



E84AYCET

EtherCAT®

Inverter Drives 8400 -----



13429553

Contents

1	About this documentation	4
1.1	Document history	6
1.2	Conventions used	7
1.3	Terminology used	8
1.4	Notes used	9
2	Safety instructions	10
2.1	General safety and application notes	10
2.2	Device- and application-specific safety instructions	11
2.3	Residual hazards	11
3	Product description	12
3.1	Application as directed	12
3.2	Identification	12
3.3	Product features	13
3.4	Terminals and interfaces	14
4	Technical data	15
4.1	General data and operating conditions	15
4.2	Protective insulation	16
4.3	Protocol data	19
4.4	Communication time	19
4.5	Dimensions	20
5	Installation	21
5.1	Mechanical installation	22
5.1.1	Mounting for standard devices 0.25 kW and 0.37 kW	22
5.1.2	Mounting for standard devices from 0.55 kW onwards	23
5.1.3	Replacing the communication module	24
5.2	Electrical installation	25
5.2.1	EMC-compliant wiring	25
5.2.2	Network topology	26
5.2.3	EtherCAT connection	27
5.2.4	Specification of the Ethernet cable	29
5.2.5	External voltage supply	31
6	Commissioning	33
6.1	Before initial switch-on	33
6.2	Configuring the Controller (EtherCAT master)	34
6.2.1	Installing device description files	34
6.2.2	Automatic device identification	35
6.2.3	Configuring process data	35
6.2.4	Determining the cycle time	35
6.3	Address allocation	36
6.4	Synchronisation with "Distributed Clocks" (DC)	37
6.4.1	DC configuration in the master	38
6.4.2	DC configuration in the Inverter Drive 8400 (slave)	38
6.4.3	Response of the Lenze EtherCAT nodes during start-up	39
6.5	Establishing an online connection with the »Engineer«	40
6.5.1	Gateway Controller -> configure EtherCAT	41
6.5.2	Gateway Controller -> configure EtherCAT ADS (Beckhoff)	43
6.6	EtherCAT ADS communication parameters in »TwinCAT« and »Engineer«	45
6.6.1	Example: Structure without a Beckhoff Controller	45
6.6.2	Example: Structure with a Beckhoff DIN rail IPC CX1020	49
6.7	Initial switch-on	53

Contents

7	Data transfer	54
7.1	EtherCAT-Frames	55
7.2	EtherCAT datagrams	56
7.3	EtherCAT state machine	57
8	Process data transfer	59
8.1	Accessing process data / PDO mapping	60
8.2	Preconfigured port interconnection of the process data objects (PDO)	61
8.3	Free configuration of the port interconnection of the process data objects (PDO)	62
8.4	CiA402 PDO mapping in case of Inverter Drive 8400 TopLine P	66
9	Parameter data transfer	68
9.1	Establishing a connection between master and slave	68
9.2	Reading and writing parameters	69
9.2.1	Reading parameters (SDO upload)	70
9.2.2	Writing parameters (SDO download)	74
9.3	Implemented CoE objects	79
9.4	EtherCAT objects of the communication module	80
9.5	SDO abort codes (Abort codes)	81
10	Monitoring	82
10.1	Interruption of EtherCAT communication	82
10.2	Sync frame failure detection	83
10.3	Interruption of internal communication	83
11	Diagnostics	84
11.1	LED status displays	84
11.1.1	Module status displays	85
11.1.2	Fieldbus status displays	86
11.1.3	Status displays at X246 and X247	88
11.2	Diagnostics with the »Engineer«	89
11.3	Emergency requests / Emergency messages	90
11.3.1	Structure of the Emergency message	90
11.3.2	Emergency messages (overview)	91
12	Error messages	92
12.1	Short overview of the EtherCAT error messages	92
12.2	Possible causes and remedies	93
13	Parameter reference	96
13.1	Communication-relevant parameters of the standard device	96
13.2	Parameters of the communication module	97
13.3	Table of attributes	103
	Index	105
	Your opinion is important to us	108

1 About this documentation

Contents

This documentation only contains descriptions for the E84AYCET communication module (EtherCAT®).



Note!

This documentation supplements the **Mounting Instructions** supplied with the communication module and the **"Inverter Drives 8400" Hardware Manual**.

The features and functions of the communication module are described in detail.

Examples illustrate typical applications.

The theoretical concepts are only explained to the level of detail required to understand the function of the communication module.

This documentation does not describe any software provided by other manufacturers. No warranty can be given for corresponding data provided in this documentation. For information on how to use the software, please refer to the control system documents (Controller, EtherCAT master).

All product names mentioned in this documentation are trademarks of their corresponding owners.



Tip!

Detailed information about EtherCAT can be found on the website of the EtherCAT Technology Group:

www.EtherCAT.org

Target group

This documentation is intended for all persons who plan, install, commission and maintain the networking and remote servicing of a machine.



Tip!

Current documentation and software updates with regard to Lenze products can be found in the download area at:

www.Lenze.com

Validity information

The information in this documentation applies to the following devices:

Extension module	Type designation	From hardware version	From software version
EtherCAT communication module	E84AYCET	VC	01.00...03
		1A	01.04/05 02.00

Screenshots/application examples

All screenshots in this documentation are application examples. Depending on the firmware version of the communication module and the software version of the engineering tools installed (e.g. »Engineer«), the screenshots in this documentation may differ from the actual screen representation.

1.1

Document history

Version			Description
1.0	08/2009	TD17	First edition
2.0	10/2009	TD17	Amendment of the information on the synchronisation via the fieldbus and general revision
3.0	04/2010	TD17	General revision
4.0	11/2010	TD17	Information about the EtherCAT register " AL Status Code " (book 58) supplemented.
4.1	11/2012	TD17	EtherCAT® is a registered trademark by Beckhoff Automation GmbH, Germany.
5.0	05/2013	TD17	• Revision of firmware version V02.00 • New layout

1 About this documentation

1.2 Conventions used

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

Type of information	Identification	Examples/notes
Numbers		
Decimal separator	Point	In general, the decimal point is used. Example: 1234.56
Hexadecimal	0x[0 ... 9, A ... F]	Example: 0x60F4
Binary • Nibble	In inverted commas Point	Example: '100' Example: '0110.0100'
Text		
Version information	Text colour blue	All pieces of information that only apply to or from a specific software version of the inverter are highlighted accordingly in this documentation. Example: This function extension is available from software version V3.0!
Program name	» «	The Lenze PC software »Engineer«...
Window	<i>italics</i>	The <i>message window</i> ... / The <i>Options</i> dialog box ...
Variable name		Setting <i>bEnable</i> to TRUE...
Control element	Bold	The OK button ... / The Copy command ... / The Properties tab ... / The Name input field ...
Sequence of menu commands		If several successive commands are required for executing a function, the individual commands are separated from each other by an arrow: Select the command File → Open to...
Hyperlink	<u>Underlined</u>	Optically highlighted reference to another topic. Can be activated with a mouse-click in this online documentation.
Icons		
Page reference	 5	Optically highlighted reference to another page. Can be activated with a mouse-click in this online documentation.
Step-by-step instructions		Step-by-step instructions are marked by a pictograph.

1.3

Terminology used

Term	Meaning
Inverter	Lenze inverters of the "Inverter Drives 8400" series
Standard device	
Code	Parameter which serves to parameterise and monitor the drive. In normal usage, the term is usually referred to as "Index".
Subcode	If a code contains several parameters, they are stored in so-called "subcodes". This manual uses a slash "/" as a separator between code and subcode (e.g. "C00118/3"). In normal usage, the term is also referred to as "Subindex".
CoE	CANopen over EtherCAT
DC	"Distributed clocks" for EtherCAT synchronisation
»Engineer«	PC software from Lenze which supports you during engineering (parameterisation, diagnostics, and configuration) throughout the entire life cycle, i.e. from planning to maintenance of the commissioned machine.
»PLC Designer«	
ESI	EtherCAT slave information (device description file in XML format)
	EtherCAT® is a real-time capable Ethernet system with the highest performance. EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
EtherCAT master	Control system (Controller)
HW	Hardware
I-1600.8	CoE index (hexadecimal representation) In the example: index 0x1600, subindex 8
Lenze setting	Settings with which the device is preconfigured ex works.
Standard setting	
PDO	Process data object
SDO	Service data object
SW	Software
»TwinCAT«	Beckhoff PC software for EtherCAT configuration

1.4

Notes used

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

Safety instructions

Layout of the safety instructions:

**Pictograph and signal word!**

(characterise the type and severity of danger)

Note

(describes the danger and suggests how to prevent dangerous situations)

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

Application notes

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

2

Safety instructions

**Note!**

Always observe the specified safety measures to avoid severe injury to persons and damage to property!

Always keep this documentation to hand in the vicinity of the product during operation.

2.1

General safety and application notes

**Danger!**

If you ignore the following basic safety measures, severe injury to persons and damage to material assets may result.

Lenze drive and automation components ...

- shall only be used as directed.
 - ▶ [Application as directed](#) (12)
- must never be commissioned if they display any signs of damage.
- must never be technically modified.
- must never be commissioned if they are not fully mounted.
- must never be operated without the covers required.
- during and after operation can have live, moving and rotating parts, depending on their degree of protection. Surfaces can be hot.

For Lenze drive components ...

- only use the accessories approved.
- only use genuine spare parts supplied by the manufacturer of the product.

Observe all specifications contained in the enclosed documentation and related documentation.

- This is the precondition for a safe and trouble-free operation and for obtaining the product features specified.
 - ▶ [Product features](#) (13)
- The specifications, processes, and circuitry described in this document are for guidance only and must be adapted to your own specific application. Lenze does not take responsibility for the suitability of the process and circuit proposals.

All works on and with Lenze drive and automation components must only be carried out by qualified personnel. According to IEC 60364 or CENELEC HD 384 these are persons who ...

- who are familiar with the installation, assembly, commissioning and operation of the product.
- who have the corresponding qualifications for their work.
- who know all regulations for the prevention of accidents, directives and laws applicable on site and are able to apply them.

2.2

Device- and application-specific safety instructions

- During operation, the communication module must be firmly connected to the standard device.
- With external voltage supply, always use a separate power supply unit, safely separated to EN 61800-5-1 in every control cabinet (SELV/PELV).
- Only use cables that correspond to the given specifications.
 - ▶ [Specification of the Ethernet cable](#) (□ 29)

**Documentation for the standard device, control system, plant/machine**

All the other measures prescribed in this documentation must also be implemented. Observe the safety instructions and application notes stated in this manual.

2.3

Residual hazards**Protection of persons**

If Inverter Drives 8400 are used on a phase earthed mains with a rated mains voltage of ≥ 400 V, external measures need to be implemented in order to provide reliable protection against accidental contact.

- ▶ [Protective insulation](#) (□ 16)

Device protection

The communication module contains electronic components that can be damaged or destroyed by electrostatic discharge.

- ▶ [Installation](#) (□ 21)

3 Product description

3.1 Application as directed

3 Product description

3.1 Application as directed

The communication module ...

- is an accessory module which can be used with the following standard devices:

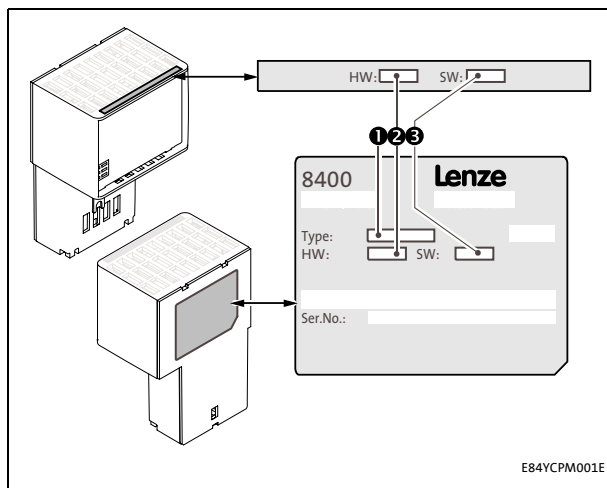
Product series	Type designation	From software version
Inverter Drives 8400 StateLine	E84AVSCxxxxx	03.00
Inverter Drives 8400 HighLine	E84AVHCxxxxx	02.00
Inverter Drives 8400 TopLine	E84AVTCxxxxx	01.00
Inverter Drives 8400 TopLine P (with CiA402)	E84AVPCxxxxx	01.00

- is an item of equipment intended for use in industrial power systems.
- may only be operated under the operating conditions specified in this documentation.
- may only be used in EtherCAT networks.

Any other use shall be deemed inappropriate!

3.2 Identification

The type designation as well as the hardware and software version of the communication module are specified on the nameplate:



1 Type designation (type)

- E84 Product series
- A Version
- Y Module identification: Extension module
- C Module type: Communication module
- ET EtherCAT
- V/S V: Coated version
- S: Standard version

2 Hardware version (HW)

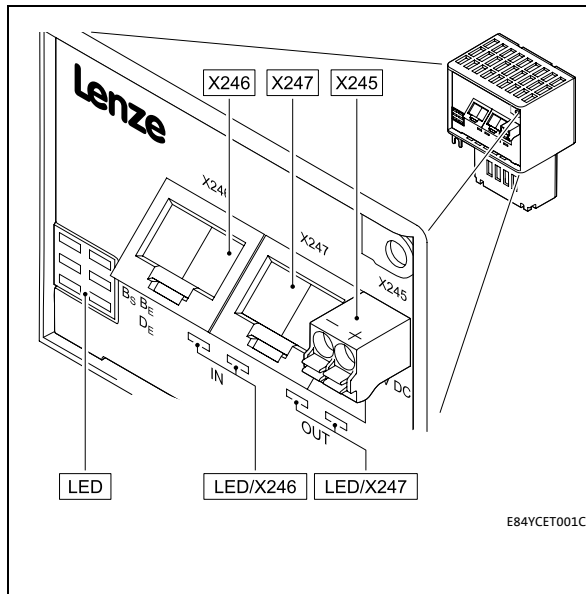
3 Software version (SW)

[3-1] Identification data

3.3 Product features

- Interface module for the EtherCAT communication system to the expansion slots of the Inverter Drives 8400
- The communication module can be supplied internally by the standard device and externally via a separate voltage source.
- Supports the "Distributed Clocks" (DC) functionality for synchronisation via the fieldbus
- Cycle times:
 - 1 ms or an integer multiple of 1 ms
 - max. 15 ms if "Distributed Clocks" (DC) are used
- PDO transfer with CoE (CANopen over EtherCAT)
- Up to 16 process data words (16 bits/word) per direction can be exchanged.
- Access to all Lenze parameters with CoE (CANopen over EtherCAT)

3.4 Terminals and interfaces



[3-2] E84AYCET communication module (EtherCAT)

X245 External voltage supply of the communication module

2-pole plug connector with spring connection

► [External voltage supply](#) (□ 31)

X246 EtherCAT input (IN)

X247 EtherCAT output (OUT)

- RJ45 sockets according to IEC/EN 60603-7
- with 2 LED status displays each for diagnostics

► [Network topology](#) (□ 26)

► [EtherCAT connection](#) (□ 27)

► [Status displays at X246 and X247](#) (□ 88)

MS 5 LED status displays for diagnostics

ME ► [Module status displays](#) (□ 85)

BS ► [Fieldbus status displays](#) (□ 86)

BE

DE

4 Technical data

4.1 General data and operating conditions

Area	Values
Order designation	E84AYCET
Communication profile	EtherCAT
Supported device profile and mailbox protocol	CANopen over EtherCAT (CoE)
Communication medium	S/FTP (Screened Foiled Twisted Pair, ISO/IEC 11801 or EN 50173), CAT 5e
Interface for communication	RJ45: Standard Ethernet (in accordance with IEEE 802.3), 100Base-TX (Fast Ethernet)
Network topology	Line, switch
Bus device type	EtherCAT slave
Number of nodes	Max. 65535 (in the entire network)
Max. cable length between two EtherCAT nodes	100 m (typically)
Vendor ID [hex]	0x3B
Product ID	Depending on the standard device used (see chapter Automatic device identification (□ 35)).
Revision ID	Depending on the software version of the communication module (see chapter Identification (□ 12)).
Baud rate	100 Mbps, full duplex
Cycle times	1 ms or an integer multiple of 1 ms, max. 15 ms if "Distributed clocks" (DC) are used
Voltage supply	External supply via a separate power supply unit • "+": V = 24 V DC (20.4 V - 0 % ... 28.8 V + 0 %), I = 140 mA • "-": Reference potential for external voltage supply
Conformity, approvals	CE UL (see hardware manual "Inverter Drives 8400")



"Inverter Drives 8400" Hardware Manual

Here you can find the **ambient conditions** and information on the **electromagnetic compatibility (EMC)** which also apply to the communication module.

4.2

Protective insulation

**Danger!****Dangerous voltage**

If Inverter Drives 8400 are used on a phase earthed mains with a rated mains voltage ≥ 400 V, external measures need to be implemented in order to provide reliable protection against accidental contact.

Possible consequences:

Death or severe injury

Protective measures:

If protection against accidental contact is required for the control terminals of the inverter and the connections of the plugged device modules, ...

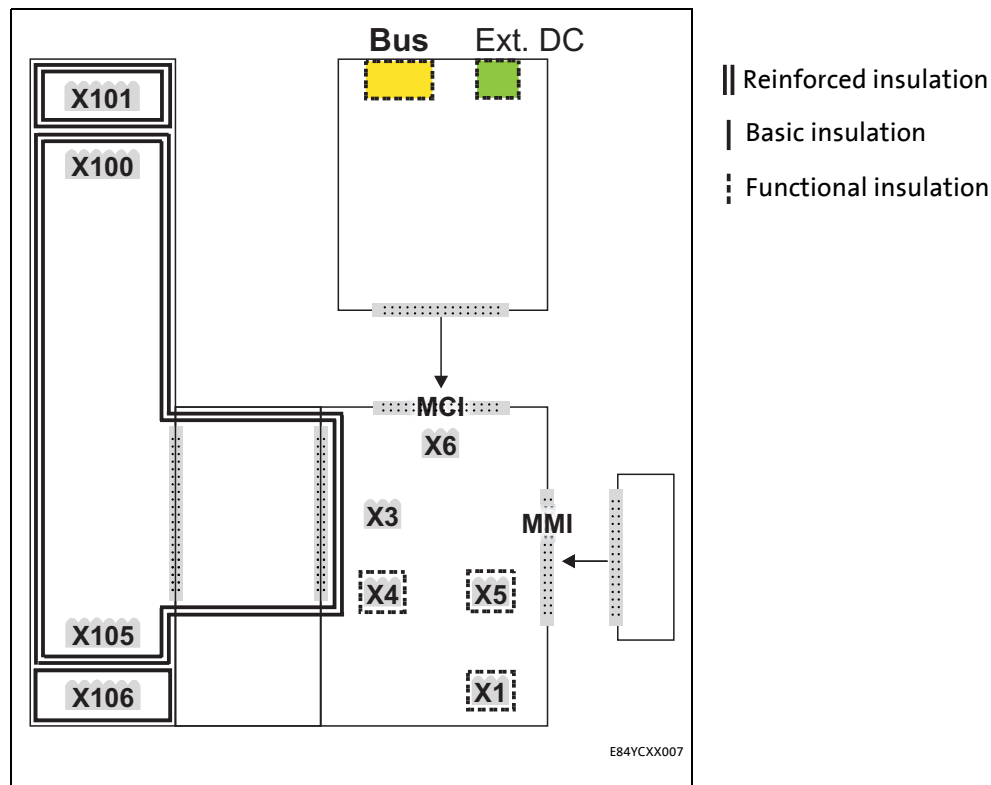
- a double isolating distance must be available.
- the components to be connected must be provided with a second isolating distance.

**Note!**

The protective insulation provided in Inverter Drives 8400 is realised in accordance with EN 61800-5-1.

The following illustration ...

- shows the arrangement of the terminal strips and the separate potential areas of the inverter.
- serves to determine the decisive protective insulation between two terminals located in differently insulated separate potential areas.



[4-1] Protective insulation in accordance with EN61800-5-1

Terminal strip	Connection
X100	Mains / DC bus connection
X101	Relay contact
X105	Motor/brake resistor
X106	Motor PTC
X1	System bus (CAN)
X3	Analog inputs/outputs
X4	Digital outputs
X5	Digital inputs
X6	Diagnostics
MCI	Slot for communication module
MMI	Slot for memory module

Example

Which type of protective insulation is used between the bus terminal of the device module in slot MCI and the mains terminal X100?

The separate potential area with the better protective insulation is decisive.

- The separate potential area of the bus terminal of the device module has a "basic insulation".
- The separate potential area of the mains terminal has a "reinforced insulation".

Result: The insulation between the X100 mains terminal and the bus terminal is of the "reinforced insulation" type.

4.3 Protocol data

Area	Values
Process data words	1 ... 16 process data words per direction (max. 32 bytes, 16 bits/word)
Parameter data (mailbox size for CoE transfer)	Max. 128 bytes

4.4 Communication time

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

The communication times in the EtherCAT network depend on the ...

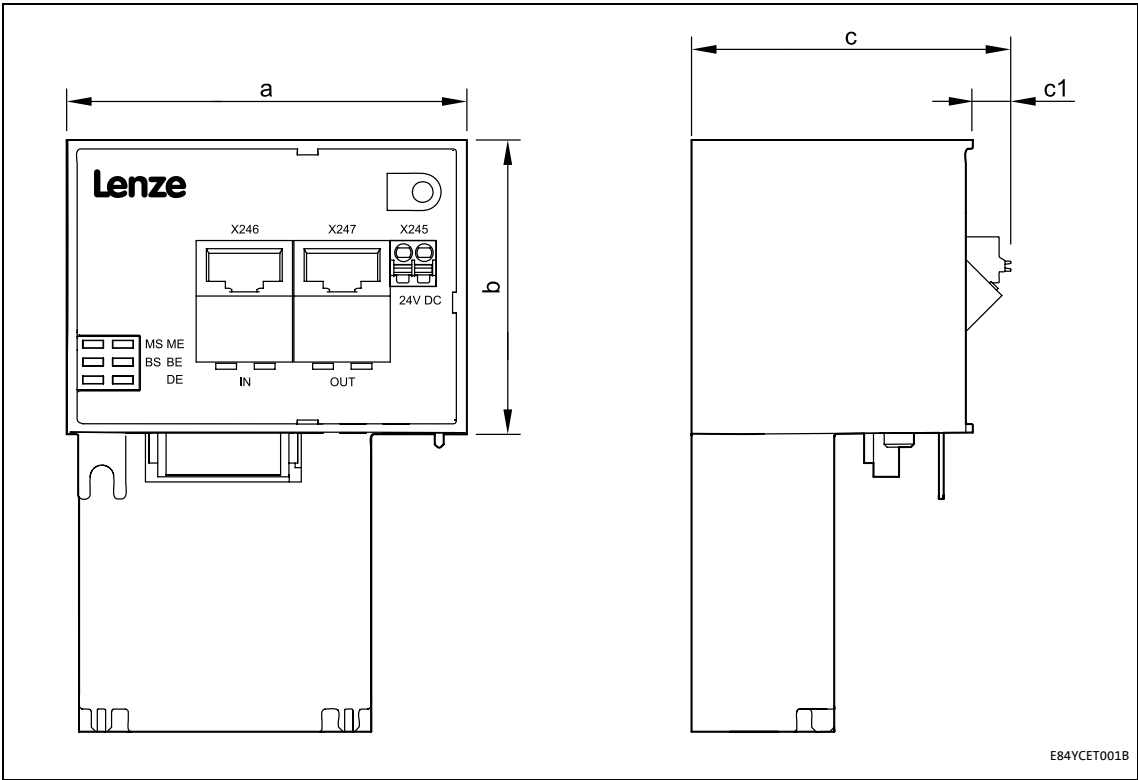
- processing time in the inverter;
- telegram runtime (baud rate / telegram length).

processing time within the inverter

Data	Processing time	
Process data for synchronised operation	0.2 ms	Processing time in the module
	+ 1 ... x ms	Runtime of the application task of the technology application used (tolerance)
Process data for non-synchronised operation	2 ms	Processing time in the module
	+ 1 ... x ms	Runtime of the application task of the technology application used (tolerance)
Parameter data	approx. 30 ms	Processing time
	+ 20 ms	Tolerance (typical) Some codes may require a longer processing time (see software manual/online help for the Inverter Drive 8400).

4.5

Dimensions



[4-2] Dimensions

Type	Dimensions [mm]			
	a	b	c	c1
E84AYCET	67	50	57	8

5 Installation



Stop!

Electrostatic discharge

Electronic components in the communication module can be damaged or destroyed by electrostatic discharge.

Possible consequences:

- The communication module is defective.
- Fieldbus communication is faulty or not possible.

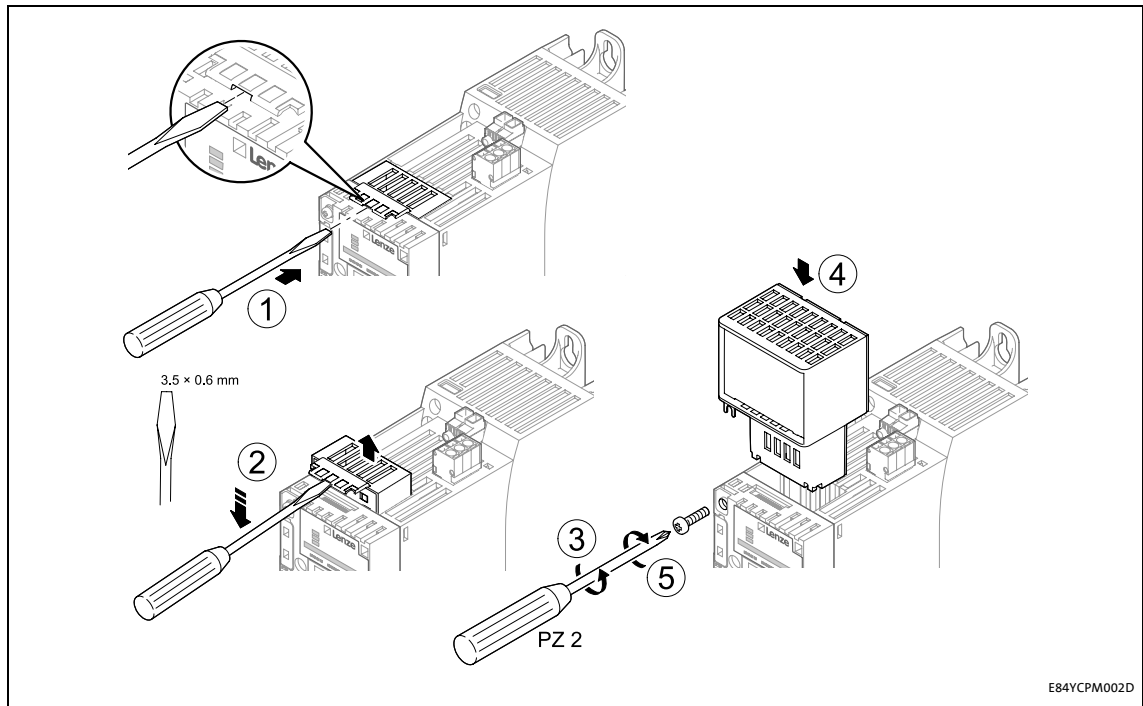
Protective measures:

Before touching the module, make sure that you are free of electrostatic charge.

5.1 Mechanical installation

The communication module can be plugged into the MCI slot or removed while the frequency inverter is switched on. When the module is plugged in, it is recognised automatically and checked for plausibility regarding its function and version.

5.1.1 Mounting for standard devices 0.25 kW and 0.37 kW

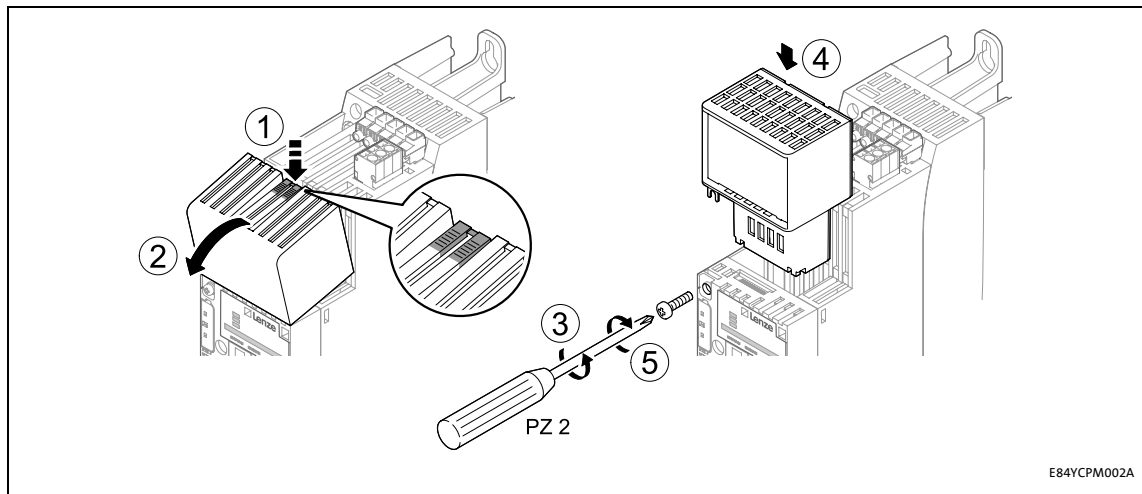


[5-1] Mounting for standard devices 0.25 kW and 0.37 kW

Mounting steps

1. Use a screwdriver to lever out the cover of the MCI slot of the standard device and remove it (1, 2).
2. Loosen the securing screw for the communication module at the standard device (3).
3. Insert the communication module into the MCI slot of the standard device (4).
4. Tighten the securing screw again (5).

5.1.2 Mounting for standard devices from 0.55 kW onwards

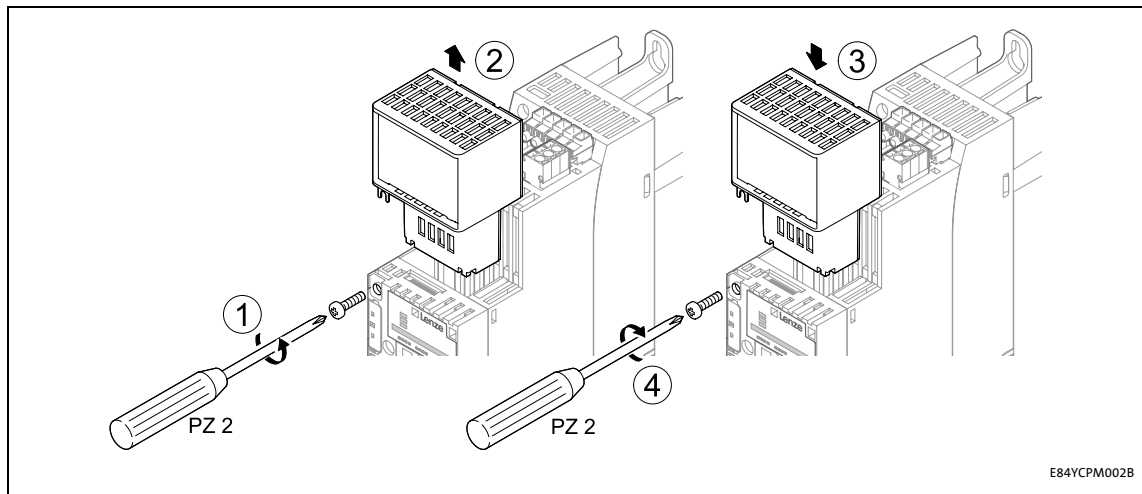


[5-2] Mounting for standard devices from 0.55 kW onwards

Mounting steps

1. Slightly impress the pressure surface of the top side of the MCI slot cover of the standard device (1).
2. Bend the cover forward and remove it from the standard device (2).
3. Loosen the securing screw for the communication module at the standard device (3).
4. Insert the communication module into the MCI slot of the standard device (4).
5. Tighten the securing screw again (5).

5.1.3 Replacing the communication module



[5-3] Replacing the communication module

Mounting steps

1. Loosen the securing screw for the communication module at the standard device (1).
2. Remove the communication module from the MCI slot of the standard device (2).
3. Insert the new communication module into the MCI slot of the standard device (3).
4. Tighten the securing screw again (4).

5.2

Electrical installation

Documentation for the standard device, control system, plant/machine

Observe the notes and wiring instructions stated.

5.2.1

EMC-compliant wiring

In typical systems, standard shielding is sufficient for Ethernet cables.

However, in environments with a very high level of interference, EMC resistance can be improved by earthing both sides of the cable shield as well.

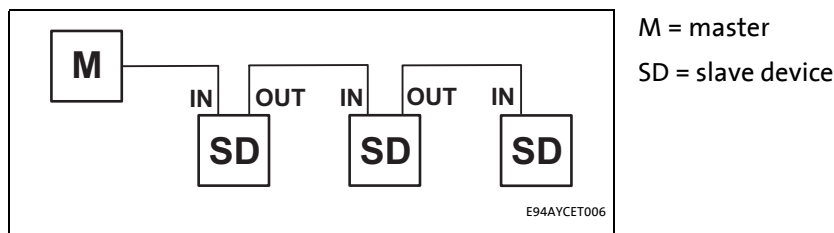
For this purpose, observe the following notes:

1. Remove the plastic sheath of the cable on a length of 2 cm.
2. Fasten the cable shield to the shield support of the basic device.

5.2.2 Network topology

An EtherCAT frame is sent through a pair of wires from the master to the slaves. The frame is forwarded from slave to slave until it has passed through all the devices. Finally, the last slave returns the frame to the master through a second pair of wires. In this way, EtherCAT always forms a logic ring topology, irrespective of the topology used.

Line topology



[5-4] Line topology

The devices are interconnected successively.

In order to ensure trouble-free operation, it is required to assign and wire the EtherCAT inputs (IN) and EtherCAT outputs (OUT) correctly.

The receiving line is plugged into socket X246 (IN), the forwarding line into socket X247 (OUT).

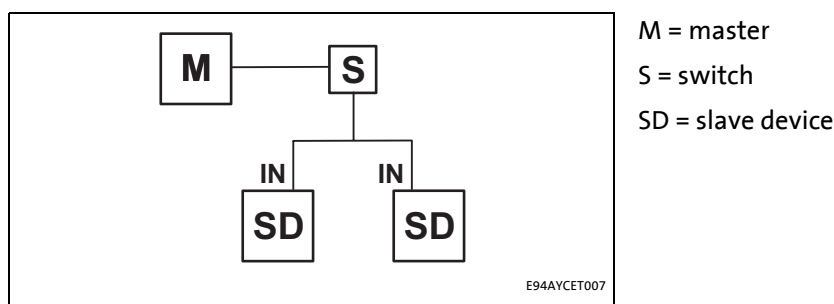
The direction of data transmission is from the master to the slaves.



Tip!

The termination of the last EtherCAT node is effected automatically by the slave.

Switch topology

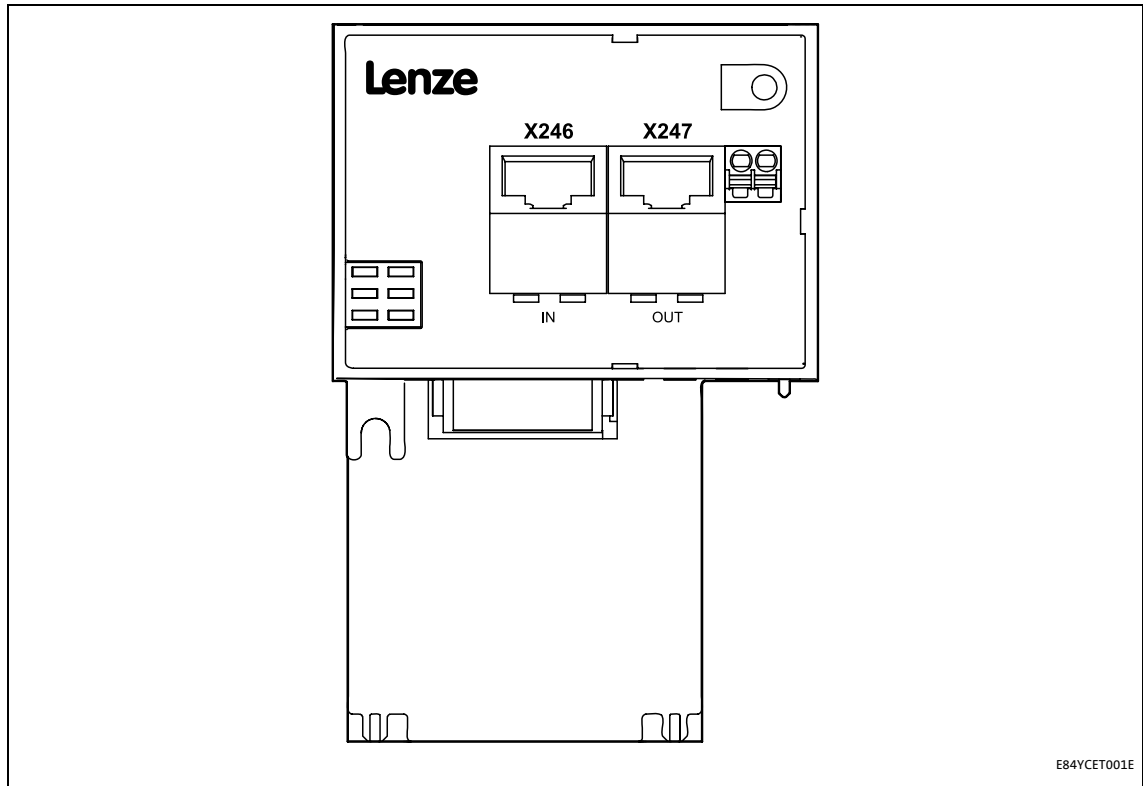


[5-5] Switch topology

The wiring can also be carried out in a star structure via an appropriate switch. For this, observe the additional runtimes.

5.2.3 EtherCAT connection

EtherCAT is connected via the RJ45 sockets **X246** (IN) and **X247** (OUT).



[5-6] EtherCAT connection

A standard Ethernet patch cable is suitable for connecting the communication module to the EtherCAT fieldbus.

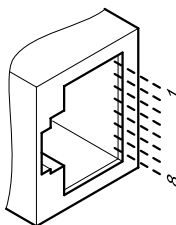
► [Specification of the Ethernet cable](#) (29)



Note!

In order to prevent damage to the RJ45 socket, plug or remove the Ethernet cable connector straight (at a right angle) into/from the socket.

Pin assignment of the RJ45 sockets

RJ45 socket	Pin	Signal
 E94AYCXX004C	1	Tx +
	2	Tx -
	3	Rx +
	4	-
	5	-
	6	Rx -
	7	-
	8	-



Tip!

The EtherCAT interfaces are provided with an auto MDIX function. This function adjusts the polarity of the RJ45 interfaces so that a connection can be established irrespective of the polarity of the opposite EtherCAT interface and irrespective of the type of cable used (standard patch cable or crossover cable).

5.2.4 Specification of the Ethernet cable

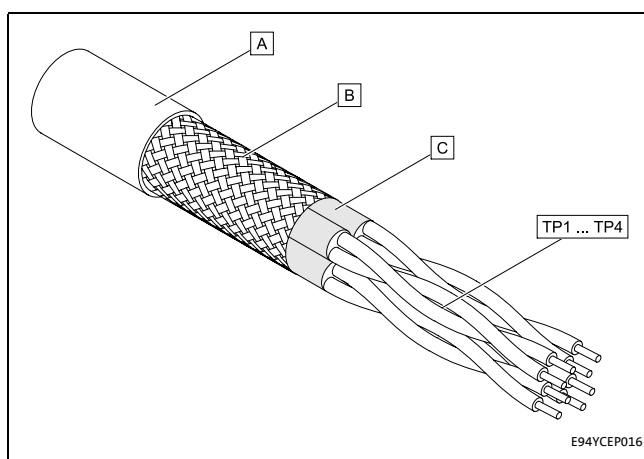


Note!

Only use cables that correspond to the given specifications.

Specification of the Ethernet cable	
Ethernet standard	Standard Ethernet (acc. to IEEE 802.3), 100Base-TX (Fast Ethernet)
Cable type	S/FTP (Screened Foiled Twisted Pair, ISO/IEC 11801 or EN 50173), CAT 5e
Damping	23.2 dB (for 100 MHz and 100 m each)
Crosstalk damping	24 dB (for 100 MHz and 100 m each)
Return loss	10 dB (100 m each)
Surge impedance	100 Ω

Structure of the Ethernet cable



A Cable insulation

B Braid

C Foil shielding

TP1 Twisted core pairs 1 ... 4

...

TP4

[5-7] Structure of the Ethernet cable (S/FTP, CAT 5e)

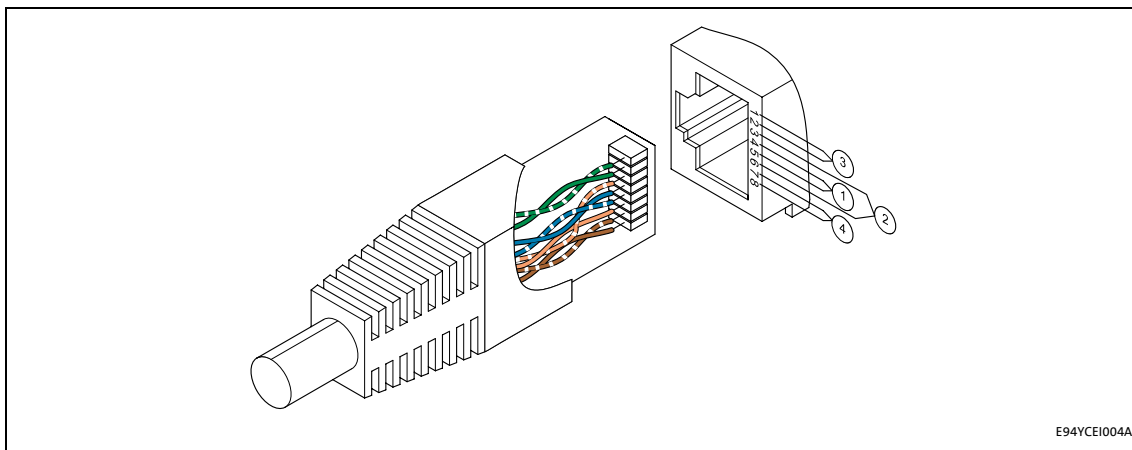
Colour code of the Ethernet cable



Note!

Wiring and colour code are standardised in EIA/TIA 568A/568B.

The use of 4-pole Ethernet cables according to industrial standard is permissible. The cable type only connects the assigned pins 1, 2, 3 and 6.



E94YCEI004A

[5-8] Ethernet plugs in accordance with EIA/TIA 568A/568B

Pair	Pin	Signal	EIA/TIA 568A	EIA/TIA 568B
3	1	Tx +	white / green	white / orange
	2	Tx -	green	orange
2	3	Rx +	white / orange	white / green
1	4		blue	blue
	5		white / blue	blue / white
2	6	Rx -	orange	green
4	7		white / brown	white / brown
	8		brown	brown

5.2.5 External voltage supply

The communication module can be supplied externally with voltage via separate supply cables at the 2-pole plug connector **X245**.



Note!

With external voltage supply, always use a separate power supply unit, safely separated to EN 61800-5-1 in every control cabinet (SELV/PELV).

External voltage supply of the communication module is necessary if the bus communication is to be continued in the event of a failure of the supply of the standard device.

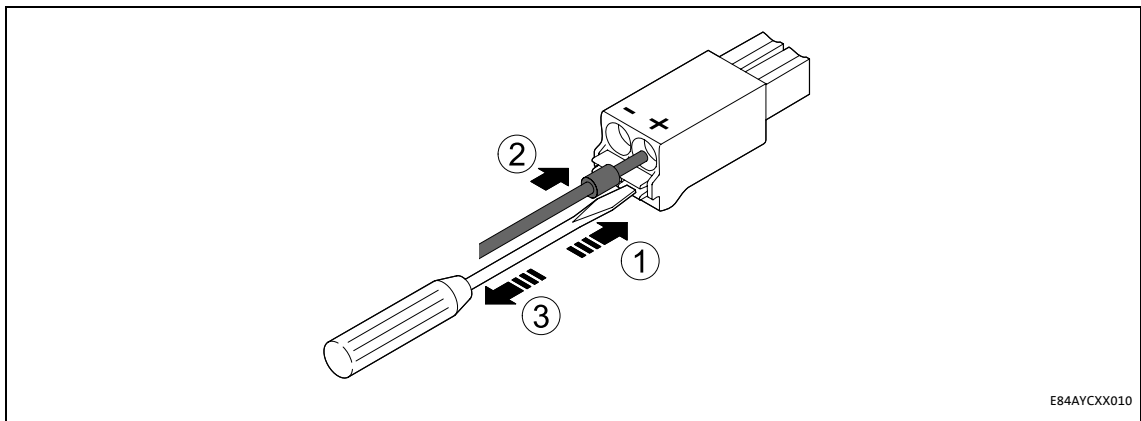
Access to parameters of a standard device disconnected from the mains is not possible.

Wiring plug connector X245



Stop!

Only wire the plug connector if the standard device is disconnected from the mains.



[5-9] Wiring of the 2-pole plug connector with spring connection

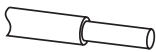
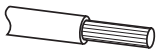
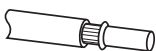
How to wire the plug connector with spring connection:

1. Place a screwdriver in the notch beneath the contact slot and keep it pressed.
2. Insert the supply cable in the contact slot.
3. Remove the screwdriver from the notch.

Assignment of the X245 plug connector

Designation	Description
+	U = 24 V DC (20.4 V - 0 % ... 28.8 V + 0 %) I = 140 mA
-	Reference potential for the external voltage supply

Terminal data

Area	Values
Electrical connection	2-pole plug connector with spring connection
Possible connections	Fixed:
	 0.2 ... 1.5 mm ² (AWG 24 ... 16)
	Flexible:
	 Without wire end ferrule 0.2 ... 1.5 mm ² (AWG 24 ... 16)
	 With wire end ferrule, without plastic sleeve 0.2 ... 1.5 mm ² (AWG 24 ... 16)
	 With wire end ferrule, with plastic sleeve 0.2 ... 1.5 mm ² (AWG 24 ... 16)
Stripping length	10 mm

6 Commissioning

6.1 Before initial switch-on

6 Commissioning

During commissioning, plant-specific data such as motor parameters, operating parameters, responses, and parameters for fieldbus communication are defined for the inverter. Lenze devices use codes for this purpose.

The codes of the inverter and for communication are saved to the memory module in a non-volatile data set.

In addition, there are codes for diagnosing and monitoring the stations.

▶ [Parameters of the communication module](#) (📖 97)

6.1 Before initial switch-on



Stop!

Before you switch on the Inverter Drive 8400 and the communication module for the first time, check the entire wiring for completeness, short circuit and earth fault.

6.2 Configuring the Controller (EtherCAT master)

The Controller (EtherCAT master) must be configured before communication with the communication module is possible.

In order to configure EtherCAT networks, you always need a configuration software for the Controller, e.g.:

- Lenze »PLC Designer«
- Beckhoff »TwinCAT«

These are software systems for the programming of control programs, EtherCAT configuration, real-time execution, and diagnostics.

The basic parameters of the communication module are stored in the internal configuration memory and can be used by the master for the node identification.

For the node search (fieldbus scan), the corresponding device descriptions of the Lenze device family are used.

6.2.1 Installing device description files

The current XML device description file required for configuring the EtherCAT node can be found in the download area at:

www.Lenze.com

The **Lenze_Inverter_8400_IO_yyyymmdd.xml** device description file has to be installed via the EtherCAT configuration software.

Wildcards in the file name	
yyyy	Year
mm	Month
dd	Day

The description file includes all EtherCAT capable devices of the Inverter Drives 8400 series (Inverter Drives 8400 with EtherCAT V01.xx and V02.xx, 8400 motec with EtherCAT V01.xx).

6.2.2 Automatic device identification

For a faultless integration of the EtherCAT slaves into a master configuration it is necessary to select the correct Lenze device in the EtherCAT configuration software.

Each EtherCAT node is identified unambiguously by the configuration software by means of the product code (equal to the CoE object I-1018.2), the manufacturer's identification mark (0x3B), and the main software version of the communication module.

► [Identification](#) (□ 12)

► [Implemented CoE objects](#) (□ 79)

In order that the configuration software select the configuration from the device description file specific for the EtherCAT node, the product code is automatically set in the identity object. An update is made after switching off/on the voltage supply.

During initialisation, the product code is transferred to the EtherCAT master. On the basis of this identification, the master can accept the corresponding settings from the device description.

Product codes for Inverter Drives 8400

The product code defines the following Inverter Drives 8400 in the device description files:

Product code [dec]						Meaning
8	4	0	0	2	2	Inverter Drive 8400 StateLine
8	4	0	0	2	3	Inverter Drive 8400 HighLine
8	4	0	0	2	4	Inverter Drive 8400 TopLine
8	4	0	0	2	5	Inverter Drive 8400 TopLine P (with CiA402)

6.2.3 Configuring process data

Inverter Drives 8400 support the configuration of max. 16 process data words (max. 32 bytes) per direction.

The process data configuration is determined during the initialisation phase of the master (PDO mapping).

The process data configuration is predefined in the device description file for each application.

The process data length can be adjusted by the user if required.

6.2.4 Determining the cycle time

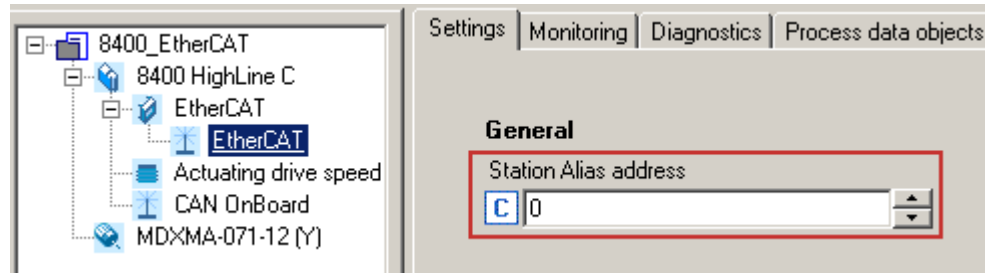
The process data objects (PDO) are transferred cyclically between the EtherCAT master and the slaves (inverters).

The cycle time is set with the EtherCAT configuration software.

6.3 Address allocation

The EtherCAT nodes are normally addressed via a fixed 16-bit address defined by the EtherCAT master. During start-up, the master assigns this address to each node, depending on the physical order in the EtherCAT network. The address is not saved and is lost when the device is switched off.

Via the **Station alias address** input field you can assign a fixed address to the EtherCAT slave.



Note!

- The station alias address must be unambiguous and may only be assigned once within the EtherCAT network.
- Use the same station alias address in the EtherCAT master and in the slave.

Valid address range: 0 ... 32767

- Address 0 means that no station alias address is assigned.
- Impermissible addresses are marked in red in the input field.
- The address is written to code [C13899](#).

In addition, specify the use of the fixed addressing on the master.

The address assigned by the master is displayed under code [C13864](#).

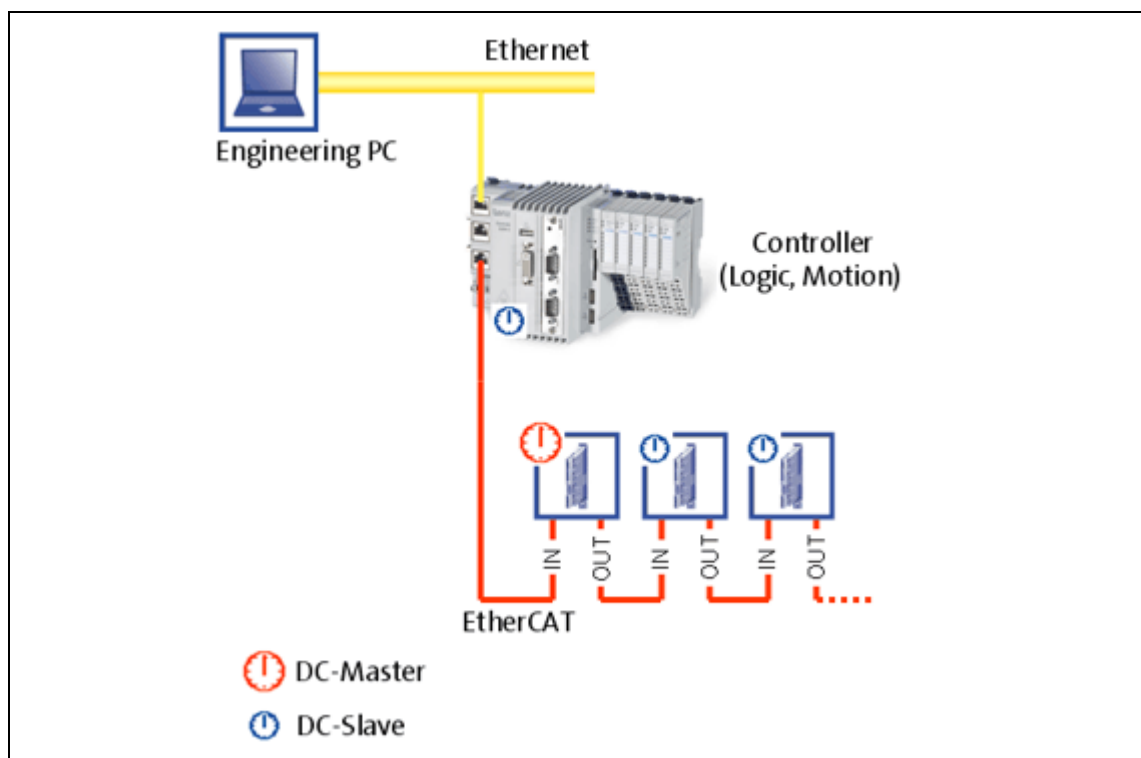
6.4 Synchronisation with "Distributed Clocks" (DC)

The "Distributed clocks" (DC) functionality enables exact time synchronisation for applications in which several axes perform a coordinated movement simultaneously. Data are incorporated synchronously with the PLC program. During DC synchronisation, all slaves are synchronised with a reference clock, the so-called "DC master".



Note!

- DC synchronisation is absolutely required for Motion applications.
- DC synchronisation can also be used for Logic applications.
- Not all slaves support the DC functionality.
- On order to be able to use the DC functionality, the first slave connected to the EtherCAT master (e.g. Lenze Controller) must have **DC master capability**. When further slaves are connected, DC-capable and non-DC-capable devices can be mixed.
- The first EtherCAT slave after the Lenze Controller must be the **DC master** that supplies the other EtherCAT nodes (incl. Controller) with the exact time.



[6-1] Example: "Distributed clocks" in the EtherCAT bus system with Lenze Controller 3231 C

The DC synchronisation is set with the EtherCAT configuration software.



"Control technology EtherCAT" communication manual

Here you can find some detailed information about the EtherCAT configuration and the commissioning of Lenze devices in the EtherCAT network.

6.4.1 DC configuration in the master

By default, the application of the DC synchronisation is deactivated in the device description ([34](#)). Parameterise the DC synchronisation in the EtherCAT configuration software («PLC Designer», »TwinCAT«).

Set the synchronisation cycle time in the master. It is mainly defined by the processing time of the master and the slaves.



Note!

The synchronisation cycle time ...

- must be an integer multiple of 1 ms;
- may be maximally 15 ms.

6.4.2 DC configuration in the Inverter Drive 8400 (slave)



Note!

The settings of the parameter sync cycle time (C01121), sync phase position (C01122), sync tolerance (C01123) and sync PLL increment (C01124) common for the Lenze system bus (CAN) cannot be made for EtherCAT. These values are automatically calculated by the EtherCAT communication module and set internally in the inverter.

In order to use the DC synchronisation in the Inverter Drive 8400, select the sync source by means of the standard device code [C01120](#):

Selection 4: MCI (synchronisation via MCI (communication module))

6.4.3 Response of the Lenze EtherCAT nodes during start-up

Code [C13883](#) indicates whether the DC synchronisation for the communication module has been activated.

If the DC synchronisation is used, the communication module only changes to the "Operational" state when the standard device has adapted its phase position to the DC signal. This process may take several seconds.



Note!

- If the communication module does not change to the "Operational" state, there might be an error in the configuration or in the EtherCAT wiring.
- The communication module compares the cycle time defined by the EtherCAT master to the internal processing time (1 ms) of the standard device. The synchronisation cycle time in the master must be identical with or an integer multiple of 1 ms.
- Furthermore it is checked whether the sync source selection in standard device code [C01120](#) is correct.
- Further information can be found in the status information or emergency messages of the master.

The state of the standard device synchronicity is displayed under code [C13884](#).

6.5 Establishing an online connection with the »Engineer«

With the »Engineer« you can establish an online connection to the individual field devices.

When an online connection has been established, you can for instance carry out parameter settings directly in the field device or diagnose the field device.



Stop!

If parameters in the »Engineer« are changed while the Engineer is connected online to the field device, the changes are directly accepted to the device!



Note!

To go online, the EtherCAT bus at least has to be in the "Pre-Operational" state.

The functions for establishing/cancelling an online connection in the »Engineer« can be executed via the **Online** menu:

Menu command	Shortcut
Online → Go online	<F4>
Online → Set communication path and go online	
Configuring a bus connection: • Gateway Controller -> configure EtherCAT (□ 41) • Gateway Controller -> configure EtherCAT ADS (Beckhoff) (□ 43)	
Online → Go offline	<Shift>+<F4>

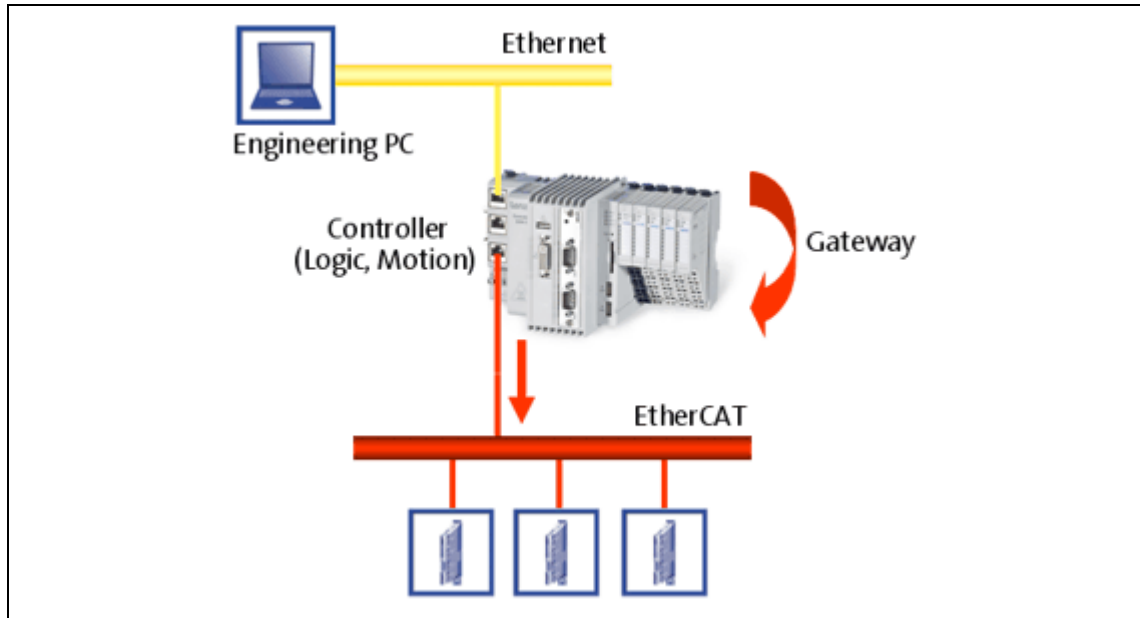


Documentation for the Lenze »Engineer«

Here you'll find further detailed information about how to establish an online connection.

6.5.1 Gateway Controller -> configure EtherCAT

The Lenze Controller provides a gateway function to establish an online connection to a field device via EtherCAT.



[6-2] Example: EtherCAT bus system with a Lenze Controller 3231 C as gateway



How to configure an online connection to a field device which is connected to the Lenze Controller via EtherCAT:

1. Go to the *Communication path* dialog box and the *Bus connection* list field, and select the entry "Gateway Controller -> EtherCAT" there.
2. Click **Search/Enter....**

The *Gateway Controller -> Set up EtherCAT bus* dialog box is shown:

IPC name	IP address	IPC interface
EL 870	172.31.207.20	ECAT1

IPC (Gateway)

Name:

IP address:

Interface:

Field device

Name:

Ethercat address:

Diagram: Ethernet --- Gateway device --- EtherCAT --- Field device

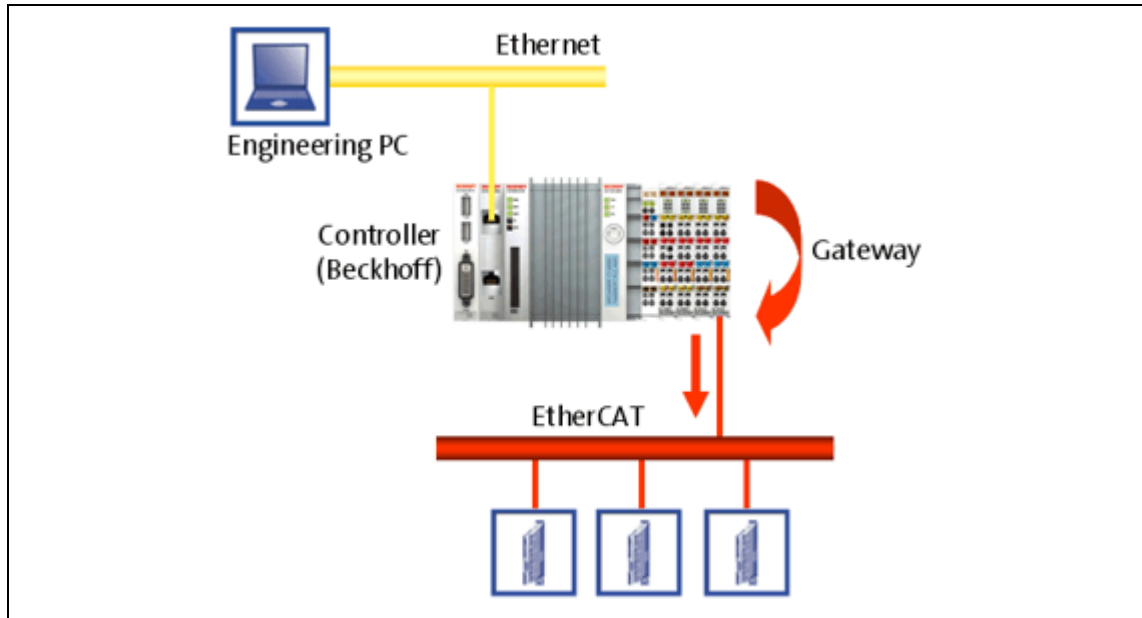
3. Enter the **IP address** of the Controller.

By clicking the **Ping** button, you can carry out a simple test which verifies whether a device can actually be reached via the IP address set

4. Click **OK**.
 - The *Enter IP address* dialog box is closed.
 - In the *Communication path* dialog box in the **Device access path** column, the corresponding device access path is shown (e.g. "IPC:172_31_207_254.ECAT.ecat1.dev1001.").

6.5.2 Gateway Controller -> configure EtherCAT ADS (Beckhoff)

The **Gateway EtherCAT ADS** bus connection makes it possible to establish an online connection to a Lenze inverter that is connected to a Beckhoff Controller via EtherCAT (gateway function).



[6-3] Example: EtherCAT bus system with a Beckhoff Controller as Gateway



How to configure an online connection to a field device which is connected to a Beckhoff Controller via EtherCAT:

1. Highlight the project root  in the project.
Alternatively: Create a new project or carry out a fieldbus scan.
2. Execute the menu command **Insert → Insert device detected online**.
3. Select **Gateway Controller -> EtherCAT ADS** as bus connection.
4. Configure access data:
 - Configure the access data applicable to the Controller via the **Insert address** button.
 - The **Search** button initiates the Controller to display the fieldbus nodes connected to the EtherCAT segment.



How to use the EtherCAT ADS communication path:

1. Highlight the desired inverter, to which a gateway connection via EtherCAT ADS is to be established, in the project tree.
2. Call the menu command **Online → Set communication path and go online**.

Device	Bus connection	Device access path	Type co
9400 HighLine~1	Gateway Controller -> ...	ADS: /192.168.0.1/192.168.0.1...	E94AFH

Project path: /New_Project/9400 HighLine~1

1 Bus connection: Gateway Controller -> EtherCAT ADS

2 User name: Name
 Password: 12345678
 IP address: 192.168. 0 . 1
 AMS Net ID: 192.168. 0 . 1 . 1 . 1

3 Address: 1001 Search/Enter...

Accept Connect Cancel

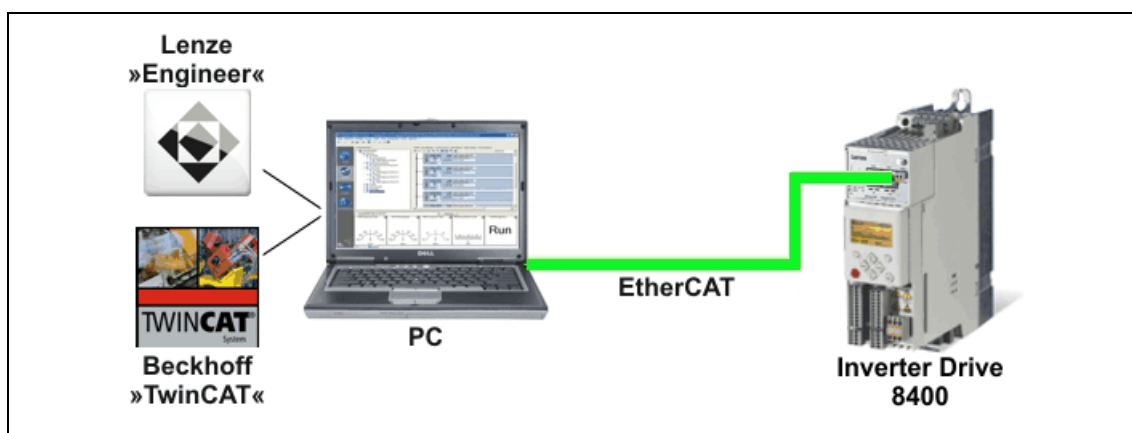
Help
 Please select the required device in the table, specify the bus connection and select the device using the "Find/Enter" button...

3. Select **1 Gateway Controller -> EtherCAT ADS** as bus connection.
4. Enter the access data applicable to the Controller in area **2**.
 Enter the user name, password, and the IP address and the AMS Net ID of the EtherCAT interface of the Controller.
5. In area **3**, specify the EtherCAT address of the field device to which the online connection is to be established.
 Alternatively you can click the **Search/Enter** button which calls the *Select Device Access Path* dialog window. By this, the »Engineer« initiates the Controller to display the devices detected on the EtherCAT segment.

6.6 EtherCAT ADS communication parameters in »TwinCAT« and »Engineer«

In the following, two example structures are used to describe where you can find the EtherCAT ADS communication parameters in the Beckhoff »TwinCAT« and in the Lenze »Engineer« EtherCAT.

6.6.1 Example: Structure without a Beckhoff Controller

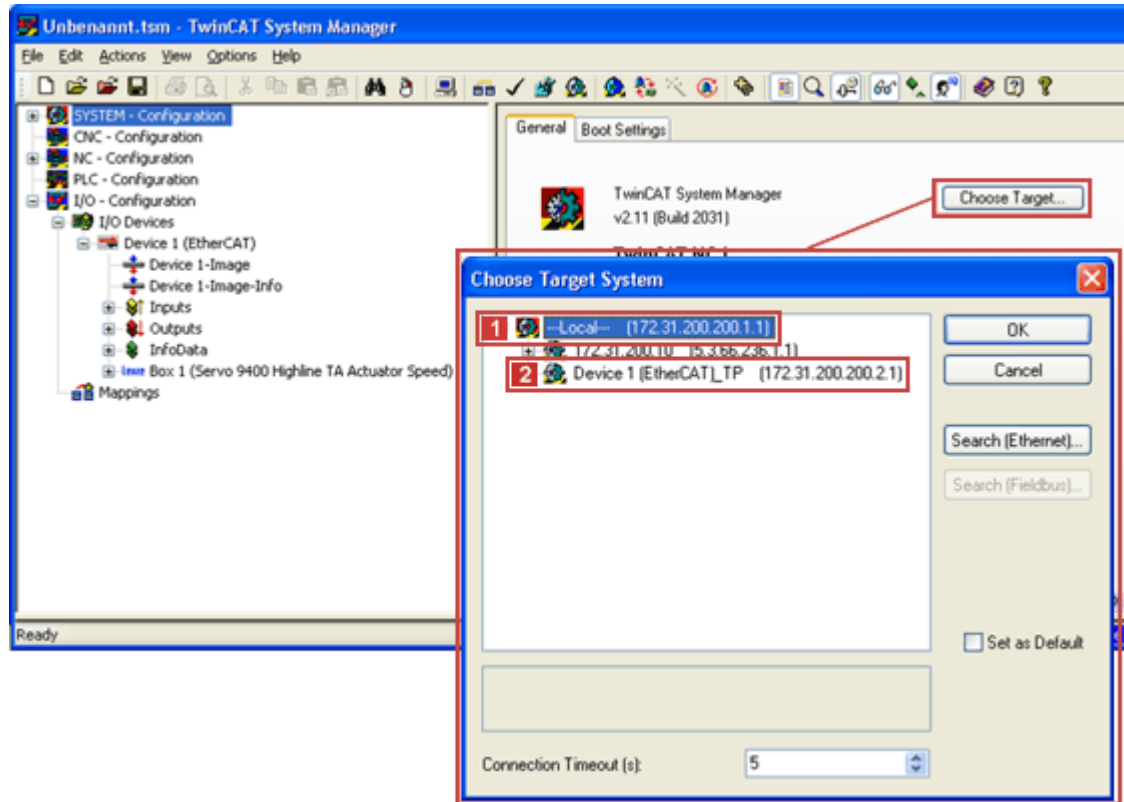


[6-4] Example: EtherCAT bus system without Beckhoff Controller

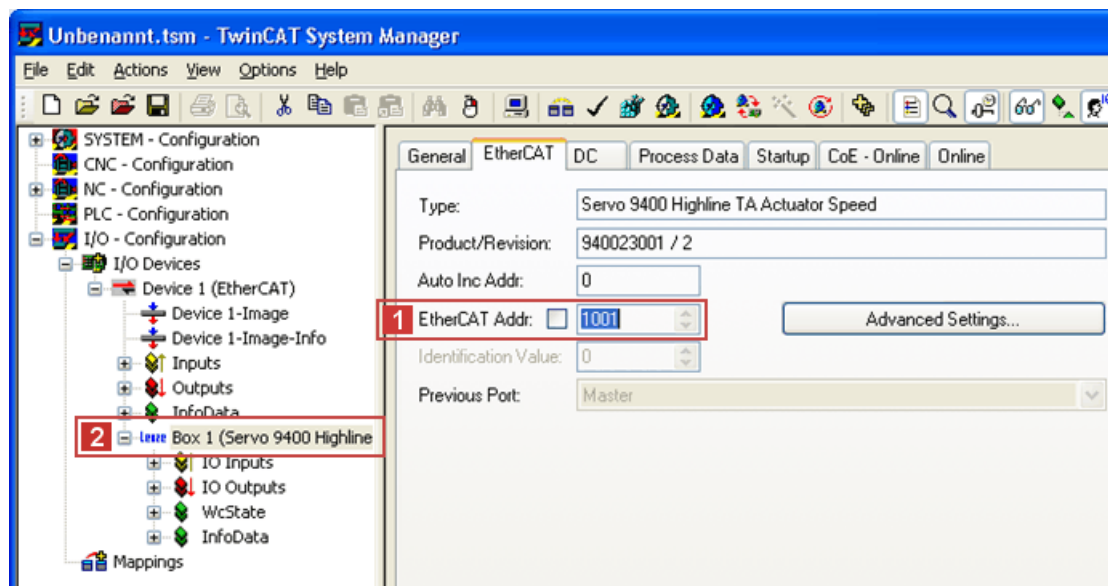
The Beckhoff Soft-PLC runs on the Microsoft Windows XP PC on which the Beckhoff »TwinCAT« and the Lenze »Engineer« are installed as well.

Display of the communication parameters in »TwinCAT«

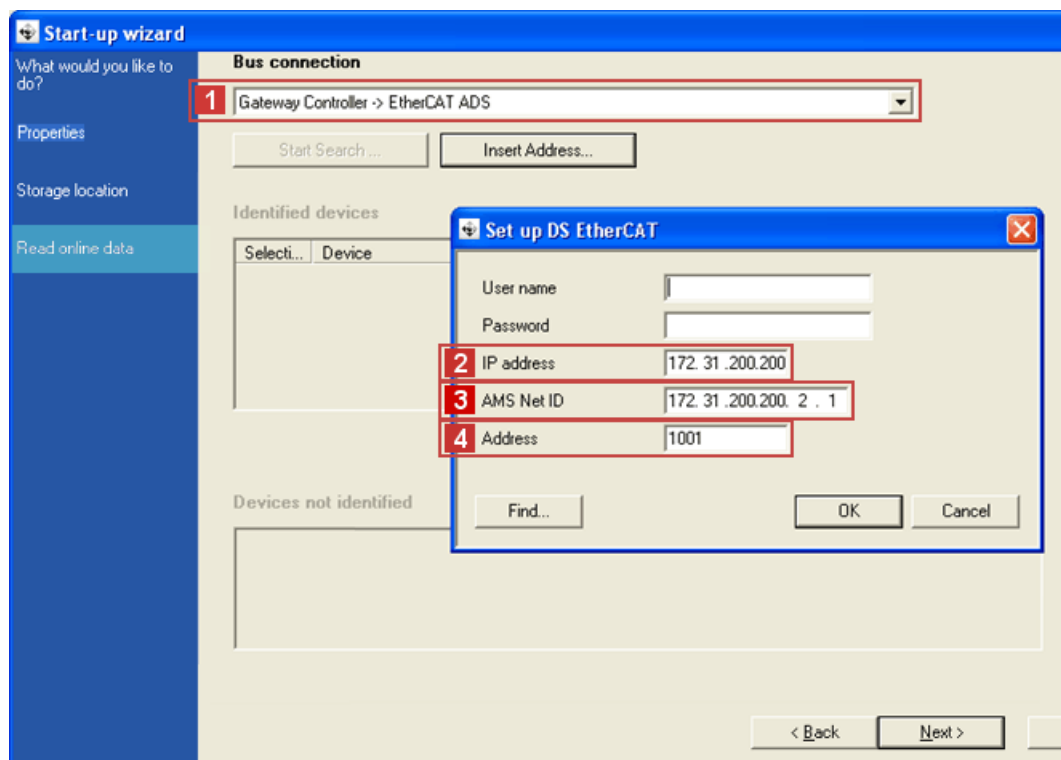
The communication parameters **1** IP address (here '172.31.200.200') and **2** EtherCAT Master Net ID (here '172.31.200.200.2.1') can be found under the target system selection:



The EtherCAT **1** Slave address (here '1001') can be found under the **EtherCAT** tab of the **2** EtherCAT slave:



Online device identification in the »Engineer« start-up wizard

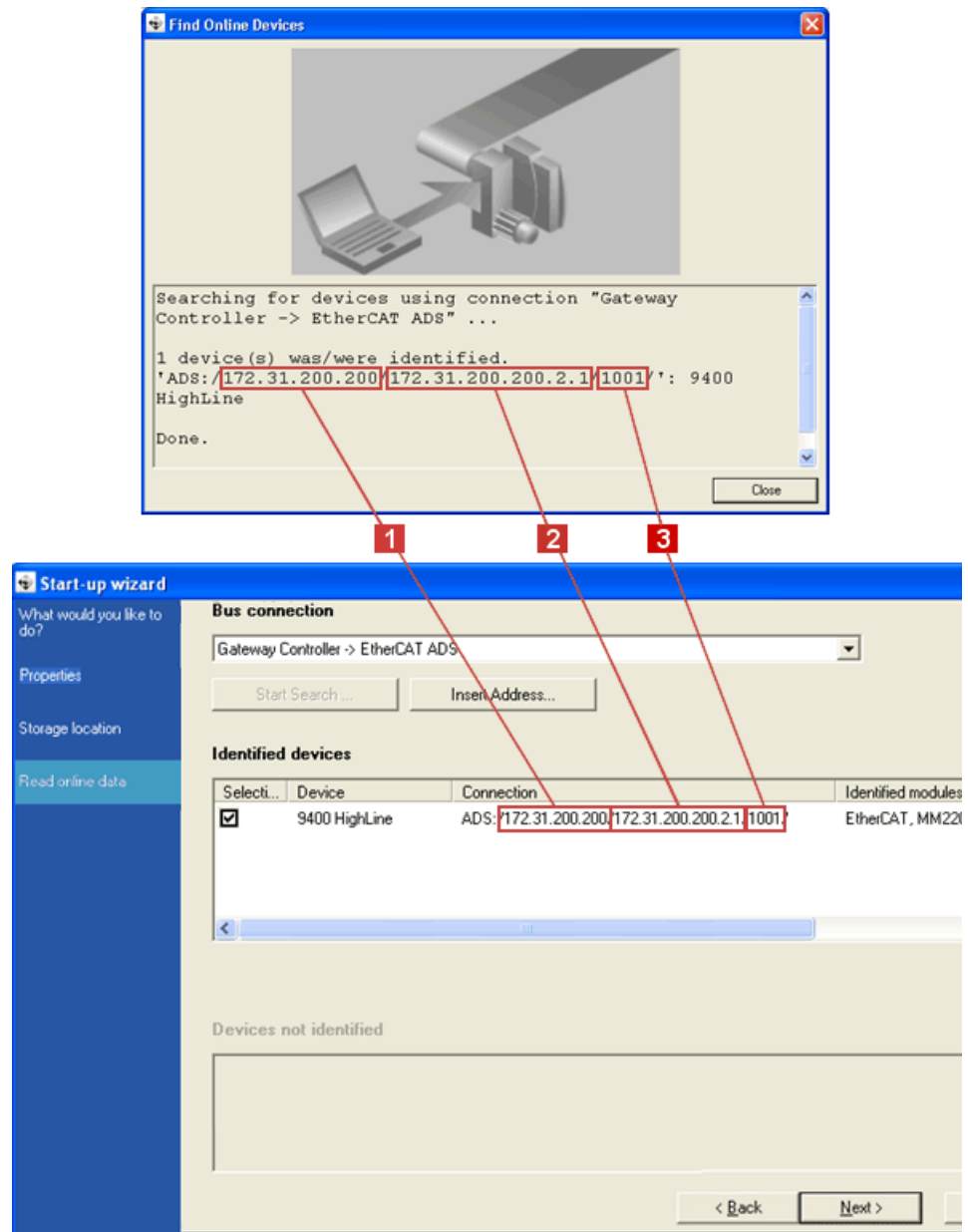


In the »Engineer« start-up wizard under the **1 Gateway Controller -> EtherCAT ADS** bus connection, a field device was detected online:

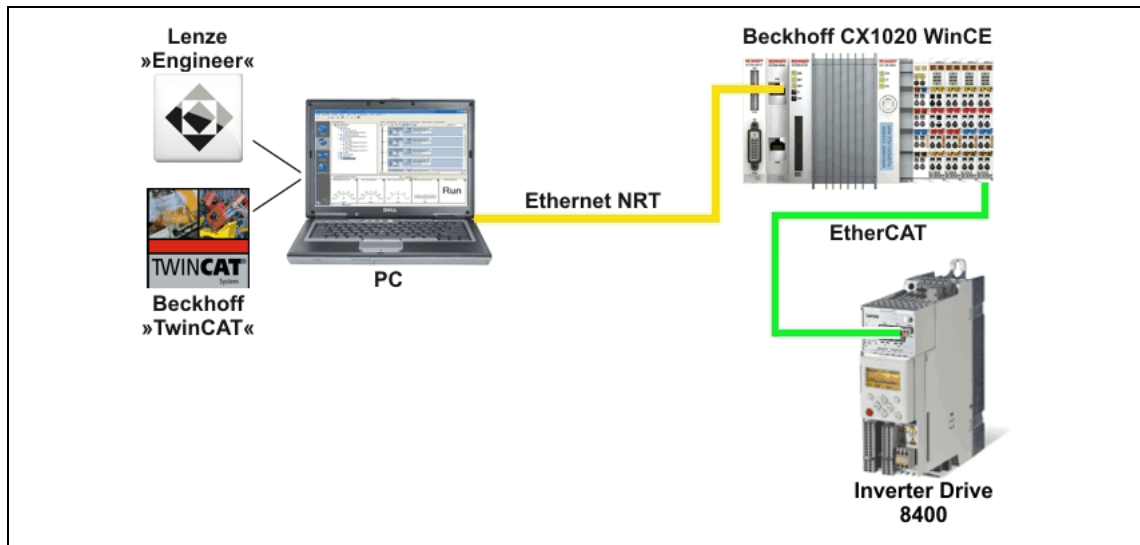
- 2 IP address:** 172.31.200.200
- 3 EtherCAT Net ID:** 172.31.200.200.2.1
- 4 EtherCAT slave address:** 1001

Display of the identification of the field device detected in the »Engineer« start-up wizard:

- 1** IP address: 172.31.200.200
- 2** EtherCAT Net ID: 172.31.200.200.2.1
- 3** EtherCAT slave address: 1001



6.6.2 Example: Structure with a Beckhoff DIN rail IPC CX1020

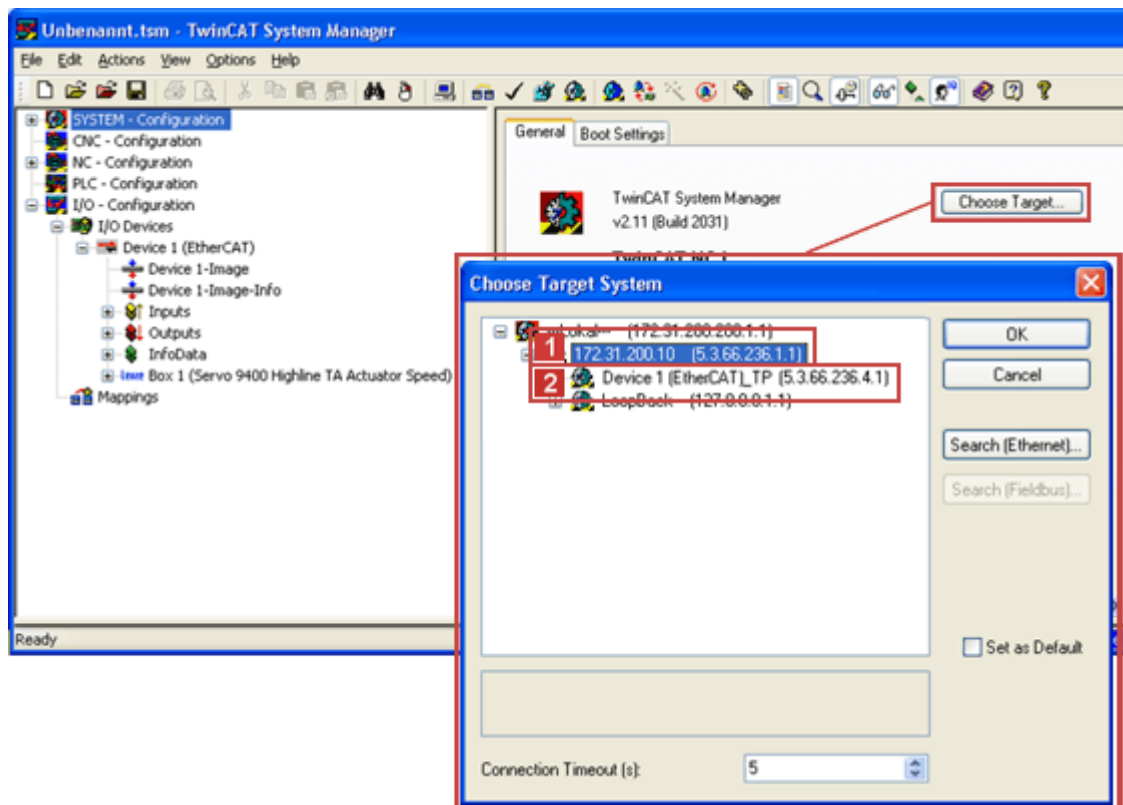


[6-5] Example: EtherCAT bus system with Beckhoff Controller

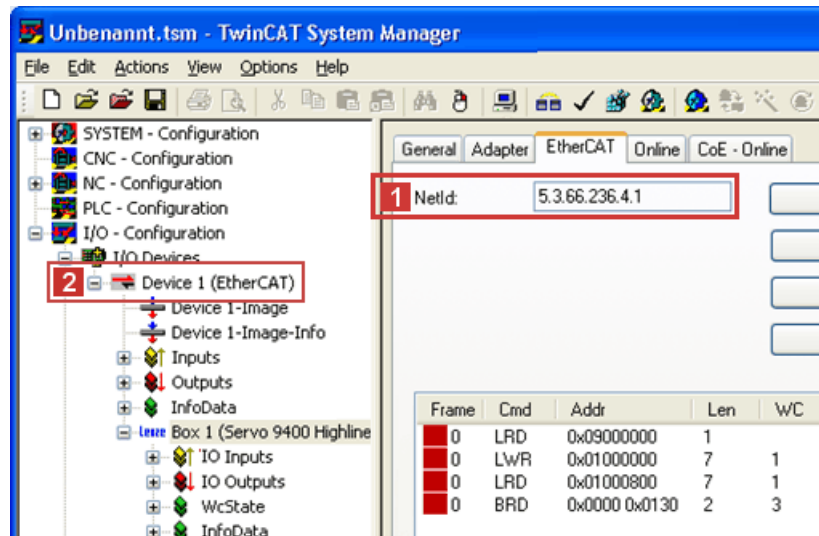
A Beckhoff DIN rail IPC with the Microsoft Windows CE operating system is used. The Beckhoff »TwinCAT« and the Lenz »Engineer« are installed on a Windows XP PC.

