d = $5C1E_h$
From software version V3.0			

Flying restart function

Setting range (min. value | unit | max. value)			Lenze setting
1	ms	6000	**60 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00994

Parameter | Name: **C00994 | Flying restart: Min. deviation**			Data type: UNSIGNED_32 Index: 23581_d = $5C1D_h$
From software version V3.0			

Flying restart function

Setting range (min. value | unit | max. value)			Lenze setting
0.00	°	90.00	**5.00 °**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C00995

Parameter | Name: **C00995 | Flying restart: Delay time**			Data type: UNSIGNED_32 Index: 23580_d = $5C1C_h$
From software version V3.0			

Flying restart function

Setting range (min. value | unit | max. value)			Lenze setting
0	ms	10000	**0 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C00998

Parameter | Name: **C00998 | VFC: Frequency setpoint**			Data type: INTEGER_32 Index: 23577_d = $5C19_h$
From software version V3.0			

V/f control

Display range (min. value | unit | max. value)			
-800.0	Hz	800.0	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C01120

Parameter | Name: **C01120 | Sync source**	Data type: UNSIGNED_8 Index: 23455_d = $5B9F_h$

Selection of the source for the synchronisation signals.
- The drive can only be synchronised by one source.

Note:
Set the selection "2: CAN module" for the communication module CANopen (E94AYCCA).

"CAN on board" system bus: sync telegram

Selection list (Lenze setting printed in bold)	
0	**Off**
1	CAN on board
2	CAN module
4	Module in MXI1
5	Module in MXI2
6	Digital input 1
7	Digital input 2
8	Digital input 3
9	Digital input 4
10	Digital input 5
11	Digital input 6
12	Digital input 7
13	Digital input 8

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01121

Parameter | Name: **C01121 | Sync cycle time**	Data type: UNSIGNED_32 Index: 23454_d = $5B9E_h$

Time interval at which the phase control loop (PLL) in the controller expects the synchronisation signals.
- The time interval set must correspond to the cycle of the synchronisation source.

Note: For synchronisation via the system bus (CANopen), only integer multiples of 1000 µs should be set.
Example: For the system bus, the interval between two synchronisation signals is set to 2 ms. If the system bus is to be used as the 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**
250	µs	20000	**1000 µs**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01122

Parameter | Name: **C01122 | Sync phase position**	Data type: UNSIGNED_32 Index: 23453_d = $5B9D_h$

The phase position defines the zero point of time for the application relating to the synchronisation signal (bus cycle). Since PDO processing is integrated in the system part of the application, the instant of the PDO acceptance also changes if the phase position is changed.
- If 0 is set, the application is started together with the synchronisation signal.
- If a value > 0 is set, the application starts by the set time interval before the synchronisation signal arrives (the phase position acts negatively).

Example: If the phase position is set to 400 µs, the system part of the application starts 400 µs before the synchronisation signal arrives.
From software version V3.0:
The effect of the sync phase position can be affected by the application cycle set in C01130. For the Lenze setting of C01130 the behaviour remains as before.

"CAN on board" system bus: sync telegram

Setting range (min. value | unit | max. value)			**Lenze setting**
0	µs	64000	**400 µs**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01123

Parameter | Name: **C01123 | Sync tolerance**			Data type: UNSIGNED_32 Index: 23452_d = $5B9C_h$

Time slot for monitoring the synchronisation signal via the **LS_SyncInput** system block
- If the last synchronisation signal amounted to approx. the expected value within this time slot, the *SYNC_bSyncInsideWindow* output of the **LS_SyncInput** system block is set to TRUE.
- This setting does not affect the synchronisation process.

"CAN on board" system bus: sync telegram

Setting range (min. value | unit | max. value)			**Lenze setting**
0	µs	1000	**0 µs**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01124

Parameter | Name: **C01124 | Sync-PLL increment**	Data type: UNSIGNED_8 Index: 23451_d = $5B9B_h$

When the cycle times of the synchronisation signal and the phase control loop (PLL) differ, this setting defines the increment to be used to readjust the phase control loop.
- If the system bus (CANopen) is used as the synchronisation source, the recommended value is 109 ns.

"CAN on board" system bus: sync telegram

Selection list (Lenze setting printed in bold)	
1	7 ns
2	15 ns
3	23 ns
4	31 ns
5	39 ns
6	46 ns
7	54 ns
8	62 ns
9	70 ns
10	78 ns
11	85 ns
12	93 ns
13	101 ns
14	**109 ns**
15	117 ns
16	125 ns
17	132 ns
18	140 ns
19	148 ns
20	156 ns
21	164 ns
22	171 ns
23	179 ns
24	187 ns
25	195 ns
26	203 ns
27	210 ns

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01125

Parameter | Name: **C01125 | Service code**	Data type: UNSIGNED_32 Index: 23450_d = $5B9A_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01126

Parameter | Name: **C01126 | Service code**	Data type: UNSIGNED_32 Index: 23449_d = $5B99_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01127

Parameter | Name: **C01127 | Service code**	Data type: UNSIGNED_32 Index: 23448_d = $5B98_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01128

Parameter | Name: **C01128 | Service code**	Data type: UNSIGNED_32 Index: 23447_d = $5B97_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01129

Parameter | Name: **C01129 | Service code**	Data type: UNSIGNED_32 Index: 23446_d = $5B96_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01130

Parameter | Name: **C01130 | CAN SYNC application cycle**	Data type: UNSIGNED_16 Index: 23445_d = $5B95_h$

From software version V3.0

This parameter affects the effect of the sync phase position (C01122) regarding the instant of acceptance of the synchronous PDOs in the application and the instant of transmission of the synchronous PDOs on the system bus (CANopen).

- The resulting PDO delay can be calculated with the following formula taking into account an internal processing time of 150 µs: PDO delay = (C01121 - C01122 + 150 µs) modulo C01130
- For the Lenze setting, the behaviour remains as before, the sync phase position (C01122) is always calculated modulo 1000.
- The set value is automatically rounded down to multiples of 1000 µs.

▶ Effect of C01130 on the sync phase position

Note: Setting the application cycle to a higher value than the sync cycle time (C01121) results in undefined behaviour. The same applies if the value set for the sync phase position (C01122) is higher than the sync cycle time (C01121). In this case, the drive usually cannot send synchronous PDOs on the system bus anymore.

▶ "CAN on board" system bus: sync telegram

Setting range (min. value | unit | max. value)			**Lenze setting**
1000	µs	65000	**1000 µs**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01190

Parameter | Name: **C01190 | Motor thermal sensor**	Data type: UNSIGNED_32 Index: 23385_d = $5B59_h$

▶ Motor temperature monitoring

Selection list (Lenze setting printed in bold)		**Information**
0	**KTY83-110**	Lenze standard KTY83-110 (MDSKX, MCS06)
1	Spec. characteristic	Characteristic defined via C01191 and C01192.
2	KTY83-110 + 2 x PTC	Lenze standard KTY83-110 + 2 x PTC 150°C (MCS09-MCS19)
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C01191

Parameter | Name: **C01191 | Spec. charact.: temperature**		Data type: UNSIGNED_32 Index: 23384_d = $5B58_h$
The special thermal sensor characteristic is selected by setting C01190="1"		
		▸ Motor temperature monitoring
Setting range (min. value | unit | max. value)		
0	°C	255
Subcodes	**Lenze setting**	**Information**
C01191/1	25 °C	Value 1 for spec. thermal sensor characteristic
C01191/2	150 °C	Value 2 for spec. thermal sensor characteristic
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C01192

Parameter | Name: **C01192 | Spec. characteristic: resistance**		Data type: UNSIGNED_32 Index: 23383_d = $5B57_h$
The special thermal sensor characteristic is selected by setting C01190="1"		
		▸ Motor temperature monitoring
Setting range (min. value | unit | max. value)		
0	Ohm	30000
Subcodes	**Lenze setting**	**Information**
C01192/1	1000 Ohm	Value 1 for spec. thermal sensor characteristic
C01192/2	2225 Ohm	Value 2 for spec. thermal sensor characteristic
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C01193

Parameter | Name: **C01193 | Motor temp. feedback system**		Data type: UNSIGNED_16 Index: 23382_d = $5B56_h$
Selection of feedback system for motor temperature detection.		
		▸ Motor temperature monitoring
Selection list (Lenze setting printed in bold)		
0	**Speed feedback**	
1	X7 (Input Resolver)	
2	X8 (Input Encoder)	
3	Reserved	
4	Reserved	
5	X7 and X8 parallel	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C01194

Parameter | Name: **C01194 | Motor operating temperature**			Data type: INTEGER_32 Index: 23381_d = $5B55_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
1	°C	200	**140 °C**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01195

Parameter | Name: **C01195 | Influence winding I²xt mon.**			Data type: UNSIGNED_32 Index: 23380_d = $5B54_h$

I^2xt motor monitoring: Influence of the winding temperature
- By setting "0 %", the time constant for the winding is not considered and the thermal model is only calculated with the time constant set for the housing/laminated core.

Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	100	**0 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01196

Parameter | Name: **C01196 | S1 torque characteristic I²xt mon.**			Data type: UNSIGNED_32 Index: 23379_d = $5B53_h$

I^2xt motor monitoring: Speed-dependent evaluation of the motor current
- By selecting a characteristic, the permissible motor current is evaluated depending on speed for calculating the thermal motor utilisation.

Setting range (min. value | unit | max. value)			
0	%	600	

Subcodes	**Lenze setting**	**Information**
C01196/1	0 %	S1 torque characteristic n1/nn
C01196/2	100 %	S1 torque characteristic I1/In
C01196/3	0 %	S1 torque characteristic n2/nn
C01196/4	100 %	S1 torque characteristic I2/In
C01196/5	100 %	S1 torque characteristic n3/nn
C01196/6	100 %	S1 torque characteristic I3/In
C01196/7	100 %	S1 torque characteristic n4/nn
C01196/8	100 %	S1 torque characteristic I4/In
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C01198

Parameter | Name: **C01198 | Async. motor: Stall protection**			Data type: UNSIGNED_32 Index: 23377_d = $5B51_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
0	%	100	**0 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01199

Parameter | Name: **C01199 | Enhanced Power**	Data type: UNSIGNED_32 Index: 23376_d = $5B50_h$

From software version V3.0

Stop!
During operation with increased continuous power, the max. permissible ambient temperature is reduced to 40 °C. The overload current must be reduced. An overload current of 180 % for 10 s is no longer permissible during operation with increased continuous power.

▶ Operation with increased continuous power

Selection list (Lenze setting printed in bold)		
0	**Enhanced Power off**	
1	Enhanced Power Mode 1 on	
2	Enhanced Power Mode 2 on	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C01200

Parameter | Name: **C01200 | Dual motor temperature**			Data type: INTEGER_32 Index: 23375_d = $5B4F_h$
From software version V7.0			

Motor temperature monitoring
Motor temperature monitoring of a second motor

Display range (min. value | unit | max. value)			
-200	°C	200	
Subcodes			**Information**
C01200/1			Motor temperature via X7
C01200/2			Motor temperature via X8
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01201

Parameter | Name: **C01201 | Delay time for fan start**		Data type: UNSIGNED_32 Index: 23374_d = $5B4E_h$
Selection list (Lenze setting printed in bold)		**Information**
0	**Via power section serial no.**	
1	500 ms	
2	1000 ms	
3	1500 ms	
4	2000 ms	
5	2500 ms	
6	3000 ms	
7	3500 ms	
8	4000 ms	
9	4500 ms	
10	5000 ms	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C01203

Parameter | Name: **C01203 | Counter: Brake chopper overload**			Data type: UNSIGNED_16 Index: 23372_d = $5B4C_h$
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01204

Parameter | Name: **C01204 | Counter: Ixt overload**			Data type: UNSIGNED_16 Index: 23371_d = $5B4B_h$
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01205

Parameter | Name: **C01205 | Counter: DC bus overvoltage**			Data type: UNSIGNED_16 Index: 23370_d = $5B4A_h$
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01206

Parameter | Name: **C01206 | Counter: Mains switching**			Data type: UNSIGNED_16 Index: 23369_d = $5B49_h$
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01208

Parameter | Name: **C01208 | Counter: Heatsink overtemp.**			Data type: UNSIGNED_16 Index: 23367_d = $5B47_h$
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01209

Parameter | Name: **C01209 | Counter: Housing overtemp.**			Data type: UNSIGNED_16 Index: 23366_d = $5B46_h$
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01210

Parameter | Name: **C01210 | Counter: Internal**			Data type: UNSIGNED_8 Index: 23365_d = $5B45_h$
Display range (min. value | unit | max. value)			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01211

Parameter | Name: **C01211 | Service code**	Data type: UNSIGNED_32 Index: 23364_d = $5B44_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01212

Parameter | Name: **C01212 | Counter: Power section overload**			Data type: UNSIGNED_16 Index: 23363_d = $5B43_h$
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01213

Parameter | Name: **C01213 | Service code DataFlash**	Data type: UNSIGNED_32 Index: 23362_d = $5B42_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01214

Parameter | Name: **C01214 | Internal clock**	Data type: VISIBLE_STRING Index: 23361_d = $5B41_h$

Display of the system time of the controller in the format "dd/mm/yyyy hh:mm:ss"
- Time and date are set via C01215.

If the MM440 memory module with real-time clock is plugged into the controller, the internal clock is adjusted to the real-time clock every time the mains is switched on and every 24 hours at 0:00 o'clock.
- The daily adjustment is executed on a low-priority level. This is why it may take some seconds until the adjusted time is displayed.
- During the adjustment process, status information of the real-time clock is also queried and entered into the logbook.

Note:
If a memory module without real-time clock is plugged into the controller, the internal clock is initialised with "01.01.1970 00:00:00" every time the mains is switched on.

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01215

Parameter | Name: **C01215 | Set time and date**	Data type: UNSIGNED_16 Index: 23360_d = $5B40_h$

From software version V1.5
Setting of the system time of the controller displayed in C01214.
- If the MM440 memory module with real-time clock is plugged into the controller, the real-time clock is set simultaneously.

Set time and date.
- Before writing to a subcode for the first time, the current time information according to the internal clock is displayed in the subcodes.
- When a value has been written into a subcode, the displays in the subcodes freeze to the last values.
- The new system time set is only accepted after a value has been written into each subcode at least once.
- After the new system time has been accepted, the display in the subcodes is updated according to the internal clock.

Note:
If a memory module without real-time clock is plugged into the controller, the internal clock is initialised with "01.01.1970 00:00:00" every time the mains is switched on.

Setting range (min. value | unit | max. value)			
0		65535	
Subcodes	**Lenze setting**		**Information**
C01215/1	0		Seconds
C01215/2	0		Minutes
C01215/3	0		Hours
C01215/4	1		Day
C01215/5	1		Month
C01215/6	1970		Year

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer Scaling factor: 1

C01217

Parameter | Name: **C01217 | Service code**	Data type: VISIBLE_STRING Index: 23358_d = $5B3E_h$

This code is used internally by the controller and must not be overwritten by the user!

C01218

Parameter | Name: **C01218 | Service code**	Data type: UNSIGNED_32 Index: 23357_d = $5B3D_h$

This code is used internally by the controller and must not be overwritten by the user!

C01220

Parameter | Name: **C01220 | MEC history: RAM address**	Data type: UNSIGNED_32 Index: 23355_d = $5B3B_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01221

Parameter | Name: **C01221 | MEC history: RAM value**	Data type: UNSIGNED_32 Index: 23354_d = $5B3A_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01222

Parameter | Name: **C01222 | MEC history: Flash value**	Data type: UNSIGNED_32 Index: 23353_d = $5B39_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01223

Parameter | Name: **C01223 | MEC history: Error number**	Data type: UNSIGNED_32 Index: 23352_d = $5B38_h$
This code is used internally by the controller and must not be overwritten by the user!	

C01230

Parameter | Name: **C01230 | Resp. to communication task overflow**		Data type: UNSIGNED_8 Index: 23345_d = $5B31_h$
Selection list (Lenze setting printed in bold)		**Information**
1	**Fault**	
2	Trouble	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
6	Information	
0	No response	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C01501

Parameter | Name: **C01501 | Resp. to comm. error with MXI1**		Data type: UNSIGNED_32 Index: 23074_d = $5A22_h$
Response to communication error between "intelligent" module in module slot 1 and basic device		
Selection list (Lenze setting printed in bold)		
0	**No response**	
1	Fault	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C01502

Parameter | Name: **C01502 | Resp. to comm. error with MXI2**	Data type: UNSIGNED_32 Index: 23073_d = $5A21_h$
Response to communication error between "intelligent" module in module slot 2 and standard device	

Selection list (Lenze setting printed in bold)	
0	**No response**
1	Fault
3	Quick stop by trouble
4	Warning locked
5	Warning

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer   Scaling factor: 1

C01510

Parameter | Name: **C01510 | Ethernet IP address client x**	Data type: VISIBLE_STRING Index: 23065_d = $5A19_h$

Display of the three possible server channels
- If a client is connected via one of these server channels, the IP address and the port of the client will be indicated in the form of "xxx.xxx.xxx.xxx : yyyy".
- If no client is connected via the server channel, "---.---.---.--- : ----" will be indicated.

Subcodes	Information
C01510/1	Server channel 1 ... 3
C01510/...	
C01510/3	

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer   Scaling factor: 1

C01511

Parameter | Name: **C01511 | Ethernet status client x**	Data type: UNSIGNED_8 Index: 23064_d = $5A18_h$

Status of the three possible server channels

Selection list (display only)	
0	Not connected
1	Connected
2	Stop
3	Unknown status

Subcodes	Information
C01511/1	State of server channels 1 ... 3
C01511/...	
C01511/3	

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer   Scaling factor: 1

C01902

Parameter | Name: **C01902 | Diagnostics X6: Max. baud rate**	Data type: UNSIGNED_32 Index: 22673_d = 5891_h

Maximum permissible baud rate of the standard device after determination of the baud rate at the diagnostic interface X6
- Communication starts with the default standard device baud rate of 19200 baud.

Selection list (Lenze setting printed in bold)	
9600	9600 baud
19200	19.200 baud
38400	38.400 baud
57600	57.600 baud
115200	115.200 baud
230400	230.400 baud
375000	375.000 baud
750000	750.000 baud
1500000	1.500.000 baud
3000000	**3.000.000 baud**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01903

Parameter | Name: **C01903 | Diagnostics X6: Change baud rate**	Data type: UNSIGNED_32 Index: 22672_d = 5890_h

New determination of the baud rate at the diagnostic interface X6

Selection list (Lenze setting printed in bold)	
1	Set a higher baud rate
0	**Ignore changes**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer Scaling factor: 1

C01905

Parameter | Name: **C01905 | Diagnostics X6: Curr. baud rate**	Data type: UNSIGNED_32 Index: 22670_d = $588E_h$

Current baud rate at diagnostics interface X6

Display range (min. value | unit | max. value)		
0		3000000

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02104

Parameter | Name: **C02104 | Program auto-start**	Data type: UNSIGNED_32 Index: 22471_d = $57C7_h$

Selection list (Lenze setting printed in bold)	
0	Off
1	**Autom. after mains connection**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02108

Parameter | Name: **C02108 | Program status**	Data type: UNSIGNED_8 Index: 22467_d = $57C3_h$

Selection list (display only)	
1	Program stopped
0	Program is running
2	Program stopped at breakpoint

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02109

Parameter | Name: **C02109 | Program runtime**			Data type: UNSIGNED_16 Index: 22466_d = $57C2_h$
Display range (min. value | unit | max. value)			
0	µs	65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02110

Parameter | Name: **C02110 | User code memory load**			Data type: UNSIGNED_32 Index: 22465_d = $57C1_h$
From software version V2.0			
Display range (min. value | unit | max. value)			
0	%	100	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02111

Parameter | Name: **C02111 | Resp. to task overflow**		Data type: UNSIGNED_8 Index: 22464_d = $57C0_h$
From software version V5.0 Response to a task overflow in the application or user task.		
Selection list (Lenze setting printed in bold)		
1	**Fault**	
3	Quick stop by trouble	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02113

Parameter | Name: **C02113 | Program name**	Data type: VISIBLE_STRING Index: 22462_d = $57BE_h$
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02121

Parameter | Name: **C02121 | Runtime ApplicationTask**			Data type: UNSIGNED_32 Index: 22454_d = $57B6_h$
▸ Runtime measurement			
Display range (min. value | unit | max. value)			
0	µs	3600000000	
Subcodes			**Information**
C02121/1			Curr. runtime ApplicationTask
C02121/2			Max. runtime ApplicationTask
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02122

Parameter | Name: **C02122 | Runtime UserTask**			Data type: UNSIGNED_32 Index: 22453_d = $57B5_h$
▸ Runtime measurement			
Display range (min. value | unit | max. value)			
0	µs	3600000000	
Subcodes			**Information**
C02122/1			Curr. runtime UserTask
C02122/2			Max. runtime UserTask
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02123

Parameter | Name: **C02123 | Runtime IdleTask**			Data type: UNSIGNED_32 Index: 22452_d = $57B4_h$
▶ Runtime measurement			
Display range (min. value | unit | max. value)			
0	µs	3600000000	
Subcodes			**Information**
C02123/1			Curr. runtime IdleTask
C02123/2			Max. runtime IdleTask
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02520

Parameter | Name: **C02520 | Gearbox factor numerator: Motor**			Data type: INTEGER_32 Index: 22055_d = 5627_h
			▶ Drive interface
Setting range (min. value | unit | max. value)			**Lenze setting**
1		2147483647	**1**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02521

Parameter | Name: **C02521 | Gearbox factor denom.: Motor**			Data type: INTEGER_32 Index: 22054_d = 5626_h
			▶ Drive interface
Setting range (min. value | unit | max. value)			**Lenze setting**
1		2147483647	**1**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02522

Parameter | Name: **C02522 | Gearbox factor num.: Pos. enc.**			Data type: INTEGER_32 Index: 22053_d = 5625_h
			▶ Drive interface
Setting range (min. value | unit | max. value)			**Lenze setting**
1		2147483647	**1**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02523

Parameter | Name: **C02523 | Gearbox fac. denom.: Pos. enc.**			Data type: INTEGER_32 Index: 22052_d = 5624_h
			▶ Drive interface
Setting range (min. value | unit | max. value)			**Lenze setting**
1		2147483647	**1**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02524

Parameter | Name: **C02524 | Feed constant**			Data type: UNSIGNED_32 Index: 22051_d = 5623_h
			▶ Drive interface
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0001	Unit/rev.	214748.3647	**360.0000 Unit/rev.**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02525

Parameter | Name: **C02525 | Unit**		Data type: UNSIGNED_32 Index: 22050_d = 5622_h

▶ Drive interface

Selection list (Lenze setting printed in bold)		Information
0	User-defined	The text entered in C02526 is displayed for the unit.
1	Incr.	
2	µm	
3	mm	
4	m	
5	inch	
6	yard	
7	**°**	

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02526

Parameter | Name: **C02526 | User-defined unit**	Data type: VISIBLE_STRING Index: 22049_d = 5621_h

User-defined unit which is displayed when C02525="0" is selected.