Display of the communication parameters in »TwinCAT«

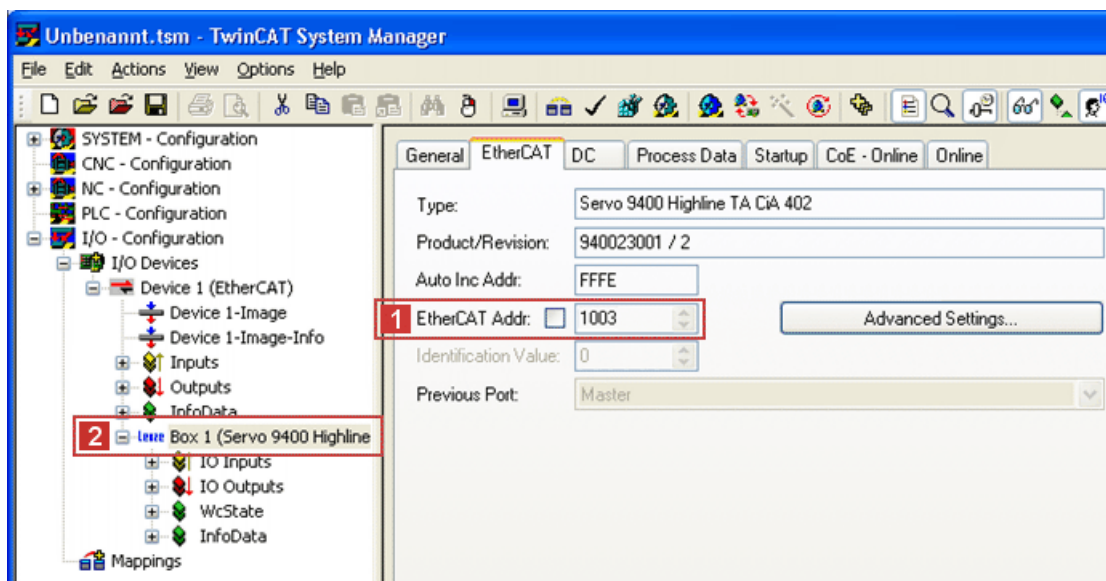
The communication parameters **1** IP address (here '172.31.200.10') and **2** EtherCAT Master Net ID (here '5.3.66.236.4.1') can be found under the target system selection:



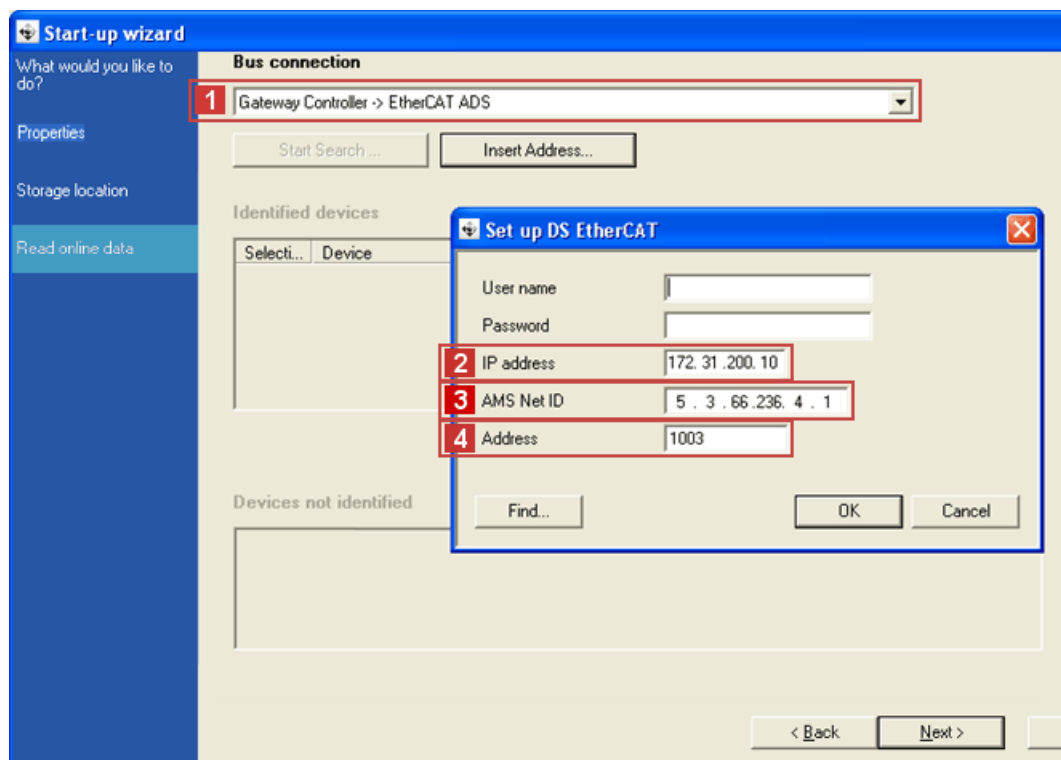
The **1** **EtherCAT Master Net ID** (here '5.3.66.236.4.1') can also be found under the **EtherCAT** tab of the **2** EtherCAT master:



The EtherCAT **1** **slave address** (here '1003') can be found under the **EtherCAT** tab of the **2** EtherCAT slave:



Online device identification in the »Engineer« start-up wizard



In the »Engineer« start-up wizard under the **1 Gateway Controller -> EtherCAT ADS** bus connection, a field device was detected online:

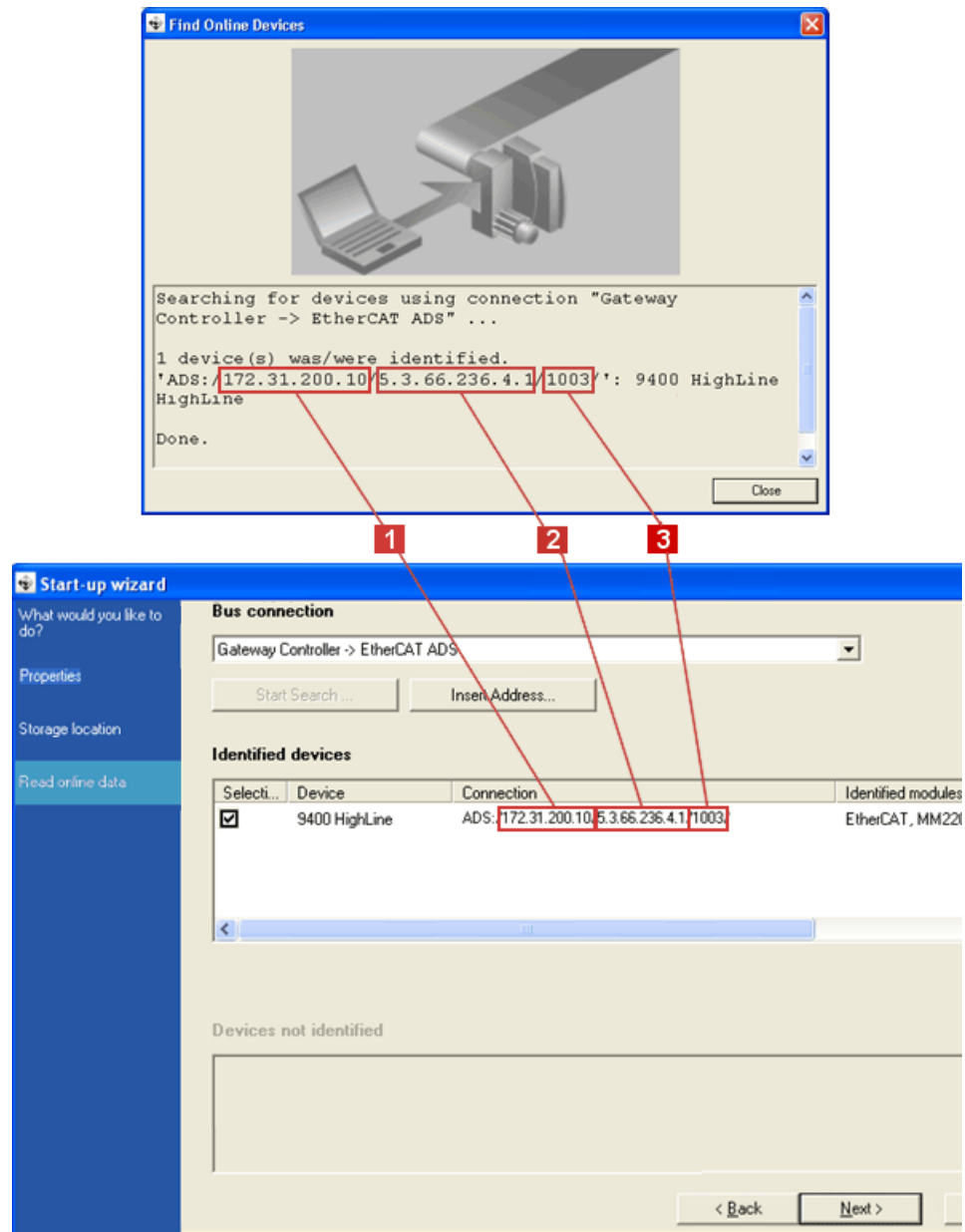
2 IP address: 172.31.200.10

3 EtherCAT Net ID: 5.3.66.236.4.1

4 EtherCAT slave address: 1003

Display of the identification of the field device detected in the »Engineer« start-up wizard:

- 1** IP address: 172.31.200.10
- 2** EtherCAT Net ID: 5.3.66.236.4.1
- 3** EtherCAT slave address: 1003



6.7

Initial switch-on

**Documentation for the standard device**

Observe the safety instructions and residual hazards stated.

**Note!****Establishing communication**

In order to establish communication via an externally supplied communication module, the standard device must be switched on as well.

After communication has been established, the externally supplied module operates independently of the power on/off state of the standard device.

Activating changed settings

In order to activate changed settings ...

- execute the device command "11: Save all parameter sets" via the standard device code **C00002** and ...
- then execute a "reset node" of the node or switch off the voltage supply of the communication module and switch it on again.

Protection against uncontrolled restart

After a fault (e.g. short-time mains failure), the restart of a drive is not always wanted and - in some cases - even not allowed.

In the Lenze setting for Inverter Drives 8400, the restart protection is active.

Via the standard device code **C00142** ("Autostart option") you can set the restart behaviour of the inverter:

C00142 = 9 (Lenze setting)

- The inverter remains inhibited (even if the fault is no longer active).
- Bit 0 (inhibit if device is ON) and bit 3 (inhibit in case of undervoltage) are set.
- An explicit inverter enable causes the drive to start up in a controlled manner: LOW-HIGH edge at digital input X4/RFR.

C00142 = 8 (enabled)

- In order to directly enable the device at switch-on, bit 0 must be set to zero (FALSE).
- An uncontrolled restart of the drive is possible.

7 Data transfer

Compared with conventional Ethernet, the collision-free transfer of frames on the fieldbus makes EtherCAT a real-time capable bus system.

Communication is always initiated by the EtherCAT master, e.g. a Lenze Controller. A frame sent by the master passes through all EtherCAT slaves. The last slave of the communication chain sends the frame back to the EtherCAT master. On the way back, the frame is directly sent to the master, without being processed in the slaves.

EtherCAT transmits data in so-called "EtherCAT frames". The EtherCAT nodes only extract the data intended for them while the EtherCAT frame passes through the device. At the same time output data are inserted into the frame while it passes through the device. Read and write accesses are only executed on a small section of the entire EtherCAT frame – the datagrams. Therefore it is not necessary to receive the complete frame before it can be processed. Processing starts as soon as possible.

EtherCAT transmits process data, parameter data, configuration data, and diagnostic data between the EtherCAT master and the inverters (slaves) that are part of the fieldbus. The data are transmitted via corresponding communication channels depending on their time-critical behaviour (see [Process data transfer](#) (□ 59) / [Parameter data transfer](#) (□ 68)).

7.1 EtherCAT-Frames

EtherCAT frames have the following structure:

Ethernet header			Ethernet data				FCS
48 bits	48 bits	16 bits	11 bits	1 bit	4 bits	48 ... 1498 bytes	32 bits
Destination	Source	EtherType	Frame header			Datagrams	
			Length	Reserved	Type		

Ethernet header

The Ethernet header contains the following information:

- Target address of the EtherCAT frame (destination)
- Source address of the EtherCAT frame (source)
- Type of the EtherCAT frame (EtherType = 0x88A4)

Ethernet data

The Ethernet data contain the following information:

- Length of the datagrams within the EtherCAT frame (Length)
- One reserved bit (Reserved)
- Type of the datagrams within the EtherCAT frame (Type)
- EtherCAT datagrams (Datagrams)

FCS

Checksum of the EtherCAT frame

7.2 EtherCAT datagrams

Read and write accesses are always only executed in one small section of the complete EtherCAT frame – the datagrams.

EtherCAT datagrams have the following structure:

EtherCAT Command header	Data	WKC
10 bytes	Max. 1486 bytes	2 bytes

EtherCAT Command header

The EtherCAT command header contains the following information:

- Command to be executed
- Addressing information
- Length of the data area (Data)
- Interrupt field

Data

The data area contains the data of the command to be executed.

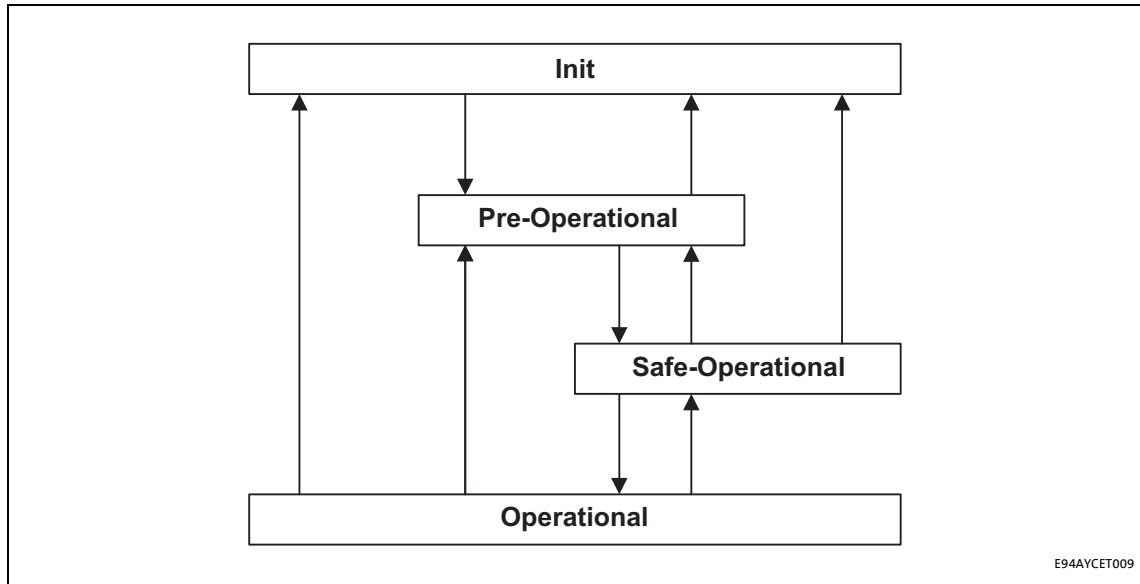
WKC (Working Counter)

The working counter is evaluated by the master for monitoring the execution of the command.

7.3

EtherCAT state machine

Before communication is possible via EtherCAT, the fieldbus passes through the EtherCAT state machine during start-up. The following illustration depicts the possible state changes from the point of view of an EtherCAT slave:



[7-1] EtherCAT state machine

State	Description
Init	- Initialisation phase - No SDO/PDO communication with the slaves - Device detection possible by means of a fieldbus scan
Pre-operational	- The fieldbus is active. - SDO communication (mailbox communication) is possible. - No PDO communication
Safe-operational	- SDO communication (mailbox communication) is possible. - PDO communication: - The input data in the process image are updated. - The output data from the process image are not transferred to the slaves.
Operational	Normal operation: - SDO communication - PDO communication - Fieldbus synchronisation successful (if used)



Note!

- A fieldbus scan can be carried out during any EtherCAT status.
- SDO communication via the EtherCAT bus is only possible if at least the "Pre-Operational" state has been reached.
- Only in the transitional phases between states can bus nodes be in different states.

The current status of the EtherCAT state machine is shown in [C13861](#) and indicated via the **BS** LED.

Possible errors at the state transitions are shown in [C13879](#). Additionally, an error message is entered in the "AL status code" EtherCAT register" ([see 58](#)).

-
- ▶ [Diagnostics with the »Engineer«](#) (📖 89)
 - ▶ [Fieldbus status displays](#) (📖 86)

AL Status Code

Information about how the "AL Status Code" EtherCAT register (address 0x0134:0x0135) can be accessed can be found in the documentation of the EtherCAT master.

These error messages can be entered in the "AL Status Code" register:

AL Status Code [hex]	Description
0x0000	No fault
0x0011	Invalid status change requested
0x0012	Unknown status requested
0x0013	"Bootstrap" status is not supported
0x0016	Invalid mailbox configuration "Pre-operational"
0x001A	Synchronisation error
0x001B	Sync manager watchdog
0x001D	Invalid output data configuration
0x001E	Invalid input data configuration
0x002B	Invalid input and output data
0x0030	Invalid configuration of DC synchronisation
0x9001	Firmware watchdog error
0x9002	Mapping error

8 Process data transfer

Process data are transmitted by means of [EtherCAT datagrams](#) (□ 56) via the process data channel.

The Inverter Drive 8400 is controlled by means of the process data.

The transmission of process data is time-critical.

Process data are transferred cyclically between the Controller (EtherCAT master) and the inverters (slaves) (continuous exchange of current input and output data).

The master can directly access the process data. In the PLC for instance, the data are directly stored in the I/O area.

Up to 16 process data words (16 bits/word) per direction can be exchanged.

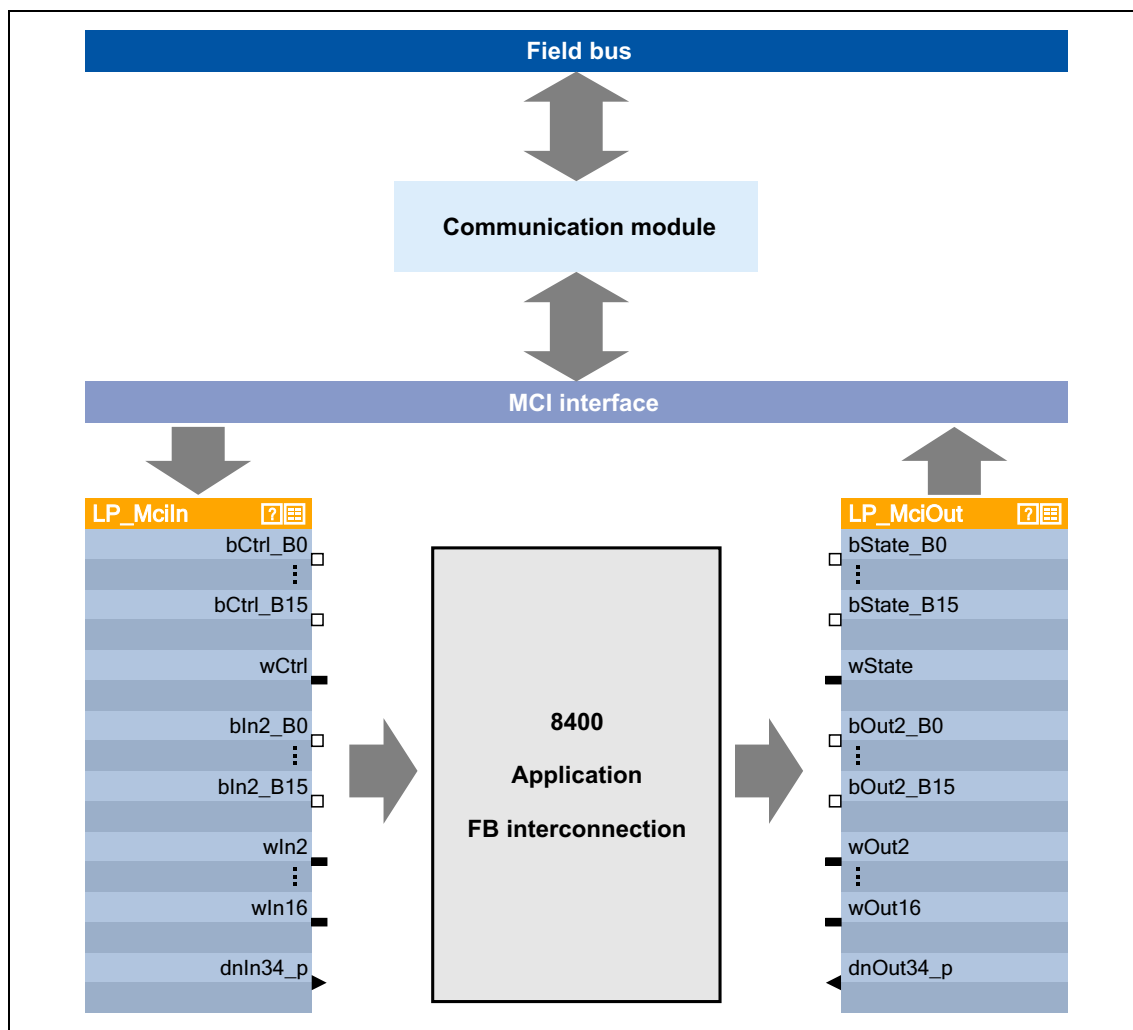
Process data are not saved in the Inverter Drive 8400.

Process data are for instance setpoints, actual values, control words, and status words.

8.1 Accessing process data / PDO mapping

Process data (MCI PDO) are transmitted via the MCI interface.

- Max. 16 words are exchanged per direction.
- The process data are accessed via the port blocks **LP_MciIn** and **LP_MciOut**. These port blocks are also referred to as process data channels.
- The port/function block interconnection of the process data objects (PDO) is made via the Lenze »Engineer«.



[8-1] Outer and inner data transfer between bus system, inverter, and application



Software manual / online help for the Inverter Drive 8400

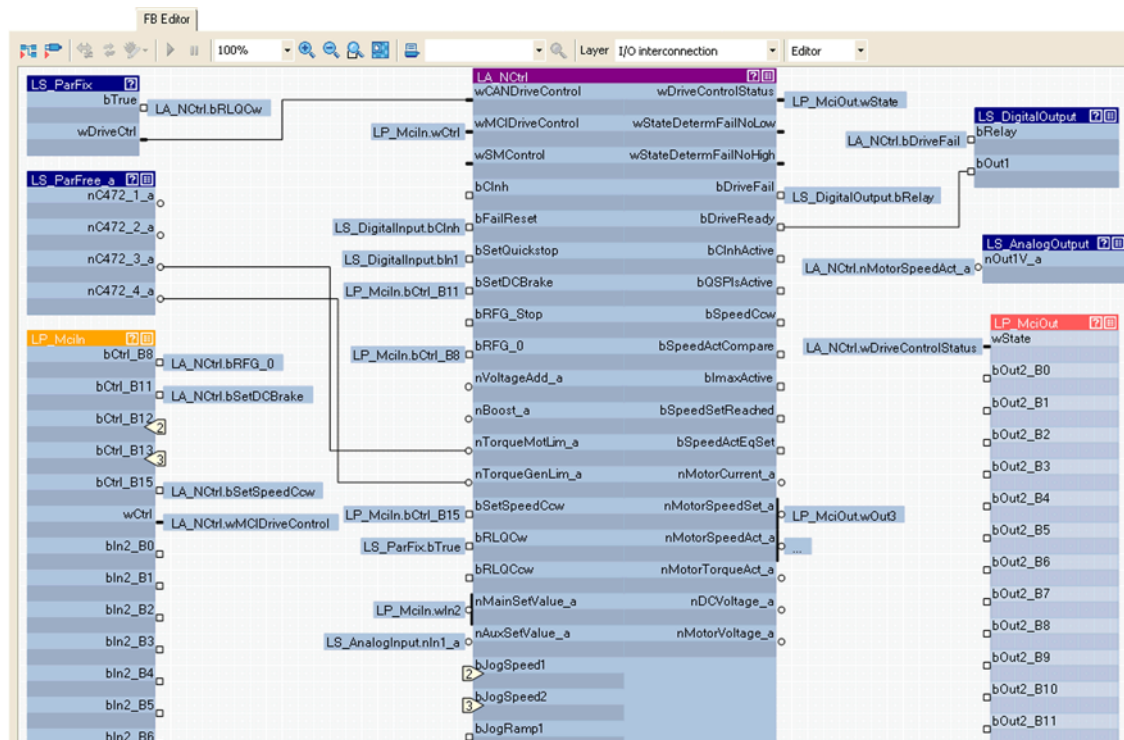
Here you can find detailed information on the port/function block interconnection in the »Engineer« and on port blocks.

8.2

Preconfigured port interconnection of the process data objects (PDO)

The preconfigured port interconnection of the process data objects can be activated by setting the standard device code **C00007 = 40: MCI**.

The »FB Editor« serves to display the port blocks "LP_MciIn" and "LP_MciOut" with the preconfigured interconnections:



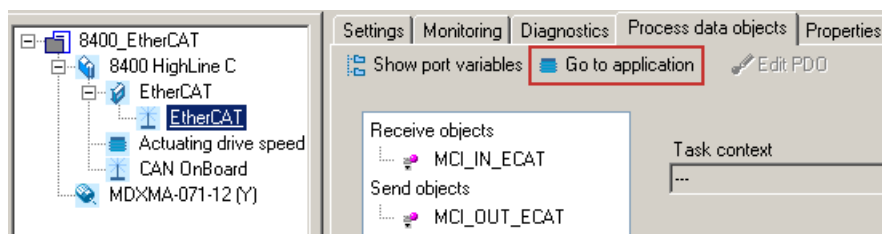
8.3

Free configuration of the port interconnection of the process data objects (PDO)

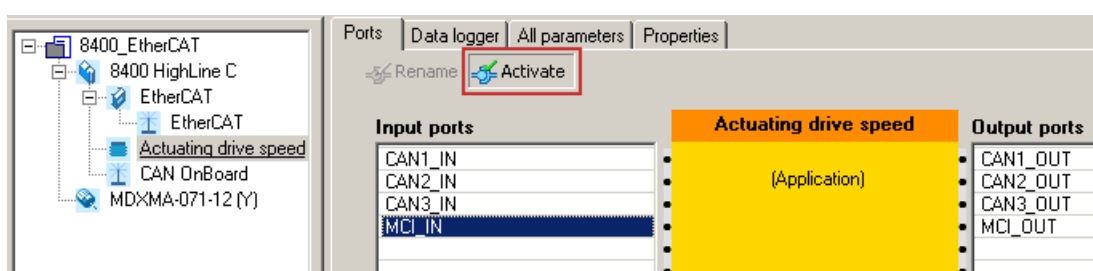


How to configure the port interconnection in the »Engineer«:

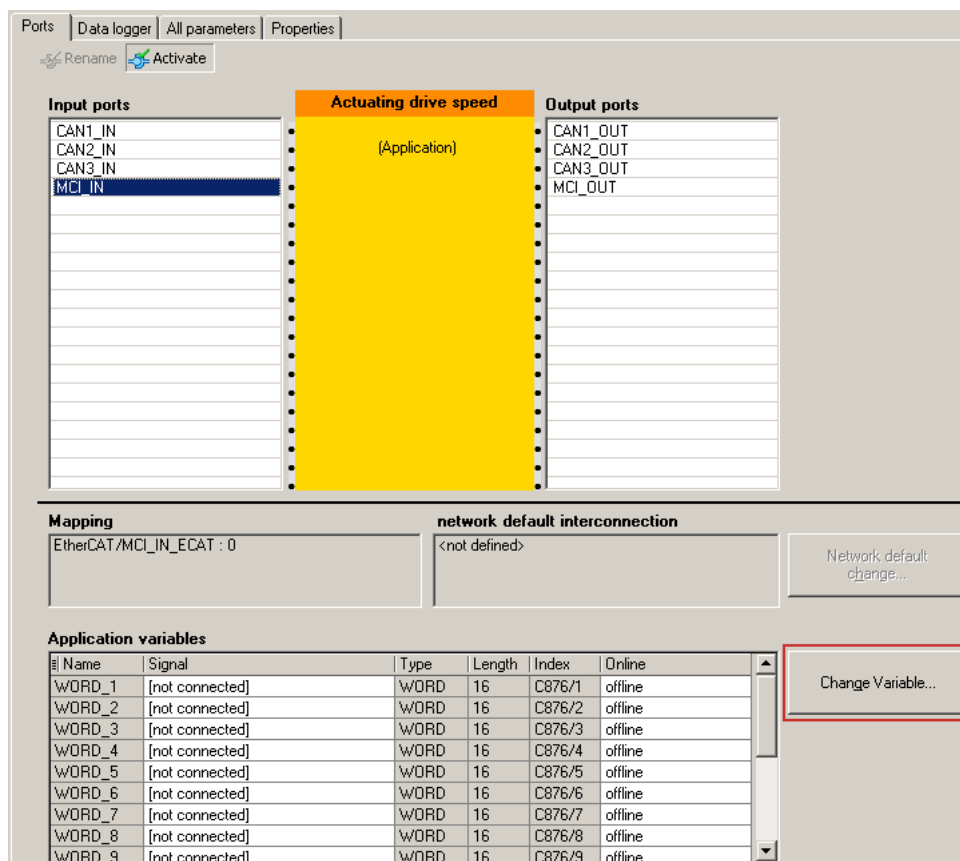
1. Under the **Process data objects** tab, click on the **Go to application** button.



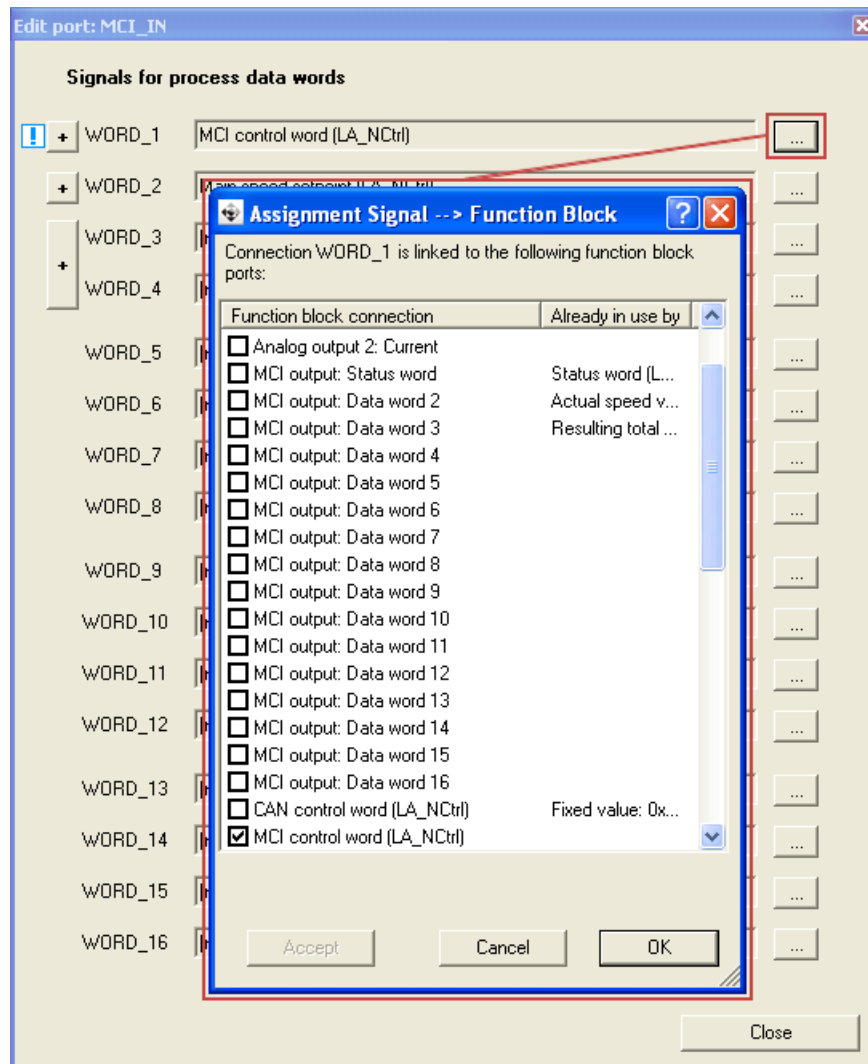
2. Under the **Ports** tab, select port block "MCI_IN" or "MCI_OUT" by mouse-click and activate it via the **Activate** button.



3. Click the **Change Variable...** button.

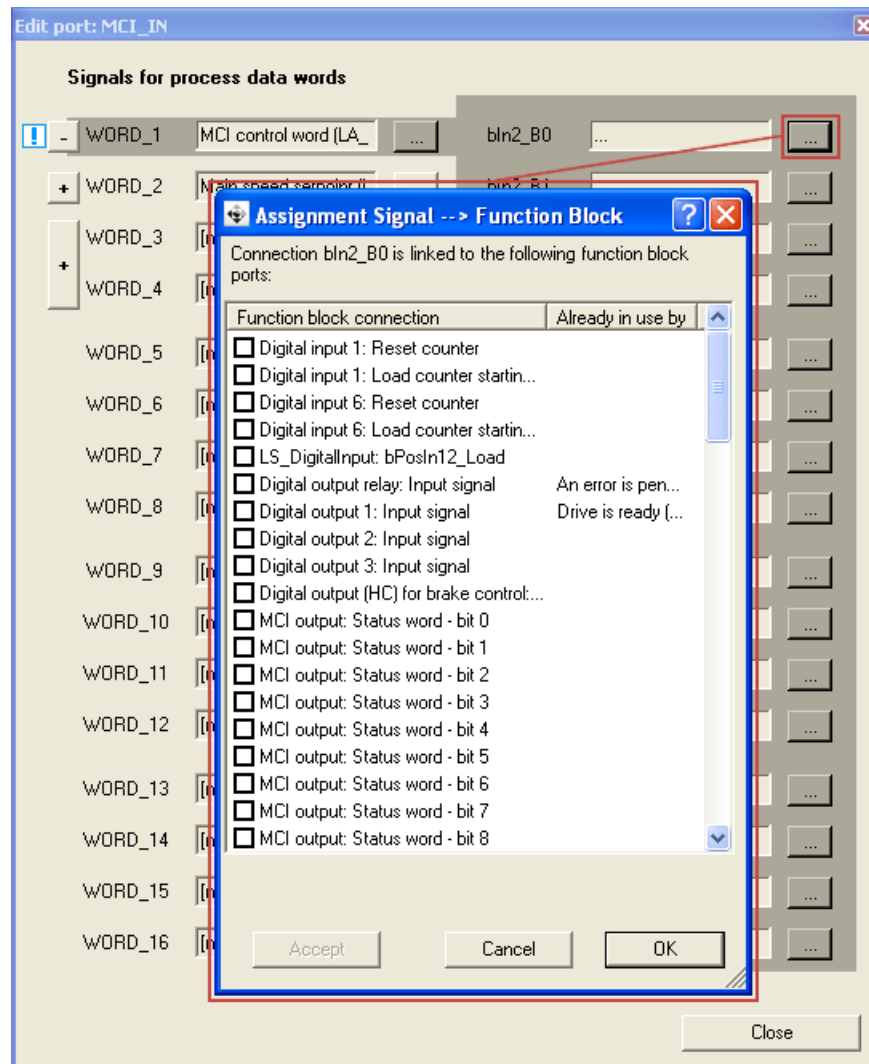


4. The  button serves to assign signals to the process data words in the *Assignment Signal* --> *Function Block* dialog box.
→ Select signals and then click the **OK** button.



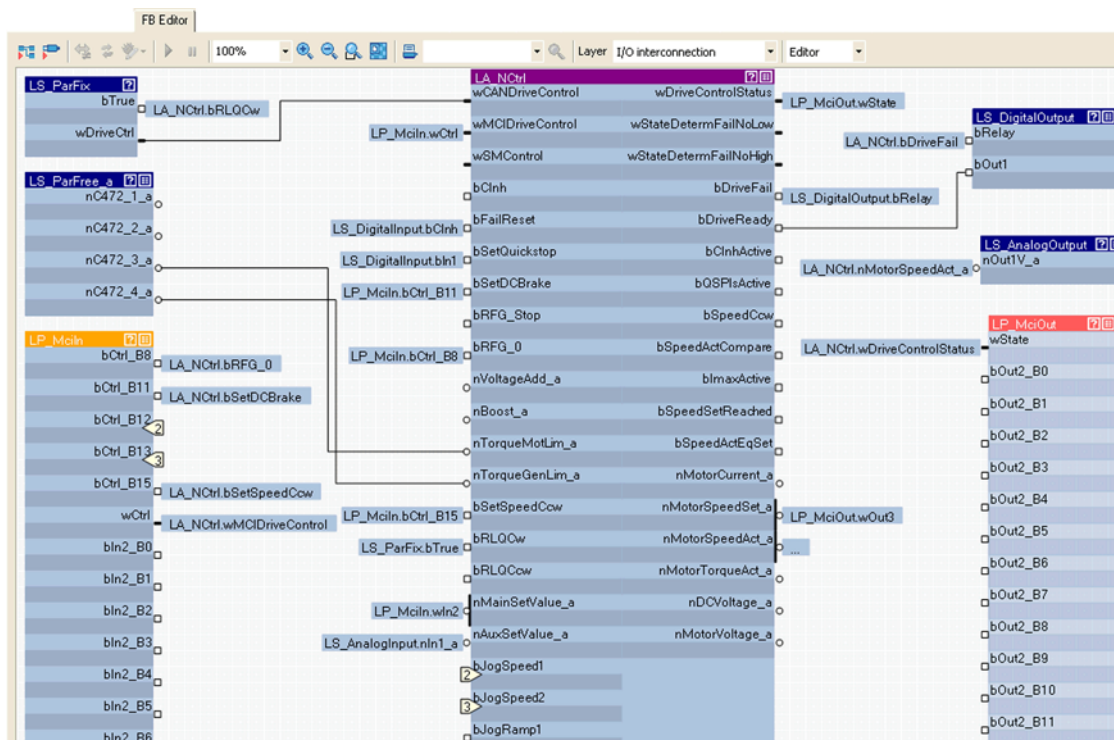
Moreover you can assign signals to the individual control and status bits at the WORD_1 and WORD_2 process data words via the  and  buttons.

→ Select the signals and then click **OK**.



**Tip!**

When the port blocks "LP_MciIn" and "LP_MciOut" are activated (see 1.), they will be visible in the »FB-Editor«. Here you can also assign signals to the process data words.



8.4

CiA402 PDO mapping in case of Inverter Drive 8400 TopLine P

The Inverter Drive 8400 TopLine P supports the CiA402 profile.

The following tables show an overview of the CiA402 objects used in the receipt PDO (Rx-PDO) and transmit PDO (Tx-PDO).


Software manual/online help for the Inverter Drive 8400 TopLine P

Here you will find detailed information on the functions of the CiA402 objects.

Receipt PDO (Rx-PDO)

Word	Index [hex]	Description
1	0x6040	Control word
2	0x6060	Operating mode
	0x60FE.1	Physical outputs
3	0x6042	Target velocity for "Velocity Mode" (is not supported yet)
4	0x607A	Target position
5		
6	0x60B1	Offset for velocity
7		
8	0x60B2	Offset for torque
9	0x60B8	Touch probe function
10	0x60E0	Positive torque limit
11	0x60E1	Negative torque limit
12	0xFFFF	Upper speed limit for "CST Mode" (is not supported yet)
13		
14	0xFFFF	Lower speed limit for "CST Mode" (is not supported yet)
15		
16	0xA580.1	Freely interconnectable

Transmit PDO (Tx-PDO)

Word	Index [hex]	Description
1	0x6041	Status word
2	0x6061	Display of the operating mode
	0x60FD	Physical inputs
3	0x6044	Current velocity for "Velocity Mode" (is not supported yet)
4	0x6064	Current position
5		
6	0x606C	Current velocity
7		
8	0x6077	Current torque
9	0x60B9	Touch probe status
10	0x60BA	Touch probe 1 positive edge
11		
12	0x60BB	Touch probe 1 negative edge
13		
14	0x603F	Error code
15	0x6074	Following error
16		

9

Parameter data transfer

Parameter data are transmitted via the fieldbus as so-called SDOs (Service Data Objects). The SDO services provide for the write and read access to the object directory.

The SDO channel provides for access to [Implemented CoE objects](#) (□ 79) and Lenze codes by means of the CoE protocol.

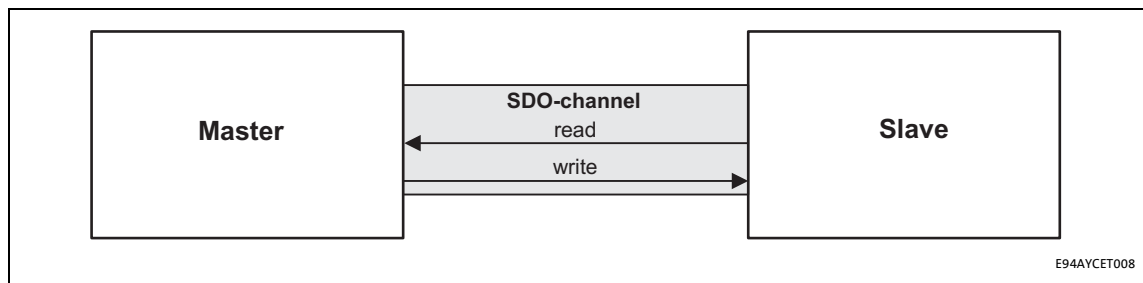
In general, the parameter data transfer is not time-critical.

Parameter data are, for instance, operating parameters, motor data, diagnostic information.

9.1

Establishing a connection between master and slave

Basically a master can always request parameter jobs from a slave if the slave is at least in the "Pre-operational" state.



[9-1] Data communication via the SDO channel

9.2 Reading and writing parameters

Parameters ...

- for instance are set for one-time system settings or if materials are changed within a machine;
- are transmitted with a low priority.

In the case of Lenze inverters, the parameters to be changed are contained in codes.

Indexing of the Lenze codes

If they are accessed via a communication module, the codes of the inverter are addressed by the index.

The index of Lenze code numbers within the manufacturer-specific area of the object directory is between 8192 (0x2000) and 24575 (0x5FFF).

Conversion formula	
Index [dec]	Index [hex]
24575 - Lenze code	0x5FFF - Lenze code _{hex}

Example of C00002 (device commands)	
Index [dec]	Index [hex]
24575 - 2 = 24573	0x5FFF - 2 = 0x5FFD

Structure of a mailbox datagram

Mailbox data are transmitted in a datagram within an EtherCAT frame. The data area of the mailbox datagram has the following structure:

Mailbox Header	CoE Header	SDO control byte	Index	Subindex	Data	Data
6 bytes	2 bytes	1 byte	2 bytes	1 byte	4 bytes	1 ... n bytes

9.2.1 Reading parameters (SDO upload)

1. The master sends "Initiate Domain Upload Request".
2. The slave acknowledges the request with a positive response ("Initiate Domain Upload Response").

In the event of an error the slave responds with "Abort Domain Transfer".



Note!

In the case of jobs for the inverter, please make sure that you convert the code into an index.

► [Indexing of the Lenze codes](#) (69)

SDO upload request

Detailed breakdown of the data for an "SDO upload request":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 Bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Reserved	BYTE	4 bits (0 ... 3)	0x00
	Complete access		1 bit (4)	0x00: The entry addressed with an index and subindex is read. 0x01: The complete object is read. (Is currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x02: Upload request
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Reserved	DWORD	4 bytes	0x00

SDO Upload Expedited Response

An "SDO Upload Expedited Response" is effected if the data length of the parameter data to be read is up to 4 bytes.

Detailed breakdown of the data for an "SDO Upload Expedited Response":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 Bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x01: Size of the data in the "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data 0x01: 3 bytes of data 0x02: 2 bytes of data 0x03: 1 byte of data
	Complete access		1 bit (4)	0x00: The entry addressed with an index and subindex is read. 0x01: The complete object is read. (Is currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x02: Upload response
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Data	DWORD	4 bytes	Data of the object

SDO Upload Normal Response

An "SDO Upload Normal Response" is effected if the data length of the parameter data to be read $\geq$ is 4 bytes.

Detailed breakdown of the data for an "SDO Upload Normal Response":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	$n \geq 0x0A$: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 Bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x01
	Transfer type		1 bit (1)	0x00: Normal transfer
	Data set size		2 bits (2, 3)	0x00
	Complete access		1 bit (4)	0x00: The entry addressed with an index and subindex is read. 0x01: The complete object is read. (Is currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x02: Upload response
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Complete size	DWORD	4 bytes	Total data length of the object
	Data	BYTE	$n - 10$ bytes	Data of the object

Example

In case of an **Upload** to index 0x5FD8 (Standard setting of C00039/1 (fixed setpoint_1) = 0x0FA0_{hex} (4000_{dec})), the transmitted response structure contains the following data:

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	0x00
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 Bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x01: Length of the data in the "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x02: 2 bytes of data
	Complete access		1 bit (4)	0x00: The entry addressed with an index and subindex is read.
	Command specifier		3 bits (5 ... 7)	0x02: Upload response
	Index	WORD	2 bytes	0xD8: Index low byte of the object 0x5F: Index high byte of the object
	Subindex	BYTE	1 byte	0x01
	Data	DWORD	2 bytes	0x0FA0

9.2.2 Writing parameters (SDO download)

1. The master sends "Initiate Domain Download Request".
2. The slave acknowledges the request with a positive response ("Initiate Domain Download Response").

In the event of an error the slave responds with "Abort Domain Transfer".



Note!

In the case of jobs for the inverter, please make sure that you convert the code into an index.

► [Indexing of the Lenze codes](#) (📖 69)

SDO Download Expedited Request

An "SDO Download Expedited Request" is effected if the data length of the parameter data to be written is up to 4 bytes.

Detailed breakdown of the data for an "SDO Download Expedited Request":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 Bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Size indicator	BYTE	1 bit (0)	0x01: Size of the data in the "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data 0x01: 3 bytes of data 0x02: 2 bytes of data 0x03: 1 byte of data
	Complete access		1 bit (4)	0x00: The entry addressed with an index and subindex is written. 0x01: The complete object is written. (Is currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x01: Download request
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Data	DWORD	4 bytes	Data of the object

SDO Download Normal Request

An "SDO Download Normal Request" is effected if the data length of the parameter data to be written $\geq$ is 4 bytes.

Detailed breakdown of the data for an "SDO Download Normal Request":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	$n \geq 0x0A$: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 Bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Size indicator	BYTE	1 bit (0)	0x01
	Transfer type		1 bit (1)	0x00: Normal transfer
	Data set size		2 bits (2, 3)	0x00
	Complete access		1 bit (4)	0x00: The entry addressed with an index and subindex is written. 0x01: The complete object is written. (Is currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x01: Download request
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Complete size	DWORD	4 bytes	Total data length of the object
	Data	BYTE	$n - 10$ bytes	Data of the object

SDO Download Response

Detailed breakdown of the data for a "SDO download response":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 Bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x0
	Transfer type		1 bit (1)	0x0
	Data set size		2 bits (2, 3)	0x0
	Complete access		1 bit (4)	0x00: The entry addressed with an index and subindex is written. 0x01: The complete object is written. (Is currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x3: Download response
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Reserved	DWORD	4 bytes	0x00

Example

In the case of a **Download** to the index 0x1600, the transmitted request structure contains the following data:

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	0x00
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 Bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Size indicator	BYTE	1 bit (0)	0x01: Size of the data in the "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data
	Complete access		1 bit (4)	0x00: The entry addressed with an index and subindex is written.
	Command specifier		3 bits (5 ... 7)	0x01: Download request
	Index	WORD	2 bytes	0x00: Index low byte of the object 0x16: Index high byte of the object
	Subindex	BYTE	1 byte	0x01: Subindex of the object
	Data	DWORD	4 bytes	0x5C930110

9 Parameter data transfer

9.3 Implemented CoE objects

9.3 Implemented CoE objects

Lenze devices can be parameterised with both Lenze codes and the manufacturer-independent "CoE objects". In order to fully comply with EtherCAT communication, you must only use the CoE objects for parameterisation. The CoE objects described in this manual are defined in the "EtherCAT Specification, Part 6 – Application Layer Protocol Specification".

Index	Designation	Subindex	Subindex name	Type	Bits	Access
0x1000	Device type	-	-	UDINT	32	R
0x1008	Device name	-	-	STRING(8)	64	R
0x1009	Hardware version	-	-	STRING(8)	64	R
0x100A	Software version	-	-	STRING(7)	56	R
0x1018	Identity	0	Subindex 000	USINT	8	R
		1	Vendor ID	UDINT	32	R
		2	Product code	UDINT	32	R
		3	Revision number	UDINT	32	R
		4	Serial number	UDINT	32	R
0x10F1	Error Settings Object	0	Subindex 000	USINT	8	R
		1	Local Error Reaction	USINT	8	R
		2	Sync Error Counter Limit	USINT	8	RW
0x1600	RxPDO 1	0	Subindex 000	USINT	8	RW
		1 ... 16	Output Object 1 ... 16	UDINT	32	RW
0x1A00	TxPDO 1	0	Subindex 000	USINT	8	RW
		1 ... 16	Input Object 1 ... 16	UDINT	32	RW
0x1C00	Sync Man Communication type	0	Subindex 000	USINT	8	R
		1	Elements	UDINT	32	R
0x1C12	Sync Man 2 Assignment	0	Subindex 000	USINT	8	R
		1	PDO Mapping object index of assigned RxPDO	UDINT	32	R
0x1C13	Sync Man 3 Assignment	0	Subindex 000	USINT	8	R
		1	PDO Mapping object index of assigned TxPDO	UDINT	32	R
0x1C32	Sync Man 2 Synchronization	0	Subindex 000	USINT	8	R
		1	Synchronization type	UINT	16	R
		2	Cycle time / ns	UDINT	32	R
		3	Shift time / ns	UDINT	32	R
		4	Sync types supported	UINT	16	R
		5	Minimum cycle time / ns	UDINT	32	R
0x1C33	Sync Man 3 Synchronization	0	Subindex 000	USINT	8	R
		1	Synchronization type	UINT	16	R
		2	Cycle time / ns	UDINT	32	R
		3	Shift time / ns	UDINT	32	R
		4	Sync types supported	UINT	16	R
		5	Minimum cycle time / ns	UDINT	32	R
		6	Minimum shift time / ns	UDINT	32	R

R: Read access only
RW: Read and write access

9.4 EtherCAT objects of the communication module

The object directory displays the [Parameters of the communication module](#) (97) as objects:

Index	Code	Index name	Subindex	Subindex name	Type	Bits	Access
0x29E5	C13850	All words to master	1 ... 16	All words to master	UNSIGNED	16	R
0x29E4	C13851	All words from master	1 ... 16	All words from master	UNSIGNED	16	R
0x29E3	C13852	All words to standard device	1 ... 16	All words to standard device	UNSIGNED	16	R
0x29E2	C13853	All words from standard device	1 ... 16	All words from standard device	UNSIGNED	16	R
0x29DC	C13859	Number of Tx PDOs	-	-	UNSIGNED	16	R
0x29DB	C13860	Number of Rx PDOs	-	-	UNSIGNED	16	R
0x29DA	C13861	Bus state	-	-	UNSIGNED	16	R
0x29D7	C13864	Active station address	-	-	UNSIGNED	16	R
0x29D4	C13867	Display of emergency data	-	-	STRING(8)	64	R
0x29C8	C13879	Bus error	-	-	UNSIGNED	16	R
0x29C7	C13880	Reaction on communication failure	2	1: Response to communication interruption 2: Response to sync telegram failure	UNSIGNED	8	RW
0x29C6	C13881	Response time when exiting "Operational"	-	-	UNSIGNED	16	RW
0x29C4	C13883	DC active	-	-	UNSIGNED	8	R
0x29C3	C13884	Synchronisation is active	-	-	UNSIGNED	8	R
0x29C2	C13885	Clear process data	-	-	UNSIGNED	8	RW
0x29C0	C13887	Suppress emergency message in case of	-	-	BITFIELD	8	RW
0x29B4	C13899	Station alias address	-	-	UNSIGNED	16	RW
0x29B3	C13900	Firmware product type	-	-	STRING(8)	64	R
0x29B2	C13901	Firmware compilation date	-	-	STRING(20)	160	R
0x29B1	C13902	Firmware version	-	-	STRING(11)	88	R

R: Read access only

RW: Read and write access

9.5 SDO abort codes (Abort codes)

If an SDO request is evaluated negatively, a corresponding error code is output.

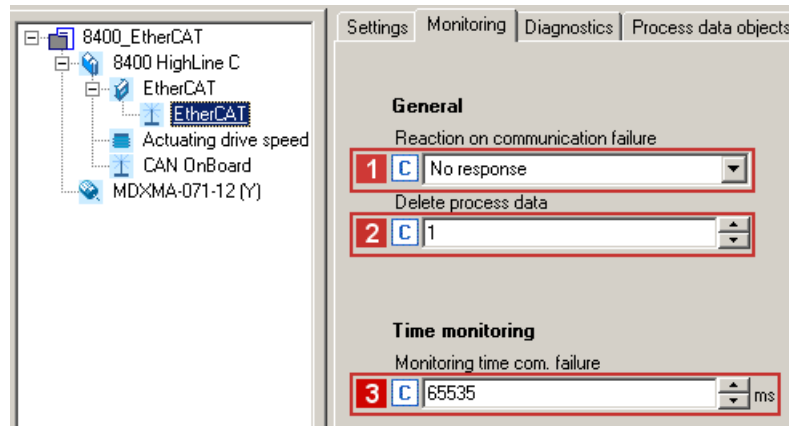
Index [hex]	Description
0x00000000	No fault
0x05030000	The status of the toggle bit has not changed
0x05040000	SDO protocol time-out
0x05040001	Invalid or unknown specification symbol for the client/server command
0x05040005	Not enough space in the main memory
0x06010000	Access to object not supported
0x06010001	Read access to a write-protected object
0x06010002	Write access to a write-protected object
0x06020000	An object does not exist in the object directory
0x06040041	An object cannot be mapped into the PDO
0x06040042	The number and/or length of the objects mapped would exceed the PDO length
0x06040043	General parameter incompatibility
0x06040047	General internal device incompatibility
0x06060000	Access has failed due to a fault in the hardware
0x06070010	The data type or the parameter length does not correspond
0x06070012	Incorrect data type (The parameter length is too large)
0x06070013	Incorrect data type (The parameter length is too small)
0x06090011	A subindex is not available
0x06090030	The value range for parameters is too great (only for write access)
0x06090031	The parameter value is too high
0x06090032	The parameter value is too low
0x06090036	The maximum value is lower than the minimum value
0x08000000	General error
0x08000020	Data cannot be transferred to the application or stored in the application
0x08000021	Due to local control, data cannot be transferred to the application or stored in the application
0x08000022	Due to the current device state, data cannot be transferred to the application or stored in the application
0x08000023	The dynamic object directory generation has failed, or no object directory is available

10 Monitoring

10.1 Interruption of EtherCAT communication

An interruption of the EtherCAT communication in the "Operational" state, e.g. due to cable break or failure of the EtherCAT master, is detected by the slave.

The response to an interrupted communication is triggered by settings in the **Monitoring** tab:



- During the initialisation of the EtherCAT communication, the sync manager watchdog monitoring time determined in the master is transferred to the slave.
If the slave does not receive any valid process data in the "Operational" status, the setting in **2** [C13885](#) is taken as a basis for the process data:
 - Value '0': The data sent last by the master are used.
 - Value '1': PDOs are set to the value '0'.
 After the watchdog monitoring time has elapsed, the slave changes to the "Safe operational" state (see [C13861](#)) and the green LED BS is activated (see [Fieldbus status displays](#) ([□ 86](#))).
There is no response in the slave.
- In order that a response is triggered in the slave, you have to set a **1** **Reaction on communication failure** ([C13880/1](#)).
- The response is delayed if you set an internal **3** **monitoring time** ([C13881](#)) in addition.
In the Lenze setting ([C13881](#) = 0), no delay is set.
The monitoring time elapses as soon as the "Operational" state is exited. (See 1.).
After the monitoring time has elapsed, the response set is executed with the error message "[Operational status quit \[0x01bc8131\]](#)" ([□ 95](#)).

10.2 Sync frame failure detection

During the [Synchronisation with "Distributed Clocks" \(DC\)](#) (📖 37), this monitoring checks whether an EtherCAT PDO telegram (Sync Manager 2 Event) has arrived between two signals.

For this purpose, the communication module comes with an internal EtherCAT telegram failure error counter. The telegram failure error counter is incremented by the value '3' in case of a telegram failure. For each PDO received correctly, the counter is decremented by '1'. If the internal telegram failure error counter reaches a threshold adjustable in CoE object 0x10F1.2, a change to "Safe-Operational" takes place and the ["Sync telegram failure \[0x01bc8700\]"](#) error message is output.

In the standard setting, monitoring is deactivated (0x10F1.2 = 0). A sensible setting is a value > '5' in order that an error will only be triggered in the inverter when a second telegram fails.

An error response can be set via code [C13880/2](#) ("0: No response" is preset here).

10.3 Interruption of internal communication

The response to a communication error between the communication module and the standard device can be set in the standard device code [C01501](#) (module in the MCI slot).

An externally supplied communication module reports a connection abort to the standard device via an emergency telegram to the master and changes to the "Safe-operational" state.

The error message ["Lost connection to 8400 \[0x01bc3100\]"](#) (📖 93) is output.

11

Diagnostics

For fault diagnostics, the communication module is provided with LEDs on the front. Furthermore, you can carry out the [Diagnostics with the »Engineer«](#) (📖 89).

11.1

LED status displays



Note!

During normal operation ...

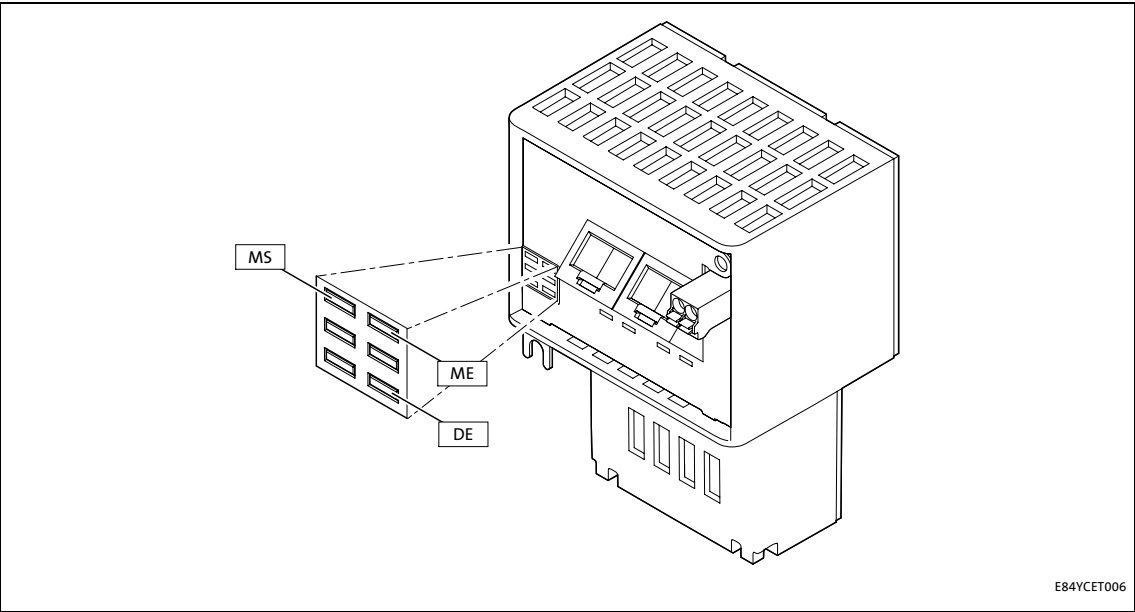
- only the LEDs **MS** (📖 85) and **BS** (📖 86) should be lit constantly.
- the green LED at the RJ45 sockets X246/X247 must be lit (📖 88).

The following status displays can be differentiated:

- [Module status displays](#) (📖 85)
- [Fieldbus status displays](#) (📖 86)
- [Status displays at X246 and X247](#) (📖 88)

11.1.1 Module status displays

The LEDs **MS**, **ME** and **DE** show the module status.



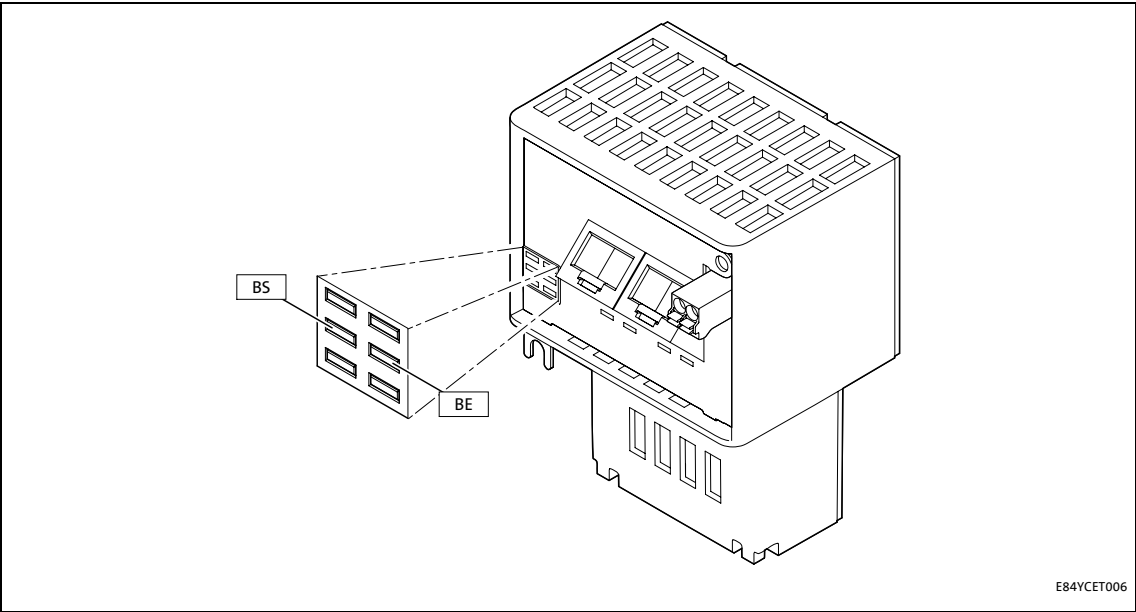
[11-1] LEDs MS, ME, DE

LED	Colour	State	Description
MS	green	on	 The communication module is supplied with voltage and is connected to the standard device.
		blinking	200 ms 200 ms The communication module is supplied with voltage, but has no connection to the standard device (the standard device is switched off, in the initialisation phase, or not available).
ME	red	on	 An error has occurred in the communication module.
DE	red	on	 The communication module is not accepted by the standard device, or the standard device is not active. (See notes in the documentation for the standard device).

11.1.2

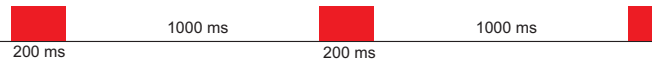
Fieldbus status displays

The LEDs **BS** and **BE** show the fieldbus status.



[11-2] LEDs BS, BE

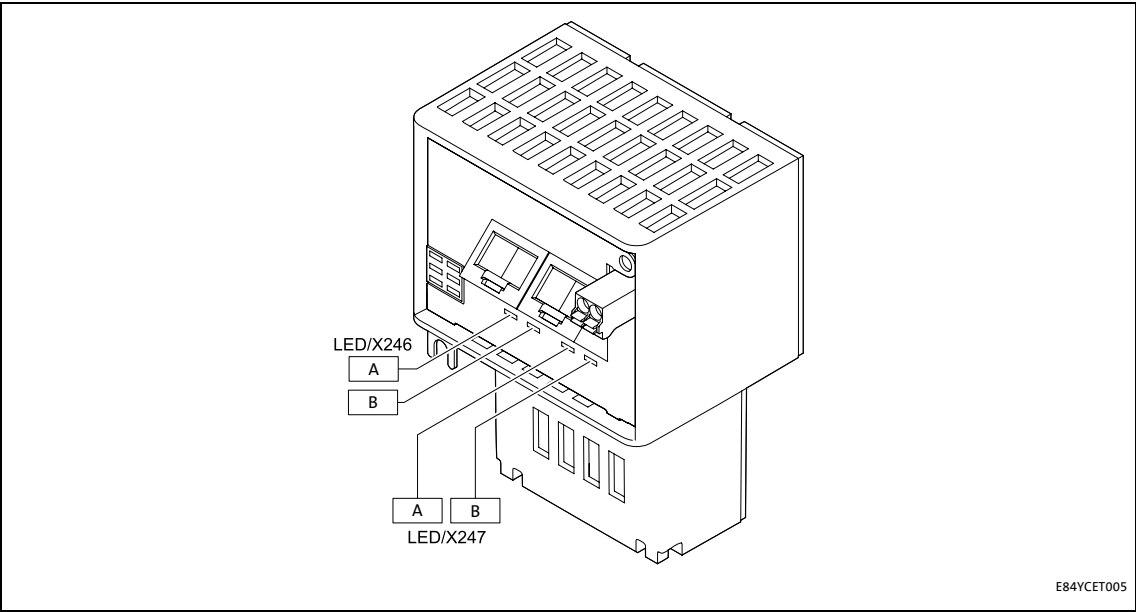
LED	Colour	State	Description
BS	green	off	The communication module is not active on the fieldbus or is in the "Init" state.
		blinking	200 ms 200 ms "Pre-operational" status is active: • Access to parameters and objects is possible. • No process data exchange.
		blinking once (single flash)	200 ms 1000 ms 200 ms 1000 ms "Safe-operational" status is active: • The data are not yet active in the standard device.
		on	The communication module is in the "Operational" state.

LED	Colour	State	Description
BE	red	off	No fault
		blinking	 The configuration is invalid/faulty.
		blinking once (single flash)	 • A non requested state change has occurred. (The slave application has autonomously changed the EtherCAT status.) • Synchronisation error (The EtherCAT node automatically changes to the "Safe-operational" state.)
		blinking twice (double flash)	 An "Application Watchdog Timeout" or a "Sync Manager Watchdog Timeout" has occurred.

11.1.3

Status displays at X246 and X247

The LEDs beneath the RJ45 sockets X246 and X247 show the EtherCAT connection status.

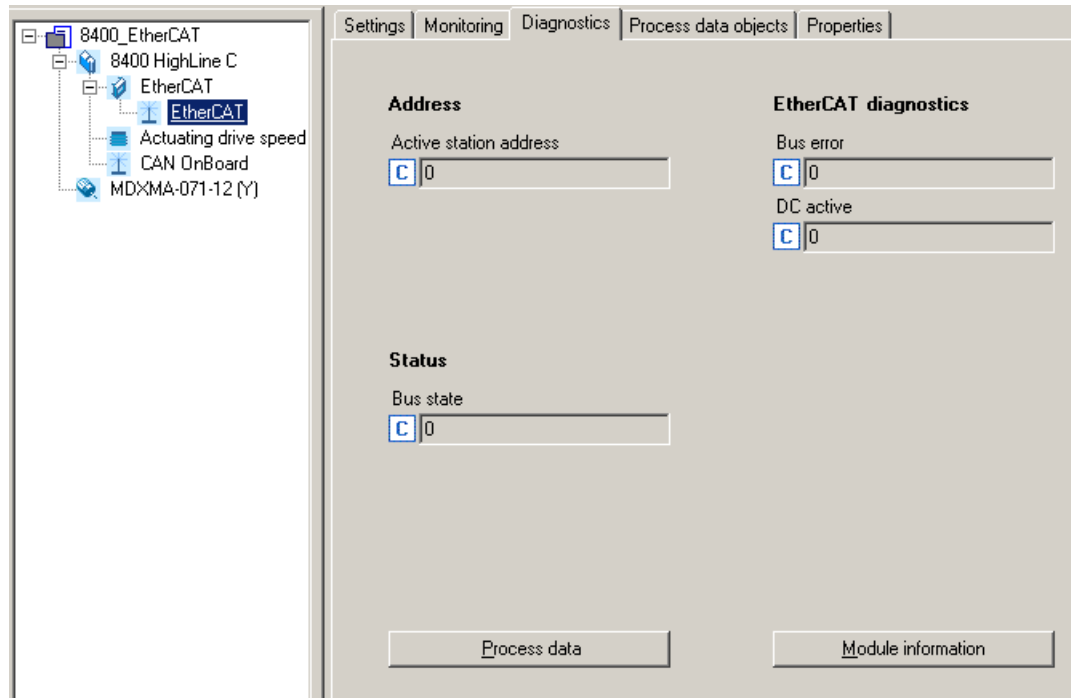


[11-3] LEDs at the RJ45 sockets X246 and X247

LED	Colour	State	Description
A	green	on	A physical EtherCAT connection is available.
		flickering	50 ms Data are being exchanged via EtherCAT.
B	red	off	This LED is not used.

11.2 Diagnostics with the »Engineer«

In the »Engineer«, the **Diagnostics** tab displays various pieces of EtherCAT diagnostic information.



11.3 Emergency requests / Emergency messages

Emergency messages are sent to the EtherCAT master once when the error status of the inverter changes, i.e. ...

- when an error in the inverter or in the communication module occurs;
- when an internal error of the communication module is eliminated.

An "Emergency Request" on the fieldbus consists of the components "Mailbox Header", "CANopen Header" and the actual "Emergency Message":

Mailbox header	CANopen header	Emergency Message
6 bytes	2 bytes	8 bytes

The emergency message last sent by the Inverter Drive 8400 is displayed in code [C13867](#).

Code [C13887](#) serves to select the error responses where no emergency messages are to be sent to the EtherCAT master.

11.3.1 Structure of the Emergency message

Example: Emergency message of the error "[Operational status quit \[0x01bc8131\]](#)"

Byte 1	Byte2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Emergency Error code		Error Register (I-1001)	Reserved	Error code Inverter Drive 8400 / E84AYCET			
Low byte	High byte	Low byte	High byte	Low word		High word	
				Low byte	High byte	Low byte	High byte
0x00	0x10	0x01	0x00	0x31	0x81	0xbc	0x01

- Bytes 1 and 2 display that an error is pending.
- Byte 3 display the contents of the error register (I-1001).
- Bytes 5 ... 8 contain the error code.



Software manual/online help for the Inverter Drive 8400

Here you will find detailed information on the error codes.

11.3.2 Emergency messages (overview)

The following emergency messages can occur:

Emergency no. [hex]	Designation	Extended error code [hex]	Description
0x1000	General error	0x01bc3100	Lost connection to 8400 [0x01bc3100] (¶ 93)
0x1000	General error	0x01bc6430	Invalid module configuration [0x01bc6430] (¶ 94)
0x1000	General error	0x01bc8131	Operational status quit [0x01bc8131] (¶ 95)
0xA000	Synchronisation error	0x01bc8265	Sync telegram failure [0x01bc8700] (¶ 95)
0x0000	Error reset	-	-

12

Error messages

This chapter provides the error messages of the communication module E84AYCET EtherCAT as a supplement to the error list in the software manual and the »Engineer« online help for the Inverter Drives 8400.

12.1

Short overview of the EtherCAT error messages


Software manual/online help for the Inverter Drive 8400

Here you can find general information on diagnostics & fault analysis and on error messages.

The following table lists all EtherCAT error messages in numerical order of the error number. Furthermore the preset error response and – if available – the parameters for setting the error response are specified.


Tip!

When you click the cross-reference in the first column, you will see a detailed description (causes and remedies) of this error message.

Error no. [hex]	Subject area no. [dec]	Error no. [dec]	Error text	Error type (Error response)	Adjustable in
0x01bc3100	444	12544	Lost connection to 8400 base device	1: Fault	-
0x01bc5531	444	21809	Memory: No access	1: Fault	-
0x01bc5532	444	21810	Memory: Read error	1: Fault	-
0x01bc5533	444	21811	Memory: Write error	1: Fault	-
0x01bc6010	444	24592	Restart by watchdog reset	1: Fault	-
0x01bc6011	444	24593	Internal error	1: Fault	-
0x01bc6100	444	24832	Internal error	1: Fault	-
0x01bc6101	444	24833	Internal error	1: Fault	-
0x01bc641f	444	25631	Invalid parameter record	1: Fault	-
0x01bc6420	444	25632	Error: Lenze setting loaded	1: Fault	-
0x01bc6430	444	25648	Invalid module configuration	1: Fault	-
0x01bc8131	444	33073	"Operational" status quit	0: No response	C13880/1
0x01bc8700	444	34560	Sync telegram failure	0: No response	C13880/2

12.2 Possible causes and remedies

This chapter lists all EtherCAT error messages in the numerical order of the error numbers. Possible causes and remedies as well as responses to the error messages are described in detail.

Lost connection to 8400 [0x01bc3100]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ Information			
Cause		Remedy	
MCI communication for the Inverter Drives 8400 is interrupted. • Inverter Drive 8400 is switched off. • The communication module has not been inserted properly into the slot of the Inverter Drives 8400.		• Switch on Inverter Drive 8400. • Check screwed connection of the communication module to the MC slot of the Inverter Drive 8400. • Send communication module and Inverter Drive 8400 with error description to Lenze.	

Memory: No access [0x01bc5531]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information			
Cause		Remedy	
Access via standard device to parameter set within the memory module was not successful.		Repeat download of the application (including module).	

Memory: Read error [0x01bc5532]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information			
Cause		Remedy	
Parameters of the memory module could not be read.		Repeat download of the application (including module).	

Memory: Write error [0x01bc5533]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information			
Cause		Remedy	
Parameters of the memory module could not be written.		Repeat download of the application (including module).	

Restart by watchdog reset [0x01bc6010]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information			
Cause		Remedy	
Communication module is defective.		Send communication module with error description to Lenze.	

Internal error [0x01bc6011]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
Communication module is defective.	Send communication module with error description to Lenze.

Internal error [0x01bc6100]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
Communication module is defective.	Send communication module with error description to Lenze.

Internal error [0x01bc6101]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
Communication module is defective.	Send communication module with error description to Lenze.

Invalid parameter set [0x01bc641f]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
No active parameter set could be loaded.	Repeat download of the application (including module).

Error: Lenze setting loaded [0x01bc6420]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
Access via standard device to parameter set within the memory module was not successful.	Repeat download of the application (including module).

Invalid module configuration [0x01bc6430]

Response (Lenze setting printed in bold)	Setting: not possible
☒ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
- Faulty module configuration. - Incorrect sync settings.	Check and correct module configuration/sync settings.

Operational status quit [0x01bc8131]

Response (Lenze setting printed in bold)	Setting: C13880/1 (☒ Adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☒ Quick stop by trouble ☒ Warning locked ☐ Warning ☒ Information	
Cause	Remedy
The EtherCAT data exchange was stopped in the "Operational" state. See also chapter " Interruption of EtherCAT communication " (82).	• Check cables and terminals. • The master has to reset the node to the "Operational" status. (If required, check a pending emergency message first).

Sync telegram failure [0x01bc8700]

Response (Lenze setting printed in bold)	Setting: C13880/2 (☒ Adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☒ Quick stop by trouble ☒ Warning locked ☐ Warning ☒ Information	
Cause	Remedy
• The cycle time settings are faulty. • The task time setting of a linked PLC program is faulty. • EtherCAT master / DC master settings are incorrect. • The sync source setting is faulty.	• Check the cycle times. • Check the task time of a linked PLC program. • Check the EtherCAT master / DC master settings. • Check the settings in C01120 of the standard device (selection of the sync source).

13 Parameter reference

This chapter supplements the parameter list and the table of attributes in the software manual and in the »Engineer« online help for the Inverter Drive 8400 by the parameters of the E84AYCET communication module (EtherCAT).



Software manual/online help for the Inverter Drive 8400

Here you will find general information about parameters.

13.1 Communication-relevant parameters of the standard device

This chapter lists the communication-relevant parameters of the Inverter Drive 8400 in numerically ascending order.

C01120

Parameter Name: C01120 Sync signal source		Data type: UNSIGNED_8 Index: 23455 _d = 5B9F _h
Selection of the source for DC synchronisation signals. • Select "4: MCI" for the communication module. • In general, only one source can synchronise the drive. ▶ Synchronisation with "Distributed Clocks" (DC) (□ 37)		
Selection list (Lenze setting printed in bold)		Info
0	Off	Synchronisation off
1	CAN on board	Synchronisation via the system bus "CAN on board"
4	MCI	Synchronisation via MCI (communication module)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C01501

Parameter Name: C01501 Reaction on communication error with MCI		Data type: UNSIGNED_8 Index: 23074 _d = 5A22 _h
Configuration of monitoring functions for the communication module		
Selection list		
0	No response	
1	Error	
2	Trouble	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
6	Information	
Subcodes	Lenze setting	Info
C01501/1	0: No response	Reaction on MCI communication failure • Response to a communication error (module in slot)
C01501/2	0: No response	Reaction on MCI module invalid • Response to an incompatible communication module or module not in slot
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

13.2 Parameters of the communication module

This chapter lists the parameters of the E84AYCET communication module (EtherCAT) in ascending numerical order.