▶ Drive interface

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02527

Parameter | Name: **C02527 | Motor mounting direction**	Data type: UNSIGNED_32 Index: 22048_d = 5620_h

▶ Drive interface

Selection list (Lenze setting printed in bold)	
0	**Motor rotating CW**
1	Motor rotating CCW

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02528

Parameter | Name: **C02528 | Traversing range**	Data type: UNSIGNED_32 Index: 22047_d = $561F_h$

▶ Drive interface

Selection list (Lenze setting printed in bold)	
0	**Unlimited**
1	Limited
2	Modulo

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02529

Parameter | Name: **C02529 | Position encoder mounting dir.**	Data type: UNSIGNED_32 Index: 22046_d = $561E_h$

▶ Drive interface

Selection list (Lenze setting printed in bold)	
0	**Encoder rotating CW**
1	Encoder rotating CCW

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02530

Parameter | Name: **C02530 | Active function state**	Data type: INTEGER_32 Index: 22045_d = $561D_h$
Displays the basic drive function that currently controls the drive.	
	▸ Basic drive functions: Internal state machine

Selection list (display only)	
0	Program stopped
1	Initial/boot state active
2	Torque follower active
3	Speed follower active
4	Position follower active
5	Setpoint follower active
6	Positioning active
7	Homing active
8	Manual jog active
9	Brake test active
10	Drive at standstill
11	Drive is stopped
12	Quick stop active
13	Reserve 1
14	Controller is not ready
15	Initialisation
16	Fault
17	Manual control open loop active
18	Pole position identification active

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02531

Parameter | Name: **C02531 | Gearbox factors (decimal)**	Data type: UNSIGNED_32 Index: 22044_d = $561C_h$

Note: In subcode 3 the effective gearbox factor resulting form the motor and the load is displayed if a separate position encoder is configured and the position control is activated (C02570="2"). For a different encoder configuration (without a separate position encoder) the value "1" is shown in subcode 3.

▸ Drive interface

Display range (min. value | unit | max. value)		
0.000		2147483.647

Subcodes	**Information**
C02531/1	Motor gearbox factor (dec.)
C02531/2	Pos. enc. gearbox factor (dec.)
C02531/3	Effective gearbox factor (dec.)

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1000

C02532

Parameter | Name: **C02532 | Resolution of a unit**	Data type: UNSIGNED_32 Index: 22043_d = $561B_h$

▸ Drive interface

Display range (min. value | unit | max. value)		
0	Incr./unit	2147483647

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02533

Parameter | Name: **C02533 | Time unit**	Data type: UNSIGNED_32 Index: 22042_d = $561A_h$

Drive interface

Selection list (display only)	
2	s

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02534

Parameter | Name: **C02534 | Used time unit**	Data type: VISIBLE_STRING Index: 22041_d = 5619_h

Display of the time unit as a character string

Drive interface

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02535

Parameter | Name: **C02535 | Used unit**	Data type: VISIBLE_STRING Index: 22040_d = 5618_h

Display of the unit set in C02525 and C02526 as a character string

Drive interface

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02536

Parameter | Name: **C02536 | Cycle**	Data type: UNSIGNED_32 Index: 22039_d = 5617_h

Drive interface

Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit	214748.3647	**360.0000 Unit**

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 10000

C02537

Parameter | Name: **C02537 | Speed unit**	Data type: VISIBLE_STRING Index: 22038_d = 5616_h

Drive interface

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02538

Parameter | Name: **C02538 | Acceleration unit**	Data type: VISIBLE_STRING Index: 22037_d = 5615_h

Drive interface

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02539

Parameter | Name: **C02539 | Max. presentable position**	Data type: INTEGER_32 Index: 22036_d = 5614_h

Drive interface

Display range (min. value | unit | max. value)		
-214748.3647	Unit	214748.3647

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 10000

C02540

Parameter | Name: **C02540 | Max. presentable speed**			Data type: INTEGER_32 Index: 22035_d = 5613_h

Drive interface

Display range (min. value | unit | max. value)			
-214748.3647	Unit/s	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02541

Parameter | Name: **C02541 | Max. presentable acceleration**			Data type: INTEGER_32 Index: 22034_d = 5612_h

Drive interface

Display range (min. value | unit | max. value)			
-214748.3647	Unit/s²	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02542

Parameter | Name: **C02542 | Load reference speed**			Data type: UNSIGNED_32 Index: 22033_d = 5611_h

Drive interface

Display range (min. value | unit | max. value)			
0.000	rpm	4294967.295	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02543

Parameter | Name: **C02543 | Load reference torque**			Data type: UNSIGNED_32 Index: 22032_d = 5610_h

Drive interface

Display range (min. value | unit | max. value)			
0.000	Nm	4294967.295	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02544

Parameter | Name: **C02544 | Reference speed**			Data type: INTEGER_32 Index: 22031_d = $560F_h$

From software version V1.5

Drive interface

Display range (min. value | unit | max. value)			
-214748.3647	Unit/s	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02545

Parameter | Name: **C02545 | Reference S-ramp time**			Data type: UNSIGNED_32 Index: 22030_d = $560E_h$

From software version V7.0

Drive interface

Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	s	2147483.647	**0.001 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02547

Parameter | Name: **C02547 | DI_dnState**			Data type: INTEGER_32 Index: 22028_d = $560C_h$
Bit coded status of the drive interface.			
Display range (min. value | unit | max. value)			
-2147483648		2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02548

Parameter | Name: **C02548 | DI_bErrors**		Data type: UNSIGNED_32 Index: 22027_d = $560B_h$
Display of the digital error signals of the drive interface.		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02548/1		DI_bResetError1
C02548/2		DI_bResetError2
C02548/3		DI_bResetError3
C02548/4		DI_bSetExternError
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02549

Parameter | Name: **C02549 | Drive interface: Signals**		Data type: UNSIGNED_32 Index: 22026_d = $560A_h$
Display of the digital signals of the drive interface.		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02549/1		DI_bSetCInh
C02549/2		Reserved
C02549/3		Reserved
C02549/4		DI_bSwitchOn
C02549/5		Reserved
C02549/6		DI_bReady
C02549/7		DI_bFailActive
C02549/8		DI_bImpActive
C02549/9		DI_bCinhActive
C02549/10		DI_bWarningActive
C02549/11		DI_bUVDetected
C02549/12		DI_bOVDetected
C02549/13		DI_bMainSupplyOk
C02549/14		DI_bReadyToSwitchOn
C02549/15		DI_bOperationEnabled
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02550

Parameter | Name: **C02550 | Setpoint interpolation**			Data type: UNSIGNED_32 Index: 22025_d = 5609_h
			▶ Motor interface
Selection list			
0	Off		
1	ON		
Subcodes	**Lenze setting**	**Information**	
C02550/1	1: ON	Position setpoint interpolat.	
C02550/2	1: ON	Speed setpoint interpolat.	
C02550/3	1: ON	Torque setpoint interpolat.	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1	

C02552

Parameter | Name: **C02552 | Position setpoint (mctrl)**			Data type: INTEGER_32 Index: 22023_d = 5607_h
			▶ Motor interface
Display range (min. value | unit | max. value)			
-214748.3647	Unit	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02553

Parameter | Name: **C02553 | Position controller gain**			Data type: UNSIGNED_32 Index: 22022_d = 5606_h
			▶ Motor interface
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	1/s	1000.00	**20.00 1/s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02554

Parameter | Name: **C02554 | Position controller reset time**			Data type: UNSIGNED_32 Index: 22021_d = 5605_h
			▶ Motor interface
Setting range (min. value | unit | max. value)			**Lenze setting**
0.001	s	60.000	**60.000 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02555

Parameter | Name: **C02555 | D component position controller**			Data type: UNSIGNED_32 Index: 22020_d = 5604_h
			▶ Motor interface
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000		100.000	**0.000**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02556

Parameter | Name: **C02556 | Pos. contr. limitation**			Data type: INTEGER_32 Index: 22019_d = 5603_h
			▶ Motor interface
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s	214748.3647	**214748.3647 Unit/s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02557

Parameter | Name: **C02557 | Phase controller output**			Data type: INTEGER_32 Index: $22018_d = 5602_h$
			▸ Motor interface
Display range (min. value | unit | max. value)			
-214748.3647	Unit/s	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02558

Parameter | Name: **C02558 | Pos. contr. output**			Data type: INTEGER_32 Index: $22017_d = 5601_h$
			▸ Motor interface
Display range (min. value | unit | max. value)			
-214748.3647	Unit/s	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02559

Parameter | Name: **C02559 | Internal torque limits**			Data type: INTEGER_32 Index: $22016_d = 5600_h$
			▸ Motor interface
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes			**Information**
C02559/1			Upper int. torque limit
C02559/2			Lower int. torque limit
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02560

Parameter | Name: **C02560 | Messages - motor interface**			Data type: UNSIGNED_32 Index: $22015_d = 55FF_h$
			▸ Motor interface
Display range (min. value | unit | max. value)			
0		4294967295	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02561

Parameter | Name: **C02561 | Speed feedforw. control gain**			Data type: INTEGER_32 Index: $22014_d = 55FE_h$
From software version V1.5 Percentage reduction of the speed feedforward control of the profile generator • Required in some applications if a 100 % speed feedforward control causes overshoots. • Only effective for the basic functions "Positioning", "Homing" and "Manual jog".			
			▸ Motor interface
Setting range (min. value | unit | max. value)			**Lenze setting**
50.00	%	100.00	**100.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02562

Parameter | Name: **C02562 | Filter time constant**			Data type: UNSIGNED_32 Index: 22013$_d$ = 55FD$_h$
From software version V7.0			
			▶ Motor interface
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	s	60.000	**0.002 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02567

Parameter | Name: **C02567 | Control mode**		Data type: UNSIGNED_32 Index: 22008$_d$ = 55F8$_h$
		▶ Motor interface
Selection list (display only)		
0	Position control	
1	Speed control	
2	Torque control	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02568

Parameter | Name: **C02568 | Motor interface: % signals**			Data type: INTEGER_32 Index: 22007$_d$ = 55F7$_h$
Display of the scaled signals of the motor interface.			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes			**Information**
C02568/1			MI_dnPosCtrlAdaptLoad_n
C02568/2			MI_dnPosCtrlAdaptMotor_n
C02568/3			MI_dnSpeedCtrlAdapt_n
C02568/4			MI_dnTorqueHighLimit_n
C02568/5			MI_dnTorqueLowLimit_n
C02568/6			MI_dnTorqueCtrlAdapt_n
C02568/7			MI_dnFluxSetpoint_n
C02568/8			MI_dnInertiaAdapt_n
C02568/9			MI_dnBoostSet_n
C02568/10			MI_dnTorqueAdd_n
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02569

Parameter | Name: **C02569 | Motor interface.: Dig. signals**	Data type: UNSIGNED_32 Index: $22006_d = 55F6_h$
Display of the digital signals of the motor interface.	

Selection list (display only)		
0	FALSE	
1	TRUE	

Subcodes	**Information**
C02569/1	Reserved
C02569/2	MI_bResetSpeedCtrlIntegrator
C02569/3	MI_bLimitationActive
C02569/4	MI_bPosCtrlLimited
C02569/5	MI_bSpeedSetPointLimited
C02569/6	MI_bSpeedCtrlLimited
C02569/7	MI_bTorqueSetpointLimited
C02569/8	MI_bCurrentSetpointLimited
C02569/9	MI_bSpeedBelowC19
C02569/10	MI_bSpeedFollowingError
C02569/11	MI_bMotorOverloadWarning
C02569/12	MI_bMotorOrientationInverse
C02569/13	MI_bFlyingSyncBusy
C02569/14	MI_bClampIsActive
C02569/15	MI_bMagnetisationFinished
C02569/16	MI_bFlyingSyncBlocked
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02570

Parameter | Name: **C02570 | Position control structure**	Data type: UNSIGNED_32 Index: $22005_d = 55F5_h$
Chapter "Controller configuration" provides you with more information on parameter setting.	

▶ Encoder evaluation

Selection list (Lenze setting printed in bold)		**Information**
1	**Phase controller is active**	Motor encoder selection is effected in C00495.
2	Position controller active (<= FW V5.xx)	Position controller selection is effected in C00490.
3	Position controller is active	Position controller selection is effected in C00490.
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02571

Parameter | Name: **C02571 | Source - actual position**	Data type: UNSIGNED_32 Index: $22004_d = 55F4_h$
This code is used internally by the controller and must not be overwritten by the user!	

C02572

Parameter | Name: **C02572 | Speed setpoint (enc. eval.)**	Data type: INTEGER_32 Index: $22003_d = 55F3_h$

▶ Encoder evaluation

Display range (min. value | unit | max. value)		
-214748.3647	Unit/s	214748.3647
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 10000

C02573

Parameter | Name: **C02573 | Position setpoint (enc. eval.)**			Data type: INTEGER_32 Index: 22002_d = $55F2_h$
			▸ Encoder evaluation
Display range (min. value | unit | max. value)			
-214748.3647	Unit	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02574

Parameter | Name: **C02574 | Actual speed (enc. eval.)**			Data type: INTEGER_32 Index: 22001_d = $55F1_h$
			▸ Encoder evaluation
Display range (min. value | unit | max. value)			
-214748.3647	Unit/s	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02575

Parameter | Name: **C02575 | Actual position (enc. eval.)**			Data type: INTEGER_32 Index: 22000_d = $55F0_h$
			▸ Encoder evaluation
Display range (min. value | unit | max. value)			
-214748.3647	Unit	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02576

Parameter | Name: **C02576 | Following error**			Data type: INTEGER_32 Index: 21999_d = $55EF_h$
			▸ Encoder evaluation
Display range (min. value | unit | max. value)			
-214748.3647	Unit	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02577

Parameter | Name: **C02577 | External actual position**			Data type: INTEGER_32 Index: 21998_d = $55EE_h$
			▸ Encoder evaluation
Display range (min. value | unit | max. value)			
-214748.3647	Unit	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02578

Parameter | Name: **C02578 | Offset actual pos. value/setp.**			Data type: INTEGER_32 Index: 21997_d = $55ED_h$
			▸ Encoder evaluation
Display range (min. value | unit | max. value)			
-214748.3647	Unit	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02579

Parameter | Name: **C02579 | Encoder eval.: Dig. signals**	Data type: UNSIGNED_32 Index: 21996_d = $55EC_h$
Display of the digital signals of the encoder evaluation.	
Selection list (display only)	
0 FALSE	
1 TRUE	
Subcodes	**Information**
C02579/1	FDB_bResolverError
C02579/2	FDB_bSinCosSignalError
C02579/3	FDB_bEncoderComError
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02580

Parameter | Name: **C02580 | Operating mode brake**		Data type: UNSIGNED_32 Index: 21995_d = $55EB_h$
		▸ Basic function "Brake control"
Selection list (Lenze setting printed in bold)		
0	**Brake control off**	
1	Directly with brake module	
2	Autom. with brake module	
11	Directly - external switching	
12	Autom. - external switching	
22	Autom. - DC brake	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02581

Parameter | Name: **C02581 | Threshold - brake activation**			Data type: INTEGER_32 Index: 21994_d = $55EA_h$
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0	rpm	50000	**50 rpm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02582

Parameter | Name: **C02582 | Brake resp. to pulse inhibit**		Data type: UNSIGNED_32 Index: 21993_d = $55E9_h$
		▸ Basic function "Brake control"
Selection list (Lenze setting printed in bold)		
0	**Activate the brake immediately**	
1	Activate brake at n < nmin	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02583

Parameter | Name: **C02583 | Status input monitoring**		Data type: UNSIGNED_32 Index: 21992_d = $55E8_h$
		▸ Basic function "Brake control"
Selection list (Lenze setting printed in bold)		
0	**Not active**	
1	Active	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02585

Parameter | Name: **C02585 | Brake control polarity**		Data type: UNSIGNED_32 Index: 21990_d = $55E6_h$
		▸ Basic function "Brake control"
Selection list (Lenze setting printed in bold)		
0	**Not inverted**	
1	Inverted	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02586

Parameter | Name: **C02586 | Starting torque 1**			Data type: INTEGER_32 Index: 21989_d = $55E5_h$
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
-21474836.47	Nm	21474836.47	**0.00 Nm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02587

Parameter | Name: **C02587 | Starting torque 2**			Data type: INTEGER_32 Index: 21988_d = $55E4_h$
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
-21474836.47	Nm	21474836.47	**0.00 Nm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02588

Parameter | Name: **C02588 | Source of starting torque**		Data type: UNSIGNED_32 Index: 21987_d = $55E3_h$
		▸ Basic function "Brake control"
Selection list (Lenze setting printed in bold)		
0	**Starting torque 1/2**	
1	Stopping value	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02589

Parameter | Name: **C02589 | Brake closing time**			Data type: UNSIGNED_32 Index: 21986_d = $55E2_h$
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	60000	**100 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02590

Parameter | Name: **C02590 | Brake opening time**			Data type: UNSIGNED_32 Index: 21985_d = $55E1_h$
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	60000	**100 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02591

Parameter | Name: **C02591 | Waiting time - state monitoring**			Data type: UNSIGNED_32 Index: 21984_d = $55E0_h$
			▶ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	60000	**100 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02593

Parameter | Name: **C02593 | Waiting time - brake activation**			Data type: UNSIGNED_32 Index: 21982_d = $55DE_h$
			▶ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	s	1000.000	**0.000 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02594

Parameter | Name: **C02594 | Test torque**			Data type: INTEGER_32 Index: 21981_d = $55DD_h$
			▶ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
-21474836.47	Nm	21474836.47	**0.00 Nm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02595

Parameter | Name: **C02595 | Permissible angle of rotation**			Data type: INTEGER_32 Index: 21980_d = $55DC_h$
			▶ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0	°	360	**5 °**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02596

Parameter | Name: **C02596 | Grinding speed**			Data type: INTEGER_32 Index: 21979_d = $55DB_h$
			▶ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0	rpm	300	**100 rpm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02597

Parameter | Name: **C02597 | Accel./decel. time - grinding**			Data type: UNSIGNED_32 Index: 21978_d = $55DA_h$
			▶ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	s	60.000	**1.000 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02598

Parameter | Name: **C02598 | Grinding ON time**			Data type: UNSIGNED_32 Index: 21977_d = $55D9_h$
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.2	s	2.0	**0.5 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02599

Parameter | Name: **C02599 | Grinding OFF time**			Data type: UNSIGNED_32 Index: 21976_d = $55D8_h$
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.2	s	2.0	**0.5 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02600

Parameter | Name: **C02600 | Acceleration time feedf. control**			Data type: UNSIGNED_32 Index: 21975_d = $55D7_h$
From software version V3.0			
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	s	1000.000	**0.000 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02601

Parameter | Name: **C02601 | Ref. for Accel. time of brake**		Data type: UNSIGNED_32 Index: 21974_d = $55D6_h$
From software version V3.0		
		▸ Basic function "Brake control"
Selection list (Lenze setting printed in bold)		
0	**Motor reference value**	
1	Current starting value	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02602