C13850

Parameter Name: C13850 All words to master			Data type: UNSIGNED_16 Index: 10725 _d = 29E5 _h
Display of the process data words (subcodes 1 ... 16) which are transferred from the communication module to the master. Only those which are configured are valid.			
Display area (min. value unit max. value)			
0		65535	
Subcodes			Info
C13850/1			
...			
C13850/16			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13851

Parameter Name: C13851 All words from master			Data type: UNSIGNED_16 Index: 10724 _d = 29E4 _h
Display of the process data words (subcodes 1 ... 16) which are transferred from the master to the communication module. Only those which are configured are valid.			
Display area (min. value unit max. value)			
0		65535	
Subcodes			Info
C13851/1			
...			
C13851/16			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13852

Parameter Name: C13852 All words to standard device			Data type: UNSIGNED_16 Index: 10723 _d = 29E3 _h
Display of the process data words (subcodes 1 ... 16) which are transferred from the communication module to the standard device.			
Display area (min. value unit max. value)			
0		65535	
Subcodes			Info
C13852/1			
...			
C13852/16			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13853

Parameter Name: C13853 All words from standard device			Data type: UNSIGNED_16 Index: 10722 _d = 29E2 _h	
Display of the process data words (subcodes 1 ... 16) which are transferred from the standard device to the communication module.				
Display area (min. value unit max. value)				
0		65535		
Subcodes			Info	
C13853/1				
...				
C13853/16				
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT				

C13859

Parameter Name: C13859 Number of Tx PDOs		Data type: UNSIGNED_16 Index: 10716 _d = 29DC _h	
Number of process data words sent			
Display area (min. value unit max. value)			
0			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13860

Parameter Name: C13860 Number of Rx PDOs			Data type: UNSIGNED_16 Index: 10715 _d = 29DB _h
Number of process data words received			
Display area (min. value unit max. value)			
0			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13861

Parameter Name: C13861 Bus state		Data type: UNSIGNED_16 Index: 10714 _d = 29DA _h
Display of the current bus status ▶ EtherCAT state machine (📖 57)		
Selection list (read only)		Info
0	Nonexistent	No online value available (communication module is offline).
1	Init	
2	Pre-operational	
3	Bootstrap	Is not supported.
4	Safe-operational	
8	Operational	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13864

Parameter Name: C13864 Active station address		Data type: UNSIGNED_16 Index: 10711 _d = 29D7 _h
Display of the station address allocated by the master		
Display area (min. value unit max. value)		
0		32767
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13867

Parameter Name: C13867 Display of emergency data		Data type: OCTET_STRING Index: 10708 _d = 29D4 _h
Display of the emergency data last sent by the inverter (string with a length of 8 bytes).		
▶ Emergency requests / Emergency messages (📖 90)		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13879

Parameter Name: C13879 Bus error		Data type: UNSIGNED_16 Index: 10696 _d = 29C8 _h
Bit coded display of the bus error		
Additionally, an error message is entered in the EtherCAT register " AL Status Code " (📖 58).		
Value is bit-coded:		Info
Bit 0	General bus error	
Bit 1	Reserved	
...	...	
Bit 31	Reserved	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13880

Parameter Name: C13880 Reaction on communication failure		Data type: UNSIGNED_8 Index: 10695 _d = 29C7 _h
The set response will be executed if ...		
• the node detects that is no longer in the "Operational" state and the monitoring time (C13881) has elapsed or • a sync telegram failure has been detected.		
Selection list		
0	No response	
1	Error	
3	Quick stop by trouble	
4	Warning locked	
6	Information	
Subcodes	Lenze setting	Info
C13880/1	0: No response	Response to communication interruption ▶ Interruption of EtherCAT communication (📖 82)
C13880/2	0: No response	Response to sync telegram failure ▶ Sync frame failure detection (📖 83)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13881

Parameter Name: C13881 Response time when exiting "Operational"			Data type: UNSIGNED_16 Index: 10694 _d = 29C6 _h	
If the "Operational" state is exited, the response parameterised with C13880 occurs after the time set here has elapsed. A change in monitoring is effective immediately.				
Setting range (min. value unit max. value)			Lenze setting	
0	ms	65535	0 ms	
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT				

C13883

Parameter Name: C13883 DC active		Data type: UNSIGNED_8 Index: 10692 _d = 29C4 _h	
The code displays if the DC synchronisation has been activated for the communication module. ▶ Synchronisation with "Distributed Clocks" (DC) (📖 37)			
Selection list (read only)		Info	
0	DC unused		
1	DC for synchronisation		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13884

Parameter Name: C13884 Base device synchronisation status		Data type: UNSIGNED_8 Index: 10691 _d = 29C3 _h	
The code displays whether the inverter is DC-synchronised. ▶ Synchronisation with "Distributed Clocks" (DC) (□ 37)			
Selection list (read only)		Info	
0	Not synchronised		
1	Synchronised		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13885

Parameter Name: C13885 Clear process data		Data type: UNSIGNED_8 Index: 10690 _d = 29C2 _h
This code serves to set which process data the slave is to process for maintaining internal communication when the EtherCAT has exited the "Operational" state.		
Selection list (Lenze setting printed in bold)		Info
0	Use of most recent master PDOs	The data last sent by the master are used.
1	PDOs are set to the value '0'	The process data contents is set to the value '0'.
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13887

Parameter Name: C13887 Suppress emergency message in case of		Data type: BITFIELD_8 Index: 10688 _d = 29C0 _h	
This code serves to prevent the emergency messages from being transmitted to the EtherCAT master. Here, errors of a certain type can be suppressed deliberately. Furthermore, all errors are entered in the logbook. A change will only be effective if no error number with the error type selected here is available in the standard device code C00165 .			
Value is bit-coded:		Info	
Bit 0	Error		
Bit 1	Trouble		
Bit 2	Quick stop by trouble		
Bit 3	Warning locked		
Bit 4	Warning		
Bit 5	Information		
Bit 6	Reserved		
Bit 7	Reserved		
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13899

Parameter Name: C13899 Station alias		Data type: UNSIGNED_16 Index: 10676 _d = 29B4 _h	
This code serves to set a station alias address. In order to use a station alias address, you must select a value > '0'. • The station alias address must only be set if the node is part of a "hot connect" group. • The station alias address must be unambiguous and may only be assigned once within the EtherCAT network. • Use the same station alias address in the EtherCAT master and in the slave. ▶ Address allocation (📖 36)			
Setting range (min. value unit max. value)		Lenze setting	
0		32767	0
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT			

C13900

Parameter Name: C13900 Firmware: Product type		Data type: VISIBLE_STRING Index: 10675 _d = 29B3 _h
The code contains a string with a length of 8 bytes. The following identification code is output: • "E84AFCET" (up to V01.05) • "E84AFYET" (from V02.00)		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13901

Parameter Name: C13901 Firmware compilation date		Data type: VISIBLE_STRING Index: 10674 _d = 29B2 _h
The code contains a string with a length of 20 bytes. The creation date ("MM DD YYYY") and the time ("hh:mm:ss") of the software are output here. Example: "Mar 21 2005 12:31:21"		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO MAP RX ☐ PDO MAP TX ☐ COM ☐ MOT		

C13902

Parameter Name: C13902 Firmware version	Data type: VISIBLE_STRING Index: 10673 _d = 29B1 _h
The code contains a string with a length of 5 bytes. The firmware version is output here. Example: "01.00"	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

13.3 Table of attributes

The table of attributes contains information required for communicating with the Inverter Drive 8400 via parameters.

How to read the table of attributes:

Column		Meaning	Entry	
Code		Parameter designation	Cxxxxx	
Name		Parameter short text (display text)	Text	
Index	dec	Index by which the parameter is addressed. The subindex for array variables corresponds to the Lenze subcode number.	24575 - Lenze- code	Only required for access via a bus system
	hex		5FFF _h - Lenze code number	
Data	DS	Data structure	E	Single variable (only one parameter element)
			A	Array variable (several parameter elements)
	DA	Number of the 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
			OCTET_STRING	
	Factor	Factor for data transmission via a bus system, depending on the number of decimal positions	Factor	1 = no decimal positions 10 = 1 decimal position 100 = 2 decimal positions 1000 = 3 decimal positions
Access	R	Read access	☒ Reading permitted	
	W	Write access	☒ Writing permitted	
	CINH	Controller inhibit (CINH) required	☒ Writing only possible if the controller is inhibited (CINH)	

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C13850	All words to master	10725	29E5	A	16	UNSIGNED_16	1	☒		
C13851	All words from master	10724	29E4	A	16	UNSIGNED_16	1	☒		
C13852	All words to standard device	10723	29E3	A	16	UNSIGNED_16	1	☒		
C13853	All words from standard device	10722	29E2	A	16	UNSIGNED_16	1	☒		
C13859	Number of Tx PDOs	10716	29DC	E	1	UNSIGNED_16	1	☒		
C13860	Number of Rx PDOs	10715	29DB	E	1	UNSIGNED_16	1	☒		
C13861	Bus state	10714	29DA	E	1	UNSIGNED_16	1	☒		
C13864	Active station address	10711	29D7	E	1	UNSIGNED_16	1	☒		
C13867	Display of emergency data	10708	29D4	E	1	OCTET_STRING		☒		
C13879	Bus error	10696	29C8	E	1	UNSIGNED_16	1	☒		
C13880	Reaction on communication failure	10695	29C7	A	2	UNSIGNED_8	1	☒	☒	
C13881	Response time when exiting "Operational"	10694	29C6	E	1	UNSIGNED_16	1	☒	☒	
C13883	DC active	10692	29C4	E	1	UNSIGNED_8	1	☒		
C13884	Synchronisation is active	10691	29C3	E	1	UNSIGNED_8	1	☒		
C13885	Clear process data	10690	29C2	E	1	UNSIGNED_8	1	☒	☒	
C13887	Suppress emergency message in case of	10688	29C0	E	1	BITFIELD_8		☒	☒	
C13899	Station alias address	10676	29B4	E	1	UNSIGNED_16	1	☒	☒	
C13900	Firmware product type	10675	29B3	E	1	VISIBLE_STRING		☒		
C13901	Firmware compilation date	10674	29B2	E	1	VISIBLE_STRING		☒		
C13902	Firmware version	10673	29B1	E	1	VISIBLE_STRING		☒		

A

Abort codes [81](#)
Accessing process data [60](#)
Activating changed settings [53](#)
Active station address (C13864) [99](#)
Address allocation [36](#)
ADS communication parameters in »TwinCAT« and »Engineer« [45](#)
AL Status Code [58](#)
All words from master (C13851) [97](#)
All words from standard device (C13853) [98](#)
All words to master (C13850) [97](#)
All words to standard device (C13852) [97](#)
Application as directed [12](#)
Application notes (representation) [9](#)
Approvals [15](#)
Automatic device identification [35](#)

B

Base device synchronisation status (C13884) [100](#)
Baud rate [15](#)
Before initial switch-on [33](#)
Bus device type [15](#)
Bus error (C13879) [99](#)
Bus state (C13861) [98](#)

C

C01120 | Sync signal source [96](#)
C01501 | Reaction on MCI communication failure [96](#)
C13850 | All words to master [97](#)
C13851 | All words from master [97](#)
C13852 | All words to standard device [97](#)
C13853 | All words from standard device [98](#)
C13859 | Number of Tx PDOs [98](#)
C13860 | Number of mapped words PDO Rx [98](#)
C13861 | Bus state [98](#)
C13864 | Active station address [99](#)
C13867 | Display of emergency data [99](#)
C13879 | Bus error [99](#)
C13880 | Reaction on communication failure [99](#)
C13881 | Response time when exiting "Operational" [100](#)
C13883 | DC active [100](#)
C13884 | Base device synchronisation status [100](#)
C13885 | Clear process data [100](#)
C13887 | Suppress emergency message in case of [101](#)
C13899 | Station alias [101](#)
C13900 | Firmware product type [101](#)
C13901 | Firmware compilation date [101](#)
C13902 | Firmware version [102](#)

Carrying out the port interconnection in the »Engineer« [62](#)
CiA402 PDO mapping - Inverter Drive 8400 TopLine P [66](#)
CiA402 receipt PDO (Rx-PDO) [66](#)
CiA402 transmit PDO (Tx-PDO) [67](#)
Clear process data (C13885) [100](#)
Codes [96](#)
CoE objects [79](#)
Colour code of the Ethernet cable [30](#)
Commissioning [33](#)
Communication medium [15](#)
Communication parameters in »TwinCAT« and »Engineer« [45](#)
Communication profile [15](#)
Communication time [19](#)
Communication-relevant parameters of the standard device [96](#)
Configuration of the master [34](#)
Configuring process data [35](#)
Configuring the Controller (EtherCAT master) [34](#)
Configuring the EtherCAT master [34](#)
Conformities [15](#)
Conventions [7](#)
Conventions used [7](#)
Cycle times [15](#)

D

Data transfer [54](#)
Datagrams [56](#)
DC active (C13883) [100](#)
DC configuration in the Inverter Drive 8400 (slave) [38](#)
DC configuration in the master [38](#)
DC master [37](#)
Determining the cycle time [35](#)
Device- and application-specific safety instructions [11](#)
Device identification [35](#)
Device profile [15](#)
Device protection [11](#)
Diagnostics [84](#)
Diagnostics with the »Engineer« [89](#)
Dimensions [20](#)
Display of emergency data (C13867) [99](#)
Distributed Clocks (DC) [37](#)
Document history [6](#)
Download [74](#)

E

Electrical installation [25](#)
E-mail to Lenze [108](#)
EMC-compliant wiring [25](#)
Emergency messages [90](#)
Emergency requests [90](#)
Error: Lenze setting loaded (error message) [94](#)
Error code [90](#)
Error messages [92](#)
 Causes and remedies [93](#)
 Short overview [92](#)
Error number
 0x01bc3100 [93](#)
 0x01bc5531 [93](#)
 0x01bc5532 [93](#)
 0x01bc5533 [93](#)
 0x01bc6010 [93](#)
 0x01bc6011 [94](#)
 0x01bc6100 [94](#)
 0x01bc6101 [94](#)
 0x01bc641f [94](#)
 0x01bc6420 [94](#)
 0x01bc6430 [94](#)
 0x01bc8131 [95](#)
 0x01bc8700 [95](#)
Establishing a master - slave connection [68](#)
Establishing an online connection to the »Engineer« [40](#)
Establishing communication [53](#)
EtherCAT ADS communication parameters in »TwinCAT« and »Engineer« [45](#)
EtherCAT connection [27](#)
EtherCAT datagrams [56](#)
EtherCAT error messages
 Causes and remedies [93](#)
 Short overview [92](#)
EtherCAT frames [55](#)
EtherCAT objects of the communication module [80](#)
EtherCAT state machine [57](#)
Ethernet cable [29](#)
Ethernet cable colour code [30](#)
Ethernet cable, structure [29](#)
External voltage supply [31](#)

F

Feedback to Lenze [108](#)
Fieldbus status displays [86](#)
Firmware compilation date (C13901) [101](#)
Firmware product type (C13900) [101](#)
Firmware version (C13902) [102](#)
Frame structure [55](#)

G

Gateway Controller -> configure EtherCAT [41](#)
Gateway Controller -> configure EtherCAT ADS (Beckhoff) [43](#)
General data [15](#)
General safety and application notes [10](#)

I

Identification [12](#)
Indexing of the Lenze codes [69](#)
Initial switch-on [53](#)
Installation [21](#)
Installing device description files [34](#)
Interface for communication [15](#)
Interfaces [14](#)
Internal error (error message) [94](#)
Interruption of EtherCAT communication [82](#)
Interruption of internal communication [83](#)
Invalid module configuration (error message) [94](#)

L

LED status displays [84](#)
Line topology [26](#)
Lost connection to 8400 base device (error message) [93](#)

M

Mailbox datagram [69](#)
Mailbox protocol [15](#)
Max. cable length [15](#)
Mechanical installation [22](#)
Memory
 No access (error message) [93](#)
 Read error (error message) [93](#)
 Write error (error message) [93](#)
Module status displays [85](#)
Monitoring [82](#)
Mounting for standard devices 0.25 kW and 0.37 kW [22](#)
Mounting for standard devices from 0.55 kW onwards [23](#)

N

Nameplate [12](#)
Network topology [15](#), [26](#)
Notes used [9](#)
Number of nodes [15](#)
Number of Rx PDOs (C13860) [98](#)
Number of Tx PDOs (C13859) [98](#)

O

Operating conditions [15](#)
Operational status quit (error message) [95](#)

P

Parameter data transfer [68](#)
Parameter reference [96](#)
Parameter set invalid (error message) [94](#)
Parameters of the communication module [97](#)
PDO mapping [60](#)
Pin assignment of the RJ45 sockets [28](#)
Process data [59](#)
Process data transfer [59](#)
Processing time [19](#)
Product codes for Inverter Drives 8400 [35](#)
Product description [12](#)
Product features [13](#)
Product ID [15](#)
Protection against uncontrolled restart [53](#)
Protection of persons [11](#)
Protective insulation [16](#)
Protocol data [19](#)

R

Reaction on communication failure (C13880) [99](#)
Reaction on MCI communication failure (C01501) [96](#)
Reading parameters (SDO upload) [70](#)
Replacing the communication module [24](#)
Residual hazards [11](#)
Response of the Lenze EtherCAT nodes during start-up [39](#)
Response time when exiting "Operational" (C13881) [100](#)
Restart by watchdog reset (error message) [93](#)
Revision ID [15](#)

S

Safety instructions [10](#)
Safety instructions (representation) [9](#)
Screenshots/application examples [5](#)
SDO abort codes (Abort codes) [81](#)
SDO download [74](#)
SDO upload [70](#)
Specification of the Ethernet cable [29](#)
State machine [57](#)
Station alias (C13899) [101](#)
Status displays (LEDs) [84](#)
Status displays at X246 and X247 [88](#)
Structure of the Emergency message [90](#)
Structure of the Ethernet cable [29](#)
Suppress emergency message in case of (C13887) [101](#)
Switch topology [26](#)
Sync frame failure detection [83](#)
Sync signal source (C01120) [96](#)
Sync telegram failure (error message) [95](#)
Synchronisation with "Distributed Clocks" (DC) [37](#)
System error messages [92](#)

T

Table of attributes [103](#)
Target group [5](#)
Technical data [15](#)
Terminals [14](#), [32](#)
Terminology used [8](#)
Terms [8](#)

U

Upload [70](#)
Using the communication module [12](#)

V

Validity of the documentation [5](#)
Vendor ID [15](#)
Voltage supply [15](#), [31](#)

W

Writing parameters (SDO download) [74](#)



Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

Perhaps we have not succeeded in achieving this objective in every respect. If you notice this, please send your suggestions and points of criticism in a short e-mail to:

feedback-docu@Lenze.de

Thank you for your support.

Your Lenze documentation team



Lenze Automation GmbH
Hans-Lenze-Str. 1
D-31855 Aerzen
Germany
☎ +49 (0)51 54 / 82-0
📠 +49 (0)51 54 / 82-28 00
✉ Lenze@Lenze.de
🌐 www.Lenze.com

Service

Lenze Service GmbH
Breslauer Straße 3
D-32699 Extertal
Germany
☎ 00 80 00 / 24 4 68 77 (24 h helpline)
📠 +49 (0)51 54 / 82-11 12
✉ Service@Lenze.de

EtherCAT®



E84DGFCTxxx

Inverter Drives 8400 motec-----

Communication Manual

EN



13564907

Contents

1	About this documentation	5
1.1	Document history	7
1.2	Conventions used	8
1.3	Terminology used	9
1.4	Notes used	10
2	Safety instructions	11
2.1	General safety and application notes	11
2.2	Device and application-specific safety instructions	12
2.3	Residual hazards	12
3	Product description	13
3.1	Application as directed	13
3.2	Features and variants	14
3.3	Connections and interfaces	15
4	Technical data	17
4.1	General data and operating conditions	17
4.2	Protocol data	18
4.3	Communication time	18
5	Installation	19
5.1	Mechanical installation	20
5.2	Electrical installation	21
5.2.1	Network topology	21
5.2.2	EtherCAT connection	23
5.2.3	External voltage supply	24
6	Commissioning	25
6.1	Before initial switch-on	25
6.2	Configuring the Controller (master)	26
6.2.1	Installing device description files	26
6.2.2	Automatic device identification	27
6.2.3	State change after "Operational" in the stay-alive operation	27
6.2.4	Configuring process data	29
6.2.5	Determining the cycle time	30
6.3	Address allocation	31
6.4	Initial switch-on	32
7	Data transfer	33
7.1	EtherCAT frames	34
7.2	EtherCAT datagrams	35
7.3	EtherCAT state machine	36
7.4	AL Status Code	37
8	Process data transfer	38
8.1	Accessing process data / PDO mapping	39
8.2	Configuring the port interconnection of the process data objects (PDO)	40
9	Parameter data transfer	44
9.1	Establishing a connection between master and slave	44
9.2	Reading and writing parameters	45
9.2.1	Reading parameters (SDO upload)	46
9.2.2	Writing parameters (SDO download)	50

Contents

9.3	Implemented CoE objects	54
9.4	EtherCAT objects of the Communication Unit	55
9.5	SDO abort codes (Abort codes)	56
10	Monitoring	57
10.1	Interruption of EtherCAT communication	57
10.2	Interruption of internal communication	58
11	Diagnostics	59
11.1	LED status displays	59
11.2	Diagnosing with the »Engineer«	61
11.3	Emergency requests / Emergency messages	62
12	Error messages	63
12.1	Short overview of the EtherCAT error messages	63
12.2	Possible causes and remedies	64
13	Parameter reference	67
13.1	Communication-relevant parameters of the operating system	67
13.2	Parameters of the Communication Unit	68
13.3	Table of attributes	72
	Index	74
	Your opinion is important to us	77

1 About this documentation

Contents

This documentation exclusively contains descriptions of the EtherCAT® bus system for the Inverter Drive 8400 motec.



Note!

This documentation supplements the **mounting instructions** and the **"Inverter Drives 8400 motec" hardware manual** supplied with the Communication Unit.

The hardware manual contains safety instructions which must be observed!

The properties and functions of EtherCAT for Inverter Drives 8400 motec are described in detail.

Examples illustrate typical applications.

The theoretical concepts are only explained to the level of detail required to understand the function of EtherCAT communication with Inverter Drives 8400 motec.

Depending on the software version of the inverter and the version of the installed engineering tool («Engineer», «PLC Designer»), the screenshots in this documentation may vary from the depiction in the engineering tool.

This documentation does not describe any software provided by other manufacturers. No liability can be accepted for corresponding data provided in this documentation. For information on how to use the software, please refer to the Controller (PLC, master) documents.

All brand names mentioned in this documentation are trademarks of their corresponding owners.



Tip!

Detailed information on EtherCAT can be found on the website of the EtherCAT Technology Group:

www.EtherCAT.org

About this documentation

Target group

This documentation addresses to persons who configure, install, commission, and maintain the networking and remote maintenance of a machine.



Tip!

Information and software updates for Lenze products are provided in the download area at:

www.Lenze.com

Information regarding the validity

The information given in this documentation is valid for the following devices:

Product series	Type designation	Version
Inverter Drives 8400 motec EtherCAT communication unit	E84DGFCTxNx	EtherCAT
	E84DGFCTxJx	EtherCAT + Safety

► [Features and variants](#) (14)

About this documentation

Document history

1.1 Document history

Version			Description
4.0	01/2019	TD23	General revision
3.0	05/2017	TD17	• From Communication Unit SW version V01.02: ▶ State change after "Operational" in the stay-alive operation (📖 27) • New layout
2.1	11/2012	TD17	EtherCAT® is a registered trademark of the Beckhoff Automation GmbH, Germany.
2.0	11/2011	TD17	Information about the EtherCAT register " AL Status Code " (📖 37) supplemented.
1.0	04/2011	TD17	First edition

About this documentation

Conventions used

1.2 Conventions used

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

Type of information	Highlighting	Examples/notes
Spelling of numbers		
Decimal	Normal spelling	Example: 1234
Hexadecimal	0x[0 ... 9, A ... F]	Example: 0x60F4
Binary • Nibble	In inverted commas Point	Example: '100' Example: '0110.0100'
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text		
Program name	» «	PC software Example: Lenze »Engineer«
Window	<i>italics</i>	The <i>message window</i> ... / The <i>Options</i> dialog box...
Control element	Bold	The OK button... / The Copy command... / The Properties tab... / The Name input field...
Sequence of menu commands		If several commands must be used in sequence to carry out a function, the individual commands are separated by an arrow: Select File → Open to...
Hyperlink	<u>underlined</u>	Optically highlighted reference to another topic. Can be activated with a mouse-click in this documentation.
Icons		
Page reference	 8	Optically highlighted reference to another page. Can be activated with a mouse-click in this documentation.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

About this documentation

Terminology used

1.3 Terminology used

Term	Meaning
 EtherCAT	EtherCAT® (E thernet for C ontroller and A utomation T echnology) is an Ethernet-based fieldbus system which fulfils the application profile for industrial real-time systems. EtherCAT® is a registered trademark and patented technology, licenced by Beckhoff Automation GmbH, Germany.
Inverter	Lenze inverters of the "Inverter Drives 8400 motec" product series
Standard device	
Drive Unit Communication unit Wiring Unit	The 8400 motec inverter has a modular structure that includes the following modules: "Drive Unit", "Communication Unit", and "Wiring Unit". • The drive unit is available in different power settings. • In case of the communication unit you can select between: • Without fieldbus (basic I/O, standard I/O, extended I/O) • AS interface (without safety/with safety STO) • CANopen (without safety/with safety STO) • EtherCAT (without safety/with safety STO) • EtherNET/IP (without safety/with safety STO) • PROFIBUS (without safety/with safety STO) • PROFINET (without safety/with safety STO) • POWERLINK (without safety/with safety STO) • The wiring unit provides flexible connection possibilities for a simple integration into the power supply of the machine.
»Engineer«	Lenze PC software which supports you during the "Engineering" process (parameterisation, diagnostics, and configuration) throughout the whole life cycle, i. e. from planning to maintenance of the machine commissioned.
»PLC Designer«	
»TwinCAT«	Beckhoff PC software for EtherCAT configuration
Code	Parameter which serves to parameterise and monitor the inverter. In normal usage, the term is usually referred to as "Index".
Subcode	If a code contains several parameters, they are stored in "subcodes". This manual uses a slash "/" as a separator between code and subcode (e.g. "C00118/3"). This term is also referred to as "subindex" in common parlance.
Lenze setting	This setting is the default factory setting of the device.
Basic setting	
HW	Hardware
SW	Software
ESI	"EtherCAT Slave Information" (device description file in XML format)
CoE	CANopen over EtherCAT
I-1600.8	CoE index (hexadecimal representation) • In the example: index 0x1600, subindex 8
TA	Technology application
PDO	Process data object
SDO	Service data object
"Hot connect"	This feature provides for removing and connecting slave nodes during operation.

About this documentation

Notes used

1.4 Notes used

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

Safety instructions

Layout of the safety instructions:



Pictograph and signal word!

(characterise the type and severity of danger)

Note

(describes the danger and gives information about how to prevent dangerous situations)

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

Application notes

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

2 Safety instructions



Note!

It is absolutely vital that the stated safety measures are implemented in order to prevent serious injury to persons and damage to material assets.

Always keep this documentation to hand in the vicinity of the product during operation.

2.1 General safety and application notes



Danger!

If the following basic safety measures are disregarded, severe injuries to persons and damage to material assets may result.

- Lenze drive and automation components ...
 - must only be used as directed.
 - ▶ [Application as directed](#) (📖 13)
 - must never be commissioned if they display signs of damage.
 - must never be technically modified.
 - must never be commissioned if they are not fully mounted.
 - must never be operated without required covers.
 - during and after operation can have live, moving and rotating parts, depending on their degree of protection. Surfaces can be hot.
- The following applies to Lenze drive components ...
 - only use the accessories approved.
 - Only use original manufacturer spare parts.
- Observe all specifications contained in the enclosed documentation and related documentation.
 - This is the precondition for safe and trouble-free operation and for obtaining the product features specified.
 - ▶ [Features and variants](#) (📖 14)
 - The specifications, processes, and circuitry described in this document are for guidance only and must be adapted to your own specific application. Lenze does not take responsibility for the suitability of the process and circuit proposals.
- Only qualified personnel may work with and on Lenze drive and automation components. According to IEC 60364 and CENELEC, these are persons ...
 - are familiar with installing, mounting, commissioning, and operating the product.
 - who have the corresponding qualifications for their work.
 - who know and can apply all regulations for the prevention of accidents, directives, and laws applicable at the place of use.

Safety instructions

Device and application-specific safety instructions

2.2 Device and application-specific safety instructions

- During operation, the Communication Unit must be connected to the Wiring Unit and the Drive Unit.
- With external voltage supply, always use a separate power supply unit, safely separated to EN 61800-5-1 in every control cabinet (SELV/PELV).



Documentation for "Inverter Drives 8400 motec", control system, plant/machine

All the other measures prescribed in this documentation must also be implemented.
Observe the safety instructions and application notes contained in this manual.

2.3 Residual hazards

Device protection

- The Communication Unit contains electronic components that can be damaged or destroyed by electrostatic discharge.
 - ▶ [Installation](#) (📖 19)

3 Product description

3.1 Application as directed

The EtherCAT communication unit ...

- is a unit that can only be used in conjunction with the following modules:

Product series	Type designation
Inverter Drives 8400 motec Drive Unit	E84DGDVxxxxxxxxx
Inverter Drives 8400 motec Wiring Unit	E84DGVNxx

- is a device intended for use in industrial power systems.
- should only be used under the operating conditions prescribed in this documentation.
- may only be used in EtherCAT networks.
- can also be used without being connected to the EtherCAT network.

Any other use shall be deemed inappropriate!

Product description

Features and variants

3.2 Features and variants

The EtherCAT communication unit is available in the following versions:

Product series	Type designation	Product features				
		Enclosure	Connection EtherCAT	I/O: Connection via terminal	I/O: Connection via M12	Safety
Inverter Drives 8400 motec EtherCAT communication unit	E84DGFCTFNP	IP 65	M12	3× DI 1× DO	2× DI	
	E84DGFCTENP	IP 65	M12	2× DI	3× DI 1× DO	
	E84DGFCTFJP	IP 65	M12	3× DI 1× DO 1× AI	2× DI	●
	E84DGFCTEJP	IP 65	M12	3× DI	2× DI 1× DO 1× AI	●

- The EtherCAT communication unit is ...
 - mounted on top of the Wiring Unit (E84DGVNxx);
 - supplied internally via the Drive Unit (E84DGDVxxxxxxx) or externally via a separate voltage source.
- The I/O connections can be brought into the device via M12 connectors or cable glands.
- Devices without an integrated safety system (safety option) have no analog input and no relay output.
- The integrated safety system can be used on machines for the protection of persons.
- SDO transfer with CoE (CANopen over EtherCAT)
- Up to 10 process data words (max 20 bytes) can be sent to the master.
- Up to 8 process data words (max. 16 bytes) can be sent from the master.
- Communication with the Lenze »Engineer« (access to all Lenze parameters) is executed via the diagnostic interface of the Drive Unit.
- Access to all Lenze parameters with CoE (CANopen over EtherCAT)
- Cycle times: 1 ms or an integer multiple of 1 ms



"Inverter Drives 8400 motec" hardware manual

Here you will find detailed information on the integrated safety system (safety option).

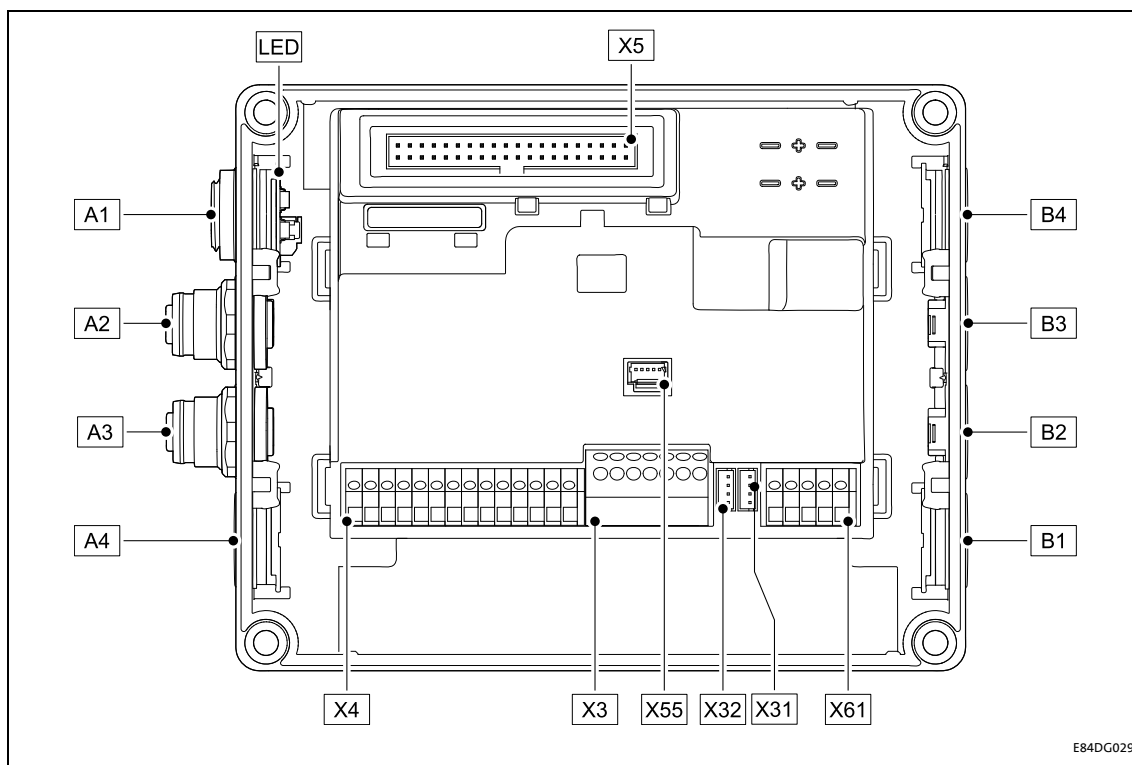
Reference manual / »Engineer« online help "Inverter Drives 8400 motec"

Here you will find detailed information on how to configure the safety system (safety option).

Product description

Connections and interfaces

3.3 Connections and interfaces



[3-1] EtherCAT communication unit

Pos.	Description
A1 / LED	Position for LEDs for EtherCAT status display ► LED status displays (□ 59)
A2	IN: EtherCAT input (M12 socket, 5-pole, D-coded) ► EtherCAT connection (□ 23)
A3	OUT: EtherCAT output (M12 socket, 5-pole, D-coded) ► EtherCAT connection (□ 23)
A4 B1 ... B4	Positions for further freely designable inputs and outputs: • Digital inputs • Digital output • Analog input (only for E84DGFCTxJx) • Relay output (only for E84DGFCTxJx) • Connection of "Safety Option" safety system (only for E84DGFCTxJx)
X3 / X4 / X61	Terminal strips for wiring the connections at A4 and B1 ... B4
X5	Plug connector for connection to the Drive Unit
X31	Plug connector for wiring the EtherCAT input (IN) to A2
X32	Plug connector for wiring the EtherCAT output (OUT) to A3
X55	Plug connector for the wiring of the LEDs to A1

Product description

Connections and interfaces

- By default, the EtherCAT connections and the LEDs for the EtherCAT status displays are already mounted and wired:
 - EtherCAT input to plug connector X31
 - EtherCAT output to plug connector X32
 - LEDs on plug connector X55
- At positions A1 ... A4 and B1 ... B4 it is also possible to connect the EtherCAT terminals and other connections (e.g. digital inputs) freely.
- For the connections, 5-pin M12 connectors or - alternatively - cable glands (cable cross-section max. 1.0 mm², AWG 18) can be used.
- The M12 connectors, cable glands and prefabricated system cables can be obtained from various manufacturers.
- Wire the M12 connectors or cable glands used to the corresponding contacts of the terminal strips/plug connectors X3, X4 and X61.



"Inverter Drives 8400 motec" hardware manual

Observe the notes and wiring instructions contained in this documentation.

4 Technical data



"Inverter Drives 8400 motec" hardware manual

Here you will find the **ambient conditions** and information on the **electromagnetic compatibility (EMC)** that also apply to the Communication Unit.

4.1 General data and operating conditions

Range	Values
Order designation	- E84DGFCTxNx (EtherCAT) - E84DGFCTxJx (EtherCAT + Safety)
Communication profile	EtherCAT
Supported device profile and mailbox protocol	CANopen over EtherCAT (CoE)
Communication medium	S/FTP (Screened Foiled Twisted Pair, ISO/IEC 11801 or EN 50173), CAT 5e
Interface for communication	- EtherCAT input (IN): M12 socket, 5-pole, D-coded - EtherCAT output (OUT): M12 socket, 5-pole, D-coded
Network topology	Line, switch
Type of node	EtherCAT slave
Number of nodes	Max. 65535 (in the entire network)
Max. cable length between two EtherCAT nodes	100 m (typical)
Vendor ID [hex]	0x3B
Product ID	841020
Revision ID	Dependent on the software version of the Communication Unit
Baud rate	100 Mbps, full duplex
Cycle times	1 ms or an integer multiple of 1 ms
External voltage supply	- U = 24 V DC (20 V - 0 % ... 29 V + 0 %) - I_{max} = 120 mA
Conformities, approvals	- CE - UR / cUR (see also hardware manual)

4.2 Protocol data

Range	Values
Process data words	1 ... 10 process data words to master (max. 20 bytes, 16 bits / word) 1 ... 8 process data words from master (max. 16 bytes, 16 bits / word)
Parameter data (mailbox size for CoE transfer)	Max. 128 bytes

4.3 Communication time

Parameter data (SDO)

The communication time for parameter data is the time between the transmission of an SDO request and the arrival of the corresponding response.

- The processing time in the inverter is approx. 10 ms + a tolerance of 20 ms (typically)
- Some codes may require a longer processing time (see reference manual / »Engineer« online help "Inverter Drive 8400 motec").

Process data (PDO)

The communication time for process data is the time between the reception of a PDO with setpoints and the return of a PDO with the current actual values.

The communication times for process data depend on ...

- processing time in the inverter (interval time of the application task, process data mode)
- Runtime on the bus (telegram length, number of nodes, PDO update time, instant of transmission of the EtherCAT-frame)

The processing time starts when the setpoints are taken over by the inverter at a point in time which is not synchronised with the EtherCAT master, and ends when the current actual values are provided at the EtherCAT interface.

The following processing time arises:

1.3 ms + 1.0 ms (tolerance) + interval time of the application task

5 Installation



Stop!

Electrostatic discharge

Electronic components within the Communication Unit can be damaged or destroyed by electrostatic discharge.

Possible consequences:

- The Communication Unit is defective.
- Fieldbus communication is not possible or faulty.
- I/O signals are faulty.
- The safety function is faulty.

Protective measures

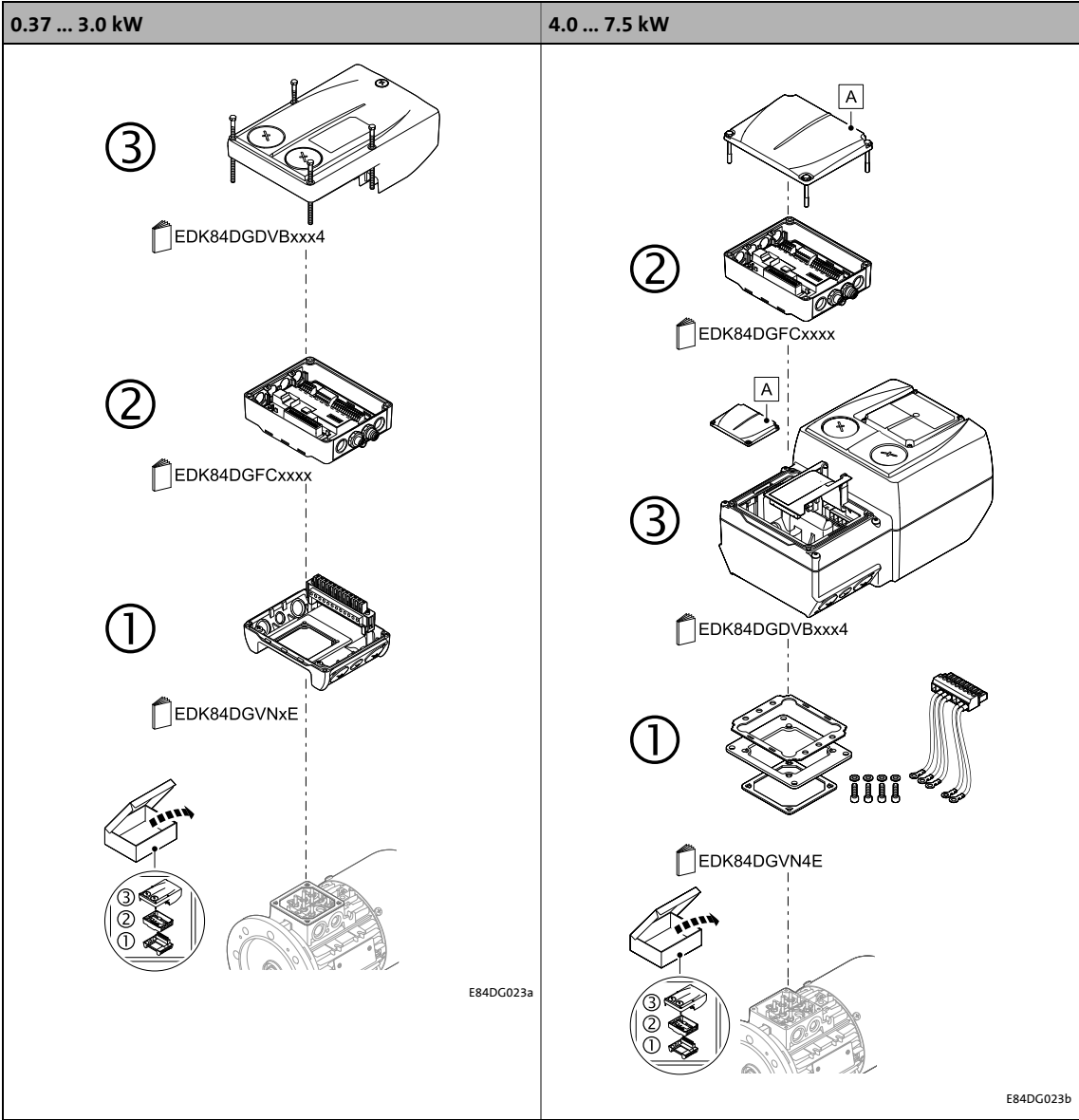
Discharge electrostatic charges before touching the Communication Unit.

5.1 Mechanical installation



Mounting instructions "Inverter Drives 8400 motec"

Here you will find detailed information on the installation.



[5-1] Mechanical installation of the 8400 motec components

Legend for fig. [5-1]	
1	Drive Unit
2	Communication unit
3	Wiring Unit
A	Cover of the Drive Unit
EDK84DG...	Mounting instructions of the Drive Unit, Communication Unit, Wiring Unit

5.2 Electrical installation



"Inverter Drives 8400 motec" hardware manual

Here you can find detailed information on ...

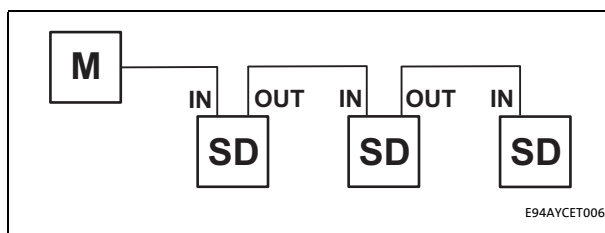
- the digital and analog inputs/outputs;
- the relay output;
- the integrated safety system (safety option);
- the wiring of the terminals.

Observe the notes and wiring instructions contained in this documentation.

5.2.1 Network topology

An EtherCAT telegram is sent through a pair of wires from the master to the slaves. The telegram is forwarded from slave to slave until it has passed through all the devices. Finally, the last slave returns the telegram to the master through a second pair of wires. In this way, EtherCAT always forms a logic ring topology, independent of the topology used.

Line topology



M = master

SD = slave device

[5-2] Line topology

- The devices are interconnected successively.
- In order to ensure trouble-free operation, it is required to assign and wire the EtherCAT inputs (IN) and EtherCAT outputs (OUT) correctly.
The receiving line is plugged into the X31 socket (IN), the forwarding line into the X32 socket (OUT).
- The direction of data transmission is from the master to the slaves.



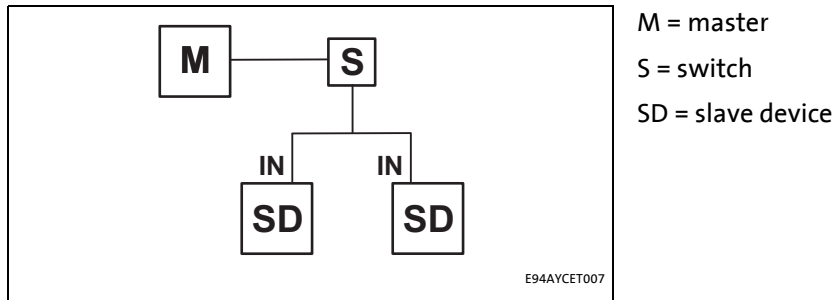
Tip!

The termination of the last EtherCAT node is effected automatically by the slave.

Installation

Electrical installation

Switch topology

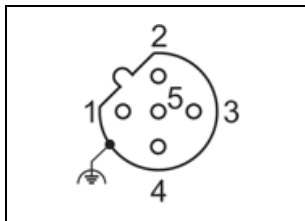


[5-3] Switch topology

The wiring can also be carried out in a star structure via an appropriate switch. For this, observe the additional runtimes.

5.2.2 EtherCAT connection

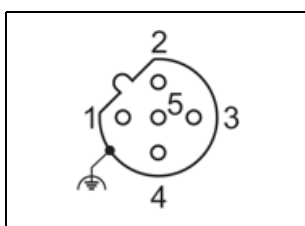
EtherCAT input (IN)



- M12 socket, 5-pole, D-coded
- Wiring at terminal strip X31

Pin	Signal	Description
1	Tx +	Data line (transmitted data, plus)
2	Rx +	Data line (received data, plus)
3	Tx -	Data line (transmitted data, minus)
4	Rx -	Data line (received data, minus)
5	-	not assigned

EtherCAT output (OUT)



- M12 socket, 5-pole, D-coded
- Wiring at terminal strip X32

Pin	Signal	Description
1	Tx +	Data line (transmitted data, plus)
2	Rx +	Data line (received data, plus)
3	Tx -	Data line (transmitted data, minus)
4	Rx -	Data line (received data, minus)
5	-	not assigned

5.2.3 External voltage supply

- The external voltage supply can be used to establish EtherCAT communication for commissioning and to query the data of the digital and analog inputs.
- Furthermore the external voltage supply serves to maintain EtherCAT communication if the main supply fails.
- The digital inputs RFR, DI1 ... DI5 and the analog input can continue to be evaluated.
- The external voltage supply is done via the terminals 24E and GND of the terminal strip X3.
- Permissible voltage (DC) / max. current:
 - $U = 24 \text{ V DC (} 20 \text{ V} - 0 \% \dots 29 \text{ V} + 0 \% \text{)}$
 - $I_{\text{max}} = 120 \text{ mA}$
- Access to parameters of a device that is disconnected from the mains is not possible.



"Inverter Drives 8400 motec" hardware manual

Here you can find detailed information on how to wire the Communication Unit.

6 Commissioning

During commissioning, system-related data such as motor parameters, operating parameters, responses, and parameters for fieldbus communication are defined for the inverter. For Lenze devices, this is done via the codes.

The codes of the inverter and communication are saved non-volatily as a data set in the memory module.

In addition, there are codes for diagnosing and monitoring the stations.

▶ [Parameter reference](#) (📖 67)

The data from the inverter or the memory module can only be read with the main voltage supply (400/500 V AC).

For commissioning with 24 V DC, only the data of the digital and analog inputs in the last two data words are valid and readable (see [Configuring process data](#) (📖 29)).

6.1 Before initial switch-on



Stop!

Before switching on the inverter for the first time, check the entire wiring with regard to completeness, short circuit, and earth fault.

6.2 Configuring the Controller (master)

For communication with the Communication Unit, the Controller (master) must be configured first. In order to configure EtherCAT networks, you always need a configuration software for the Controller, e.g.:

- Lenze »PLC Designer«
- Beckhoff »TwinCAT«

These are software systems for the programming of control programs, EtherCAT configuration, real-time execution and diagnostics.

- The basic parameters of the Communication Unit are stored in the internal configuration memory and can be used by the master for the node identification.
- For node detection (fieldbus scan), the corresponding device descriptions of the Lenze device family are used.

6.2.1 Installing device description files

The current XML device description files required the configuration of the EtherCAT node can be found in the download area on:

www.Lenze.com

The device description file **Lenze_Inverter_8400_IO_yyyymmdd.xml** must be installed via the EtherCAT configuration software.

Wildcards in the file name	
yyyy	Year
mm	Month
dd	Day

The description file summarises all EtherCAT-capable devices of the Inverter Drives 8400 series (Inverter Drives 8400 with EtherCAT from V01.xx, 8400 motec with EtherCAT from V01.xx).

6.2.2 Automatic device identification

- For a faultless integration of the EtherCAT slaves into a master configuration it is necessary to select the correct Lenze device in the EtherCAT configuration software.
- Each EtherCAT node is unambiguously identified by the configuration software by means of the product code (equal to the CoE object I-1018.2), the manufacturer's identification mark (0x3B) and the main software version of the Communication Unit.
 - ▶ [Implemented CoE objects](#) (66 54)
- In order that the configuration software selects the configuration specific for the EtherCAT node from the device description file, the product code is automatically set in the identity object.
- During initialisation, the product code is transferred to the master. Based on identification, the master can accept the corresponding settings from the device description.
- Product code of the Inverter Drives 8400 motec: 841020

6.2.3 State change after "Operational" in the stay-alive operation

[To be used from Communication Unit SW version V01.02!](#)

If the inverter connected via EtherCAT is not supplied with power when switching on the Lenze Controller, but the EtherCAT Communication Unit is externally supplied with 24 V (stay-alive operation), the inverter does not change to the "Operational" EtherCAT state.

The inverter remains in "Safe-Operational". The Controller resets the EtherCAT state and thus the entire interconnection from "Operational" to "Pre-Operational".

In order to enable the controller to switch the whole EtherCAT bus to "Operational", the "0x2995: Reach Operational" parameter ([C13930](#)) must be set to '1' (66 28). The parameter is a volatile parameter that is not saved and must therefore be reset to '1' each time the controller is restarted.

Error messages

- Logbook message: "Inverter_8400_Motec: CoE - Emergency request. id=0x0, len=8, errCode=0x1000, ErrReg=0x1, data: 0x0 0x0 0x31 0xbc 0x1."
- Message of the communication unit: [Quit "Operational" status \[0x01bc8131\] \(nt04: COM fault 4\)](#) (66 66)

Commissioning

Configuring the Controller (master)



How to configure the state change in the »PLC Designer«:

Preconditions:

- No further start parameters assigned to the standard device must be transmitted to the inverter. Only mapping data via the corresponding indices are permissible.
- All device parameters must be written from the application via function blocks and only if IOData1.Bit15 = TRUE (the control electronics of the inverter is active; the device parameters can be accessed).
- IOData1.Bit15 must be polled cyclically from the application. If the value is FALSE, no power supply is available and the control electronics of the inverter is switched off. However, the inverter continues to stay in the "Operational" state.

1. Add the **1 0x2995** parameter ([C13930](#)) to the list of the start parameters of the inverter.
2. Set the **2** parameter value to '1'.
3. Complete the entry with OK.

The screenshot shows the 'Inverter_8400_Motec' software window with the 'Startup Parameters' tab selected. A 'Select Item from Object Directory' dialog box is open, displaying a list of parameters. The parameter '(C13930) Reach Operational' is selected and highlighted with a red box. Below the list, the 'Name' field is set to '(13930) Reach Operational', 'Index' is '2995', 'SubIndex: 16#' is '0', and 'Value' is '1'. The 'Bitlength' is set to '8'. The 'Byte Array' checkbox is unchecked. The 'OK' button is visible.

Line	Index/Subindex	Name	Value	Bitlength	Comment
	16#1600:16#00	RxPDO 1			
	16#1A00:16#00	TxPDO 1			
	16#1C32:16#00	Sync Man 2 Synchronization			
	16#1C33:16#00	Sync Man 3 Synchronization			
1	16#2995:16#00	(C13930) Reach Operational			
	16#29B4:16#00	(C13899) Station Alias address			
	16#29C2:16#00	(C13885) Delete process data			
	16#29C6:16#00	(C13881) Process data monitoring time			
	16#29C7:16#00	(C13880) Process data monitoring reaction			
	16#5F96:16#00	(C0105) Deceleration time quick stop>			
	16#5FA4:16#00	(C0091) Motor coine phi>			
	16#5FA5:16#00	(C0090) Rated motor voltage>			
	16#5FA6:16#00	(C0089) Rated motor frequency>			
	16#5FA7:16#00	(C0088) Rated motor current>			
	16#5FA8:16#00	(C0087) Rated motor speed>			
	16#5FAE:16#00	(C0081) Rated motor power>			

Name: (13930) Reach Operational
Index: 2995 Bitlength: 8
SubIndex: 16# 0 Value: 1
☐ Byte Array

OK Cancel

6.2.4 Configuring process data

- The process data configuration is determined during the initialisation phase of the master (PDO mapping).
- Up to 10 process data words (max 20 bytes) can be sent to the master.
- Up to 8 process data words (max. 16 bytes) can be sent from the master.
- Independent of the configured length of the process data from the Inverter Drive 8400 motec to the master, the I/O data are always entered into the last two words:

Data word	Bits	Function	Value / description	
Word 1	0 ... 9	Analog input value (0 ... 10 V)	10 V = 1000	
	10	Digital input 3	0 (FALSE)	Open
			1 (TRUE)	Closed
	11	Digital input 4	0 (FALSE)	Open
			1 (TRUE)	Closed
	12	Digital input 5	0 (FALSE)	Open
			1 (TRUE)	Closed
	13	Reserved		
	14	I/O status	0 (FALSE)	Data in word 1/2 are not valid.
			1 (TRUE)	Data in word 1/2 are valid.
	15	Connection status of the inverter	0 (FALSE)	Inverter is offline ("stay-alive" operation)
			1 (TRUE)	Inverter is online
Word 2	0	RFR	0 (FALSE)	Open
			1 (TRUE)	Closed
	1	Digital input 1	0 (FALSE)	Open
			1 (TRUE)	Closed
	2	Digital input 2	0 (FALSE)	Open
			1 (TRUE)	Closed
	3	Digital input 3	0 (FALSE)	Open
			1 (TRUE)	Closed
	4	Digital input 4	0 (FALSE)	Open
			1 (TRUE)	Closed
	5	Digital input 5	0 (FALSE)	Open
			1 (TRUE)	Closed
	6 ... 13	Reserved		
	14	I/O status	0 (FALSE)	Data in word 1/2 are not valid.
			1 (TRUE)	Data in word 1/2 are valid.
	15	Connection status of the inverter	0 (FALSE)	Inverter is offline ("stay-alive" operation)
			1 (TRUE)	Inverter is online

- The process data configuration is predefined in the device description file for each application.
 - ▶ [Configuring the port interconnection of the process data objects \(PDO\)](#) (📖 40)
- If required, the process data length can be adapted by the user.
- The last internal information of the configured data must be deleted to shorten the configured the process data length. Process data words to the master must keep their last two I/O data words.

6.2.5 Determining the cycle time

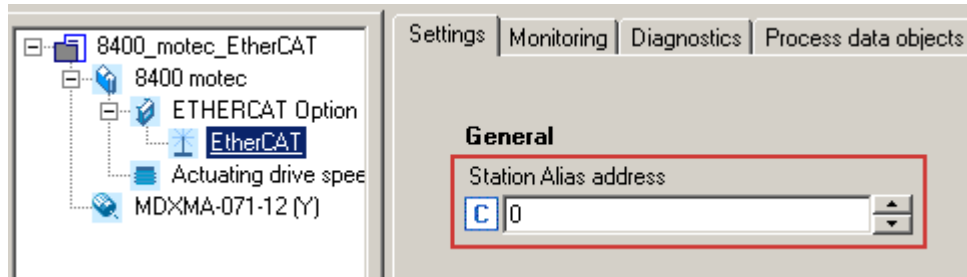
The process data objects (PDO) are transmitted cyclically between the master and the slaves.

The cycle time can be set via the EtherCAT configuration software.

6.3 Address allocation

The EtherCAT nodes are normally addressed via a fixed 16-bit address defined by the EtherCAT master. During start-up, the master assigns this address to each node, depending on the physical order in the EtherCAT network. The address is not saved and is lost when the device is switched off.

Via the **Station alias address** input field you can assign a fixed address to the EtherCAT slave.



Note!

- The station alias address must only be set if the node is part of a "hot connect" group.
- The station alias address must be unambiguous and may only be assigned once within the EtherCAT network.
- Use the same station alias address in the EtherCAT master and in the slave.

- Valid address range: 0 ... 32767
 - Address 0 means that no station alias address is assigned.
 - Impermissible addresses are marked in red in the input field.
 - The address is written to code [C13899](#).
- In addition, specify the use of the fixed addressing on the master.
- The address assigned by the master is displayed under code [C13864](#).
- Via standard device code **C00002**, execute the "**11: Save all parameter sets**" device command to activate the changed station alias address and to save it to the memory module.

6.4 Initial switch-on

Establishing communication

- To establish communication, the inverter must be supplied with mains voltage.
- EtherCAT communication requires voltage supply of the communication unit.
If this requirement is not met, the "CE04: MCI communication error" error message (error No. 01.0127.00002) is output. The error must be reset in the inverter so that EtherCAT communication can be established.
- The external voltage supply serves to keep up EtherCAT communication in the event of a main supply failure.
 - ▶ [External voltage supply](#) (📖 24)
- During mains connection, all parameters (codes) are read.
- Addressing can be carried out automatically via the EtherCAT master or manually via codes in the »Engineer«.
 - ▶ [Address allocation](#) (📖 31)

7 Data transfer

Compared with conventional Ethernet, the collision-free transfer of telegrams on the fieldbus makes EtherCAT a real-time capable bus system.

Communication is always initiated by the EtherCAT master which is the Controller. A telegram sent by the master passes through all EtherCAT slaves. The last slave of the communication chain sends the telegram back to the EtherCAT master. On the way back, the telegram is directly sent to the EtherCAT master, without being processed in the slaves.

EtherCAT transmits data in so-called "EtherCAT frames". The EtherCAT nodes only extract the data intended for them while the EtherCAT frame passes through the device. At the same time output data are inserted into the frame while it passes through the device. Read and write accesses are only executed on a small section of the entire EtherCAT frames – the datagrams. Therefore it is not necessary to receive the complete frame before it can be processed. Each datagram is transmitted with a minimum delay.

EtherCAT transmits process data, parameter data, configuration data, and diagnostic data between the controller (master) and the inverters (slaves) that are part of the fieldbus. The data are transmitted via corresponding communication channels as a function of their time-critical response (see [Process data transfer](#) (□ 38) / [Parameter data transfer](#) (□ 44)).

Data transfer

EtherCAT frames

7.1 EtherCAT frames

EtherCAT frames have the following structure:

EtherCAT header			Ethernet data				FCS
48 bits	48 bits	16 bits	11 bits	1 bit	4 bits	48 ... 1498 bytes	32 bits
Destination	Source	EtherType	Frame header			Datagrams	
			Length	Reserved	Type		

Ethernet header

The Ethernet header contains the following information:

- Target address of the EtherCAT frame (destination)
- Source address of the EtherCAT frame (source)
- Type of the EtherCAT frame (EtherType = 0x88A4)

Ethernet data

The Ethernet data contain the following information:

- Length of the datagrams within the EtherCAT frame (Length)
- One reserved bit (Reserved)
- Type of the datagrams within the EtherCAT frame (Type)
- EtherCAT datagrams (Datagrams)

FCS

Checksum of the EtherCAT frame

7.2 EtherCAT datagrams

EtherCAT datagrams have the following structure:

EtherCAT Command header	Data	WKC
10 bytes	Max. 1486 bytes	2 bytes

EtherCAT command header

The EtherCAT command header contains the following information:

- Command to be executed
- Addressing information
- Length of the data area (Data)
- Interrupt field

Data

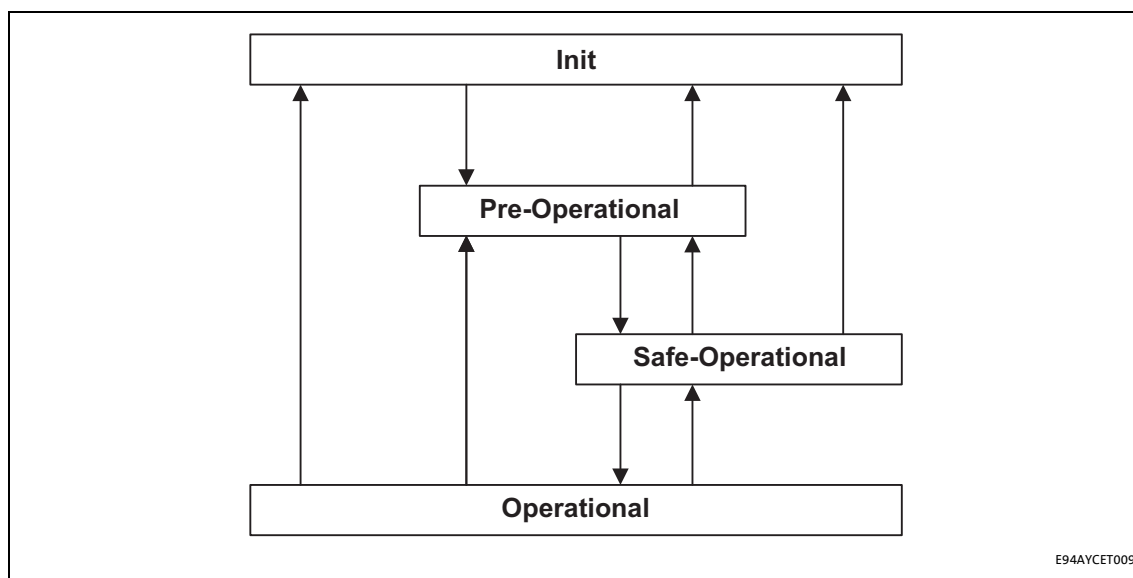
The data area contains the data of the command to be executed.

WKC

The working counter is evaluated by the master for monitoring the execution of the command.

7.3 EtherCAT state machine

Before communication is possible via EtherCAT, the fieldbus passes through the EtherCAT state machine during start-up. The following illustration depicts the possible state changes from the point of view of an EtherCAT slave:



[7-1] EtherCAT state machine

Status	Description
Init	- Initialisation phase - No SDO/PDO communication with the slaves - Device can be detected by fieldbus scan
Pre-Operational	- The fieldbus is active. - SDO communication (mailbox communication) is possible. - No PDO communication
Safe-operational	- SDO communication (mailbox communication) is possible. - PDO communication: - The input data in the process image are updated. - The output data from the process image are not transferred to the slaves.
Operational	- Normal operation - SDO communication - PDO communication - Fieldbus synchronisation has been successful (if used)



Note!

- An EtherCAT fieldbus scan is possible in every state.
- The SDO communication via the EtherCAT bus is only possible if at least the "Pre-Operational" state has been reached.
- Only in the transitional phases between states can bus nodes be in different states.

The current EtherCAT state is displayed in [C13861](#) and signalled via the "RUN" LED.

Possible errors in the state transitions are displayed in [C13879](#). Additionally, an error message is entered in the EtherCAT register "AL Status Code".

► [Diagnosing with the »Engineer«](#) (📖 61)

► [LED status displays](#) (📖 59)

7.4 AL Status Code

Information on how to access the EtherCAT register "AL Status Code" (address 0x0134:0x0135) can be found in the documentation of the EtherCAT master.

These error messages can be entered in the "AL Status Code" register:

AL Status Code [hex]	Description
0x0000	No error
0x0011	Invalid status change requested
0x0012	Unknown status requested
0x0013	"Bootstrap" status is not supported
0x0016	Invalid mailbox configuration "Pre-operational"
0x001A	Synchronisation error
0x001B	Sync manager watchdog
0x001D	Invalid output data configuration
0x001E	Invalid input data configuration
0x002B	Invalid input and output data
0x0030	Invalid configuration of DC synchronisation
0x9001	Firmware watchdog error
0x9002	Mapping error

8 Process data transfer

- Process data are transmitted by means of [EtherCAT datagrams](#) (35) via the process data channel.
- Process data serve to control the Inverter Drive 8400 motec.
- The transmission of process data is time-critical.
- Process data are cyclically transferred between the Controller (master) and the inverters (slaves) (continuous exchange of current input and output data).
- The master can directly access the process data. In the PLC for instance, the data are directly stored in the I/O area.
- Up to 10 process data words (max 20 bytes) can be sent to the master.
- Up to 8 process data words (max. 16 bytes) can be sent from the master.
- Process data are not saved in the Inverter Drive 8400 motec.
- Process data are e.g. setpoints, actual values, control words, and status words.

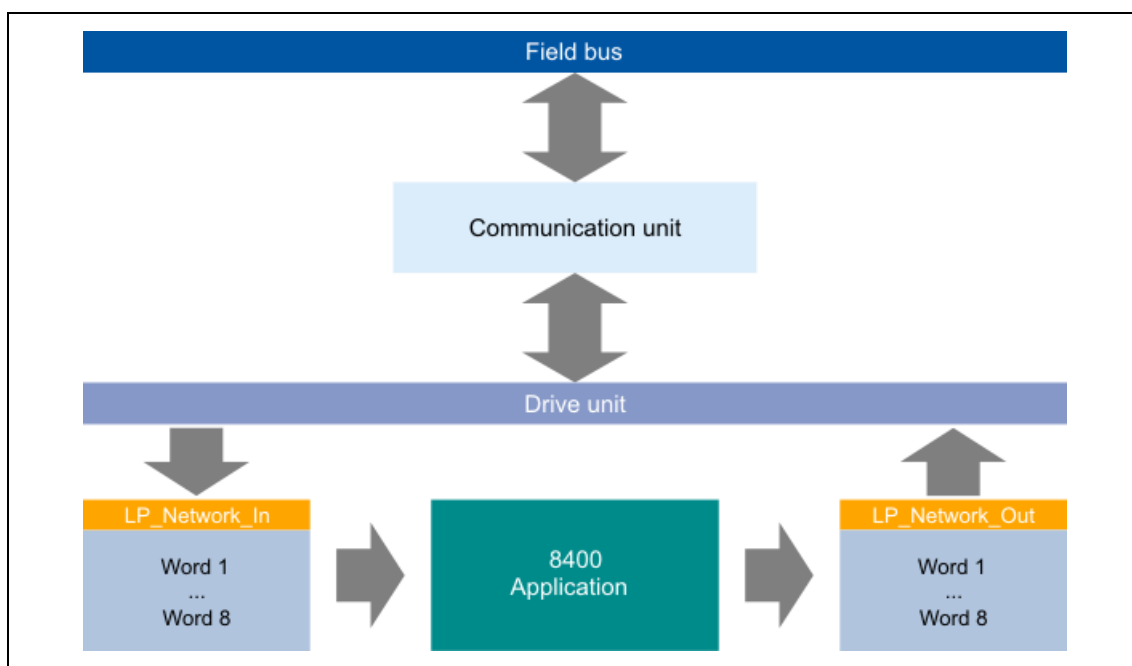
Process data transfer

Accessing process data / PDO mapping

8.1 Accessing process data / PDO mapping

Process data are transferred via the MCI/CAN interface.

- Max. 8 words (16 bits/word) per direction can be exchanged.
- The process data are accessed via the **LP_Network_In** and **LP_Network_Out** port blocks. These port blocks are also called process data channels.
- Port block **LP_Network_In** maps the MCI PDOs received.
- Port block **LP_Network_Out** maps the MCI PDOs to be transmitted.
- The port/function blocks of the process data objects (PDO) are interconnected via the Lenze »Engineer«.



[8-1] External and internal data transfer between the bus system, inverter, and application



Reference manual / »Engineer« online help for "Inverter Drive 8400 motec"

Here you will find detailed information on the port/function block interconnection in the »Engineer« and on the port blocks.

Process data transfer

Configuring the port interconnection of the process data objects (PDO)

8.2 Configuring the port interconnection of the process data objects (PDO)



Note!

The »Engineer« screenshots shown on the following pages are only examples of the setting sequence and the resulting screens.

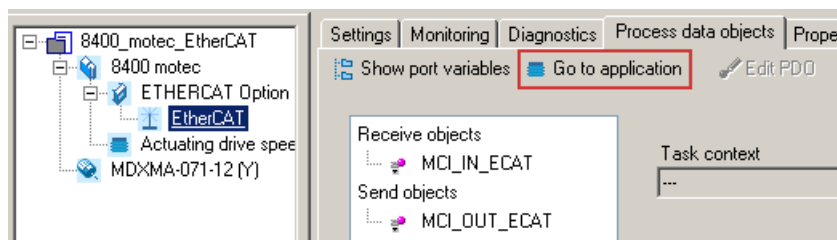
Depending on the software version of the inverter and the version of the »Engineer« software installed, the screenshots in this documentation may differ from the actual »Engineer« screens.

The preconfigured port interconnection of the process data objects is activated by setting code **C00007 = 40: Network (MCI/CAN)**.

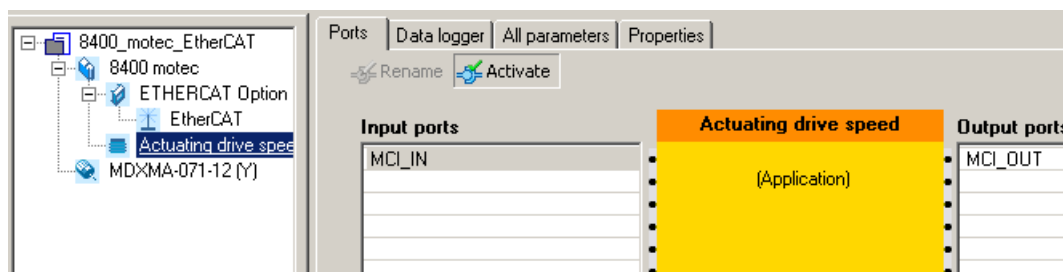


How to configure the port interconnection in the »Engineer«:

1. Go to the **Process data objects** tab and click **Go to application**.



2. The **Ports** tab displays the port blocks **MCI_IN** and **MCI_OUT**.



Process data transfer

Configuring the port interconnection of the process data objects (PDO)

3. Select the port to be configured and click the **Change Variable ...** button.

Ports | Data logger | All parameters | Properties

Rename Activate

Input ports

MCI_IN

Actuating drive speed

(Application)

Output ports

MCI_OUT

Mapping

EtherCAT/MCI_IN_ECAT : 0

Network default interconnection

<not defined>

Network default change...

Application variables

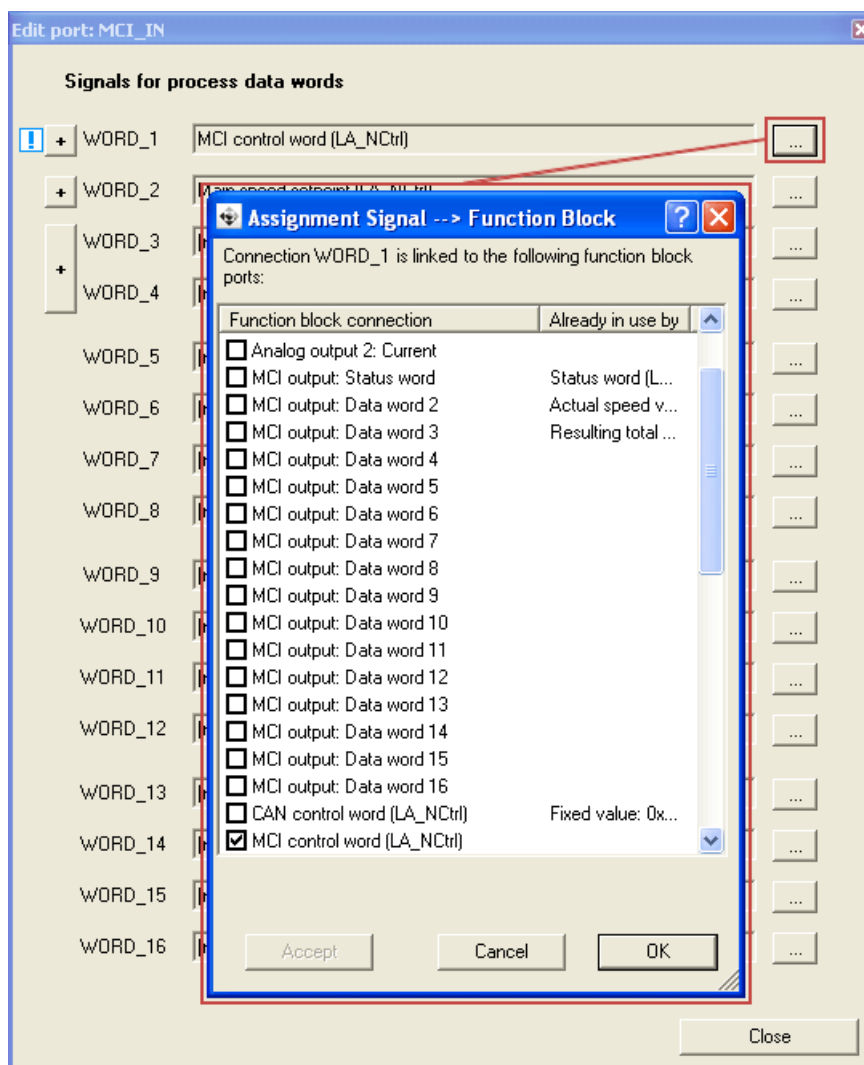
Name	Signal	Type	Length	Index	Online
WORD_1	LA_NCtrl:Network(MCI/CAN)...	WORD	16	C876/1	offline
WORD_2	LA_NCtrl: nMainSetValue_a	WORD	16	C876/2	offline
WORD_3	[not connected]	WORD	16	C876/3	offline
WORD_4	[not connected]	WORD	16	C876/4	offline
WORD_5	[not connected]	WORD	16	C876/5	offline
WORD_6	[not connected]	WORD	16	C876/6	offline
WORD_7	[not connected]	WORD	16	C876/7	offline
WORD_8	[not connected]	WORD	16	C876/8	offline
hCtrl_RR	hRFG_0	BOOL	1	...	offline

Change Variable...

Process data transfer

Configuring the port interconnection of the process data objects (PDO)

4. Via the  button, you can assign signals to the process data words in the *Assignment Signal --> Function Block* dialog window.
→ Select the signals and then confirm the selection with **OK**.

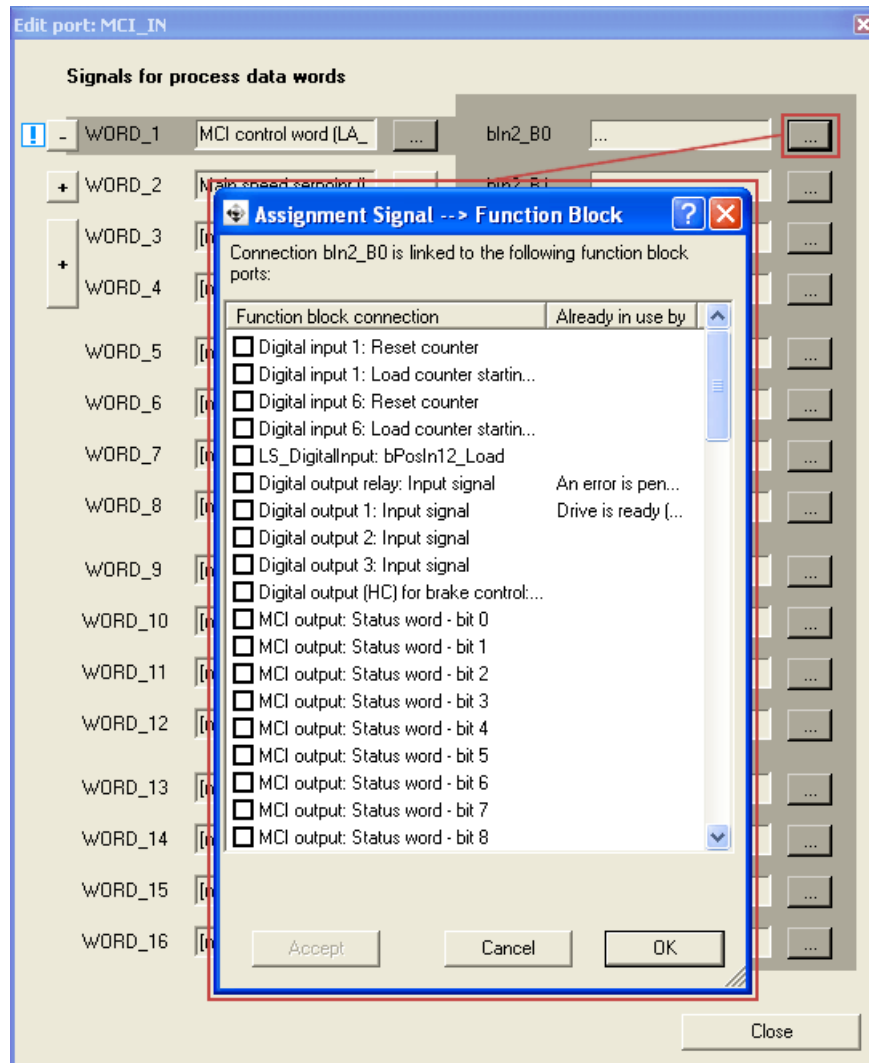


Process data transfer

Configuring the port interconnection of the process data objects (PDO)

For some process data words, you can also assign signals to the individual bits via the  and  buttons.

→ Select the signals and then confirm the selection with **OK**.



The current interconnection is only displayed if the following has been set for the control mode in code **C00007 = 40: Network (MCI/CAN)**.

5. Select the standard device code **C00002** and execute the device command "**11: Save all parameter sets**" to activate the changed port interconnection and save it in the memory module.

Parameter data transfer

Establishing a connection between master and slave

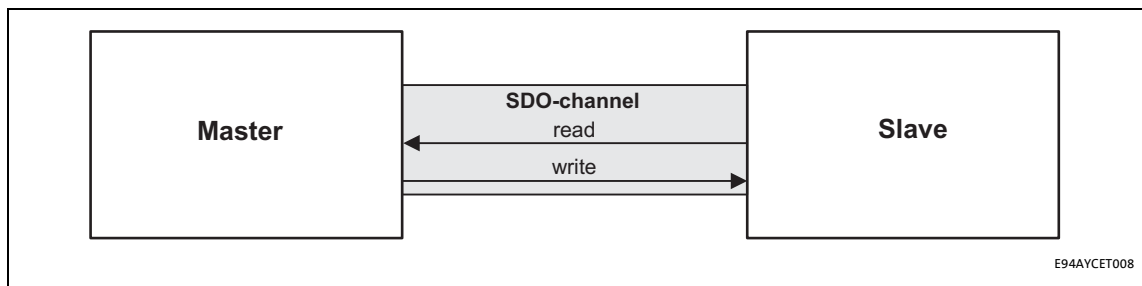
9 Parameter data transfer

Parameter data are transmitted via the fieldbus as so-called SDOs (Service Data Objects). The SDO services provide for the write and read access to the object directory.

- The SDO channel provides for access to [Implemented CoE objects](#) (📖 54) and Lenze codes by means of the CoE protocol.
- The transmission of parameter data is usually not time-critical.
- Parameter data are, for instance, operating parameters, motor data, diagnostic information.

9.1 Establishing a connection between master and slave

Basically a master can always request parameter jobs from a slave if the slave is at least in the "Pre-operational" state.



[9-1] Data communication via the SDO channel

Parameter data transfer

Reading and writing parameters

9.2 Reading and writing parameters

Parameters ...

- for instance are set for one-time system settings or if materials are changed within a machine;
- are transmitted with a low priority.

In the case of Lenze inverters, the parameters to be changed are contained in codes.

Indexing of the Lenze codes

If they are accessed via the Communication Unit, the codes of the Inverter Drive 8400 motec are addressed by the index.

The index for Lenze code numbers is in the manufacturer-specific area of the object directory between 8192 (0x2000) and 24575 (0x5FFF).

Conversion formula	
Index [dec]	Index [hex]
24575 - Lenze code	0x5FFF - Lenze code _{hex}

Example for C00002 (device commands)	
Index [dec]	Index [hex]
24575 - 2 = 24573	0x5FFF - 2 = 0x5FFD

Structure of a mailbox datagram

In a datagram, mailbox data are transferred within an EtherCAT frame. The data area of the mailbox datagram has the following structure:

Mailbox header	CoE header	SDO control byte	Index	Subindex	Data	Data
6 bytes	2 bytes	1 byte	2 bytes	1 byte	4 bytes	1 ... n bytes

Parameter data transfer

Reading and writing parameters

9.2.1 Reading parameters (SDO upload)

1. The master sends "Initiate Domain Upload Request".
2. The slave acknowledges the request with a positive response ("Initiate Domain Upload Response").

In the event of an error the slave responds with "Abort Domain Transfer".



Note!

In the case of jobs for the inverter, please make sure that you convert the code into an index.

► [Indexing of the Lenze codes](#) (45)

SDO Upload Request

Detailed breakdown of the data for an "SDO Upload Request":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Reserved	BYTE	4 bits (0 ... 3)	0x00
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is read. 0x01: The complete object is read. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x02: Upload request
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Reserved	DWORD	4 bytes	0x00

Parameter data transfer

Reading and writing parameters

SDO Upload Expedited Response

An "SDO Upload Expedited Response" is effected if the data length of the parameter data to be read is up to 4 bytes.

Detailed breakdown of the data for an "SDO Upload Expedited Response":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7) ... 0x03: Highest priority	
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x01: Size of the data in "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data 0x01: 3 bytes of data 0x02: 2 bytes of data 0x03: 1 byte of data
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is read. 0x01: The complete object is read. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x02: Upload response
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Data	DWORD	4 bytes	Data of the object

Parameter data transfer

Reading and writing parameters

SDO Upload Expedited Response

An "SDO Upload Normal Response" is effected if the data length of the parameter data to be read is ≥ 4 bytes.

Detailed breakdown of the data for an "SDO Upload Normal Response":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	$n \geq 0x0A$: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x01
	Transfer type		1 bit (1)	0x00: Normal transfer
	Data set size		2 bits (2, 3)	0x00
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is read. 0x01: The complete object is read. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x02: Upload response
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Complete size	DWORD	4 bytes	Total data length of the object
	Data	BYTE	$n - 10$ bytes	Data of the object

Parameter data transfer

Reading and writing parameters

Example

In the case of an **upload** to the index 0x5FD8 (standard value of C00039/1, preset setpoint_1 = 0x0FA0), the transmitted response structure contains the following data:

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	0x00
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x01: Length of the data in "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x02: 2 bytes of data
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is read.
	Command specifier		3 bits (5 ... 7)	0x02: Upload response
	Index	WORD	2 bytes	0xD8: Index low byte of the object 0x5F: Index high byte of the object
	Subindex	BYTE	1 byte	0x01
	Data	DWORD	2 bytes	0x0FA0

Parameter data transfer

Reading and writing parameters

9.2.2 Writing parameters (SDO download)

1. The master sends "Initiate Domain Download Request".
2. The slave acknowledges the request with a positive response ("Initiate Domain Download Response").

In the event of an error the slave responds with "Abort Domain Transfer".



Note!

In the case of jobs for the inverter, please make sure that you convert the code into an index.

► [Indexing of the Lenze codes](#) (45)

SDO Download Expedited Request

An "SDO Download Expedited Request" is effected if the data length of the parameter data to be written is up to 4 bytes.

Detailed breakdown of the data for an "SDO Download Expedited Request":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7) ... 0x03: Highest priority	
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Size indicator	BYTE	1 bit (0)	0x01: Size of the data in "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data 0x01: 3 bytes of data 0x02: 2 bytes of data 0x03: 1 byte of data
			Complete access	1 bit (4)
	Command specifier		3 bits (5 ... 7)	0x01: Download request
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Data	DWORD	4 bytes	Data of the object

Parameter data transfer

Reading and writing parameters

SDO Download Expedited Request

An "SDO Download Normal Request" is effected if the data length of the parameter data to be written is ≥ 4 bytes.

Detailed breakdown of the data for an "SDO Download Normal Request":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	$n \geq 0x0A$: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Size indicator	BYTE	1 bit (0)	0x01
	Transfer type		1 bit (1)	0x00: Normal transfer
	Data set size		2 bits (2, 3)	0x00
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is written. 0x01: The complete object is written. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x01: Download request
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Complete size	DWORD	4 bytes	Total data length of the object
	Data	BYTE	n - 10 bytes	Data of the object

Parameter data transfer

Reading and writing parameters

SDO Download Response

Detailed breakdown of the data for an "SDO Download Response":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the instructing party. Station address of the target if an EtherCAT slave is the instructing party.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x0
	Transfer type		1 bit (1)	0x0
	Data set size		2 bits (2, 3)	0x0
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is written. 0x01: The complete object is written. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x3: Download response
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Reserved	DWORD	4 bytes	0x00

Parameter data transfer

Reading and writing parameters

Example

In case of a **download**, the request structure transmitted to the index 0x1600 contains the following data:

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	0x00
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Size indicator	BYTE	1 bit (0)	0x01: Size of the data in "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is written.
	Command specifier		3 bits (5 ... 7)	0x01: Download request
	Index	WORD	2 bytes	0x00: Index low byte of the object 0x16: Index high byte of the object
	Subindex	BYTE	1 byte	0x01: Subindex of the object
	Data	DWORD	4 bytes	0x5C930110

Parameter data transfer

Implemented CoE objects

9.3 Implemented CoE objects

Lenze devices can be parameterised with both Lenze codes and the manufacturer-independent "CoE objects". In order to fully comply with EtherCAT communication, you must only use the CoE objects for parameterisation. The CoE objects described in this manual are defined in the "EtherCAT Specification, Part 6 – Application Layer Protocol Specification".