Parameter | Name: **C02602 | Source for feedf. control brake**		Data type: UNSIGNED_32 Index: 21973_d = $55D5_h$
From software version V3.0		
		▸ Basic function "Brake control"
Selection list (Lenze setting printed in bold)		
0	**Torque**	
1	Speed	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02603

Parameter | Name: **C02603 | Threshold 1 for opening brake**			Data type: INTEGER_32 Index: 21972_d = $55D4_h$
From software version V3.0			
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
-50000	rpm	50000	**0 rpm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02604

Parameter | Name: **C02604 | Threshold 2 for opening brake**			Data type: INTEGER_32 Index: 21971_d = $55D3_h$
From software version V3.0			
			▸ Basic function "Brake control"
Setting range (min. value | unit | max. value)			**Lenze setting**
-50000	rpm	50000	**0 rpm**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02607

Parameter | Name: **C02607 | BRK_dnState**			Data type: INTEGER_32 Index: 21968_d = $55D0_h$
Bit coded status of the basic function "Brake control".			
Display range (min. value | unit | max. value)			
-2147483648		2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02608

Parameter | Name: **C02608 | BRK_dnTorqueAdd_n**			Data type: INTEGER_32 Index: 21967_d = $55CF_h$
Display of the additive torque value of the basic function "brake control".			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02609

Parameter | Name: **C02609 | Brake control: Dig. signals**		Data type: UNSIGNED_32 Index: 21966_d = $55CE_h$
Display of the digital signals of the basic function "brake control".		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02609/1		BRK_bReleaseBrake
C02609/2		BRK_bStartingTorque2
C02609/3		BRK_bBrakeApplied
C02609/4		BRK_bBrakeTest
C02609/5		BRK_bBrakeGrindIn
C02609/6		BRK_bReleaseBrakeOut
C02609/7		BRK_bBrakeReleased
C02609/8		BRK_bError
C02609/9		BRK_bCInhActive
C02609/10		BRK_bDisableStop
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02610

Parameter | Name: **C02610 | Deceleration time for stop**			Data type: UNSIGNED_32 Index: 21965_d = $55CD_h$
			▸ Basic function "Stop"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	s	1000.000	**1.000 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02611

Parameter | Name: **C02611 | S-ramp time for stop**			Data type: UNSIGNED_32 Index: 21964_d = $55CC_h$
			▶ Basic function "Stop"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	s	10.000	**0.100 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02612

Parameter | Name: **C02612 | Ref. for decel. time of stop**		Data type: UNSIGNED_32 Index: 21963_d = $55CB_h$
		▶ Basic function "Stop"
Selection list (Lenze setting printed in bold)		
0	**Motor reference speed (C00011)**	
1	Current speed	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02616

Parameter | Name: **C02616 | STP_dnState**			Data type: INTEGER_32 Index: 21959_d = $55C7_h$
Bit coded status of the basic function "Stop".			
Display range (min. value | unit | max. value)			
-2147483648		2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02617

Parameter | Name: **C02617 | STP_bStopActive**		Data type: UNSIGNED_32 Index: 21958_d = $55C6_h$
Status of the basic function "Stop".		
Selection list (display only)		
0	Normal stop not active	
1	Normal stop active	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02619

Parameter | Name: **C02619 | Quick stop: Dig. signals**		Data type: UNSIGNED_32 Index: 21956_d = $55C4_h$
Display of the digital signals of the basic function "Quick stop".		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02619/1		QSP_bActivate1
C02619/2		QSP_bActivate2
C02619/3		QSP_bActivate3
C02619/4		QSP_bActive
C02619/5		QSP_bActivateDCBrake
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02620

Parameter | Name: **C02620 | Manual jog: Speed 1**			Data type: INTEGER_32 Index: 21955_d = $55C3_h$
			▸ Basic function "Manual jog"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s	214748.3647	**360.0000 Unit/s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02621

Parameter | Name: **C02621 | Manual jog: Speed 2**			Data type: INTEGER_32 Index: 21954_d = $55C2_h$
			▸ Basic function "Manual jog"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s	214748.3647	**720.0000 Unit/s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02622

Parameter | Name: **C02622 | Manual jog: Acceleration**			Data type: INTEGER_32 Index: 21953_d = $55C1_h$
			▸ Basic function "Manual jog"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s²	214748.3647	**360.0000 Unit/s²**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02623

Parameter | Name: **C02623 | Manual jog: Deceleration**			Data type: INTEGER_32 Index: 21952_d = $55C0_h$
			▸ Basic function "Manual jog"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s²	214748.3647	**1440.0000 Unit/s²**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02624

Parameter | Name: **C02624 | Manual jog: S-ramp time**			Data type: UNSIGNED_32 Index: 21951_d = $55BF_h$
			▸ Basic function "Manual jog"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	s	10.000	**0.100 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02625

Parameter | Name: **C02625 | Manual jog: Step size**			Data type: INTEGER_32 Index: 21950_d = $55BE_h$
From software version V5.0 Step distance for "Manual jog with step limitation" mode.			
			▸ Basic function "Manual jog"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit	214748.3647	**360.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02626

Parameter | Name: **C02626 | Manual jog:Index Stop position**	Data type: INTEGER_32 Index: 21949_d = $55BD_h$

From software version V5.0
Selection of the breakpoint positions for "Manual jog with breakpoint" mode.
- In connection with a function block instance of type **L_PosPositionerTable**:
  The index [1...75] of the table position in the VTPOS table has to be specified, which contains the intermediate stop position x that is to be used.
- In connection with a function block instance of type **L_PosProfileTable**:
  The index [1...4] of the profile data set in the VTPOS table has to be specified, which contains the intermediate stop position x that is to be used.

▶ Basic function "Manual jog"

Setting range (min. value | unit | max. value)			
0		75	
Subcodes	**Lenze setting**		**Information**
C02626/1	0		Index of the breakpoint positions 1 ... 16
C02626/...			
C02626/16			
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02627

Parameter | Name: **C02627 | Manual jog:Selected Stop position**	Data type: INTEGER_32 Index: 21948_d = $55BC_h$

From software version V5.0
Display of the breakpoint positions selected via C02626/1...16 for "Manual jog with breakpoints".

▶ Basic function "Manual jog"

Display range (min. value | unit | max. value)			
-214748.3648	Unit	214748.3647	
Subcodes			**Information**
C02627/1			Breakpoint position 1 ... 16
C02627/...			
C02627/16			
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02637

Parameter | Name: **C02637 | MAN_dnSpeedOverride_n**	Data type: INTEGER_32 Index: 21938_d = $55B2_h$

From software version V5.0
Display of the speed override for the basic function "Manual jog".

Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02638

Parameter | Name: **C02638 | Manual jog: Status**	Data type: INTEGER_32 Index: 21937_d = $55B1_h$

Status of the basic function "Manual jog".

Display range (min. value | unit | max. value)			
0		8	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02639

Parameter | Name: **C02639 | Manual jog: Dig. signals**	Data type: UNSIGNED_32 Index: 21936_d = $55B0_h$
Display of the digital signals of the basic function "Manual jog".	
Selection list (display only)	
0 FALSE	
1 TRUE	
Subcodes	**Information**
C02639/1	MAN_bEnable
C02639/2	MAN_bJogPositive
C02639/3	MAN_bJogNegative
C02639/4	MAN_bActivateJogSpeed2
C02639/5	MAN_bReleaseLimitSwitch
C02639/6	MAN_bEnabled
C02639/7	MAN_bActive
C02639/8	MAN_bStepMode
C02639/9	MAN_bIntermediateStopMode
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02640

Parameter | Name: **C02640 | Homing mode**	Data type: UNSIGNED_32 Index: 21935_d = $55AF_h$

Selection of the way homing is to be carried out.

▸ Basic function "Homing"

Selection list (Lenze setting printed in bold)		Information
0	cw_Rn_TP	Positive direction - via home mark - to TP ▸ Process description
1	ccw_Rn_TP	Negative direction - via home mark - to TP ▸ Process description
2	cw_Lp_ccw_Rn_TP	Pos. direction - reversing to limit switch - via home mark - to TP ▸ Process description
3	ccw_Ln_cw_Rn_TP	Neg. direction - reversing to limit switch - via home mark - to TP ▸ Process description
4	cw_Rp_ccw_Rn_TP	Pos. direction - reversing to home mark - to TP ▸ Process description
5	ccw_Rp_cw_Rn_TP	Neg. direction - reversing to home mark - to TP ▸ Process description
8	cw_TP	Positive direction to touch probe ▸ Process description
9	ccw_TP	Negative direction to touch probe ▸ Process description
10	cw_Lp_ccw_TP	Pos. direction - reversing to limit switch - to TP ▸ Process description
11	ccw_Ln_cw_TP	Neg. direction - reversing to limit switch - to TP ▸ Process description
12	cw_Lp	Positive direction to limit switch ▸ Process description
13	ccw_Ln	Negative direction to limit switch ▸ Process description
14	cw_Trq_Lim	Positive direction to torque limit ▸ Process description
15	ccw_Trq_Lim	Negative direction to torque limit ▸ Process description
100	**Set home pos. directly**	Set home pos. directly ▸ Process description
1001	DS402 homing method 01	From software version V3.0 also the homing methods in accordance with DS402 are provided. ▸ Overview of DS402 homing modes

Parameter | Name: **C02640 | Homing mode**		Data type: UNSIGNED_32 Index: 21935_d = $55AF_h$
1002	DS402 homing method 02	
1003	DS402 homing method 03	
1004	DS402 homing method 04	
1005	DS402 homing method 05	
1006	DS402 homing method 06	
1007	DS402 homing method 07	
1008	DS402 homing method 08	
1009	DS402 homing method 09	
1010	DS402 homing method 10	
1011	DS402 homing method 11	
1012	DS402 homing method 12	
1013	DS402 homing method 13	
1014	DS402 homing method 14	
1015	DS402 homing method 15	
1016	DS402 homing method 16	
1017	DS402 homing method 17	
1018	DS402 homing method 18	
1019	DS402 homing method 19	
1020	DS402 homing method 20	
1021	DS402 homing method 21	
1022	DS402 homing method 22	
1023	DS402 homing method 23	
1024	DS402 homing method 24	
1025	DS402 homing method 25	
1026	DS402 homing method 26	
1027	DS402 homing method 27	
1028	DS402 homing method 28	
1029	DS402 homing method 29	
1030	DS402 homing method 30	
1031	DS402 homing method 31	
1032	DS402 homing method 32	
1033	DS402 homing method 33	
1034	DS402 homing method 34	
1035	DS402 homing method 35	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02641

Parameter | Name: **C02641 | Action after detect Home position**		Data type: UNSIGNED_32 Index: 21934_d = $55AE_h$
From software version V4.0		
		▸ Basic function "Homing"
Selection list (Lenze setting printed in bold)		
0	**Move absolute on target position**	
1	Move relative by Target position	
2	Stop immediately	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02642

Parameter | Name: **C02642 | Home position**			Data type: INTEGER_32 Index: 21933_d = $55AD_h$
			▶ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
-214748.3647	Unit	214748.3647	**0.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02643

Parameter | Name: **C02643 | Homing: Target position**			Data type: INTEGER_32 Index: 21932_d = $55AC_h$
			▶ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
-214748.3647	Unit	214748.3647	**0.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02644

Parameter | Name: **C02644 | Homing: Speed 1**			Data type: INTEGER_32 Index: 21931_d = $55AB_h$
			▶ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s	214748.3647	**360.0000 Unit/s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02645

Parameter | Name: **C02645 | Homing: Acceleration 1**			Data type: INTEGER_32 Index: 21930_d = $55AA_h$
			▶ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s²	214748.3647	**720.0000 Unit/s²**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02646

Parameter | Name: **C02646 | Homing: Speed 2**			Data type: INTEGER_32 Index: 21929_d = $55A9_h$
			▶ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s	214748.3647	**0.0000 Unit/s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02647

Parameter | Name: **C02647 | Homing: Acceleration 2**			Data type: INTEGER_32 Index: 21928_d = $55A8_h$
			▶ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s²	214748.3647	**360.0000 Unit/s²**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02648

Parameter | Name: **C02648 | Homing: S-ramp time**			Data type: INTEGER_32 Index: 21927_d = $55A7_h$
			▸ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	10000	**100 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02649

Parameter | Name: **C02649 | Homing: Torque limit**			Data type: INTEGER_32 Index: 21926_d = $55A6_h$
			▸ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.00	%	200.00	**10.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02650

Parameter | Name: **C02650 | Homing: Blocking time**			Data type: UNSIGNED_32 Index: 21925_d = $55A5_h$
			▸ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.000	s	120.000	**1.000 s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1000

C02651

Parameter | Name: **C02651 | Homing: TP configuration**			Data type: UNSIGNED_32 Index: 21924_d = $55A4_h$
			▸ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
0		4294967295	**16**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02652

Parameter | Name: **C02652 | Ref. pos. after mains switching**		Data type: UNSIGNED_32 Index: 21923_d = $55A3_h$
		▸ Basic function "Homing"
Selection list (Lenze setting printed in bold)		
0	**Delete**	
1	Retain	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02653

Parameter | Name: **C02653 | Max. rot. ang. aft. mns. swtch.**			Data type: INTEGER_32 Index: 21922_d = $55A2_h$
			▸ Basic function "Homing"
Setting range (min. value | unit | max. value)			**Lenze setting**
0	°	1000000	**180 °**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02655

Parameter | Name: **C02655 | HM_dnSpeedOverride_n**			Data type: INTEGER_32 Index: 21920_d = $55A0_h$
From software version V5.0			
Display of the speed override for the basic function "Homing".			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02656

Parameter | Name: **C02656 | Actual position (homing)**			Data type: INTEGER_32 Index: 21919_d = $559F_h$
			▸ Basic function "Homing"
Display range (min. value | unit | max. value)			
-214748.3647	Unit	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02657

Parameter | Name: **C02657 | HM_dnState**			Data type: INTEGER_32 Index: 21918_d = $559E_h$
Bit coded status of the basic function "Homing".			
Display range (min. value | unit | max. value)			
-2147483648		2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02658

Parameter | Name: **C02658 | HM_dnHomePos_p**			Data type: INTEGER_32 Index: 21917_d = $559D_h$
Display of the *HM_dnHomePos_p* input signal of the basic function "Homing".			
Display range (min. value | unit | max. value)			
-214748.3647		214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02659

Parameter | Name: **C02659 | Homing: Dig. signals**		Data type: UNSIGNED_32 Index: 21916_d = $559C_h$
Display of the digital signals of the basic function "Homing".		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02659/1		HM_bEnable
C02659/2		HM_bActivateHoming
C02659/3		HM_bHomingMark
C02659/4		HM_bLoadHomePos
C02659/5		HM_bResetHomePos
C02659/6		HM_bEnabled
C02659/7		HM_bActive
C02659/8		HM_bDone
C02659/9		HM_bHomePosAvailable
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02670

Parameter | Name: **C02670 | Tolerance for POS_bActPosInTarget**			Data type: INTEGER_32 Index: 21905_d = 5591_h
From software version V1.5 Tolerance window for actual value-based evaluation "Target position reached" (Output *POS_bActPosInTarget*) ▶ Basic function "Positioning"			
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit	214748.3647	**0.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02671

Parameter | Name: **C02671 | Tolerance for POS_bDriveInTarget**			Data type: INTEGER_32 Index: 21904_d = 5590_h
From software version V5.0 Tolerance window for actual value and setpoint-based evaluation "Drive in the target" (Output *POS_bDriveInTarget*) ▶ Basic function "Positioning"			
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0001	Unit	214748.3647	**2.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02672

Parameter | Name: **C02672 | Hysteresis for POS_bDriveInTarget**			Data type: INTEGER_32 Index: 21903_d = $558F_h$
From software version V5.0 Hysteresis window for actual value and setpoint-based evaluation "Drive in the target" (Output *POS_bDriveInTarget*) ▶ Basic function "Positioning"			
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit	214748.3647	**1.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02673

Parameter | Name: **C02673 | Activate DriveInTarget Modulo**		Data type: UNSIGNED_32 Index: 21902_d = $558E_h$
From software version V5.0 For actual value and setpoint-based evaluation "Drive in target" (output *POS_bDriveInTarget*): Definition how the modulo evaluation is to be carried out if the actual position value enters the tolerance and hysteresis window again. ▶ Basic function "Positioning"		
Selection list (Lenze setting printed in bold)		
0	Only setpoint Cycle	
1	**All cycles**	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02674

Parameter | Name: **C02674 | POS_dwActualProfileNumber**			Data type: UNSIGNED_32 Index: 21901_d = $558D_h$
Current profile of the basic function "Positioning".			
Display range (min. value | unit | max. value)			
0		1000	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02675

Parameter | Name: **C02675 | POS_dnState**			Data type: INTEGER_32 Index: 21900$_d$ = 558C$_h$
Bit coded status of the basic function "Positioning".			
Display range (min. value | unit | max. value)			
-2147483648		2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02676

Parameter | Name: **C02676 | POS_dnProfileSpeed_s**			Data type: INTEGER_32 Index: 21899$_d$ = 558B$_h$
Display of the max. speed of the current profile of the basic function "Positioning".			
Display range (min. value | unit | max. value)			
-214748.3647		214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02677

Parameter | Name: **C02677 | Positioning: % signals**			Data type: INTEGER_32 Index: 21898$_d$ = 558A$_h$
Display of the scaled signals of the basic function "Positioning".			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes			**Information**
C02677/1			POS_dnSpeedOverride_n
C02677/2			POS_dnAccOverride_n
C02677/3			POS_dnDecOverride_n
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02678

Parameter | Name: **C02678 | Positioning: Pos. signals**			Data type: INTEGER_32 Index: 21897$_d$ = 5589$_h$
Display of the position signals of the basic function "Positioning".			
Display range (min. value | unit | max. value)			
-214748.3647	Unit	214748.3647	
Subcodes			**Information**
C02678/1			POS_dnSetPos_p
C02678/2			POS_dnProfileTarget_p
C02678/3			POS_dnActPosRelative_p
C02678/4			POS_dnSetPosRelative_p
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02679

Parameter | Name: **C02679 | Positioning: Dig. signals**	Data type: UNSIGNED_32 Index: 21896_d = 5588_h
Display of the digital signals of the basic function "Positioning".	
Selection list (display only)	
0 FALSE	
1 TRUE	
Subcodes	**Information**
C02679/1	POS_bEnable
C02679/2	POS_bStart
C02679/3	POS_bAbort
C02679/4	POS_bRestart
C02679/5	POS_bEnableOverride
C02679/6	POS_bDisableTP
C02679/7	POS_bEnabled
C02679/8	POS_bActive
C02679/9	POS_bDone
C02679/10	POS_bInTarget
C02679/11	POS_bActPosInTarget
C02679/12	POS_bDriveInTarget
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02680

Parameter | Name: **C02680 | Source position setpoint**	Data type: UNSIGNED_32 Index: 21895_d = 5587_h
	▸ Basic function "Positioning"
Selection list (Lenze setting printed in bold)	
0 Position setpoint input	
1 From add. speed	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02681

Parameter | Name: **C02681 | Source add. speed**	Data type: UNSIGNED_32 Index: 21894_d = 5586_h
	▸ Basic function "Positioning"
Selection list (Lenze setting printed in bold)	
0 Add. speed input	
1 From position setpoint	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02685

Parameter | Name: **C02685 | PF_dnMotorAcc_x**	Data type: INTEGER_32 Index: 21890_d = 5582_h
Display of the motor acceleration of the basic function "Position follower".	
Display range (min. value | unit | max. value)	
-7680000.0 | | 7680000.0	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 10

C02686

Parameter | Name: **C02686 | PF_dnSpeedAdd1_s**			Data type: INTEGER_32 Index: 21889_d = 5581_h
Display of the speed feedforward control value of the basic function "Position follower".			
Display range (min. value | unit | max. value)			
-480000.0	rpm	480000.0	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02687

Parameter | Name: **C02687 | Position follower: % signals**			Data type: INTEGER_32 Index: 21888_d = 5580_h
Display of the scaled signals of the basic function "Position follower".			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes			**Information**
C02687/1			PF_dnSpeedAdd2_n
C02687/2			PF_dnTorqueAdd_n
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02688

Parameter | Name: **C02688 | PF_dnPositionSet_p**			Data type: INTEGER_32 Index: 21887_d = $557F_h$
Display of the position signals of the basic function "Position follower".			
Display range (min. value | unit | max. value)			
-214748.3648	Revolution	214748.3647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02689

Parameter | Name: **C02689 | Position follower: Dig. signals**		Data type: UNSIGNED_32 Index: 21886_d = $557E_h$
Display of the digital signals of the basic function "Position follower".		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02689/1		PF_bEnable
C02689/2		PF_bEnabled
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02692

Parameter | Name: **C02692 | SF_dnMotorAcc_x**			Data type: INTEGER_32 Index: 21883_d = $557B_h$
Display of the motor acceleration of the basic function "Speed follower".			
Display range (min. value | unit | max. value)			
-7680000.0		7680000.0	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02693

Parameter | Name: **C02693 | SF_dnSpeedAdd_s**			Data type: INTEGER_32 Index: 21882_d = $557A_h$
Display of the additive speed setpoint of the basic function "Speed follower".			
Display range (min. value | unit | max. value)			
-480000.0	rpm	480000.0	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02694

Parameter | Name: **C02694 | Speed follower: % signals**			Data type: INTEGER_32 Index: 21881_d = 5579_h
Display of the scaled signals of the basic function "Speed follower".			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes			**Information**
C02694/1			SF_dnSpeedSet_n
C02694/2			SF_dnTorqueAdd_n
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02695

Parameter | Name: **C02695 | Speed follower: Dig. signals**		Data type: UNSIGNED_32 Index: 21880_d = 5578_h
Display of the digital signals of the basic function "Speed follower".		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02695/1		SF_bEnable
C02695/2		SF_bEnabled
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02698

Parameter | Name: **C02698 | Torque follower: % signals**			Data type: INTEGER_32 Index: 21877_d = 5575_h
Display of the scaled signals of the basic function "Torque follower".			
Display range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes			**Information**
C02698/1			TF_TorqueSet_n
C02698/2			TF_dnSpeedHighLimit_n
C02698/3			TF_dnSpeedLowLimit_n
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02699

Parameter | Name: **C02699 | Torque follower: Dig. signals**		Data type: UNSIGNED_32 Index: 21876_d = 5574_h
Display of the digital signals of the basic function "Torque follower".		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02699/1		TF_bEnable
C02699/2		TF_bEnabled
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02700

Parameter | Name: **C02700 | Software limits pos. effective**		Data type: UNSIGNED_32 Index: 21875_d = 5573_h
		▸ Basic function "Limiter"
Selection list (Lenze setting printed in bold)		
0	**Deactivated**	
1	Activated	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02701

Parameter | Name: **C02701 | Software limit positions**			Data type: INTEGER_32 Index: 21874_d = 5572_h
			▸ Basic function "Limiter"
Setting range (min. value | unit | max. value)			
-214748.3647	Unit	214748.3647	
Subcodes	**Lenze setting**		**Information**
C02701/1	0.0000 Unit		Positive software limit position
C02701/2	0.0000 Unit		Negative software limit position
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02702

Parameter | Name: **C02702 | Limitations effective**		Data type: UNSIGNED_32 Index: 21873_d = 5571_h
		▸ Basic function "Limiter"
Selection list (Lenze setting printed in bold)		
0	**Deactivated**	
1	Activated	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02703

Parameter | Name: **C02703 | Max. speed**			Data type: INTEGER_32 Index: 21872_d = 5570_h
			▸ Basic function "Limiter"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s	214748.3647	**3600.0000 Unit/s**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02704

Parameter | Name: **C02704 | Max. speed [rpm]**			Data type: INTEGER_32 Index: 21871_d = $556F_h$
			Basic function "Limiter"
Display range (min. value | unit | max. value)			
0.0	rpm	214748364.7	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02705

Parameter | Name: **C02705 | Max. acceleration**			Data type: INTEGER_32 Index: 21870_d = $556E_h$
			Basic function "Limiter"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit/s²	214748.3647	**3600.0000 Unit/s²**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02706

Parameter | Name: **C02706 | Min. S-ramp time**			Data type: UNSIGNED_32 Index: 21869_d = $556D_h$
			Basic function "Limiter"
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	10000	**100 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02707

Parameter | Name: **C02707 | Permissible direction of rot.**		Data type: UNSIGNED_32 Index: 21868_d = $556C_h$
		Basic function "Limiter"
Selection list (Lenze setting printed in bold)		
0	**Positive and negative**	
1	Positive only	
2	Negative only	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02708

Parameter | Name: **C02708 | Limited speed**			Data type: INTEGER_32 Index: 21867_d = $556B_h$
			Basic function "Limiter"
Setting range (min. value | unit | max. value)			
0.0000	Unit/s	214748.3647	
Subcodes	**Lenze setting**		**Information**
C02708/1	3600.0000 Unit/s		Limited speed 1 ... 4
C02708/2	7200.0000 Unit/s		
C02708/3	14400.0000 Unit/s		
C02708/4	28800.0000 Unit/s		
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02709

Parameter | Name: **C02709 | Limited speed [rpm]**			Data type: INTEGER_32 Index: 21866_d = $556A_h$
			▸ Basic function "Limiter"
Display range (min. value | unit | max. value)			
0.0	rpm	214748364.7	
Subcodes			**Information**
C02709/1			Limited speed 1 ... 4
C02709/2			
C02709/3			
C02709/4			
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02710

Parameter | Name: **C02710 | Delay lim. speed**			Data type: UNSIGNED_32 Index: 21865_d = 5569_h
			▸ Basic function "Limiter"
Setting range (min. value | unit | max. value)			
0.0000	Unit/s²	214748.3647	
Subcodes	**Lenze setting**		**Information**
C02710/1	0.0100 Unit/s²		Delays for limited speed 1 ... 4
C02710/2	0.0100 Unit/s²		
C02710/3	0.0100 Unit/s²		
C02710/4	0.0100 Unit/s²		
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02711

Parameter | Name: **C02711 | S-ramp time lim. speed**			Data type: UNSIGNED_32 Index: 21864_d = 5568_h
			▸ Basic function "Limiter"
Setting range (min. value | unit | max. value)			
0	ms	10000	
Subcodes	**Lenze setting**		**Information**
C02711/1	100 ms		S-ramp times for limited speed 1 ... 4
C02711/2	100 ms		
C02711/3	100 ms		
C02711/4	100 ms		
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02712

Parameter | Name: **C02712 | Decel. time lim. speed**			Data type: UNSIGNED_32 Index: 21863_d = 5567_h
			▸ Basic function "Limiter"
Display range (min. value | unit | max. value)			
0	ms	10000	
Subcodes			**Information**
C02712/1			Deceleration times for limited speed 1 ... 4
C02712/2			
C02712/3			
C02712/4			
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02713

Parameter | Name: **C02713 | Max. dist. manual control**			Data type: UNSIGNED_32 Index: 21862_d = 5566_h
			▶ Basic function "Limiter"
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	Unit	214748.3647	**360.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10000

C02714

Parameter | Name: **C02714 | Max. dist. manual jog [inc.]**			Data type: UNSIGNED_32 Index: 21861_d = 5565_h
			▶ Basic function "Limiter"
Display range (min. value | unit | max. value)			
0	Incr.	2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02715

Parameter | Name: **C02715 | Limitation active**		Data type: UNSIGNED_32 Index: 21860_d = 5564_h
		▶ Basic function "Limiter"
Selection list (display only)		
0	Deactivated	
1	Activated	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02716

Parameter | Name: **C02716 | Resp. to limitation**		Data type: UNSIGNED_32 Index: 21859_d = 5563_h
		▶ Basic function "Limiter"
Selection list		
1	Fault	
2	Trouble	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
6	Information	
0	No response	
Subcodes	**Lenze setting**	**Information**
C02716/1	6: Information	Resp. to rotation limitation
C02716/2	3: Quick stop by trouble	Resp. to SW lim. pos. exceeded
C02716/3	6: Information	Resp. to max. value exceeded
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02717

Parameter | Name: **C02717 | LIM_dwControl**			Data type: UNSIGNED_32 Index: 21858_d = 5562_h
Bit coded control word of the basic function "Limiter".			
Display range (min. value | unit | max. value)			
0		4294967295	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02718

Parameter | Name: **C02718 | LIM_dnState**			Data type: INTEGER_32 Index: 21857_d = 5561_h
Status of the basic function "Limiter".			
Display range (min. value | unit | max. value)			
0		1	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02719

Parameter | Name: **C02719 | Limiter: Dig. signals**		Data type: UNSIGNED_32 Index: 21856_d = 5560_h
Display of the digital input signals of the basic function "Limiter".		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02719/1		LIM_bLimitSwitchPositive
C02719/2		LIM_bLimitSwitchNegative
C02719/3		LIM_bActivateLimitedSpeed1
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02720

Parameter | Name: **C02720 | observation software limit positions**		Data type: UNSIGNED_32 Index: 21855_d = $555F_h$
From software version V4.0		
		▶ Basic function "Limiter"
Selection list (Lenze setting printed in bold)		
0	**Based on set value**	
1	Based on set and actual value	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02730

Parameter | Name: **C02730 | Analog inputs: Gain**			Data type: INTEGER_32 Index: 21845_d = 5555_h
Setting range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes	**Lenze setting**		**Information**
C02730/1	100.00 %		Gain - analog input 1
C02730/2	100.00 %		Analog input 2: Gain
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02731

Parameter | Name: **C02731 | Analog inputs: Offset**			Data type: INTEGER_32 Index: 21844_d = 5554_h
Setting range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes	**Lenze setting**		**Information**
C02731/1	0.00 %		Offset - analog input 1
C02731/2	0.00 %		Analog input 2: Offset
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02732

Parameter | Name: **C02732 | Analog inputs: Dead band**			Data type: INTEGER_32 Index: 21843$_d$ = 5553$_h$
Setting range (min. value | unit | max. value)			
0.00	%	100.00	
Subcodes	**Lenze setting**		**Information**
C02732/1	0.00 %		Dead band - analog input 1
C02732/2	0.00 %		Analog input 2: Dead band
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02733

Parameter | Name: **C02733 | Analog outputs: Gain**			Data type: INTEGER_32 Index: 21842$_d$ = 5552$_h$
Setting range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes	**Lenze setting**		**Information**
C02733/1	100.00 %		Gain - analog output 1
C02733/2	100.00 %		Analog output 2: Gain
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02734

Parameter | Name: **C02734 | Analog outputs: Offset**			Data type: INTEGER_32 Index: 21841$_d$ = 5551$_h$
Setting range (min. value | unit | max. value)			
-200.00	%	200.00	
Subcodes	**Lenze setting**		**Information**
C02734/1	0.00 %		Offset - analog output 1
C02734/2	0.00 %		Analog output 2: Offset
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02760

Parameter | Name: **C02760 | Activate encoder**		Data type: UNSIGNED_32 Index: 21815$_d$ = 5537$_h$
From software version V7.0		
		▸ Encoder evaluation: Provision of the encoder signal of input X8
Selection list (Lenze setting printed in bold)		
0	**Deactivated**	
1	Activated	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02761

Parameter | Name: **C02761 | Multiturn resolution**			Data type: UNSIGNED_32 Index: 21814$_d$ = 5536$_h$
From software version V7.0			
			▸ Encoder evaluation: Provision of the encoder signal of input X8
Display range (min. value | unit | max. value)			
0	Rev	2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02762

Parameter | Name: **C02762 | Encoderpos**			Data type: INTEGER_32 Index: $21813_d = 5535_h$
From software version V7.0			
			▸ Encoder evaluation: Provision of the encoder signal of input X8
Display range (min. value | unit | max. value)			
-2147483647	Steps	2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02763

Parameter | Name: **C02763 | Encoderrev**			Data type: INTEGER_32 Index: $21812_d = 5534_h$
From software version V7.0			
			▸ Encoder evaluation: Provision of the encoder signal of input X8
Display range (min. value | unit | max. value)			
-2147483647	Steps	2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02764

Parameter | Name: **C02764 | Encoderspeed**			Data type: INTEGER_32 Index: $21811_d = 5533_h$
From software version V7.0			
			▸ Encoder evaluation: Provision of the encoder signal of input X8
Display range (min. value | unit | max. value)			
-214748364.7	rpm	214748364.7	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02765

Parameter | Name: **C02765 | Enc_bError**		Data type: UNSIGNED_32 Index: $21810_d = 5532_h$
From software version V7.0		
		▸ Encoder evaluation: Provision of the encoder signal of input X8
Selection list (display only)		
0	FALSE	
1	TRUE	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02770

Parameter | Name: **C02770 | Operating mode**		Data type: UNSIGNED_32 Index: $21805_d = 552D_h$
From software version V7.0		
		▸ Basic function "Manual jog open loop"
Selection list (Lenze setting printed in bold)		
0	Deactivate	
1	Activate	
Subcodes	**Lenze setting**	**Information**
C02770/1	0: Deactivate	EnableManualMode
C02770/2	0: Deactivate	JogPositive
C02770/3	0: Deactivate	JogNegative
C02770/4	0: Deactivate	SelectTab1
C02770/5	0: Deactivate	SelectTab2
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02771

Parameter | Name: **C02771 | Frequency**			Data type: INTEGER_32 Index: 21804_d = $552C_h$
From software version V7.0			
			▸ Basic function "Manual jog open loop"
Setting range (min. value | unit | max. value)			
0.0	Hz	1000.0	
Subcodes	**Lenze setting**		**Information**
C02771/1	1.0 Hz		Frequency 1
C02771/2	1.0 Hz		Frequency 2
C02771/3	1.0 Hz		Frequency 3
C02771/4	1.0 Hz		Frequency 4
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02772

Parameter | Name: **C02772 | Starting angle**			Data type: INTEGER_32 Index: 21803_d = $552B_h$
From software version V7.0			
			▸ Basic function "Manual jog open loop"
Setting range (min. value | unit | max. value)			
-180.0	°	180.0	
Subcodes	**Lenze setting**		**Information**
C02772/1	0.0 °		Startangle 1
C02772/2	0.0 °		Startangle 2
C02772/3	0.0 °		Startangle 3
C02772/4	0.0 °		Startangle 4
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02773

Parameter | Name: **C02773 | Current**			Data type: INTEGER_32 Index: 21802_d = $552A_h$
From software version V7.0 100 % ½ I_{max_device} (C00022)			
			▸ Basic function "Manual jog open loop"
Setting range (min. value | unit | max. value)			
0.00	%	100.00	
Subcodes	**Lenze setting**		**Information**
C02773/1	10.00 %		Current 1
C02773/2	10.00 %		Current 2
C02773/3	10.00 %		Current 3
C02773/4	10.00 %		Current 4
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02774

Parameter | Name: **C02774 | Acceleration time**	Data type: INTEGER_32 Index: 21801_d = 5529_h

From software version V7.0

▸ Basic function "Manual jog open loop"

Setting range (min. value | unit | max. value)			
0.001	s	2147483.647	
Subcodes	**Lenze setting**		**Information**
C02774/1	1.000 s		Acceleration time 1
C02774/2	1.000 s		Acceleration time 2
C02774/3	1.000 s		Acceleration time 3
C02774/4	1.000 s		Acceleration time 4

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1000

C02775

Parameter | Name: **C02775 | Deceleration time**	Data type: INTEGER_32 Index: 21800_d = 5528_h

From software version V7.0

▸ Basic function "Manual jog open loop"

Setting range (min. value | unit | max. value)			
0.001	s	2147483.647	
Subcodes	**Lenze setting**		**Information**
C02775/1	1.000 s		Deceleration time 1
C02775/2	1.000 s		Deceleration time 2
C02775/3	1.000 s		Deceleration time 3
C02775/4	1.000 s		Deceleration time 4

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1000

C02776

Parameter | Name: **C02776 | Time**	Data type: INTEGER_32 Index: 21799_d = 5527_h

From software version V7.0

▸ Basic function "Manual jog open loop"

Setting range (min. value | unit | max. value)			
0.001	s	2147483.647	
Subcodes	**Lenze setting**		**Information**
C02776/1	1.000 s		Max. activation time 1
C02776/2	1.000 s		Max. activation time 2
C02776/3	1.000 s		Max. activation time 3
C02776/4	1.000 s		Max. activation time 4

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1000

C02779

Parameter | Name: **C02779 | MOL_SetpointCurrent**	Data type: UNSIGNED_32 Index: 21796_d = 5524_h

From software version V7.0
Maximum current of the selected profile parameter set.

▸ Basic function "Manual jog open loop"

Display range (min. value | unit | max. value)		
0.00	A	42949672.95

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 100

C02780

Parameter | Name: **C02780 | MOL_dnState**			Data type: INTEGER_32 Index: 21795_d = 5523_h
From software version V7.0 Status of the basic function "Manual jog open loop".			
Display range **(min. value | unit | max. value)**			
-2147483648		2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02781

Parameter | Name: **C02781 | ManualJogOpenLoop: Dig. signals**		Data type: UNSIGNED_32 Index: 21794_d = 5522_h
From software version V7.0 Display of the digital signals of the basic function "Manual jog open loop".		
Selection list **(display only)**		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02781/1		MOL_bEnable
C02781/2		MOL_bJogPositive
C02781/3		MOL_bJogNegative
C02781/4		MOL_bSelectTab1
C02781/5		MOL_bSelectTab2
C02781/6		MOL_bEnabled
C02781/7		MOL_bActive
C02781/8		MOL_bDone
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02785

Parameter | Name: **C02785 | PPI activation**		Data type: UNSIGNED_32 Index: 21790_d = $551E_h$
From software version V7.0 ▸ Basic function "pole position identification"		
Selection list **(Lenze setting printed in bold)**		
0	**PPI disabled**	
1	PPI active	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02786

Parameter | Name: **C02786 | PPI mode**		Data type: UNSIGNED_32 Index: 21789_d = $551D_h$
From software version V7.0 ▸ Basic function "pole position identification"		
Selection list **(Lenze setting printed in bold)**		
0	**360°**	
1	<20°	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02787

Parameter | Name: **C02787 | PPI_dnState**			Data type: INTEGER_32 Index: 21788_d = $551C_h$
From software version V7.0 Status of the basic function "pole position identification".			
Display range (min. value | unit | max. value)			
-2147483648		2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02788

Parameter | Name: **C02788 | PolePosition setpoint**			Data type: INTEGER_32 Index: 21787_d = $551B_h$
From software version V7.0			▶ Basic function "pole position identification"
Display range (min. value | unit | max. value)			
-214748364.8	°	214748364.7	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02789

Parameter | Name: **C02789 | PolePositionIdentification: Dig. signals**		Data type: UNSIGNED_32 Index: 21786_d = $551A_h$
From software version V7.0 Display of the digital signals of the basic function "pole position identification".		
Selection list (display only)		
0	FALSE	
1	TRUE	
Subcodes		**Information**
C02789/1		PPI_bEnable
C02789/2		PPI_bStart
C02789/3		PPI_bLoadPolePosition
C02789/4		PPI_bResetPolePosition
C02789/5		PPI_bEnabled
C02789/6		PPI_bActive
C02789/7		PPI_bDone
C02789/8		PPI_bError
C02789/9		PPI_bPolePositionAvailable
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02800

Parameter | Name: **C02800 | Analog input x: Input signal**			Data type: INTEGER_16 Index: 21775_d = $550F_h$
Scaling: -16384 ≡ -100 %, +16383 ≡ +100 %			
Display range (min. value | unit | max. value)			
-16384		16383	
Subcodes			**Information**
C02800/1			Input signal - analog input 1
C02800/2			Analog input 2: Input signal
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02801

Parameter | Name: **C02801 | Analog output x: Output signal**	Data type: INTEGER_16 Index: 21774_d = $550E_h$
Scaling: -16384 ≡ -100 %, +16383 ≡ +100 %	
Display range (min. value | unit | max. value)	
-16384 | | 16383	
Subcodes	**Information**
C02801/1	Output signal - analog output 1
C02801/2	Analog output 2: Output signal
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02802

Parameter	Name: **C02802	Status word: Digital outputs**		Data type: BITFIELD_32 Index: 21773_d = $550D_h$

Display of the hexadecimal value of the digital outputs
- Important: All digital levels are indicated without considering the level logic. Internal signals are displayed as well.

Display area		
0x00000000		0xFFFFFFFF

Value is bit-coded:	
Bit 0	Dig. output. 1: Terminal state
Bit 1	Dig. output. 2: Terminal state
Bit 2	Dig. output. 3: Terminal state
Bit 3	Dig. output. 4: Terminal state
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
Bit16	Reserved
Bit 17	Reserved
Bit 18	Reserved
Bit 19	Reserved
Bit 20	Reserved
Bit 21	Reserved
Bit 22	Reserved
Bit 23	Reserved
Bit 24	Reserved
Bit 25	Reserved
Bit 26	Reserved
Bit 27	Reserved
Bit 28	Reserved
Bit 29	Reserved
Bit 30	Reserved
Bit 31	Reserved

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02803

Parameter | Name: **C02803 | Status word: Digital inputs**		Data type: BITFIELD_32 Index: $21772_d = 550C_h$
Display of the hexadecimal value of the digital inputs • Important: All digital levels are indicated without considering the level logic. Internal signals are displayed as well.		
Display area		
0x00000000		0xFFFFFFFF
Value is bit-coded:		
Bit 0	Dig. input. 1: Terminal state	
Bit 1	Dig. input. 2: Terminal state	
Bit 2	Dig. input. 3: Terminal state	
Bit 3	Dig. input. 4: Terminal state	
Bit 4	Dig. input. 5: Terminal state	
Bit 5	Dig. input. 6: Terminal state	
Bit 6	Dig. input. 7: Terminal state	
Bit 7	Dig. input. 8: Terminal state	
Bit 8	Reserved	
Bit 9	Reserved	
Bit 10	Reserved	
Bit 11	Reserved	
Bit 12	Reserved	
Bit 13	Reserved	
Bit 14	Reserved	
Bit 15	Reserved	
Bit16	Reserved	
Bit 17	Reserved	
Bit 18	Reserved	
Bit 19	Reserved	
Bit 20	Reserved	
Bit 21	Reserved	
Bit 22	Reserved	
Bit 23	Reserved	
Bit 24	Reserved	
Bit 25	Reserved	
Bit 26	Reserved	
Bit 27	Reserved	
Bit 28	Reserved	
Bit 29	Reserved	
Bit 30	Reserved	
Bit 31	Reserved	

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02810

Parameter | Name: **C02810 | Touch probe x: Delay time**		Data type: UNSIGNED_32 Index: 21765_d = 5505_h

The delay time set will be considered when the position is determined at the time of touch probe and will be used to compensate for dead times, if necessary.
- Please observe the setting of the input filter for the digital inputs (C02830).