Index	Name	Subindex	Subindex name	Type	Bits	Access
0x1000	Device Type	-	-	UDINT	32	R
0x1001	Error register	-	-	USINT	8	R
0x1008	Device name	-	-	STRING(8)	64	R
0x1009	Hardware version	-	-	STRING(8)	64	R
0x100A	Software version	-	-	STRING(7)	56	R
0x1018	Identity	0	Number of elements	USINT	8	R
		1	Vendor ID	UDINT	32	R
		2	Product Code	UDINT	32	R
		3	Revision number	UDINT	32	R
		4	Serial Number	UDINT	32	R
0x1600	RxPDO 1	0	Number of elements	USINT	8	RW
		1 ... 8	Output object 1 ... 8	UDINT	32	RW
0x1A00	TxPDO 1	0	Number of elements	USINT	8	RW
		1 ... 10	Input object 1 ... 10	UDINT	32	RW
0x1C00	Sync Man Communication type	0	Number of elements	USINT	8	R
		1	Elements	UDINT	32	R
0x1C10	Sync Man 0 Assignment	0	-	UINT	16	R
0x1C11	Sync Man 1 Assignment	0	-	UINT	16	R
0x1C12	Sync Man 2 Assignment	0	Number of assigned RxPDOs	USINT	8	R
		1	PDO Mapping object index of assigned RxPDO	UDINT	32	R
0x1C13	Sync Man 3 Assignment	0	Number of assigned TxPDOs	USINT	8	R
		1	PDO Mapping object index of assigned TxPDO	UDINT	32	R
0x1C32	Sync Man 2 Synchronization	0	Number of elements	USINT	8	R
		1	Synchronization type	UINT	16	R
		2	Cycle time / ns	UDINT	32	R
		3	Shift time / ns	UDINT	32	R
		4	Sync types supported	UINT	16	R
		5	Minimum cycle time / ns	UDINT	32	R
		6	Minimum shift time / ns	UDINT	32	R
0x1C33	Sync Man 3 Synchronization	0	Number of elements	USINT	8	R
		1	Synchronization type	UINT	16	R
		2	Cycle time / ns	UDINT	32	R
		3	Shift time / ns	UDINT	32	R
		4	Sync types supported	UINT	16	R
		5	Minimum cycle time / ns	UDINT	32	R
		6	Minimum shift time / ns	UDINT	32	R

R: Read access only

RW: Read and write access

Parameter data transfer

EtherCAT objects of the Communication Unit

9.4 EtherCAT objects of the Communication Unit

The object directory displays the [Parameters of the Communication Unit](#) (□ 68) as objects:

Index	Name	Subindex	Subindex name	Type	Bits	Access	Index
0x29E5	C13850	All words from the inverter to the master	1 ... 10	All words from drive to master	UNSIGNED	16	R
0x29E4	C13851	All words from the master to the inverter	1 ... 8	All words from master to drive	UNSIGNED	16	R
0x29DC	C13859	Number of PDO words Tx	-	-	UNSIGNED	16	R
0x29DB	C13860	Number of PDO words Rx	-	-	UNSIGNED	16	R
0x29DA	C13861	Bus status	-	-	UNSIGNED	16	R
0x29D7	C13864	Active station address	-	-	UNSIGNED	16	R
0x29D4	C13867	Display of emergency data	-	-	STRING(8)	64	R
0x29C8	C13879	Bus error	-	-	UNSIGNED	16	R
0x29C7	C13880	Response to interruption of communication	1	-	UNSIGNED	8	RW
0x29C6	C13881	Monitoring time of data failure	-	-	UNSIGNED	16	RW
0x29C2	C13885	Delete process data	-	-	UNSIGNED	8	RW
0x29B4	C13899	Station alias address	-	-	UNSIGNED	16	RW
0x29B3	C13900	Firmware product type	-	-	STRING(8)	64	R
0x29B2	C13901	Firmware: Creation date	-	-	STRING(20)	160	R
0x29B1	C13902	Firmware version	-	-	STRING(11)	88	R
0x2995	C13930	Reach Operational	-	-	UNSIGNED	8	RW

R: Read access only

RW: Read and write access

Parameter data transfer

SDO abort codes (Abort codes)

9.5 SDO abort codes (Abort codes)

If an SDO request is evaluated negatively, a corresponding error code is output.

Index [hex]	Description
0x00000000	No error
0x05030000	The status of the toggle bit has not changed.
0x05040000	SDO protocol time-out
0x05040001	Invalid or unknown specification symbol for the client/server command
0x05040005	The space in the main memory is not sufficient.
0x06010000	Access to object not supported
0x06010001	Read access to a write-only object
0x06010002	Write access to a read-only object
0x06020000	An object does not exist in the object directory
0x06040041	An object cannot be mapped into the PDO
0x06040042	The number and/or length of the objects mapped would exceed the PDO length
0x06040043	General parameter incompatibility
0x06040047	General internal device incompatibility
0x06060000	Access has failed due to a fault in the hardware
0x06070010	The data type or the parameter length does not correspond
0x06070012	Incorrect data type (The parameter length is too large)
0x06070013	Incorrect data type (The parameter length is too small)
0x06090011	A subindex is not available
0x06090030	The value range for parameters is too great (only for write access)
0x06090031	The parameter value is too high
0x06090032	The parameter value is too low
0x06090036	The maximum value is lower than the minimum value
0x08000000	General error
0x08000020	Data cannot be transferred or saved to the application.
0x08000021	Data cannot be transferred or saved to the application because of local control.
0x08000022	Due to the current device state, data cannot be transferred to the application or stored in the application
0x08000023	The dynamic generation of an object directory has failed, or no object directory is available.

10 Monitoring

10.1 Interruption of EtherCAT communication

An interruption of the EtherCAT communication in the "Operational" state, e.g. due to cable break or failure of the EtherCAT master, is detected by the slave.



The response to the interruption of communication is controlled via the following settings:

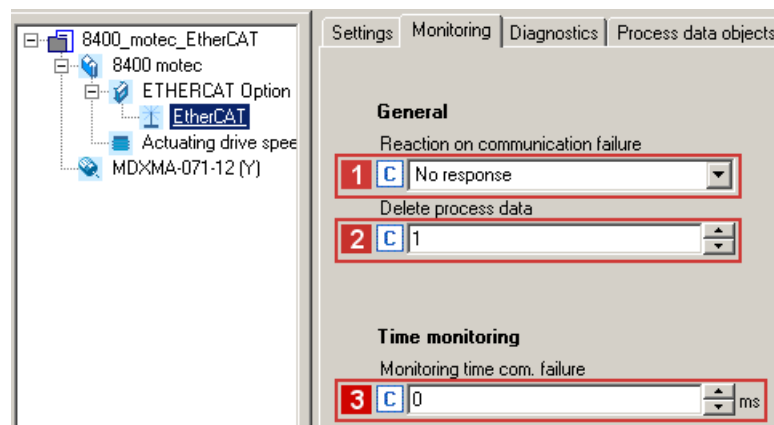
1. During the initialisation of the EtherCAT communication, the sync manager watchdog monitoring time determined in the master is transferred to the slave.

If the slave does not receive any valid process data in the "Operational" state, the process data are treated according to the setting in **2** [C13885](#). (The data sent last by the master can be used or reset to zero.)

When the watchdog monitoring time has elapsed, the slave changes to the "Error safe-operational" state (see [C13861](#)). The RUN (green) and ERR (red) LEDs are activated (see [LED status displays](#) (p 59)).

There is no response by the slave.

2. In order to trigger a response in the slave, you can set an additional **1** [response of the Inverter Drive 8400 motec](#) ([C13880](#)) in the »Engineer« on the **Monitoring** tab.



Set a **response delay** **3** ([C13881](#)) to delay execution of the response.

- A Lenze setting of "No response" deactivates this monitoring.
- Setting a response will activate the monitoring as long as a response time < 65356 ms is set.
- A change of monitoring will be effective immediately.
- The monitoring time expires as soon as communication in the "Operational" state is interrupted.

Monitoring

Interruption of internal communication

After the monitoring time has elapsed, the response set is executed with the error message "[Quit "Operational" status \[0x01bc8131\] \(nt04: COM fault 4\)](#)" ([📖 66](#)).

3. Via standard device code **C00002**, execute the "**11: Save all parameter sets**" device command to activate the changed parameter settings and to save it to the memory module.

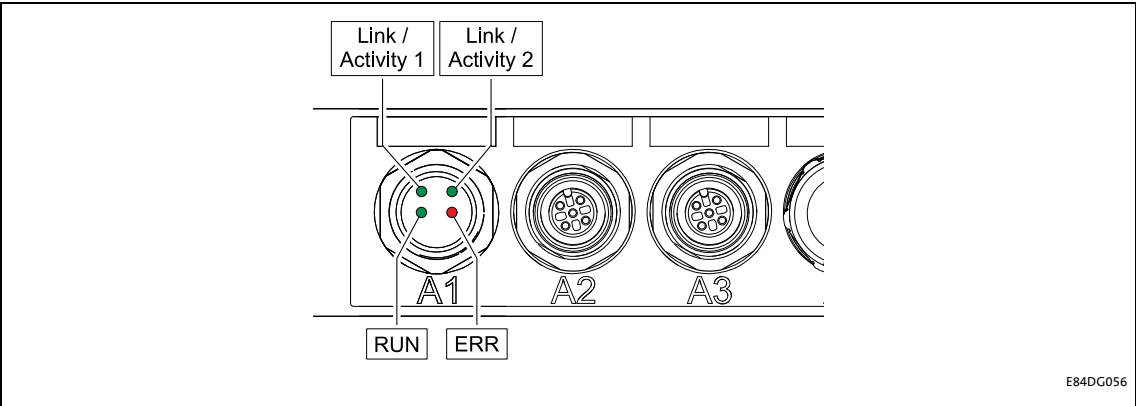
10.2 Interruption of internal communication

- The response in the case of a communication error between the Communication Unit and the Drive Unit can be set via code [C01501](#).
- The Communication Unit reports interrupted communication via an emergency telegram to the master and changes to the "Safe-Operational" state.
- The error message "[Connection to 8400 lost \[0x01bc3100\]](#)" ([📖 64](#)) is output.

11 Diagnostics

EtherCAT communication faults can be diagnosed via the LEDs of the communication unit.
Moreover, the »Engineer« provides diagnostic EtherCAT information.

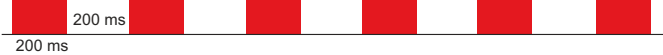
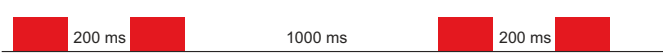
11.1 LED status displays



LED	Colour	Status	Description
Link / Activity 1 (A2)	green	Off	- No cable is connected to the EtherCAT input (IN). - No communication
		On	A cable is connected to the EtherCAT input (IN).
		Flickers	50 ms Communication at the EtherCAT input (IN) is active.
Link / Activity 2 (A3)	green	Off	- No cable is connected to the EtherCAT output (OUT). - No communication
		On	A cable is connected to the EtherCAT output (OUT).
		Flickers	50 ms Communication at the EtherCAT output (OUT) is active.

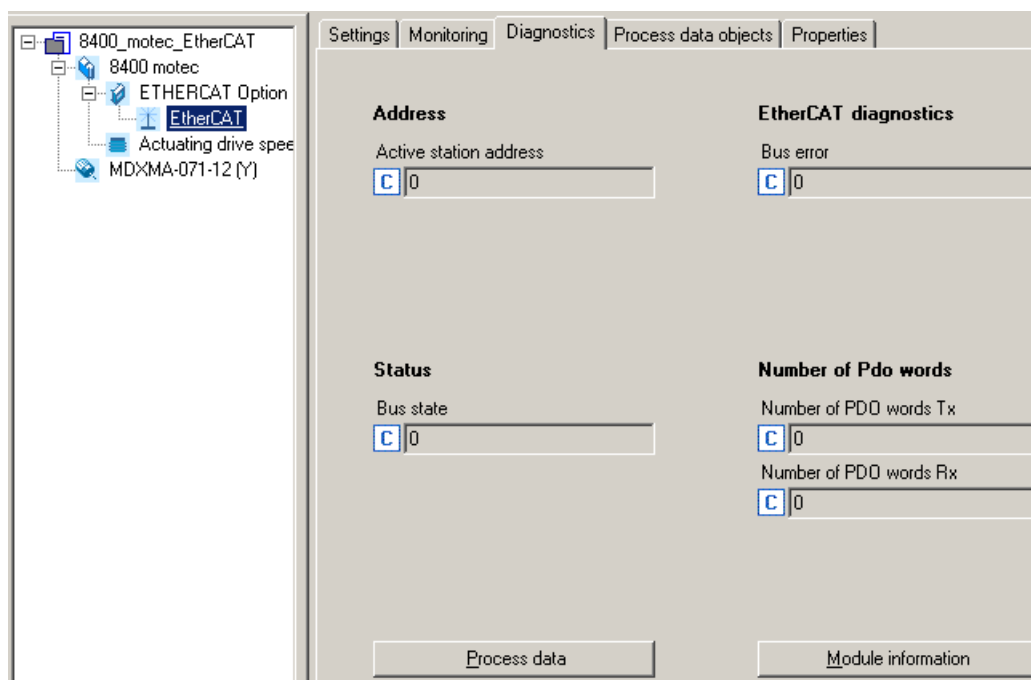
Diagnostics

LED status displays

LED	Colour	Status	Description
RUN	green	Off	The Communication Unit is not active on the fieldbus or is in the "Init" state.
		On	 The Communication Unit is in the "Operational" status
		blinking	 "Pre-operational" status is active: • Access to parameters and objects is possible. • No process data exchange.
		blinking once (single flash)	 "Safe-operational" status is active: • The data is transmitted from the inverter to the master. The data to the inverter are not yet active.
ERR	red	Off	No error
		blinking	 The configuration is invalid/faulty.
		blinking once (single flash)	 • A non requested state change has occurred. (The slave application has autonomously changed the EtherCAT status.) • Synchronisation error (The EtherCAT node automatically changes to the "Safe-operational" state.)
		blinking twice (double flash)	 An "Application Watchdog Timeout" or a "Sync Manager Watchdog Timeout" has occurred.

11.2 Diagnosing with the »Engineer«

In the »Engineer« under the **Diagnostics** tab, you will find EtherCAT diagnostics information.



11.3 Emergency requests / Emergency messages

Emergency messages are sent to the EtherCAT master once when the error status changes, i.e. ...

- if an error of the inverter or Communication Unit occurs;
- if an internal error of the Communication Unit is removed.

An "Emergency Request" on the fieldbus consists of the components "Mailbox Header", "CANopen Header" and the actual "Emergency Message":

Mailbox header	CANopen header	Emergency Message
6 bytes	2 bytes	8 bytes

Structure of the Emergency message

Example: Emergency message of the error "[Quit "Operational" status \[0x01bc8131\] \(nt04: COM fault 4\)](#)"

Byte 1	Byte2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Emergency Error code		Error Register (I-1001)	Reserved	Error code Inverter Drive 8400 motec / E84DGFCTxxx			
Low byte	High byte	Low byte	High byte	LOW word		High word	
				Low byte	High byte	Low byte	High byte
0x00	0x10	0x01	0x00	0x31	0x81	0xbc	0x01

- Bytes 1 and 2 display that there is an error.
- Byte 3 display the contents of the error register (I-1001).
- The error code is entered in bytes 5 ... 8.



Reference manual/»Engineer« Online help for the Inverter Drive 8400 motec
Detailed information on the error codes is provided here.

Error messages

Short overview of the EtherCAT error messages

12 Error messages

This chapter complements the error list in the reference manual and the »Engineer« online help for the Inverter Drive 8400 motec by EtherCAT error messages.

12.1 Short overview of the EtherCAT error messages



Reference manual/»Engineer« Online help for the Inverter Drive 8400 motec

Here you will find general information on diagnostics & fault analysis and on error messages.

The following table lists all EtherCAT error messages in numerical order of the error number. Furthermore the preset error response and – if available – the parameters for setting the error response are specified.



Tip!

If you click on the cross-reference in the first column, you will get a detailed description (causes and remedies) of the corresponding error message.

Error number			Error text	Error type	Adjustable in
hex	dec (Subject area no.)	dec (Error no.)			
0x01bc3100	444	12544	Connection to 8400 lost	1: No Response	C01501/2
0x01bc5531	444	21809	Memory: No access	1: No Response	C01501/2
0x01bc5532	444	21810	Memory: Read error	1: No Response	C01501/2
0x01bc5533	444	21811	Memory: Write error (nt14: COM fault 14)	1: No Response	C01501/2
0x01bc6010	444	24592	Restart by watchdog reset	1: No Response	C01501/2
0x01bc6011	444	24593	Internal error	1: No Response	C01501/2
0x01bc6100	444	24832	Internal error	1: No Response	C01501/2
0x01bc6101	444	24833	Internal error	1: No Response	C01501/2
0x01bc641f	444	25631	Invalid parameter record	1: No Response	C01501/2
0x01bc6420	444	25632	Error: Lenze settings loaded	1: No Response	-
0x01bc6430	444	25648	Invalid module configuration	1: No Response	C01501/2
0x01bc8131	444	33073	State "Operational" left (nt04: COM fault 4)	0: No Response	C13880

Error messages

Possible causes and remedies

12.2 Possible causes and remedies

This chapter contains all EtherCAT error messages in numerical order of the error number. Possible causes and remedies as well as responses to the error messages are described in detail.

Connection to 8400 lost [0x01bc3100]

Response (Lenze setting printed in bold)		Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information		
Cause	Remedy	
• The Communication Unit is supplied with external voltage, but the Inverter Drive 8400 motec is not supplied with voltage. • The Communication Unit is not connected correctly to the Drive Unit.	• Switch off and on again the voltage supply of the Inverter Drive 8400 motec. • Check wiring and terminals. • Check internal plug connection between Communication Unit and Drive Unit. For this purpose, the Inverter Drive 8400 motec must be unscrewed. Please observe the information in the mounting instructions of the Communication Unit and the Drive Unit! • If this error continues to occur, please contact the Lenze Service. (if required, the Communication Unit must be replaced.)	

Memory: No access [0x01bc5531]

Response (Lenze setting printed in bold)		Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information		
Cause	Remedy	
Access to memory was not possible.	Send the device and a description of the fault to Lenze.	

Memory: Read error [0x01bc5532]

Response (Lenze setting printed in bold)		Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information		
Cause	Remedy	
Parameter could not be read.	• Download application again (including module). • Send the device and a description of the fault to Lenze.	

Memory: Write error [0x01bc5533] (nt14: COM fault 14)

Response (Lenze setting printed in bold)		Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information		
Cause	Remedy	
Parameter could not be written.	• Download application again (including module). • Send the device and a description of the fault to Lenze.	

Error messages

Possible causes and remedies

Restart by watchdog reset [0x01bc6010]

Response (Lenze setting printed in bold)	Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information	
Cause	Remedy
Device is defective.	Send the device and a description of the fault to Lenze.

Internal error [0x01bc6011]

Response (Lenze setting printed in bold)	Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information	
Cause	Remedy
Device is defective.	Send the device and a description of the fault to Lenze.

Internal error [0x01bc6100]

Response (Lenze setting printed in bold)	Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information	
Cause	Remedy
Device is defective.	Send the device and a description of the fault to Lenze.

Internal error [0x01bc6101]

Response (Lenze setting printed in bold)	Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information	
Cause	Remedy
Device is defective.	Send the device and a description of the fault to Lenze.

Invalid parameter set [0x01bc641f]

Response (Lenze setting printed in bold)	Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information	
Cause	Remedy
No active parameter set could be loaded.	• Download application again (including module). • Send the device and a description of the fault to Lenze.

Error: Lenze setting loaded [0x01bc6420]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ warning ☐ Information	
Cause	Remedy
Access to parameter set was not possible.	• Download application again (including module). • Send the device and a description of the fault to Lenze.

Error messages

Possible causes and remedies

Invalid module configuration [0x01bc6430]

Response (Lenze setting printed in bold)	Setting: C01501/2 (☒ adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ warning ☐ Information	
Cause	Remedy
The number of process data words configured via EtherCAT does not comply with the length saved in the memory module.	Save parameter set with C00002/11 = 1 .

Quit "Operational" status [0x01bc8131] (nt04: COM fault 4)

Response (Lenze setting printed in bold)	Setting: C13880 (☒ Adjustable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☒ Quick stop by trouble ☒ Warning locked ☐ Warning ☒ Information	
Cause	Remedy
The EtherCAT data exchange was stopped in the "Operational" state. • Also see the chapter "Interruption of EtherCAT communication" (📖 57).	• Check cables and terminals. • The master must reset the inverter to the "Operational" status (C13930 = 1). If required, check a pending emergency message first. ▶ State change after "Operational" in the stay-alive operation (📖 27)

Parameter reference

Communication-relevant parameters of the operating system

13 Parameter reference

This chapter complements the parameter list and the table of attributes in the reference manual and the online help for the Inverter Drive 8400 motec by the parameters for EtherCAT communication.



Reference manual/online help for the Inverter Drive 8400 motec

Here you will find general information on parameters.

13.1 Communication-relevant parameters of the operating system

This chapter lists the communication-relevant parameters of the 8400 motec operating system in numerically ascending order.

C01501

Parameter Name:		Data type: UNSIGNED_8 Index: 23074 _d = 5A22 _h
C01501 Response in case of communication fault with MCI		
Configuration of monitoring functions for the Communication Unit		
Selection list (Lenze setting printed in bold)		
0	No response	
1	Error	
4	Warning Locked	
Subcodes	Lenze setting	Info
C01501/1	1: No Response	Resp. to MCI fault 1 • Response to a communication fault.
C01501/2	1: No Response	Resp. to MCI fault 2 • Response to a fault in the Communication Unit.
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

C01503

Parameter Name:		Data type: UNSIGNED_16 Index: 23072 _d = 5A20 _h
C01503 MCI timeout		
Setting range (min. value unit max. value)		
0	ms	1000
Subcodes	Lenze setting	Info
C01503/1	200 ms	MCI timeout
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1		

Parameter reference

Parameters of the Communication Unit

13.2 Parameters of the Communication Unit

This chapter lists the parameters of the Communication Unit E84DGFCTxxx (EtherCAT) in numerically ascending order.

C13850

Parameter Name: C13850 All words to master			Data type: UNSIGNED_16 Index: 10725 _d = 29E5 _h		
Display of the process data words (subcodes 1 ... 10) which are transferred from the inverter to the master. Only those which are configured are valid.					
Display range (min. value unit max. value)					
0		65535			
Subcodes			Info		
C13850/1			1st word		
...			...		
C13850/8			8th word		
C13850/9			I/O data 1		
C13850/10			I/O data 2		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT					

C13851

Parameter Name: C13851 All words from master			Data type: UNSIGNED_16 Index: 10724 _d = 29E4 _h		
Display of the process data words (subcodes 1 ... 8) which are transferred from the inverter to the master. Only those which are configured are valid.					
Display range (min. value unit max. value)					
0		65535			
Subcodes			Info		
C13851/1			1st word		
...			...		
C13851/8			8th word		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO MAP RX ☐ PDO MAP TX ☐ COM ☐ MOT					

C13859

Parameter Name: C13859 All words to standard device		Data type: UNSIGNED_16 Index: 10716 _d = 29DC _h	
Number of process data words to be sent			
Display range (min. value unit max. value)			
0			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO MAP RX ☐ PDO MAP TX ☐ COM ☐ MOT			

Parameter reference

Parameters of the Communication Unit

C13860

Parameter Name: C13860 All words from standard device			Data type: UNSIGNED_16 Index: 10715 _d = 29DB _h		
Number of process data words to be received					
Display range (min. value unit max. value)					
0		8			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT					

C13861

Parameter Name: C13861 Bus status			Data type: UNSIGNED_16 Index: 10714 _d = 29DA _h		
Display of the current bus status ▶ EtherCAT state machine (📖 36)					
Display range (min. value unit max. value)					
0		65535			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT					

C13864

Parameter Name: C13864 Active station address			Data type: UNSIGNED_16 Index: 10711 _d = 29D7 _h
Display of the station address allocated by the master			
Display range (min. value unit max. value)			
0		32767	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13867

Parameter Name: C13867 Display of emergency data		Data type: OCTET_STRING Index: 10708 _d = 29D4 _h
Display of the emergency data sent by the inverter (string with a length of 8 bytes). ▶ Emergency requests / Emergency messages (📘 62)		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13879

Parameter Name: C13879 Bus error		Data type: UNSIGNED_16 Index: 10696 _d = 29C8 _h
Bit coded display of the bus error Additionally, an error message is entered in the EtherCAT register " AL Status Code " ( 37).		
Value is bit-coded:		Info
Bit 0	General bus error	
Bit 1	Reserved	
...	...	
Bit 31	Reserved	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

Parameter reference

Parameters of the Communication Unit

C13880

Parameter Name: C13880 Monitoring reaction		Data type: UNSIGNED_8 Index: 10695 _d = 29C7 _h
The set response will be executed if the node detects that it is no longer in the "Operational" state and the monitoring time (C13881) has elapsed. The notes in code C13881 must be observed! ► Interruption of EtherCAT communication (57)		
Selection list (Lenze setting printed in bold)		
0	No response	
1	Error	
4	Warning Locked	
Subcodes	Lenze setting	Info
C13880/1	0: No Response	
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13881

Parameter Name: C13881 Response time when exiting "Operational"		Data type: UNSIGNED_16 Index: 10694 _d = 29C6 _h
If the "Operational" state is exited, the response parameterised with C13880 occurs after the time set here has elapsed. • A value of "0" or "65535" in this code deactivates the monitoring. ► Interruption of EtherCAT communication (57)		
Setting range (min. value unit max. value)		Lenze setting
0	ms	65535 0 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13885

Parameter Name: C13885 Clear process data		Data type: UNSIGNED_8 Index: 10690 _d = 29C2 _h
This code serves to set the process data which the slave must process for maintaining internal communication when the EtherCAT has exited the "Operational" state. • 0: The data last sent by the master are used. • 1: The process data contents is set to a value of "0". ► Interruption of EtherCAT communication (57)		
Setting range (min. value unit max. value)		Lenze setting
0		1 1
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13899

Parameter Name: C13899 Station Alias address		Data type: UNSIGNED_16 Index: 10676 _d = 29B4 _h
This code serves to set a station alias address. In order to use a station alias address, you must select a value > "0". • The station alias address must only be set if the node is part of a "hot connect" group. • The station alias address must be unambiguous and may only be assigned once within the EtherCAT network. • Use the same station alias address in the EtherCAT master and in the slave. ► Address allocation (31)		
Setting range (min. value unit max. value)		Lenze setting
0		32767 0
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

Parameter reference

Parameters of the Communication Unit

C13900

Parameter Name: C13900 Firmware product type	Data type: VISIBLE_STRING Index: 10675 _d = 29B3 _h
The code contains a string with a length of 8 bytes. The following identification code is displayed: "E84DFFET".	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13901

Parameter Name: C13901 Firmware compilation date	Data type: VISIBLE_STRING Index: 10674 _d = 29B2 _h
The code contains a string with a length of 20 bytes. Here, the compilation date ("MM DD YYYY") and time ("hh:mm:ss") of the software are provided. Example: "Mar 21 2005 12:31:21"	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13902

Parameter Name: C13902 Firmware version	Data type: VISIBLE_STRING Index: 10673 _d = 29B1 _h
The code contains a string with a length of 11 bytes. Here, the firmware version is provided. Example: "01.00.00.00"	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13930

Parameter Name: C13930 Reach Operational		Data type: UNSIGNED_8 Index: 10645 _d = 2995 _h
▶ State change after "Operational" in the stay-alive operation (📖 27) The parameter is a volatile parameter that is not saved and must therefore be reset to '1' each time the controller (master) is restarted. • Setting '1': State change after "Operational" • Setting '0': No state change		
Setting range (min. value unit max. value)		Lenze setting
0		1 0
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

Parameter reference

Table of attributes

13.3 Table of attributes

The table of attributes contains information that is required for communication with the inverter via parameters.

How to read the table of attributes:

Column		Meaning	Entry	
Code		Parameter name	Cxxxxx	
Name		Parameter short text (display text)	Text	
Index	dec	Index under which the parameter is addressed. The subindex for array variables corresponds to the Lenze subcode number.	24575 - Lenze code number	Is only required for access via a bus system.
	hex		5FFF _h - Lenze code number	
Data	DS	Data structure	E	Single variable (only one parameter element)
			A	Array variable (several parameter elements)
	DA	Number of array elements (subcodes)	Number	
	DT	Data type	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
	Factor	Factor for data transmission via a bus system, depending on the number of decimal positions	VISIBLE_STRING	ASCII string
			OCTET_STRING	
Access	R	Read access	☒ Reading permitted	
	W	Write access	☒ Writing permitted	
	CINH	Controller inhibit (CINH) required	☒ Writing is only possible when the controller is inhibited (CINH)	

Parameter reference

Table of attributes

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	Data type	Factor	R	W	CINH
C13850	All words from drive to master	10725	29E5	A	9	UNSIGNED_16	1	☒		
C13851	All words from master to drive	10724	29E4	A	8	UNSIGNED_16	1	☒		
C13859	All words to the basic device	10716	29DC	E	1	UNSIGNED_16	1	☒		
C13860	All words to the basic device	10715	29DB	E	1	UNSIGNED_16	1	☒		
C13861	Bus status	10714	29DA	E	1	UNSIGNED_16	1	☒		
C13864	Active station address	10711	29D7	E	1	UNSIGNED_16	1	☒		
C13867	Display of emergency data	10708	29D4	E	1	OCTET_STRING		☒		
C13879	Bus error	10696	29C8	E	1	UNSIGNED_16	1	☒		
C13880	Reaction on communication failure	10695	29C7	A	1	UNSIGNED_8	1	☒	☒	
C13881	Response time when exiting "Operational"	10694	29C6	E	1	UNSIGNED_16	1	☒	☒	
C13885	Delete process data	10690	29C2	E	1	UNSIGNED_8	1	☒	☒	
C13899	Station alias address	10676	29B4	E	1	UNSIGNED_16	1	☒	☒	
C13900	Firmware product type	10675	29B3	E	1	VISIBLE_STRING		☒		
C13901	Firmware compilation date	10674	29B2	E	1	VISIBLE_STRING		☒		
C13902	Firmware version	10673	29B1	E	1	VISIBLE_STRING		☒		
C13930	Reach Operational	10645	2995	A	1	UNSIGNED_8	1	☒	☒	

A

Abort codes [56](#)
Accessing process data [39](#)
Active station address (C13864) [69](#)
Address allocation [31](#)
AL Status Code [37](#)
All words from drive to master (C13850) [68](#)
All words from master to drive (C13851) [68](#)
All words from standard device (C13860) [69](#)
All words to standard device (C13859) [68](#)
Application as directed [13](#)
Application notes (representation) [10](#)
Application of the Communication Unit [13](#)
Approvals [17](#)
Automatic device identification [27](#)

B

Baud rate [17](#)
Before initial switch-on [25](#)
Bus Error (C13879) [69](#)
Bus state (C13861) [69](#)

C

C01501 | Resp. to communication error with MCI [67](#)
C01503 | MCI timeout [67](#)
C13850 | All words from drive to master [68](#)
C13851 | All words from master to drive [68](#)
C13859 | All words to standard device [68](#)
C13860 | All words from standard device [69](#)
C13861 | Bus state [69](#)
C13864 | Active station address [69](#)
C13867 | Display of emergency data [69](#)
C13879 | Bus error [69](#)
C13880 | Reaction on communication failure [70](#)
C13881 | Response time when exiting "Operational" [70](#)
C13885 | Delete process data [70](#)
C13899 | Station Alias address [70](#)
C13900 | Firmware
 Product type [71](#)
C13901 | Firmware
 Compilation date [71](#)
C13902 | Firmware
 Version [71](#)
C13930 | Reach Operational [71](#)
Codes [67](#)
CoE objects [54](#)
Commissioning [25](#)
Communication medium [17](#)
Communication profile [17](#)
Communication time [18](#)
Communication-relevant parameters of the operating system [67](#)
Configuration of the master [26](#)
Configuring process data [29](#)

Configuring the Controller (master) [26](#)
Configuring the port interconnection of the process data objects (PDO) [40](#)
Conformities [17](#)
Connections [15](#)
Conventions [8](#)
Conventions used [8](#)
Cycle times [17](#)

D

Data transfer [33](#)
Datagrams [35](#)
Delete process data (C13885) [70](#)
Determining the cycle time [30](#)
Device and application-specific safety instructions [12](#)
Device identification [27](#)
Device profile [17](#)
Device protection [12](#)
Diagnostics [59](#)
Diagnostics with the »Engineer« [61](#)
Display of emergency data (C13867) [69](#)
Document history [7](#)
Download [50](#)

E

Electrical installation [21](#)
E-mail to Lenze [77](#)
Emergency messages [62](#)
Emergency requests [62](#)
Error code [62](#)
Error Lenze setting loaded (error message) [65](#)
Error messages [63](#)
 Causes and remedies [64](#)
 Short overview [63](#)
Error number
 0x01bc3100 [64](#)
 0x01bc5531 [64](#)
 0x01bc5532 [64](#)
 0x01bc5533 [64](#)
 0x01bc6010 [65](#)
 0x01bc6011 [65](#)
 0x01bc6100 [65](#)
 0x01bc6101 [65](#)
 0x01bc641f [65](#)
 0x01bc6420 [65](#)
 0x01bc6430 [66](#)
 0x01bc8131 [66](#)
Establishing a master - slave connection [44](#)
Establishing communication [32](#)
EtherCAT connection [23](#)
EtherCAT datagrams [35](#)
EtherCAT error messages
 Causes and remedies [64](#)
 Short overview [63](#)
EtherCAT frames [34](#)

EtherCAT input (IN) [23](#)
EtherCAT objects of the Communication Unit [55](#)
EtherCAT output (OUT) [23](#)
EtherCAT state machine [36](#)
Exist. conn. to 8400 lost (error message) [64](#)
External voltage supply [24](#)

F

Feedback to Lenze [77](#)
Firmware
 Compilation date (C13901) [71](#)
 Product type (C13900) [71](#)
 Version (C13902) [71](#)
Frame structure [34](#)

G

General data [17](#)
General safety and application notes [11](#)

I

I/O data [29](#)
Indexing of the Lenze codes [45](#)
Initial switch-on [32](#)
Installation [19](#)
Installing device description files [26](#)
Interface for communication [17](#)
Interfaces [15](#)
Internal error (error message) [65](#)
Interruption of EtherCAT communication [57](#)
Interruption of internal communication [58](#)
Invalid module configuration (error message) [66](#)
Invalid Parameter Set (error message) [65](#)

L

LED status displays [59](#)
Line topology [21](#)

M

Mailbox datagram [45](#)
Mailbox protocol [17](#)
Max. cable length [17](#)
MCI timeout (C01503) [67](#)
Mechanical installation [20](#)
Memory
 No access (error message) [64](#)
 Read error (error message) [64](#)
 Write error (error message) [64](#)
Monitoring [57](#)

N

Network topology [17](#), [21](#)
Notes used [10](#)
nt04 - COM fault 4 (error message) [66](#)
nt14 - COM fault 14 (error message) [64](#)

Number of nodes [17](#)

O

Operating conditions [17](#)
Operational in the stay-alive operation [27](#)

P

Parameter data transfer [44](#)
Parameter reference [67](#)
Parameters of the Communication Unit [68](#)
PDO mapping [39](#)
Process data [38](#)
Process data transfer [38](#)
Product description [13](#)
Product features [14](#)
Product ID [17](#)
Protocol data [18](#)

Q

Quit "Operational" status (error message) [66](#)

R

Reach Operational (C13930) [71](#)
Reaction on communication failure (C13880) [70](#)
Reading parameters (SDO upload) [46](#)
Residual hazards [12](#)
Resp. to communication error with MCI (C01501) [67](#)
Response time when exiting "Operational" (C13881) [70](#)
Restart by watchdog reset (error message) [65](#)
Revision ID [17](#)

S

Safety instructions [11](#)
Safety instructions (representation) [10](#)
SDO abort codes (Abort codes) [56](#)
SDO download [50](#)
SDO upload [46](#)
State change after "Operational" in the stay-alive operation [27](#)
State machine [36](#)
Station Alias address (C13899) [70](#)
Status displays (LEDs) [59](#)
Stay-Alive operation
 State change after "Operational" [27](#)
Structure of the Emergency message [62](#)
Switch topology [22](#)
System error messages [63](#)

T

Table of attributes [72](#)
Target group [6](#)
Technical data [17](#)
Terminology used [9](#)
Terms [9](#)
Type of node [17](#)

Index

U

Upload [46](#)

V

Validity of the documentation [6](#)

Vendor ID [17](#)

Versions [14](#)

Voltage supply [17](#), [24](#)

W

Writing parameters (SDO download) [50](#)



Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

Perhaps we have not succeeded in achieving this objective in every respect. If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.com

Thank you very much for your support.

Your Lenze documentation team

Lenze Drives GmbH
Postfach 10 13 52, D-31763 Hameln
Breslauer Straße 3, D-32699 Extertal
Germany
HR Lemgo B 6478
 +49 5154 82-0
 +49 5154 82-2800
 sales.de@lenze.com
 www.lenze.com

Lenze Service GmbH
Breslauer Straße 3, D-32699 Extertal
Germany
 008000 24 46877 (24 h helpline)
 +49 5154 82-1112
 service.de@lenze.com



EtherCAT®

E94AYCET

Servo Drives 9400

Communication Manual

EN



13518011

Contents

1	About this documentation	4
1.1	Document history	6
1.2	Conventions used	7
1.3	Terminology used	8
1.4	Definition of the notes used	9
2	Safety instructions	10
2.1	General safety and application notes	10
2.2	Device and application-specific safety instructions	11
2.3	Residual hazards	11
3	Product description	12
3.1	Application as directed	12
3.2	Identification	12
3.3	Product features	13
3.4	Terminals and interfaces	14
4	Technical data	15
4.1	General data and operating conditions	15
4.2	Protective insulation	16
4.3	Protocol data	19
4.4	Communication time	19
4.5	Dimensions	21
5	Installation	22
5.1	Mechanical installation	23
5.1.1	Mounting	23
5.1.2	Disassembly	23
5.2	Electrical installation	24
5.2.1	Wiring according to EMC guidelines	24
5.2.2	Network topology	25
5.2.3	EtherCAT connection	26
5.2.4	Ethernet cable specification	27
5.2.5	External voltage supply	29
6	Commissioning	30
6.1	Before initial switch-on	30
6.2	Configuring the Controller (EtherCAT master)	31
6.2.1	Installing device description files	31
6.2.2	Automatic device identification	32
6.2.3	Configuring process data	33
6.2.4	Configuring safety process data	33
6.2.5	Determining the cycle time	33
6.3	Settings in »Engineer«	34
6.4	Address allocation	35
6.5	Synchronisation with "Distributed clocks" (DC)	36
6.5.1	DC configuration in the master	37
6.5.2	DC configuration in the Servo Drive 9400 (slave)	38
6.5.3	Behaviour of the communication module during the start	39
6.5.4	Process data mode	40
6.6	Establishing an online connection with the »Engineer«	41
6.6.1	Gateway Controller -> configure EtherCAT	42
6.6.2	Gateway Controller -> configure EtherCAT ADS (Beckhoff)	44

Contents

6.7	EtherCAT ADS communication parameters in »TwinCAT« and »Engineer«	46
6.7.1	Example: Structure without a Beckhoff Controller	46
6.7.2	Example: Structure with a Beckhoff DIN rail IPC CX1020	50
6.8	Initial switch-on	55
7	Data transfer	56
7.1	EtherCAT telegrams	57
7.2	EtherCAT datagrams	58
7.3	EtherCAT state machine	59
8	Process data transfer	61
8.1	Configuring process data (PDO mapping)	62
8.2	Setting PDO mapping with the »Engineer«	63
8.3	Mapping indices and codes	65
8.3.1	Structure of the mapping codes	66
8.3.2	FSOE mapping indices	66
9	Parameter data transfer	67
9.1	Establishing a connection between master and slave	67
9.2	Reading and writing parameters	68
9.2.1	Reading parameters (SDO upload)	69
9.2.2	Writing parameters (SDO download)	73
9.3	Communication objects (object directory)	78
9.4	SDO abort codes (Abort codes)	80
10	Safety over EtherCAT	81
11	Monitoring	82
11.1	Interruption of EtherCAT communication	82
11.2	Interruption of internal communication	83
11.3	Sync telegram failure detection	83
11.4	EtherCAT synchronisation loss	84
12	Diagnostics	85
12.1	LED status displays	85
12.1.1	Module status displays	86
12.1.2	EtherCAT status displays	87
12.1.3	Status display at the RJ45 sockets (X246 / X247)	89
12.2	Diagnostics with the »Engineer«	90
12.3	Emergency requests / Emergency messages	91
13	Error messages	92
13.1	Short overview of the EtherCAT error messages	92
13.2	Possible causes and remedies	93
14	Parameter reference	97
14.1	Parameters of the standard device that are relevant to communication	97
14.2	Parameters of the communication module for slot MXI1	100
14.3	Parameters of the communication module for slot MXI2	118
14.4	Table of attributes	136
	Index	139
	Your opinion is important to us	144

1 About this documentation

Contents

This documentation only contains descriptions for the E94AYCET communication module (EtherCAT®).



Note!

This documentation supplements the **mounting instructions** supplied with the communication module and the **Servo Drives 9400 hardware manual**.

The mounting instructions contain safety instructions which must be observed!

The features and functions of the communication module are described in detail.

Examples illustrate typical applications.

The theoretical context is only explained as far as it is required for understanding the function of the communication module.

This documentation does not describe any software provided by other manufacturers. No warranty can be given for corresponding data provided in this documentation. For information on how to use the software, please refer to the control system documents (Controller, EtherCAT master).

All brand names mentioned in this documentation are trademarks of their corresponding owners.



Tip!

Detailed information about EtherCAT can be found on the website of the EtherCAT Technology Group:

www.ethercat.org

Target group

This documentation addresses to persons who configure, install, commission, and maintain the networking and remote maintenance of a machine.



Tip!

Current documentation and software updates with regard to Lenze products can be found in the download area at:

www.lenze.com

Information regarding the validity

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

Extension module	Type designation	From hardware version	From software version upwards
EtherCAT communication module	E94AYCET	VE	03.00

Screenshots/application examples

All screenshots in this documentation are application examples. Depending on the firmware version of the communication module and the software version of the engineering tools installed (e.g. »Engineer«, »TwinCAT«), the screenshots in this documentation may differ from the actual screen representation.

1.1

Document history

Version			Description
10.2	11/2018	TD06	Corrections in chapter • Application as directed (□ 12)
10.1	08/2016	TD17	Safety over EtherCAT® logo added. • Terminology used (□ 8)
10.0	05/2016	TD17	• Update for software version 04.00 • Safety over EtherCAT (□ 81) • Information if the SM302 safety module is used
9.0	02/2014	TD17	General updates
8.0	01/2013	TD17	• Update for C13861 / C14861 (bus status) • New layout
7.0	07/2012	TD17	• General revision • Information about the EtherCAT register "AL Status Code" (□ 60) supplemented. • Chapter EtherCAT ADS communication parameters in »TwinCAT« and »Engineer« (□ 46) supplemented.
6.0	03/2011	TD17	General revision
5.0	11/2010	TD17	• General revision • Update for software version 03.00
4.0	04/2010	TD17	General revision
3.0	11/2009	TD17	Amendment of the information on the fieldbus synchronisation and general revision
2.0	07/2009	TD17	General revision
1.0	04/2008	TD17	First edition

1 About this documentation

1.2 Conventions used

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

Type of information	Highlighting	Examples/notes
Spelling of numbers		
Decimal	Normal spelling	Example: 1234
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Hexadecimal	0x[0 ... 9, A ... F]	Example: 0x60F4
Binary • Nibble	In inverted commas Point	Example: '100' Example: '0110.0100'
Text		
Version information	Blue text colour	All information that applies to from a certain software version of the device onwards is marked accordingly in this documentation. Example: This function extension is available from software version V3.0!
Program name	» «	The Lenze »Engineer« PC software...
Control element	Bold	The OK button... / the Copy command... / the Characteristics tab... / the Name input field...
Sequence of menu commands		If several commands are required in succession for executing a function, the single commands are separated by an arrow: Select the File → Open command to...
Hyperlink	Underlined	Optically highlighted reference to another topic. It is activated with a mouse-click in this online documentation.
Icons		
Page reference	(9)	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.

1.3

Terminology used

Term	Meaning
Inverters	Lenze inverters of the "Servo Drives 9400" product series
Standard device	
Code	Parameter which serves to parameterise and monitor the inverter. In normal usage, the term is usually referred to as "Index".
Subcode	If a code contains several parameters, they are stored in "subcodes". In the documentation, the slash "/" is used as a separator between the code and the subcode (e.g. "C00118/3"). In everyday language, the term is also referred to as "subindex".
CoE	CANopen over EtherCAT
Controllers	EtherCAT master
Control system	
DC	"Distributed clocks" for EtherCAT synchronisation
»Engineer«	Lenze PC software which supports you during the "Engineering" process (parameterisation, diagnostics, and configuration) throughout the whole life cycle, i. e. from planning to maintenance of the machine commissioned.
»PLC Designer«	
ESI	"EtherCAT slave information" (device description file in XML format)
	EtherCAT® (E thernet for C ontroller and A utomation T echnology) is an Ethernet-based fieldbus system which fulfils the application profile for industrial real-time systems. EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
	Safety over EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
"Hot connect"	This feature makes it possible to remove and connect slave field devices during operation.
HW	Hardware
I-1600.8	CoE index (hexadecimal representation) In the example: index 0x1600, subindex 8
Lenze setting	Default settings of the device, preconfigured ex works.
Standard setting	
PDO	Process data object
SDO	Service data object
SW	Software
»TwinCAT«	Beckhoff PC software for EtherCAT configuration

1.4 Definition of the notes used

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

Safety instructions

Structure 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!	Danger of property damage Reference to a possible danger that may result in property damage if the corresponding measures are not taken.

Application notes

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

2

Safety instructions

**Note!**

It is absolutely vital that the stated safety measures are implemented in order to prevent serious injury to persons and damage to material assets.

Always keep this documentation to hand in the vicinity of the product during operation.

2.1

General safety and application notes

**Danger!**

If the following basic safety measures are disregarded, severe injuries to persons and damage to material assets may result.

Lenze drive and automation components ...

- must only be used as directed.
 - [Application as directed](#) (12)
- must never be commissioned if they display any signs of damage.
- must never be technically modified.
- must never be commissioned if they are not fully mounted.
- must never be operated without required covers.
- during and after operation can have live, moving and rotating parts, depending on their degree of protection. Surfaces can be hot.

For Lenze drive components ...

- only use the accessories approved.
- only use genuine spare parts supplied by the manufacturer of the product.

observe all specifications contained in the enclosed documentation and related documentation.

- This is the precondition for safe and trouble-free operation and for obtaining the product features specified.
 - [Product features](#) (13)
- The specifications, processes, and circuitry described in this document are for guidance only and must be adapted to your own specific application. Lenze does not take responsibility for the suitability of the process and circuit proposals.

All work with or on Lenze drive components and automation components must be carried out only by qualified specialist personnel. According to IEC 60364 or CENELEC HD 384, these are persons ...

- are familiar with installing, mounting, commissioning, and operating the product.
- who have the corresponding qualifications for their work.
- who know and can apply all regulations for the prevention of accidents, directives, and laws applicable at the place of use.

2.2**Device and application-specific safety instructions**

- During operation, the communication module must be securely connected to the standard device.
- With external voltage supply, always use a separate power supply unit, safely separated to EN 61800-5-1 in every control cabinet (SELV/PELV).
 - ▶ [External voltage supply](#) (📖 29)
- A safety bus system (FSOE) can only be operated via the upper module slot (MXI1) of the Servo Drive 9400.
- Only use cables that meet the specifications listed.
 - ▶ [Ethernet cable specification](#) (📖 27)

**Documentation for the standard device, control system, system/machine**

All the other measures prescribed in this documentation must also be implemented. Observe the safety instructions and application notes contained in this manual.

2.3**Residual hazards****Protection of persons**

If Servo Drives 9400 are used on a phase earthed mains with a rated mains voltage ≥ 400 V, protection against accidental contact is not guaranteed without external measures.

- ▶ [Protective insulation](#) (📖 16)

Device protection

The communication module contains electronic components which may be damaged or destroyed by electrostatic discharge.

- ▶ [Installation](#) (📖 22)

3 Product description

3.1 Application as directed

3 Product description

3.1 Application as directed

The communication module ...

- is an accessory module that can be used in conjunction with the following standard devices:

Product series	Type designation	From hardware version	From software version upwards
Servo Drives 9400 HighLine	E94AxHxxxx	VA	01.51
Servo Drives 9400 PLC	E94AxPExxxx	VA	02.00
regenerative power supply module commissioning guidelines	E94ARNxxxx	VA	01.00

- is a device intended for use in industrial power systems.
- should only be used under the operating conditions prescribed in this documentation.
- may only be used in EtherCAT networks.

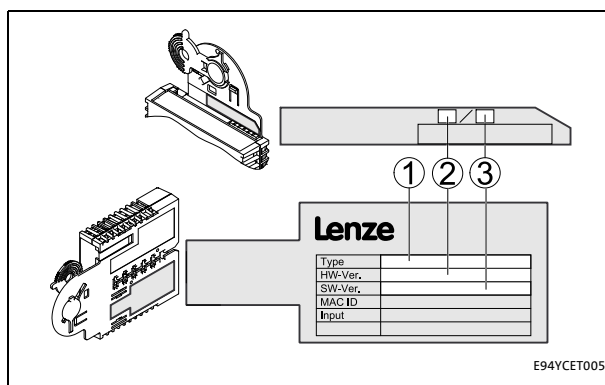
Any other use shall be deemed inappropriate!

From **SW version 04.00**, it is possible to communicate via FSoE in connection with a **SM302** safety module:

Product series	Type designation	From hardware version	From software version upwards
SM302 safety module	E94AYAF	VA	1.0

3.2 Identification

The type designation and the hardware and software versions of the communication module are indicated on the nameplate:



1 Type designation (type)

E94 Product series

A Version

Y Module identification: Extension module

C Module type: Communication module

ET EtherCAT

2 Hardware version (HW)

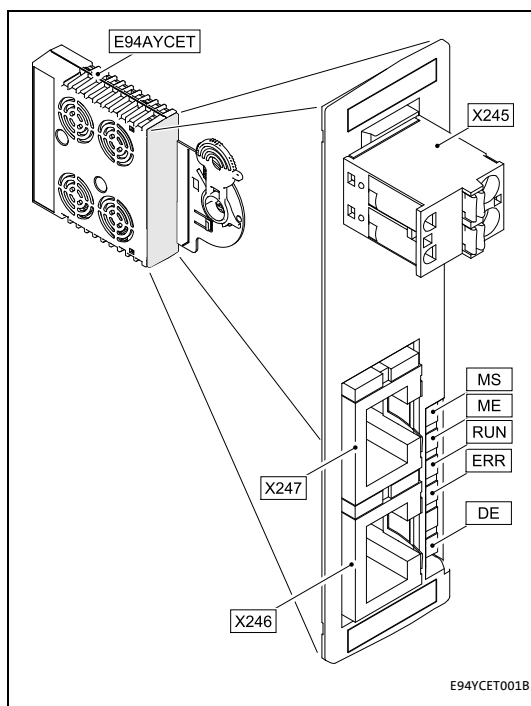
3 Software version (SW)

[3-1] Identification data

3.3**Product features**

- Interface module for the EtherCAT communication system to be connected to the expansion slots of the Servo Drives 9400
- The communication module can either be supplied internally by the standard device or externally by a separate voltage source.
- The CANopen device profile CiA402 in connection with Servo Drives 9400 HighLine is available from software version 7.0.
- Supports the "Distributed Clocks" (DC) functionality for synchronisation via the fieldbus
- Cycle times:
 - 1 ms or an integer multiple of 1 ms
 - max. 15 ms if "Distributed Clocks" (DC) are used
- Up to 32 process data words (16 bits/word) per direction can be exchanged.
- PDO transfer with CoE (CANopen over EtherCAT)
- Access to all Lenze parameters with CoE (CANopen over EtherCAT)
- [From SW version 04.00](#): Transmission of safe information via the FSoE protocol if the **SM302** safety module (E94AYAF) is used simultaneously

3.4 Terminals and interfaces



[3-2] E94AYCET communication module (EtherCAT)

X245 External voltage supply of the communication module

2-pin plug connector with spring connection

► [External voltage supply](#) (8 29)

X246 EtherCAT input (IN)

X247 EtherCAT output (OUT)

- RJ45 sockets

- each with 2 LED status displays for diagnostics

► [Network topology](#) (8 25)

► [EtherCAT connection](#) (8 26)

► [Status display at the RJ45 sockets \(X246 / X247\)](#) (8 89)

MS 5 LED status displays for diagnostics

ME ► [Module status displays](#) (8 86)

RUN ► [EtherCAT status displays](#) (8 87)

ERR

EN

4

Technical data

4.1

General data and operating conditions

Range	Values
Order designation	E94AYCET
Communication profile	EtherCAT
Supported device profile and mailbox protocol	CANopen over EtherCAT (CoE)
Safety over EtherCAT (from SW version 04.00)	Protocol according to the "ETG.5100" specification, version 1.2.0
Communication medium	S/FTP (Screened Foiled Twisted Pair, ISO/IEC 11801 or EN 50173), CAT 5e
Interface for communication	RJ45: Standard Ethernet (in accordance with IEEE 802.3), 100Base-TX (Fast Ethernet)
Network topology	Line, switch
Number of devices	Max. 65535 (in the entire network)
Max. cable length between two EtherCAT devices	100 m (typical)
Type of device	EtherCAT slave
Vendor ID [hex]	0x3B
Product ID	Depending on the standard device used and the selected technology application (see chapter Automatic device identification (□ 32)).
Revision ID	Depending on the main software version of the communication module (see chapter Identification (□ 12)).
Baud rate	100 Mbps, full duplex
Cycle times	1 ms or an integer multiple of 1 ms, max. 15 ms if "Distributed clocks" (DC) are used
Voltage supply	External supply via separate power supply unit +: U = 24 V DC (20.4 V - 0 % ... 28.8 V + 0 %), I = 130 mA -: reference potential for external voltage supply
Conformities, approvals	CE 2004/108/EC, EMC Directive 2006/95/EC, Low-Voltage Directive
	EAC Eurasian conformity TR CU: Technical Regulations of Customs Union
	TP TC 004/2011 (TR CU 004/2011) About the safety of low voltage equipment
	TP TC 020/2011 (TR CU 020/2011) Electromagnetic compatibility of technical means
	UL UL 508C, Power Conversion Equipment (file no. E132659)



Hardware manual "Servo Drives 9400"

Here you can find the **ambient conditions** and data on the **electromagnetic compatibility (EMC)**, which also apply to the communication module.

4.2

Protective insulation

**Danger!****Dangerous voltage**

If the Servo Drives 9400 are operated on a phase earthed mains with a rated mains voltage ≥ 400 V, external measures need to be implemented in order to ensure protection against accidental contact.

Possible consequences:

Death or severe injuries

Protective measures:

If protection against accidental contact is required for the control terminals of the inverter and the connections of the plugged device modules, ...

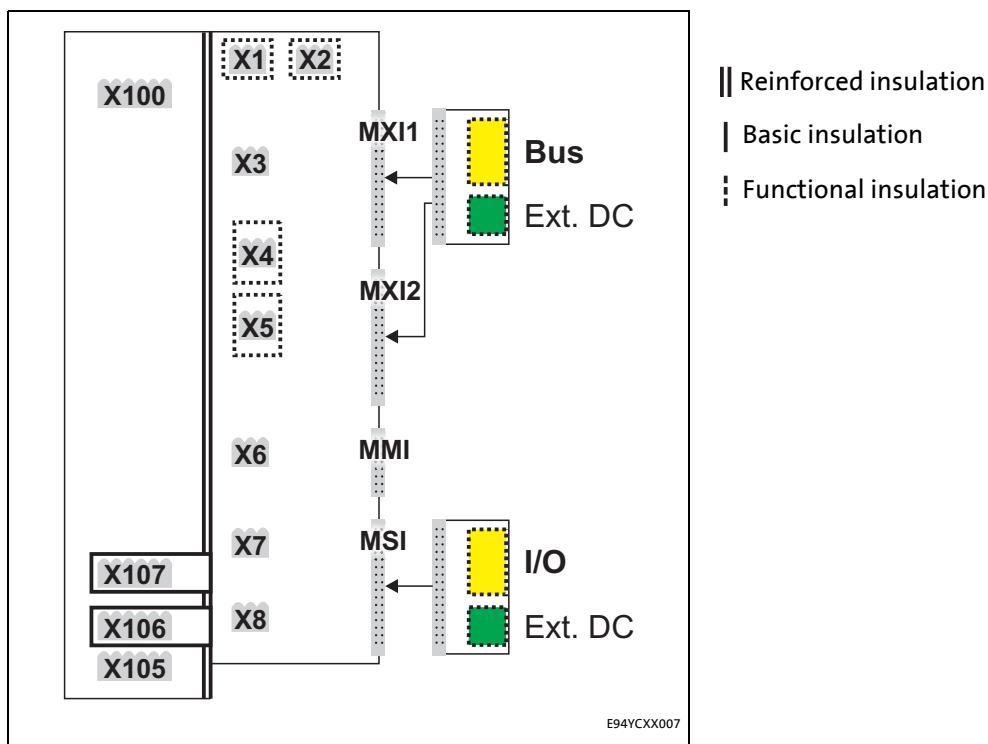
- a double isolating distance must exist.
- the components to be connected must be provided with the second isolating distance.

**Note!**

The protective insulation provided in Servo Drives 9400 is implemented in accordance with EN 61800-5-1.

The following illustration ...

- shows the arrangement of the terminal strips and the separate potential areas of the inverter.
- serves to determine the decisive protective insulation between two terminals located in differently insulated separate potential areas.



[4-1] Protective insulation in accordance with EN61800-5-1

Terminal strip	Terminal	Terminal strip	Terminal
X100	L1, L2, L3 (Single Drive only)	X1	CAN on board 9400
	+UG, -UG	X2	State bus
X105	U, V, W		24 V (ext.)
	Rb1, Rb2 (Single Drive only)	X3	Analog inputs/outputs
X106	Motor PTC	X4	Digital outputs
X107	Control of the motor holding brake	X5	Digital inputs
		X6	Diagnostics
		X7	Resolver
		X8	Encoder
		MXI1, MXI2	Extension module
		MMI	Memory module
		MSI	Safety module

Example

Which type of protective insulation is used between the bus terminal of the device module in slot MXI1 or MXI2 and the mains terminal X100?

The separate potential area with the better protective insulation is decisive.

- The separate potential area of the bus terminal of the device module has a "functional insulation".
- The separate potential area of the mains terminal has a "reinforced insulation".

Result: The insulation between mains terminal X100 and the bus terminal is of the "reinforced insulation" type.

4.3 Protocol data

Range	Values
Process data	1 ... 32 process data words for each direction (max. 64 bytes, 16 bits / word)
Parameter data (mailbox size for CoE transfer)	Max. 128 bytes
FSOE data if the SM302 safety module is used (from SW version 04.00)	Safety outputs: 11 bytes
	Safety inputs: 19 bytes

4.4 Communication time

Parameter data (SDO)

The communication time for parameter data is the time between the transmission of an SDO request and the arrival of the corresponding response.

The processing time in the inverter is approx. 30 ms + 20 ms tolerance (typical)

For some codes, the processing time may be longer (see reference manual/»Engineer« online help for the Servo Drive 9400).

Process data (PDO)

The communication time for process data is the time between the reception of a PDO with setpoints and the return of a PDO with the current actual values.

The communication times for process data depend on ...

- processing time in the inverter (interval time of the application task, process data mode);
- Runtime on the fieldbus (frame length, number of devices, PDO update time, instant of transmission of the EtherCAT telegram);
- Use of the "Distributed clocks" (DC) functionality for synchronised operation.
 - ▶ [Synchronisation with "Distributed clocks" \(DC\)](#) (36)

The basic conditions mentioned above cause the following processing times for the process data in the inverter:

- DC use (synchronised operation):

The processing time starts with the acceptance of the setpoints by the inverter with the Sync0 event (DC synchronisation cycle) and ends with the provision of the current actual values in the EtherCAT interface.

Depending on the selected [Process data mode](#) (40) ([C13892](#) / [C14892](#)), the resulting processing times are as follows:

- Deterministic mode: 2.1 ms + interval time of the application task
- Optimised mode: 1.3 ms + interval time of the application task

- No DC use (non-synchronised operation):

The processing time starts with the acceptance of the setpoints by the inverter at a time that is not synchronised with the EtherCAT master and ends with the provision of the current actual values in the EtherCAT interface.

Hence, the following holds true for the processing time:

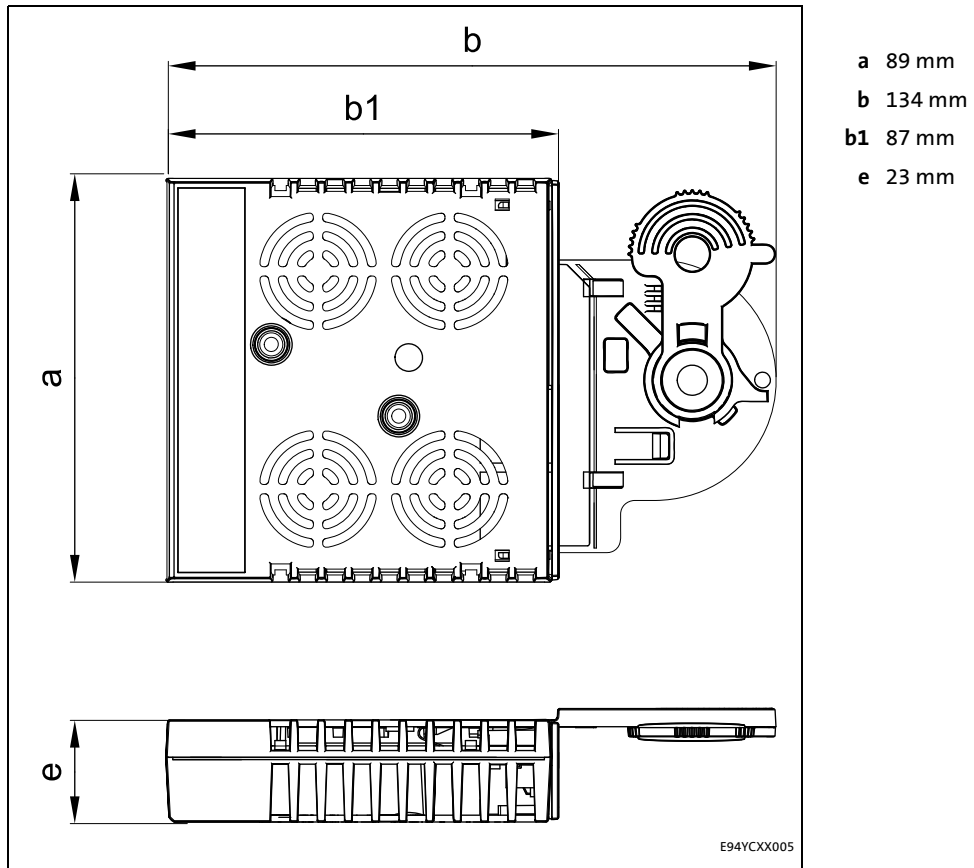
1.3 ms + 1.0 ms (tolerance) + interval time of the application task



Documentation of the 9400 function library

Here you can find detailed information on how to set the interval time of the application task.

4.5 Dimensions



[4-2] Dimensions

5 Installation



Stop!

Electrostatic discharge

Electronic components within the communication module can be damaged or destroyed by electrostatic discharge.

Possible consequences:

- The communication module is defective.
- Fieldbus communication is not possible or faulty.

Protective measures:

Before touching the module, be sure that you are free of electrostatic charge.

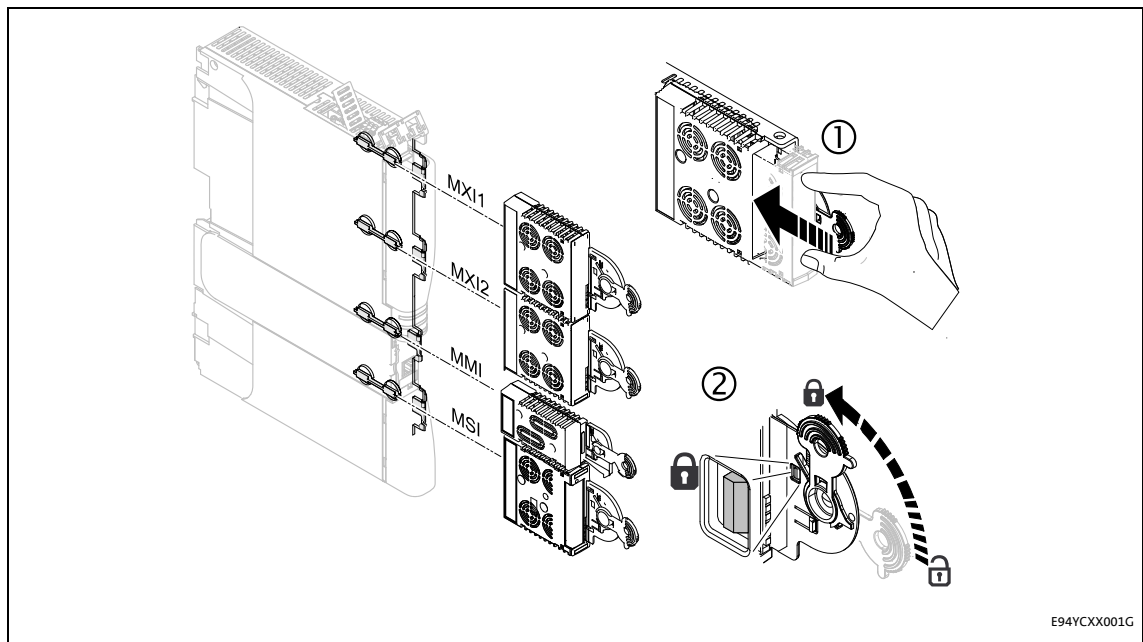
5.1 Mechanical installation

5.1.1 Mounting



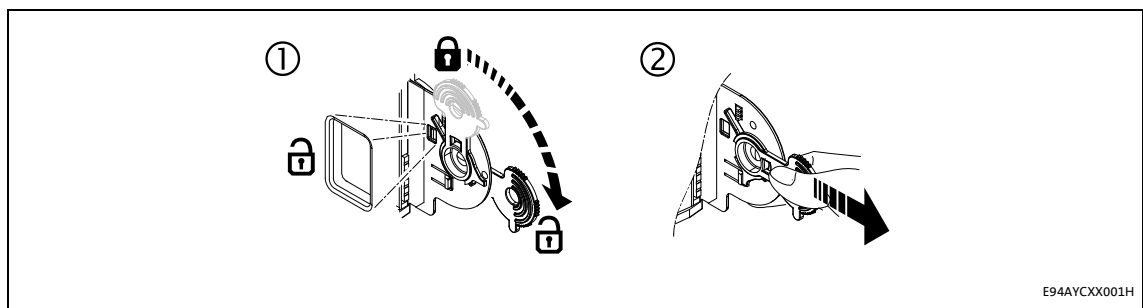
Note!

A safety bus system (FSOE) can only be operated via the upper module slot (MXI1) of the Servo Drive 9400.



[5-1] Mounting

5.1.2 Disassembly



[5-2] Disassembly

5.2

Electrical installation


Documentation for the standard device, control system, system/machine

Observe the notes and wiring instructions contained in this documentation.

5.2.1

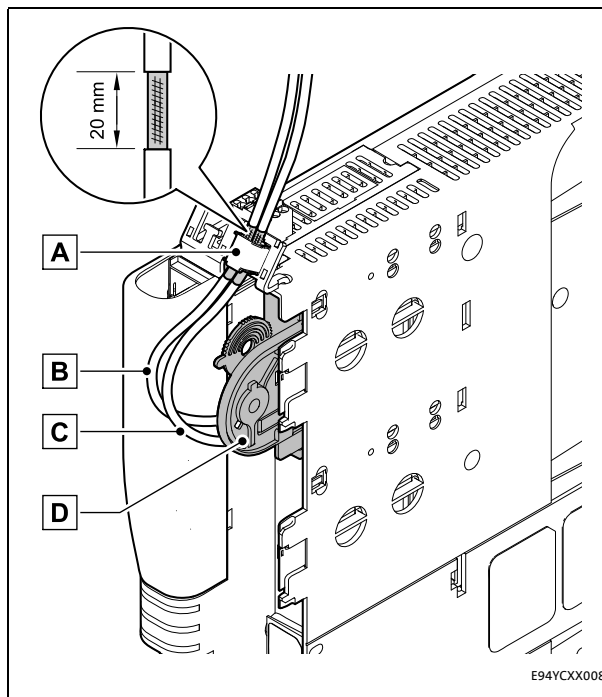
Wiring according to EMC guidelines

In typical systems, standard shielding is sufficient for Ethernet cables.

However, in environments with a very high level of interference, EMC resistance can be improved by additionally earthing the cable shield on both sides.

For this observe the following notes:

1. The distance between the additional earthing and the Ethernet plug depends on the module slot and is as follows:
 - approx. 10 cm for the upper slot (MXI1)
 - approx. 20 cm for the lower slot (MXI2)
2. Measure the appropriate distance along the cable and, starting from this point, remove 2 cm of the cable's plastic sheath.
3. Connect the cable shield to the shield sheet of the Servo Drive 9400.



A Connection to the shield sheet of the Servo Drive 9400

B Outgoing EtherCAT line at X247 (OUT)

C Incoming EtherCAT line at X246 (IN)

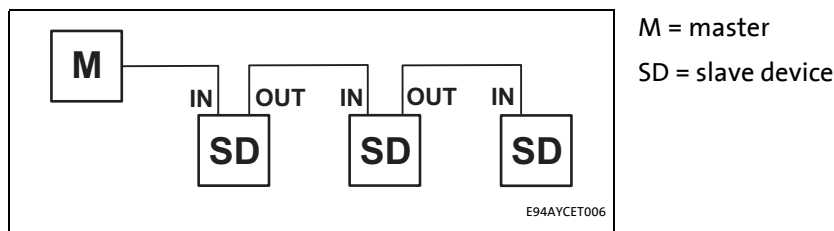
D Communication module in slot MXI1 of the Servo Drive 9400

[5-3] Wiring according to EMC guidelines

5.2.2 Network topology

An EtherCAT frame is sent through a pair of wires from the master to the slaves. The frame is forwarded from slave to slave until it has passed through all the devices. Finally, the last slave returns the frame to the master through a second pair of wires. In this way, EtherCAT always forms a logic ring topology, irrespective of the topology used.

Line topology



[5-4] Line topology

The devices are interconnected successively.

In order to ensure trouble-free operation, it is required to assign and wire the EtherCAT inputs (IN) and EtherCAT outputs (OUT) correctly.

The receiving line is plugged into socket X246 (IN), the forwarding line into socket X247 (OUT).

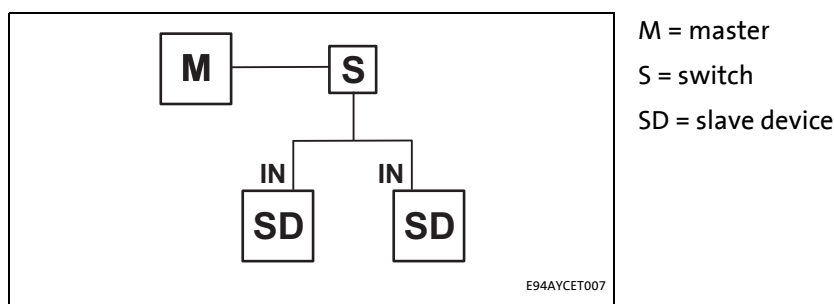
The direction of data transmission is from the master to the slaves.



Tip!

The termination of the last EtherCAT device is effected automatically by the slave.

Switch topology



[5-5] Switch topology

The wiring can also be carried out in a star structure via an appropriate switch. For this, observe the additional runtimes.

5.2.3 EtherCAT connection

The EtherCAT connection is established via the RJ45 sockets **X246 (IN)** and **X247 (OUT)**.

A standard Ethernet patch cable is suitable for connecting the communication module to the EtherCAT fieldbus.

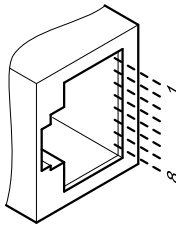
► [Ethernet cable specification](#) (📖 27)



Note!

To prevent the RJ45 socket from being damaged, insert or remove the Ethernet cable connector straight (at a right angle) into or from the socket.

Pin assignment of the RJ45 sockets

RJ45 socket	Pin	Signal
 E94AYCXX004C	1	Tx +
	2	Tx -
	3	Rx +
	4	-
	5	-
	6	Rx -
	7	-
	8	-

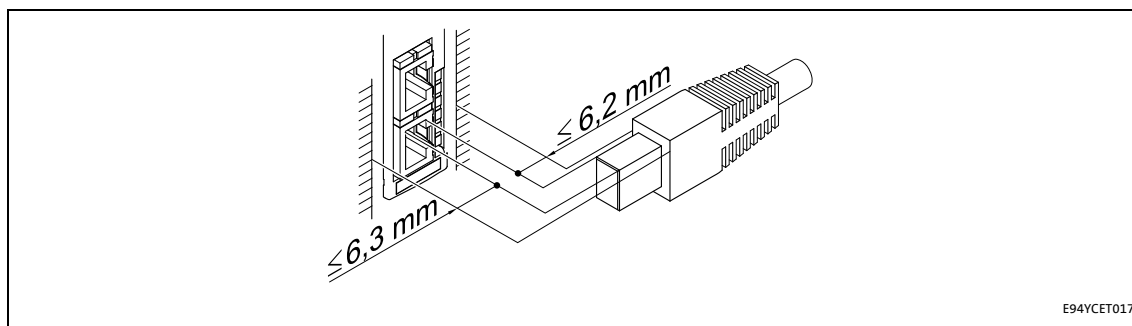


Tip!

The EtherCAT interfaces are provided with an auto MDIX function. This function adjusts the polarity of the RJ45 interfaces so that a connection can be established irrespective of the polarity of the opposite EtherCAT interface and irrespective of the type of cable used (standard patch cable or crossover cable).

Mounting clearance

When ordering and using your Ethernet cable, note the amount of free space available.



[5-6] Mounting clearance

5.2.4 Ethernet cable specification

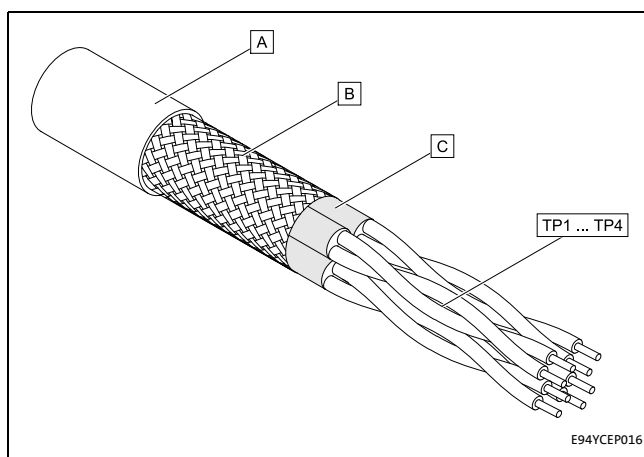


Note!

Only use cables that meet the specifications listed.

Ethernet cable specification	
Ethernet standard	Standard Ethernet (in accordance with IEEE 802.3), 100Base-TX (Fast Ethernet)
Cable type	S/FTP (Screened Foiled Twisted Pair, ISO/IEC 11801 or EN 50173), CAT 5e
Damping	23.2 dB (for 100 MHz and 100 m each)
Crosstalk damping	24 dB (at 100 MHz and per 100 m)
Return loss	10 dB (per 100 m)
Surge impedance	100 Ω

Structure of the Ethernet cable



A Cable insulation

B Braid

C Foil shielding

TP1 Twisted core pairs 1 ... 4

... [▶ Colour code of the Ethernet cable](#)

TP4 [\(28\)](#)

[5-7] Structure of the Ethernet cable (S/FTP, CAT 5e)

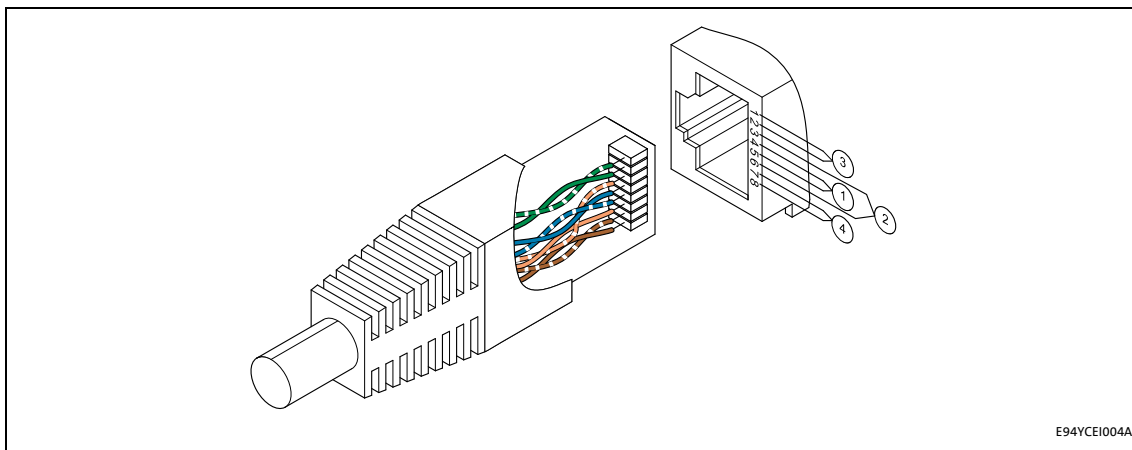
Colour code of the Ethernet cable



Note!

Wiring and colour code are standardised in EIA/TIA 568A/568B.

In accordance with the industrial standard, the use of 4-pin Ethernet cables is permissible. The cable type only connects the assigned pins 1, 2, 3 and 6 to one another.



E94YCEI004A

[5-8] Ethernet plug in accordance with EIA/TIA 568A/568B

Pair	Pin	Signal	EIA/TIA 568A	EIA/TIA 568B
3	1	Tx +	White / Green	White / Orange
	2	Tx -	Green	orange
2	3	Rx +	White / Orange	White / Green
1	4		blue	blue
	5		White / Blue	Blue / White
2	6	Rx -	orange	Green
4	7		White / Brown	White / Brown
	8		brown	brown

5.2.5 External voltage supply

The communication module can be supplied externally with voltage via separate supply cables at the 2-pole plug connector **X245**.



Note!

With external voltage supply, always use a separate power supply unit, safely separated to EN 61800-5-1 in every control cabinet (SELV/PELV).

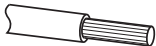
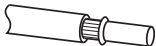
External voltage supply of the communication module is required if the communication via the bus should be maintained when the supply of the standard device fails.

Access to parameters of a standard device disconnected from the mains is not possible.

Assignment of the X245 plug connector

Name	Description
+	U = 24 V DC (20.4 V -0 % ... 28.8 V +0 %) I = 130 mA
-	Reference potential for the external voltage supply

Terminal data

Range	Values
Electrical connection	2-pole plug connector (spring connection/screw connection)
Possible connections	Rigid:
	 1.5 mm ² (AWG 16)
	Flexible:
	 Without wire end ferrule 1.5 mm ² (AWG 16)
	 With wire end ferrule, without plastic sleeve 1.5 mm ² (AWG 16)
	 With wire end ferrule, with plastic sleeve 0.5 mm ² (AWG20)
Starting torque	0.5 ... 0.6 Nm / 4.4 ... 5.3 lb-in (only with screw connection)
Stripping length	6 mm with screw connection
	9 mm with spring connection

6

Commissioning

During commissioning, plant-specific data such as motor parameters, operating parameters, responses, and parameters for fieldbus communication are defined for the inverter. Lenze devices use codes for this purpose.

The codes of the inverter and the communication module are saved to the memory module in a non-volatile data set.

In addition, there are codes for diagnosing and monitoring the stations.

**Note!**

When parameterising the communication module, please note that the code number depends on the slot of the Servo Drive 9400 into which the communication module is plugged.

The first two digits of the code number indicate the slot:

- C13nnn for slot MXI1
 - ▶ [Parameters of the communication module for slot MXI1](#) (📖 100)
- C14nnn for slot MXI2
 - ▶ [Parameters of the communication module for slot MXI2](#) (📖 118)

You also have to set the [Parameters of the standard device that are relevant to communication](#) (📖 97).

6.1

Before initial switch-on

**Stop!**

Before switching on the Servo Drive 9400 and the communication module for the first time, check the entire wiring for completeness, short circuit and earth fault.

6.2 Configuring the Controller (EtherCAT master)

The Controller (EtherCAT master) must be configured before communication with the communication module is possible.

In order to configure EtherCAT networks, you always need a configuration software for the Controller, e.g.:

- Lenze »PLC Designer«
- Beckhoff »TwinCAT«

These are software systems for the programming of control programs, EtherCAT configuration, real-time execution, and diagnostics.

The parameters of the communication module are stored in the internal configuration memory and can be used by the master for the device identification.

For device detection (fieldbus scan), the corresponding device descriptions of the Lenze device family are used.



"Controller-based Automation EtherCAT" communication manual

Here you'll find some detailed information relating to the EtherCAT configuration with the Lenze »PLC Designer«.

6.2.1 Installing device description files

The current XML device description files required for configuring the EtherCAT device can be found in the download area at:

www.lenze.com

The following device description file can be installed via the EtherCAT configuration software.

Device description file	Used for ...
Lenze_E94AYCET_V100_yymmdd.xml	SW version 01.01 of the E94AYCET communication module
Lenze_E94AYCET_IO_yyyymmdd.xml	From SW version 02.00 of the E94AYCET communication module in connection with Servo Drives 9400
Lenze_E94AYCET_MOTION_yyyymmdd.xml	From SW version 02.00 of the E94AYCET communication module in connection with the technology application "CiA402" and Servo Drives 9400

Wildcards in the file name	
yyyy	Year
mm	Month
dd	Day

6.2.2 Automatic device identification

For a faultless integration of the EtherCAT slaves into a master configuration it is necessary to select the correct Lenze device in the EtherCAT configuration software.

Each EtherCAT device is identified unambiguously by the configuration software by means of the product code (equal to the CoE object I-1018.2), the manufacturer's identification mark (0x3B), and the main software version of the communication module.

► [Identification](#) (□ 12)

► [Communication objects \(object directory\)](#) (□ 78)

In order that the configuration software selects the configuration specific for the EtherCAT device from the device description file, the product code is automatically set in the identity object according to the technology application selected in the inverter (code C00218) and is updated after the voltage supply is switched off/on or after every application download.

During initialisation, the product code is transferred to the EtherCAT master. On the basis of this identification, the master can accept the corresponding settings from the device description.

Product codes for Servo Drives 9400

The product code defines the following devices in the device description files, depending on the technology application selected in the Servo Drive 9400:

Product code [dec]									Meaning
9	4	0	0	2	3	x	x	x	Servo Drive 9400 HighLine
									Applications:
									0 0 0 Empty application (appl. ID 100000000)
									0 0 1 Actuating drive speed (appl. ID 100102101)
									0 0 2 Actuating drive torque (appl. ID 100202101)
									0 0 3 Electronic gearbox (appl. ID 100302202 / 100302102)
									0 0 4 Synchronism with mark synchronisation (appl. ID 100402202 / 100402102)
									0 0 5 Table positioning (appl. ID 100502101)
									0 0 6 Positioning sequence control (appl. ID 100602103)
									1 0 1 CiA402 (appl. ID 110102102)
9	4	0	0	2	5	0	0	7	Servo Drive 9400 PLC with PLC application (appl. ID 100700204)
9	4	0	0	2	6	2	0	1	Servo Drive 9400 V/R (regenerative power supply module) with V/R application (appl. ID 120100101)

6.2.3 Configuring process data

Servo Drives 9400 support the configuration of max. 32 process data words (max. 64 bytes) per direction.

The process data configuration is determined during the initialisation phase of the master (PDO mapping).

The process data configuration is specifically predefined in the device description file for each application.

The process data length can be adapted by the user, if required.

6.2.4 Configuring safety process data



Note!

A safety bus system (FSoE) can only be operated via the upper module slot (MXI1) of the Servo Drive 9400.

From [SW version 04.00](#) and with the use of an **SM302** safety module (E94AYAF), safety process data can be configured optionally in addition to the process data. The structure of the safety process data is predefined in the device description files and cannot be changed.

The safety process data are transmitted with the same cycle time as the process data (PDOs).

▶ [Safety over EtherCAT](#) (📖 81)

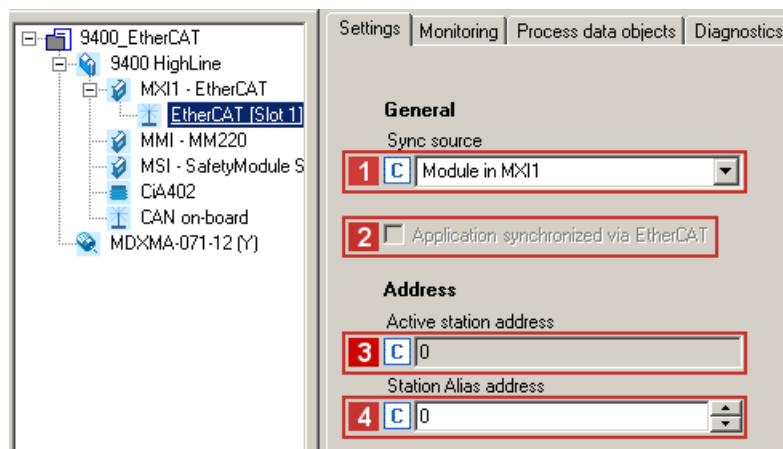
▶ [FSoE mapping indices](#) (📖 66)

6.2.5 Determining the cycle time

The process data objects (PDO) and the optional safety process data are transmitted cyclically between the EtherCAT master and the slaves (inverters).

The cycle time is set by means of the EtherCAT configuration software.

6.3 Settings in »Engineer«

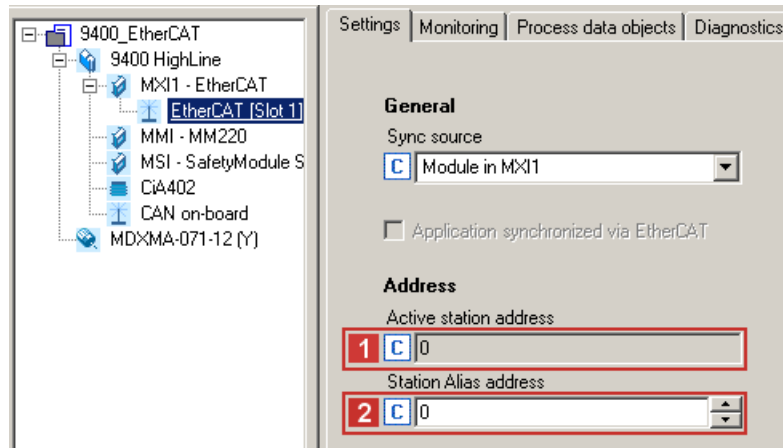


Settings / display		Code
1	Selection of the sync source for DC synchronisation See also DC configuration in the Servo Drive 9400 (slave) (□ 38).	C01120
2	Display of the status of synchronism of the Servo Drive 9400 Only if Synchronisation with "Distributed clocks" (DC) (□ 36) is used.	C13884 / C14884
3	Display of the active station address See also Address allocation (□ 35).	C13864 / C14864
4	Entry of the station alias address See also Address allocation (□ 35).	C13899 / C14899

6.4 Address allocation

The EtherCAT slaves are normally addressed via a fixed 16-bit address defined by the EtherCAT master. During start-up, the master assigns this address to each slave, depending on the physical order in the EtherCAT network. The address is not saved and is lost when the device is switched off.

In the »Engineer«, the address assigned is shown under the **Settings** tab in the **1 Active station address** display field ([C13864](#) / [C14864](#)).



Defining Explicit Device Identification

There are two ways to assign the slave to an identifier (station alias):

- A. Via the EtherCAT master with the assignment of the "Configured Station Alias" (from [SW version 04.00](#) of the communication module).
- B. Via the **2 Station alias address** input field.



Note!

- The station alias must only be set if the EtherCAT slave is part of a "hot connect" group or integrated in the modular machine configuration of Lenze.
- The station alias must be unambiguous and must only be assigned once within the EtherCAT network.
- Use the same station alias in the EtherCAT master and in the slave.

Valid address range: 0 ... 32767

- Address '0' means that no station alias is assigned.
- If the value is > '0', the address cannot be assigned by the EtherCAT master.
- Impermissible addresses are marked in red in the input field.
- The address is written to code [C13899](#) / [C14899](#).

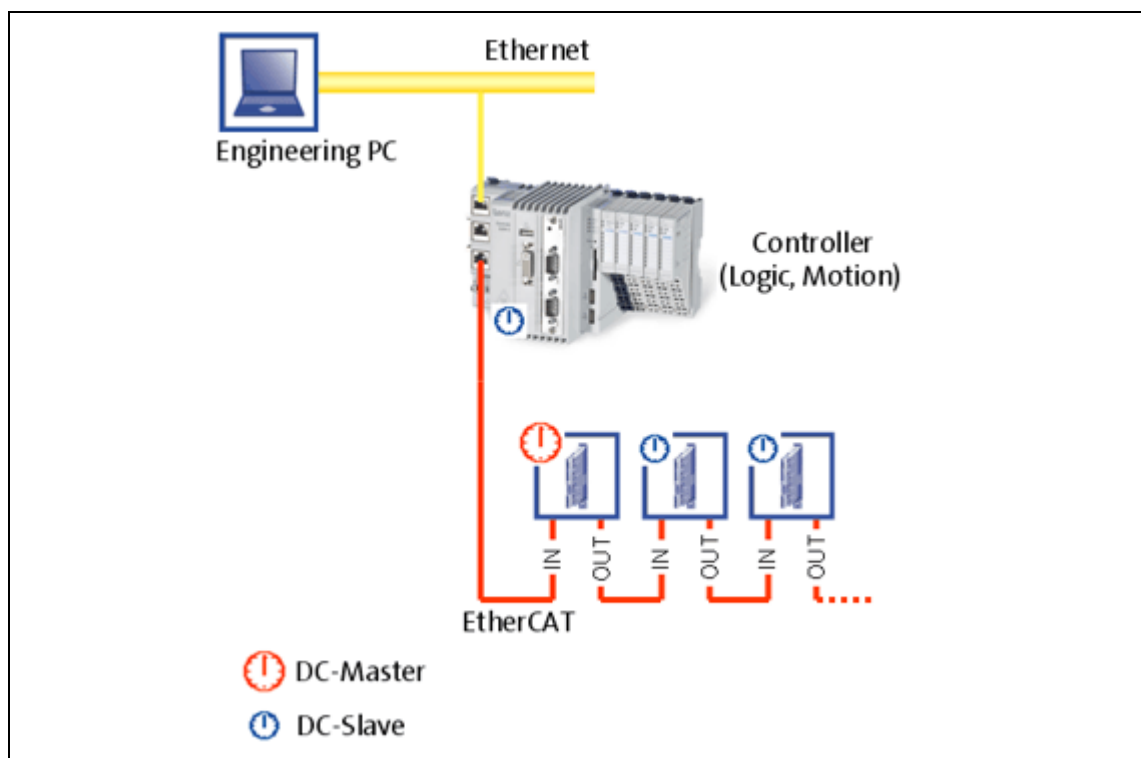
6.5 Synchronisation with "Distributed clocks" (DC)

The "Distributed clocks" (DC) functionality enables exact time synchronisation for applications in which several axes perform a coordinated movement simultaneously. Data are incorporated synchronously with the PLC program. During DC synchronisation, all slaves are synchronised with a reference clock, the so-called "DC master".



Note!

- DC synchronisation is absolutely required for Motion applications.
- DC synchronisation can also be used for Logic applications.
- Not all slaves support the DC functionality.
- On order to be able to use the DC functionality, the first slave connected to the EtherCAT master (e.g. Lenze Controller) must have **DC master capability**. When further slaves are connected, DC-capable and non-DC-capable devices can be mixed.
- The first EtherCAT slave after the Lenze Controller must be the **DC master** that supplies the other EtherCAT devices (incl. the controller) with the exact time.



[6-1] Example: "Distributed clocks" in the EtherCAT bus system with Lenze Controller 3231 C

The DC synchronisation is set with the EtherCAT configuration software (e.g. »PLC Designer«, »TwinCAT«).



"Controller-based Automation EtherCAT" communication manual

Here you'll find some detailed information relating to the EtherCAT configuration with the Lenze »PLC Designer«.

6.5.1 DC configuration in the master

By default, the use of the DC synchronisation is deactivated in the device description ([31](#)).

Parameterise the DC synchronisation in the EtherCAT configuration software (e.g. »PLC Designer«, »TwinCAT«).

Set the synchronisation cycle time in the master. It is mainly defined by the processing time of the master and the slaves.



Note!

The synchronisation cycle time ...

- must be an integer multiple of 1 ms;
- may be maximally 15 ms.

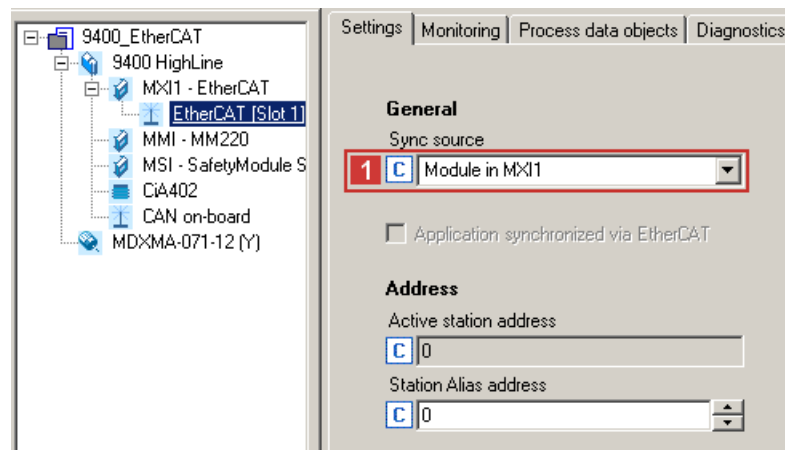
6.5.2 DC configuration in the Servo Drive 9400 (slave)

**Note!**

The settings of the parameters sync cycle time (C01121), sync phase position (C01122), sync tolerance (C01123) and sync PLL increment (C01124) common for the Lenze system bus (CAN) cannot be made for EtherCAT. These values are automatically calculated by the EtherCAT communication module and set internally in the inverter.

In order to be able to use the DC synchronisation in the Servo Drive 9400, go to the **Settings** tab in the **1 Sync source** selection field and select the sync source ([C01120](#)):

- Selection 4: Module in slot MXI1 of the Servo Drive 9400
- Selection 5: Module in slot MXI2 of the Servo Drive 9400



6.5.3 Behaviour of the communication module during the start

Code [C13883](#) / [C14883](#) shows whether the DC synchronisation has been activated for the communication module.

If the DC synchronisation is used, the communication module only changes to the "Operational" state when the standard device has adapted its phase position to the DC signal. This process may take several seconds.



Note!

- If the communication module does not change to the "Operational" state, there might be an error in the configuration or in the EtherCAT wiring.
- The communication module compares the cycle time defined by the EtherCAT master to the internal processing time (1 ms) of the standard device. The synchronisation cycle time in the master must be identical with or an integer multiple of 1 ms.
- Furthermore it is checked whether the sync source selection in standard device code [C01120](#) is correct.
- Further information can be found in the status information or emergency messages of the master.

The cycle time (in μs) set by the master on the fieldbus is displayed with code [C13870](#) / [C14870](#).

The state of the standard device synchronicity is displayed in code [C13884](#) / [C14884](#).

6.5.4 Process data mode

Code [C13892](#) / [C14892](#) serves to influence the internal PDO processing time of the actual values.

Deterministic mode ([C13892](#) / [C14892](#) = 1, Lenze setting)

In the deterministic mode, the actual values are copied to the EtherCAT interface with the next Sync0 event.

Optimised mode ([C13892](#) / [C14892](#) = 0)

In the optimised mode, the actual values are not copied to the EtherCAT interface with the next Sync0 event but already 320 µs after the last Sync0 event. Thus, the actual values are provided one cycle earlier in the EtherCAT interface.

Conditions for the optimised mode:

- The EtherCAT telegram generated by the master has a low jitter and is sent within a window of 320 µs after the Sync0 event and shortly before the next Sync0 event.
- The EtherCAT cycle time must be 1 ms.



Note!

- The process data mode is only evaluated with DC synchronisation.
- The setpoints are always processed with the Sync0 event.
- When Lenze EtherCAT controllers are used, we recommend setting the deterministic mode ([C13892](#) / [C14892](#) = 1).

6.6 Establishing an online connection with the »Engineer«

With the »Engineer« you can establish an online connection to the individual field devices.

When an online connection has been established, you can for instance carry out parameter settings directly in the field device or diagnose the field device.



Stop!

If parameters in the »Engineer« are changed while the Engineer is connected online to the field device, the changes are directly accepted to the device!



Note!

To go online, the EtherCAT bus at least has to be in the "Pre-Operational" state.

The functions for establishing/cancelling an online connection in the »Engineer« can be executed via the **Online** menu:

Menu command	Shortcut
Online → Go online	<F4>
Online → Set communication path and go online	
Configuring a bus connection: • Gateway Controller -> configure EtherCAT (□ 42) • Gateway Controller -> configure EtherCAT ADS (Beckhoff) (□ 44)	
Online → Go offline	<Shift>+<F4>

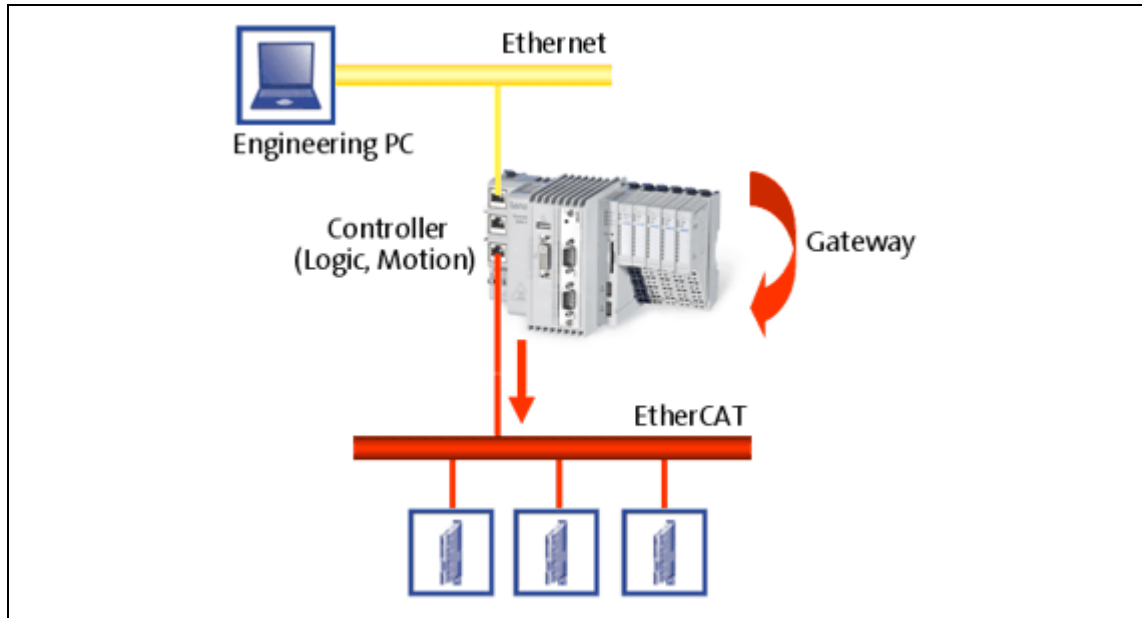


Documentation for the Lenze »Engineer«

Here you'll find further detailed information about how to establish an online connection.

6.6.1 Gateway Controller -> configure EtherCAT

The Lenze Controller provides a gateway function to establish an online connection to a field device via EtherCAT.



[6-2] Example: EtherCAT bus system with a Lenze Controller 3231 C as gateway



How to configure an online connection to a field device which is connected to the Lenze Controller via EtherCAT:

1. Go to the "Communication path" dialog box and select the entry "Gateway controller -> EtherCAT" from the "Bus connection" list field.
2. Click **Search/Enter....**

The "Gateway Controller -> Set up EtherCAT bus" dialog box is shown:

IPC name	IP address	IPC interface
EL 870	172.31.207.20	ECAT1

IPC (Gateway)

Name:

IP address:

Interface:

Field device

Name:

Ethercat address:

Diagram: Gateway device (Lenze controller) connected to Field device (9400 HighLine Master) via Ethernet and EtherCAT.

OK Cancel

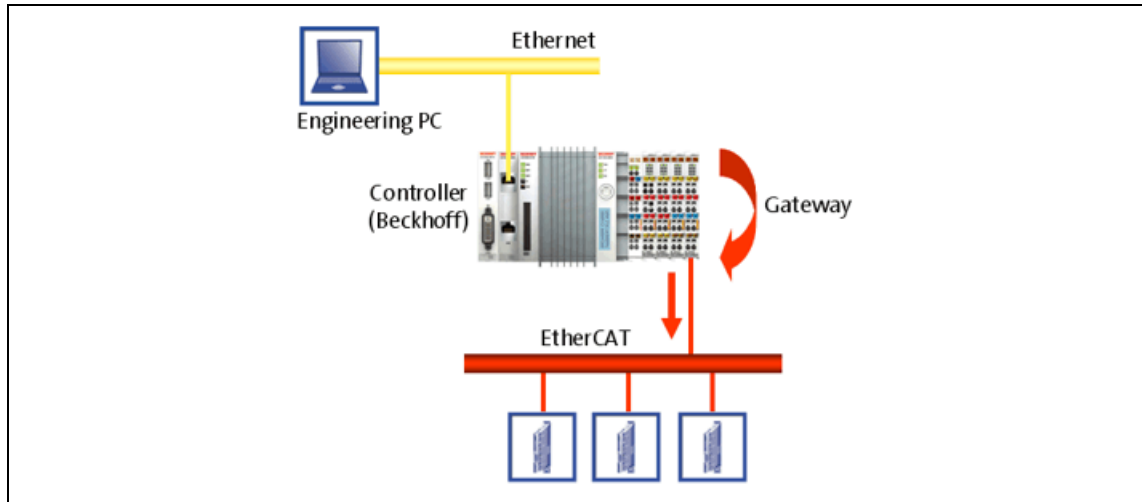
3. Enter the **IP address** of the controller.

By clicking the **Ping** button, you can carry out a simple test which verifies whether a device can actually be reached via the IP address set.

4. Click **OK**.
 - The "Enter IP address" dialog box is closed.
 - In the **Device access path** column of the "Communication path" dialog box, the corresponding device access path is shown (e.g. "IPC:172_31_207_254.ECAT.ecat1.dev1001.").

6.6.2 Gateway Controller -> configure EtherCAT ADS (Beckhoff)

The **Gateway EtherCAT ADS** bus connection makes it possible to establish an online connection to a Lenze inverter that is connected to a Beckhoff Controller via EtherCAT (gateway function).



[6-3] Example: EtherCAT bus system with a Beckhoff Controller as Gateway



How to configure an online connection to a field device which is connected to a Beckhoff controller via EtherCAT:

1. Highlight the project root  in the project.
Alternatively: Create a new project or carry out a fieldbus scan.
2. Execute the menu command **Insert → Insert device detected online**.
3. Select **Gateway Controller -> EtherCAT ADS** as bus connection.
4. Configure access data:
 - Configure the access data applicable to the controller via the **Insert address** button.
 - The **Search** button initiates the Controller to display the fieldbus devices connected to the EtherCAT segment.



How to use the EtherCAT ADS communication path:

1. Highlight the desired inverter, to which a gateway connection via EtherCAT ADS is to be established, in the project tree.
2. Call the menu command **Online → Set communication path and go online**.

3. Select **1 Gateway Controller -> EtherCAT ADS** as bus connection.
4. Enter the access data applicable to the controller in area **2**.
Enter the user name, password, and the IP address and the AMS Net ID of the EtherCAT interface of the Controller.
5. In area **3**, specify the EtherCAT address of the field device to which the online connection is to be established.

Alternatively you can click the **Search/Enter** button which calls the "Select Device Access Path" dialog box. By this, the »Engineer« initiates the controller to display the devices detected on the EtherCAT segment.



Note!

If several PCs access a target device (slave) simultaneously under a Beckhoff control, superpositions cause sporadic transmission errors in the communication protocol.



Tip!

If you have installed »TwinCAT« V2 or V3 but no connection to the Beckhoff control has been prepared, use the »TwinCAT« system manager or the version-specific »TwinCAT« route tool (accessible via the »TwinCAT« tray icon) for creating the required ADS remote connection.

As "AMSNetId" select the "NetId" of the EtherCAT device (usually "a.b.c.d.2.1").

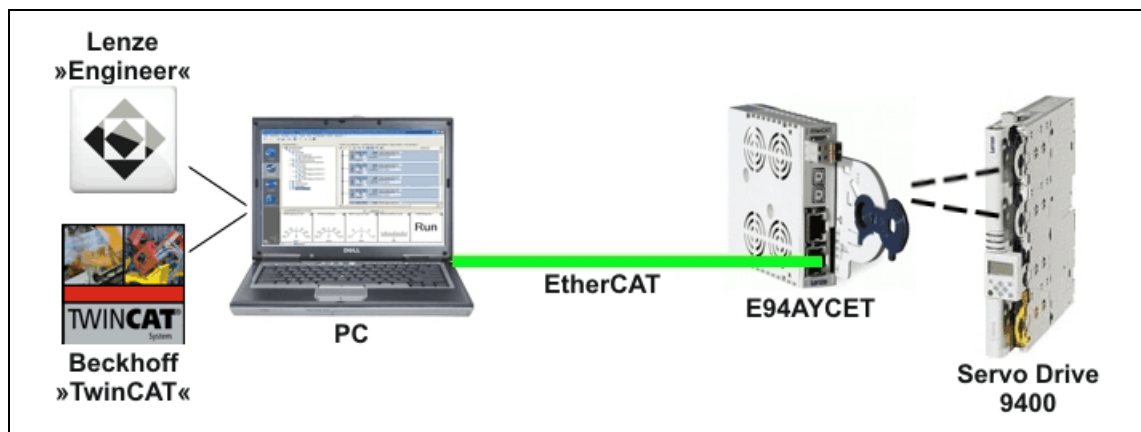
6.7

EtherCAT ADS communication parameters in »TwinCAT« and »Engineer«

In the following, two example structures are used to describe where you can find the EtherCAT ADS communication parameters in the Beckhoff »TwinCAT« and in the Lenze »Engineer« EtherCAT.

6.7.1

Example: Structure without a Beckhoff Controller

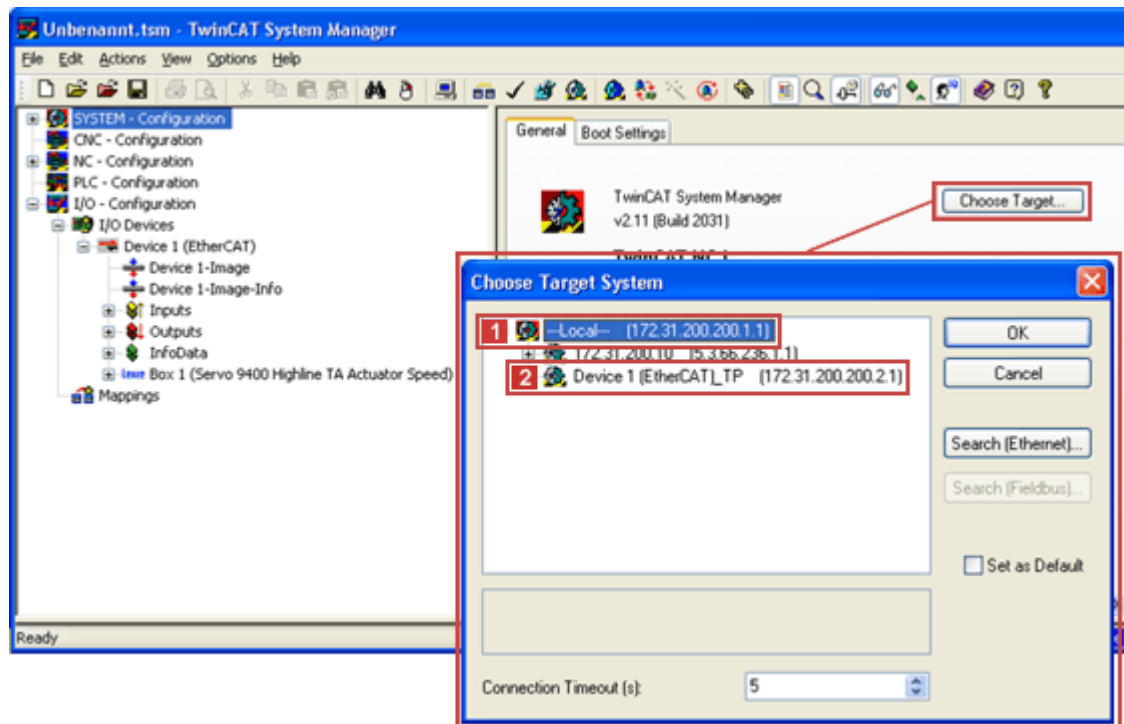


[6-4] Example: EtherCAT bus system without Beckhoff Controller

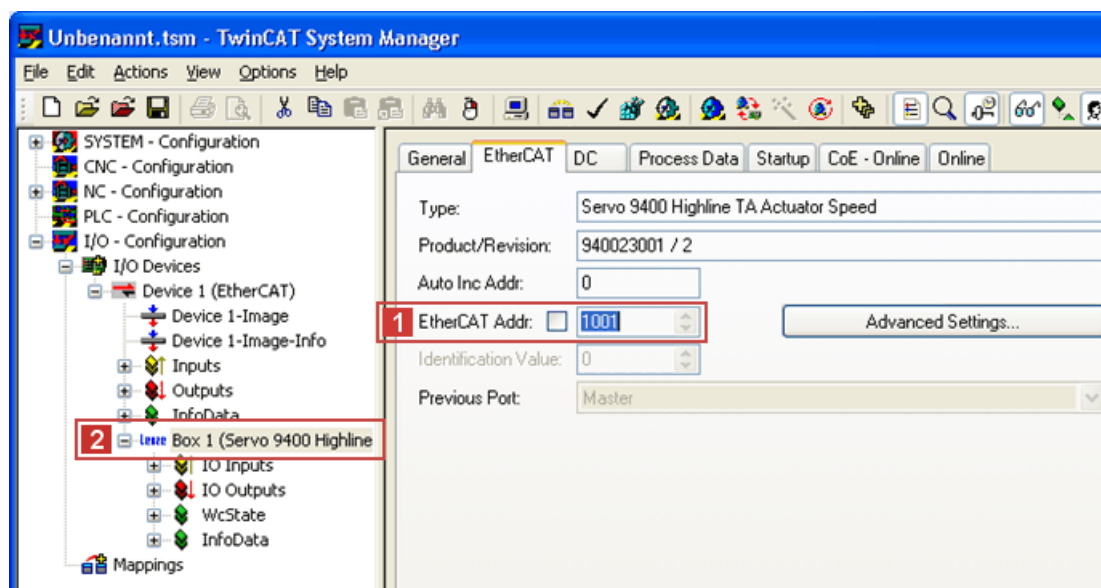
The Beckhoff Soft-PLC runs on the Microsoft Windows XP PC on which the Beckhoff »TwinCAT« and the Lenze »Engineer« are installed as well.

Display of the communication parameters in »TwinCAT«

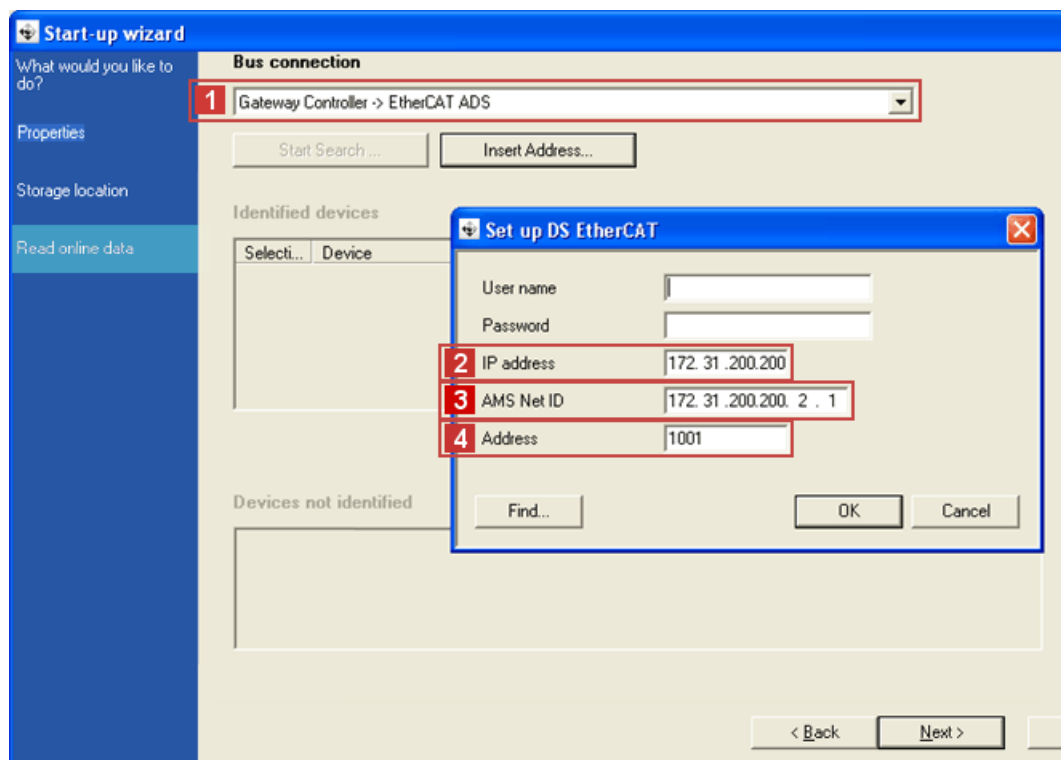
The communication parameters **1** IP address (here '172.31.200.200') and **2** EtherCAT Master Net ID (here '172.31.200.200.2.1') can be found under the target system selection:



The EtherCAT **1** Slave address (here '1001') can be found under the EtherCAT tab of the **2** EtherCAT slave:



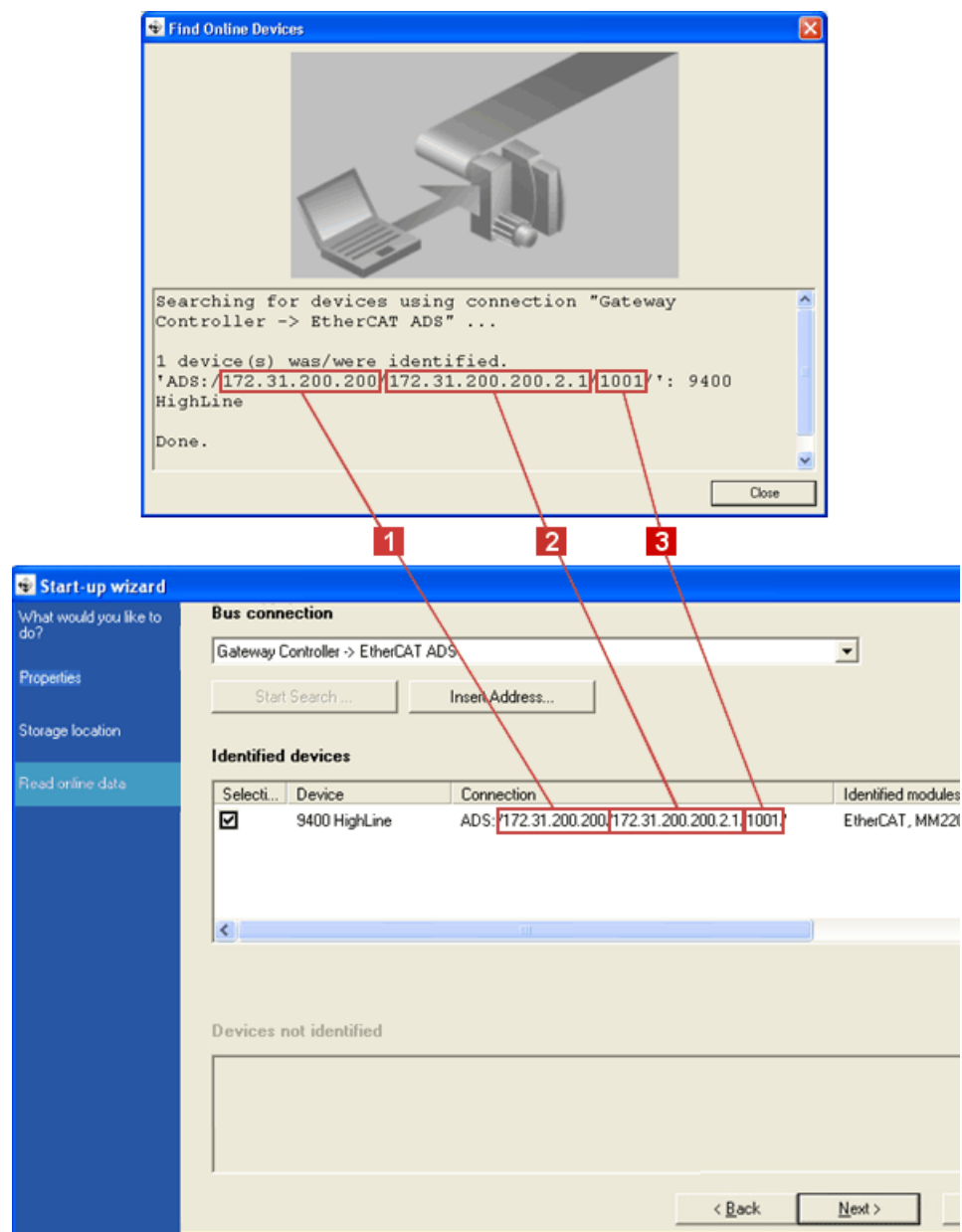
Online device identification in the »Engineer« start-up wizard



In the »Engineer« start-up wizard under the **1 Gateway Controller -> EtherCAT ADS** bus connection, a field device was detected online:

- 2 IP address:** 172.31.200.200
- 3 EtherCAT Net ID:** 172.31.200.200.2.1
- 4 EtherCAT slave address:** 1001

Display of the identification of the field device detected in the »Engineer« start-up wizard:



1 IP address: 172.31.200.200

2 EtherCAT Net ID: 172.31.200.200.2.1

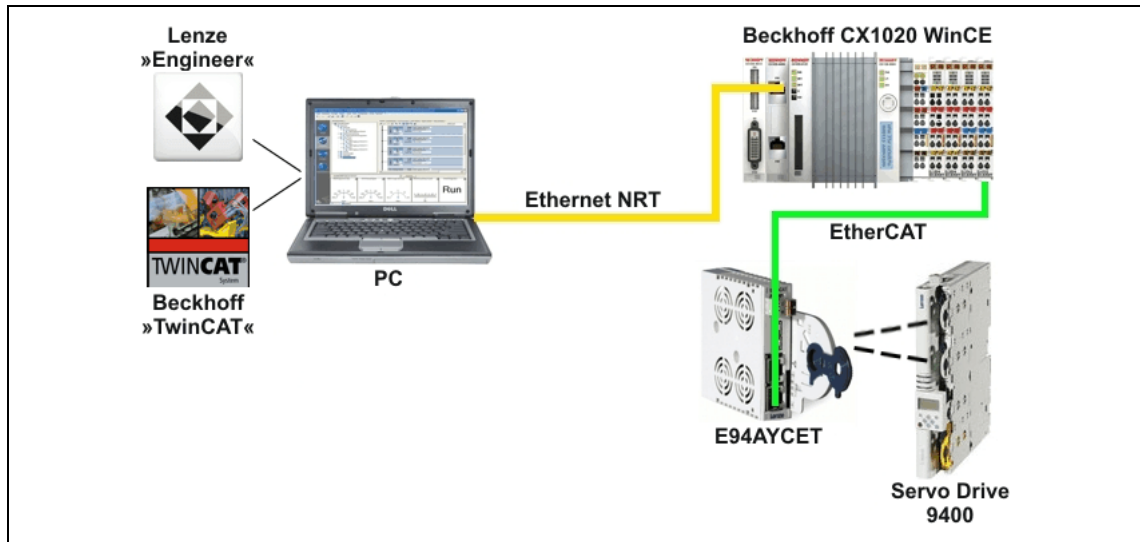
3 EtherCAT slave address: 1001

6.7.2

Example: Structure with a Beckhoff DIN rail IPC CX1020

**Note!**

If several PCs access a target device under a Beckhoff control, superpositions cause sporadic transmission errors in the communication protocol.



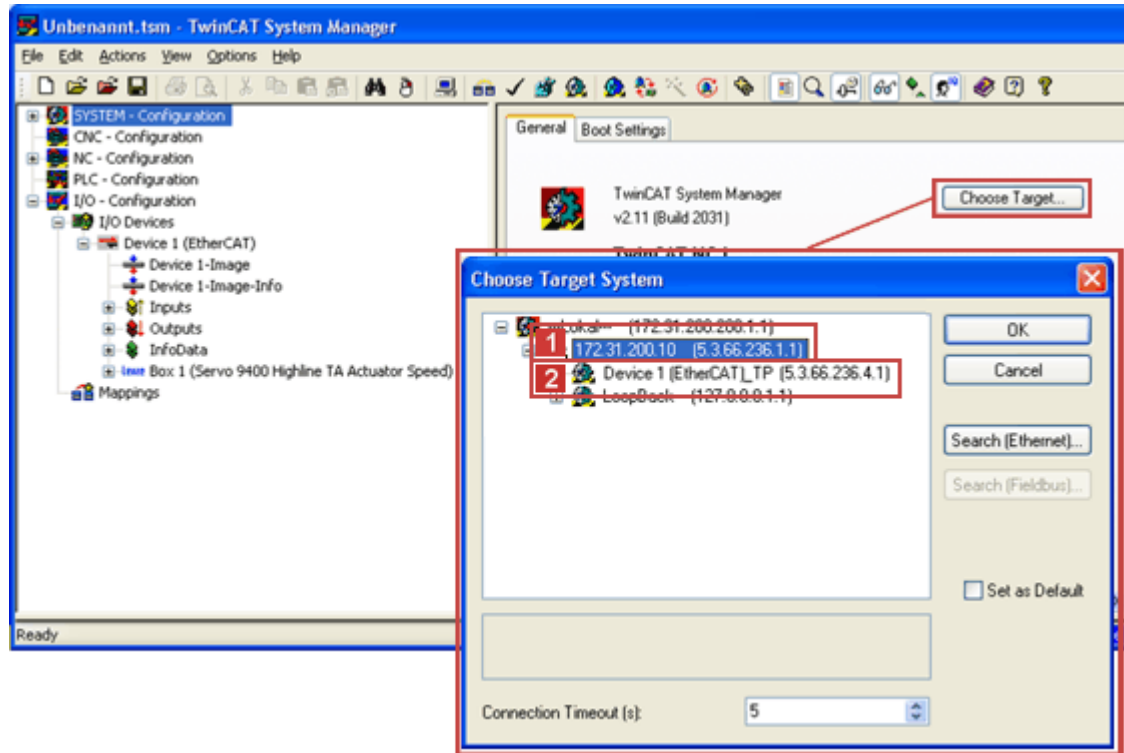
[6-5] Example: EtherCAT bus system with Beckhoff Controller

A Beckhoff DIN rail IPC with the Microsoft Windows CE operating system is used.

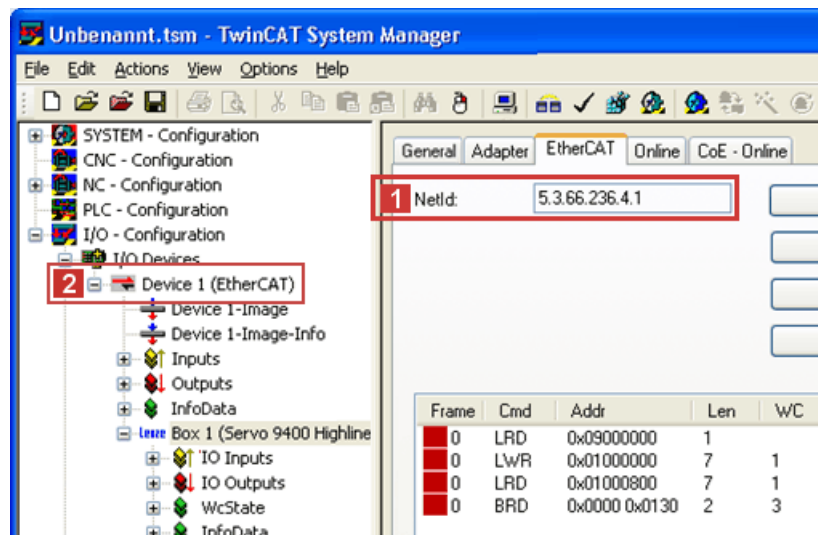
The Beckhoff »TwinCAT« and the Lenze »Engineer« are installed on a Windows XP PC.

Display of the communication parameters in »TwinCAT«

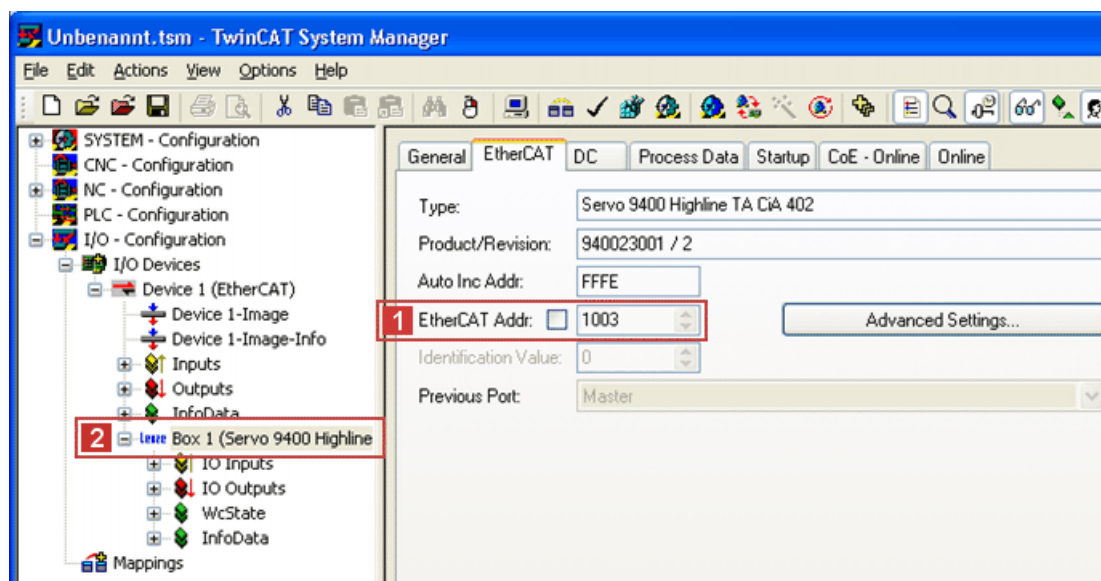
The communication parameters **1** IP address (here '172.31.200.10') and **2** EtherCAT Master Net ID (here '5.3.66.236.4.1') can be found under the target system selection:



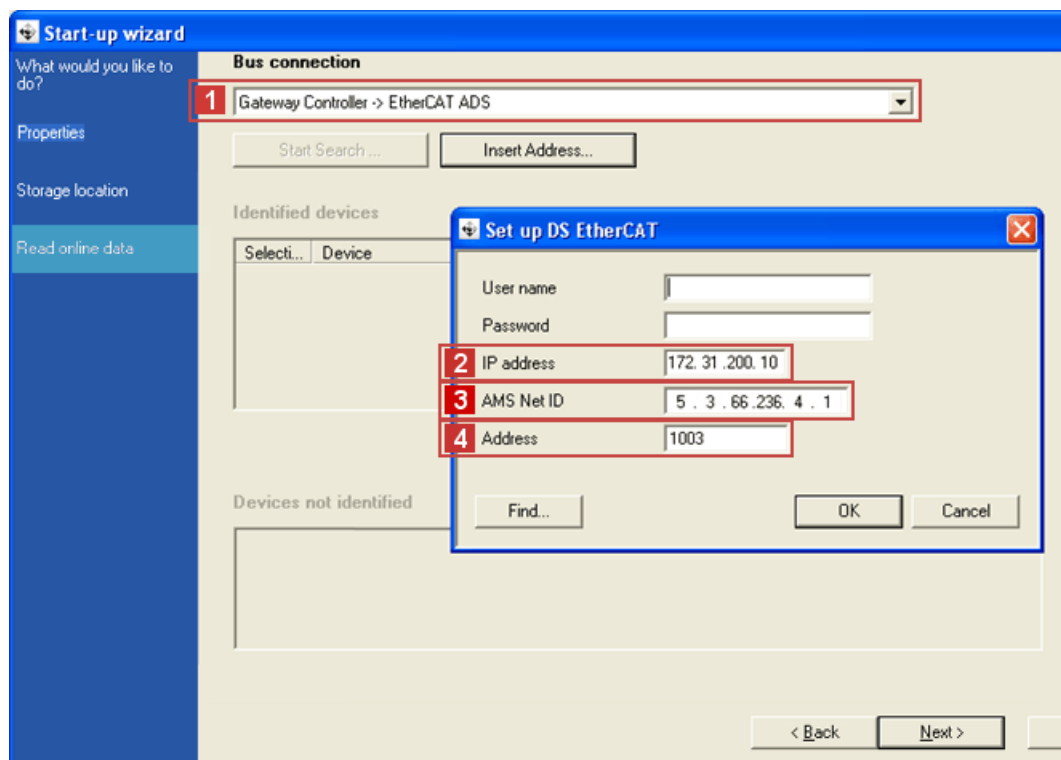
The **1** EtherCAT Master Net ID (here '5.3.66.236.4.1') can also be found under the **EtherCAT** tab of the **2** EtherCAT master:



The EtherCAT **1** slave address (here '1003') can be found under the **EtherCAT** tab of the **2** EtherCAT slave:



Online device identification in the »Engineer« start-up wizard



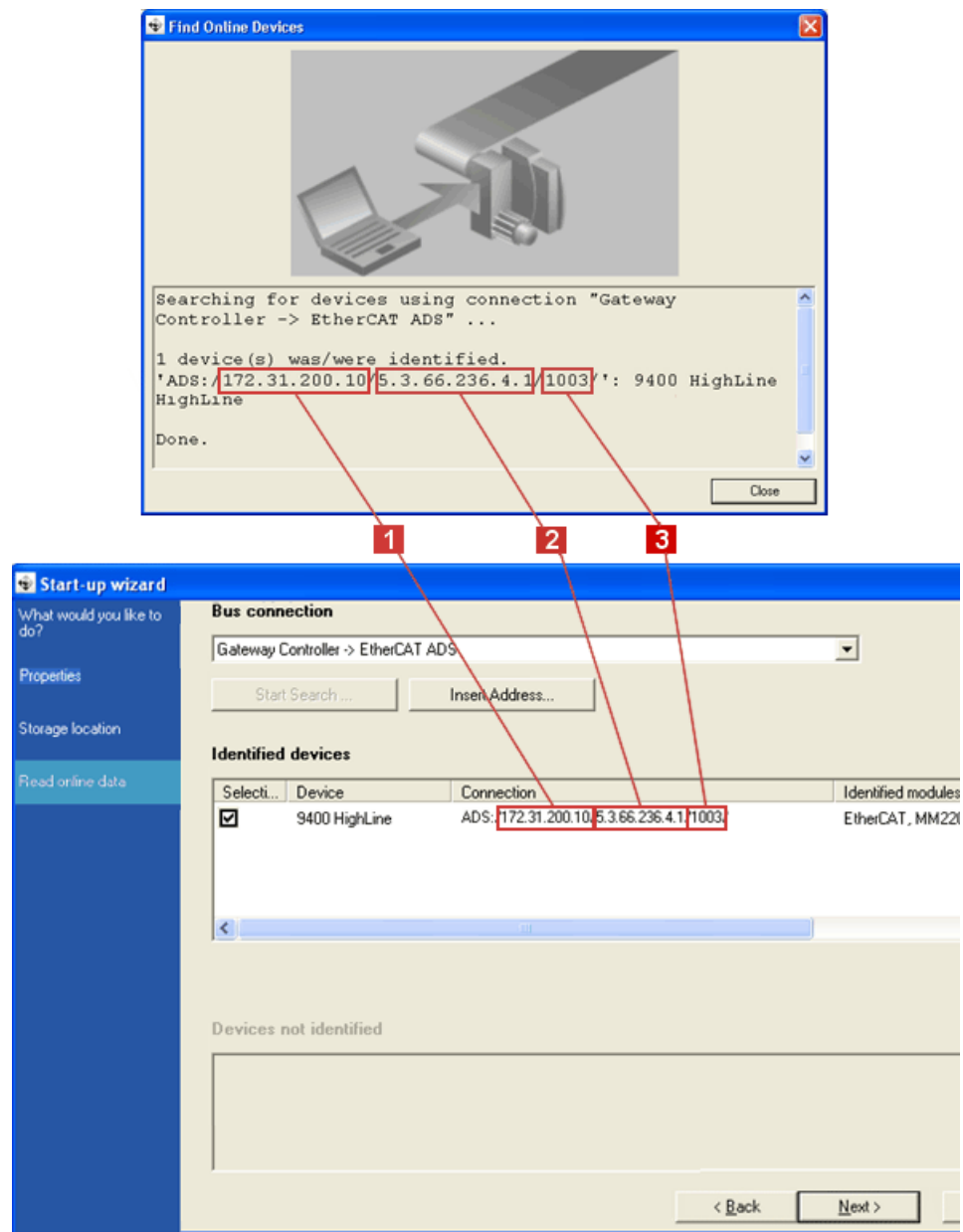
In the »Engineer« start-up wizard under the **1 Gateway Controller -> EtherCAT ADS** bus connection, a field device was detected online:

2 IP address: 172.31.200.10

3 EtherCAT Net ID: 5.3.66.236.4.1

4 EtherCAT slave address: 1003

Display of the identification of the field device detected in the »Engineer« start-up wizard:



1 IP address: 172.31.200.10

2 EtherCAT Net ID: 5.3.66.236.4.1

3 EtherCAT slave address: 1003

6.8

Initial switch-on

**Documentation for the Servo Drive 9400**

Observe the safety instructions and information on residual hazards.

**Note!****Establishing communication**

In order to establish communication via an externally supplied communication module, the standard device must be switched on as well.

For further communication of the externally supplied module it is not relevant whether the standard device is switched on or not.

Activating changed settings

To activate changed settings ...

- execute the device command "11: Save start parameter" via the standard device code **C00002** and ...
- then reset the bus device or switch off and on again the voltage supply of the communication module.

Protection against uncontrolled restart

After a fault (e.g. short-term mains failure), it is sometimes undesirable or even impermissible for the drive to restart.

In the Lenze setting of Servo Drives 9400, the restart protection is activated.

Via the standard device code **C00142** ("Auto restart after mains connection") you can set the restart behaviour of the inverter:

C00142 = "0: Inhibited" (Lenze setting)

- The inverter remains inhibited (even if the fault is no longer active).
- An explicit controller enable causes the drive to start up in a controlled manner: LOW-HIGH edge at digital input X5/RFR.

C00142 = "1: Enabled"

- An uncontrolled restart of the drive is possible.

7 Data transfer

Compared with conventional Ethernet, the collision-free transfer of telegrams on the fieldbus makes EtherCAT a real-time capable bus system.

Communication is always initiated by the EtherCAT master, e.g. a Lenze Controller. A telegram sent by the master passes through all EtherCAT slaves. The last slave of the communication chain sends the telegram back to the EtherCAT master. On the way back, the telegram is directly sent to the master, without being processed in the slaves.

EtherCAT transmits data in standard Ethernet frames. The EtherCAT devices only extract the data intended for them while the frame passes through the device. At the same time, output data are inserted into the frame while it passes through the device. Read and write accesses are only executed on a small section of the entire frame - the datagrams. Therefore, it is not necessary to receive the complete frame before it is processed. Each datagram is forwarded with a minimum delay.

EtherCAT transmits process data, parameter data, configuration data and diagnostic data between the EtherCAT master and the inverters (slaves) that are part of the fieldbus. The data is transmitted via corresponding communication channels depending on their time-critical behaviour (see [Process data transfer](#) (□ 61) / [Parameter data transfer](#) (□ 67)).

7.1 EtherCAT telegrams

An EtherCAT telegram is based on a standard Ethernet frame with the Ether type '0x88A4' and has the following structure:

Ethernet header			Ethernet data				FCS
48 bits	48 bits	16 bits	11 bits	1 bit	4 bits	48 ... 1498 bytes	32 bits
Destination	Source	Ether type	Frame header			Datagrams	
			Length	Reserved	Type		

Ethernet header

The Ethernet header contains the following information:

- Target address of the EtherCAT telegram (destination)
- Source address of the EtherCAT telegram (source)
- Type of the EtherCAT telegram (Ether type = 0x88A4)

Ethernet data

The Ethernet data contain the following information:

- Length of the datagram within the EtherCAT telegram (length)
- One reserved bit (Reserved)
- Type of the datagrams within the EtherCAT frame (type)
- Datagrams

FCS

Checksum of the EtherCAT frame

7.2 EtherCAT datagrams

Read and write accesses are always only executed in one small section of the complete EtherCAT frame – the datagrams.

EtherCAT datagrams have the following structure:

EtherCAT Command header	Data	WKC
10 bytes	Max. 1486 bytes	2 bytes

EtherCAT Command header

The EtherCAT command header contains the following information:

- Command to be executed
- Addressing information
- Length of the data area (Data)
- Interrupt field

Data

The data area contains the data of the command to be executed.

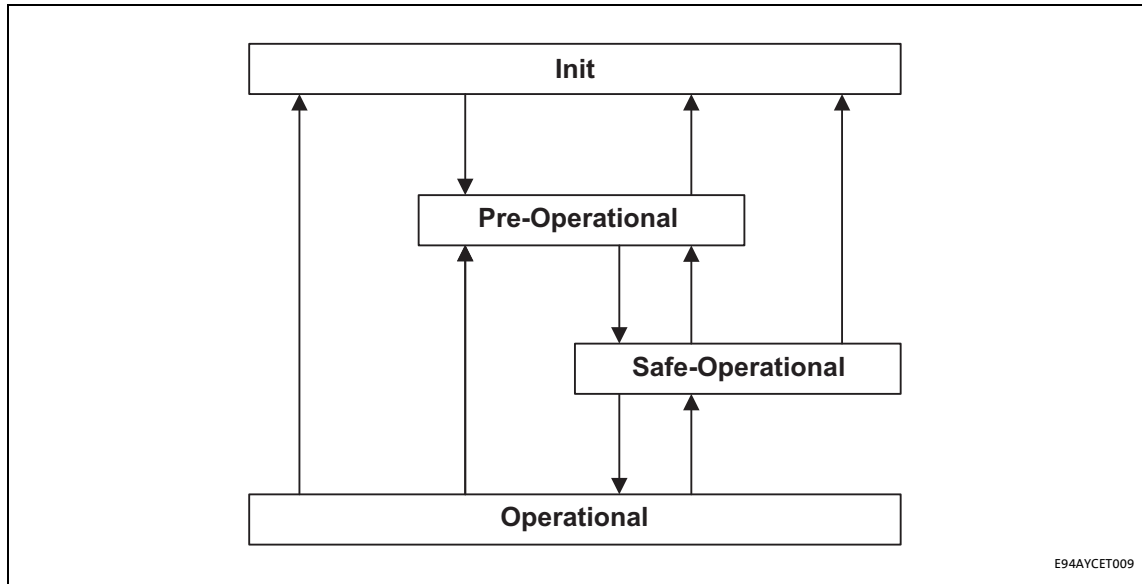
WKC (Working Counter)

The working counter is evaluated by the master for monitoring the execution of the command.

7.3

EtherCAT state machine

Before communication is possible via EtherCAT, the fieldbus passes through the EtherCAT state machine during start-up. The following illustration depicts the possible state changes from the point of view of an EtherCAT slave:



[7-1] EtherCAT state machine

Status	Description
Init	- Initialisation phase - No SDO/PDO communication with the slaves - Device can be detected by fieldbus scan
Pre-Operational	- The fieldbus is active. - SDO communication (mailbox communication) is possible. - No PDO communication
Safe-operational	- SDO communication (mailbox communication) is possible. - PDO communication: - The input data in the process image are updated. - The output data from the process image are not transferred to the slaves.
Operational	Normal operation: - SDO communication - PDO communication - Fieldbus synchronisation has been successful (if used)



Note!

- A fieldbus scan can be carried out during any EtherCAT status.
- SDO communication via the EtherCAT bus is only possible if at least the "Pre-Operational" state has been reached.

The current status of the EtherCAT state machine is shown in [C13861](#) / [C14861](#) and is indicated via the **RUN** LED.

Possible errors at the state transitions are shown in [C13879](#) / [C14879](#). Additionally, an error message is entered in the EtherCAT "AL state code" register (see below).

► [Diagnostics with the »Engineer«](#) (📖 90)

► [EtherCAT status displays](#) (📖 87)

AL Status Code

Information about how the "AL Status Code" EtherCAT register (address 0x0134:0x0135) can be accessed can be found in the documentation of the EtherCAT master.

These error messages can be entered in the "AL Status Code" register:

AL Status Code [hex]	Description
0x0000	No error
0x0011	Invalid status change requested
0x0012	Unknown status requested
0x0013	"Bootstrap" status is not supported
0x0016	Invalid mailbox configuration "Pre-operational"
0x001A	Synchronisation error
0x001B	Sync manager watchdog
0x001D	Invalid output data configuration
0x001E	Invalid input data configuration
0x002B	Invalid input and output data
0x0030	Invalid configuration of DC synchronisation
0x9001	Firmware watchdog error
0x9002	Mapping error

8 Process data transfer

Process data are transmitted by means of [EtherCAT datagrams](#) (☐ 58) via the process data channel.

Process data serve to control the Servo Drive 9400.

The transmission of process data is time-critical.

Process data are transferred cyclically between the Controller (EtherCAT master) and the inverters (slaves) (continuous exchange of current input and output data).

The master can directly access the process data. In the PLC for instance, the data are directly stored in the I/O area.

Up to 32 process data words (16 bits/word) per direction can be exchanged.

Process data are not saved in the Servo Drive 9400.

Process data are e.g. setpoints, actual values, control words, and status words.

8.1

Configuring process data (PDO mapping)

**Stop!**

No application must be downloaded in the "Safe-Operational" or "Operational" state, since the PDO mapping of the EtherCAT master may differ from the PDO mapping of the inverter (slave).

Individual setting of the PDO mapping via the Lenze »Engineer«

The PDO mapping must be set in the »Engineer« if ...

- the "empty application" has been selected in the basic device as a starting point for a technology application;
- own ports are created for a technology application of the standard device. (Only possible with activated application in the »FB Editor«.)
 - ▶ [Setting PDO mapping with the »Engineer«](#) (📖 63)

Setting of the PDO mapping exclusively via the EtherCAT configuration software**Note!**

In the »Engineer« "Process data object structure: PDO_RX0 / PDO_TX0" dialog, mapping is not displayed if it has been set exclusively through the EtherCAT-configuration software.

▶ [Configuring the Controller \(EtherCAT master\)](#) (📖 31)

The current mapping is entered in the mapping codes.

▶ [Mapping indices and codes](#) (📖 65)

In order that the predefined PDO mapping from the device description file can be used, a standard technology application must be configured in the standard device. Moreover, the standard ports of the configured standard technology application must be used.

8.2 Setting PDO mapping with the »Engineer«

The Servo Drives 9400 enable the individual mapping of process data. For this purpose, the »Engineer« is provided with a port configurator.



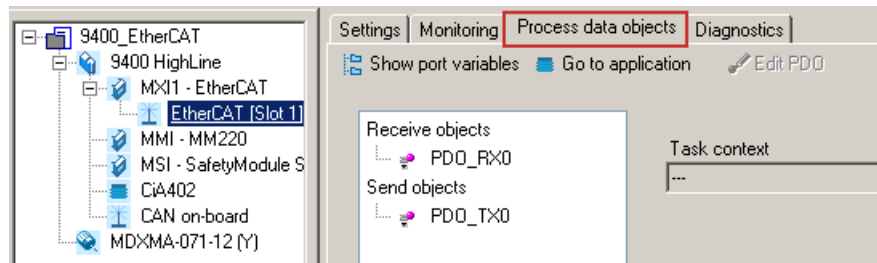
Note

The port mapping is not a configuration, which can be carried out online for the Servo Drive 9400 HighLine. For this purpose, the »Engineer« project must be updated and then the application must be downloaded.

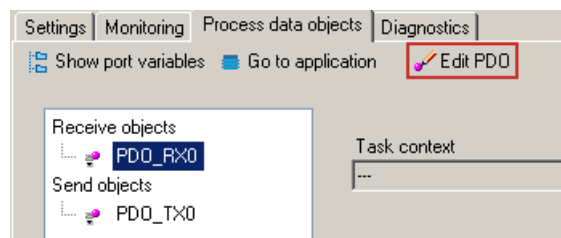


How to set the PDO mapping with the »Engineer«:

1. The process data mapping to the port variables is executed in the »Engineer« under the **Process data objects** tab of the corresponding fieldbus communication module:

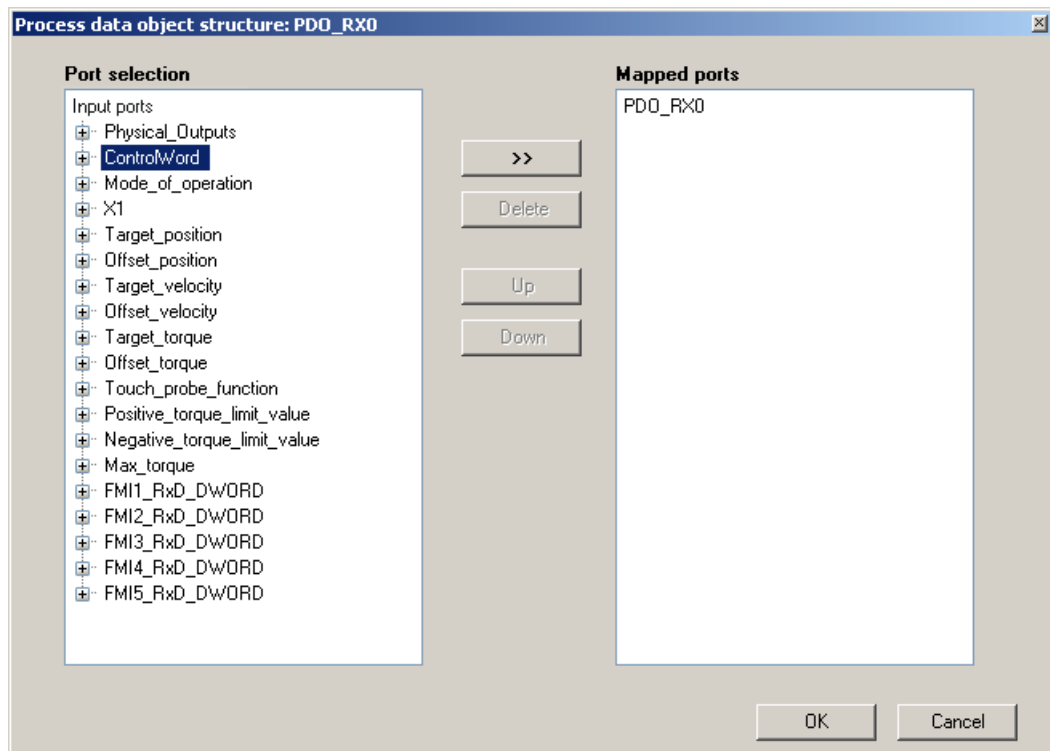


2. Select the receive object **PDO_RX0**:



3. Click the **Edit PDO** button.

The selection window for editing the process data object is opened:



Here you can transfer the individual ports from the **Port Selection** list to the "PDO_RX0" receive PDO by clicking the >> button.

The **Up** and **Down** buttons serve to shift the sequence of the ports within the PDO.

4. Confirm the selection with **OK**.
5. Repeat the steps 2. to 4. for the transmit object **PDO_TX0**.
6. Now link the ports to the application signals in the selected technology application.

Activate the »FB Editor« via the multiplexer codes (> C03000) if this has not been done yet.

If the »FB Editor« is activated, the multiplexer codes (> C03000) are no longer available. In this case, you must carry out the interconnection directly in the »FB Editor«.

8.3 Mapping indices and codes

The settings caused by the previous steps result in mapping indices that are required for the configuration of the master and are stored as codes.

The mapping indices always show the currently valid PDO mapping for the inverter and are displayed on CoE objects 0x1600 (RPDOs) and 0x1A00 (TPDOs) on the EtherCAT.

10 mapping channels each are available for RPDOs and TPDOs. Only one channel at a time can be activated by the EtherCAT master.



Note!

- If the PDO mapping is set via the »Engineer«, the EtherCAT configuration software must load the mapping from the inverter in order that it can be written back to the controller when the network is started.
This ensure that the mapping indices in the EtherCAT master and slave are identical.
- The mapping codes must not be modified by the user.

Slot in the Servo Drive 9400	Code	Correspondent CoE object	Description
MXI1	C13231	0x1600.01..0x20	Mapping entries RPDO-1 ... 10
	... C13240	0x1609.01..0x20	
	C13531	0xA100.01..0x20	Mapping entries TPDO-1 ... 10
	... C13540	0xA109.01..0x20	
	C13484	0x1C12.01	Active RPDO channel
	C13489/1	0x1600.00	Number of RPDO mapping entries RPDO-1 ... 10
	... C13489/10	0x1609.00	
	C13784	0x1C13.01	Active TPDO channel
	C13789/1	0x1A00.00	Number of TPDO mapping entries TPDO-1 ... 10
	... C13789/10	0x1A09.00	
MXI2	C14231	0x1600.01..0x20	Mapping entries RPDO-1 ... 10
	... C13240	0x1609.01..0x20	
	C14531	0xA100.01..0x20	Mapping entries TPDO-1 ... 10
	... C14540	0xA109.01..0x20	
	C14484	0x1C12.01	Active RPDO channel
	C14489/1	0x1600.00	Number of RPDO mapping entries RPDO-1 ... 10
	... C14489/10	0x1609.00	
	C14784	0x1C13.01	Active TPDO channel
	C14789/1	0x1A00.00	Number of TPDO mapping entries TPDO-1 ... 10
	... C14789/10	0x1A09.00	

8.3.1 Structure of the mapping codes

The mapping codes [C13231 ... C13240](#) and [C14231 ... C14240](#) have the following structure:

Octet offset	Name	Description
0 / 1	Index	Index of the object to be mapped
2	Subindex	Subindex of the object to be mapped
3	Reserved	
4 / 5	Offset	Offset regarding the PDO start in bits
6 / 7	Length	Length of the object to be mapped in bits

Example

In the RPDO channel 1, the index 0xA640.0x03 is transmitted with a length of 4 bytes from an offset of 20 bytes.

- Entry in the mapping code **C1x231/8**: 0x002000A00003A640

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Index		Subindex	Reserved	Offset		Length	
0x40	0xA6	0x03	0x00	0xA0	0x00	0x20	0x00

- Correspondent entry in the CoE object **0x1600/8**: 0xA6400320

8.3.2 FSoE mapping indices

The FSoE mapping indices can be used from [SW version 04.00](#) and show the PDO mapping for the safety PDOs. On the EtherCAT side, they are mapped onto the CoE objects **0x1700** (Safety RxPDO) and **0x1B00** (Safety TxPDO).

The FSoE mapping is activated by the configurator via the indices **0x1C12.02 = 0x1700** and **0x1C13.02 = 0x1B00**.

The FSoE mapping is only possible with the simultaneous use of a valid PDO mapping.

► [Safety over EtherCAT](#) (81)

9

Parameter data transfer

Parameter data are transmitted via the fieldbus as so-called SDOs (Service Data Objects). The SDO services provide for the write and read access to the object directory.

The SDO channel provides for access to [Communication objects \(object directory\)](#) (78) and Lenz codes with the CoE protocol.

If the "CiA402" technology application is used in the inverter, all CANopen CiA402 objects implemented can be accessed. (See documentation on the "CiA402" technology application in the reference manual/online help for the Servo Drive 9400.)

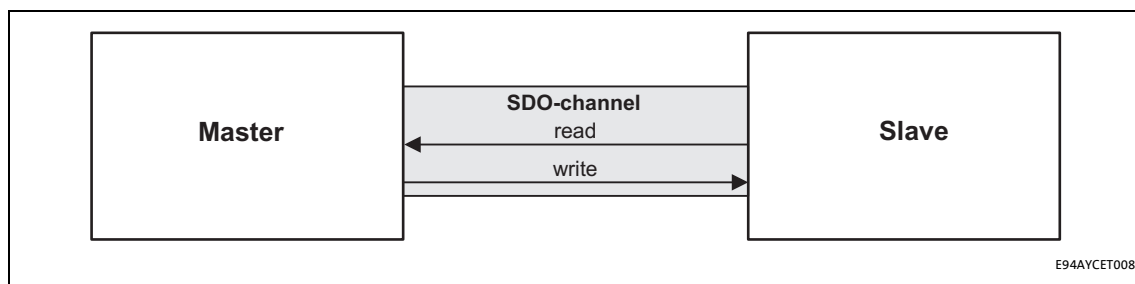
The transmission of parameter data is usually not time-critical.

Parameter data are, for instance, operating parameters, motor data, diagnostic information.

9.1

Establishing a connection between master and slave

Basically a master can always request parameter jobs from a slave if the slave is at least in the "Pre-operational" state.



[9-1] Data communication via the SDO channel

9.2 Reading and writing parameters

Parameters ...

- for instance are set for one-time system settings or if materials are changed within a machine;
- are transmitted with a low priority.

In the case of Lenze inverters, the parameters to be changed are contained in codes or in the case of the CANopen device profile "CiA402" as device profile objects.

Indexing of the Lenze codes

If they are accessed via a communication module, the codes of the inverter are addressed by the index.

The index for Lenze code numbers is in the manufacturer-specific area of the object directory between 8192 (0x2000) and 24575 (0x5FFF).

Conversion formula	
Index [dec]	Index [hex]
24575 - Lenze code	0x5FFF - Lenze code _{hex}

Example for C00002 (device commands)	
Index [dec]	Index [hex]
24575 - 2 = 24573	0x5FFF - 2 = 0x5FFD

Structure of a mailbox datagram

In a datagram, mailbox data are transferred within an EtherCAT telegram. The data area of the mailbox datagram has the following structure:

Mailbox header	CoE header	SDO control byte	Index	Subindex	Data	Data
6 bytes	2 bytes	1 byte	2 bytes	1 byte	4 bytes	1 ... n bytes

9.2.1 Reading parameters (SDO upload)

1. The master sends "Initiate Domain Upload Request".
2. The slave acknowledges the request with a positive response ("Initiate Domain Upload Response").

In the event of an error the slave responds with "Abort Domain Transfer".



Note!

In the case of jobs for the inverter, please make sure that you convert the code into an index.

► [Indexing of the Lenze codes](#) (68)

SDO Upload Request

Detailed breakdown of the data for an "SDO Upload Request":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the initiator. Station address of the target if an EtherCAT slave is the initiator.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Reserved	BYTE	4 bits (0 ... 3)	0x00
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is read. 0x01: The complete object is read. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x02: Upload request
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Reserved	DWORD	4 bytes	0x00

SDO Upload Expedited Response

An "SDO Upload Expedited Response" is effected if the data length of the parameter data to be read is up to 4 bytes.

Detailed breakdown of the data for an "SDO Upload Expedited Response":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the initiator. Station address of the target if an EtherCAT slave is the initiator.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x01: Size of the data in "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data 0x01: 3 bytes of data 0x02: 2 bytes of data 0x03: 1 byte of data
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is read. 0x01: The complete object is read. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x02: Upload response
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Data	DWORD	4 bytes	Data of the object

SDO Upload Expedited Response

An "SDO Upload Normal Response" is effected if the data length of the parameter data to be read $\geq$ is 4 bytes.

Detailed breakdown of the data for an "SDO Upload Normal Response":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	$n \geq 0x0A$: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the initiator. Station address of the target if an EtherCAT slave is the initiator.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits 0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x01
	Transfer type		1 bit (1)	0x00: Normal transfer
	Data set size		2 bits (2, 3)	0x00
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is read. 0x01: The complete object is read. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x02: Upload response
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Complete size	DWORD	4 bytes	Total data length of the object
	Data	BYTE	n - 10 bytes	Data of the object

Example

The response structure transferred to index 0x5FC2 (standard setting of C00061/0 (heatsink temperature) = 0x0000002B (43 °C)) in the case of an **Upload** contains the following data:

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	0x00
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x01: Length of the data in "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is read.
	Command specifier		3 bits (5 ... 7)	0x02: Upload response
	Index	WORD	2 bytes	0xC2: Index low byte of the object 0x5F: Index high byte of the object
	Subindex	BYTE	1 byte	0x00
	Data	DWORD	4 bytes	0x0000002B

9.2.2 Writing parameters (SDO download)

1. The master sends "Initiate Domain Download Request".
2. The slave acknowledges the request with a positive response ("Initiate Domain Download Response").

In the event of an error the slave responds with "Abort Domain Transfer".



Note!

In the case of jobs for the inverter, please make sure that you convert the code into an index.

► [Indexing of the Lenze codes](#) (📖 68)

SDO Download Expedited Request

An "SDO Download Expedited Request" is effected if the data length of the parameter data to be written is up to 4 bytes.

Detailed breakdown of the data for an "SDO Download Expedited Request":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the initiator. Station address of the target if an EtherCAT slave is the initiator.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Size indicator	BYTE	1 bit (0)	0x01: Size of the data in "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data 0x01: 3 bytes of data 0x02: 2 bytes of data 0x03: 1 byte of data
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is written. 0x01: The complete object is written. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x01: Download request
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Data	DWORD	4 bytes	Data of the object

SDO Download Expedited Request

An "SDO Download Normal Request" is effected if the data length of the parameter data to be written $\geq$ is 4 bytes.

Detailed breakdown of the data for an "SDO Download Normal Request":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	$n \geq 0x0A$: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the initiator. Station address of the target if an EtherCAT slave is the initiator.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Size indicator	BYTE	1 bit (0)	0x01
	Transfer type		1 bit (1)	0x00: Normal transfer
	Data set size		2 bits (2, 3)	0x00
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is written. 0x01: The complete object is written. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x01: Download request
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Complete size	DWORD	4 bytes	Total data length of the object
	Data	BYTE	n - 10 bytes	Data of the object

SDO Download Response

Detailed breakdown of the data for an "SDO Download Response":

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	Station address of the source if an EtherCAT master is the initiator. Station address of the target if an EtherCAT slave is the initiator.
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority ... 0x03: Highest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x03: SDO response
SDO	Size indicator	BYTE	1 bit (0)	0x0
	Transfer type		1 bit (1)	0x0
	Data set size		2 bits (2, 3)	0x0
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is written. 0x01: The complete object is written. (Currently not supported.)
	Command specifier		3 bits (5 ... 7)	0x3: Download response
	Index	WORD	2 bytes	Index of the object
	Subindex	BYTE	1 byte	Subindex of the object 0x00 or 0x01 if "Complete access" = 0x01.
	Reserved	DWORD	4 bytes	0x00

Example

The request structure transferred in the case of a **Download** to the index 0x5FA7 (C00088/0, rated motor current $I = 10.2$ A) contains the following data:

SDO frame area	Data field	Data type / length		Value / description
Mailbox header	Length	WORD	2 bytes	0x0A: Length of the mailbox service data
	Address	WORD	2 bytes	0x00
	Channel	WORD	6 bits (0 ... 5)	0x00: Reserved
	Priority		2 bits (6, 7)	0x00: Lowest priority
	Type		4 bits (8 ... 11)	0x03: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x00
CANopen header	Number	WORD	9 bits (0 ... 8)	0x00
	Reserved		3 bits (9 ... 11)	0x00
	Service		4 bits (12 ... 15)	0x02: SDO request
SDO	Size indicator	BYTE	1 bit (0)	0x01: Size of the data in "Data set size"
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes of data
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is written.
	Command specifier		3 bits (5 ... 7)	0x01: Download request
	Index	WORD	2 bytes	0xA7: Index low byte of the object 0x5F: Index high byte of the object
	Subindex	BYTE	1 byte	0x00: Subindex of the object
	Data	DWORD	4 bytes	0x00000066 (10.2 x 10 = 102)

9.3

Communication objects (object directory)

Index	Name	Subindex	Subindex name	Type	Bits	Access
0x1000	Device Type	0	-	UDINT	32	R
0x1001	Error register	0	-	USINT	8	R
0x1008	Device name	0	-	STRING(8)	64	R
0x1009	Hardware version	0	-	STRING(8)	64	R
0x100A	Software version	0	-	STRING(7)	56	R
0x1018	Identity	0	Number of elements	USINT	8	R
		1	Vendor ID	UDINT	32	R
		2	Product Code	UDINT	32	R
		3	Revision number	UDINT	32	R
		4	Serial Number	UDINT	32	R
0x10F1	Error Settings *	0	Number of elements	USINT	8	R
		1	Local Error Reaction	UDINT	32	RW
		2	Sync Error Counter Limit	UINT	16	RW
0x1600 ... 0x1609	RxPDO 1 ... 10	0	Number of elements	USINT	8	RW **
		1 ... 32	Output object 1 ... 32	UDINT	32	RW **
0x1700	Safety RxPDO	0	Number of elements	USINT	8	R ***
		1 ... 36	Safety Output Objects	UDINT	32	R ***
0x1A00 ... 0x1A09	TxPDO 1 ... 10	0	Number of elements	USINT	8	RW
		1 ... 32	Input object 1 ... 32	UDINT	32	RW
0x1B00	Safety TxPDO	0	Number of elements	USINT	8	R ***
		1 ... 76	Safety Input Objects	UDINT	32	R ***
0x1C00	Sync Man Communication type	0	Number of elements	USINT	8	R
		1 ... 4	Elements	USINT	8	R
0x1C12	Sync Man 2 Assignment	0	Number of assigned RxPDOs	USINT	8	RW **
		1	PDO Mapping object index of assigned RxPDO	UINT	16	RW **
		2	PDO Mapping index of assigned Safety RxPDO	UINT	16	RW *
0x1C13	Sync Man 3 Assignment	0	Number of assigned TxPDOs	USINT	8	RW **
		1	PDO Mapping object index of assigned TxPDO	UINT	16	RW **
		2	PDO Mapping index of assigned Safety TxPDO	UINT	16	RW **
0x1C32	Sync Man 2 Synchronization	0	Number of elements	USINT	8	R
		1	Synchronization type	UINT	16	R
		2	Cycle time / ns	UDINT	32	R
		3	Shift time / ns	UDINT	32	R
		4	Sync types supported	UINT	16	R
		5	Minimum cycle time / ns	UDINT	32	R
		6	Minimum shift time / ns	UDINT	32	R
		9	Delay time / ns	UDINT	32	R *
		12	Cycle time too small	UINT	16	R *

Index	Name	Subindex	Subindex name	Type	Bits	Access
0x1C33	Sync Man 3 Synchronization	0	Number of elements	USINT	8	R
		1	Synchronization type	UINT	16	R
		2	Cycle time / ns	UDINT	32	R
		3	Shift time / ns	UDINT	32	R
		4	Sync types supported	UINT	16	R
		5	Minimum cycle time / ns	UDINT	32	R
		6	Minimum shift time / ns	UDINT	32	R
		9	Delay time / ns	UDINT	32	R *
		12	Cycle time too small	UINT	16	R *
0xE600	FSOE Slave Frame Elements	0	Number of elements	UINT	8	R ***
		1	FSOE Slave Command	UINT	16	R ***
		2	FSOE Slave Connection ID	UINT	16	R ***
		3	FSOE Slave CRC_0	UINT	16	R ***
		...	...			
		9	FSOE Slave CRC_6			
0xE700	FSOE Master Frame Elements	0	Number of elements	UINT	8	R ***
		1	FSOE Master Command	UINT	8	R ***
		2	FSOE Master Connection ID	UINT	16	R ***
		3	FSOE Master CRC_0	UINT	16	R ***
		4	FSOE Master CRC_1			
0xE901	FSOE Connection Communication Parameter	0	Number of elements	USINT	8	R ***
		1	Version	UINT	16	R ***
		2	Safety Address	UINT	16	R ***
		3	Connection ID	UINT	16	R ***
		4	Watchdog Time	UINT	16	R ***
		5	Reserved (Unique ID)	VSTRING		R ***
		6	Connection Type	UINT	8	R ***
		7	Com Parameter Length	UINT	16	R ***
		8	Appl Parameter Length	UINT	16	R ***
0xEA00	FSOE Connection Diagnosis	0	Number of elements	USINT	8	R ***
		1	Connection state	UINT	16	R ***
0xF980	Device Safety Address	0	Number of elements	USINT	8	R ***
		1	FSOE Address	UINT	16	R ***

R: Read access only

RW: Read and write access

*: from SW version 04.00

**: from SW version 04.00 only in the "Pre-Operational" state

***: from SW version 04.00 only available if FSoE is used

9.4 SDO abort codes (Abort codes)

If an SDO request is evaluated negatively, a corresponding error code is output.