Setting range (min. value | unit | max. value)		
0	µs	7000

Subcodes	Lenze setting	Information
C02810/1	0 µs	TP1 (DI1): Delay time
C02810/2	0 µs	TP2 (DI2):Delay time
C02810/3	0 µs	TP3 (DI3):Delay time
C02810/4	0 µs	TP4 (DI4):Delay time
C02810/5	0 µs	TP5 (DI5):Delay time
C02810/6	0 µs	TP6 (DI6): Delay time
C02810/7	0 µs	TP7 (DI7): Delay time
C02810/8	0 µs	TP8 (DI8): Delay time
C02810/9	0 µs	TPM (motor encoder): Delay time
C02810/10	0 µs	TPL (pos. encoder): Delay time

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02830

Parameter | Name: **C02830 | Digital inputs: Delay time**		Data type: UNSIGNED_8 Index: 21745_d = $54F1_h$

Input filter for digital inputs
- Can be used to filter out "spikes" at the digital inputs, if necessary.
- Each digital input is assigned to a subcode.
- Since the filter is a "counting" filter, the indicated times are only approximate values.

Selection list (Lenze setting printed in bold)		Information
0	2 µs	Filter time
1	4 µs	
2	8 µs	
3	16 µs	
4	32 µs	
5	64 µs	
6	128 µs	
7	256 µs	
8	512 µs	
9	1024 µs	
10	2048 µs	
11	4096 µs	
12	8192 µs	
13	16384 µs	
14	32768 µs	

Subcodes	Lenze setting	Information
C02830/1	0: 2 µs	Setting for digital input 1 ... 8
C02830/...		
C02830/8		

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02850

Parameter | Name: **C02850 | Service code**	Data type: UNSIGNED_32 Index: 21725_d = $54DD_h$
This code is used internally by the controller and must not be overwritten by the user!	

C02851

Parameter | Name: **C02851 | Service code**	Data type: UNSIGNED_32 Index: 21724_d = $54DC_h$
This code is used internally by the controller and must not be overwritten by the user!	

C02852

Parameter | Name: **C02852 | Service code**	Data type: UNSIGNED_16 Index: 21723_d = $54DB_h$
This code is used internally by the controller and must not be overwritten by the user!	

C02853

Parameter | Name: **C02853 | Lss sat. characteristic**		Data type: UNSIGNED_16 Index: 21722_d = $54DA_h$
Setting range (min. value | unit | max. value)		
0	%	400
Subcodes	**Lenze setting**	**Information**
C02853/1	100 %	Saturation characteristic to correct the leakage inductance and the current controller parameters. • The saturation characteristic is defined by 17 interpolation points which are distributed linearly on the x axis. • Interpolation point 17 represents 100 % of the maximum motor current in the process (C02855). • The values to be entered in the subcodes represent the y values of the interpolation points 1 ... 17. ▸ Correction of the leakage inductance via saturation characteristic
C02853/2	100 %	
C02853/3	100 %	
C02853/4	100 %	
C02853/5	100 %	
C02853/6	100 %	
C02853/7	100 %	
C02853/8	100 %	
C02853/9	100 %	
C02853/10	100 %	
C02853/11	100 %	
C02853/12	100 %	
C02853/13	100 %	
C02853/14	100 %	
C02853/15	100 %	
C02853/16	100 %	
C02853/17	100 %	
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02854

Parameter | Name: **C02854 | Service code**	Data type: UNSIGNED_32 Index: 21721_d = $54D9_h$
This code is used internally by the controller and must not be overwritten by the user!	

C02855

Parameter | Name: **C02855 | Imax f. Lss sat. characteristic**			Data type: UNSIGNED_32 Index: 21720_d = $54D8_h$

Maximum motor current in the process
- Defines the interpolation point 17 of the saturation characteristic set in C02853.

▶ Correction of the leakage inductance via saturation characteristic

Setting range (min. value | unit | max. value)			Lenze setting
0.0	A	6000.0	**5.4 A**
☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 10

C02856

Parameter | Name: **C02856 | Service code**	Data type: VISIBLE_STRING Index: 21719_d = $54D7_h$

This code is used internally by the controller and must not be overwritten by the user!

C02857

Parameter | Name: **C02857 | Service code**	Data type: VISIBLE_STRING Index: 21718_d = $54D6_h$

This code is used internally by the controller and must not be overwritten by the user!

C02858

Parameter | Name: **C02858 | Electronic nameplate status**	Data type: UNSIGNED_8 Index: 21717_d = $54D5_h$

This code is used internally by the controller and must not be overwritten by the user!

C02859

Parameter | Name: **C02859 | Activate Lss sat. char.**	Data type: UNSIGNED_8 Index: 21716_d = $54D4_h$

▶ Correction of the leakage inductance via saturation characteristic

Selection list (Lenze setting printed in bold)	
0	**Off**
1	ON

☑ Read access ☑ Write access ☑ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02860

Parameter | Name: **C02860 | Rr adjustment**			Data type: UNSIGNED_32 Index: 21715_d = $54D3_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
50.00	%	200.00	**100.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02861

Parameter | Name: **C02861 | Lh adjustment**			Data type: UNSIGNED_32 Index: 21714_d = $54D2_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
50.00	%	200.00	**100.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02862

Parameter | Name: **C02862 | Service code**			Data type: UNSIGNED_16 Index: 21713$_d$ = 54D1$_h$
From software version V7.0 ▶ Resolver error compensation			
Setting range (min. value | unit | max. value)			
0		100	
Subcodes	**Lenze setting**		**Information**
C02862/1	100		Gain of cosine track
C02862/2	100		Gain of sine track
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02863

Parameter | Name: **C02863 | Service code**			Data type: INTEGER_16 Index: 21712$_d$ = 54D0$_h$
From software version V7.0 ▶ Resolver error compensation			
Setting range (min. value | unit | max. value)			**Lenze setting**
-100		100	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02864

Parameter | Name: **C02864 | Service code**			Data type: INTEGER_32 Index: 21711$_d$ = 54CF$_h$
Display range (min. value | unit | max. value)			
-2147483647		2147483647	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02865

Parameter | Name: **C02865 | Adaptation of Ur**			Data type: UNSIGNED_32 Index: 21710$_d$ = 54CE$_h$
Setting range (min. value | unit | max. value)			**Lenze setting**
50.00	%	200.00	**100.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 100

C02866

Parameter | Name: **C02866 | Curr. control par. of C75 C76**	Data type: UNSIGNED_8 Index: 21709$_d$ = 54CD$_h$
This code is used internally by the controller and must not be overwritten by the user!	

C02900

Parameter | Name: **C02900 | User Password**	Data type: VISIBLE_STRING Index: 21675$_d$ = 54AB$_h$

From software version V3.0

If the cam data are provided with a user password, the defined user password must be entered once to execute the following actions:

- Changing the cam data via parameter setting
- Loading/saving the cam data

Validity

The user password entered is maintained until the next download, mains switching, or until reset by the user (logout).

- You can "logout" deliberately by entering an invalid password.

▶ Basic function "Cam data management"

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer

C02901

Parameter | Name: **C02901 | CamMemory**	Data type: UNSIGNED_32 Index: 21674_d = $54AA_h$
From software version V5.0	
	▶ Basic function "Cam data management": Memory mapping
Display range (min. value | unit | max. value)	
0 | | 4294967295	
Subcodes	**Information**
C02901/1	Size of the memory for quick download to RAM
C02901/2	Size of the memory for "Online change"
C02901/3	Size of the memory from which the cam data are processed.
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02902

Parameter | Name: **C02902 | Timestamp**	Data type: UNSIGNED_32 Index: 21673_d = $54A9_h$
From software version V3.0	
	▶ Basic function "Cam data management"
Display range (min. value | unit | max. value)	
0 | | 4294967295	
Subcodes	**Information**
C02902/1	Time stamp of the cam data in the controller
C02902/2	Time stamp of the cam data which are currently being processed in the controller.
C02902/3	Time stamp of the cam data in the controller which have already been converted into the internal format.
C02902/4	Time stamp of the cam data in the memory module
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02903

Parameter | Name: **C02903 | GUID**	Data type: OCTET_STRING Index: 21672_d = $54A8_h$
From software version V3.0	
	▶ Basic function "Cam data management"
Display range (min. value | unit | max. value)	
Subcodes	**Information**
C02903/1	GUID of the cam data in the controller
C02903/2	GUID of the cam data which are currently being processed in the controller.
C02903/3	GUID of the cam data in the controller which have already been converted into the internal format.
C02903/4	GUID of the cam data in the memory module
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C02905

Parameter | Name: **C02905 | Online Change Mode**	Data type: UNSIGNED_32 Index: 21670_d = $54A6_h$

From software version V3.0

▸ Basic function "Cam data management": Online change mode

Selection list (Lenze setting printed in bold)	
10	Manual activation
15	Automatic activation
16	**Automatic activation with controller inhibit**

☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02906

Parameter | Name: **C02906 | Online Change State**	Data type: UNSIGNED_32 Index: 21669_d = $54A5_h$

From software version V3.0

▸ Basic function "Cam data management": Online change mode

Selection list (display only)	
0	Ready
5	Initialisation
7	Saving is active
8	Loading is active
11	Waiting for controlled acceptance

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02908

Parameter | Name: **C02908 | Product Count**	Data type: UNSIGNED_32 Index: 21667_d = $54A3_h$

From software version V3.0

Display of the highest product number +1 of the cam data currently being processed

▸ Basic function "Cam data management"

Display range (min. value | unit | max. value)		
0		0

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02909

Parameter | Name: **C02909 | Active Product**	Data type: UNSIGNED_32 Index: 21666_d = $54A2_h$

From software version V3.0

Display of the product number of the active product of the cam data currently being processed

▸ Basic function "Cam data management"

Display range (min. value | unit | max. value)		
0		0

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02910

Parameter | Name: **C02910 | Product Name**	Data type: VISIBLE_STRING Index: 21665_d = $54A1_h$

From software version V3.0

▸ Basic function "Cam data management"

☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C02911

Parameter | Name: **C02911 | Product Choice**			Data type: UNSIGNED_32 Index: 21664_d = $54A0_h$
From software version V3.0			
			Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
0		47	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 1

C02912

Parameter | Name: **C02912 | Number Of Products**			Data type: UNSIGNED_32 Index: 21663_d = $549F_h$
From software version V3.0			
			Changing cam data via parameterisation
Display range (min. value | unit | max. value)			
0		65536	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02919

Parameter | Name: **C02919 | Number Of Cam Tracks**			Data type: UNSIGNED_32 Index: 21656_d = 5498_h
From software version V3.0			
			Changing cam data via parameterisation
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02920

Parameter | Name: **C02920 | Cam Track Choice**			Data type: UNSIGNED_32 Index: 21655_d = 5497_h
From software version V3.0			
			Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
0		65535	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 1

C02921

Parameter | Name: **C02921 | Cam Track Type**		Data type: UNSIGNED_32 Index: 21654_d = 5496_h
From software version V3.0		
		Changing cam data via parameterisation
Selection list (display only)		
1	Linear	
5	Spline	
11	LinearPC	
15	SplinePC	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02922

Parameter | Name: C02922 | Number Of Cam Data Points			Data type: UNSIGNED_32 Index: 21653_d = 5495_h
From software version V3.0			
			Changing cam data via parameterisation
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02923

Parameter | Name: C02923 | Cam Data Point Choice			Data type: UNSIGNED_32 Index: 21652_d = 5494_h
From software version V3.0			
			Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
0		65535	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 1

C02924

Parameter | Name: C02924 | Change Cam Data Point X			Data type: INTEGER_32 Index: 21651_d = 5493_h
From software version V3.0			
			Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
-214748.3647	Unit	214748.3647	**0.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 10000

C02925

Parameter | Name: C02925 | Change Cam Data Point Y			Data type: INTEGER_32 Index: 21650_d = 5492_h
From software version V3.0			
			Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
-214748.3647	Unit	214748.3647	**0.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 10000

C02926

Parameter | Name: C02926 | Change Cam Data Point M			Data type: INTEGER_32 Index: 21649_d = 5491_h
From software version V3.0			
			Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
-200.00	%	200.00	**0.00 %**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 100

C02927

Parameter | Name: C02927 | Auto Inc Cam Data Points		Data type: UNSIGNED_32 Index: 21648_d = 5490_h
From software version V3.0		
		Changing cam data via parameterisation
Selection list (Lenze setting printed in bold)		
0	**Deactivate**	
1	Activate	
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer		Scaling factor: 1

C02939

Parameter | Name: **C02939 | Number Of Cont Tracks**			Data type: UNSIGNED_32 Index: 21636_d = 5484_h
From software version V3.0			
			Changing cam data via parameterisation
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02940

Parameter | Name: **C02940 | Cont Track Choice**			Data type: UNSIGNED_32 Index: 21635_d = 5483_h
From software version V3.0			
			Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
0		65535	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 1

C02941

Parameter | Name: **C02941 | Cont Type**		Data type: UNSIGNED_32 Index: 21634_d = 5482_h
From software version V3.0		
		Changing cam data via parameterisation
Selection list (display only)		
1	Pos. position cam	
2	Neg. position cam	
3	Bidirect. position cam	
11	Pos. time cam	
12	Neg. time cam	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C02942

Parameter | Name: **C02942 | Number Of Cont Data Points**			Data type: UNSIGNED_32 Index: 21633_d = 5481_h
From software version V3.0			
			Changing cam data via parameterisation
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02943

Parameter | Name: **C02943 | Cont Data Point Choice**			Data type: UNSIGNED_32 Index: 21632_d = 5480_h
From software version V3.0			
			Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
0		65535	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 1

C02944

Parameter | Name: **C02944 | Cont Pos X0**			Data type: INTEGER_32 Index: 21631_d = $547F_h$
From software version V3.0			
			▸ Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
-214748.3647	Unit	214748.3647	**0.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 10000

C02945

Parameter | Name: **C02945 | Cont Pos X1**			Data type: INTEGER_32 Index: 21630_d = $547E_h$
From software version V3.0			
			▸ Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
-214748.3647	Unit	214748.3647	**0.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 10000

C02946

Parameter | Name: **C02946 | Cont Time**			Data type: UNSIGNED_32 Index: 21629_d = $547D_h$
From software version V3.0			
			▸ Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
0.0000	ms	214748.3647	**0.0000 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 10000

C02959

Parameter | Name: **C02959 | Number of Pos Tracks**			Data type: UNSIGNED_32 Index: 21616_d = 5470_h
From software version V3.0			
			▸ Changing cam data via parameterisation
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02960

Parameter | Name: **C02960 | Pos Track Choice**			Data type: UNSIGNED_32 Index: 21615_d = $546F_h$
From software version V3.0			
			▸ Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
0		65535	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 1

C02962

Parameter | Name: **C02962 | Number of Pos Data Points**			Data type: UNSIGNED_32 Index: 21613_d = $546D_h$
From software version V3.0			
			▸ Changing cam data via parameterisation
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C02963

Parameter | Name: **C02963 | Pos Data Point Choice**			Data type: UNSIGNED_32 Index: 21612_d = $546C_h$
From software version V3.0			
			▸ Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
0		65535	**0**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 1

C02964

Parameter | Name: **C02964 | Change Pos Data Point X**			Data type: INTEGER_32 Index: 21611_d = $546B_h$
From software version V3.0			
			▸ Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
-214748.3647	Unit	214748.3647	**0.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 10000

C02965

Parameter | Name: **C02965 | Change Pos Data Point Y**			Data type: INTEGER_32 Index: 21610_d = $546A_h$
From software version V3.0			
			▸ Changing cam data via parameterisation
Setting range (min. value | unit | max. value)			**Lenze setting**
-214748.3647	Unit	214748.3647	**0.0000 Unit**
☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☑ No transfer			Scaling factor: 10000

C02996

Parameter | Name: **C02996 | Service code**	Data type: UNSIGNED_32 Index: 21579_d = $544B_h$
This code is used internally by the controller and must not be overwritten by the user!	

C02997

Parameter | Name: **C02997 | Service code**	Data type: UNSIGNED_32 Index: 21578_d = $544A_h$
This code is used internally by the controller and must not be overwritten by the user!	

C02998

Parameter | Name: **C02998 | Service code**	Data type: UNSIGNED_32 Index: 21577_d = 5449_h
This code is used internally by the controller and must not be overwritten by the user!	

C02999

Parameter | Name: **C02999 | Service code**	Data type: UNSIGNED_32 Index: 21576_d = 5448_h
This code is used internally by the controller and must not be overwritten by the user!	

14.3 Attribute table

The Attribute table contains information required for communicating with the controller via parameters.

How to read the table of attributes:

Column		Meaning	Entry	
Code		Parameter designation	Cxxxxx	
Name		Short parameter text (display text)	*Text*	
Index	dec	Index under which the parameter is addressed. The subindex of array variables corresponds to the Lenze subcode number.	24575 - Lenze code number	Only required for access via 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	BITFIELD_8	1 byte bit-coded
			BITFIELD_16	2 bytes bit-coded
			BITFIELD_32	4 bytes bit-coded
			INTEGER_8	1 byte with sign
			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	ASCII string
	Factor	Factor for data transmission via 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
Access	R	Read access	☑ Reading allowed	
	W	Write access	☑ Writing allowed	
	CINH	Controller inhibit required	☑ Writing only possible when controller is inhibited	

Attribute table

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00002	Device commands	24573	5FFD	E	1	UNSIGNED_32	1	☑	☑	
C00003	Device command status	24572	5FFC	E	1	UNSIGNED_32	1	☑		
C00004	Service password	24571	5FFB	E	1	UNSIGNED_32	1	☑	☑	
C00005	Application selection	24570	5FFA	E	1	INTEGER_32	1	☑	☑	
C00006	Select motor control	24569	5FF9	E	1	UNSIGNED_32	1	☑	☑	☑
C00007	Active application	24568	5FF8	E	1	INTEGER_32	1	☑		
C00008	Device command progress	24567	5FF7	E	1	UNSIGNED_32	1	☑		
C00011	Motor reference speed	24564	5FF4	E	1	UNSIGNED_32	1	☑	☑	
C00018	Switching frequency	24557	5FED	E	1	UNSIGNED_32	1	☑	☑	
C00019	Threshold - standstill recognition	24556	5FEC	E	1	UNSIGNED_32	1	☑	☑	
C00022	Maximum current	24553	5FE9	E	1	UNSIGNED_32	100	☑	☑	
C00034	Config. analog input 1	24541	5FDD	E	1	UNSIGNED_32	1	☑	☑	
C00050	Speed setpoint [rpm]	24525	5FCD	A	2	INTEGER_32	1	☑		
C00051	Actual speed [rpm]	24524	5FCC	E	1	INTEGER_32	1	☑		
C00052	Motor voltage	24523	5FCB	E	1	UNSIGNED_32	1	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00053	DC-bus voltage	24522	5FCA	E	1	UNSIGNED_32	1	☑		
C00054	Motor current	24521	5FC9	E	1	UNSIGNED_32	100	☑		
C00055	Phase currents	24520	5FC8	A	4	INTEGER_32	100	☑		
C00056	Torque setpoint	24519	5FC7	E	1	INTEGER_32	100	☑		
C00057	Torque	24518	5FC6	A	2	UNSIGNED_32	1000	☑		
C00058	Pole position	24517	5FC5	A	3	INTEGER_32	10	☑	☑	
C00059	Motor - number of pole pairs	24516	5FC4	E	1	UNSIGNED_32	1	☑		
C00060	Rotor position	24515	5FC3	E	1	INTEGER_32	1	☑		
C00061	Heatsink temperature	24514	5FC2	E	1	INTEGER_32	1	☑		
C00062	Temperature inside the controller	24513	5FC1	E	1	INTEGER_32	1	☑		
C00063	Motor temperature	24512	5FC0	E	1	INTEGER_32	1	☑		
C00064	Device utilisation (Ixt)	24511	5FBF	E	1	UNSIGNED_32	1	☑		
C00065	Ext. 24-V voltage	24510	5FBE	E	1	INTEGER_32	10	☑		
C00066	Thermal motor load (I^2xt)	24509	5FBD	E	1	UNSIGNED_32	1	☑		
C00068	Capacitor temperature	24507	5FBB	E	1	INTEGER_32	1	☑		
C00069	CPU temperature	24506	5FBA	E	1	INTEGER_32	1	☑		
C00070	Speed controller gain	24505	5FB9	E	1	UNSIGNED_32	100000	☑	☑	
C00071	Speed controller reset time	24504	5FB8	E	1	UNSIGNED_32	10	☑	☑	
C00072	Speed controller rate time	24503	5FB7	E	1	UNSIGNED_32	100	☑	☑	
C00074	Feedfwd. ctrl. - current contr.	24501	5FB5	E	1	UNSIGNED_8	1	☑	☑	
C00075	Current controller gain	24500	5FB4	E	1	UNSIGNED_32	100	☑	☑	
C00076	Current contr. reset time	24499	5FB3	E	1	UNSIGNED_32	100	☑	☑	
C00077	Field controller gain	24498	5FB2	E	1	UNSIGNED_32	100	☑	☑	
C00078	Field contr. reset time	24497	5FB1	E	1	UNSIGNED_32	10	☑	☑	
C00079	Mutual motor inductance	24496	5FB0	E	1	UNSIGNED_32	10	☑		
C00080	Resolver - number of pole pairs	24495	5FAF	E	1	UNSIGNED_32	1	☑	☑	☑
C00081	Rated motor power	24494	5FAE	E	1	UNSIGNED_32	100	☑	☑	☑
C00082	Motor rotor resistance	24493	5FAD	E	1	UNSIGNED_32	10000	☑		
C00083	Motor - rotor time constant	24492	5FAC	E	1	UNSIGNED_32	100	☑		
C00084	Motor stator resistance	24491	5FAB	E	1	UNSIGNED_32	10000	☑	☑	☑
C00085	Motor stator leakage induct.	24490	5FAA	E	1	UNSIGNED_32	1000	☑	☑	☑
C00087	Rated motor speed	24488	5FA8	E	1	UNSIGNED_32	1	☑	☑	☑
C00088	Rated motor current	24487	5FA7	E	1	UNSIGNED_32	100	☑	☑	☑
C00089	Rated motor frequency	24486	5FA6	E	1	UNSIGNED_32	10	☑	☑	☑
C00090	Rated motor voltage	24485	5FA5	E	1	UNSIGNED_32	1	☑	☑	☑
C00091	Motor - cosine phi	24484	5FA4	E	1	UNSIGNED_32	100	☑	☑	☑
C00092	Motor magnetising current	24483	5FA3	E	1	UNSIGNED_32	100	☑		
C00093	Field weakening for SM	24482	5FA2	E	1	UNSIGNED_32	1	☑	☑	
C00099	Firmware version	24476	5F9C	E	1	VISIBLE_STRING	1	☑		
C00100	Resol. of an encoder revolution	24475	5F9B	E	1	UNSIGNED_32	1	☑	☑	☑
C00105	Quick stop decel. time	24470	5F96	E	1	UNSIGNED_32	1000	☑	☑	
C00106	Quick stop S-ramp time	24469	5F95	E	1	UNSIGNED_32	100	☑	☑	
C00107	Ref. for quick stop dec. time	24468	5F94	E	1	UNSIGNED_8	1	☑	☑	
C00114	Digital inp. x - terminal pol.	24461	5F8D	A	8	UNSIGNED_8	1	☑	☑	
C00118	Dig. output x: terminal pol.	24457	5F89	A	4	UNSIGNED_8	1	☑	☑	
C00120	Motor overload protection (I^2xt)	24455	5F87	E	1	UNSIGNED_32	1	☑	☑	
C00121	Motor temp. warning threshold	24454	5F86	E	1	UNSIGNED_32	1	☑	☑	
C00122	Heatsink temp. warn. threshold	24453	5F85	E	1	UNSIGNED_32	1	☑	☑	
C00123	Warning threshold device util.	24452	5F84	E	1	UNSIGNED_32	1	☑	☑	
C00126	CPU temp. warning threshold	24449	5F81	E	1	UNSIGNED_32	1	☑	☑	
C00127	Mot. overload warning threshold	24448	5F80	E	1	UNSIGNED_32	1	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00128	Therm. motor time constant	24447	5F7F	A	2	UNSIGNED_32	10	☑	☑	
C00129	Brake resistor value	24446	5F7E	E	1	INTEGER_32	10	☑	☑	
C00130	Rated power of brake resistor	24445	5F7D	E	1	INTEGER_32	1	☑	☑	
C00131	Rated quantity of heat for brake res.	24444	5F7C	E	1	INTEGER_32	1	☑	☑	
C00133	Ref.: Brake chopper utilisation	24442	5F7A	E	1	UNSIGNED_8	1	☑	☑	
C00134	Min. brake resistance	24441	5F79	E	1	INTEGER_32	10	☑		
C00137	Brake transistor utilisation	24438	5F76	E	1	INTEGER_32	1	☑		
C00138	Brake resistor utilisation	24437	5F75	E	1	INTEGER_32	1	☑		
C00142	Autom. restart after mains ON	24433	5F71	E	1	UNSIGNED_32	1	☑	☑	
C00150	Status word device control 1	24425	5F69	E	1	BITFIELD_16	1	☑		
C00155	Status word device control 2	24420	5F64	E	1	BITFIELD_16	1	☑		
C00156	Status/Control word MCTRL	24419	5F63	A	2	UNSIGNED_32	1	☑		
C00158	Controller inhibit by (source)	24417	5F61	E	1	BITFIELD_16	1	☑		
C00159	Quick stop by (source)	24416	5F60	E	1	BITFIELD_16	1	☑		
C00162	Masked error number	24413	5F5D	A	3	UNSIGNED_32	1	☑		
C00166	Error description	24409	5F59	E	1	VISIBLE_STRING	1	☑		
C00168	Error number	24407	5F57	E	1	UNSIGNED_32	1	☑		
C00169	Logbook event filter	24406	5F56	E	1	BITFIELD_32	1	☑	☑	
C00173	Mains voltage	24402	5F52	E	1	UNSIGNED_8	1	☑	☑	
C00174	Undervoltage (LU) threshold	24401	5F51	E	1	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	Service code	24395	5F4B	E	1	VISIBLE_STRING		☑	☑	
C00181	Red. brake chopper threshold	24394	5F4A	E	1	UNSIGNED_32	1	☑	☑	
C00183	Device state	24392	5F48	E	1	UNSIGNED_32	1	☑		
C00185	Mains recov. detect. threshold	24390	5F46	E	1	UNSIGNED_32	1	☑	☑	
C00186	ENP: Identified motor type	24389	5F45	E	1	VISIBLE_STRING	1	☑		
C00187	ENP: Identified serial number	24388	5F44	E	1	VISIBLE_STRING	1	☑		
C00188	ENP: Status	24387	5F43	E	1	UNSIGNED_8	1	☑		
C00199	Device name	24376	5F38	E	1	VISIBLE_STRING	1	☑	☑	
C00200	Firmware product type	24375	5F37	E	1	VISIBLE_STRING	1	☑		
C00201	Firmware compilation date	24374	5F36	E	1	VISIBLE_STRING	1	☑		
C00202	Autom. ENP data transfer	24373	5F35	E	1	UNSIGNED_32	1	☑	☑	
C00203	HW product types	24372	5F34	A	9	VISIBLE_STRING	1	☑		
C00204	HW serial numbers	24371	5F33	A	9	VISIBLE_STRING	1	☑		
C00205	HW descriptions	24370	5F32	A	6	VISIBLE_STRING	1	☑		
C00206	HW manufacturing data	24369	5F31	A	8	VISIBLE_STRING	1	☑		
C00208	HW manufacturer	24367	5F2F	A	6	VISIBLE_STRING	1	☑		
C00209	HW countries of origin	24366	5F2E	A	6	VISIBLE_STRING	1	☑		
C00210	HW versions	24365	5F2D	A	6	VISIBLE_STRING	1	☑		
C00211	Application: Version	24364	5F2C	E	1	VISIBLE_STRING	1	☑		
C00212	Application: Type code	24363	5F2B	E	1	VISIBLE_STRING	1	☑		
C00213	Application: compilation date	24362	5F2A	E	1	VISIBLE_STRING	1	☑		
C00214	Required safety module	24361	5F29	E	1	UNSIGNED_8	1	☑	☑	☑
C00218	Application: ID number	24357	5F25	E	1	UNSIGNED_32	1	☑		
C00227	Behav. at parameter set changeover	24348	5F1C	A	2	UNSIGNED_32	1	☑	☑	
C00254	Phase controller gain	24321	5F01	E	1	UNSIGNED_32	100	☑	☑	
C00270	Freq. - current setpoint filter	24305	5EF1	A	2	UNSIGNED_32	10	☑	☑	
C00271	Width - current setp. filter	24304	5EF0	A	2	UNSIGNED_32	10	☑	☑	
C00272	Depth - current setp. filter	24303	5EEF	A	2	UNSIGNED_32	1	☑	☑	
C00273	Moment of inertia	24302	5EEE	A	2	UNSIGNED_32	100	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00274	Max. acceleration change	24301	5EED	E	1	UNSIGNED_32	10	☑	☑	
C00275	Signal source - speed setpoint	24300	5EEC	E	1	UNSIGNED_16	1	☑	☑	
C00276	Signal source - torque setpoint	24299	5EEB	E	1	UNSIGNED_16	1	☑	☑	
C00280	Filter time const. DC detection	24295	5EE7	E	1	UNSIGNED_32	10	☑	☑	
C00281	Filter for PWM adjustment	24294	5EE6	E	1	UNSIGNED_8	1	☑	☑	
C00311	CAN TPDO1 mask byte x	24264	5EC8	A	8	BITFIELD_8	1	☑	☑	
C00312	CAN TPDO2 mask byte x	24263	5EC7	A	8	BITFIELD_8	1	☑	☑	
C00313	CAN TPDO3 mask byte x	24262	5EC6	A	8	BITFIELD_8	1	☑	☑	
C00314	CAN TPDO4 mask byte x	24261	5EC5	A	8	BITFIELD_8	1	☑	☑	
C00320	CAN TPDOx identifier	24255	5EBF	A	4	BITFIELD_32	1	☑	☑	
C00321	CAN RPDOx identifier	24254	5EBE	A	4	BITFIELD_32	1	☑	☑	
C00322	CAN TPDOx Tx mode	24253	5EBD	A	4	UNSIGNED_8	1	☑	☑	
C00323	CAN RPDOx Rx mode	24252	5EBC	A	4	UNSIGNED_8	1	☑	☑	
C00324	CAN TPDOx delay time	24251	5EBB	A	4	UNSIGNED_16	1	☑	☑	
C00343	CAN TPDO counter	24232	5EA8	A	4	UNSIGNED_32	1	☑		
C00344	CAN RPDO counter	24231	5EA7	A	4	UNSIGNED_32	1	☑		
C00345	CAN error	24230	5EA6	E	1	UNSIGNED_8	1	☑		
C00346	CAN heartbeat activity	24229	5EA5	E	1	BITFIELD_32	1	☑		
C00347	CAN heartbeat status	24228	5EA4	A	32	UNSIGNED_8	1	☑		
C00348	CAN status DIP switch	24227	5EA3	E	1	UNSIGNED_8	1	☑		
C00349	CAN setting of DIP switch	24226	5EA2	A	2	UNSIGNED_8	1	☑		
C00350	CAN node address	24225	5EA1	E	1	UNSIGNED_8	1	☑	☑	
C00351	CAN baud rate	24224	5EA0	E	1	UNSIGNED_8	1	☑	☑	
C00352	CAN slave/master	24223	5E9F	E	1	UNSIGNED_8	1	☑	☑	
C00356	CAN TPDOx cycle time	24219	5E9B	A	4	UNSIGNED_16	1	☑	☑	
C00357	CAN RPDOx monitoring time	24218	5E9A	A	4	UNSIGNED_16	1	☑	☑	
C00359	CAN status	24216	5E98	E	1	UNSIGNED_8	1	☑		
C00360	CAN telegram and error counter	24215	5E97	A	8	UNSIGNED_16	1	☑		
C00361	CAN bus load	24214	5E96	A	6	UNSIGNED_32	1	☑		
C00367	CAN SYNC Rx identifier	24208	5E90	E	1	UNSIGNED_32	1	☑	☑	
C00368	CAN SYNC Tx identifier	24207	5E8F	E	1	UNSIGNED_32	1	☑	☑	
C00369	CAN sync transmission cycle time	24206	5E8E	A	3	UNSIGNED_16	1	☑	☑	
C00372	CAN SDO server Rx identifier	24203	5E8B	A	10	BITFIELD_32	1	☑	☑	
C00373	CAN SDO server Tx identifier	24202	5E8A	A	10	BITFIELD_32	1	☑	☑	
C00374	CAN SDO client node address	24201	5E89	A	10	UNSIGNED_8	1	☑	☑	
C00375	CAN SDO client Rx identifier	24200	5E88	A	10	BITFIELD_32	1	☑	☑	
C00376	CAN SDO client Tx identifier	24199	5E87	A	10	BITFIELD_32	1	☑	☑	
C00377	CAN SDO server node address	24198	5E86	A	10	UNSIGNED_8	1	☑	☑	
C00378	CAN delay boot-up - operational	24197	5E85	E	1	UNSIGNED_16	1	☑	☑	
C00381	CAN Heartbeat Producer Time	24194	5E82	E	1	UNSIGNED_16	1	☑	☑	
C00382	CAN guard time	24193	5E81	E	1	UNSIGNED_16	1	☑	☑	
C00383	CAN Life Time Factor	24192	5E80	E	1	UNSIGNED_8	1	☑	☑	
C00385	CAN Heartbeat Consumer Time	24190	5E7E	A	32	BITFIELD_32	1	☑	☑	
C00386	CAN Node Guarding	24189	5E7D	A	32	BITFIELD_32	1	☑	☑	
C00387	CAN Node Guarding Activity	24188	5E7C	E	1	BITFIELD_32	1	☑		
C00388	CAN node guarding status	24187	5E7B	A	32	UNSIGNED_8	1	☑		
C00390	CAN Error Register (DS301V402)	24185	5E79	E	1	BITFIELD_8	1	☑		
C00391	CAN Emergency Object	24184	5E78	E	1	BITFIELD_32	1	☑	☑	
C00392	CAN emergency delay time	24183	5E77	E	1	UNSIGNED_16	1	☑	☑	
C00393	CAN result - bus scan	24182	5E76	A	128	UNSIGNED_8	1	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00394	CAN Predefined Error Field (DS301V402)	24181	5E75	A	10	UNSIGNED_32	1	☑		
C00398	Test mode motor control	24177	5E71	E	1	UNSIGNED_32	1	☑	☑	☑
C00399	Settings for test mode	24176	5E70	A	2	INTEGER_32	10	☑	☑	
C00413	Hiperface: detected TypeCode	24162	5E62	E	1	UNSIGNED_32	1	☑		
C00414	Hiperface: TypeCode	24161	5E61	E	1	UNSIGNED_32	1	☑	☑	
C00415	Hiperface: number of rev.	24160	5E60	E	1	UNSIGNED_32	1	☑	☑	
C00417	Dynamic of resolver evaluation	24158	5E5E	E	1	UNSIGNED_32	1	☑	☑	
C00418	Activate resolver error compensation	24157	5E5D	E	1	UNSIGNED_8	1	☑	☑	
C00420	Encoder - number of increments	24155	5E5B	E	1	UNSIGNED_16	1	☑	☑	☑
C00421	Encoder voltage	24154	5E5A	E	1	UNSIGNED_16	10	☑	☑	☑
C00422	Encoder type	24153	5E59	E	1	UNSIGNED_16	1	☑	☑	☑
C00423	SSI encoder: Bit rate	24152	5E58	E	1	UNSIGNED_32	1	☑	☑	☑
C00424	Ssi-encoder: Data word length	24151	5E57	E	1	UNSIGNED_32	1	☑	☑	☑
C00427	TTL encoder signal evaluation	24148	5E54	E	1	UNSIGNED_16	1	☑	☑	☑
C00435	SSI encoder: Partword starting position	24140	5E4C	A	8	UNSIGNED_8	1	☑	☑	
C00436	SSI encoder: Partword length	24139	5E4B	A	8	UNSIGNED_8	1	☑	☑	
C00437	SSI encoder: Partword data coding	24138	5E4A	A	8	UNSIGNED_8	1	☑	☑	
C00443	Status: Digital inputs	24132	5E44	A	12	UNSIGNED_8	1	☑		
C00444	Status: Digital outputs	24131	5E43	A	18	UNSIGNED_8	1	☑		
C00464	Keypad: Mode	24111	5E2F	E	1	UNSIGNED_16	1	☑	☑	
C00465	Keypad: Welcome screen time-out	24110	5E2E	E	1	UNSIGNED_8	1	☑	☑	
C00466	Keypad: Default parameters	24109	5E2D	E	1	UNSIGNED_16	1	☑	☑	
C00467	Keypad: Default welcome screen	24108	5E2C	E	1	UNSIGNED_8	1	☑	☑	
C00469	Keypad: Fct. STOP key	24106	5E2A	E	1	UNSIGNED_8	1	☑	☑	
C00490	Position encoder selection	24085	5E15	E	1	UNSIGNED_16	1	☑	☑	☑
C00494	Motor standstill time constant	24081	5E11	E	1	UNSIGNED_32	1	☑	☑	
C00495	Motor encoder selection	24080	5E10	E	1	UNSIGNED_16	1	☑	☑	☑
C00497	Act. speed value time constant	24078	5E0E	E	1	UNSIGNED_32	10	☑	☑	
C00569	Resp. brake trans. Ixt > C00570	24006	5DC6	E	1	UNSIGNED_32	1	☑	☑	
C00570	Warning thres. brake transistor	24005	5DC5	E	1	UNSIGNED_32	1	☑	☑	
C00571	Resp. brake res. i²xt > C00572	24004	5DC4	E	1	UNSIGNED_32	1	☑	☑	
C00572	Warning thres. brake resistor	24003	5DC3	E	1	UNSIGNED_32	1	☑	☑	
C00573	Resp. to overload brake trans.	24002	5DC2	E	1	UNSIGNED_32	1	☑	☑	
C00574	Resp. to overtemp. brake resist.	24001	5DC1	E	1	UNSIGNED_32	1	☑	☑	
C00576	Speed monitoring tolerance	23999	5DBF	E	1	UNSIGNED_32	1	☑	☑	
C00577	Field weakening controller gain	23998	5DBE	E	1	UNSIGNED_32	1000	☑	☑	
C00578	Field weak. contr. reset time	23997	5DBD	E	1	UNSIGNED_32	10	☑	☑	
C00579	Resp. to speed monitoring	23996	5DBC	E	1	UNSIGNED_32	1	☑	☑	
C00580	Resp. to encoder open circuit	23995	5DBB	E	1	UNSIGNED_32	1	☑	☑	
C00581	Resp. to external fault	23994	5DBA	E	1	UNSIGNED_32	1	☑	☑	