Index [hex]	Description
0x00000000	No error
0x05030000	The status of the toggle bit has not changed
0x05040000	SDO protocol time-out
0x05040001	Invalid or unknown specification symbol for the client/server command
0x05040005	The space in the main memory is not sufficient.
0x06010000	Access to object not supported
0x06010001	Read access to a write-only object
0x06010002	Write access to a read-only object
0x06010004	Complete Access (is not supported)
0x06020000	An object does not exist in the object directory
0x06040041	An object cannot be entered/mapped in the PDO.
0x06040042	The number and/or length of the objects entered/mapped would exceed the PDO length.
0x06040043	General parameter incompatibility
0x06040047	General internal device incompatibility
0x06060000	Access has failed due to a fault in the hardware
0x06070010	The data type or the parameter length does not correspond
0x06070012	Incorrect data type (The parameter length is too large)
0x06070013	Incorrect data type (The parameter length is too small)
0x06090011	A subindex is not available
0x06090030	The value range for parameters is too great (only for write access)
0x06090031	The parameter value is too high
0x06090032	The parameter value is too low
0x06090036	The maximum value is lower than the minimum value
0x08000000	General error
0x08000020	Data cannot be transferred to the application or stored in the application
0x08000021	Due to local control, data cannot be transferred to the application or stored in the application
0x08000022	Due to the current device state, data cannot be transferred to the application or stored in the application
0x08000023	The dynamic generation of an object directory has failed, or no object directory is available.

10 Safety over EtherCAT



Note!

A safety bus system (FSoE) can only be operated via the upper module slot (MXI1) of the Servo Drive 9400.

From SW version 04.00, FSoE (Fail Safe over EtherCAT) provides for transmission of safe information via the "Safety over EtherCAT" protocol according to the "ETG.5100" specification, version 1.2.0.

The FSoE data is configured within a safety PDO and sent in the same Sync unit as the standard PDOs.

One bit of the FSoE data is always used to control specific safety functions.

The structure of the FSoE data is specified and cannot be changed by the user.

If the **SM302** safety module (E94AYAF) is used, the length of the ...

- safety outputs is permanently 11 bytes;
- safety inputs is permanently 19 bytes.

The Servo Drive 9400 forwards the FSoE data to the safety module for a safe evaluation.

▶ [Configuring safety process data](#) (📖 33)



Documentation for the SM302 safety module (E94AYAF)

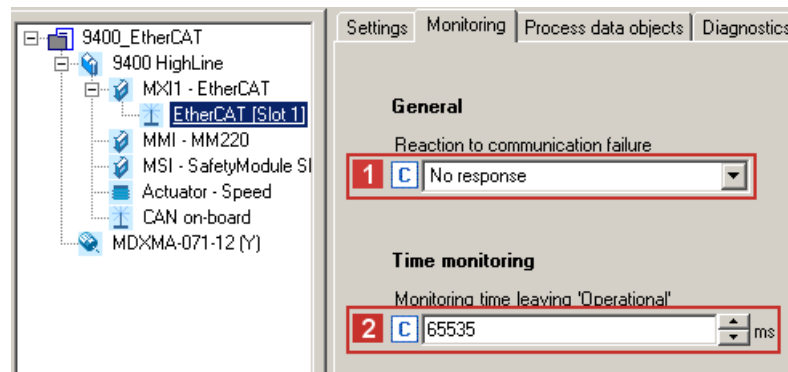
Here you can find detailed information on the FSoE connection.

11 Monitoring

11.1 Interruption of EtherCAT communication

An interruption of the EtherCAT communication in the "Operational" state, e.g. due to cable break or failure of the EtherCAT master, is detected by the slave.

The response to an interrupted communication is triggered by settings in the **Monitoring** tab:



- During initialisation of the EtherCAT communication, the PDO watchdog monitoring time preset in the master ([C13882](#) / [C14882](#)) is transmitted to the slave.

If the slave does not receive any valid process data in the "Operational" state, the setting in [C13885](#) / [C14885](#) is taken as a basis for the process data.

- Value '0': The data sent last by the master are used.
- Value '1': PDOs are set to the value '0'.

After the watchdog monitoring time has elapsed, the slave changes to the "Safe-Operational" state (see [C13861](#) / [C14861](#)) and the green RUN LED is activated (see [EtherCAT status displays](#) ([□ 87](#))).

There is no response in the slave.

- In order that a response is triggered in the slave, you have to set a **1 Reaction on communication failure** ([C13880](#) / [C14880](#)).
- The response is delayed if you set an internal **2 monitoring time** ([C13881](#) / [C14881](#)) in addition.

The monitoring time elapses as soon as the "Operational" state is exited. (See 1.).

After the monitoring time has elapsed, the set response is executed with the error message "[EtherCAT: Quit 'Operational' state \[0x00c88131\]](#)" ([□ 95](#)).

11.2 Interruption of internal communication

The response in the case of a communication error between the communication module and the standard device can be set via codes [C01501](#) (module in slot MXI1) and [C01502](#) (module in slot MXI2).

An externally supplied communication module reports a connection abort to the standard device via an emergency telegram to the master and changes to the "Safe-operational" state.

11.3 Sync telegram failure detection

During the [Synchronisation with "Distributed clocks" \(DC\)](#) (36), this monitoring mode checks whether an EtherCAT PDO telegram (Sync Manager 2 Event) has arrived between two signals.

For this purpose, the communication module comes with an internal EtherCAT telegram failure error counter. The telegram failure error counter is incremented by the value '3' in case of a telegram failure. For each PDO received correctly, the counter is decremented by '1'.

If the telegram failure error counter reaches a threshold, – adjustable via the EtherCAT master in the CoE object 0x10F1.2 – the state changes to "Safe-Operational" and the "Sync telegram failure [0x01bc8700]" error message is output.

In the standard setting, monitoring is deactivated (0x10F1.2 = 0). A sensible setting is a value > '5' in order that an error will only be triggered in the inverter when a second telegram fails.

An error response can be set via the code [C13880/2](#) / [C14880/2](#) ("0: No response" is preset here).

The application serves to detect telegram failures by evaluating the signals **SYNC_bProcessDataExpected** and **SYNC_bProcessDataInvalid** of the **LS_SyncInput** system block.

This monitoring can be used, for instance, for extrapolating the setpoints (position, speed and torque).



Tip!

The function blocks **L_SdInterExtrapolatePosition** or **L_SdInterExtrapolateAny** are available as extrapolation blocks.

The frame failure detection is implemented by default in the "CiA402" technology application.

11.4 EtherCAT synchronisation loss

The "Sync0" synchronisation signal serves to synchronise the start of the inverter application with the application of the EtherCAT master.

This synchronisation is monitored by the inverter in a permanent time slot.

If impermissible deviations of the synchronisation cycle time are detected within four successive cycles, the inverter generates the error message "[EtherCAT: Sync loss detected with standard device \[0x00c88132\]](#)" ([📖 95](#)).

The permissible jitter of the synchronisation cycle time is 10 µs.

Code [C13884](#) / [C14884](#) displays whether the Servo Drive 9400 is synchronised.



Tip!

The synchronisation can be monitored in the application by the **LS_SyncInput** system block.

12

Diagnostics

For purposes of fault diagnostics, the communication module is provided with the LEDs on the front. Furthermore, you can carry out the [Diagnostics with the »Engineer«](#) (📖 90).

12.1

LED status displays



Note!

During normal operation ...

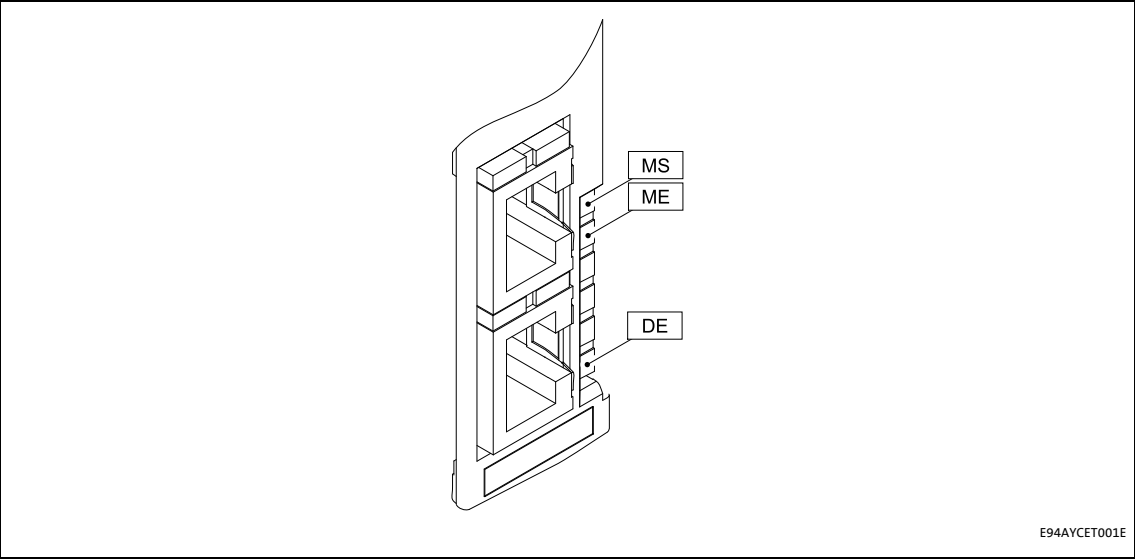
- only the LEDs **MS** (📖 86) and **BS** (📖 87) should be lit constantly.
- the green LED on the RJ45 sockets X246 / X247 must be lit or blinking (📖 89).

The following status displays can be differentiated:

- ▶ [Module status displays](#) (📖 86)
- ▶ [EtherCAT status displays](#) (📖 87)
- ▶ [Status display at the RJ45 sockets \(X246 / X247\)](#) (📖 89)

12.1.1 Module status displays

The module status displays are indicated by the LEDs **MS**, **ME** and **DE**.



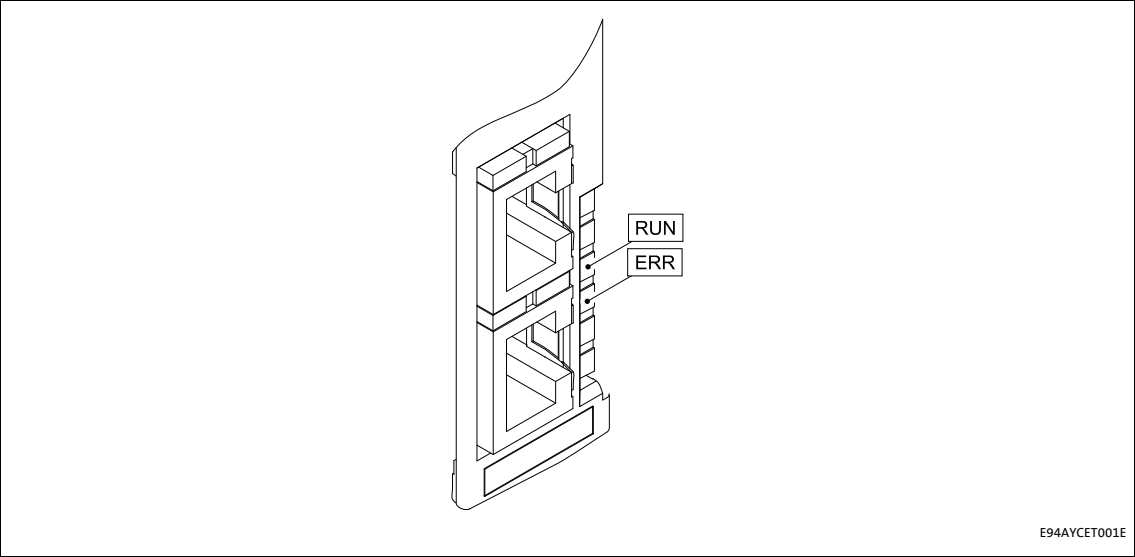
[12-1] Module status displays

Pos.	Colour	Status	Description
MS	Green	On	 The communication module is supplied with voltage and is connected to the standard device.
		Blinking	 The communication module is supplied with voltage, but is not connected to the standard device. (Standard device is switched off, in the initialisation phase, or not available.)
ME	Red	On	 An error has occurred in the communication module.
EN	Red	On	 The communication module is not accepted by the basic device or the basic device is not active (see notes in the documentation relating to the basic device.)

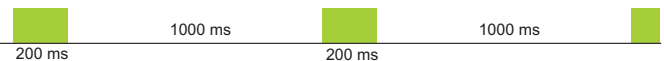
12.1.2 EtherCAT status displays

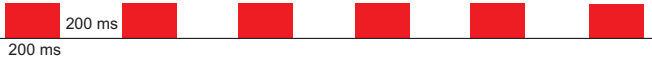
EtherCAT status displays are indicated by the LEDs **RUN** and **ERR**.

- The **RUN** LED shows the current state of the [EtherCAT state machine](#) (59).
- The **ERR** LED shows the current EtherCAT error.



[12-2] EtherCAT status displays

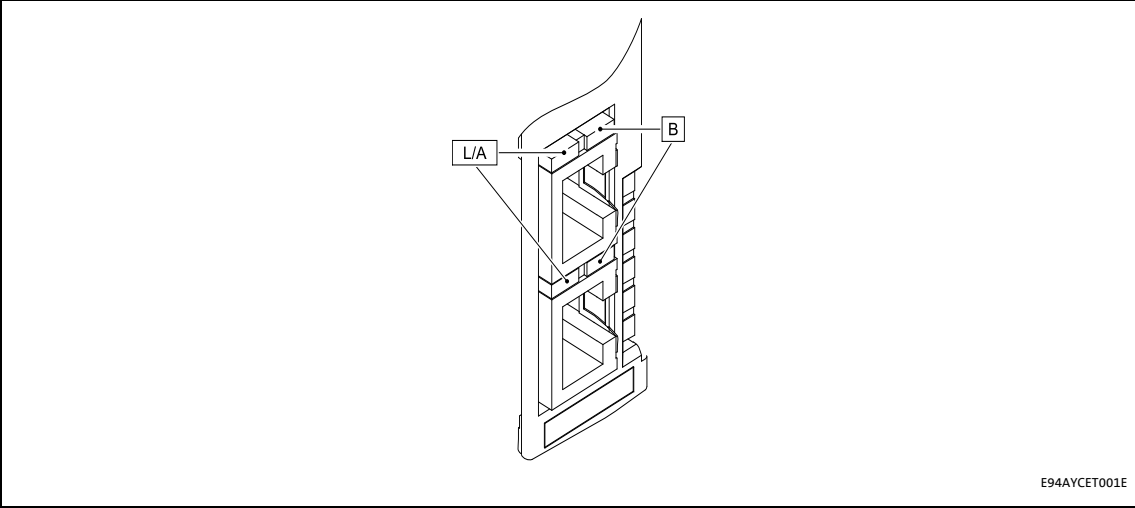
LED	Colour	Status	Description
RUN	Green	Off	The communication module is not active on the fieldbus or is in the "Init" state.
		Blinking	 "Pre-operational" status is active: • Access to parameters and objects is possible. • No process data exchange.
		blinking once (single flash)	 "Safe-operational" status is active: • The data are not yet active in the standard device.
		On	 The communication module is in the "Operational" state.

LED	Colour	Status	Description
ERR	Red	Off	No error
		Blinking	 The configuration is invalid/faulty.
		blinking once (single flash)	 • A non requested state change has occurred. (The slave application has autonomously changed the EtherCAT status.) • Synchronisation error (The EtherCAT device automatically changes to the "Safe-operational" state.)
		blinking twice (double flash)	 An "Application Watchdog Timeout" or a "Sync Manager Watchdog Timeout" has occurred.

12.1.3

Status display at the RJ45 sockets (X246 / X247)

The physical status of the EtherCAT connection with the corresponding communication partner is displayed by the L/A (Link/Activity) LED.

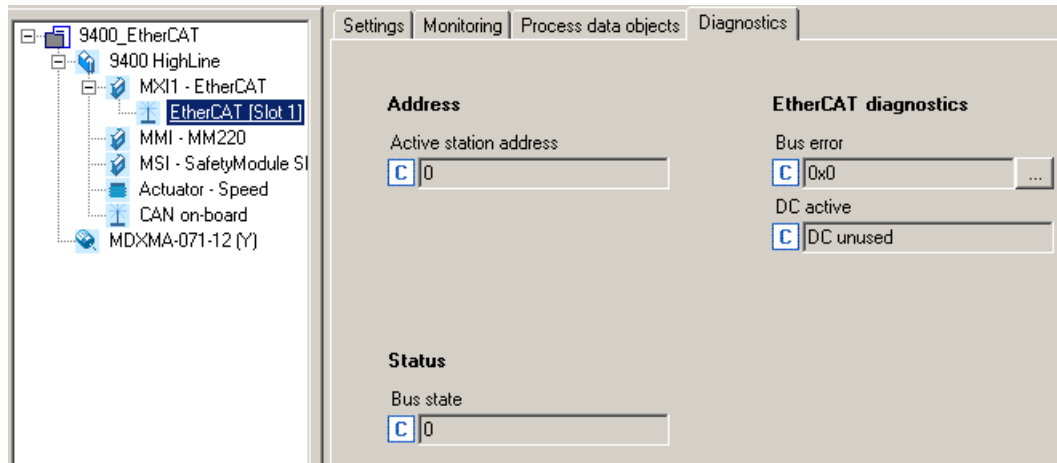


[12-3] Status of the EtherCAT connection

LED	Colour	Status	Description
L/A	Green	On	A physical EtherCAT connection is available.
		Flickers	50 ms Data are being exchanged via EtherCAT.
B	Red	Off	This LED is not used.

12.2 Diagnostics with the »Engineer«

In the »Engineer«, the **Diagnostics** tab displays various pieces of EtherCAT diagnostic information.



12.3 Emergency requests / Emergency messages

Emergency messages are sent to the EtherCAT master once when the error status of the controller changes, i.e. ...

- when an error in the inverter or in the communication module occurs or
- when an internal error of the communication module is eliminated.

An "Emergency Request" on the fieldbus consists of the components "Mailbox Header", "CANopen Header" and the actual "Emergency Message":

Mailbox header	CANopen header	Emergency Message
6 bytes	2 bytes	8 bytes

The emergency message last sent by the Servo Drive 9400 is displayed in code [C13867](#) / [C14867](#).

Structure of the Emergency message

Example: Emergency message of the "EtherCAT error: Quit Operational state [0x00c88131]"

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Emergency Error code		Error Register (I-1001)	Reserved	Error code Servo Drive 9400 / E94AYCET			
Low byte	High byte	Low byte	High byte	Low word		High word	
				Low byte	High byte	Low byte	High byte
0x00	0x10	0x01	0x00	0x31	0x81	0xc8	0x00

- Bytes 1 and 2 display that an error is pending.
- Byte 3 display the contents of the error register (I-1001).
- Bytes 5 ... 8 contain the error code.



Reference manual/online help for the Servo Drive 9400

Here you will find detailed information on error codes.

13 Error messages

This chapter supplements the error list in the reference manual and the »Engineer« online help for the Servo Drive 9400 by the error messages of the E94AYCET (EtherCAT) communication module.



Reference manual/online help for the Servo Drive 9400

Here you will find general information on diagnostics & fault analysis and on error messages.

13.1 Short overview of the EtherCAT error messages

The following table lists all EtherCAT error messages in numerical order of the error number. Furthermore the preset error response and – if available – the parameters for setting the error response are specified.



Tip!

If you click on the cross-reference in the first column, you will get a detailed description (causes and remedies) of the corresponding error message.

Error number		Name	Response (Lenze setting)	Adjustable in
hex	dec			
0x00c83100	13119744	EtherCAT: Connection to 9400 lost	No response	-
0x00c85114	13127956	EtherCAT: Internal supply voltage too low	Information	-
0x00c85531	13129009	EtherCAT: Memory: No access	Information	-
0x00c85532	13129010	EtherCAT: Memory: Error while reading	Information	-
0x00c85533	13129011	EtherCAT: Memory: Error while writing	Information	-
0x00c86010	13131792	EtherCAT: Restart after watchdog reset	Fault	-
0x00c86011	13131793	EtherCAT: Internal error	Fault	-
0x00c86100	13132032	EtherCAT: Internal error	Information	-
0x00c86101	13132033	EtherCAT: Internal error	Fault	-
0x00c86110	13132048	EtherCAT: Internal mapping error	Warning Locked	-
0x00c8641f	13132831	EtherCAT: Parameter set invalid	Fault	-
0x00c86420	13132832	EtherCAT: Error: Lenze setting loaded	Fault	-
0x00c86610	13133328	EtherCAT: Communication time-out with safety module (from SW version 04.00)	Warning Locked	
0x00c86612	13133330	EtherCAT: Initialisation error with safety module (from SW version 04.00)	Warning Locked	
0x00c88131	13140273	EtherCAT: Quit the 'Operational' status	No response	C13880/1 / C14880/1
0x00c88132	13140274	EtherCAT: Sync loss detected in standard device	Warning Locked	-
0x00c88700	13141760	EtherCAT: Sync telegram failure	No response	C13880/2 / C14880/2

13.2 Possible causes and remedies

This chapter lists all EtherCAT error messages in the numerical order of the error numbers. Possible causes and remedies as well as responses to the error messages are described in detail.

EtherCAT: Connection to 9400 lost [0x00c83100]

Response (Lenze setting printed in bold)	Setting: not possible
☒ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
The MXI communication to the Servo Drive 9400 is interrupted. • Servo Drive 9400 is switched off. • The communication module is not connected correctly at the MXI slot of the Servo Drive 9400.	• Switch on Servo Drive 9400. • Plug in the communication module correctly at the MXI slot of the Servo Drive 9400. • Send communication module and Servo Drive 9400 with error description to Lenze.

EtherCAT: Internal supply voltage too low [0x00c85114]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ Information	
Cause	Remedy
Voltage loss of the standard device	Check voltage supply.

EtherCAT: Memory: No access [0x00c85531]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ Information	
Cause	Remedy
Access to memory was not possible.	Send communication module together with a description of the fault to Lenze.

EtherCAT: Memory: Error while reading [0x00c85532]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ Information	
Cause	Remedy
Parameter could not be read.	Send communication module together with a description of the fault to Lenze.

EtherCAT: Memory: Error while writing [0x00c85533]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ Information	
Cause	Remedy
Parameter could not be written.	Send communication module together with a description of the fault to Lenze.

EtherCAT: Restart by watchdog reset [0x00c86010]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
Communication module is defective.	Send communication module together with a description of the fault to Lenze.

EtherCAT: Internal error [0x00c86011]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
Communication module is defective.	Send communication module together with a description of the fault to Lenze.

EtherCAT: Internal error [0x00c86100]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ Information	
Cause	Remedy
Internal error.	If the error occurs permanently, please contact Lenze.

EtherCAT: Internal error [0x00c86101]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
Internal error.	Send communication module together with a description of the fault to Lenze.

EtherCAT: Internal mapping error [0x00c86110]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
The selected PDO mapping is invalid: • An object not supporting the required properties has been specified. • More than 64 bytes of data have been entered/mapped.	Re-execute PDO mapping. ▶ Configuring process data (PDO mapping) (62)

EtherCAT: Parameter set invalid [0x00c8641f]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
No active parameter set could be loaded.	Download application again (including module).

EtherCAT: Error: Lenze setting loaded [0x00c86420]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☒ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
Access via standard device to the parameter set in the memory module failed.	Download application again (including module).

EtherCAT: Communication time-out with safety module [0x00c86610]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
The communication to the SM30x safety module is disturbed.	Send communication module with error description to Lenze.

EtherCAT: Initialisation error with safety module [0x00c86612]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
The communication to the SM30x safety module could not be initialised correctly.	Check if the SM30x safety module is plugged in and supplied with voltage

EtherCAT: Quit 'Operational' state [0x00c88131]

Response (Lenze setting printed in bold)	Setting: C13880/1 / C14880/1 (☑ Configurable response)
☒ None ☐ System fault ☒ Fault ☐ Trouble ☒ Quick stop by trouble ☒ Warning locked ☐ Warning ☒ Information	
Cause	Remedy
The EtherCAT data exchange was stopped in the "Operational" state. Also see the chapter " Interruption of EtherCAT communication " (□ 82).	• Check cables and terminals. • The master has to reset the device to the "Operational" status. (If required, check a pending emergency message first).

EtherCAT: Sync loss detected with standard device [0x00c88132]

Response (Lenze setting printed in bold)	Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information	
Cause	Remedy
• The cycle time settings are faulty. • The task time setting of a linked PLC program is faulty. • EtherCAT master / DC master settings are incorrect. • The sync source setting is faulty.	• Check the cycle times. • Check the task time of a linked PLC program. • Check the EtherCAT master / DC master settings. • Check the settings of the standard device in C01120 (selection of the sync source).

EtherCAT: Sync telegram failure [0x00c88700]

Response (Lenze setting printed in bold)		Setting: C13880/2 / C14880/2 (☒ Configurable response)	
☒ None ☐ System fault ☒ Fault ☐ Trouble ☒ Quick stop by trouble ☒ Warning locked ☐ Warning ☒ Information			
Cause		Remedy	
• EtherCAT-Sync0 signal has failed. • PDO sent or received too late as the instant of transmission of the EtherCAT frame by the master jitters too much.		• Check EtherCAT master configuration if the DC mode has been set. • Restart EtherCAT configuration. • Postpone PDO instant of reception via the user shift time. • Increase sync error counter monitoring limit (0x10F1.2).	

14 Parameter reference

This chapter supplements the parameter list and the table of attributes contained in the reference manual and in the »Engineer« online help for the Servo Drive 9400 with the parameters of the E94AYCET (EtherCAT) communication module.



Reference manual/»Engineer« online help for the Servo Drive 9400

Here you will find general information on parameters.

14.1 Parameters of the standard device that are relevant to communication

In this chapter communication-relevant parameters of the Servo Drive 9400 are listed in numerically ascending order.

C00615

Parameter Name:		Data type: UNSIGNED_32 Index: 23960 = 0x5D98
C00615 Resp. to imp. device conf.		
Response to impermissible device configuration		
Selection list		
1	Fault	
3	Quick stop by trouble	
4	Warning Locked	
6	Information	
0	No response	
Subcodes	Lenze setting	Info
C00615/1	0: No Response	Reserved
C00615/2	1: No Response	Resp. to imp. module in MXI1
C00615/3	1: No Response	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		

C00636

Parameter Name:		Data type: UNSIGNED_32 Index: 23939 = 0x5D83
C00636 Resp. to new module in MXI1		
Response for the case where a new module has been plugged into slot 1 of the standard device.		
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		

C00637

Parameter Name: C00637 Resp. to new module in MXI2		Data type: UNSIGNED_32 Index: 23939 = 0x5D83
Response if a new module has been plugged into module slot 2 of the standard device.		
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		

C01120

Parameter Name: C01120 Sync source		Data type: UNSIGNED_8 Index: 23455 = 0x5B9F
Selection of the source for DC synchronisation signals. - Depending on the slot in the standard device, set the selection "4: Module in MXI1" or "5: Module in MXI2" for the communication module. - The drive can only be synchronised by one source. ▶ Synchronisation with "Distributed clocks" (DC) (□ 36)		
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	
...	...	
13	Digital input 8	
☒ Read access ☒ Write access ☐ RSP ☐ PLC-STOP ☐ No transfer Scaling factor: 1		

C01501

Parameter Name: C01501 Resp. to comm. error with MXI1		Data type: UNSIGNED_32 Index: 23074 = 0x5A22
Response to a communication error between an "intelligent" module in slot 1 and the 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		

C01502

Parameter Name: C01502 Resp. to comm. error with MXI2		Data type: UNSIGNED_32 Index: 23074 = 0x5A22
Response to a communication error between an "intelligent" module in slot 2 and the 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		

14.2

Parameters of the communication module for slot MXI1

This chapter lists the parameters of the E94AYCET communication module (EtherCAT) for slot MXI2 of the Servo Drive 9400 in numerically ascending order.

C13231

Parameter Name: C13231 Mapping entries RPDO-1 (I-1600)		Data type: UNSIGNED_64 Index: 11344 = 0x2C50
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 1. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13231/1	0	1. Mapping entry RPDO-1 (I-1600.01)
...	0	...
C13231/32	0	32. Mapping entry RPDO-1 (I-1600.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13232

Parameter Name: C13232 Mapping entries RPDO-2		Data type: UNSIGNED_64 Index: 11343 = 0x2C4F
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 2. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13232/1	0	1. Mapping entry RPDO-2 (I-1601.01)
...	0	...
C13232/32	0	32. Mapping entry RPDO-2 (I-1601.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13233

Parameter Name: C13233 Mapping entries RPDO-3		Data type: UNSIGNED_64 Index: 11342 = 0x2C4E
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 3. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13233/1	0	1. Mapping entry RPDO-3 (I-1602.01)
...	...	...
C13233/32	0	32. Mapping entry RPDO-3 (I-1602.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13234

Parameter Name: C13234 Mapping entries RPDO-4		Data type: UNSIGNED_64 Index: 11341 = 0x2C4D
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 4. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13234/1	0	1. Mapping entry RPDO-4 (I-1603.01)
...	0	...
C13234/32	0	32. Mapping entry RPDO-4 (I-1603.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13235

Parameter Name: C13235 Mapping entries RPDO-5		Data type: UNSIGNED_64 Index: 11340 = 0x2C4C
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 5. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13235/1	0	1. Mapping entry RPDO-5 (I-1604.01)
...	...	...
C13235/32	0	32. Mapping entry RPDO-5 (I-1604.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13236

Parameter Name: C13236 Mapping entries RPDO-6		Data type: UNSIGNED_64 Index: 11339 = 0x2C4B
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 6. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13236/1	0	1. Mapping entry RPDO-6 (I-1605.01)
...	...	...
C13236/32	0	32. Mapping entry RPDO-6 (I-1605.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13237

Parameter Name: C13237 Mapping entries RPDO-7		Data type: UNSIGNED_64 Index: 11338 = 0x2C4A
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 7. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13237/1	0	1. Mapping entry RPDO-7 (I-1606.01)
...	...	...
C13237/32	0	32. Mapping entry RPDO-7 (I-1606.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13238

Parameter Name: C13238 Mapping entries RPDO-8		Data type: UNSIGNED_64 Index: 11337 = 0x2C49
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 8. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13238/1	0	1. Mapping entry RPDO-8 (I-1607.01)
...	...	...
C13238/32	0	32. Mapping entry RPDO-8 (I-1607.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13239

Parameter Name: C13239 Mapping entries RPDO-9		Data type: UNSIGNED_64 Index: 11336 = 0x2C48
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 9. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13239/1	0	1. Mapping entry RPDO-9 (I-1608.01)
...	...	...
C13239/32	0	32. Mapping entry RPDO-9 (I-1608.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13240

Parameter Name: C13240 Mapping entries RPDO-10		Data type: UNSIGNED_64 Index: 11335 = 0x2C47
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 10. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13240/1	0	1. Mapping entry RPDO-10 (I-1609.01)
...	0	...
C13240/32	0	32. Mapping entry RPDO-10 (I-1609.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13484

Parameter Name: C13484 Active RPDO channel		Data type: UNSIGNED_8 Index: 11091 = 0x2B53
This code is for device-internal use only and must not be written to by the user!		
Selection of the active RPDO channel		
Selection list		Info
0	No RPDO	
1	RPDO-1 (I-1600)	
...	...	
10	RPDO-10 (I-1609)	
Subcodes	Lenze setting	Info
C13484/1	1: RPDO-1 (I-1600)	Active RPDO channel
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13489

Parameter Name: C13489 No. of RPDO mapping entries		Data type: UNSIGNED_8 Index: 11086 = 0x2B4E
This code is for device-internal use only and must not be written to by the user!		
Subcodes 1 ... 10 specify the number of valid mapping entries of the RPDO channels 1 ... 10 (valid subcodes in C13231 ... C13240).		
Setting range (min. value unit max. value)		
0		32
Subcodes	Lenze setting	Info
C13489/1	0	No. of mapping entries for RPDO-1
...	...	...
C13489/10	0	No. of mapping entries for RPDO-10
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13531

Parameter Name: C13531 Mapping entries TPDO-1		Data type: UNSIGNED_64 Index: 11044 = 0x2B24
This code is for device-internal use only and must not be written to by the user!		
Subcodes 1 ... 32 specify the mapping for the TPDO channel 1.		
- Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. - The master must have the same mapping entries as the slave.		
▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13531/1	0	1. mapping entry TPDO-1 (I-1A00.01)
...	...	...
C13531/32	0	32. mapping entry TPDO-1 (I-1A00.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13532

Parameter Name: C13532 Mapping entries TPDO-2		Data type: UNSIGNED_64 Index: 11043 = 0x2B23
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 2. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13532/1	0	1. Mapping entry TPDO-2 (I-1A01.01)
...	...	...
C13532/32	0	32. Mapping entry TPDO-2 (I-1A01.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13533

Parameter Name: C13533 Mapping entries TPDO-3		Data type: UNSIGNED_64 Index: 11042 = 0x2B22
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 3. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13533/1	0	1. Mapping entry TPDO-3 (I-1A02.01)
...	...	...
C13533/32	0	32. Mapping entry TPDO-3 (I-1A02.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13534

Parameter Name: C13534 Mapping entries TPDO-4		Data type: UNSIGNED_64 Index: 11041 = 0x2B21
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 4. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13534/1	0	1. Mapping entry TPDO-4 (I-1A03.01)
...	...	...
C13534/32	0	32. Mapping entry TPDO-4 (I-1A03.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13535

Parameter Name: C13535 Mapping entries TPDO-5		Data type: UNSIGNED_64 Index: 11040 = 0x2B20
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 5. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13535/1	0	1. Mapping entry TPDO-5 (I-1A04.01)
...	...	...
C13535/32	0	32. Mapping entry TPDO-5 (I-1A04.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13536

Parameter Name: C13536 Mapping entries TPDO-6		Data type: UNSIGNED_64 Index: 11039 = 0x2B1F
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 6. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13536/1	0	1. Mapping entry TPDO-6 (I-1A05.01)
...	...	...
C13536/32	0	32. Mapping entry TPDO-6 (I-1A05.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13537

Parameter Name: C13537 Mapping entries TPDO-7		Data type: UNSIGNED_64 Index: 11038 = 0x2B1E
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 7. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13537/1	0	1. Mapping entry TPDO-7 (I-1A06.01)
...	0	...
C13537/32	0	32. Mapping entry TPDO-7 (I-1A06.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13538

Parameter Name: C13538 Mapping entries TPDO-8		Data type: UNSIGNED_64 Index: 11037 = 0x2B1D
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 8. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13538/1	0	1. Mapping entry TPDO-8 (I-1A07.01)
...	...	...
C13538/32	0	32. Mapping entry TPDO-8 (I-1A07.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13539

Parameter Name: C13539 Mapping entries TPDO-9		Data type: UNSIGNED_64 Index: 11036 = 0x2B1C
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 9. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13539/1	0	1. Mapping entry TPDO-9 (I-1A08.01)
...	...	...
C13539/32	0	32. Mapping entry TPDO-9 (I-1A08.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13540

Parameter Name: C13540 Mapping entries TPDO-10		Data type: UNSIGNED_64 Index: 11035 = 0x2B1B
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 10. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C13540/1	0	1. Mapping entry TPDO-10 (I-1A09.01)
...	...	...
C13540/32	0	32. Mapping entry TPDO-10 (I-1A09.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13784

Parameter Name: C13784 Active TPDO channel		Data type: UNSIGNED_8 Index: 10791 = 0x2A27
This code is for device-internal use only and must not be written to by the user! Selection of the active TPDO channel		
Selection list		Info
0	No TPDO	
1	TPDO-1 (I-1A00)	
...	...	
10	TPDO-10 (I-1A09)	
Subcodes	Lenze setting	Info
C13784/1	1: TPDO-1 (I-1A00)	Active TPDO channel
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13789

Parameter Name: C13789 No. of TPDO mapping entries		Data type: UNSIGNED_8 Index: 10786 = 0x2A22
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 10 specify the number of valid mapping entries of the TPDO channels 1 ... 10 (valid subcodes in C13531 ... C13540).		
Setting range (min. value unit max. value)		
0		32
Subcodes	Lenze setting	Info
C13789/1	0	No. of mapping entries for TPDO-1
...	...	...
C13789/10	0	No. of mapping entries for TPDO-10
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13850

Parameter Name: C13850 All words to master			Data type: UNSIGNED_16 Index: 10725 = 0x29E5
Display of the process data words (subcodes 1 ... 32) which are transferred from the communication module to the master. Only those which are configured are valid.			
Display range (min. value unit max. value)			
0		65535	
Subcodes			Info
C13850/1			
...			
C13850/32			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13851

Parameter Name: C13851 All words from master			Data type: UNSIGNED_16 Index: 10724 = 0x29E4
Display of the process data words (subcodes 1 ... 32) which are transferred from the master to the communication module. Only those which are configured are valid.			
Display range (min. value unit max. value)			
0		65535	
Subcodes			Info
C13851/1			
...			
C13851/32			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13852

Parameter Name: C13852 All words to standard device			Data type: UNSIGNED_16 Index: 10723 = 0x29E3
Display of the process data words (subcodes 1 ... 32) which are transferred from the communication module to the standard device.			
Display range (min. value unit max. value)			
0		65535	
Subcodes			Info
C13852/1			
...			
C13852/32			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13853

Parameter Name: C13853 All words from standard device			Data type: UNSIGNED_16 Index: 10722 = 0x29E2	
Display of the process data words (subcodes 1 ... 32) which are transferred from the standard device to the communication module.				
Display range (min. value unit max. value)				
0		65535		
Subcodes			Info	
C13853/1				
...				
C13853/32				
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT				

C13858

Parameter Name: C13858 Safety-over-EtherCAT (FSOE) active		Data type: UNSIGNED_8 Index: 10717 = 0x29DD
Available from SW version 04.00. Display of the activity of Safety over EtherCAT (🔒 81)		
Selection list (read only)		Info
0	Not active	
1	Enable	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13859

Parameter Name: C13859 Length of the configured input data			Data type: UNSIGNED_8 Index: 10716 = 0x29DC
Available from SW version 04.00. Display of the length of the configured input data in bytes			
Display range (min. value unit max. value)			
0		255	
Subcodes			Info
C13859/1			TxPDO length
C13859/2			Safety-TxPDO length
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13860

Parameter Name: C13860 Length of the configured output data			Data type: UNSIGNED_8 Index: 10715 = 0x29DB		
Available from SW version 04.00. Display of the length of the configured output data in bytes					
Display range (min. value unit max. value)					
0		255			
Subcodes			Info		
C13859/1			RxPDO length		
C13859/2			Safety-RxPDO length		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT					

C13861

Parameter Name: C13861 Bus status		Data type: UNSIGNED_16 Index: 10714 = 0x29DA
Display of the current bus status ▶ EtherCAT state machine (📘 59)		
Selection list (read only)		Info
0	Nonexistent	No online value available (communication module is offline).
1	Init	
2	Pre-Operational	
3	Bootstrap	Is not supported.
4	Safe-operational	
8	Operational	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13864

Parameter Name: C13864 Active station address		Data type: UNSIGNED_16 Index: 10711 = 0x29D7
Display of the station address allocated by the master		
Display range (min. value unit max. value)		
0		65535
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13867

Parameter Name: C13867 Display: Most recent emergency data		Data type: OCTET_STRING Index: 10708 = 0x29D4
Display of the emergency data last sent by the inverter (string with a length of 8 bytes). ▶ Emergency requests / Emergency messages (📘 91)		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13870

Parameter Name: C13870 Cycle time display		Data type: UNSIGNED_32 Index: 10705 = 0x29D1
Available from software version 03.00! Display of the synchronisation cycle time in μ s set by the master. • This time is only valid if "Distributed clocks" (DC) are used. • If no DC is used, the time is displayed is "0". ▶ Synchronisation with "Distributed clocks" (DC) (📘 36)		
Display range (min. value unit max. value)		
0	μ s	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13879

Parameter Name: C13879 Bus error		Data type: UNSIGNED_32 Index: 10696 = 0x29C8
Bit-coded display of the bus error Additionally, an error message is entered in the EtherCAT register " AL Status Code " (□ 60).		
Value is bit-coded:		Info
Bit 0	General bus error	
Bit 1	Reserved	
...	...	
Bit 31	Reserved	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13880

Parameter Name: C13880 Resp. to communication failure		Data type: UNSIGNED_8 Index: 10695 = 0x29C7
The set response will be executed if ... • a sync telegram failure has been detected <u>or</u> • the device detects that is no longer in the "Operational" state and the monitoring time (C13881) has elapsed.		
Selection list		
0	No response	
1	Fault	
3	Quick stop by trouble	
4	Warning Locked	
6	Information	
Subcodes	Lenze setting	Info
C13880/1	0: No Response	Resp. to Interruption of EtherCAT communication (□ 82)
C13880/2	0: No Response	Resp. to Sync telegram failure detection (□ 83) (available from SW version 04.00)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13881

Parameter Name: C13881 Response time when exiting 'Operational'		Data type: UNSIGNED_16 Index: 10694 = 0x29C6
If the "Operational" state is exited, the response parameterised with C13880 occurs after the time set here has elapsed. A change of monitoring will be effective immediately.		
Setting range (min. value unit max. value)		Lenze setting
0	ms	65535 0 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13882

Parameter Name: C13882 Display of PDO watchdog time		Data type: UNSIGNED_32 Index: 10693 = 0x29C5
Available from software version 03.00! Display of the PDO watchdog monitoring time defined in the master. During initialisation of the EtherCAT communication, the watchdog monitoring time is transmitted to the slave.		
Display range (min. value unit max. value)		
0	µs	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13883

Parameter Name: C13883 DC active		Data type: UNSIGNED_8 Index: 10692 = 0x29C4
The code displays if the DC synchronisation has been activated for the communication module. ► Synchronisation with "Distributed clocks" (DC) (□ 36)		
Selection list (read only)		Info
0	DC unused	
1	DC for synchronisation	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13884

Parameter Name: C13884 Base device synchronisation status		Data type: UNSIGNED_8 Index: 10691 = 0x29C3
Available from software version 03.00! The code displays whether the inverter is DC-synchronised. ► Synchronisation with "Distributed clocks" (DC) (□ 36)		
Selection list (read only)		Info
0	Not synchronised	
1	Synchronised	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13885

Parameter Name: C13885 Clear process data		Data type: UNSIGNED_8 Index: 10690 = 0x29C2
This code serves to set which process data the slave is to process for maintaining internal communication when the EtherCAT has exited the "Operational" state.		
Selection list (Lenze setting printed in bold)		Info
0	Use of most recent master PDOs	
1	PDOs are set to the value 0'	
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13887

Parameter Name: C13887 Suppress emergency message in case of		Data type: BITFIELD_8 Index: 10688 = 0x29C0
This code serves to prevent the emergency messages from being transmitted to the EtherCAT master. Here, errors of a certain type can be suppressed deliberately. Furthermore, all errors are entered in the logbook. A change will only be effective if no error number with the error type selected here is available in the standard device code C00165.		
Value is bit-coded:		Info
Bit 0	Fault	
Bit 1	Fault	
Bit 2	Quick stop by trouble	
Bit 3	Warning Locked	
Bit 4	Warning	
Bit 5	Information	
Bit 6	Reserved	
Bit 7	Reserved	
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13892

Parameter Name: C13892 Process data mode		Data type: UNSIGNED_8 Index: 10683 = 0x29BB
Mode selection for a time optimisation of the process data transfer This code is only evaluated if "Distributed clocks" are used. ► Synchronisation with "Distributed clocks" (DC) (□ 36) ► Process data mode (□ 40)		
Selection list (Lenze setting printed in bold)		Info
0	Optimised mode	This mode leads to a reduction of the dead time by one EtherCAT cycle. Preconditions: • The EtherCAT telegram generated by the master has a low jitter and is sent within a window of 320 µs after the Sync0 event and shortly before the next Sync0 event. • The EtherCAT cycle time must be 1 ms.
1	Deterministic mode	This mode is the standard mode. It does not support a time optimisation of the process data transfer. The actual values are copied to the EtherCAT interface with the next Sync0 event.
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13899

Parameter Name: C13899 Station alias		Data type: UNSIGNED_16 Index: 10676 = 0x29B4
This code serves to set a station alias. In order to use a station alias, you must select a value > "0". • The station alias must only be set if the device is part of a "hot connect" group. • The station alias must be unambiguous and must only be assigned once within the EtherCAT network. • Use the same station alias in the EtherCAT master and in the slave. ► Address allocation (□ 35)		
Setting range (min. value unit max. value)		Lenze setting
0		32767 0
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C13900

Parameter Name: C13900 Firmware: Product type	Data type: VISIBLE_STRING Index: 10675 = 0x29B3
The code contains a string with a length of 8 bytes. The following identification code is output: "E94AFCET".	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13901

Parameter Name: C13901 Firmware compilation date	Data type: VISIBLE_STRING Index: 10674 = 0x29B2
The code contains a string with a length of 20 bytes. The creation date ("MM DD YYYY") and the time ("hh:mm:ss") of the software are output here. Example: "Mar 21 2005 12:31:21"	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13902

Parameter Name: C13902 Firmware version	Data type: VISIBLE_STRING Index: 10673 = 0x29B1
The code contains a string with a length of 11 bytes. The firmware version is output here. Example: "02.00.00.36"	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

14.3

Parameters of the communication module for slot MXI2

This chapter lists the parameters of the E94AYCET communication module (EtherCAT) for slot MXI2 of the Servo Drive 9400 in numerically ascending order.

C14231

Parameter Name: C14231 Mapping entries RPDO-1 (I-1600)		Data type: UNSIGNED_64 Index: 10344 = 0x2868
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 1. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14231/1	0	1. Mapping entry RPDO-1 (I-1600.01)
...	...	...
C14231/32	0	32. Mapping entry RPDO-1 (I-1600.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14232

Parameter Name: C14232 Mapping entries RPDO-2		Data type: UNSIGNED_64 Index: 10343 = 0x2867
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 2. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14232/1	0	1. Mapping entry RPDO-2 (I-1601.01)
...	...	...
C14232/32	0	32. Mapping entry RPDO-2 (I-1601.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14233

Parameter Name: C14233 Mapping entries RPDO-3		Data type: UNSIGNED_64 Index: 10342 = 0x2866
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 3. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14233/1	0	1. Mapping entry RPDO-3 (I-1602.01)
...	...	...
C14233/32	0	32. Mapping entry RPDO-3 (I-1602.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14234

Parameter Name: C14234 Mapping entries RPDO-4		Data type: UNSIGNED_64 Index: 10341 = 0x2865
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 4. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14234/1	0	1. Mapping entry RPDO-4 (I-1603.01)
...	...	...
C14234/32	0	32. Mapping entry RPDO-4 (I-1603.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14235

Parameter Name: C14235 Mapping entries RPDO-5		Data type: UNSIGNED_64 Index: 10340 = 0x2864
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 5. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14235/1	0	1. Mapping entry RPDO-5 (I-1604.01)
...	...	...
C14235/32	0	32. Mapping entry RPDO-5 (I-1604.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14236

Parameter Name: C14236 Mapping entries RPDO-6		Data type: UNSIGNED_64 Index: 10339 = 0x2863
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 6. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14236/1	0	1. Mapping entry RPDO-6 (I-1605.01)
...	...	...
C14236/32	0	32. Mapping entry RPDO-6 (I-1605.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14237

Parameter Name: C14237 Mapping entries RPDO-7		Data type: UNSIGNED_64 Index: 10338 = 0x2862
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 7. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14237/1	0	1. Mapping entry RPDO-7 (I-1606.01)
...	...	...
C14237/32	0	32. Mapping entry RPDO-7 (I-1606.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14238

Parameter Name: C14238 Mapping entries RPDO-8		Data type: UNSIGNED_64 Index: 10337 = 0x2861
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 8. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14238/1	0	1. Mapping entry RPDO-8 (I-1607.01)
...	...	...
C14238/32	0	32. Mapping entry RPDO-8 (I-1607.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14239

Parameter Name: C14239 Mapping entries RPDO-9		Data type: UNSIGNED_64 Index: 10336 = 0x2860
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 9. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14239/1	0	1. Mapping entry RPDO-9 (I-1608.01)
...	...	...
C14239/32	0	32. Mapping entry RPDO-9 (I-1608.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14240

Parameter Name: C14240 Mapping entries RPDO-10		Data type: UNSIGNED_64 Index: 10335 = 0x285F
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the RPDO channel 10. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14240/1	0	1. Mapping entry RPDO-10 (I-1609.01)
...	...	...
C14240/32	0	32. Mapping entry RPDO-10 (I-1609.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14484

Parameter Name: C14484 Active RPDO channel		Data type: UNSIGNED_8 Index: 10091 = 0x276B
This code is for device-internal use only and must not be written to by the user!		
Selection of the active RPDO channel		
Selection list		Info
0	No RPDO	
1	RPDO-1 (I-1600)	
...	...	
10	RPDO-10 (I-1609)	
Subcodes	Lenze setting	Info
C14484/1	1: RPDO-1 (I-1600)	Active RPDO channel
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14489

Parameter Name: C14489 No. of RPDO mapping entries		Data type: UNSIGNED_8 Index: 10086 = 0x2766
This code is for device-internal use only and must not be written to by the user!		
Subcodes 1 ... 10 specify the number of valid mapping entries of RPDO channels 1 ... 10 (valid subcodes in C14231 ... C14240).		
Setting range (min. value unit max. value)		
0		32
Subcodes	Lenze setting	Info
C14489/1	0	No. of mapping entries for RPDO-1
...	0	...
C14489/10	0	No. of mapping entries for RPDO-10
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14531

Parameter Name: C14531 Mapping entries TPDO-1		Data type: UNSIGNED_64 Index: 10044 = 0x273C
This code is for device-internal use only and must not be written to by the user!		
Subcodes 1 ... 32 specify the mapping for the TPDO channel 1.		
- Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. - The master must have the same mapping entries as the slave.		
▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14531/1	0	1. mapping entry TPDO-1 (I-1A00.01)
...	...	...
C14531/32	0	32. mapping entry TPDO-1 (I-1A00.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14532

Parameter Name: C14532 Mapping entries TPDO-2		Data type: UNSIGNED_64 Index: 10043 = 0x273B
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 2. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14532/1	0	1. Mapping entry TPDO-2 (I-1A01.01)
...	...	...
C14532/32	0	32. Mapping entry TPDO-2 (I-1A01.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14533

Parameter Name: C14533 Mapping entries TPDO-3		Data type: UNSIGNED_64 Index: 10042 = 0x273A
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 3. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14533/1	0	1. Mapping entry TPDO-3 (I-1A02.01)
...	...	...
C14533/32	0	32. Mapping entry TPDO-3 (I-1A02.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14534

Parameter Name: C14534 Mapping entries TPDO-4		Data type: UNSIGNED_64 Index: 10041 = 0x2739
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 4. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14534/1	0	1. Mapping entry TPDO-4 (I-1A03.01)
...	...	...
C14534/32	0	32. Mapping entry TPDO-4 (I-1A03.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14535

Parameter Name: C14535 Mapping entries TPDO-5		Data type: UNSIGNED_64 Index: 10040 = 0x2738
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 5. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14535/1	0	1. Mapping entry TPDO-5 (I-1A04.01)
...	...	...
C14535/32	0	32. Mapping entry TPDO-5 (I-1A04.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14536

Parameter Name: C14536 Mapping entries TPDO-6		Data type: UNSIGNED_64 Index: 10039 = 0x2737
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 6. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14536/1	0	1. Mapping entry TPDO-6 (I-1A05.01)
...	...	...
C14536/32	0	32. Mapping entry TPDO-6 (I-1A05.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14537

Parameter Name: C14537 Mapping entries TPDO-7		Data type: UNSIGNED_64 Index: 10038 = 0x2736
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 7. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14537/1	0	1. Mapping entry TPDO-7 (I-1A06.01)
...	...	...
C14537/32	0	32. Mapping entry TPDO-7 (I-1A06.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14538

Parameter Name: C14538 Mapping entries TPDO-8		Data type: UNSIGNED_64 Index: 10037 = 0x2735
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 8. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14538/1	0	1. Mapping entry TPDO-8 (I-1A07.01)
...	...	...
C14538/32	0	32. Mapping entry TPDO-8 (I-1A07.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14539

Parameter Name: C14539 Mapping entries TPDO-9		Data type: UNSIGNED_64 Index: 10036 = 0x2734
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 9. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14539/1	0	1. Mapping entry TPDO-9 (I-1A08.01)
...	...	...
C14539/32	0	32. Mapping entry TPDO-9 (I-1A08.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14540

Parameter Name: C14540 Mapping entries TPDO-10		Data type: UNSIGNED_64 Index: 10035 = 0x2733
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 32 specify the mapping for the TPDO channel 10. • Servo Drives 9400 exclusively support mapping objects with a data length of 8 bits (1 byte) or a multiple of this value. Incompatible objects must be mapped to variables with a compatible data length. • The master must have the same mapping entries as the slave. ▶ Structure of the mapping codes (□ 66)		
Value is bit-coded:		Info
Bit 0	Bit 0	
...	...	
Bit 63	Bit 63	
Subcodes	Lenze setting	Info
C14540/1	0	1. Mapping entry TPDO-10 (I-1A09.01)
...	...	...
C14540/32	0	32. Mapping entry TPDO-10 (I-1A09.20)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14784

Parameter Name: C14784 Active TPDO channel		Data type: UNSIGNED_8 Index: 9791 = 0x263F
This code is for device-internal use only and must not be written to by the user! Selection of the active TPDO channel		
Selection list		Info
0	No TPDO	
1	TPDO-1 (I-1A00)	
...	...	
10	TPDO-10 (I-1A09)	
Subcodes	Lenze setting	Info
C14784/1	1: TPDO-1 (I-1A00)	Active TPDO channel
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14789

Parameter Name: C14789 No. of TPDO mapping entries		Data type: UNSIGNED_8 Index: 9786 = 0x263A
This code is for device-internal use only and must not be written to by the user! Subcodes 1 ... 10 specify the number of valid mapping entries of TPDO channels 1 ... 10 (valid subcodes in C14531 ... C14540).		
Setting range (min. value unit max. value)		
0		32
Subcodes	Lenze setting	Info
C14789/1	0	No. of mapping entries for TPDO-1
...	...	...
C14789/10	0	No. of mapping entries for TPDO-10
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C14850

Parameter Name: C14850 All words to master			Data type: UNSIGNED_16 Index: 9725 = 0x25FD
Display of the process data words (subcodes 1 ... 32) which are transferred from the communication module to the master. Only those which are configured are valid.			
Display range (min. value unit max. value)			
0		65535	
Subcodes			Info
C14850/1			
...			
C14850/32			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14851

Parameter Name: C14851 All words from master			Data type: UNSIGNED_16 Index: 9724 = 0x25FC
Display of the process data words (subcodes 1 ... 32) which are transferred from the master to the communication module. Only those which are configured are valid.			
Display range (min. value unit max. value)			
0		65535	
Subcodes			Info
C14851/1			
...			
C14851/32			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14852

Parameter Name: C14852 All words to standard device			Data type: UNSIGNED_16 Index: 9723 = 0x25FB
Display of the process data words (subcodes 1 ... 32) which are transferred from the communication module to the standard device.			
Display range (min. value unit max. value)			
0		65535	
Subcodes			Info
C14852/1			
...			
C14852/32			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14853

Parameter Name: C14853 All words from standard device			Data type: UNSIGNED_16 Index: 9722 = 0x25FA	
Display of the process data words (subcodes 1 ... 32) which are transferred from the standard device to the communication module.				
Display range (min. value unit max. value)				
0		65535		
Subcodes			Info	
C14853/1				
...				
C14853/32				
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT				

C14858

Parameter Name: C14858 Safety-over-EtherCAT (FSOE) active		Data type: UNSIGNED_8 Index: 9717 = 0x25F5
Available from SW version 04.00. Display of the activity of Safety over EtherCAT (🔍 81)		
Selection list (read only)		Info
0	Not active	
1	Enable	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14859

Parameter Name: C14859 Length of the configured input data			Data type: UNSIGNED_8 Index: 9716 = 0x25F4
Available from SW version 04.00.			
Display of the length of the configured input data in bytes			
Display range (min. value unit max. value)			
0		255	
Subcodes			Info
C14859/1			TxPDO length
C14859/2			Safety-TxPDO length
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14860

Parameter Name: C14860 Length of the configured output data			Data type: UNSIGNED_8 Index: 9715 = 0x25F3	
Available from SW version 04.00. Display of the length of the configured output data in bytes				
Display range (min. value unit max. value)				
0		255		
Subcodes			Info	
C13859/1			RxPDO length	
C13859/2			Safety-RxPDO length	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT				

C14861

Parameter Name: C14861 Bus status		Data type: UNSIGNED_16 Index: 9714 = 0x25F2
Display of the current bus status ► EtherCAT state machine (📖 59)		
Selection list (read only)		Info
0	Nonexistent	No online value available (communication module is offline).
1	Init	
2	Pre-Operational	
3	Bootstrap	Is not supported.
4	Safe-operational	
8	Operational	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14864

Parameter Name: C14864 Active station address		Data type: UNSIGNED_16 Index: 9711 = 0x25EF
Display of the station address allocated by the master		
Display range (min. value unit max. value)		
0		65535
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14867

Parameter Name: C14867 Display: Most recent emergency data		Data type: OCTET_STRING Index: 9708 = 0x25EC
Display of the emergency data last sent by the inverter (string with a length of 8 bytes). ► Emergency requests / Emergency messages (📖 91)		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14870

Parameter Name: C14870 Cycle time display		Data type: UNSIGNED_32 Index: 9705 = 0x25E9
Available from software version 03.00! Display of the synchronisation cycle time in μ s set by the master. • This time is only valid if "Distributed clocks" (DC) are used. • If no DC is used, the time is displayed is "0". ► Synchronisation with "Distributed clocks" (DC) (📖 36)		
Display range (min. value unit max. value)		
0	μ s	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14879

Parameter Name: C14879 Bus error		Data type: UNSIGNED_32 Index: 9696 = 0x25E0
Bit-coded display of the bus error Additionally, an error message is entered in the EtherCAT register " AL Status Code " (□ 60).		
Value is bit-coded:		Info
Bit 0	General bus error	
Bit 1	Reserved	
...	...	
Bit 31	Reserved	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14880

Parameter Name: C14880 Resp. to communication failure		Data type: UNSIGNED_8 Index: 9695 = 0x25DF
The set response will be executed if ... • a sync telegram failure has been detected <u>or</u> • the device detects that is no longer in the "Operational" state and the monitoring time (C13881) has elapsed.		
Selection list		
0	No response	
1	Fault	
3	Quick stop by trouble	
4	Warning Locked	
6	Information	
Subcodes	Lenze setting	Info
C13880/1	0: No Response	Resp. to Interruption of EtherCAT communication (□ 82)
C14880/2	0: No Response	Resp. to Sync telegram failure detection (□ 83) (available from SW version 04.00)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14881

Parameter Name: C14881 Response time when exiting 'Operational'		Data type: UNSIGNED_16 Index: 9694 = 0x25DE
If the "Operational" state is exited, the response parameterised with C14880 occurs after the time set here has elapsed. A change of monitoring will be effective immediately.		
Setting range (min. value unit max. value)		Lenze setting
0	ms	65535 0 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14882

Parameter Name: C14882 Display of PDO watchdog time		Data type: UNSIGNED_32 Index: 9693 = 0x25DD
Available from software version 03.00! Display of the PDO watchdog monitoring time defined in the master. During initialisation of the EtherCAT communication, the watchdog monitoring time is transmitted to the slave.		
Display range (min. value unit max. value)		
0	µs	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14883

Parameter Name: C14883 DC active		Data type: UNSIGNED_8 Index: 9692 = 0x25DC
The code displays if the DC synchronisation has been activated for the communication module. ► Synchronisation with "Distributed clocks" (DC) (□ 36)		
Selection list (read only)		Info
0	DC unused	
1	DC for synchronisation	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14884

Parameter Name: C14884 Base device synchronisation status		Data type: UNSIGNED_8 Index: 9691 = 0x25DB
Available from software version 03.00! The code displays whether the inverter is DC-synchronised. ► Synchronisation with "Distributed clocks" (DC) (□ 36)		
Selection list (read only)		Info
0	Not synchronised	
1	Synchronised	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14885

Parameter Name: C14885 Clear process data		Data type: UNSIGNED_8 Index: 9690 = 0x25DA
This code serves to set which process data the slave is to process for maintaining internal communication when the EtherCAT has exited the "Operational" state.		
Selection list (Lenze setting printed in bold)		Info
0	Use of most recent master PDOs	
1	PDOs are set to the value 0'	
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14887

Parameter Name: C14887 Suppress emergency message in case of		Data type: BITFIELD_8 Index: 9688 = 0x25D8
This code serves to prevent the emergency messages from being transmitted to the EtherCAT master. Here, errors of a certain type can be suppressed deliberately. Furthermore, all errors are entered in the logbook. A change will only be effective if no error number with the error type selected here is available in the standard device code C00165.		
Value is bit-coded:		Info
Bit 0	Fault	
Bit 1	Fault	
Bit 2	Quick stop by trouble	
Bit 3	Warning Locked	
Bit 4	Warning	
Bit 5	Information	
Bit 6	Reserved	
Bit 7	Reserved	
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14892

Parameter Name: C14892 Process data mode		Data type: UNSIGNED_8 Index: 9683 = 0x25D3
Mode selection for a time optimisation of the process data transfer This code is only evaluated if "Distributed clocks" are used. ► Synchronisation with "Distributed clocks" (DC) (□ 36) ► Process data mode (□ 40)		
Selection list (Lenze setting printed in bold)		Info
0	Optimised mode	This mode leads to a reduction of the dead time by one EtherCAT cycle. Preconditions: • The EtherCAT telegram generated by the master has a low jitter and is sent within a window of 320 µs after the Sync0 event and shortly before the next Sync0 event. • The EtherCAT cycle time must be 1 ms.
1	Deterministic mode	This mode is the standard mode. It does not support a time optimisation of the process data transfer. The actual values are copied to the EtherCAT interface with the next Sync0 event.
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14899

Parameter Name: C14899 Station alias			Data type: UNSIGNED_16 Index: 9676 = 0x25CC
This code serves to set a station alias. In order to use a station alias, you must select a value > "0". • The station alias must only be set if the device is part of a "hot connect" group <u>or</u> integrated in the modular machine configuration of Lenze. • The station alias must be unambiguous and must only be assigned once within the EtherCAT network. • Use the same station alias in the EtherCAT master and in the slave. ► Address allocation (□ 35)			
Setting range (min. value unit max. value)			Lenze setting
0		32767	0
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT			

C14900

Parameter Name: C14900 Firmware: Product type			Data type: VISIBLE_STRING Index: 9675 = 0x25CB
The code contains a string with a length of 8 bytes. The following identification code is output: "E94AFCET".			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14901

Parameter Name: C14901 Firmware compilation date			Data type: VISIBLE_STRING Index: 9674 = 0x25CA
The code contains a string with a length of 20 bytes. The creation date ("MM DD YYYY") and the time ("hh:mm:ss") of the software are output here. Example: "Mar 21 2005 12:31:21"			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14902

Parameter Name: C14902 Firmware version			Data type: VISIBLE_STRING Index: 9673 = 0x25C9
The code contains a string with a length of 11 bytes. The firmware version is output here. Example: "02.00.00.36"			
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

14.4 Table of attributes

The table of attributes contains information required for communication with the inverter via parameters.

How to read the table of attributes:

Column		Meaning	Entry	
Code		Parameter name	Cxxxxx	
Name		Parameter short text (display text)	Text	
Index	dec	Index under which the parameter is addressed. The subindex for array variables corresponds to the Lenze subcode number.	24575 - Lenze code number	Is only required for access via a bus system.
	hex		0x5FFF - 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
	OCTET_STRING			
Factor	Factor for data transmission via a bus system, depending on the number of decimal positions	Factor	1 ≡ No decimal positions 10 ≡ 1 decimal position 100 ≡ 2 decimal positions 1000 ≡ 3 decimal positions	
Access	R	Read access	☒ Reading permitted	
	W	Write access	☒ Writing permitted	
	CINH	Controller inhibit required	☒ Writing is only possible if controller inhibit is set	

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C13231	Mapping entries RPDO-1	11344	0x2C50	A	32	UNSIGNED_64		☒	☒	
C13232	Mapping entries RPDO-2	11343	0x2C4F	A	32	UNSIGNED_64		☒	☒	
C13233	Mapping entries RPDO-3	11342	0x2C4E	A	32	UNSIGNED_64		☒	☒	
C13234	Mapping entries RPDO-4	11341	0x2C4D	A	32	UNSIGNED_64		☒	☒	
C13235	Mapping entries RPDO-5	11340	0x2C4C	A	32	UNSIGNED_64		☒	☒	
C13236	Mapping entries RPDO-6	11339	0x2C4B	A	32	UNSIGNED_64		☒	☒	
C13237	Mapping entries RPDO-7	11338	0x2C4A	A	32	UNSIGNED_64		☒	☒	
C13238	Mapping entries RPDO-8	11337	0x2C49	A	32	UNSIGNED_64		☒	☒	
C13239	Mapping entries RPDO-9	11336	0x2C48	A	32	UNSIGNED_64		☒	☒	
C13240	Mapping entries RPDO-10	11335	0x2C47	A	32	UNSIGNED_64		☒	☒	
C13484	Active RPDO channel	11091	0x2B53	A	1	UNSIGNED_8	1	☒	☒	
C13489	Number of RPDO mapping entries	11086	0x2B4E	A	10	UNSIGNED_8	1	☒	☒	
C13531	Mapping entries TPDO-1	11044	0x2B24	A	32	UNSIGNED_64		☒	☒	
C13532	Mapping entries TPDO-2	11043	0x2B23	A	32	UNSIGNED_64		☒	☒	
C13533	Mapping entries TPDO-3	11042	0x2B22	A	32	UNSIGNED_64		☒	☒	
C13534	Mapping entries TPDO-4	11041	0x2B21	A	32	UNSIGNED_64		☒	☒	
C13535	Mapping entries TPDO-5	11040	0x2B20	A	32	UNSIGNED_64		☒	☒	
C13536	Mapping entries TPDO-6	11039	0x2B1F	A	32	UNSIGNED_64		☒	☒	
C13537	Mapping entries TPDO-7	11038	0x2B1E	A	32	UNSIGNED_64		☒	☒	
C13538	Mapping entries TPDO-8	11037	0x2B1D	A	32	UNSIGNED_64		☒	☒	
C13539	Mapping entries TPDO-9	11036	0x2B1C	A	32	UNSIGNED_64		☒	☒	
C13540	Mapping entries TPDO-10	11035	0x2B1B	A	32	UNSIGNED_64		☒	☒	
C13784	Active TPDO channel	10791	0x2A27	A	1	UNSIGNED_8	1	☒	☒	
C13789	Number of TPDO mapping entries	10786	0x2A22	A	10	UNSIGNED_8	1	☒	☒	
C13850	All words from drive to master	10725	0x29E5	A	32	UNSIGNED_16	1	☒		
C13851	All words from master to drive	10724	0x29E4	A	32	UNSIGNED_16	1	☒		
C13852	All words to the basic device	10723	0x29E3	A	32	UNSIGNED_16	1	☒		
C13853	All words to the basic device	10722	0x29E2	A	32	UNSIGNED_16	1	☒		
C13858	Safety-over-EtherCAT (FSofE) active	10717	0x29DD	E	1	UNSIGNED_8	1	☒		
C13859	Length of the configured input data	10716	0x29DC	A	2	UNSIGNED_8	1	☒		
C13860	Length of the configured output data	10715	0x29DB	A	2	UNSIGNED_8	1	☒		
C13861	Bus status	10714	0x29DA	E	1	UNSIGNED_16	1	☒		
C13864	Active station address	10711	0x29D7	E	1	UNSIGNED_16	1	☒		
C13867	Display: Most recent emergency data	10708	0x29D4	E	1	OCTET_STRING		☒		
C13870	Cycle time display	10705	0x29D1	E	1	UNSIGNED_32	1	☒		
C13879	Bus error	10696	0x29C8	E	1	UNSIGNED_32		☒		
C13880	Resp. to communication failure	10695	0x29C7	A	1	UNSIGNED_8	1	☒	☒	
C13881	Response time when exiting 'Operational'	10694	0x29C6	E	1	UNSIGNED_16	1	☒	☒	
C13882	Display of PDO watchdog time	10693	0x29C5	E	1	UNSIGNED_32	1	☒		
C13883	DC active	10692	0x29C4	E	1	UNSIGNED_8	1	☒		
C13884	Base device synchronisation status	10691	0x29C3	E	1	UNSIGNED_8	1	☒		
C13885	Delete process data	10690	0x29C2	E	1	UNSIGNED_8	1	☒	☒	
C13887	Suppress emergency message in case of	10688	0x29C0	E	1	BITFIELD_8		☒	☒	
C13892	Process data mode	10683	0x29BB	E	1	UNSIGNED_8	1	☒	☒	
C13899	Station alias	10676	0x29B4	E	1	UNSIGNED_16	1	☒	☒	
C13900	Firmware product type	10675	0x29B3	E	1	VISIBLE_STRING		☒		
C13901	Firmware compilation date	10674	0x29B2	E	1	VISIBLE_STRING		☒		
C13902	Firmware version	10673	0x29B1	E	1	VISIBLE_STRING		☒		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C14231	Mapping entries RPDO-1	10344	0x2868	A	32	UNSIGNED_64		☒	☒	
C14232	Mapping entries RPDO-2	10343	0x2867	A	32	UNSIGNED_64		☒	☒	
C14233	Mapping entries RPDO-3	10342	0x2866	A	32	UNSIGNED_64		☒	☒	
C14234	Mapping entries RPDO-4	10341	0x2865	A	32	UNSIGNED_64		☒	☒	
C14235	Mapping entries RPDO-5	10340	0x2864	A	32	UNSIGNED_64		☒	☒	
C14236	Mapping entries RPDO-6	10339	0x2863	A	32	UNSIGNED_64		☒	☒	
C14237	Mapping entries RPDO-7	10338	0x2862	A	32	UNSIGNED_64		☒	☒	
C14238	Mapping entries RPDO-8	10337	0x2861	A	32	UNSIGNED_64		☒	☒	
C14239	Mapping entries RPDO-9	10336	0x2860	A	32	UNSIGNED_64		☒	☒	
C14240	Mapping entries RPDO-10	10335	0x285F	A	32	UNSIGNED_64		☒	☒	
C14484	Active RPDO channel	10091	0x276B	A	1	UNSIGNED_8	1	☒	☒	
C14489	Number of RPDO mapping entries	10086	0x2766	A	10	UNSIGNED_8	1	☒	☒	
C14531	Mapping entries TPDO-1	10044	0x273C	A	32	UNSIGNED_64		☒	☒	
C14532	Mapping entries TPDO-2	10043	0x273B	A	32	UNSIGNED_64		☒	☒	
C14533	Mapping entries TPDO-3	10042	0x273A	A	32	UNSIGNED_64		☒	☒	
C14534	Mapping entries TPDO-4	10041	0x2739	A	32	UNSIGNED_64		☒	☒	
C14535	Mapping entries TPDO-5	10040	0x2738	A	32	UNSIGNED_64		☒	☒	
C14536	Mapping entries TPDO-6	10039	0x2737	A	32	UNSIGNED_64		☒	☒	
C14537	Mapping entries TPDO-7	10038	0x2736	A	32	UNSIGNED_64		☒	☒	
C14538	Mapping entries TPDO-8	10037	0x2735	A	32	UNSIGNED_64		☒	☒	
C14539	Mapping entries TPDO-9	10036	0x2734	A	32	UNSIGNED_64		☒	☒	
C14540	Mapping entries TPDO-10	10035	0x2733	A	32	UNSIGNED_64		☒	☒	
C14784	Active TPDO channel	9791	0x263F	A	1	UNSIGNED_8	1	☒	☒	
C14789	Number of TPDO mapping entries	9786	0x263A	A	10	UNSIGNED_8	1	☒	☒	
C14850	All words from drive to master	9725	0x25FD	A	32	UNSIGNED_16	1	☒		
C14851	All words from master to drive	9724	0x25FC	A	32	UNSIGNED_16	1	☒		
C14852	All words to the basic device	9723	0x25FB	A	32	UNSIGNED_16	1	☒		
C14853	All words to the basic device	9722	0x25FA	A	32	UNSIGNED_16	1	☒		
C14858	Safety-over-EtherCAT (FSoE) active	9717	0x25F5	E	1	UNSIGNED_8	1	☒		
C14859	Length of the configured input data	9716	0x25F4	A	2	UNSIGNED_8	1	☒		
C14860	Length of the configured output data	9715	0x25F3	A	2	UNSIGNED_8	1	☒		
C14861	Bus status	9714	0x25F2	E	1	UNSIGNED_16	1	☒		
C14864	Active station address	9711	0x25EF	E	1	UNSIGNED_16	1	☒		
C14867	Display: Most recent emergency data	9708	0x25EC	E	1	OCTET_STRING		☒		
C14870	Cycle time display	9705	0x25E9	E	1	UNSIGNED_32	1	☒		
C14879	Bus error	9696	0x25E0	E	1	UNSIGNED_32		☒		
C14880	Resp. to communication failure	9695	0x25DF	A	1	UNSIGNED_8	1	☒	☒	
C14881	Response time when exiting 'Operational'	9694	0x25DE	E	1	UNSIGNED_16	1	☒	☒	
C14882	Display of PDO watchdog time	9693	0x25DD	E	1	UNSIGNED_32	1	☒		
C14883	DC active	9692	0x25DC	E	1	UNSIGNED_8	1	☒		
C14884	Base device synchronisation status	9691	0x25DB	E	1	UNSIGNED_8	1	☒		
C14885	Delete process data	9690	0x25DA	E	1	UNSIGNED_8	1	☒	☒	
C14887	Suppress emergency message in case of	9688	0x25D8	E	1	BITFIELD_8		☒	☒	
C14892	Process data mode	9683	0x25D3	E	1	UNSIGNED_8	1	☒	☒	
C14899	Station alias	9676	0x25CC	E	1	UNSIGNED_16	1	☒	☒	
C14900	Firmware product type	9675	0x25CB	E	1	VISIBLE_STRING		☒		
C14901	Firmware compilation date	9674	0x25CA	E	1	VISIBLE_STRING		☒		
C14902	Firmware version	9673	0x25C9	E	1	VISIBLE_STRING		☒		

A

Abort codes [80](#)
Activating changed settings [55](#)
Active RPDO channel (C13484) [105](#)
Active RPDO channel (C14484) [123](#)
Active station address (C13864) [113](#)
Active station address (C14864) [131](#)
Active TPDO channel (C13784) [110](#)
Active TPDO channel (C14784) [128](#)
Address allocation [35](#)
ADS communication parameters in »TwinCAT« and »Engineer« [46](#)
AL Status Code [60](#)
All words from drive to master (C13850) [111](#)
All words from master to drive (C13851) [111](#)
All words from the master (C14851) [129](#)
All words from the standard device (C14853) [130](#)
All words to the basic device (C13852) [111](#)
All words to the basic device (C13853) [112](#)
All words to the master (C14850) [129](#)
All words to the standard device (C14852) [129](#)
Application as directed [12](#)
Application notes [9](#)
Approvals [15](#)
Automatic device identification [32](#)

B

Base device synchronisation status (C13884) [115](#)
Base device synchronisation status (C14884) [133](#)
Baud rate [15](#)
Before initial switch-on [30](#)
Behaviour of the communication module during the start [39](#)
Bus Error (C13879) [114](#)
Bus error (C14879) [132](#)
Bus state (C13861) [113](#)
Bus state (C14861) [131](#)

C

C00615 | Resp. to imp. device conf. [97](#)
C00636 | Resp. to new module in MXI1 [97](#)
C00637 | Resp. to new module in MXI2 [98](#)
C01120 | Sync source [98](#)
C01501 | Resp. to comm. error with MXI1 [98](#)
C01502 | Resp. to comm. error with MXI2 [99](#)
C13231 | Mapping entries RPDO-1 (I-1600) [100](#)
C13232 | Mapping entries RPDO-2 [100](#)
C13233 | Mapping entries RPDO-3 [101](#)
C13234 | Mapping entries RPDO-4 [101](#)
C13235 | Mapping entries RPDO-5 [102](#)
C13236 | Mapping entries RPDO-6 [102](#)
C13237 | Mapping entries RPDO-7 [103](#)
C13238 | Mapping entries RPDO-8 [103](#)
C13239 | Mapping entries RPDO-9 [104](#)

C13240 | Mapping entries RPDO-10 [104](#)
C13484 | Active RPDO channel [105](#)
C13489 | No. of RPDO mapping entries [105](#)
C13531 | Mapping entries TPDO-1 [105](#)
C13532 | Mapping entries TPDO-2 [106](#)
C13533 | Mapping entries TPDO-3 [106](#)
C13534 | Mapping entries TPDO-4 [107](#)
C13535 | Mapping entries TPDO-5 [107](#)
C13536 | Mapping entries TPDO-6 [108](#)
C13537 | Mapping entries TPDO-7 [108](#)
C13538 | Mapping entries TPDO-8 [109](#)
C13539 | Mapping entries TPDO-9 [109](#)
C13540 | Mapping entries TPDO-10 [110](#)
C13784 | Active TPDO channel [110](#)
C13789 | No. of TPDO mapping entries [110](#)
C13850 | All words from drive to master [111](#)
C13851 | All words from master to drive [111](#)
C13852 | All words to the basic device [111](#)
C13853 | All words to the basic device [112](#)
C13858 | Safety-over-EtherCAT (FSOE) active [112](#)
C13859 | Length of the configured input data [112](#)
C13860 | Length of the configured output data [112](#)
C13861 | Bus state [113](#)
C13864 | Active station address [113](#)
C13867 | Display
 Most recent emergency data [113](#)
C13870 | Cycle time display [113](#)
C13879 | Bus error [114](#)
C13880 | Resp. to communication failure [114](#)
C13881 | Response time when exiting 'Operational' [114](#)
C13882 | Display of PDO watchdog time [115](#)
C13883 | DC active [115](#)
C13884 | Base device synchronisation status [115](#)
C13885 | Delete process data [115](#)
C13887 | Suppress emergency message in case of [116](#)
C13892 | Process data mode [116](#)
C13899 | Station alias [116](#)
C13900 | Firmware
 Product type [117](#)
C13901 | Firmware
 Compilation date [117](#)
C13902 | Firmware
 Version [117](#)
C14231 | Mapping entries RPDO-1 (I-1600) [118](#)
C14232 | Mapping entries RPDO-2 [118](#)
C14233 | Mapping entries RPDO-3 [119](#)
C14234 | Mapping entries RPDO-4 [119](#)
C14235 | Mapping entries RPDO-5 [120](#)
C14236 | Mapping entries RPDO-6 [120](#)
C14237 | Mapping entries RPDO-7 [121](#)
C14238 | Mapping entries RPDO-8 [121](#)
C14239 | Mapping entries RPDO-9 [122](#)
C14240 | Mapping entries RPDO-10 [122](#)

C14484 | Active RPDO channel [123](#)
C14489 | No. of RPDO mapping entries [123](#)
C14531 | Mapping entries TPDO-1 [123](#)
C14532 | Mapping entries TPDO-2 [124](#)
C14533 | Mapping entries TPDO-3 [124](#)
C14534 | Mapping entries TPDO-4 [125](#)
C14535 | Mapping entries TPDO-5 [125](#)
C14536 | Mapping entries TPDO-6 [126](#)
C14537 | Mapping entries TPDO-7 [126](#)
C14538 | Mapping entries TPDO-8 [127](#)
C14539 | Mapping entries TPDO-9 [127](#)
C14540 | Mapping entries TPDO-10 [128](#)
C14784 | Active TPDO channel [128](#)
C14789 | No. of TPDO mapping entries [128](#)
C14850 | All words to master [129](#)
C14851 | All words from master [129](#)
C14852 | All words to standard device [129](#)
C14853 | All words from standard device [130](#)
C14858 | Safety-over-EtherCAT (FSOE) active [130](#)
C14859 | Length of the configured input data [130](#)
C14860 | Length of the configured output data [130](#)
C14861 | Bus state [131](#)
C14864 | Active station address [131](#)
C14867 | Display
 Most recent emergency data [131](#)
C14870 | Cycle time display [131](#)
C14879 | Bus error [132](#)
C14880 | Resp. to communication failure [132](#)
C14881 | Response time when exiting 'Operational' [132](#)
C14882 | Display of PDO watchdog time [133](#)
C14883 | DC active [133](#)
C14884 | Base device synchronisation status [133](#)
C14885 | Clear process data [133](#)
C14887 | Suppress emergency message in case of [134](#)
C14892 | Process data mode [134](#)
C14899 | Station alias [135](#)
C14900 | Firmware
 Product type [135](#)
C14901 | Firmware
 Compilation date [135](#)
C14902 | Firmware
 Version [135](#)
Clear process data (C14885) [133](#)
Codes [97](#)
Colour code of the Ethernet cable [28](#)
Commissioning [30](#)
Communication medium [15](#)
Communication objects (object directory) [78](#)
Communication parameters in »TwinCAT« and »Engineer« [46](#)
Communication profile [15](#)
Communication time [19](#)
Configuration of the master [31](#)
Configuring process data [33](#)

Configuring process data (PDO mapping) [62](#)
Configuring safety process data [33](#)
Configuring the Controller (EtherCAT master) [31](#)
Configuring the EtherCAT master [31](#)
Conformities [15](#)
Connections [14](#)
Conventions [7](#)
Conventions used [7](#)
Cycle time display (C13870) [113](#)
Cycle time display (C14870) [131](#)
Cycle times [15](#)

D

Data transfer [56](#)
Datagrams [58](#)
DC active (C13883) [115](#)
DC active (C14883) [133](#)
DC configuration in the master [37](#)
DC configuration in the Servo Drive 9400 (slave) [38](#)
DC master [36](#)
Define identifier (station alias) [35](#)
Define station alias [35](#)
Defining Explicit Device Identification [35](#)
Delete process data (C13885) [115](#)
Determining the cycle time [33](#)
Device and application-specific safety instructions [11](#)
Device identification [32](#)
Device profile [15](#)
Device protection [11](#)
Diagnostics [85](#)
Diagnostics with the »Engineer« [90](#)
Dimensions [21](#)
Disassembly [23](#)
Display
 Most recent emergency data (C13867) [113](#)
 Most recent emergency data (C14867) [131](#)
Display of PDO watchdog time (C13882) [115](#)
Display of PDO watchdog time (C14882) [133](#)
Distributed clocks (DC) [36](#)
Document history [6](#)
Download [73](#)

E

Electrical installation [24](#)
E-mail to Lenze [144](#)
Emergency messages [91](#)
Emergency requests [91](#)
Error code [91](#)
Error messages [92](#)
 Causes and remedies [93](#)
 Short overview [92](#)
Error number
 0x00c83100 [93](#)

0x00c85114 [93](#)
0x00c85531 [93](#)
0x00c85532 [93](#)
0x00c85533 [93](#)
0x00c86010 [94](#)
0x00c86011 [94](#)
0x00c86100 [94](#)
0x00c86101 [94](#)
0x00c86110 [94](#)
0x00c8641f [94](#)
0x00c86420 [95](#)
0x00c86610 [95](#)
0x00c86612 [95](#)
0x00c88131 [95](#)
0x00c88132 [95](#)
0x00c88700 [96](#)
Establishing a master - slave connection [67](#)
Establishing an online connection with the »Engineer« [41](#)
Establishing communication [55](#)
EtherCAT
 Communication time-out with safety module (error message) [95](#)
 Connection to 9400 got lost (error message) [93](#)
 Fault
 Error
 Lenze Setting Loaded (error message) [95](#)
 Initialisation error with safety module (error message) [95](#)
 Internal error (error message) [94](#)
 Internal mapping error (error message) [94](#)
 Internal supply voltage too low (error message) [93](#)
 Invalid Parameter Set (error message) [94](#)
 Memory
 No access (error message) [93](#)
 Read error (error message) [93](#)
 Write error (error message) [93](#)
 Quit 'Operational' status (error message) [95](#)
 Restart by watchdog reset (error message) [94](#)
 Sync loss detected with standard device (error message) [95](#)
 Sync telegram failure (error message) [96](#)
EtherCAT ADS communication parameters in »TwinCAT« and »Engineer« [46](#)
EtherCAT connection [26](#)
EtherCAT datagrams [58](#)
EtherCAT error messages
 Causes and remedies [93](#)
 Short overview [92](#)
EtherCAT state machine [59](#)
EtherCAT status displays [87](#)
EtherCAT synchronisation loss [84](#)
EtherCAT telegrams [57](#)
Ethernet cable [27](#)
Ethernet cable specification [27](#)
Ethernet cable, colour code [28](#)
Ethernet cable, structure [27](#)

External voltage supply [29](#)

F

Feedback to Lenze [144](#)

Firmware

 Compilation date (C13901) [117](#)

 Compilation date (C14901) [135](#)

 product type (C13900) [117](#)

 Product type (C14900) [135](#)

 version (C13902) [117](#)

 Version (C14902) [135](#)

FSoE mapping indices [66](#)

G

Gateway Controller -> configure EtherCAT [42](#)

Gateway Controller -> configure EtherCAT ADS (Beckhoff) [44](#)

General data [15](#)

General safety and application notes [10](#)

I

Identification [12](#)

Indexing of the Lenze codes [68](#)

Initial switch-on [55](#)

Installation [22](#)

Installing device description files [31](#)

Interface for communication [15](#)

Interfaces [14](#)

Interruption of EtherCAT communication [82](#)

Interruption of internal communication [83](#)

L

Layout of the safety instructions [9](#)

LED status displays [85](#)

Length of the configured input data (C13859) [112](#)

Length of the configured input data (C14859) [130](#)

Length of the configured output data (C13860) [112](#)

Length of the configured output data (C14860) [130](#)

Line topology [25](#)

M

Mailbox datagram [68](#)

Mailbox protocol [15](#)

Mapping codes [66](#)

Mapping entries RPDO-1 (I-1600) (C13231) [100](#)

Mapping entries RPDO-1 (I-1600) (C14231) [118](#)

Mapping entries RPDO-10 (C13240) [104](#)

Mapping entries RPDO-10 (C14240) [122](#)

Mapping entries RPDO-2 (C13232) [100](#)

Mapping entries RPDO-2 (C14232) [118](#)

Mapping entries RPDO-3 (C13233) [101](#)

Mapping entries RPDO-3 (C14233) [119](#)

Mapping entries RPDO-4 (C13234) [101](#)

Mapping entries RPDO-4 (C14234) [119](#)

Mapping entries RPDO-5 (C13235) [102](#)

Mapping entries RPDO-5 (C14235) [120](#)
Mapping entries RPDO-6 (C13236) [102](#)
Mapping entries RPDO-6 (C14236) [120](#)
Mapping entries RPDO-7 (C13237) [103](#)
Mapping entries RPDO-7 (C14237) [121](#)
Mapping entries RPDO-8 (C13238) [103](#)
Mapping entries RPDO-8 (C14238) [121](#)
Mapping entries RPDO-9 (C13239) [104](#)
Mapping entries RPDO-9 (C14239) [122](#)
Mapping entries TPDO-1 (C13531) [105](#)
Mapping entries TPDO-1 (C14531) [123](#)
Mapping entries TPDO-10 (C13540) [110](#)
Mapping entries TPDO-10 (C14540) [128](#)
Mapping entries TPDO-2 (C13532) [106](#)
Mapping entries TPDO-2 (C14532) [124](#)
Mapping entries TPDO-3 (C13533) [106](#)
Mapping entries TPDO-3 (C14533) [124](#)
Mapping entries TPDO-4 (C13534) [107](#)
Mapping entries TPDO-4 (C14534) [125](#)
Mapping entries TPDO-5 (C13535) [107](#)
Mapping entries TPDO-5 (C14535) [125](#)
Mapping entries TPDO-6 (C13536) [108](#)
Mapping entries TPDO-6 (C14536) [126](#)
Mapping entries TPDO-7 (C13537) [108](#)
Mapping entries TPDO-7 (C14537) [126](#)
Mapping entries TPDO-8 (C13538) [109](#)
Mapping entries TPDO-8 (C14538) [127](#)
Mapping entries TPDO-9 (C13539) [109](#)
Mapping entries TPDO-9 (C14539) [127](#)
Mapping indices [65](#)
Mapping indices (FSoE) [66](#)
Max. cable length [15](#)
Mechanical installation [23](#)
Module status displays [86](#)
Monitoring [82](#)
Mounting [23](#)
Mounting clearance [27](#)

N

Nameplate [12](#)
Network topology [15](#), [25](#)
No. of RPDO mapping entries (C13489) [105](#)
No. of RPDO mapping entries (C14489) [123](#)
Number of devices [15](#)
Number of TPDO mapping entries (C13789) [110](#)
Number of TPDO mapping entries (C14789) [128](#)

O

Object directory (communication objects) [78](#)
Operating conditions [15](#)

P

Parameter data transfer [67](#)

Parameter reference [97](#)

Parameters of the communication module for slot MXI1 [100](#)

Parameters of the communication module for slot MXI2 [118](#)

Parameters of the standard device that are relevant to communication [97](#)

PDO mapping [62](#)

Pin assignment of the RJ45 sockets [26](#)

Process data [61](#)

Process data mode [40](#)

Process data mode (C13892) [116](#)

Process data mode (C14892) [134](#)

Process data transfer [61](#)

Processing time [19](#)

Product codes for Servo Drives 9400 [32](#)

Product description [12](#)

Product features [13](#)

Product ID [15](#)

Protection against uncontrolled restart [55](#)

Protection of persons [11](#)

Protective insulation [16](#)

Protocol data [19](#)

R

Reading parameters (SDO upload) [69](#)

Residual hazards [11](#)

Resp. to comm. error with MXI1 (C01501) [98](#)

Resp. to comm. error with MXI2 (C01502) [99](#)

Resp. to communication failure (C13880) [114](#)

Resp. to communication failure (C14880) [132](#)

Resp. to imp. device config. (C00615) [97](#)

Resp. to new module in MXI1 (C00636) [97](#)

Resp. to new module in MXI2 (C00637) [98](#)

Response time when exiting 'Operational' (C13881) [114](#)

Response time when exiting 'Operational' (C14881) [132](#)

Revision ID [15](#)

S

Safety instructions [9](#), [10](#)

Safety over EtherCAT [81](#)

Safety-over-EtherCAT (FSoE) active (C13858) [112](#)

Safety-over-EtherCAT (FSoE) active (C14858) [130](#)

Screenshots/application examples [5](#)

SDO abort codes (Abort codes) [80](#)

SDO download [73](#)

SDO upload [69](#)

Setting PDO mapping with the »Engineer« [63](#)

Settings in the »Engineer« [34](#)

State machine [59](#)

Station alias (C13899) [116](#)

Station alias (C14899) [135](#)

Status display at the RJ45 sockets (X246 / X247) [89](#)

Status displays (LEDs) [85](#)

Structure of the Emergency message [91](#)

Structure of the Ethernet cable [27](#)
Suppress emergency message in case of (C13887) [116](#)
Suppress emergency message in case of (C14887) [134](#)
Switch topology [25](#)
Sync source (C01120) [98](#)
Sync telegram failure detection [83](#)
Synchronisation loss [84](#)
Synchronisation with "Distributed clocks" (DC) [36](#)
System error messages [92](#)

T

Table of attributes [136](#)
Target group [5](#)
Technical data [15](#)
Telegram failure detection (Sync) [83](#)
Terminal data [29](#)
Terminology used [8](#)
Terms [8](#)
Type of device [15](#)

U

Upload [69](#)
Using the communication module [12](#)

V

Validity of the documentation [5](#)
Vendor ID [15](#)
Voltage supply [15](#), [29](#)

W

Wiring according to EMC guidelines [24](#)
Writing parameters (SDO download) [73](#)



Your opinion is important to us

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

Perhaps we have not succeeded in achieving this objective in every respect. If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.com

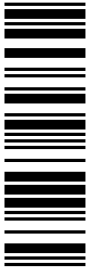
Thank you very much for your support.

Your Lenze documentation team

Lenze Automation GmbH
Postfach 10 13 52, 31763 Hameln
Hans-Lenze-Straße 1, 31855 Aerzen
GERMANY
HR Hannover B 205381
☎ +49 5154 82-0
📠 +49 5154 82-2800
✉ lenze.de@lenze.com
🌐 www.lenze.com

Service

Lenze Service GmbH
Breslauer Straße 3, 32699 Extertal
GERMANY
☎ 008000 24 46877 (24 h helpline)
📠 +49 5154 82-1112
✉ service.de@lenze.com



PC-based automation



EtherCAT®

EtherCAT control technology
Commissioning & configuration

Contents

1	About this documentation	7
1.1	Document history	9
1.2	Conventions used	10
1.3	Terminology used	11
1.4	Notes used	12
2	Safety instructions	13
3	The "PC-based automation" system	14
4	The Lenze control system with EtherCAT	16
4.1	Brief description of EtherCAT	16
4.1.1	Features	16
4.1.2	Structure of the EtherCAT bus system	17
4.1.3	Communication	18
4.2	Required hardware components	22
4.2.1	The industrial PC - the central component	22
4.2.2	Field devices	23
4.2.3	EtherCAT product codes	24
4.2.4	EtherCAT hardware for the industrial PC	26
4.3	Required engineering tools	28
4.4	Interaction of the components	30
4.4.1	The state machine of the Lenze control technology	30
4.4.2	Communication between engineering PC and field devices	32
5	Technical data	34
5.1	General data of the EtherCAT bus	34
5.2	MC-ETC communication card	34
5.3	Communication times and drive-specific data	35
6	Synchronisation with "Distributed clocks"	36
6.1	Synchronous communication	37
6.2	Adjusting task cycle time and DC cycle time	38
6.3	Setting the DC synchronisation with the »EtherCAT Configurator«	39
6.4	Check of the DC synchronicity	41

7	<u>Commissioning of the system</u>	42
7.1	<u>Overview of commissioning steps</u>	43
7.2	<u>Detailed commissioning steps</u>	45
7.2.1	<u>Planning the bus topology</u>	45
7.2.2	<u>Installing field devices</u>	45
7.2.3	Creating a project folder	46
7.2.4	<u>Determining the physical EtherCAT configuration (fieldbus scan)</u>	47
7.2.5	<u>Configuration in the »Engineer«</u>	54
7.2.6	<u>Inserting devices available on the fieldbus into the »EtherCAT Configurator« project</u>	55
7.2.7	<u>Creating the configuration in the »EtherCAT Configurator«</u>	56
7.2.8	<u>Configuration in the »PLC Designer«</u>	68
7.3	<u>Checking the system startup</u>	80
7.3.1	<u>Evaluation of the boot-up error message of the SM DriveBasic.lib library</u>	80
7.3.2	<u>Evaluation of the Axis IO Group state</u>	81
7.3.3	<u>Evaluation of the axis state</u>	82
7.4	<u>Typical commissioning scenarios</u>	83
7.4.1	<u>Switching on a completely configured system</u>	83
7.4.2	<u>Switching on a system with an incomplete configuration</u>	85
7.4.3	<u>Updating the PLC application while the EtherCAT configuration remains unchanged</u>	87
7.4.4	<u>Stopping and starting the PLC while the configuration remains unchanged</u>	89
7.5	<u>Detailed overview of the commissioning steps</u>	90
8	<u>EtherCAT with CANopen or PROFIBUS</u>	91
8.1	<u>Addressing the PROFIBUS and CANopen stations</u>	92
8.2	<u>Addressing EtherCAT nodes using CANopen/PROFIBUS nodes</u>	93
9	<u>EtherCAT function libraries</u>	96
9.1	<u>Usability</u>	96
9.2	<u>Function blocks/functions required for a »PLC Designer« project (overview)</u>	97
9.3	<u>Properties of function blocks</u>	98
9.4	<u>The EC T STATE structure</u>	98
9.5	<u>Function blocks and functions for master/slave states</u>	99
9.5.1	<u>ecatStartAsync (FB)</u>	100
9.5.2	<u>ecatStopAsync (FB)</u>	101
9.5.3	<u>ecatSetMasterStateAsync (FB)</u>	102
9.5.4	<u>ecatSetSlaveStateAsync (FB)</u>	103
9.5.5	<u>ecatGetMasterState (FUN)</u>	105
9.5.6	<u>ecatGetSlaveState (FUN)</u>	106
9.5.7	<u>ecatGetSlaveStateAsync (FB)</u>	107

9.6	Functions for the network management	108
9.6.1	ecatMasterIsConfigured (FUN)	108
9.6.2	ecatGetSlaveId (FUN)	109
9.6.3	ecatGetSlaveIdAtPosition (FUN)	110
9.6.4	ecatGetSlaveProp (FUN)	111
9.7	Function blocks and functions for diagnosing the network	112
9.7.1	ecatGetNumConfiguredSlaves (FUN)	112
9.7.2	ecatGetNumConnectedSlaves (FUN)	113
9.7.3	ECATDiagnostic (FB)	113
9.7.4	ResetMasterStatus (FB)	114
9.7.5	SMC_ETCErrorString (FUN)	115
9.7.6	L_ECATA_ReadErrCnt (FB)	116
9.7.7	L_ECATA_ResetErrCnt (FB)	117
9.7.8	The global EtherCAT master structure ECAT_MASTER	118
9.8	Function blocks for CANopen over EtherCAT (CoE)	122
9.8.1	ecatCoeSdoDownloadReq (FB)	122
9.8.2	ecatCoeSdoUploadReq (FB)	123
10	Defining the minimum cycle time of the PLC project	125
10.1	Calculating the total access time to the peripheral devices ($T_{\text{Correction}}$)	125
10.2	Detecting the task utilisation of the application ($T_{\text{Task utilisation}}$)	126
10.2.1	Display of the system utilisation in the »PLC Designer« with the task editor	126
10.2.2	Detecting the task utilisation	127
10.3	Calculating the minimum cycle time	128
10.4	Optimising the system	129
11	Diagnostics	130
11.1	Diagnostics with the »EtherCAT Configurator«	131
11.1.1	"Diagnostics" tab	131
11.1.2	Representation in the online mode	133
11.2	Diagnostics with the »PLC Designer«	134
11.2.1	VISU_ETHERCATMaster visualisation template	135
11.2.2	VISU_ECATADiagnostic visualisation template	136
11.2.3	The global variable wState	137
11.2.4	Error scenario (example)	139
11.3	Diagnostic codes	140
11.4	Logbook of the IPC	140
11.4.1	Displaying the EtherCAT entries of the logbook	141
11.4.2	Messages in the logbook of the industrial PC	142

11.5	Error counters of the EtherCAT slaves	143
11.5.1	Error types: "Errors" and "Forwarded Errors"	143
11.5.2	Error counter reset from the application	144
11.6	Error scenarios	145
11.6.1	Compilation error in »PLC Designer«	147
11.6.2	EtherCAT bus does not enter the Pre-Operational state	147
11.6.3	Control unit/PLC does not enter the RUN state	147
11.6.4	EtherCAT bus does not enter the Operational state	148
11.6.5	Error during EtherCAT data transmission	148
11.6.6	Shafts make clicking noises	149
11.6.7	Shafts do not rotate	150
11.6.8	Logbook message: "Cannot spawn Remote API Server"	151
11.6.9	Logbook message: "Ethernet cable not connected"	152
11.6.10	Logbook message: "Ethernet cable connected"	153
11.6.11	Logbook messages: "Slave at index X missing" with "Cyclic command WKC error ..."	154
11.6.12	Logbook message: "Cyclic command WKC error ..."	156
11.7	System error messages	157
11.7.1	IPC logbook messages	157
11.7.2	General error codes (0x00000000_{hex}, 0x98110001 ... 0x98110038_{hex})	159
11.7.3	CANOpen over EtherCAT (CoE) SDO error codes (0x98110040 ... 0x9811005D_{hex})	161
11.7.4	Remote API error codes (0x98110181 ... 0x98110196_{hex})	163
11.8	SDO abort codes	164
12	Parameter reference	165
12.1	Parameters of the MC-ETC communication card in slot 1	166
12.2	Interface parameters of the MC-ETC communication card in slot 1	167
12.3	Parameters of the MC-ETC communication card in slot 2	182
12.4	Interface parameters of the MC-ETC communication card in slot 2	183
13	Index	198

1 About this documentation



Note!

For industrial PCs of the **EL 1xx**, **EL x8xx**, **CS x8xx** and **CPC x8xx** series in control technology **release 2.5**, EtherCAT is not supported.

This documentation ...

- This documentation contains detailed information on how to commission, configure and diagnose the EtherCAT bus system within the scope of Lenze's control technology.
- belongs to the "PC-based automation" manual collection which consists of the following documentation:

Documentation	Subject
System manuals "PC-based automation"	• Control technology - <i>system structure & configuration</i> • Control technology - <i>system structure & components</i>
Communication manuals "PC-based automation"	• CANopen control technology • Control technology PROFIBUS • EtherCAT control technology
(Software) Manual "PC-based automation"	• Industrial PC - <i>parameter setting & configuration</i>
Operating Instructions "Embedded Line Panel PC"	• EL x7xx - built-in panel-PC with TFT display
Operating Instructions "Command Station"	• CS x7xx - stand-alone operator terminal
Operating Instructions "Control Cabinet PC"	• CPC x7xx - control cabinet PC
Operating Instructions "HMI EL 100"	• EL 1xx - HMI with Windows® CE
Further software manuals	• »Global Drive Control« (»GDC«) – IPC as gateway - <i>parameter setting & configuration</i> • »Engineer« • »PLC Designer« / »PLC Designer« / »PLC Designer - CANopen for Runtime Systems« • »VisiWinNET® Smart«

Further technical documentations for Lenze components

More information about Lenze components that can be used together with "PC-based automation" can be found in the following documents:

Mounting & wiring
 MAs for Inverter Drives 8400
 MAs for Servo Drives 9400
 MA EPM-Txxx (I/O system IP20)
 MA EPM-Sxxx (I/O system 1000)
 MA 8200 vector
 Wiring according to EMC, 8200 vector
 MAs for the ECS servo system
 MA communication card MC-CAN2
 MA communication card MC-ETC
 MA communication card MC-ETH
 MA communication card MC-PBM
 MA communication card MC-PBS
 MA communication card MC-MPI
 MAs for communication modules

Legend:

-  Printed documentation
-  Online help/PDF

Abbreviations used:

- SHB System Manual
- BA Operating Instructions
- MA Mounting Instructions
- SW Software manual
- KHB Communication manual

Parameter setting, configuration, commissioning
 SW Inverter Drive 8400 Baseline / StateLine / HighLine / TopLine
 SW Servo Drive 9400 HighLine / PLC
 Commissioning guide 9400 HighLine
 SHB I/O system IP20 (EPM-Txxx)
 SHB I/O system 1000 (EPM-Sxxx)
 SHB 8200 vector
 BAs for the ECS servo system
 KHBs for communication modules

Programming
 SW 9400 function library

Creating a network
 KHBs for communication modules



Tip!

Manuals and software updates for Lenze products can be found in the download area at:

<http://www.Lenze.com>

Target group

This documentation is intended for all persons who plan, install, commission and maintain the networking of devices in the field of control technology.

1.1 Document history

Material no.	Version			Description
-	1.0	09/2008	TD11 TD17	First edition
13296253	2.0	05/2009	TD17	General revision
13317335	3.0	10/2009	TD17	General revision
13369406	3.1	01/2011	TD17	• General updates • Information on control technology release 2.5 has been added.

Your opinion is important to us!

These instructions were created to the best of our knowledge and belief to give you the best possible support for handling our product.

If you have suggestions for improvement, please e-mail us to:

feedback-docu@lenze.de

Thank you for your support.

Your Lenze documentation team

1.2 Conventions used

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

Type of information	Writing	Examples/notes
Spelling of numbers		
Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text		
Version information	Blue text colour	All information valid for or from a certain software version, is indicated accordingly in this documentation. Example: This function extension is available from software version V3.0!
Program name	» «	The Lenze PC software »Engineer«...
Window	<i>Italics</i>	The <i>Message window</i> ... / The <i>Options</i> dialog box...
Variable identifier		By setting <i>bEnable</i> to TRUE...
Control element	Bold	The OK button... / the Copy command... / the Characteristics tab... / the Name input field...
Sequence of menu commands		If the execution of a function requires several commands in a row, the individual commands are separated by an arrow: Select File → Open to ...
Shortcut	< Bold >	Use < F1 > to open the online help. If a key combination is required for a command, a "+" is placed between the key identifiers: With < Shift >+< ESC >...
Program code	Courier	IF var1 < var2 THEN a = a + 1 END IF
Keyword	Courier bold	
Hyperlink	<u>Underlined</u>	Optically highlighted reference to another topic. It is activated with a mouse-click in this documentation.
Symbols		
Page reference	 10	Optically highlighted reference to another page. It is activated with a mouse-click in this documentation.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

1.3 Terminology used

Term	Meaning
»Engineer«	Lenze engineering tools supporting you during the entire life cycle of a machine - from the planning phase to maintenance.
»EtherCAT Configurator«	
»Global Drive Control« (GDC)	
»PLC Designer«	
Code	"Container" for one or several parameters used for Lenze Servo Drives parameter setting or monitoring.
Subcode	If a code contains several parameters, they are stored in "subcodes". In the documentation the diagonal slash "/" is used as a separator between the designation of the code and subcode (e.g. "C00118/3").
IPC	Industrial PC
PLC	Programmable Logic Controller
AT-EM	EtherCAT master
CoE	CANopen over EtherCAT
DC	Distributed clocks (distributed synchronised clocks)
EoE	Ethernet over EtherCAT
FoE	File access over EtherCAT
MCF	Master configuration file (XML file for EtherCAT bus configuration)
SoE	Servo drive profile over EtherCAT
FB	Function block (contained in a function library)
FUN	Function (contained in a function library)

1.4 Notes used

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

Safety instructions

Structure of safety instructions:



Pictograph and signal word!

(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 injuries through dangerous electrical voltage Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
	Danger!	Danger of personal injury through a general source of danger Reference to an imminent danger that may result in death or serious personal injury if the corresponding measures are not taken.
	Stop!	Danger of property damage Reference to a possible danger that may result in property damage if the corresponding measures are not taken.

Application notes

Pictograph	Signal word	Meaning
	Note!	Important note for trouble-free operation
	Tip!	Useful tip for easy handling
		Reference to another documentation

2 Safety instructions

Please observe the following safety instructions when you want to commission a controller or system using an industrial PC.



Read the documentation supplied with the system components thoroughly before starting to commission the devices and the industrial PC!

The System Manual contains safety instructions which must be observed!



Danger!

According to our present level of knowledge it is not possible to ensure the absolute freedom from errors of a software.

If necessary, systems with built-in controllers must be provided with additional monitoring and protective equipment according to relevant safety regulations (e.g. law on technical equipment, regulations for the prevention of accidents) so that an impermissible operating status does not endanger persons or facilities.

During commissioning persons must keep a safe distance from the motor or the machine parts driven by the motor. Otherwise there would be a risk of injury by the moving machine parts.



Stop!

If you change parameters in an engineering tool during an existing online connection to a device, the changes are directly added to the device!

A wrong parameter setting can cause unpredictable motor movements. By unintentional direction of rotation, too high speed or jerky operation, the driven machine parts may be damaged!

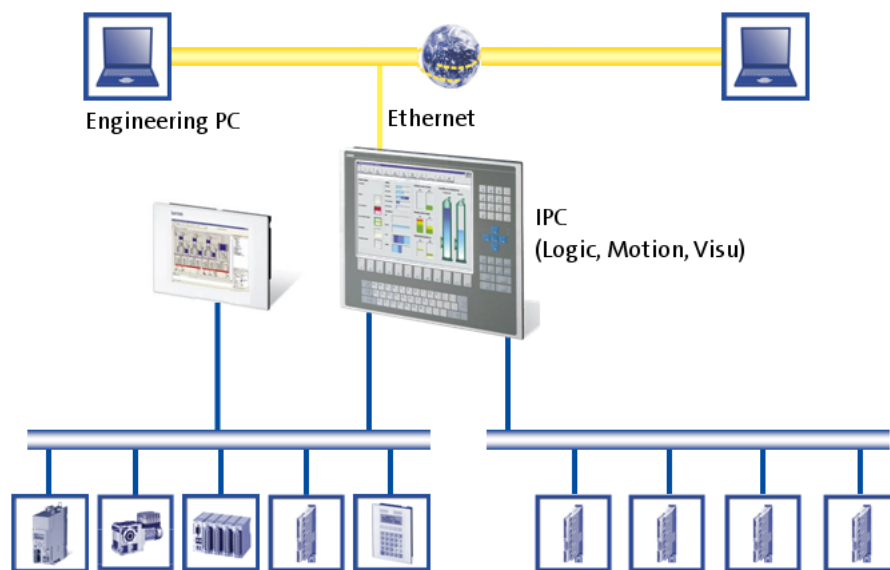
3 The "PC-based automation" system

Industrial PCs (IPCs) are increasingly finding their way into automation technology. With their scaling characteristics and the possible combination of visualisation and control in one device, industrial PCs offer clear advantages for many applications.

Lenze industrial PCs are available with the following software equipments:

- ▶ Industrial PC as component (optional with operating system) without any further software
- ▶ Industrial PC as visualisation system
- ▶ Industrial PC as control and visualisation system

The "PC-based automation" system allows the central control of Logic and Motion systems.



For this purpose, Lenze provides coordinated system components:

- ▶ Industrial PCs as control and visualisation system
 - The IPC is the central component of the PC-based automation which control the Logic and Motion functionalities by means of the runtime software.
 - The IPC communicates with the field devices via the fieldbus.
 - The IPCs are available in different designs.



Note!

Moreover, the HMI series EL 1xx PLC belongs to the "PC based automation" system. These devices differ considerably from the industrial PCs in performance and various other details. However the devices of the HMI series EL 1xx PLC are able to fulfil smaller control functions.

- ▶ Engineering tools for the Engineering PC
 - The Engineering PC communicates with the IPC via Ethernet.
 - The different engineering tools are used to configure and parameterise the system.
- ▶ Fieldbuses
- ▶ Field devices

4 The Lenze control system with EtherCAT



Note!

For industrial PCs of the **EL 1xx**, **EL x8xx**, **CS x8xx** and **CPC x8xx** series in control technology **release 2.5**, EtherCAT is not supported.

This chapter provides basic information about ...

- ▶ the EtherCAT bus system;
- ▶ the structure of the Lenze control system with the EtherCAT bus system;
- ▶ the Lenze engineering tools required for commissioning.
- ▶ the interaction of the components.

4.1 Brief description of EtherCAT



Tip!

More detailed information about EtherCAT can be found on the Internet website of the EtherCAT Technology Group under:

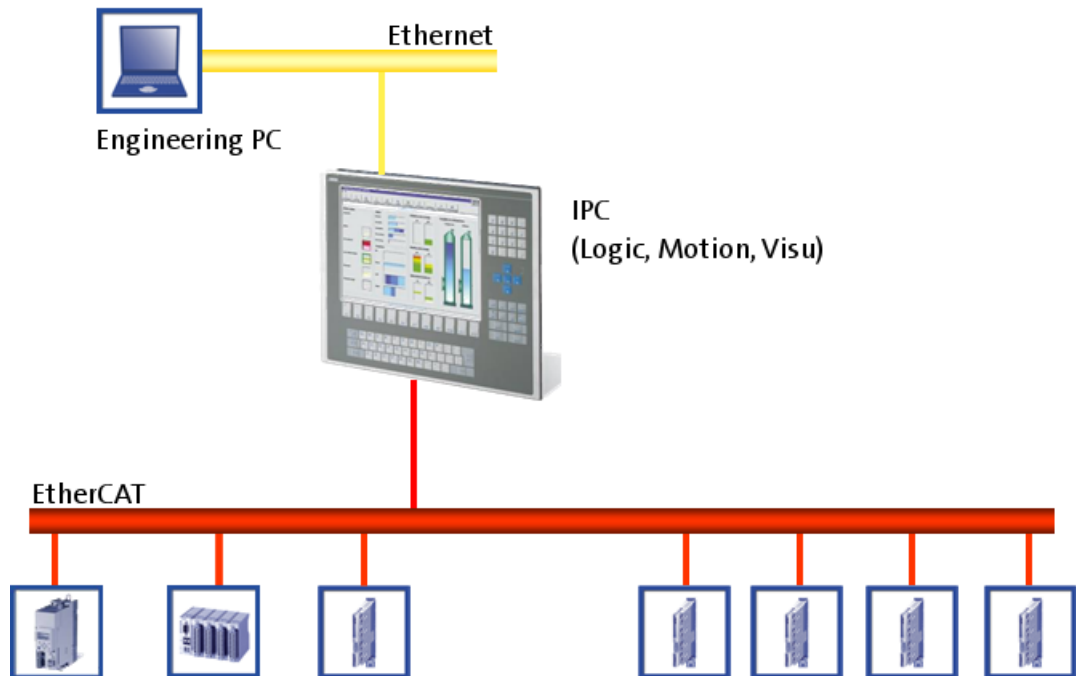
www.EtherCAT.org

4.1.1 Features

- ▶ EtherCAT is a high-performance bus system based on Ethernet.
- ▶ Thanks to the integrated synchronisation mechanisms via "distributed clocks", EtherCAT offers excellent real-time characteristics.
 - ▶ [Synchronisation with "Distributed clocks"](#) (36)
- ▶ EtherCAT provides a higher bandwidth compared to CANopen:
 - This enables motion and logic applications to be operated by the same bus.
 - The number of the nodes to be controlled is higher.
 - The maximally possible bus length is longer.
- ▶ EtherCAT can access all field devices via a common interface. Therefore, a division into Logic fieldbus and MotionBus is not required.

4.1.2 Structure of the EtherCAT bus system

Basic structure



Physical structure

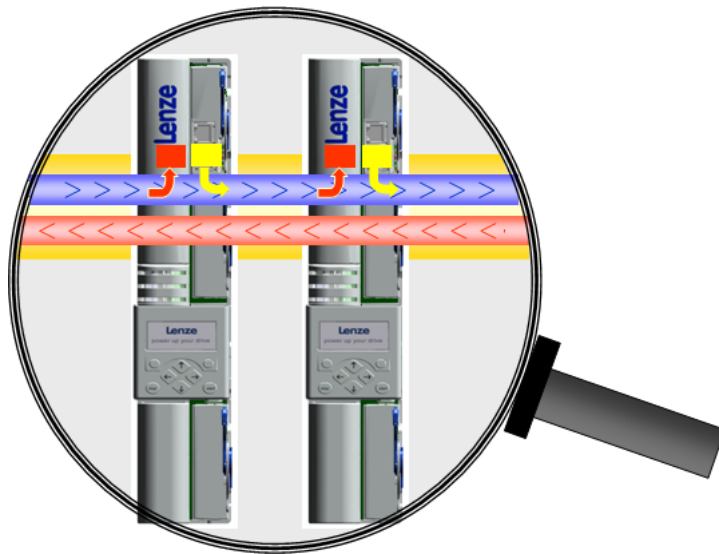
- ▶ An EtherCAT master can communicate with one or more nodes ("slaves").
- ▶ Internally, the EtherCAT bus has a ring topology. Since Ethernet cables are provided with a going and a return conductor within one cable, for the installer the topology seems to be a line. The last slave closes the ring.
- ▶ Switches, hubs or other infrastructure components known from the Ethernet standard must not be used because they impair the real-time performance.

4.1.3 Communication

Compared with the conventional Ethernet, the collision-free transfer of telegrams on the bus makes EtherCAT a real-time capable bus system.

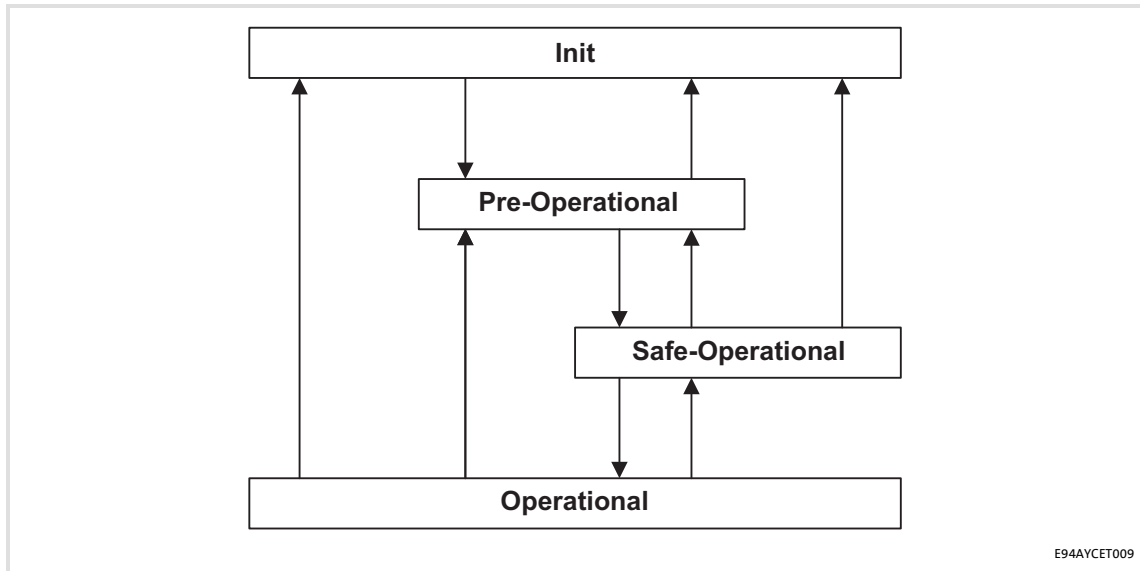
Communication is always initiated by the EtherCAT master, i.e. the industrial PC. A telegram sent by the master passes through all EtherCAT slaves. The last slave of the communication chain sends the telegram back to the EtherCAT master. On the way back, the telegram is directly sent to the EtherCAT master, without being processed in the slaves.

With EtherCAT, telegram processing completely takes place on the hardware level. The slaves take the data intended for them from the Ethernet frame and write their data back to the Ethernet frame. Every datagram can be passed on with a minimum delay.



4.1.3.1 The EtherCAT state machine

Before communication via EtherCAT is possible, the bus runs through the EtherCAT state machine during power-up. The following illustration shows the possible state change from an EtherCAT slave's point of view:



Status	Description
Init	- Initialisation phase - No SDO/PDO communication with the slaves - Device can be detected by fieldbus scan
Pre-Operational	- The fieldbus is active. - SDO communication (mailbox communication) is possible. - No PDO communication
Safe-Operational	- SDO communication (mailbox communication) is possible. - PDO communication: –The input data is transmitted to the master and evaluated. –The output data have the "Safe" state. It is not forwarded to the basic device.
Operational	- Normal operation –SDO communication –PDO communication –Fieldbus synchronisation has been successful (if used)



Note!

- Scanning of the EtherCAT fieldbus is possible in all states:
 - ▶ [Determining the physical EtherCAT configuration \(fieldbus scan\)](#) (47)
- SDO communication via the EtherCAT fieldbus is only possible when at least the *Pre-Operational* state has been reached.

4.1.3.2 Addressing of the slaves

The EtherCAT system uses two types of addressing for the slaves:

- ▶ Auto-increment addressing
- ▶ Fixed-address addressing

The auto-increment addressing is used by the master during the initialisation phase of the fieldbus. When the *Pre-Operational* state has been reached, the master uses the Fixed-Address addressing.

Auto-increment addressing

The auto-increment addressing is based on the bus topology. Each slave can be addressed by means of its physical position within the fieldbus.

- ▶ Slave 1 = address 0
- ▶ Slave 2 = address -1
- ▶ Slave 3 = address -2 etc.

The master transmits a telegram to the slave address. Each slave increments the address during the telegram cycle. A slave to which a telegram is addressed recognises the telegram by means of the address 0.

A configuration example is given under:

- ▶ [Determining the physical EtherCAT configuration \(fieldbus scan\)](#) (47)

Fixed-address addressing

With the fixed-address addressing, the slaves are addressed via the station address distributed by the master in the start-up phase.

In the EtherCAT bus topology in the »PLC Designer«, the first slave gets the address 1001, the second slave the address 1002, etc. The EtherCAT addresses cannot be changed.

The EtherCAT address of the master is 0. Master objects with this address can also be accessed via CoE.

Example

The first slave of a configuration obtains the following addresses:

- ▶ 0 by the auto-increment mode
- ▶ 1001 by the fixed addressing mode (default address of the first slave in the »EtherCAT Configurator«).

4.1.3.3 Working counter

Each EtherCAT datagram contains a working counter (WKC) which is incremented by each slave after the data have been processed successfully.

In every cycle, the control unit compares the expected value of the working counter with the value read back via the fieldbus. If the read-back value is lower than the expected value, the telegram has not reached all addressed slaves. The control unit recognises this and signals an error.

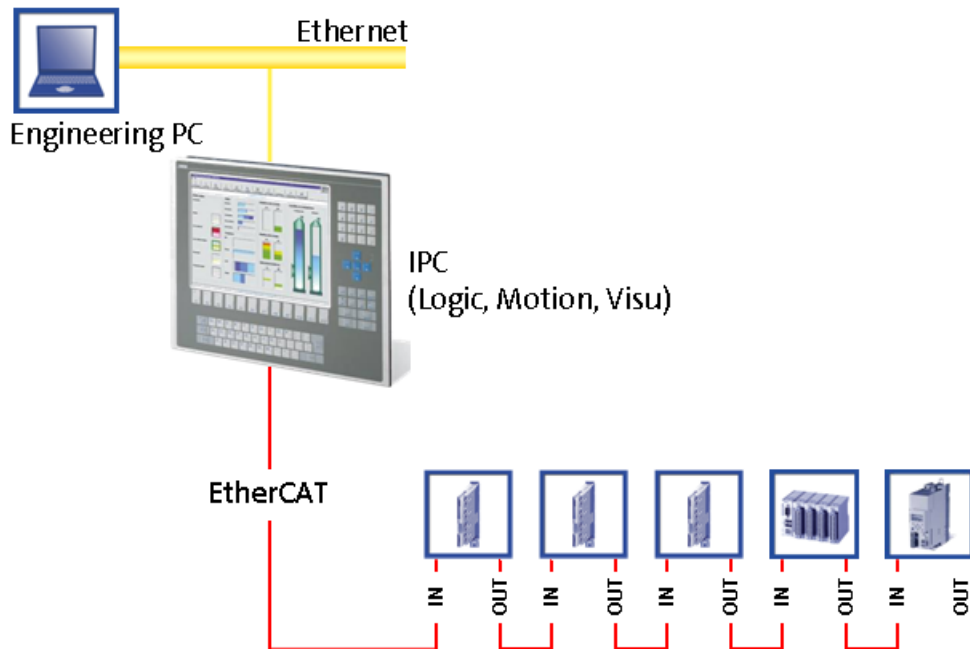
The working counter (WKC) can be used as a diagnostics option to check the processing of the EtherCAT telegrams by the slaves.

Example

- ▶ 10 slaves read/write process data in the *Operational* status
Expected value of the WKC: 10
- ▶ A cable break between the 8th and 9th slave causes the master to be unable to access slave 9 and slave 10:
 - Value of the restored WKC: 8
 - An error response is initiated in the control.

4.2 Required hardware components

4.2.1 The industrial PC - the central component



The industrial PC is the central component in the EtherCAT bus system:

- ▶ The industrial PC is the EtherCAT master.
- ▶ The industrial PC acts as EtherCAT gateway to be able to access the field devices from the engineering PC via Ethernet and EtherCAT.
- ▶ The devices must be connected in a line. To ensure that the system operates properly, the physical arrangement of the EtherCAT field devices must comply with the bus topology created in the »EtherCAT Configurator«.
- ▶ Each EtherCAT slave has two EtherCAT ports.
 - In contrast to Ethernet, one port is assigned as input, the other one as output.
 - Input (IN) and output (OUT) must not be reversed!
- ▶ A bus termination at the last slave is not required since the bus system at the last slave is terminated automatically.

4.2.2 Field devices

The Lenze control system supports the following EtherCAT-capable logic and motion components:

Field devices		EtherCAT bus	
		Logic	Motion
Industrial PCs	EL x1xx PLC	●	-
	EL x7xx	●	●
	CS x7xx	●	●
	CPC x7xx	●	●
Servo Drives 9400 1)	HighLine	●	
	Highline with CiA402	●	●
	PLC	●	
Inverter Drives 8400 2)	BaseLine	●	
	StateLine	●	
	HighLine	●	
	TopLine	●	
I/O system 1000	EPM-Sxxx	●	
ECS servo system 3)	ECSxE	●	
	ECSxS (Speed & Torque)	●	
	ECSxP (Posi & Shaft)	●	
	ECSxM (Motion)		●
	ECSxA (Application)	●	

1) With EtherCAT E94AYCET communication module

2) With EtherCAT E84AYCET communication module

3) With EtherCAT EMF2192IB communication module

Field devices of other manufacturers can be integrated as logic nodes if they provide a standard-compliant EtherCAT device description.

4.2.3 EtherCAT product codes

The product codes serve to assign device descriptions to the corresponding devices. Device descriptions can be installed via the device repository.

► [Importing missing field devices](#) (45 59)

► The product codes are part of the device ID.

► Structure of the device ID: *<Manufacturer ID>_<Productcode><Revision number>*

Identification	Meaning
Manufacturer ID	Clear identification for the manufacturer, for Lenze devices: 0x3B _{hex} (59 _{dec})
Product code	Product code of the product range/the device
Revision number	Revision number, consists of Major Revision (CANopen behaviour) and Minor Revision (device version)

If, for instance, a device available at the fieldbus without an installed device description is detected during a [Fieldbus scan with the »EtherCAT Configurator«](#) (47), a message with the device ID as hexadecimal value is displayed.

Attention! The device was not found in the repository	DeviceID: 3B_38079CD900000002
---	-------------------------------

In the example, the device description for a Lenze Servo Drive 9400 HighLine, actuator – speed, is not installed (0x38079CD9_{hex} = 940023001_{dec}).

Product codes for Servo Drives 9400

Product code [dec]	Meaning
9 4 0 0 2 1	Servo Drive 9400 in general
9 4 0 0 2 2	Servo Drive 9400 StateLine
9 4 0 0 2 3	Servo Drive 9400 HighLine
9 4 0 0 2 4	Servo Drive 9400 TopLine
9 4 0 0 2 5	Servo Drive 9400 PLC
9 4 0 0 2 6	Servo Drive 9400 V/R (regenerative power supply module)

Applications:

0	0	0	Empty application
0	0	1	Actuating drive speed
0	0	2	Actuating drive torque
0	0	3	Electronic gearbox
0	0	4	Synchronism with mark synchronisation
0	0	5	Table positioning
0	0	6	Positioning sequence control
0	0	7	PLC application
0	0	8	Reserved
...	...	...	...
0	9	9	Reserved
1	x	x	Reserved for device profiles
1	0	1	CiA402
2	x	x	Reserved for Lenze applications
2	0	1	Regenerative power supply module application

Product codes for Inverter Drives 8400

Product code [dec]	Meaning
8 4 0 0 2 1	Inverter Drive 8400 BaseLine
8 4 0 0 2 2	Inverter Drive 8400 StateLine
8 4 0 0 2 3	Inverter Drive 8400 HighLine
8 4 0 0 2 4	Inverter Drive 8400 TopLine

Product codes for the I/O system 1000

Product code [dec]	Meaning
1 3 0 0	I/O system EPM-S130

Product codes for the ECS servo system

Product code [dec]	Meaning
2 1 9 2 0 7 0 0	ECSxA axis module "Application"
2 1 9 2 0 7 0 1	ECSxM axis module "Motion"
2 1 9 2 0 7 0 2	ECSxP axis module "Posi & Shaft"
2 1 9 2 0 7 0 3	ECSxS axis module "Speed & Torque"
2 1 9 2 0 7 1 1	ECSxE power supply module

4.2.4 EtherCAT hardware for the industrial PC

MC-ETC communication card

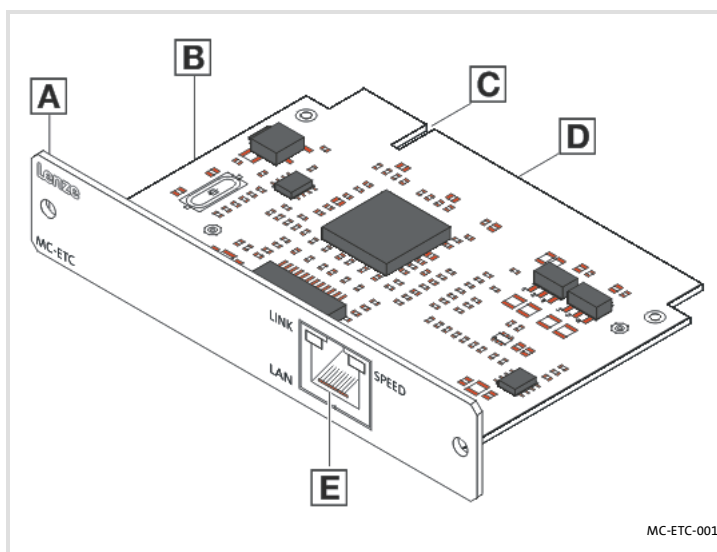
The MC-ETC communication card is a plug-in card for connecting an industrial PC to an EtherCAT network.



Note!

In case of a correct connection to the field devices, the LEDs of the communication card are lit.

► [Connection RJ45 socket / LEDs](#) (34)



A Front panel

B Board

C Coding

D Connection

E EtherCAT connection

► [Connection RJ45 socket / LEDs](#) (34)

► [Technical data of the MC-ETC communication card](#) (34)

Possible applications

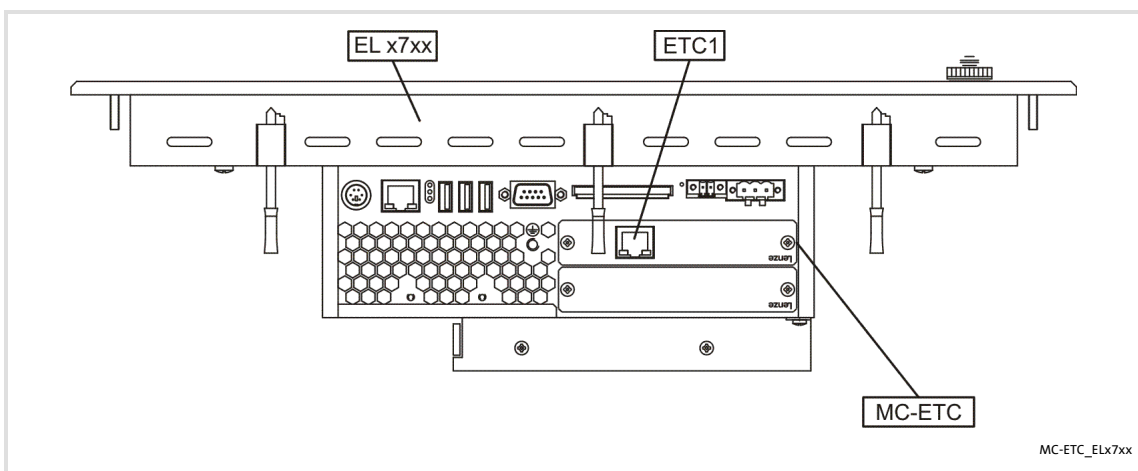


Note!

The industrial PC only supports one communication card MC-ETC!

Industrial PC	Can be combined with CANopen	Communication card for CANopen
EL x7xx	yes	MC-CAN2
CS x7xx	no	
CPC 2700	yes	MC-CAN2

Example: Industrial PC EL x7xx with MC-ETC

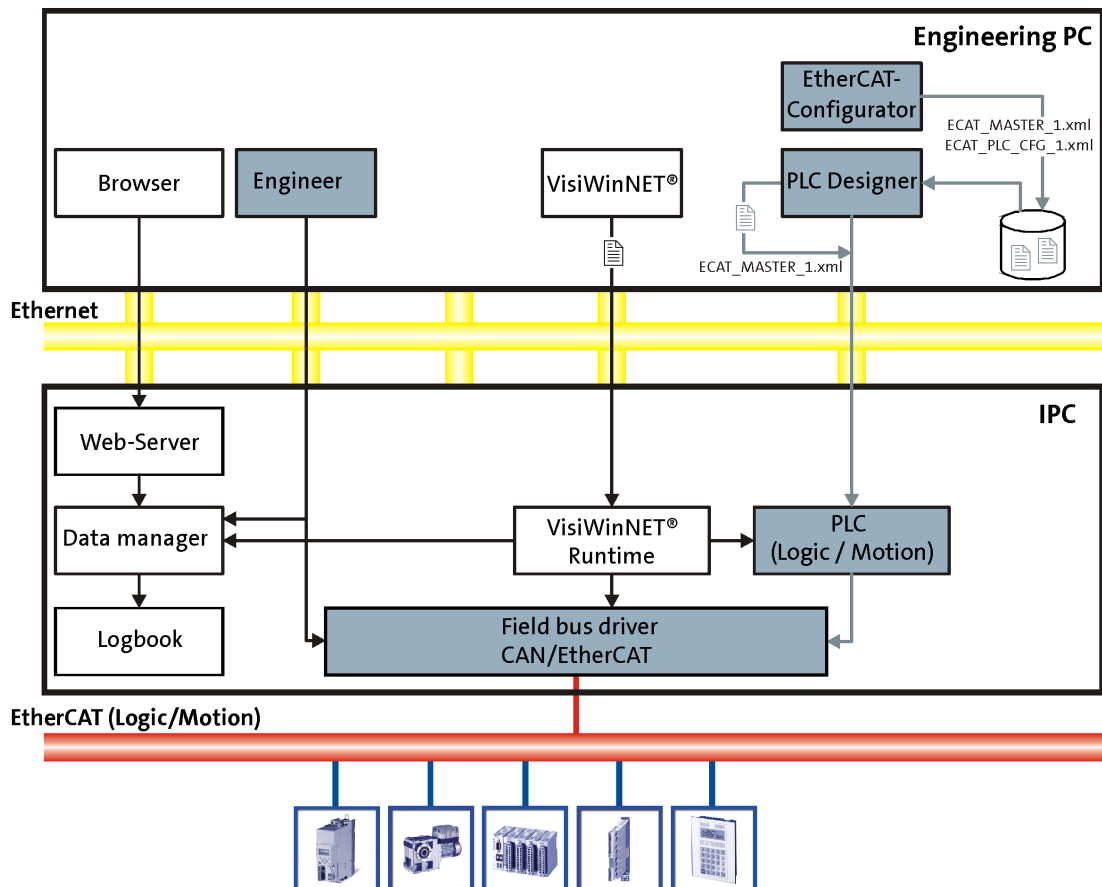


Legend

EL x7xx	Industrial PC of the EL x7xx series
ETC1	EtherCAT network connection
MC-ETC	MC-ETC communication card

4.3 Required engineering tools

- The engineering tools required for configuration and parameter setting are installed on the engineering PC.
- »Engineer«, »EtherCAT Configurator« and »PLC Designer« are engineering PC tools which are independent of each other.
- The EtherCAT bus, the industrial PC and the EtherCAT-capable field devices are configured with the engineering tools highlighted in grey.



Brief description of the engineering tools



L-force »Engineer«

With the »Engineer«, you can ...

- ▶ parameterise, configure and diagnose ...
 - Servo Drives 9400;
 - Inverter Drives 8400;
 - the I/O system 1000.
- ▶ access the supported field devices via the gateway function of the industrial PC.



»Global Drive Control« ((GDC))

With (GDC) you can ...

- ▶ parameterise, configure and diagnose controllers which are not supported by the »Engineer« (e.g. devices of the ECS servo system).
- ▶ access the supported field devices via the gateway function of the industrial PC (not with PROFIBUS).



L-force »PLC Designer«

The »PLC Designer« is needed to ...

- ▶ create the control program for the industrial PC;
- ▶ transfer the PLC projects to the industrial PC.

4.4 Interaction of the components

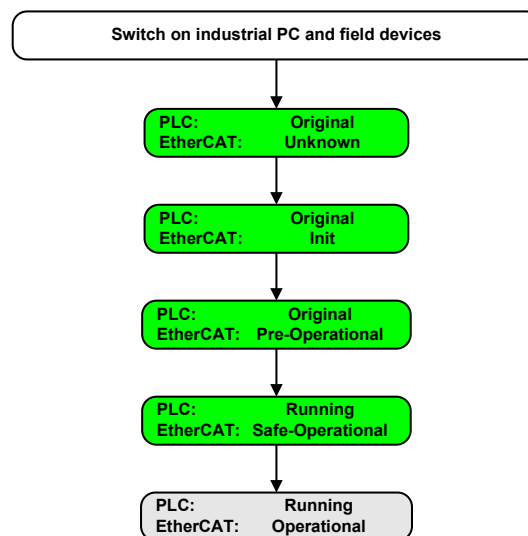
4.4.1 The state machine of the Lenze control technology

In the Lenze control technology, the states of the PLC and the EtherCAT fieldbuses are coupled. The PLC controls the fieldbus.

After switch-on the system automatically powers up if the following conditions are fulfilled:

- ▶ The IPC is provided with the configuration file for the EtherCAT master (master-XML file), the contents of which corresponds to the real bus topology.
- ▶ The IPC is provided with an executable PLC boot project.
- ▶ The slaves at the fieldbus can be accessed.

The following illustration shows the linkage of the states in the state machine of the Lenze control technology when the conditions for the automatic acceleration of the system are fulfilled (boot project with EtherCAT BusInterface and EtherCAT master configuration):



Legend

	Transitional state, automatic change to next state
	Stationary state, change to next state by external actions
PLC	State of the PLC
EtherCAT	State of the EtherCAT fieldbus

Explanation of the transitions during system start

While a state is passed through, different tests are carried out (e.g. it is tested whether the physical topology complies with the configuration). If the tests are successful, the system automatically changes to the next state.

Status		What happens?	What is tested?
PLC	EtherCAT		
Origin	Unknown	The system starts.	Is a master XML file available?
Origin	Init	EtherCAT is initialised. Master XML file is imported. Bus scan is executed	Does the imported master XML comply with the result of the bus scan?
Origin	Pre-Operational	EtherCAT is active. SDO communication is possible.	Is an executable boot project available?
Running	Safe-Operational	The PLC program is being loaded. The PLC is running. The inputs are transferred, the outputs are still in a safe state.	Are all inputs ok and initialised?
Running	Operational	The system is running.	

Detailed information about the possible bus states and the required commissioning steps can be found here:

► [Commissioning of the system](#) (42)

4.4.2 Communication between engineering PC and field devices

For commissioning of the field devices, an online connection is required between the engineering PC and the corresponding field device. Depending on the state of the EtherCAT bus, there are two options:

- ▶ [EtherCAT bus not in operation](#) (📖 32)
- ▶ [EtherCAT bus in operation \(gateway function\)](#) (📖 33)

4.4.2.1 EtherCAT bus not in operation

You can communicate serially or via CANopen.

Condition:

- ▶ Serial communication:
 - You require the E94AZCUS diagnostic adapter.
 - Field device and engineering PC (USB interface) must be connected via the diagnostic adapter.
- ▶ Communication via CANopen
 - You required the EMF2177IB USB system bus adapter .
 - Field devices and the engineering PC are connected via the system bus adapter - either via a point-to-point connection or via the bus system.

Advantage:

- ▶ Quick option of communication without commissioning of the EtherCAT bus.

Disadvantage:

- ▶ You require additional hardware.



Tip!

As soon as the fieldbus has been commissioned and at least achieved the *Pre-Operational* state, this communication path comes second. We recommend to commission the EtherCAT bus as soon as possible to be able to use the gateway function.

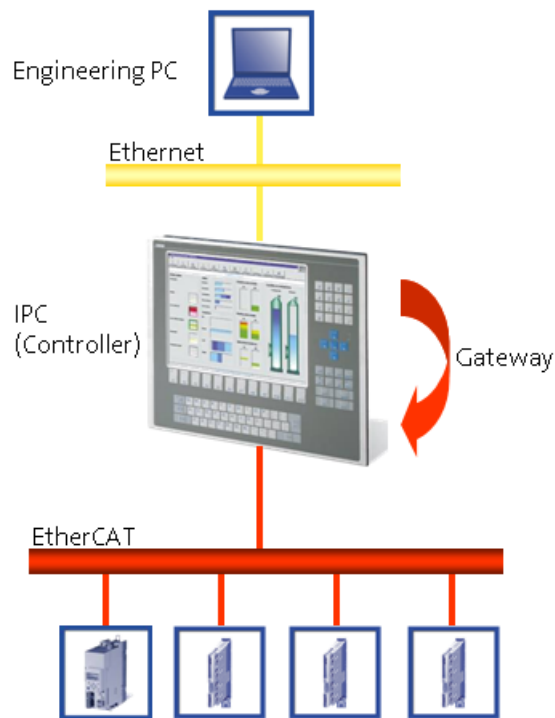
4.4.2.2 EtherCAT bus in operation (gateway function)

You directly communicate via EtherCAT and use the industrial PC as gateway.



Note!

A PLC program does not need to run to be able to use the gateway function.



Condition

- ▶ The bus configuration has been created using the »PLC Designer« and corresponds to the hardware configuration.
- ▶ The bus configuration has been loaded onto the industrial PC using the »PLC Designer« and is active.
- ▶ The fieldbus state is at least *Pre-Operational*.

Advantage:

- ▶ You do not require any additional hardware.
- ▶ The entire communication (process data, parameter data, and diagnostic data) are transferred at the same time using a single bus connection.

5 Technical data

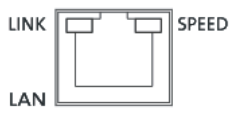
5.1 General data of the EtherCAT bus

Field	Values
Communication medium	S/FTP (Screened Foiled Twisted Pair, ISO/IEC 11801 or EN 50173), CAT5e
Network topology	Line
Number of nodes	Max. 65535 (in the entire network)
Max. cable length	100 m between two nodes
Baud rate	100 Mbit/s
EtherCAT module	Direct mode
Communication profile	CoE (CANopen over EtherCAT)
Synchronisation	Distributed clocks

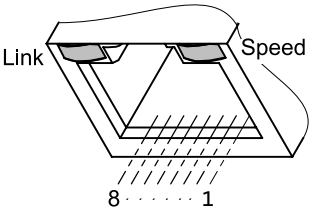
5.2 MC-ETC communication card

Field	Values
Possible baud rate	100 Mbit/s
Type within the network	Master
Connection	RJ45 socket in accordance with EN 50173

Connection RJ45 socket / LEDs

View	Description	Cable type
 MC-ETH-001	EtherCAT connection - LED "Link": – On: connection ok – Blinking: data exchange - LED "Speed": – Green: 100 MBit/s	- CAT5e S/FTP network cable (recommended) in accordance with ISO/IEC 11801 or EN 50173 - Cable length: max. 100 m

Pin assignment

RJ45 socket	Pin	Assignment
 E94YCEP018	1	Tx +
	2	Tx -
	3	Rx +
	4	-
	5	-
	6	Rx -
	7	-
	8	-

5.3 Communication times and drive-specific data

Field	Values
User data per frame	1344 bytes
Process data words (PZD) for Servo Drives 9400 HighLine	max. 32 words (64 bytes)
Parameter data (SDO) transfer	Max. 128 bytes
Permissible EtherCAT cycle times	1 ... 10 ms
Max. number of Servo Drives 9400 HighLine per frame	User data of the frame (1344 bytes) divided by the process data length resulting from setpoints and actual values of the drives: - for 32 Tx/Rx bytes: 1344 bytes / 64 bytes = 21 drives - for 16 Tx/Rx bytes: 1344 bytes / 32 bytes = 42 drives
Total signal runtime for a cycle time of 1ms Drive → Control → Drive	5 ms
Runtime of the setpoints Control → Drive	2 ms
Runtime of the actual values Drive → Control	3ms
Cross communication	Not possible
Cycle synchronisation with locked PLL (jitter)	+/-1 µs

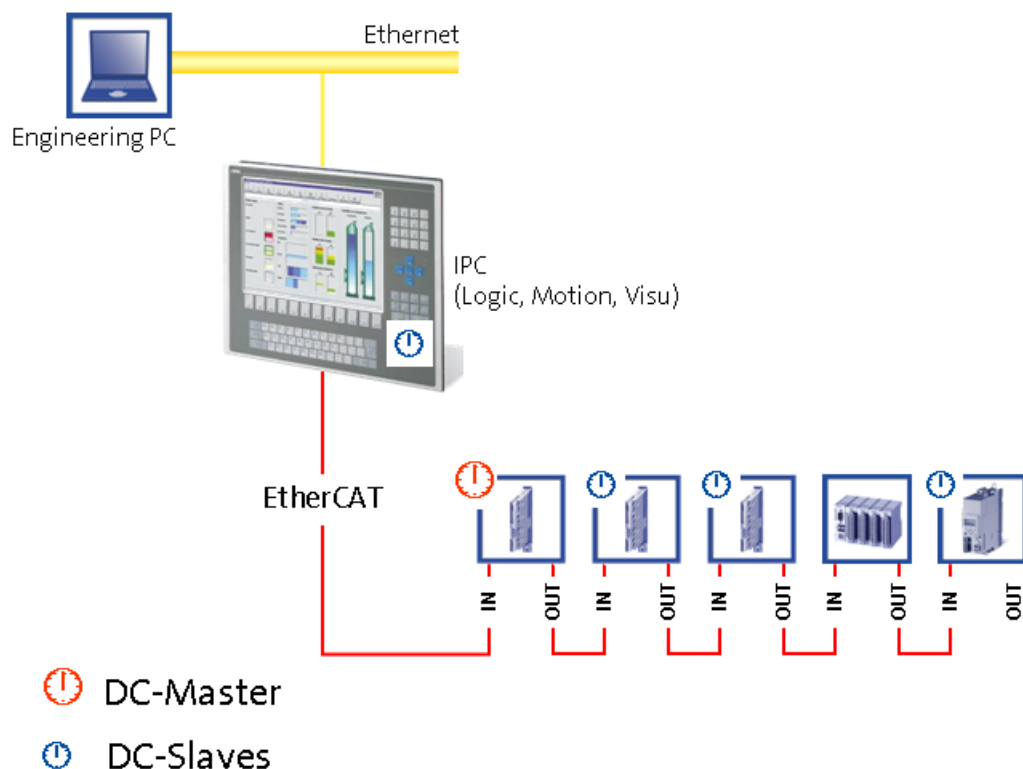
6 Synchronisation with "Distributed clocks"

The "Distributed clocks" (DC) function enables an exact time adjustment for applications where several auxiliary axes carry out a coordinated movement at the same time. The data is accepted synchronously with the PLC program. With the DC synchronisation, all slaves are synchronised with a reference clock, called the "DC master".



Note!

- Motion applications always require DC synchronisation.
- DC synchronisation can also be used for Logic applications.
- Some slaves do not support the DC functionality.
 - To be able to use the DC function, the first slave connected to the EtherCAT master (IPC) must be **DC-master-capable**.
 - In the arrangement of the slaves following then, DC-capable and non-DC-capable devices can be mixed.
- The first EtherCAT node connected to the IPC must be the **DC master** which provides the other EtherCAT nodes (including the IPC) with the exact time.



The settings for the DC synchronisation are made with the »EtherCAT Configurator«.

- ▶ [Adjusting task cycle time and DC cycle time](#) (38)
- ▶ [Setting the DC synchronisation with the »EtherCAT Configurator«](#) (39)

6.1 Synchronous communication

The DC synchronisation provides for a phase-synchronous operation of master and slaves: Within one bus cycle the setpoints are accepted and the actual values are detected in the fieldbus at exactly the same time.

If the control (IPC) is synchronous to the distributed clocks, the data (actual values) collected by the slave are assigned to the master at the end of the bus cycle and data from the master (setpoints) are sent to the slaves for processing.

When the next DC synchronisation event occurs, the data are accepted.



Note!

State change and DC synchronisation for Servo Drives 9400 HighLine

During the state change from *Operational* to *Pre-Operational*, the DC synchronisation is deactivated ($C13883/C14883 = 0$).

In order to re-activate the sync pulses adjust your PLC program in the following way:

- Call the function block **ResetAxisGroup**.
 - In this way, the EtherCAT fieldbus is reinitialised.
 - The DC synchronisation is active again ($C13883/C14883 = 1$).

6.2 Adjusting task cycle time and DC cycle time

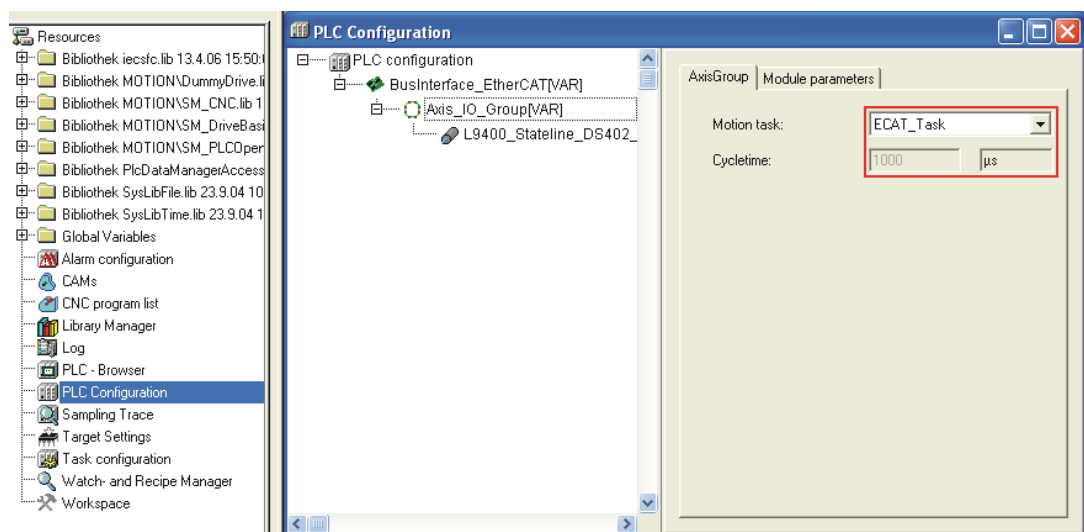
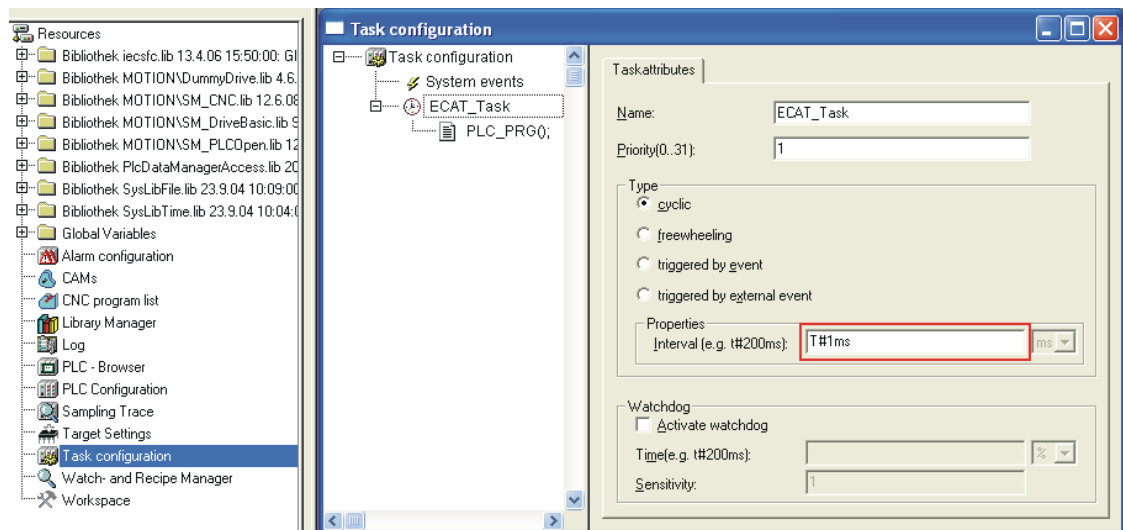
The industrial PC is the fieldbus master in the EtherCAT network. The clock pulse of the EtherCAT bus system is determined by the cycle time of the task which is assigned to the drives (slaves) in the »PLC Designer«.

The task settings in the »PLC Designer« only support integer millisecond cycles and the smallest possible bus cycle is 1 millisecond. This cycle time can be defined via the **task configuration** of the »PLC Designer«.



Note!

- The DC cycle time to be set in the »EtherCAT Configurator« must comply with EtherCAT task cycle time set in the »PLC Designer«.
- Select the cycle times, according to the technical data, between 1 and 10 ms.



6.3 Setting the DC synchronisation with the »EtherCAT Configurator«



Note!

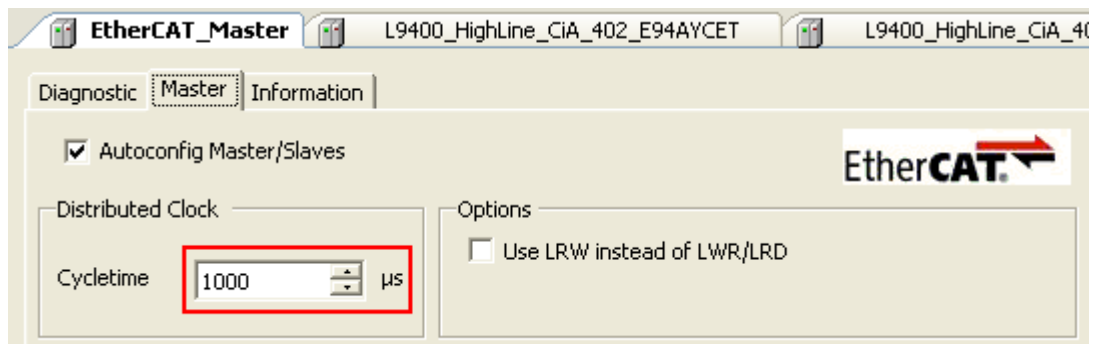
The manual configuration of the slave DC features requires a detailed knowledge of EtherCAT and the field device. Thus, DC settings should only be made by experts.

A faulty configuration can cause maloperation and negative influences on the system.



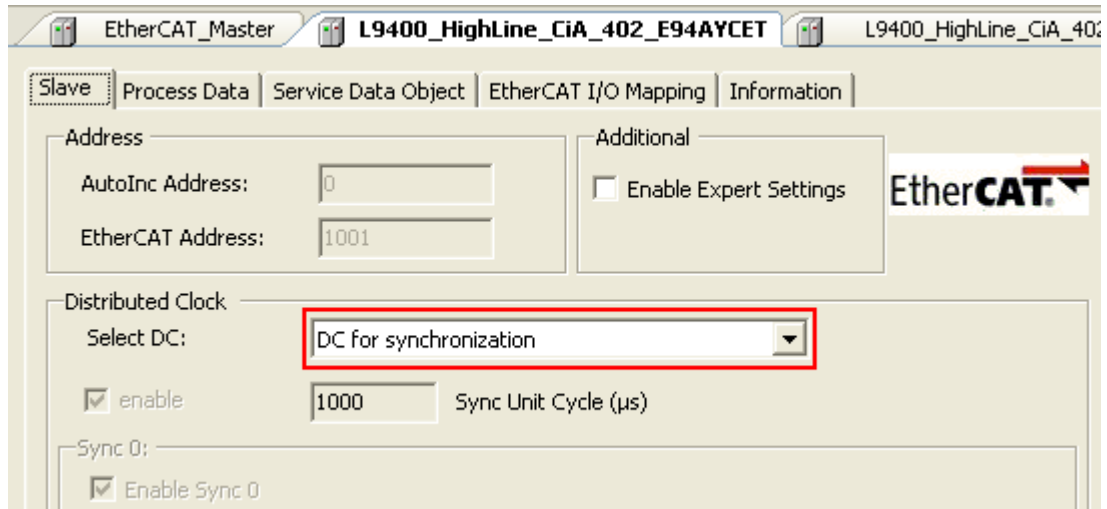
How to set the DC synchronisation:

1. Set the *Distributed clocks (DC)* cycle time at the **EtherCAT Master**:



- The DC cycle time to be set in the »EtherCAT Configurator« must comply with the EtherCAT task cycle time set in the »PLC Designer«.
- Select the cycle times, according to the technical data, between 1 and 10 ms.
- The (basic) cycle time set here is valid for all Logic and Motion nodes synchronised by distributed clocks.
- For field devices with communication modules (e.g. Servo Drives 9400 or Inverter Drives 8400), a synchronisation source must be selected via code **C01120**. If the DC setting and the selection of the sync source differ from each other (C01120 = MXI1 and "DC nused"), the devices cannot be set to the *Operational* state.
- For **Servo Drives 9400**, code **C13892/C14892 = 1** must be set (process data mode = "deterministic mode").

2. Activate the DC functionality for the **first** slave (DC master) connected to the bus master (select "DC for synchronization"):



3. Also activate the DC functionality (select "DC for synchronization") for all other slave devices which are to use the DC synchronisation.



Note!

Maintain all other basic DC settings for the slaves to ensure a correct DC synchronisation.

6.4 Check of the DC synchronicity

The DC synchronicity can only be checked in the *Operational* bus state.

You can check the DC synchronicity via the "ECAT DC: Status" parameter (code [C1082/1](#) / [C1582/1](#)) or via the [Notifications](#) ([120](#)) `bEC_NOTIFY_DC_STATUS` and `bEC_NOTIFY_DC_SLV_SYNC`.

7 Commissioning of the system

This chapter provides information about how to commission the Lenze control system with EtherCAT.

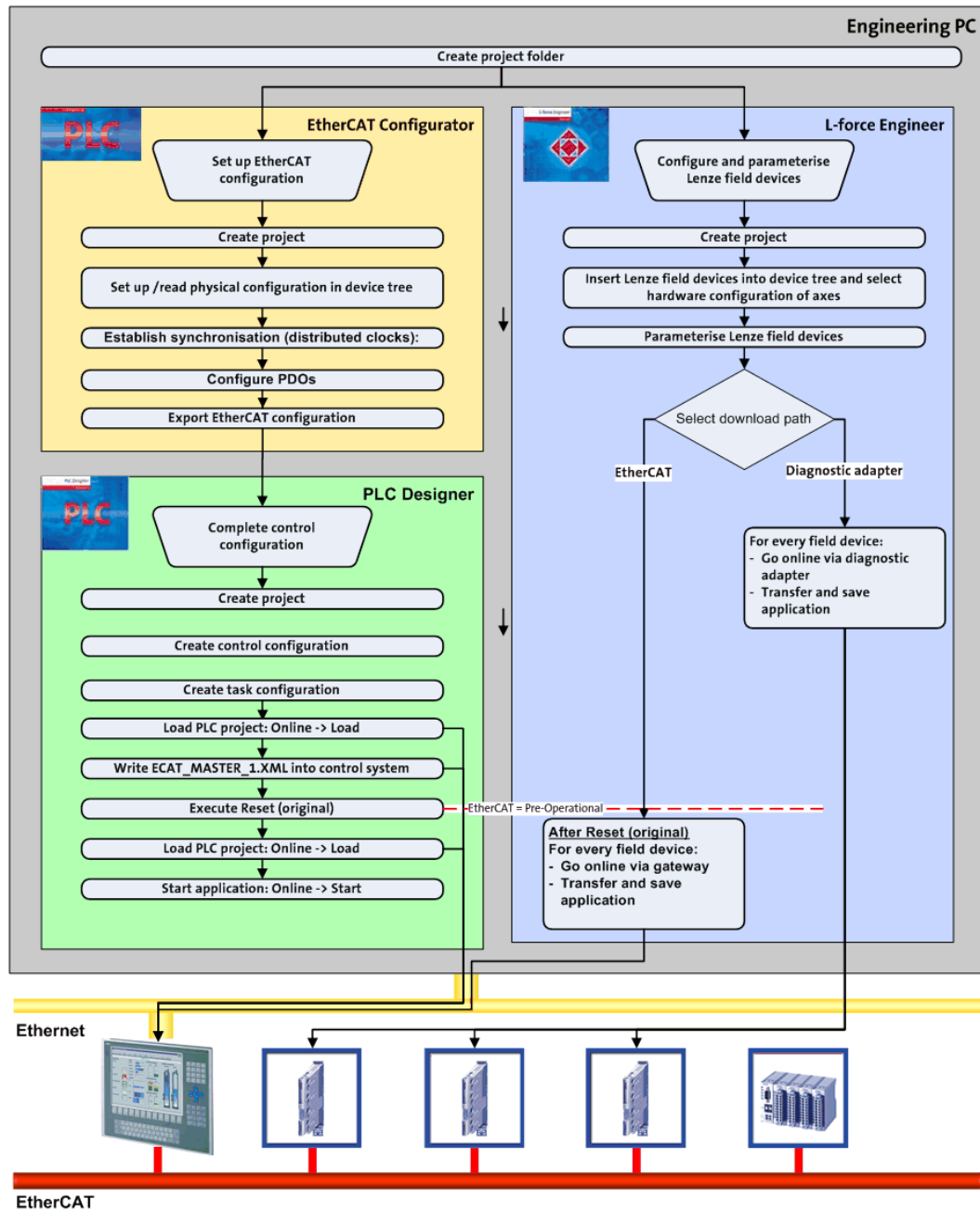
Commissioning of the system requires the following Lenze engineering tools:

- ▶ »EtherCAT Configurator«
- ▶ »PLC Designer«
- ▶ »Engineer«

An overview of the commissioning steps is given in the next section [Overview of commissioning steps](#) (43). Follow the instructions of these sections step by step in order to commission your system.

At the end of this chapter you will find a chart showing a [Detailed overview of the commissioning steps](#) (90) with regard to the different Lenze engineering tools.

7.1 Overview of commissioning steps



The main commissioning steps are listed in the following table:

Step	Activity	Software used
1.	Installing field devices (📖 45)	
2.	Creating a project folder (📖 46)	
3.	Fieldbus scan with the »EtherCAT Configurator« (📖 47)	»EtherCAT Configurator«
	Fieldbus scan with the »scandf« command line tool (📖 51)	Command line tool »scandf«
4.	Inserting devices available on the fieldbus into the »EtherCAT Configurator« project (📖 55)	»EtherCAT Configurator«
	Creating the configuration in the »EtherCAT Configurator« (📖 56)	
5.	Export of EtherCAT configuration (📖 67)	»EtherCAT Configurator«
6.	Configuration in the »Engineer« (📖 54)	»Engineer«
8.	Configuration in the »PLC Designer« (📖 68)	»PLC Designer«
9.	Loading the control configuration onto the IPC (📖 78)	»PLC Designer«
10.	Loading the master configuration onto the EtherCAT master (📖 79)	»PLC Designer«
11.	Loading and starting the PLC program (📖 79)	»PLC Designer«

7.2 Detailed commissioning steps

In the following sections, the individual commissioning steps are described.



More detailed information about how to work with the Lenze engineering tools can be found in the corresponding manuals and online helps.

7.2.1 Planning the bus topology

Before you start to set up an EtherCAT network, first create a plan of your EtherCAT bus.

► [Brief description of EtherCAT](#) (16)



How to plan the bus topology for your configuration:

1. Create an overview of the planned EtherCAT network with all field devices to be integrated.
2. Start with the industrial PC.
3. Add the other field devices below.
4. You have to distinguish between the following two cases:
 - **Operation without synchronisation via distributed clocks:**
DC synchronisation is not required if exclusively Logic field devices are to be operated on the network. The sequence of the field device interface connections on the bus can freely be selected.
 - **Operation with synchronisation via distributed clocks:**
DC synchronisation is required if Motion and Logic field devices are to be operated on the network. The first node connected to the control IPC must be capable of being a DC master. The sequence of the other Logic and Motion field device interface connections can freely be selected.

7.2.2 Installing field devices

For the installation of a field device, follow the mounting instructions for the respective device.



Note!

- The EtherCAT interfaces of all devices must be wired according to the planned topology. The inputs (IN) and outputs (OUT) must not be reversed because otherwise the topology changes.
► [Communication](#) (18).
- The structure of the EtherCAT configuration must be identical with the order of the devices in the »EtherCAT Configurator«.
- The master automatically assigns the node addresses to the slaves. Therefore, a manual address assignment is not required.

7.2.3 Creating a project folder

Create a project folder on the engineering PC.

Use this project folder to store the below data generated in the different project configuration steps:

- ▶ The project file created in the »EtherCAT Configurator«
- ▶ The configuration files exported from the »EtherCAT Configurator«:
 - ECAT_PLC_CFG_1.XML
 - ECAT_MASTER_1.XML
- ▶ The project data created in the »Engineer«
- ▶ The project file created in the »PLC Designer«



Tip!

Create a separate project folder for every EtherCAT configuration and store the project and configuration files in this folder.

7.2.4 Determining the physical EtherCAT configuration (fieldbus scan)

You can execute a fieldbus scan with the »EtherCAT Configurator« on the IPC in order to check the physical EtherCAT configuration. Alternatively, the command line tool »scandf« ([51](#)) is also available on the IPC.



Note!

- Scanning of the EtherCAT fieldbus is also possible without an appropriate EtherCAT configuration.
- Communication to field devices via the EtherCAT fieldbus is only possible if at least the *Pre-Operational* state has been reached.

7.2.4.1 Fieldbus scan with the »EtherCAT Configurator«

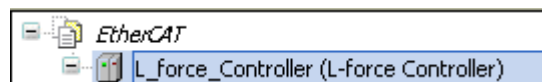
The »EtherCAT Configurator« offers the possibility to execute an online search for devices which are connected to the EtherCAT fieldbus.