C00582	Resp. to heatsink temp. > C00122	23993	5DB9	E	1	UNSIGNED_32	1	☑	☑	
C00583	Resp. to motor KTY overtemp.	23992	5DB8	E	1	UNSIGNED_32	1	☑	☑	
C00584	Resp. to motor temp. > C00121	23991	5DB7	E	1	UNSIGNED_32	1	☑	☑	
C00585	Resp. to motor PTC overtemp.	23990	5DB6	E	1	UNSIGNED_32	1	☑	☑	
C00586	Resp. to resolver open circuit	23989	5DB5	E	1	UNSIGNED_32	1	☑	☑	
C00587	Fan control status	23988	5DB4	E	1	BITFIELD_8	1	☑		
C00588	Resp. to failure t. sensor drive	23987	5DB3	E	1	UNSIGNED_32	1	☑	☑	
C00589	Resp. to CPU temp > C00126	23986	5DB2	E	1	UNSIGNED_32	1	☑	☑	
C00591	Resp. to CAN-RPDOx error	23984	5DB0	A	4	UNSIGNED_8	1	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00594	Reaktion Resp. temp. sensor motor X7/X8	23981	5DAD	E	1	UNSIGNED_32	1	☑	☑	
C00595	Resp. to CAN bus OFF	23980	5DAC	E	1	UNSIGNED_8	1	☑	☑	
C00596	Threshold max. speed reached	23979	5DAB	E	1	UNSIGNED_32	1	☑	☑	
C00597	Resp. to motor phase failure	23978	5DAA	E	1	UNSIGNED_32	1	☑	☑	
C00598	Resp. to open circuit AIN1	23977	5DA9	E	1	UNSIGNED_32	1	☑	☑	
C00599	Motor phase failure threshold	23976	5DA8	E	1	INTEGER_32	10	☑	☑	
C00600	Resp. to DC bus overvoltage	23975	5DA7	E	1	UNSIGNED_32	1	☑	☑	
C00601	Resp. to encoder comm. error	23974	5DA6	E	1	UNSIGNED_32	1	☑	☑	
C00604	Resp. to device overload > C00123	23971	5DA3	E	1	UNSIGNED_32	1	☑	☑	
C00606	Resp. to motor overload > C00127	23969	5DA1	E	1	UNSIGNED_32	1	☑	☑	
C00607	Resp. to max. speed reached	23968	5DA0	E	1	UNSIGNED_32	1	☑	☑	
C00610	Resp. to failure heatsink fan	23965	5D9D	E	1	UNSIGNED_32	1	☑	☑	
C00611	Resp. to failure integral fan	23964	5D9C	E	1	UNSIGNED_32	1	☑	☑	
C00612	Resp. to CAN node guarding error	23963	5D9B	A	32	UNSIGNED_8	1	☑	☑	
C00613	Resp. to CAN heartbeat error	23962	5D9A	A	32	UNSIGNED_8	1	☑	☑	
C00614	Resp. to CAN life guarding error	23961	5D99	E	1	UNSIGNED_8	1	☑	☑	
C00615	Resp. to imp. device config.	23960	5D98	A	5	UNSIGNED_32	1	☑	☑	
C00618	No. of CRC cycles	23957	5D95	E	1	UNSIGNED_32	1	☑		
C00619	Resp. to motor current > C00620	23956	5D94	E	1	UNSIGNED_32	1	☑	☑	
C00620	Ultimate motor current I_ult	23955	5D93	E	1	UNSIGNED_32	10	☑	☑	
C00621	Resp. to encoder pulse deviation	23954	5D92	E	1	UNSIGNED_32	1	☑	☑	
C00625	CAN behaviour in case of fault	23950	5D8E	E	1	UNSIGNED_8	1	☑	☑	
C00635	Resp to new firmw. standard dev.	23940	5D84	E	1	UNSIGNED_32	1	☑	☑	
C00636	Resp. to new module in MXI1	23939	5D83	E	1	UNSIGNED_32	1	☑	☑	
C00637	Resp. to new module in MXI2	23938	5D82	E	1	UNSIGNED_32	1	☑	☑	
C00640	Resp. to pole pos. id. monit.	23935	5D7F	E	1	UNSIGNED_32	1	☑	☑	
C00641	PolePosId 360° current amp.	23934	5D7E	E	1	UNSIGNED_32	1	☑	☑	
C00642	PolePosId 360° ramp time	23933	5D7D	E	1	UNSIGNED_32	1	☑	☑	
C00643	PolePosId 360° travel dir.	23932	5D7C	E	1	UNSIGNED_32	1	☑	☑	
C00644	PolePosId 360° fault tol.	23931	5D7B	E	1	INTEGER_32	10	☑	☑	
C00645	PolePosId 360° absolute cur. amp.	23930	5D7A	E	1	UNSIGNED_32	100	☑		
C00646	PolePosId min.mov. cur. amp.	23929	5D79	E	1	UNSIGNED_32	1	☑	☑	
C00647	PolePosId min.mov. cur.rise rate	23928	5D78	E	1	UNSIGNED_32	1	☑	☑	
C00648	PolePosId min.mov. gain	23927	5D77	E	1	UNSIGNED_32	100	☑	☑	
C00649	PLI min. mov. reset time	23926	5D76	E	1	UNSIGNED_32	100	☑	☑	
C00650	PolePosId min.mov. max.perm.mov.	23925	5D75	E	1	INTEGER_32	1	☑	☑	
C00651	PolePosId min.mov. absolute cur. amp.	23924	5D74	E	1	UNSIGNED_32	100	☑		
C00691	Total speed setpoint	23884	5D4C	E	1	INTEGER_32	100	☑		
C00692	Speed setpoint [%]	23883	5D4B	E	1	INTEGER_32	100	☑		
C00693	Actual speed [%]	23882	5D4A	E	1	INTEGER_32	100	☑		
C00694	Speed controller output	23881	5D49	E	1	INTEGER_32	100	☑		
C00695	Total torque setpoint	23880	5D48	E	1	INTEGER_32	100	☑		
C00696	Torque setpoint [%]	23879	5D47	E	1	INTEGER_32	100	☑		
C00697	Filtered torque setpoint	23878	5D46	E	1	INTEGER_32	100	☑		
C00698	Actual torque [%]	23877	5D45	E	1	INTEGER_32	100	☑		
C00770	MCTRL_dnMotorPosAct	23805	5CFD	A	2	UNSIGNED_32	1	☑		
C00771	MCTRL_dnLoadPosAct	23804	5CFC	A	2	UNSIGNED_32	1	☑		
C00772	MCTRL_dnMotorSpeedAct	23803	5CFB	E	1	INTEGER_32	1	☑		
C00773	MCTRL_dnLoadSpeedAct	23802	5CFA	E	1	INTEGER_32	1	☑		
C00774	MCTRL_dnTorqueAct	23801	5CF9	E	1	INTEGER_32	100	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00775	MCTRL_dnOutputSpeedCtrl	23800	5CF8	E	1	INTEGER_32	100	☑		
C00776	MCTRL_dnInputJerkCtrl	23799	5CF7	E	1	INTEGER_32	100	☑		
C00777	MCTRL_dnInputTorqueCtrl	23798	5CF6	E	1	INTEGER_32	100	☑		
C00778	MCTRL_dnFluxAct	23797	5CF5	E	1	INTEGER_32	100	☑		
C00779	MCTRL_dnDCBusVoltage	23796	5CF4	E	1	INTEGER_32	1	☑		
C00780	MCTRL_dnImotAct	23795	5CF3	E	1	INTEGER_32	100	☑		
C00781	MCTRL_dwMaxMotorSpeed	23794	5CF2	E	1	UNSIGNED_32	1	☑		
C00782	MCTRL_dwMaxMotorTorque	23793	5CF1	E	1	UNSIGNED_32	1000	☑		
C00783	MCTRL_dwMotorVoltageAct	23792	5CF0	E	1	UNSIGNED_32	1	☑		
C00784	MCTRL_dnMotorFreqAct	23791	5CEF	E	1	INTEGER_32	10	☑		
C00786	MCTRL_dnIxtLoad	23789	5CED	E	1	INTEGER_32	100	☑		
C00787	MCTRL_dnFlyingSpeedAct	23788	5CEC	E	1	INTEGER_32	1	☑		
C00788	MCTRL_dwMaxEffMotorTorque	23787	5CEB	E	1	INTEGER_32	1000	☑		
C00789	MCTRL_dwMaxDeviceCurrent	23786	5CEA	E	1	INTEGER_32	100	☑		
C00790	MCTRL_dnI2xtLoad	23785	5CE9	E	1	INTEGER_32	100	☑		
C00791	MCTRL_dnDeltaMotorPos_p	23784	5CE8	E	1	INTEGER_32	1	☑		
C00792	MCTRL_dnOutputPosCtrlMotor_s	23783	5CE7	E	1	INTEGER_32	1	☑		
C00800	MCTRL_dnPosSet	23775	5CDF	A	2	UNSIGNED_32	1	☑		
C00802	MCTRL_dnSpeedAdd	23773	5CDD	E	1	INTEGER_32	1	☑		
C00803	MCTRL_dnTorqueAdd	23772	5CDC	E	1	INTEGER_32	1000	☑		
C00804	MCTRL_dnAccelerationAdd	23771	5CDB	E	1	INTEGER_32	1000	☑		
C00805	MCTRL_dnSpeedLowLimit	23770	5CDA	E	1	INTEGER_32	1	☑		
C00806	MCTRL_dnTorqueLowLimit	23769	5CD9	E	1	INTEGER_32	100	☑		
C00807	MCTRL_dnTorqueHighLimit	23768	5CD8	E	1	INTEGER_32	100	☑		
C00808	MCTRL_dnPosCtrlOutLimit	23767	5CD7	E	1	INTEGER_32	1	☑		
C00809	MCTRL_dnTorqueCtrlAdapt	23766	5CD6	E	1	INTEGER_32	100	☑		
C00810	MCTRL_dnSpeedCtrlAdapt	23765	5CD5	E	1	INTEGER_32	100	☑		
C00811	MCTRL_dnPosCtrlAdapt	23764	5CD4	E	1	INTEGER_32	100	☑		
C00812	MCTRL_dnMotorPosRefValue	23763	5CD3	A	2	UNSIGNED_32	1	☑		
C00813	MCTRL_dnLoadPosRefValue	23762	5CD2	A	2	UNSIGNED_32	1	☑		
C00814	MCTRL_dnBoost	23761	5CD1	E	1	INTEGER_32	1	☑		
C00815	MCTRL_dnSpeedCtrlIntegrator	23760	5CD0	E	1	INTEGER_32	1000	☑		
C00816	MCTRL_dnFieldWeak	23759	5CCF	E	1	INTEGER_32	100	☑		
C00817	MCTRL_dnSpeedSet_s	23758	5CCE	E	1	INTEGER_32	1	☑		
C00818	MCTRL_dnMvorAdapt	23757	5CCD	E	1	INTEGER_32	100	☑		
C00854	ID status	23721	5CA9	E	1	UNSIGNED_32	1	☑		
C00878	Status DCTRL control input	23697	5C91	A	5	UNSIGNED_8	1	☑		
C00909	Speed limitation	23666	5C72	A	2	INTEGER_16	10	☑	☑	
C00950	VFC: V/f characteristic shape	23625	5C49	E	1	UNSIGNED_32	1	☑	☑	
C00951	VFC: V/f base frequency	23624	5C48	E	1	INTEGER_32	1	☑	☑	
C00952	VFC: Frequency interpol. point n	23623	5C47	A	11	INTEGER_32	1	☑	☑	☑
C00953	VFC: Voltage interpol. point n	23622	5C46	A	11	INTEGER_32	100	☑	☑	☑
C00954	VFC: Activat. interpol. point n	23621	5C45	A	11	UNSIGNED_32	1	☑	☑	☑
C00955	VFC: Vmax reduction	23620	5C44	E	1	UNSIGNED_32	1	☑	☑	
C00957	VFC: VVC current setpoint	23618	5C42	E	1	INTEGER_32	100	☑	☑	
C00958	VFC: VVC gain	23617	5C41	E	1	UNSIGNED_32	100	☑	☑	
C00959	VFC: VVC reset time	23616	5C40	E	1	UNSIGNED_32	100	☑	☑	
C00960	VFC: V/f voltage boost	23615	5C3F	E	1	INTEGER_32	1	☑	☑	
C00961	VFC: Load - cw/ccw-operation	23614	5C3E	E	1	UNSIGNED_32	1	☑	☑	☑
C00962	VFC: Load adjustment	23613	5C3D	E	1	UNSIGNED_32	100	☑	☑	
C00963	VFC: Gain - Imax controller	23612	5C3C	E	1	UNSIGNED_32	1000	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C00964	VFC: Reset time - Imax controller	23611	5C3B	E	1	UNSIGNED_32	10	☑	☑	
C00965	VFC: Gain - slip compensation	23610	5C3A	E	1	INTEGER_32	100	☑	☑	
C00966	VFC: Time const. - slip compens.	23609	5C39	E	1	UNSIGNED_32	1	☑	☑	
C00967	VFC: Gain - oscillation damping	23608	5C38	E	1	INTEGER_32	1	☑	☑	
C00968	VFC: Time const. - oscill. damp.	23607	5C37	E	1	INTEGER_32	1	☑	☑	
C00969	VFC: Limitation - oscill. damp.	23606	5C36	E	1	INTEGER_32	10	☑	☑	
C00970	VFC: ramp-end frequ. - oscill. damp.	23605	5C35	E	1	INTEGER_32	1	☑	☑	
C00971	VFC: Influence - speed controller	23604	5C34	E	1	UNSIGNED_32	100	☑	☑	
C00972	VFC: Gain - speed controller	23603	5C33	E	1	UNSIGNED_32	1000	☑	☑	
C00973	VFC: Reset time - speed contr.	23602	5C32	E	1	UNSIGNED_32	10	☑	☑	
C00974	DC brake: Current	23601	5C31	E	1	INTEGER_32	100	☑	☑	
C00975	DC brake: Current for quick stop	23600	5C30	E	1	INTEGER_32	100	☑	☑	
C00976	DC brake: Activat. by quick stop	23599	5C2F	E	1	UNSIGNED_32	1	☑	☑	
C00977	Min. inh-time aft. overvolt.	23598	5C2E	E	1	UNSIGNED_32	1	☑	☑	
C00980	VFC: Override point of field weakening	23595	5C2B	E	1	INTEGER_32	1	☑	☑	
C00985	SLVC: Gain - flux controller	23590	5C26	E	1	UNSIGNED_32	100	☑	☑	
C00986	SLVC: Gain - cross curr. contr.	23589	5C25	E	1	UNSIGNED_32	100	☑	☑	
C00987	SLVC: Gain - torque controller	23588	5C24	E	1	UNSIGNED_32	10000	☑	☑	
C00988	SLVC: Torque controller reset time	23587	5C23	E	1	UNSIGNED_32	100	☑	☑	
C00989	SLVC: Time const.- Para. adj.	23586	5C22	A	2	UNSIGNED_32	1	☑	☑	
C00990	Flying restart: Activation	23585	5C21	E	1	UNSIGNED_32	1	☑	☑	
C00991	Flying restart: Current	23584	5C20	E	1	INTEGER_32	1	☑	☑	
C00992	Flying restart: Start frequency	23583	5C1F	E	1	INTEGER_32	10	☑	☑	
C00993	Flying restart: Integration time	23582	5C1E	E	1	UNSIGNED_32	1	☑	☑	
C00994	Flying restart: Min. deviation	23581	5C1D	E	1	UNSIGNED_32	100	☑	☑	
C00995	Flying restart: Delay time	23580	5C1C	E	1	UNSIGNED_32	1	☑	☑	
C00998	VFC: Frequency setpoint	23577	5C19	E	1	INTEGER_32	10	☑		
C01120	Sync source	23455	5B9F	E	1	UNSIGNED_8	1	☑	☑	
C01121	Sync cycle time	23454	5B9E	E	1	UNSIGNED_32	1	☑	☑	
C01122	Sync phase position	23453	5B9D	E	1	UNSIGNED_32	1	☑	☑	
C01123	Sync tolerance	23452	5B9C	E	1	UNSIGNED_32	1	☑	☑	
C01124	Sync-PLL increment	23451	5B9B	E	1	UNSIGNED_8	1	☑	☑	
C01130	CAN SYNC application cycle	23445	5B95	E	1	UNSIGNED_16	1	☑	☑	
C01190	Motor thermal sensor	23385	5B59	E	1	UNSIGNED_32	1	☑	☑	
C01191	Spec. characteristic: temperature	23384	5B58	A	2	UNSIGNED_32	1	☑	☑	
C01192	Spec. characteristic: resistance	23383	5B57	A	2	UNSIGNED_32	1	☑	☑	
C01193	Motor temp. feedback system	23382	5B56	E	1	UNSIGNED_16	1	☑	☑	☑
C01194	Motor operating temperature	23381	5B55	E	1	INTEGER_32	1	☑	☑	
C01195	Influence winding I²xt mon.	23380	5B54	E	1	UNSIGNED_32	1	☑	☑	
C01196	S1 torque characteristic I²xt mon.	23379	5B53	A	8	UNSIGNED_32	1	☑	☑	
C01198	Async. motor: Stall protection	23377	5B51	E	1	UNSIGNED_32	1	☑	☑	
C01199	Enhanced power	23376	5B50	E	1	UNSIGNED_32	1	☑	☑	☑
C01200	Dual motor temperature	23375	5B4F	A	2	INTEGER_32	1	☑		
C01201	Delay time for fan start	23374	5B4E	E	1	UNSIGNED_32	1	☑	☑	
C01203	Counter: Brake chopper overload	23372	5B4C	E	1	UNSIGNED_16	1	☑		
C01204	Counter: Ixt overload	23371	5B4B	E	1	UNSIGNED_16	1	☑		
C01205	Counter: DC bus overvoltage	23370	5B4A	E	1	UNSIGNED_16	1	☑		
C01206	Counter: Mains switching	23369	5B49	E	1	UNSIGNED_16	1	☑		
C01208	Counter: Heatsink overtemp.	23367	5B47	E	1	UNSIGNED_16	1	☑		
C01209	Counter: Housing overtemp.	23366	5B46	E	1	UNSIGNED_16	1	☑		
C01210	Counter: internal	23365	5B45	E	1	UNSIGNED_8	1	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C01212	Counter: Power section overload	23363	5B43	E	1	UNSIGNED_16	1	☑		
C01214	Internal clock	23361	5B41	E	1	VISIBLE_STRING	1	☑		
C01215	Set time and date	23360	5B40	A	6	UNSIGNED_16	1	☑	☑	
C01230	Resp. to communication task overflow	23345	5B31	E	1	UNSIGNED_8	1	☑	☑	
C01501	Resp. to comm. error with MXI1	23074	5A22	E	1	UNSIGNED_32	1	☑	☑	
C01502	Resp. to comm. error with MXI2	23073	5A21	E	1	UNSIGNED_32	1	☑	☑	
C01510	Ethernet IP address client x	23065	5A19	A	3	VISIBLE_STRING	1	☑		
C01511	Ethernet status client x	23064	5A18	A	3	UNSIGNED_8	1	☑		
C01902	Diagnostics X6: Max. baud rate	22673	5891	E	1	UNSIGNED_32	1	☑	☑	
C01903	Diagnostics X6: Change baud rate	22672	5890	E	1	UNSIGNED_32	1	☑	☑	
C01905	Diagnostics X6: Curr. baud rate	22670	588E	E	1	UNSIGNED_32	1	☑		
C02104	Program auto-start	22471	57C7	E	1	UNSIGNED_32	1	☑	☑	
C02108	Program status	22467	57C3	E	1	UNSIGNED_8	1	☑		
C02109	Program runtime	22466	57C2	E	1	UNSIGNED_16	1	☑		
C02110	User code: Memory utilisation	22465	57C1	E	1	UNSIGNED_32	1	☑		
C02111	Resp. to task overflow	22464	57C0	E	1	UNSIGNED_8	1	☑	☑	
C02113	Program name	22462	57BE	E	1	VISIBLE_STRING	1	☑		
C02121	Runtime ApplicationTask	22454	57B6	A	2	UNSIGNED_32	1	☑		
C02122	Runtime UserTask	22453	57B5	A	2	UNSIGNED_32	1	☑		
C02123	Runtime IdleTask	22452	57B4	A	2	UNSIGNED_32	1	☑		
C02520	Gearbox factor numerator: Motor	22055	5627	E	1	INTEGER_32	1	☑	☑	☑
C02521	Gearbox factor denom.: Motor	22054	5626	E	1	INTEGER_32	1	☑	☑	☑
C02522	Gearbox factor num.: Pos. enc.	22053	5625	E	1	INTEGER_32	1	☑	☑	☑
C02523	Gearbox fac. denom.: Pos. enc.	22052	5624	E	1	INTEGER_32	1	☑	☑	☑
C02524	Feed constant	22051	5623	E	1	UNSIGNED_32	10000	☑	☑	☑
C02525	Unit	22050	5622	E	1	UNSIGNED_32	1	☑	☑	
C02526	User-defined unit	22049	5621	E	1	VISIBLE_STRING	1	☑	☑	
C02527	Motor mounting direction	22048	5620	E	1	UNSIGNED_32	1	☑	☑	☑
C02528	Traversing range	22047	561F	E	1	UNSIGNED_32	1	☑	☑	☑
C02529	Position encoder mounting dir.	22046	561E	E	1	UNSIGNED_32	1	☑	☑	☑
C02530	Active function state	22045	561D	E	1	INTEGER_32	1	☑		
C02531	Gearbox factors (decimal)	22044	561C	A	3	UNSIGNED_32	1000	☑		
C02532	Resolution of a unit	22043	561B	E	1	UNSIGNED_32	1	☑		
C02533	Time unit	22042	561A	E	1	UNSIGNED_32	1	☑		
C02534	Used time unit	22041	5619	E	1	VISIBLE_STRING	1	☑		
C02535	Used unit	22040	5618	E	1	VISIBLE_STRING	1	☑		
C02536	Cycle	22039	5617	E	1	UNSIGNED_32	10000	☑	☑	☑
C02537	Speed unit	22038	5616	E	1	VISIBLE_STRING	1	☑		
C02538	Acceleration unit	22037	5615	E	1	VISIBLE_STRING	1	☑		
C02539	Max. presentable position	22036	5614	E	1	INTEGER_32	10000	☑		
C02540	Max. presentable speed	22035	5613	E	1	INTEGER_32	10000	☑		
C02541	Max. presentable acceleration	22034	5612	E	1	INTEGER_32	10000	☑		
C02542	Load reference speed	22033	5611	E	1	UNSIGNED_32	1000	☑		
C02543	Load reference torque	22032	5610	E	1	UNSIGNED_32	1000	☑		
C02544	Reference speed	22031	560F	E	1	INTEGER_32	10000	☑		
C02545	Reference Jerktime	22030	560E	E	1	UNSIGNED_32	1000	☑	☑	
C02547	DI_dnState	22028	560C	E	1	INTEGER_32	1	☑		
C02548	DI_bErrors	22027	560B	A	4	UNSIGNED_32	1	☑		
C02549	Drive interface: Signals	22026	560A	A	15	UNSIGNED_32	1	☑		
C02550	Setpoint interpolation	22025	5609	A	3	UNSIGNED_32	1	☑	☑	
C02552	Position setpoint (motor interface)	22023	5607	E	1	INTEGER_32	10000	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C02553	Position controller gain	22022	5606	E	1	UNSIGNED_32	100	☑	☑	
C02554	Position controller reset time	22021	5605	E	1	UNSIGNED_32	1000	☑	☑	
C02555	D component position controller	22020	5604	E	1	UNSIGNED_32	1000	☑	☑	
C02556	Pos. contr. limitation	22019	5603	E	1	INTEGER_32	10000	☑	☑	
C02557	Phase controller output	22018	5602	E	1	INTEGER_32	10000	☑		
C02558	Pos. contr. output	22017	5601	E	1	INTEGER_32	10000	☑		
C02559	Internal torque limits	22016	5600	A	2	INTEGER_32	100	☑		
C02560	Messages - motor interface	22015	55FF	E	1	UNSIGNED_32	1	☑		
C02561	Speed feedforw. control gain	22014	55FE	E	1	INTEGER_32	100	☑	☑	
C02562	Filter time constant	22013	55FD	E	1	UNSIGNED_32	1000	☑	☑	
C02567	Control mode	22008	55F8	E	1	UNSIGNED_32	1	☑		
C02568	Motor interface: % signals	22007	55F7	A	10	INTEGER_32	100	☑		
C02569	Motor interface.: Dig. signals	22006	55F6	A	16	UNSIGNED_32	1	☑		
C02570	Position control structure	22005	55F5	E	1	UNSIGNED_32	1	☑	☑	☑
C02572	Speed setpoint (enc. eval.)	22003	55F3	E	1	INTEGER_32	10000	☑		
C02573	Position setpoint (enc. eval.)	22002	55F2	E	1	INTEGER_32	10000	☑		
C02574	Actual speed (enc. eval.)	22001	55F1	E	1	INTEGER_32	10000	☑		
C02575	Actual position (enc. eval.)	22000	55F0	E	1	INTEGER_32	10000	☑		
C02576	Following error	21999	55EF	E	1	INTEGER_32	10000	☑		
C02577	External actual position	21998	55EE	E	1	INTEGER_32	10000	☑		
C02578	Offset actual pos. value/setp.	21997	55ED	E	1	INTEGER_32	10000	☑		
C02579	Encoder eval.: Dig. signals	21996	55EC	A	3	UNSIGNED_32	1	☑		
C02580	Operating mode brake	21995	55EB	E	1	UNSIGNED_32	1	☑	☑	☑
C02581	Threshold - brake activation	21994	55EA	E	1	INTEGER_32	1	☑	☑	
C02582	Brake resp. to pulse inhibit	21993	55E9	E	1	UNSIGNED_32	1	☑	☑	
C02583	Status input monitoring	21992	55E8	E	1	UNSIGNED_32	1	☑	☑	
C02585	Brake control polarity	21990	55E6	E	1	UNSIGNED_32	1	☑	☑	
C02586	Starting torque 1	21989	55E5	E	1	INTEGER_32	100	☑	☑	
C02587	Starting torque 2	21988	55E4	E	1	INTEGER_32	100	☑	☑	
C02588	Source of starting torque	21987	55E3	E	1	UNSIGNED_32	1	☑	☑	
C02589	Brake closing time	21986	55E2	E	1	UNSIGNED_32	1	☑	☑	
C02590	Brake opening time	21985	55E1	E	1	UNSIGNED_32	1	☑	☑	
C02591	Waiting time - status monitoring	21984	55E0	E	1	UNSIGNED_32	1	☑	☑	
C02593	Waiting time - brake activation	21982	55DE	E	1	UNSIGNED_32	1000	☑	☑	
C02594	Test torque	21981	55DD	E	1	INTEGER_32	100	☑	☑	
C02595	Permissible angle of rotation	21980	55DC	E	1	INTEGER_32	1	☑	☑	
C02596	Grinding speed	21979	55 db	E	1	INTEGER_32	1	☑	☑	
C02597	Accel./decel. time - grinding	21978	55DA	E	1	UNSIGNED_32	1000	☑	☑	
C02598	Grinding ON time	21977	55D9	E	1	UNSIGNED_32	10	☑	☑	
C02599	Grinding OFF time	21976	55D8	E	1	UNSIGNED_32	10	☑	☑	
C02600	Acceleration time feedf. control	21975	55D7	E	1	UNSIGNED_32	1000	☑	☑	
C02601	Ref. for Accel. time of brake	21974	55D6	E	1	UNSIGNED_32	1	☑	☑	
C02602	Source for feedf. control brake	21973	55D5	E	1	UNSIGNED_32	1	☑	☑	
C02603	Threshold 1 for opening brake	21972	55D4	E	1	INTEGER_32	1	☑	☑	
C02604	Threshold 2 for opening brake	21971	55D3	E	1	INTEGER_32	1	☑	☑	
C02607	BRK_dnState	21968	55D0	E	1	INTEGER_32	1	☑		
C02608	BRK_dnTorqueAdd_n	21967	55CF	E	1	INTEGER_32	100	☑		
C02609	Brake control: Dig. signals	21966	55CE	A	10	UNSIGNED_32	1	☑		
C02610	Deceleration time for stop	21965	55CD	E	1	UNSIGNED_32	1000	☑	☑	
C02611	S-ramp time for stop	21964	55CC	E	1	UNSIGNED_32	1000	☑	☑	
C02612	Ref. for decel. time of stop	21963	55CB	E	1	UNSIGNED_32	1	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C02616	STP_dnState	21959	55C7	E	1	INTEGER_32	1	☑		
C02617	STP_bStopActive	21958	55C6	E	1	UNSIGNED_32	1	☑		
C02619	Quick stop: Dig. signals	21956	55C4	A	5	UNSIGNED_32	1	☑		
C02620	Manual jog: Speed 1	21955	55C3	E	1	INTEGER_32	10000	☑	☑	
C02621	Manual jog: Speed 2	21954	55C2	E	1	INTEGER_32	10000	☑	☑	
C02622	Manual jog: Acceleration	21953	55C1	E	1	INTEGER_32	10000	☑	☑	
C02623	Manual jog: Deceleration	21952	55C0	E	1	INTEGER_32	10000	☑	☑	
C02624	Manual jog: S-ramp time	21951	55BF	E	1	UNSIGNED_32	1000	☑	☑	
C02625	Manual jog: Step size	21950	55BE	E	1	INTEGER_32	10000	☑	☑	
C02626	Manual jog: Index stop position	21949	55BD	A	16	INTEGER_32	1	☑	☑	
C02627	Manual jog: Selected stop position	21948	55BC	A	16	INTEGER_32	10000	☑		
C02637	MAN_dnSpeedOverride_n	21938	55B2	E	1	INTEGER_32	100	☑		
C02638	Manual jog: Status	21937	55B1	E	1	INTEGER_32	1	☑		
C02639	Manual jog: Dig. signals	21936	55B0	A	9	UNSIGNED_32	1	☑		
C02640	Homing mode	21935	55AF	E	1	UNSIGNED_32	1	☑	☑	
C02641	Action after detect home position	21934	55AE	E	1	UNSIGNED_32	1	☑	☑	
C02642	Home position	21933	55AD	E	1	INTEGER_32	10000	☑	☑	
C02643	Homing: Target position	21932	55AC	E	1	INTEGER_32	10000	☑	☑	
C02644	Homing: Speed 1	21931	55AB	E	1	INTEGER_32	10000	☑	☑	
C02645	Homing: Acceleration 1	21930	55AA	E	1	INTEGER_32	10000	☑	☑	
C02646	Homing: Speed 2	21929	55A9	E	1	INTEGER_32	10000	☑	☑	
C02647	Homing: Acceleration 2	21928	55A8	E	1	INTEGER_32	10000	☑	☑	
C02648	Homing: S-ramp time	21927	55A7	E	1	INTEGER_32	1	☑	☑	
C02649	Homing: Torque limit	21926	55A6	E	1	INTEGER_32	100	☑	☑	
C02650	Homing: Blocking time	21925	55A5	E	1	UNSIGNED_32	1000	☑	☑	
C02651	Homing: TP configuration	21924	55A4	E	1	UNSIGNED_32	1	☑	☑	
C02652	Ref. pos. after mains switching	21923	55A3	E	1	UNSIGNED_32	1	☑	☑	
C02653	Max. rot. ang. aft. mns. swtch.	21922	55A2	E	1	INTEGER_32	1	☑	☑	
C02655	HM_dnSpeedOverride_n	21920	55A0	E	1	INTEGER_32	100	☑		
C02656	Actual position (homing)	21919	559F	E	1	INTEGER_32	10000	☑		
C02657	HM_dnState	21918	559E	E	1	INTEGER_32	1	☑		
C02658	HM_dnHomePos_p	21917	559D	E	1	INTEGER_32	10000	☑		
C02659	Homing: Dig. signals	21916	559C	A	9	UNSIGNED_32	1	☑		
C02670	Tolerance for POS_bActPosInTarget	21905	5591	E	1	INTEGER_32	10000	☑	☑	
C02671	Tolerance for POS_bDriveInTarget	21904	5590	E	1	INTEGER_32	10000	☑	☑	
C02672	Hysteresis for POS_bDriveInTarget	21903	558F	E	1	INTEGER_32	10000	☑	☑	
C02673	Activate DriveInTarget Modulo	21902	558E	E	1	UNSIGNED_32	1	☑	☑	
C02674	POS_dwActualProfileNumber	21901	558D	E	1	UNSIGNED_32	1	☑		
C02675	POS_dnState	21900	558C	E	1	INTEGER_32	1	☑		
C02676	POS_dnProfileSpeed_s	21899	558B	E	1	INTEGER_32	10000	☑		
C02677	Positioning: % signals	21898	558A	A	3	INTEGER_32	100	☑		
C02678	Positioning: Pos. signals	21897	5589	A	4	INTEGER_32	10000	☑		
C02679	Positioning: Dig. signals	21896	5588	A	12	UNSIGNED_32	1	☑		
C02680	Source position setpoint	21895	5587	E	1	UNSIGNED_32	1	☑	☑	
C02681	Source add. speed	21894	5586	E	1	UNSIGNED_32	1	☑	☑	
C02685	PF_dnMotorAcc_x	21890	5582	E	1	INTEGER_32	10	☑		
C02686	PF_dnSpeedAdd1_s	21889	5581	E	1	INTEGER_32	10	☑		
C02687	Position follower: % signals	21888	5580	A	2	INTEGER_32	100	☑		
C02688	PF_dnPositionSet_p	21887	557F	E	1	INTEGER_32	10000	☑		
C02689	Position follower: Dig. signals	21886	557E	A	2	UNSIGNED_32	1	☑		
C02692	SF_dnMotorAcc_x	21883	557B	E	1	INTEGER_32	10	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C02693	SF_dnSpeedAdd_s	21882	557A	E	1	INTEGER_32	10	☑		
C02694	Speed follower: % signals	21881	5579	A	2	INTEGER_32	100	☑		
C02695	Speed follower: Dig. signals	21880	5578	A	2	UNSIGNED_32	1	☑		
C02698	Torque follower: % signals	21877	5575	A	3	INTEGER_32	100	☑		
C02699	Torque follower: Dig. signals	21876	5574	A	2	UNSIGNED_32	1	☑		
C02700	Software limits pos. effective	21875	5573	E	1	UNSIGNED_32	1	☑	☑	
C02701	Software limit positions	21874	5572	A	2	INTEGER_32	10000	☑	☑	
C02702	Limitations effective	21873	5571	E	1	UNSIGNED_32	1	☑	☑	
C02703	Max. speed	21872	5570	E	1	INTEGER_32	10000	☑	☑	
C02704	Max. speed [rpm]	21871	556F	E	1	INTEGER_32	10	☑		
C02705	Max. acceleration	21870	556E	E	1	INTEGER_32	10000	☑	☑	
C02706	Min. S-ramp time	21869	556D	E	1	UNSIGNED_32	1	☑	☑	
C02707	Permissible direction of rot.	21868	556C	E	1	UNSIGNED_32	1	☑	☑	
C02708	Limited speed	21867	556B	A	4	INTEGER_32	10000	☑	☑	
C02709	Limited speed [rpm]	21866	556 A	A	4	INTEGER_32	10	☑		
C02710	Delay lim. speed	21865	5569	A	4	UNSIGNED_32	10000	☑	☑	
C02711	S-ramp time lim. speed	21864	5568	A	4	UNSIGNED_32	1	☑	☑	
C02712	Decel. time lim. speed	21863	5567	A	4	UNSIGNED_32	1	☑		
C02713	Max. dist. manual control	21862	5566	E	1	UNSIGNED_32	10000	☑	☑	
C02714	Max. dist. manual jog [inc.]	21861	5565	E	1	UNSIGNED_32	1	☑		
C02715	Limitation active	21860	5564	E	1	UNSIGNED_32	1	☑		
C02716	Resp. to limitation	21859	5563	A	3	UNSIGNED_32	1	☑	☑	
C02717	LIM_dwControl	21858	5562	E	1	UNSIGNED_32	1	☑		
C02718	LIM_dnState	21857	5561	E	1	INTEGER_32	1	☑		
C02719	Limiter: Dig. signals	21856	5560	A	3	UNSIGNED_32	1	☑		
C02720	Software limit position monitoring	21855	555F	E	1	UNSIGNED_32	1	☑	☑	
C02730	Analog inputs: Gain	21845	5555	A	2	INTEGER_32	100	☑	☑	
C02731	Analog inputs: Offset	21844	5554	A	2	INTEGER_32	100	☑	☑	
C02732	Analog inputs: Dead band	21843	5553	A	2	INTEGER_32	100	☑	☑	
C02733	Analog outputs: Gain	21842	5552	A	2	INTEGER_32	100	☑	☑	
C02734	Analog outputs: Offset	21841	5551	A	2	INTEGER_32	100	☑	☑	
C02760	Activate Encoder	21815	5537	E	1	UNSIGNED_32	1	☑	☑	
C02761	Resolution Multiturn	21814	5536	E	1	UNSIGNED_32	1	☑		
C02762	Encoder position	21813	5535	E	1	INTEGER_32	1	☑		
C02763	Encoder revolution	21812	5534	E	1	INTEGER_32	1	☑		
C02764	Encoderspeed	21811	5533	E	1	INTEGER_32	10	☑		
C02765	ENC_bError	21810	5532	E	1	UNSIGNED_32	1	☑		
C02770	Mode of operation	21805	552D	A	5	UNSIGNED_32	1	☑	☑	
C02771	Frequency	21804	552C	A	4	INTEGER_32	10	☑	☑	
C02772	Start angle	21803	552B	A	4	INTEGER_32	10	☑	☑	
C02773	Current	21802	552A	A	4	INTEGER_32	100	☑	☑	
C02774	Acceleration time	21801	5529	A	4	INTEGER_32	1000	☑	☑	
C02775	Deceleration time	21800	5528	A	4	INTEGER_32	1000	☑	☑	
C02776	Duration time	21799	5527	A	4	INTEGER_32	1000	☑	☑	
C02779	MoI_SetpointCurrent	21796	5524	E	1	UNSIGNED_32	100	☑		
C02780	MoI_dnState	21795	5523	E	1	INTEGER_32	1	☑		
C02781	ManualJogOpenLoop: Dig. signals	21794	5522	A	8	UNSIGNED_32	1	☑		
C02785	Activation of PPI	21790	551E	E	1	UNSIGNED_32	1	☑	☑	
C02786	Mode of PPI	21789	551D	E	1	UNSIGNED_32	1	☑	☑	
C02787	Ppi_dnState	21788	551C	E	1	INTEGER_32	1	☑		
C02788	PolePosition Setpoint	21787	551B	E	1	INTEGER_32	10	☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C02789	PolePositionIdentification: Dig. signals	21786	551A	A	9	UNSIGNED_32	1	☑		
C02800	Analog input x: Input signal	21775	550F	A	2	INTEGER_16	1	☑		
C02801	Analog output x: Output signal	21774	550E	A	2	INTEGER_16	1	☑		
C02802	Status word: Digital outputs	21773	550D	E	1	BITFIELD_32	1	☑		
C02803	Status word: Digital inputs	21772	550C	E	1	BITFIELD_32	1	☑		
C02810	Touch probe x: Delay time	21765	5505	A	10	UNSIGNED_32	1	☑	☑	
C02830	Digital inputs: Delay time	21745	54F1	A	8	UNSIGNED_8	1	☑	☑	
C02853	Lss sat. characteristic	21722	54DA	A	17	UNSIGNED_16	1	☑	☑	☑
C02855	Imax f. Lss sat. characteristic	21720	54D8	E	1	UNSIGNED_32	10	☑	☑	☑
C02859	Activate Lss sat. char.	21716	54D4	E	1	UNSIGNED_8	1	☑	☑	☑
C02860	Rr adjustment	21715	54D3	E	1	UNSIGNED_32	100	☑	☑	
C02861	Lh adjustment	21714	54D2	E	1	UNSIGNED_32	100	☑	☑	
C02862	Service code	21713	54D1	A	2	UNSIGNED_16	1	☑	☑	
C02863	Service code	21712	54D0	E	1	INTEGER_16	1	☑	☑	
C02864	Service code	21711	54CF	E	1	INTEGER_32	1	☑		
C02865	Adaptation of Ur	21710	54CE	E	1	UNSIGNED_32	100	☑	☑	
C02900	User Password	21675	54AB	E	1	VISIBLE_STRING		☑	☑	
C02901	CamMemory	21674	54AA	A	3	UNSIGNED_32	1	☑		
C02902	Time stamp of cam data	21673	54A9	A	4	UNSIGNED_32	1	☑		
C02903	GUID cam data	21672	54A8	A	4	OCTET_STRING	1	☑		
C02905	Online Change Mode	21670	54A6	E	1	UNSIGNED_32	1	☑	☑	
C02906	Online change status	21669	54A5	E	1	UNSIGNED_32	1	☑		
C02908	Number Of Products	21667	54A3	E	1	UNSIGNED_32	1	☑		
C02909	Active Product	21666	54A2	E	1	UNSIGNED_32	1	☑		
C02910	Product Name	21665	54A1	E	1	VISIBLE_STRING	1	☑		
C02911	Product Choice	21664	54A0	E	1	UNSIGNED_32	1	☑	☑	
C02912	Number Of Products	21663	549F	E	1	UNSIGNED_32	1	☑		
C02919	Number of curve tracks	21656	5498	E	1	UNSIGNED_32	1	☑		
C02920	Cam Track Choice	21655	5497	E	1	UNSIGNED_32	1	☑	☑	
C02921	Cam Track Type	21654	5496	E	1	UNSIGNED_32	1	☑		
C02922	Number of Cam Data Points	21653	5495	E	1	UNSIGNED_32	1	☑		
C02923	Cam Data Point Choice	21652	5494	E	1	UNSIGNED_32	1	☑	☑	
C02924	Change Cam Data Point X	21651	5493	E	1	INTEGER_32	10000	☑	☑	
C02925	Change Cam Data Point Y	21650	5492	E	1	INTEGER_32	10000	☑	☑	
C02926	Change Cam Data Point M	21649	5491	E	1	INTEGER_32	100	☑	☑	
C02927	Auto Inc Cam Data Points	21648	5490	E	1	UNSIGNED_32	1	☑	☑	
C02939	Number of Cont Tracks	21636	5484	E	1	UNSIGNED_32	1	☑		
C02940	Cont Track Choice	21635	5483	E	1	UNSIGNED_32	1	☑	☑	
C02941	Cont Type	21634	5482	E	1	UNSIGNED_32	1	☑		
C02942	Number of Cont Data Points	21633	5481	E	1	UNSIGNED_32	1	☑		
C02943	Cont Data Point Choice	21632	5480	E	1	UNSIGNED_32	1	☑	☑	
C02944	Cont Pos X0	21631	547F	E	1	INTEGER_32	10000	☑	☑	
C02945	Cont Pos X1	21630	547E	E	1	INTEGER_32	10000	☑	☑	
C02946	Cont Time	21629	547D	E	1	UNSIGNED_32	10000	☑	☑	
C02959	Number of Position Tracks	21616	5470	E	1	UNSIGNED_32	1	☑		
C02960	Pos Track Choice	21615	546F	E	1	UNSIGNED_32	1	☑	☑	
C02962	Number of Pos Data Points	21613	546D	E	1	UNSIGNED_32	1	☑		
C02963	Pos Data Point Choice	21612	546C	E	1	UNSIGNED_32	1	☑	☑	
C02964	Change Pos Data Point X	21611	546B	E	1	INTEGER_32	10000	☑	☑	
C02965	Change Pos Data Point Y	21610	546A	E	1	INTEGER_32	10000	☑	☑	

15 Index

C

J

N

O

T