In order to search for devices you must first establish an online connection of the »EtherCAT Configurator« with the industrial PC.

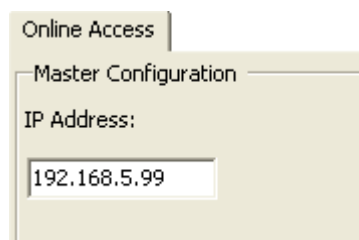


How to execute a fieldbus scan with the »EtherCAT Configurator«:

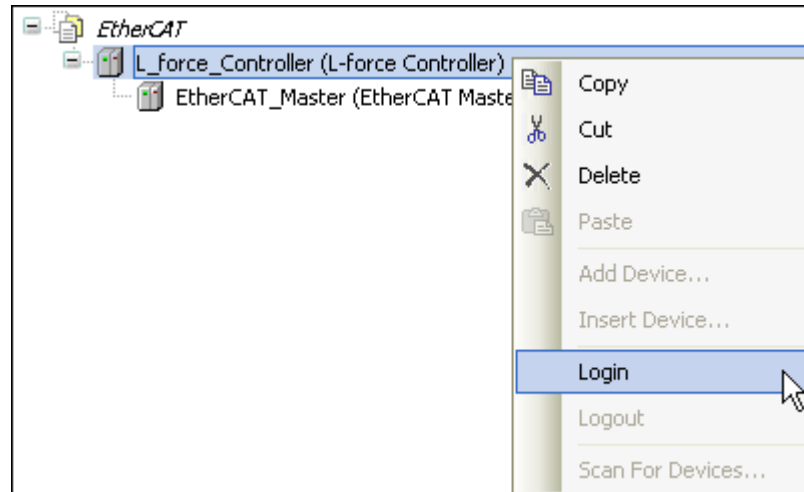
1. Enter the IP address of the industrial PC:
 - Select the *L-force Controller* by double-clicking in the device tree:



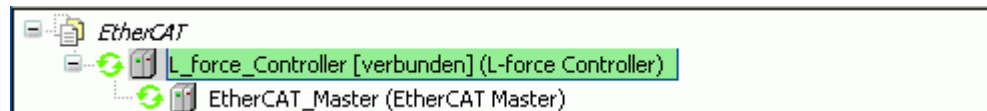
- Enter the IP address of the industrial PC which should serve as control unit of the configuration into the *Online access* configuration dialog box in the *Master configuration* area:



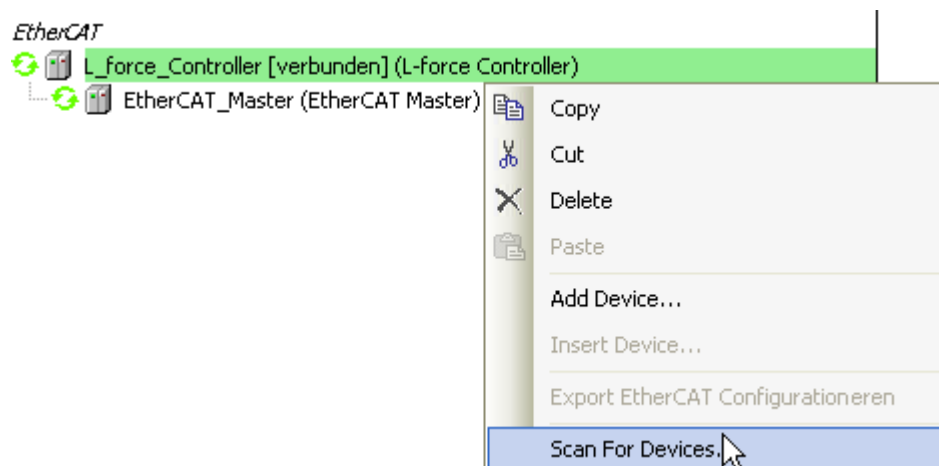
2. Execute the **Login** command in the context menu of the *L-force Controller*:



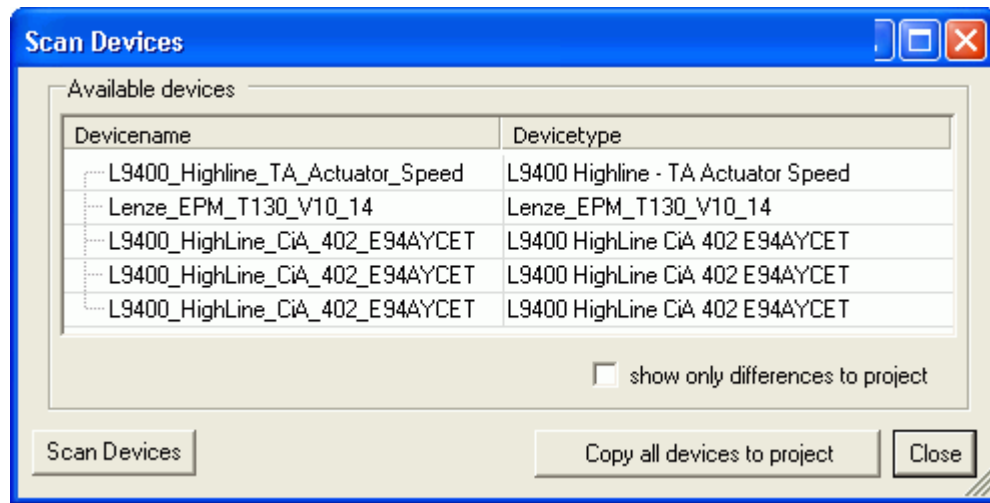
After a successful login, the *L-force Controller* (the industrial PC) is connected to the engineering PC:



3. After a successful login, execute the **Start Search** command in the context menu of the *L-force Controller* :



The »EtherCAT Configurator« determines the EtherCAT nodes available at the fieldbus. In the *Start Search* dialog box, the devices are listed according to the physical order at the fieldbus:



Further information on the *Start Search* dialog box can be found here:

► [Inserting devices available on the fieldbus into the »EtherCAT Configurator« project](#) (55)

Missing device descriptions

If a device available at the fieldbus is not present in the device repository, an error message within the *Start Search* dialog box will inform you about it:

Attention! The device was not found in the repository	DeviceID: 3B_38079CD900000002
---	-------------------------------

- ▶ The device cannot be interpolated into the project as the corresponding device description has not been installed.
 - ▶ In order to install the device in the device repository, the corresponding device description file from the manufacturer is required. The device identification (device ID) can be helpful to identify the device (see also [EtherCAT product codes](#) (📖 24)).
- ▶ [Importing missing field devices](#) (📖 59)

7.2.4.2 Fieldbus scan with the »scanfd« command line tool

Fieldbus scanning command

Command: `scanfd <interface> /option`

The interface for the EtherCAT slaves to be scanned on the fieldbus is called `ECAT`.

Option	Output on the console
<code>/i</code>	Determine available interfaces
<code>/t</code>	Display device information for every network node.
<code>/f</code>	Display additional device parameters for every device if available.
<code>/n</code>	Suppress prompt before execution of the fieldbus scan.



How to carry out a fieldbus scan:

1. Establish a Telnet connection to the IPC.
More detailed information is provided in the documentation for the IPC.
2. Call `scanfd.exe` via the command line box of the IPC to scan the fieldbus.
3. Determine the available interfaces of the IPC:
 - Enter `scanfd /i`.

Output example:

```
\> scanfd /i
The following interfaces are available:
ECAT
CAN2:
CAN1:
\> _
```

The fieldbus scanner displays the available interfaces:

- `ECAT` for the EtherCAT-capable field devices,
- `CAN1` and `CAN2` for the CAN field devices.

4. Determine the field devices physically available at the ECAT interface:

- `scanfd ecat /t /f /n`

Output example:

```
\> scanfd ecat /t /f /n
5 devices at interface 'ecat'!
DeviceID=0
Vendor=Lenze
ProductCode=#x38079cd9
RevisionNumber=#x00000002
SerialNumber=#x0000005c

DeviceID=-1
Vendor=Lenze
ProductCode=#x00000001
RevisionNumber=#x0000000e
SerialNumber=#x00000000

DeviceID=-2
Vendor=Lenze
ProductCode=#x38079d3d
RevisionNumber=#x00000002
SerialNumber=#x000000e2

DeviceID=-3
Vendor=Lenze
ProductCode=#x38079d3d
RevisionNumber=#x00000002
SerialNumber=#x000000cc

DeviceID=-4
Vendor=Lenze
ProductCode=#x38079d3d
RevisionNumber=#x00000002
SerialNumber=#x00000000

\>
```

After executing the `scanfd.exe` file, the console displays the result of the fieldbus scan. The first output line contains the number (x) of slaves found on the fieldbus: x devices at interface 'ecat'!

The further output lines provide information on each slave:

```
DeviceID=0
Vendor=Lenze
ProductCode=#x38079cd9
RevisionNumber=#x00000002
SerialNumber=#x0000005c
```

Information about the field devices

- The individual slaves are generally listed according to the auto-increment addressing scheme in their physically order.
- The first slave connected to the master has the node address 0. The second slave has the negative node address -1. The other slaves have negative node addresses (-n) corresponding to their positions on the fieldbus.

Information	Function
DeviceID	Node address of the slave: • Display of the auto-increment address. • The first slave has the node address 0.
Vendor	Name of the manufacturer
ProductCode	Product designation in hexadecimal format (see also EtherCAT product codes (□ 24)) Examples: #x38079cd9: Lenze Servo Drive 9400 HighLine TA speed acuating drive #x38079d3d: Lenze Servo Drive 9400 HighLine CiA402
RevisionNumber	Version number in hexadecimal format
SerialNumber	Serial number in hexadecimal format

Troubleshooting

If the fieldbus scan does not find any field devices at the selected interface, the following error message appears:

```
> scanfd ecat /t /n
No devices detected at interface 'ecat'!
Interface Status: unknown
> _
```

**How to check the physical configuration:**

1. Analyse the messages in the [Logbook of the IPC](#) (□ 140). ► [IPC logbook messages](#) (□ 142)
2. Check the wiring.

7.2.5 Configuration in the »Engineer«

The »Engineer« is used to configure and parameterise the Lenze field devices connected to the EtherCAT bus.



Note!

PDO mapping settings

The mapping required for a cross communication must be created in the »EtherCAT Configurator«/»PLC Designer«.

During start-up of the PLC, the complete configuration/PDO mapping is written into the EtherCAT slaves. Mapping entries, e.g. from the »Engineer«, are not overwritten.

▶ [Executing PDO mapping](#) (📖 65)

▶ [Editing the EtherCAT I/O image](#) (📖 66)



How to configure the drives in the »Engineer«:

1. Create a new »Engineer« project or open an already existing project.
2. Insert the Lenze field devices in the device tree and select the hardware configuration of the axes.

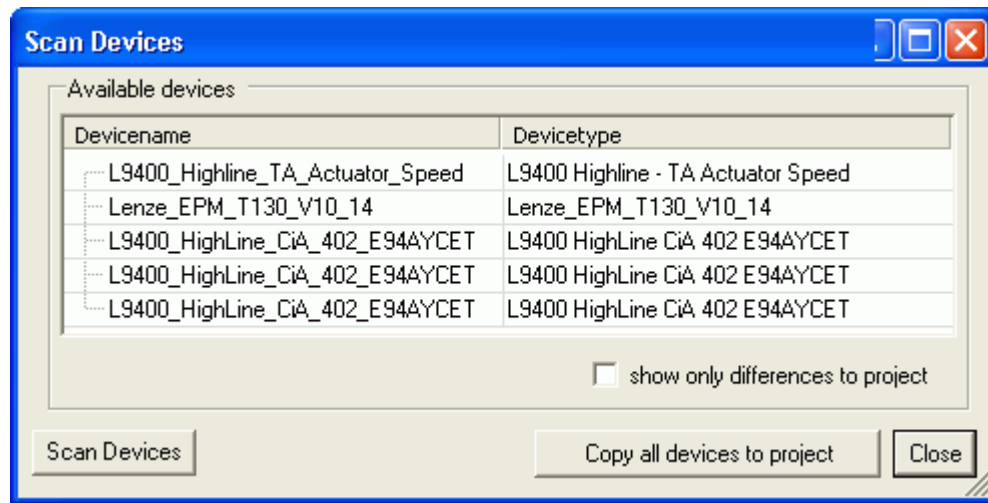
For Servo Drives 9400:

- Select inverter.
- Select motor.
- Select modules.

3. Assign an application to the field devices and set the drive parameters.
4. Save the »Engineer« project to the project folder.

7.2.6 Inserting devices available on the fieldbus into the »EtherCAT Configurator« project

After a [Fieldbus scan with the »EtherCAT Configurator«](#) (47) the EtherCAT nodes are listed according to their physical order at the fieldbus in the *Start Search* dialog box:



Here you can...

- ▶ assign individual **unique** device names in the *Device name* column. Observe the IEC 61131 syntax (no spaces and leading digits in variable names)!
- ▶ select individual devices in the *Device name* column and copy them into your »EtherCAT Configurator« project:
 - Activate the **Copy into project** button.
 - The **Copy into project** button will only appear if one or more devices are selected.
- ▶ copy all available devices into your »EtherCAT Configurator« project. (55)
 - Activate the **Copy all devices into project** button.



Note!

We recommend to copy all devices into the project.

After pasting you must check if the order of the devices in the project corresponds to the physical order in the network.

7.2.7 Creating the configuration in the »EtherCAT Configurator«

The »EtherCAT Configurator« is used to set up the EtherCAT configuration. During this process, the ECAT_PLC_CFG_1.XML and ECAT_MASTER_1.XML configuration files are created.

These files...

- ▶ illustrate the physical structure of the EtherCAT configuration;
- ▶ contain synchronisation settings, SoftMotion parameter values (only of SoftMotion nodes) and the variable mapping of the EtherCAT nodes.

At a later time, you have to

- ▶ import the ECAT_PLC_CFG_1.XML file into the control configuration using the »PLC Designer«.
- ▶ write the ECAT_MASTER_1.XML file to the IPC.



Note!

- The order of the EtherCAT slaves in the device tree must correspond to the physical order of the EtherCAT configuration.
- In order that the system works properly, end terminals must not be used when setting up the system configuration in the device tree.
- For the integration of external devices, the »EtherCAT Configurator« only supports device descriptions meeting the standards.
- Select the cycle times, according to the technical data, from 1 ... 10 ms. The cycle times are carried out by the configurations in the »EtherCAT Configurator« and »PLC Designer«.

Observe the following conditions before you lay out a topology in the »Engineer«:

- ▶ SoftMotion operation is only possible with Servo Drives 9400 Highline CiA402.
- ▶ The »EtherCAT Configurator« supports Lenze EtherCAT slaves and EtherCAT devices of other manufacturers. The integration of third-party devices requires the respective manufacturer's device descriptions.
 - ▶ [Importing missing field devices](#) (□ 59)

7.2.7.1 Setting up the EtherCAT configuration in the device tree



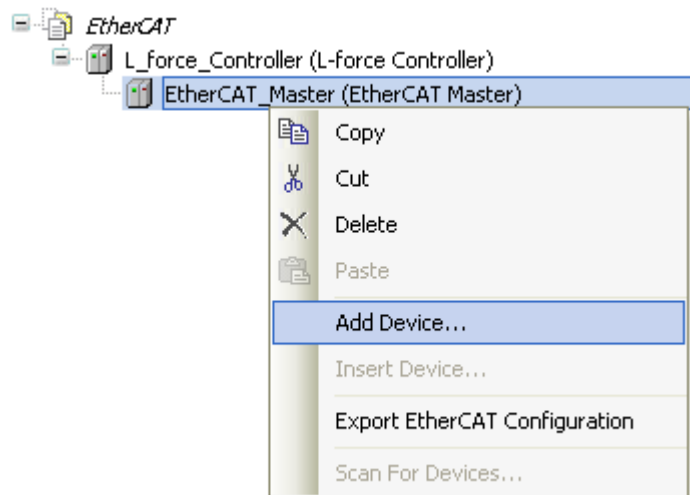
Note!

The order of the devices in the »EtherCAT Configurator« must correspond with the physical structure of the EtherCAT configuration.

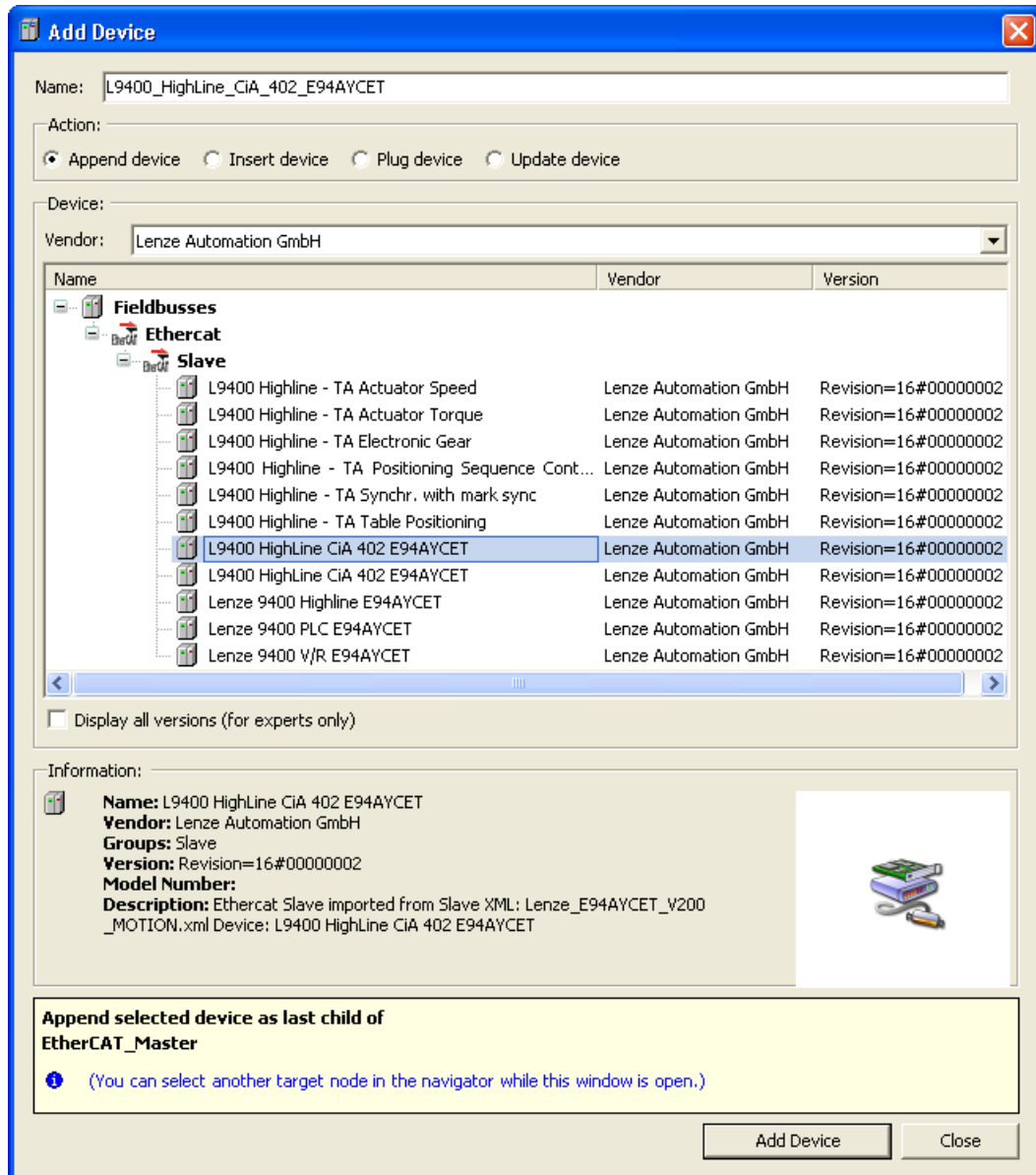


How to set up the EtherCAT configuration in the device tree:

1. Create a new project in the »EtherCAT Configurator«:
 - Menu command: **File→New Project**.
2. Map the physical configuration in the device tree:
 - Add the individual field devices to the *EtherCAT_Master* of the configuration:
Execute the **Add Device** command of the EtherCAT_Master context menu:



3. Select the device in the selection list of the *Add Device* dialog box:



4. Enter the device name in the **Name** input field (example: *X_axis_vertical*),
 - Click the **Add Device** button.
5. Add more slaves of the configuration to the device tree:
 - Select the device to which a slave is to be added. Select the **Insert Device** command from the device context menu.
 - Devices missing in the selection list can be added by importing the corresponding device description file.
 - ▶ [Importing missing field devices](#) (59)
6. Close the dialog box by clicking the **Close** button.

7.2.7.2 Importing missing field devices

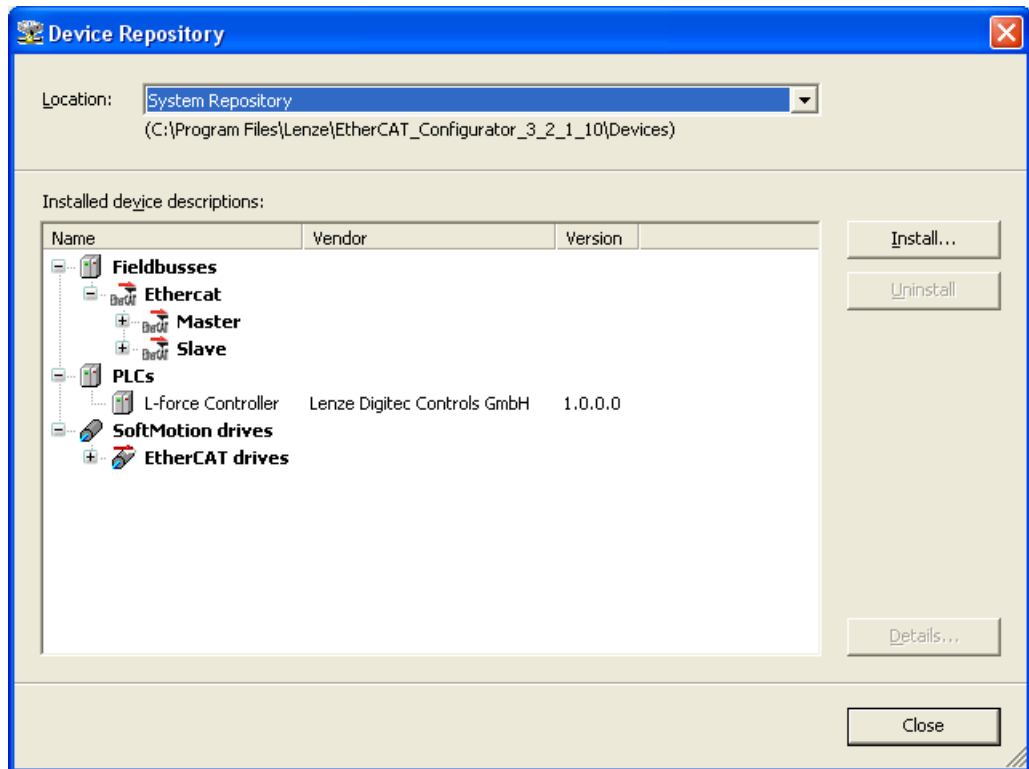
Additional device descriptions can be installed via the device repository.

The device repository manages the device description files stored locally on the system.



How to import additional device descriptions:

1. Open the device repository with the **Tools→Device Repository** menu command:



2. Click the **Install** button.
3. Select the device description file(s) to be imported from the *Installed device descriptions* dialog box appearing now.
4. Finally, click the **Open** button to execute the file import command.
 - A progress bar indicates the installation process.

7.2.7.3 Setting up a DC synchronisation



Note!

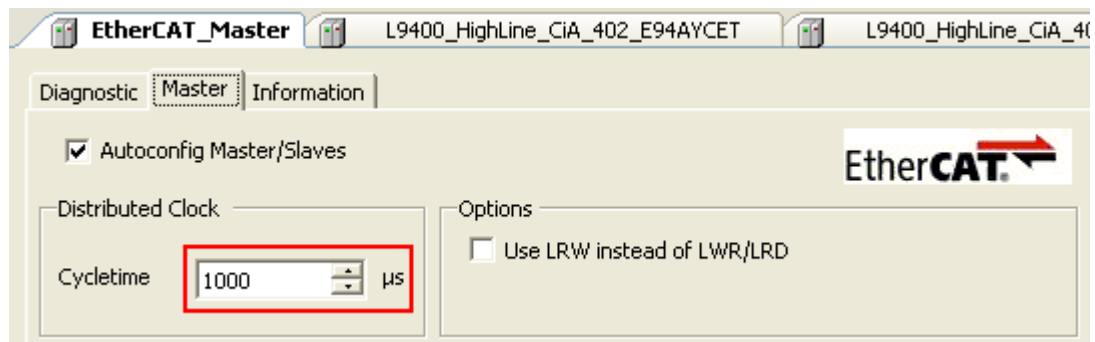
The manual configuration of the slave DC features requires a detailed knowledge of EtherCAT and the field device. Thus, DC settings should only be made by experts.

A faulty configuration can cause maloperation and negative influences on the system.

- ▶ Motion applications always require DC synchronisation.
- ▶ DC synchronisation can also be used for Logic applications.
- ▶ To be able to use the DC function, the first slave connected to the EtherCAT master (IPC) must be DC-master-capable.
The first EtherCAT node connected to the IPC is the **DC master** which provides the other bus nodes (including the IPC) with the exact time.
- ▶ In the arrangement of the slaves following then, DC-capable and non-DC-capable devices can be mixed.
- ▶ For field devices with communication modules (e.g. Servo Drives 9400 or Inverter Drives 8400), a synchronisation source must be selected via code **C01120**. If the DC setting and the selection of the sync source differ from each other (C01120 = MXI1 and "DC nused"), the devices cannot be set to the *Operational* state.
- ▶ For **Servo Drives 9400**, code **C13892/C14892 = 1** must be set (process data mode = "deterministic mode").

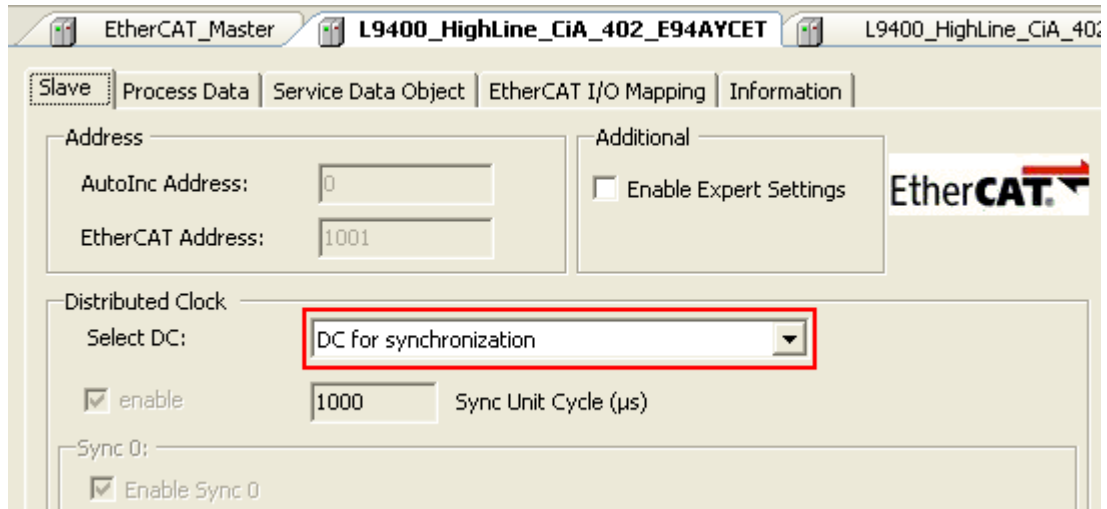
**How to set the DC synchronisation:**

1. Configure the synchronisation of the EtherCAT slaves via distributed clocks (DC).
 - Set the *Distributed clocks (DC) cycle time* at the **EtherCAT Master**:



- The DC cycle time to be set in the »EtherCAT Configurator« must be identical to the EtherCAT task cycle time set in the »PLC Designer«.
- Select the cycle times, according to the technical data, between 1 and 10 ms.
- The (basic) cycle time set here is valid for all Logic and Motion nodes synchronised by distributed clocks.

2. Activate the DC functionality for the **first** slave (DC master) connected to the bus master (select "DC for synchronization"):



3. Also activate the DC functionality (select "DC for synchronization") for all other slave devices which are to use the DC synchronisation.



Note!

Maintain all other basic DC settings for the slaves to ensure a correct DC synchronisation.

7.2.7.4 Set SoftMotion parameters.

**Note!**

The **SoftMotion Drive: Basic** tab is only available for Lenze EtherCAT slaves using the CiA402 application (*SM_Drives*).

The settings depend on the application:

Input fields	Function/description
Axis type and limits	
• Virtual mode	Activate virtual mode for the configuration to be selected.
• Rotary	Select configuration for rotary axis.
• Linear	Select configuration for linear axis.
Modulo settings (only for rotary operation)	
• Modulo value	Define SoftMotion units for rotary operation. (With the value 360.0, the drive would carry out exactly one mechanical revolution.)
Software limits (only for linear operation)	
• Activated	Activate software limit switches.
• Negative	Define value for negative software limit switch.
• Positive	Define value for positive software limit switch.
Limits for CNC (limits are only effective for CNC operation.)	
• Velocity	Define maximum (setpoint) velocity of the axis.
• Acceleration	Define maximum acceleration.
• Deceleration	Define maximum deceleration.
Velocity ramp type	
• Trapezoid	Trapezium
• $\sin^2$	Sine curve
• Parabolic	Parabola
• Jerk	Value for jerk (only for the ramp types " $\sin^2$ " and "parabolic")



A detailed description of the **velocity ramp types** can be found in the documentation/online help for the »PLC Designer« (SoftMotion).

Not all parameters required for operating a Motion drive can be set automatically via the control.

Set the following parameters for the **Servo Drives 9400 HighLine CiA402** manually via the »Engineer« or the »Global Drive Control«:

- ▶ Homing mode (C02640, to be set machine-dependent)
- ▶ Touch-probe interface (to be set machine-dependent)
- ▶ Control of a holding brake (0x60FB/2 | Brake control)
 - Depending on the setting of this parameter, the holding brake is applied for a short time after the conclusion of the home position path. In order to avoid this, set bit 2 in this parameter (disable stop => does not apply the brake in standstill).

7.2.7.5 Executing PDO mapping

**Note!**

The PDO mapping must **only** be set for the **Servo Drives 9400 HighLine CiA402**.

Set the PDO mapping via the process data tab (standard setting for Servo Drives 9400 HighLine CiA402):

Select the outputs			Select the inputs		
Name	Type	Index	Name	Type	Index
☐ 16#1600 IO Outputs (excluded by			☐ 16#1A00 IO Inputs (excluded by 1		
wControlword	UINT	16#6040:00	wStatusword	UINT	16#6041:00
dwTargetPos	UDINT	16#60C1:01	dwPosActual	DINT	16#6063:00
☒ 16#1601 CiA402 Position			☒ 16#1A01 CiA 402 Position		
wControlword	UINT	16#6040:00	Statusword	UINT	16#6041:00
Modes of operation	SINT	16#6060:00	Modes of operation display	SINT	16#6061:00
Physical outputs	USINT	16#60FE:01	Digital inputs	USINT	16#60FD:00
Interpolation data X1	DINT	16#60C1:01	Position actual value	DINT	16#6063:00
☐ 16#1602 CiA 402 Position/Velocit			☐ 16#1A02 CiA 402 Position/Velocit		
wControlword	UINT	16#6040:00	Statusword	UINT	16#6041:00
Modes of operation	SINT	16#6060:00	Modes of operation display	SINT	16#6061:00
Physical outputs	USINT	16#60FE:01	Digital inputs	USINT	16#60FD:00
Interpolation data X1	DINT	16#60C1:01	Position actual value	DINT	16#6063:00
Target Velocity	DINT	16#60FF:00	Velocity actual value	DINT	16#606C:00
☐ 16#1603 CiA 402 Torque (exclude			☐ 16#1A03 CiA 402 Torque (exclude		
wControlword	UINT	16#6040:00	Statusword	UINT	16#6041:00
Modes of operation	SINT	16#6060:00	Modes of operation display	SINT	16#6061:00
Physical outputs	USINT	16#60FE:01	Digital inputs	USINT	16#60FD:00
Target Torque	DINT	16#6071:00	Torque Actual Value	DINT	16#6077:00
☐ 16#1604 CiA 402 Position_TP (exc			☐ 16#1A04 CiA 402 Position_TP (exc		
wControlword	UINT	16#6040:00	Statusword	UINT	16#6041:00
Modes of operation	SINT	16#6060:00	Modes of operation display	SINT	16#6061:00
Physical outputs	USINT	16#60FE:01	Digital inputs	USINT	16#60FD:00
Interpolation data X1	DINT	16#60C1:01	Position actual value	DINT	16#6063:00
TPCtrlWord	UINT	16#60B8:00	TouchProbe Statusword	UINT	16#60B9:00
TorqueLimitPositiv	UINT	16#60E0:00	TPPosLatch1Positiv	DINT	16#60BA:00
TorqueLimitNegativ	UINT	16#60E1:00	TPPosLatch1Negativ	DINT	16#60BB:00
			TPPosLatch2Positiv	DINT	16#60BC:00
			TPPosLatch2Negativ	DINT	16#60BD:00
			Torque actual value	DINT	16#6077:00

- The settings for the outputs and inputs must be identical.
- In order to change the setting, you must first deselect the current setting. After that you can freely select the wanted setting.

7.2.7.6 Editing the EtherCAT I/O image

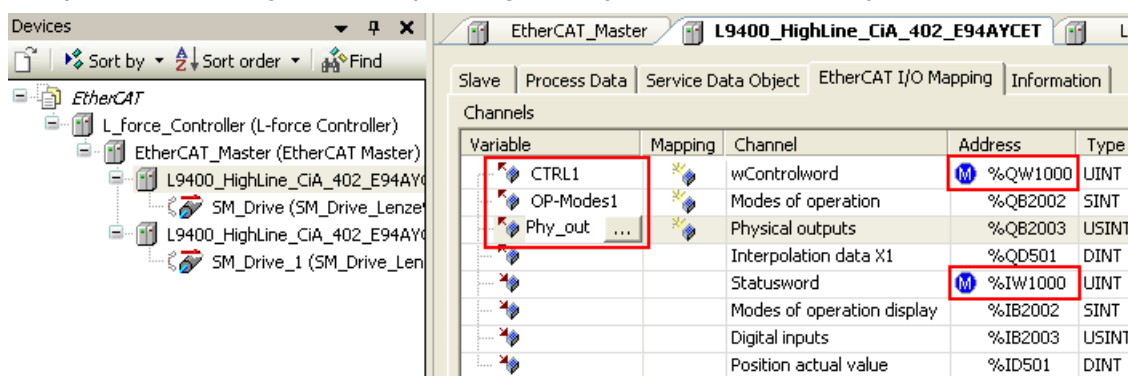
- Assign **non-ambiguous** variable names to the input and output objects according to IEC 61131 syntax (no blanks and leading digits in the variable name). After the import of the EtherCAT configuration into the »PLC Designer« control configuration, corresponding system variables are available for the PLC program.



Note!

- Always use the system variables within the PLC program in order to access the input and output objects or assign values to them.
- When copying a device in the »EtherCAT Configurator«, you must rename the variables for the device copy, otherwise there will be a compiler error during the compilation in the »PLC Designer«.

- By double-clicking the corresponding fields, you can make the adjustments:



In the example, variable names have been assigned to the first three output objects. Moreover, an offset of '1000' has been entered for the first output object (%QW...) and the first input object (%IW...).

- The »EtherCAT Configurator« does not work with byte addresses. The objects are addressed with the byte address in the control configuration:

Objects	Addresses in the »EtherCAT Configurator«	Byte addresses
Output objects	%QX1000.0	%QB1000
	%QB1000	%QB1000
	%QW1000	%QB500
	%QD1000	%QB250
Input objects	%IX1000.0	%IB1000
	%IB1000	%IB1000
	%IW1000	%IB500
	%ID1000	%IB250

7.2.7.7 Export of EtherCAT configuration

First save your project with the menu command **File→Save project to** in the project folder before you export the EtherCAT configuration.

During the export of the EtherCAT configuration, the configuration files *ECAT_PLC_CFG_1.XML* and *ECAT_MASTER_1.XML* created.

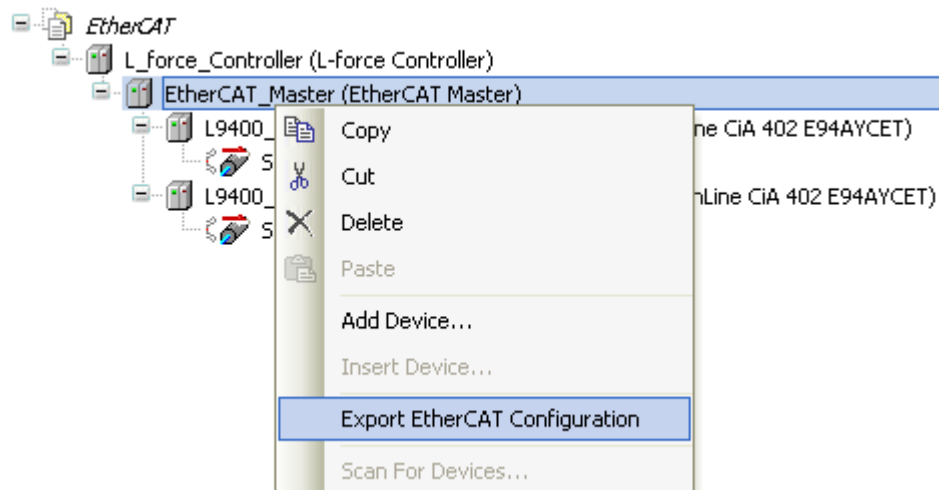
At a later time, you have to

- ▶ import the *ECAT_PLC_CFG_1.XML* file into the control configuration using the »PLC Designer«.
- ▶ write the *ECAT_MASTER_1.XML* file for the EtherCAT master stack to the IPC.



How to export the EtherCAT configuration files:

1. Select **Export EtherCAT Configuration** from the context menu of the master:



2. A directory tree appears. Select the prepared project folder for storing.
3. Confirm the selection by clicking the **OK** button.

The configuration files *ECAT_PLC_CFG_1.XML* and *ECAT_MASTER_1.XML* is now saved to the project folder.

7.2.8 Configuration in the »PLC Designer«

The »PLC Designer« serves to illustrate the field device topology in the control configuration.



Tip!

Nodes at other fieldbus systems can be configured in the »PLC Designer«.

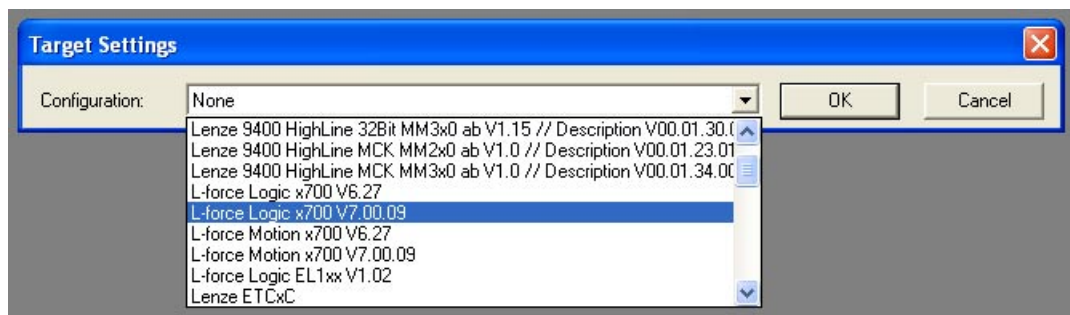
► [EtherCAT with CANopen or PROFIBUS](#) (91)

7.2.8.1 Set-up of control configuration



How to set up the control configuration in the »PLC Designer«:

1. Create a new »PLC Designer« project:
 - Menu command: **File→New**
2. Select the suitable target system from the *Target Settings* dialog box:



- **L-force_Logic_x700_Vx.xx.xx** for Logic target systems
- **L-force_Motion_x700_Vx.xx.xx** for pure Motion target systems or Motion- and Logic target systems

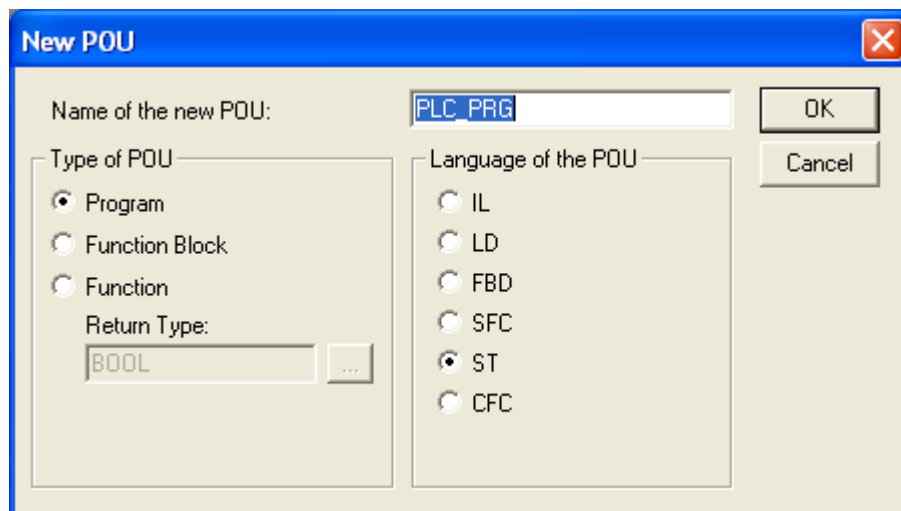
► [EtherCAT with CANopen or PROFIBUS](#) (91)

3. Confirm the configuration of the target system setting by clicking the **OK** button.

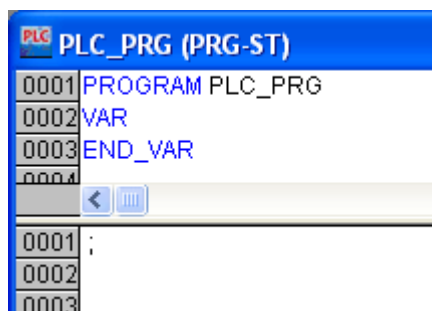
4. Create a block:

**Note!**

The program organisation unit must contain at least one instruction to function properly.



The 'New POU' dialog box is shown. It has a title bar 'New POU' with a close button. The 'Name of the new POU:' field contains 'PLC_PRG'. The 'Type of POU' section has three radio buttons: 'Program' (selected), 'Function Block', and 'Function'. Below it, the 'Return Type:' field contains 'BOOL'. The 'Language of the POU' section has six radio buttons: 'IL', 'LD', 'FBD', 'SFC', 'ST' (selected), and 'CFC'. There are 'OK' and 'Cancel' buttons on the right.

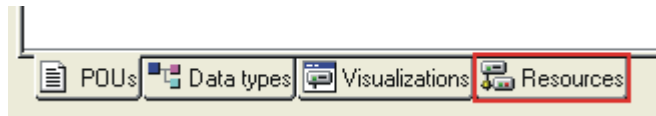


The 'PLC_PRG (PRG-ST)' ladder logic editor is shown. It has a title bar 'PLC_PRG (PRG-ST)'. The editor contains a table with the following content:

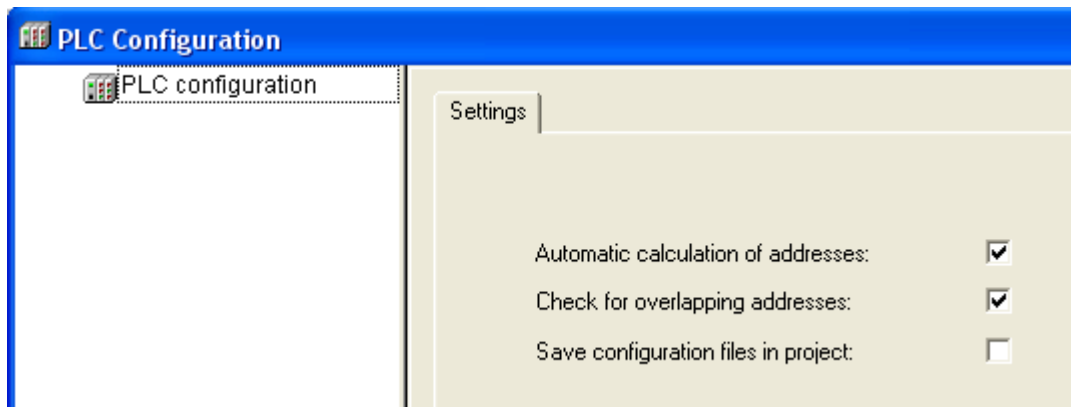
Address	Instruction
0001	PROGRAM PLC_PRG
0002	VAR
0003	END_VAR
0004	
0001	;
0002	
0003	

5. Create the control configuration:

- Open the *Resources* dialog box:



- Open the *PLC Configuration* dialog box:



Setting	Description
Automatic calculation of addresses	Every newly added module automatically gets an address which results from the address of the module integrated before and the size of this module. If a module is removed from the configuration, the addresses of the subsequent modules are adapted automatically. The Extras→Calculate addresses menu command serves to recalculate the addresses starting with the currently selected node (module).
Check for overlapping addresses	During the compilation of the project, a check for address overlapping is carried out and overlapping addresses are indicated.
Save configuration files in project	The data of the configuration file(s) *.cfg and device files on which the current control configuration is based are stored in the project.



Note!

We recommend to keep the standard setting. In case of a manual address allocation, you must ensure that each object address is **non-ambiguous** in the entire control configuration.

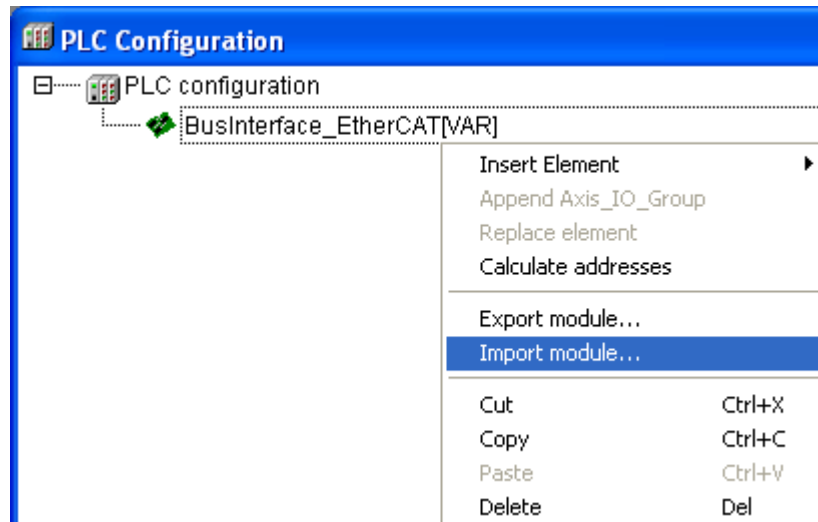
Detailed information on this can be found in the documentation for the »PLC Designer«.

6. Add the bus interface to the PLC configuration:

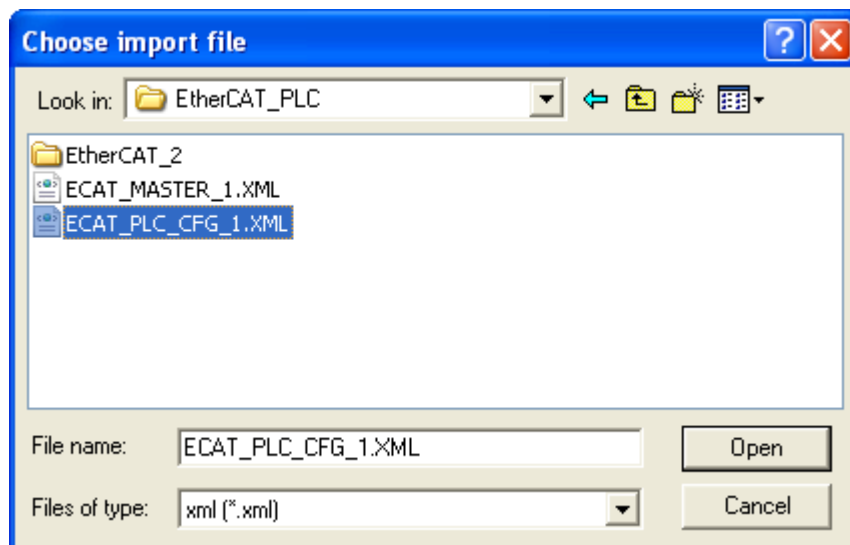
Select the **Append subelement→BusInterface_EtherCAT** command from the *PLC_configuration* context menu.

7. Import the EtherCAT components (file ECAT_PLC_CFG_1.XML) for the control configuration:

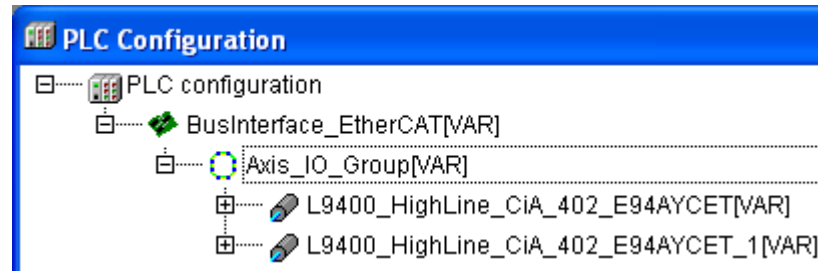
- Select the **Import module** command in the context menu of *BusInterface_EtherCAT*:



- Select the file ECAT_PLC_CFG_1.XML from the project folder and **Open**:



- When the file has been imported successfully, the *Axis_IO_Group/IO_Group* is added to the configuration:



- The configuration created in the »EtherCAT Configurator« is attached below the *Axis_IO_Group/IO_Group*.



Note!

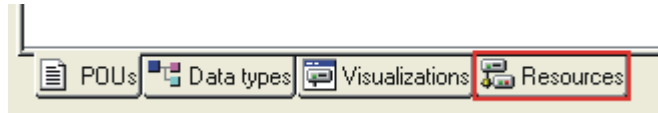
- The EtherCAT components of the control configuration (device and variable names) may only be changed in the »EtherCAT Configurator«! Changes made in the »PLC Designer« will get overwritten at the import of the ECAT_PLC_CFG_1.XML file.
- The *Axis_IO_Group/IO_Group* must not be renamed!
- The *Axis_IO_Group/IO_Group* is overwritten at every import of the ECAT_PLC_CFG_1.XML file.
- The *Axis_IO_Group/IO_Group* is created automatically at every import of the ECAT_PLC_CFG_1.XML file and assigned a default name.

7.2.8.2 Creating a task

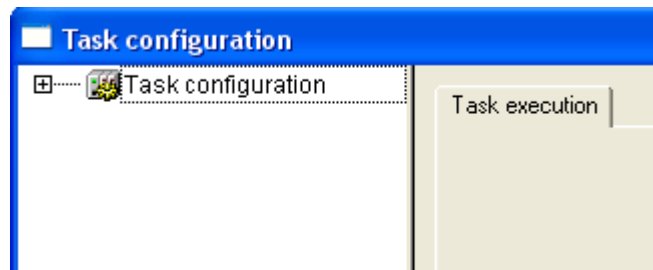


How to create a task in the »PLC Designer«:

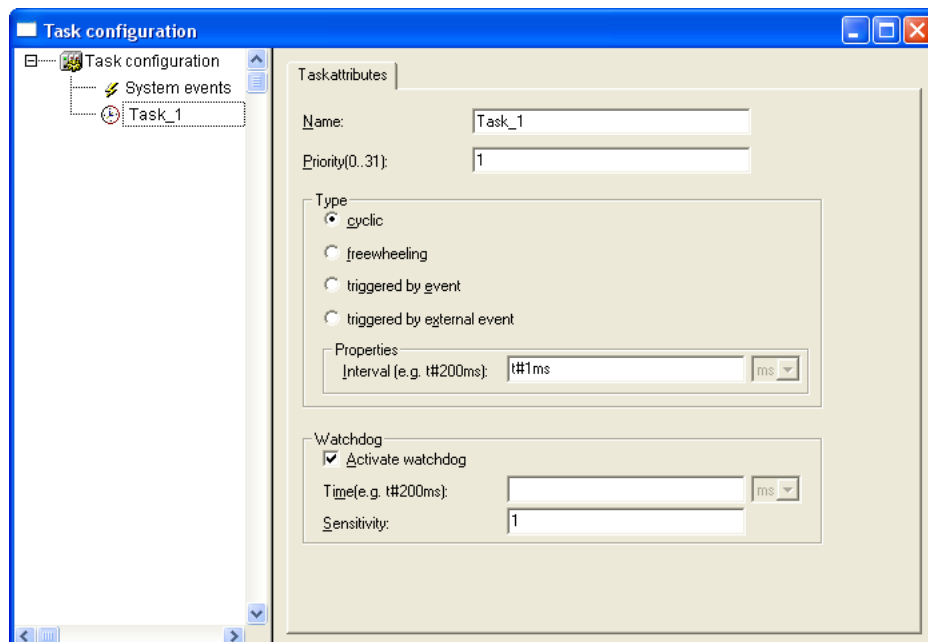
1. Select the *Resources* tab:



2. Select *Task Configuration*:



3. Select the **Append Task** command from the Task Configuration context menu.
4. Create a new task and make the settings relevant for the respective configuration:



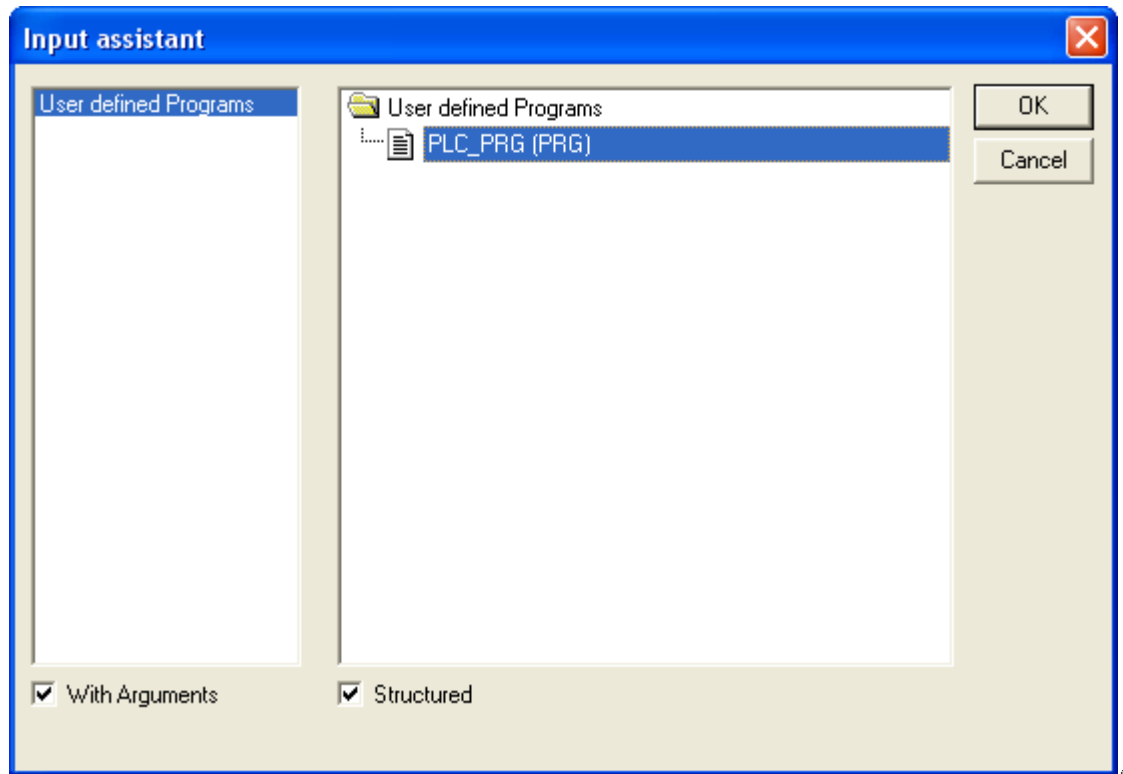
Define the task cycle time. (Example in the screenshot: 1 ms)



Note!

The task cycle time must be identical with the DC cycle time set in the »EtherCAT Configurator«.

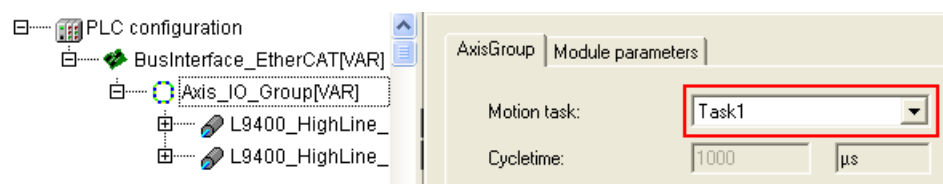
5. Assign a program call to the task:
 - Select **Attach program call** in the context menu of the task.
 - Open the **input assistance** via the  button:



- Select the required program from the list of user-defined programs.
6. Compile the project with <F11> or save the project.

► [Compiling project data](#) (77)

7. Assign a task to the *Axis_IO_Group*/IO_Group:



Use of breakpoints



Note!

The setting of breakpoints in the PLC task assigned to the EtherCAT bus is permissible.

Please observe:

- When reaching a breakpoint this task will be stopped.
- No new setpoints are calculated (Motion drives come to a standstill), no deceleration ramp is generated.
- The control still sends EtherCAT frames. After restarting the PLC task, the interrupted processing is resumed.
- Motion drives continue the interrupted motion, no acceleration ramp is generated.



Detailed information on breakpoints can be found in the documentation/online help for the »PLC Designer«.

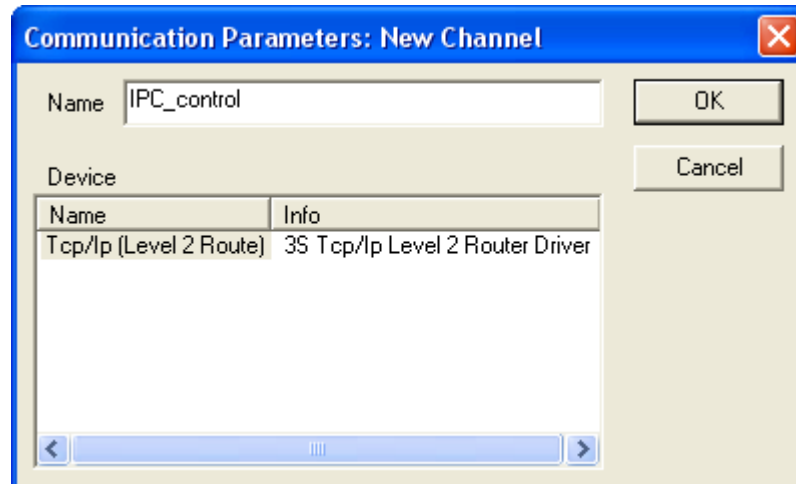
7.2.8.3 Configuring the communication parameters

Set the communication parameters to establish a connection to the respective IPC.

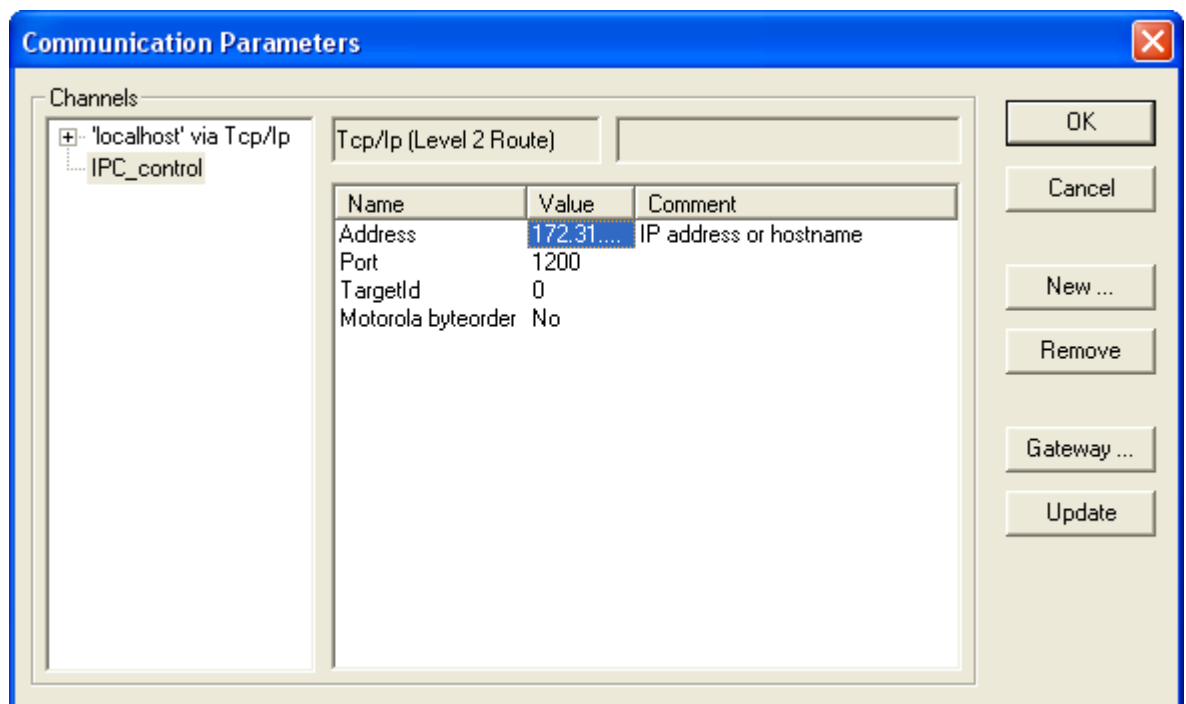


How to configure the communication parameters

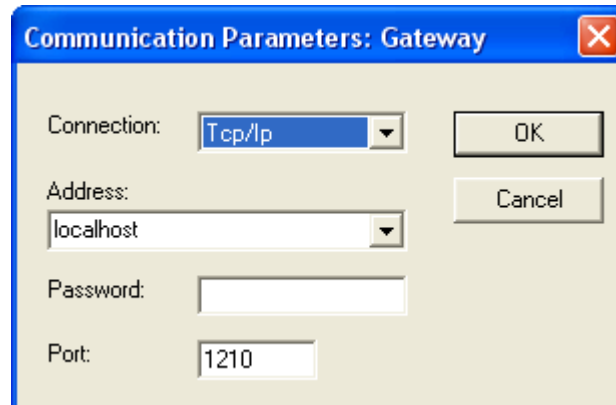
1. Select the **Online→Communication Parameters** command:



- Enter the designation of the new channel in the **Name** input field.
 - Confirm the entry by clicking the **OK** button.
2. Enter the parameters in the Values column in the *Communication parameters* dialog box.
- Double-click the respective value to edit the default values:



3. Click the **Gateway** button to configure a gateway connection:



- Select the **TCP/IP** connection type from the **Connection** selection field.
- Confirm the selection by clicking the **OK** button.

7.2.8.4 Compiling project data

To compile the project data, select the **Project→Build** menu command or press the **<F11>** function key.

- ▶ If errors occurred during the compilation process, you can locate and eliminate them by means of the »PLC Designer« error messages. Then compile the project data again.
- ▶ If no errors occurred during the compilation process, save the »PLC Designer« project to the project folder.

7.2.8.5 Logging on to the control system with the »PLC Designer«

To log the »PLC Designer« on to the control system, select the **Online→Login** menu command.

- ▶ For this, the PLC program must be error-free.
- ▶ Confirm the appearing query dialog whether the new program is to be loaded with **Yes**.

7.2.8.6 Loading the control configuration onto the IPC



How to load the control configuration onto the IPC:

1. Select the **Online→Write file to PLC** menu command.
2. Select the required file from the *Write file to PLC* dialog box.
3. Confirm the selection by clicking the **Open** button.

The file is loaded onto the IPC and saved there under the same name.



Tip!

The **Online→Read file from PLC** menu command can be used to reload a file from the IPC onto the »PLC Designer« project.

7.2.8.7 Loading the master configuration onto the EtherCAT master

The **Reset (original)** command serves to load the master configuration (ECAT_MASTER_1.XML) onto the EtherCAT master:

1. Reset (original) re-initialises the EtherCAT master stack.
2. The master compares the EtherCAT configuration with the physical bus topology. If the EtherCAT configuration is identical with the bus topology, the bus enters the *Pre-Operational* state.
3. Now it is possible to communicate with the slaves via the EtherCAT fieldbus.

Checking with »WebConfig« or »Engineer«

- Under **Master Configuration**, you can check whether the configuration file ECAT_MASTER_1.XML matches the physical bus structure.
- Depending on the slot used for the EtherCAT (MC-ETC) communication card, the parameters of the following codes should be checked:
 - MC-ETC in slot 1: codes 1080.2 / 1080.3 / 1080.4
 - MC-ETC in slot 2: codes 1580.2 / 1580.3 / 1580.4
- If the ECAT_MASTER_1.XML configuration file corresponds to the physical bus structure, the checksums of "ECAT Master Config" and "ECAT Stack Master" are identical and the "ECAT Bus Scan Match" parameter has the value 1.

Master Configuration		
1080.1	ECAT Master Configuration Date	10.03.2008 12:12:46
1080.2	ECAT Master Config Checksum	0x6C03DB77
1080.3	ECAT Stack Master Checksum	0x6C03DB77
1080.4	ECAT Bus Scan Match	1

Submit

Submit & Persist All

Refresh

7.2.8.8 Loading and starting the PLC program

- Select the **Online→Download** menu command to load the PLC program included in the configuration onto the control system.
- Select the **Online→Run** menu command to start the PLC program.

7.3 Checking the system startup



Note!

Checking of the system startup is only possible for SoftMotion target systems.

To check whether the system has started up correctly, you can do the following

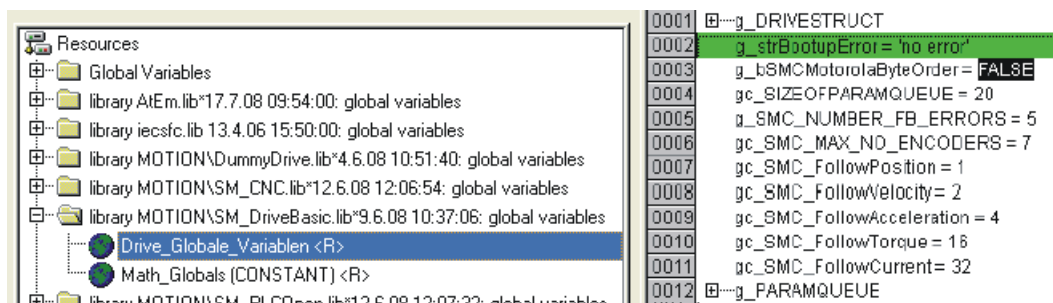
- ▶ [Evaluation of the boot-up error message of the SM_DriveBasic.lib library](#) (80);
- ▶ [Evaluation of the Axis IO Group state](#) (81);
- ▶ [Evaluation of the axis state](#) (82).

7.3.1 Evaluation of the boot-up error message of the SM_DriveBasic.lib library

To check that the system has started up correctly, you can evaluate the boot-up error. For this purpose, the scope of the global variables contained in the *SM_DriveBasic.lib* library includes the element `g_strBootupError`. This element displays error messages in text form:

`g_strBootupError = no error`

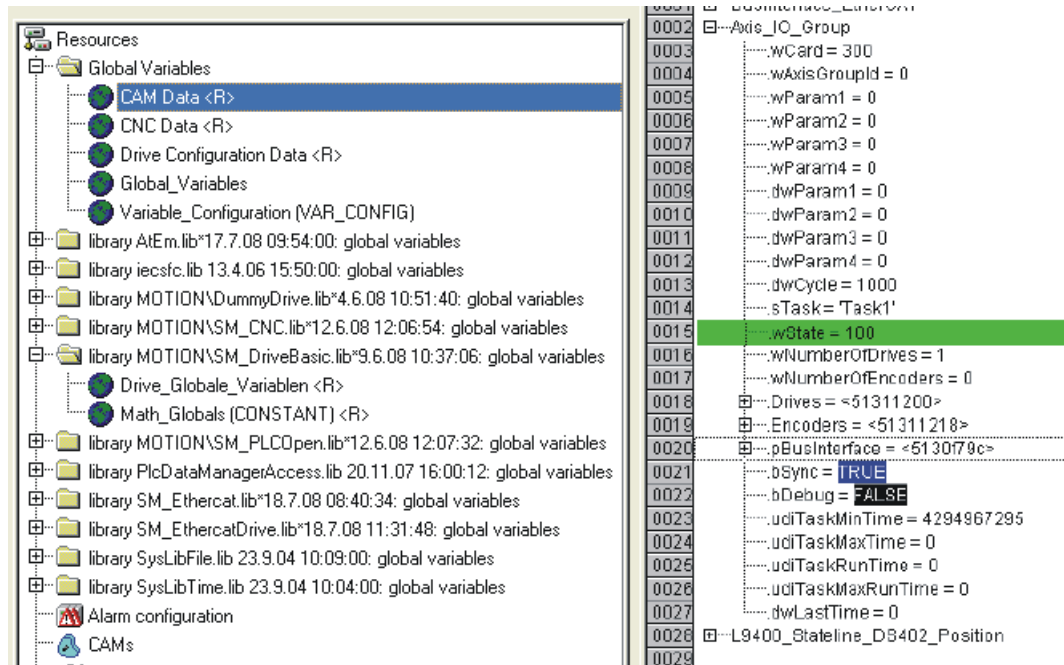
`no error` indicates that the system has started up correctly:



7.3.2 Evaluation of the Axis_IO_Group state

The Axis_IO_Group structure contains the global variable wState.

The following value ranges have to be evaluated for the system startup check:



The value of the wState variable has the following meaning:

State of the Axis_IO_Group	State of the system
wState = 0	- Initial state - Project loaded - PLC in stop
wState = 1...99	- System is starting up - Project loaded - PLC started
wState = 100	System has started up successfully
wState > 1000	Error occurred during startup, compare error message in g_strBootupError

7.4 Typical commissioning scenarios

In this chapter you will find typical commissioning scenarios and the corresponding state diagrams, which will help you to carry out simple analyses of the system behaviour.

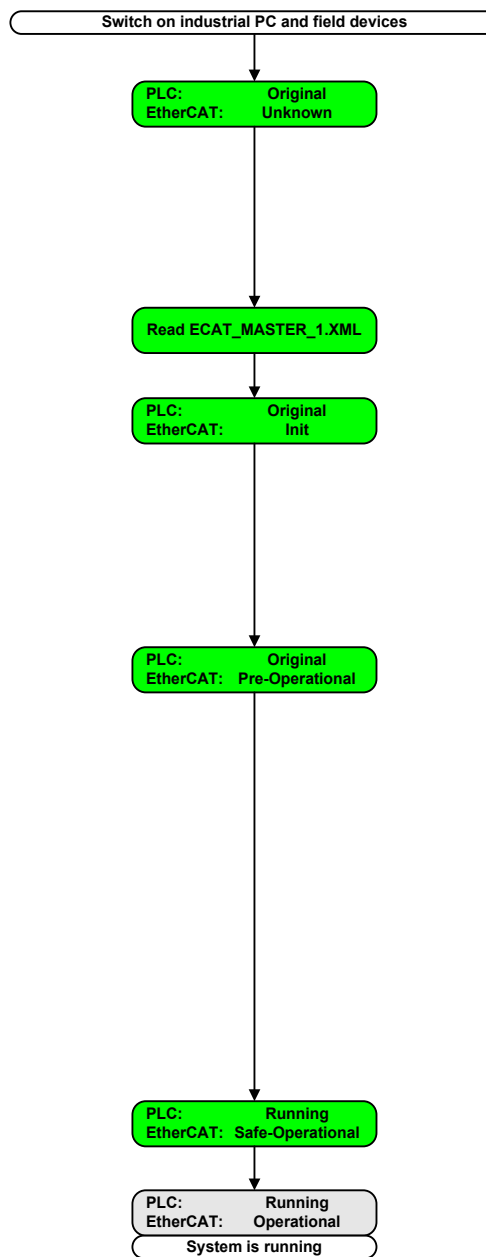
7.4.1 Switching on a completely configured system

Initial situation:

- ▶ The PLC application has been loaded on the IPC.
- ▶ The ECAT_Master_1.XML file matching the bus topology is available on the IPC.

System behaviour:

- ▶ The system starts up automatically.



Legend			
PLC: EtherCAT:	Transient state of PLC and EtherCAT Automatic change to next state	Load PLC	Load PLC application
PLC: EtherCAT:	Steady state of PLC and EtherCAT Change to next by means of an external action	Start PLC	Start PLC application
Reset (origin)	Reset the PLC. The PLC application is deleted.		

7.4.2 Switching on a system with an incomplete configuration

Initial situation:

- ▶ The PLC application has been loaded on the IPC.
- ▶ The ECAT_Master_1.XML file is not available on the IPC or does not match the physical bus topology.

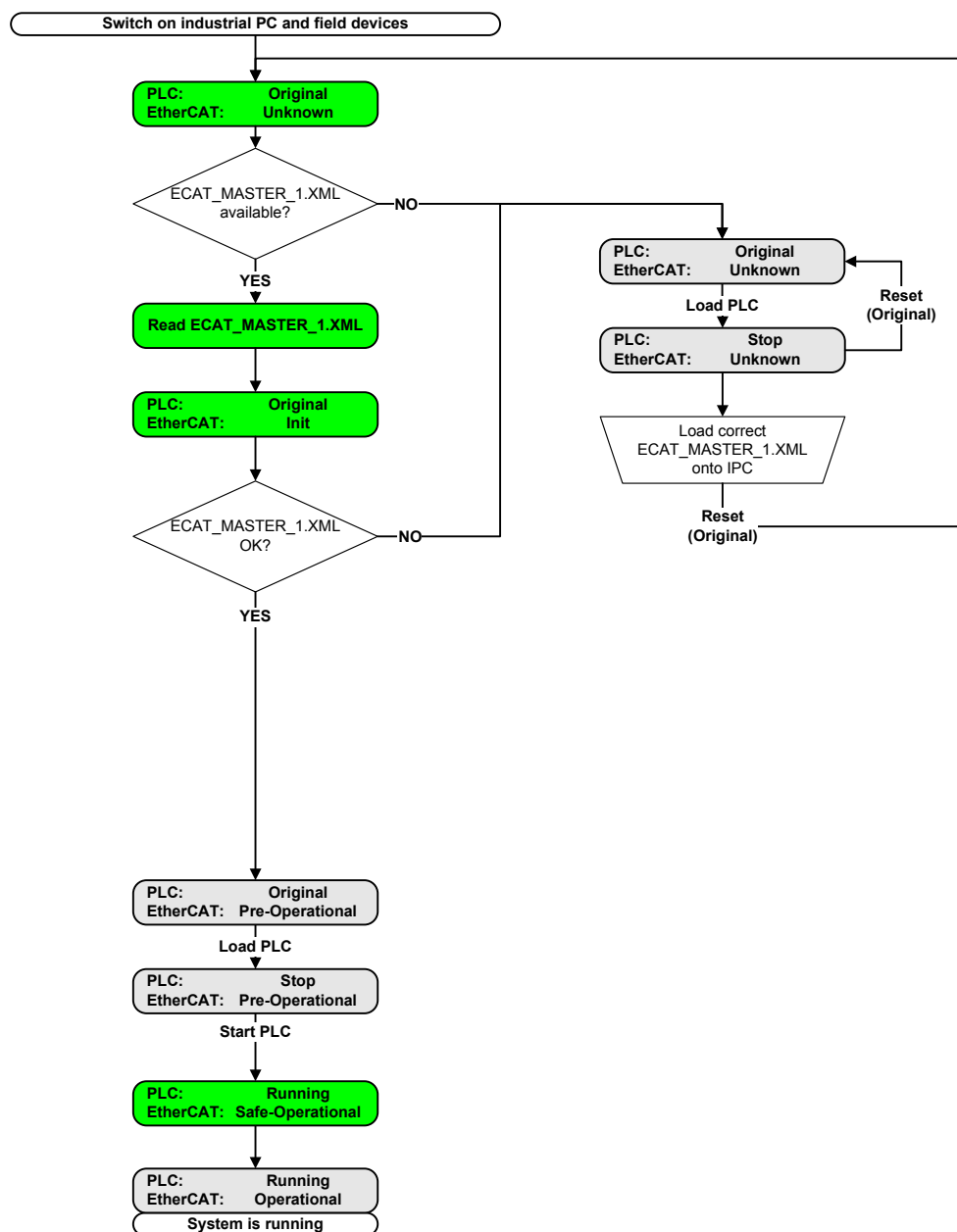
System behaviour:

- ▶ EtherCAT remains in "Unknown" state.
- ▶ The PLC does not start up.



How to complete the configuration:

1. Load the correct "ECAT_Master_1.xml" file onto the IPC.
2. Execute "Reset (Original)".
 - The PLC is reset, the PLC application is deleted.
3. The system starts up.
 - You have to manually load and start the PLC application.



Legend

PLC: EtherCAT:	Transient state of PLC and EtherCAT Automatic change to next state	Load PLC	Load PLC application
PLC: EtherCAT:	Steady state of PLC and EtherCAT Change to next by means of an external action	Start PLC	Start PLC application
Reset (origin)	Reset the PLC. The PLC application is deleted.		

7.4.3 Updating the PLC application while the EtherCAT configuration remains unchanged

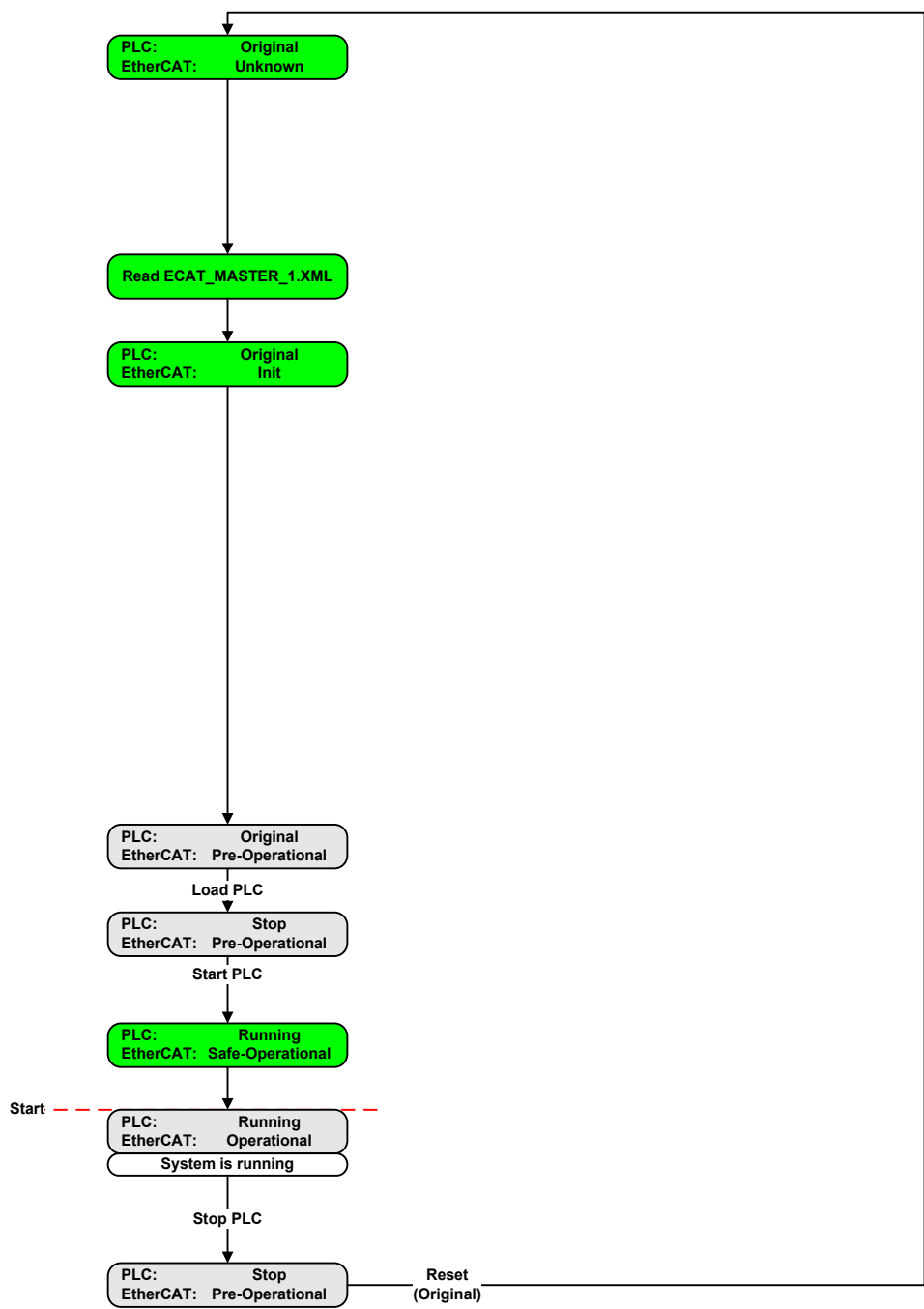
Initial situation:

- ▶ PLC and EtherCAT are running.
 - PLC state: *Running*
 - EtherCAT state: *Operational*



How to update the PLC application:

1. Stop the PLC.
2. Execute "Reset (Original)".
 - The PLC is reset, the PLC application is deleted.
3. The system starts up.
 - You have to manually load and start the PLC application.



Legend			
PLC: EtherCAT:	Transient state of PLC and EtherCAT Automatic change to next state	Load PLC	Load PLC application
PLC: EtherCAT:	Steady state of PLC and EtherCAT Change to next by means of an external action	Start PLC	Start PLC application
Reset (origin)	Reset the PLC. The PLC application is deleted.	Start	Beginning of the scenario

7.4.4 Stopping and starting the PLC while the configuration remains unchanged

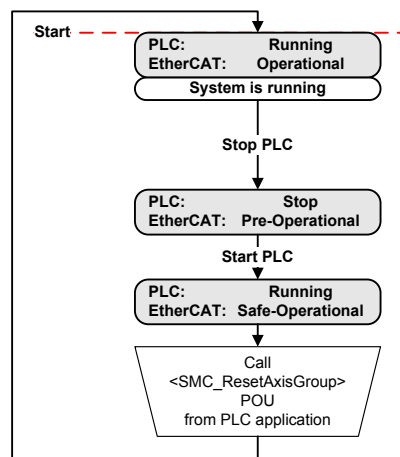
Initial situation:

- PLC and EtherCAT are running.
 - PLC state: *Running*
 - EtherCAT state: *Operational*



How to stop and start the PLC:

1. Stop the PLC.
2. Start the PLC.
3. The PLC application must call the "SMC_ResetAxisGroup" program organisation unit.
4. The system is running again.

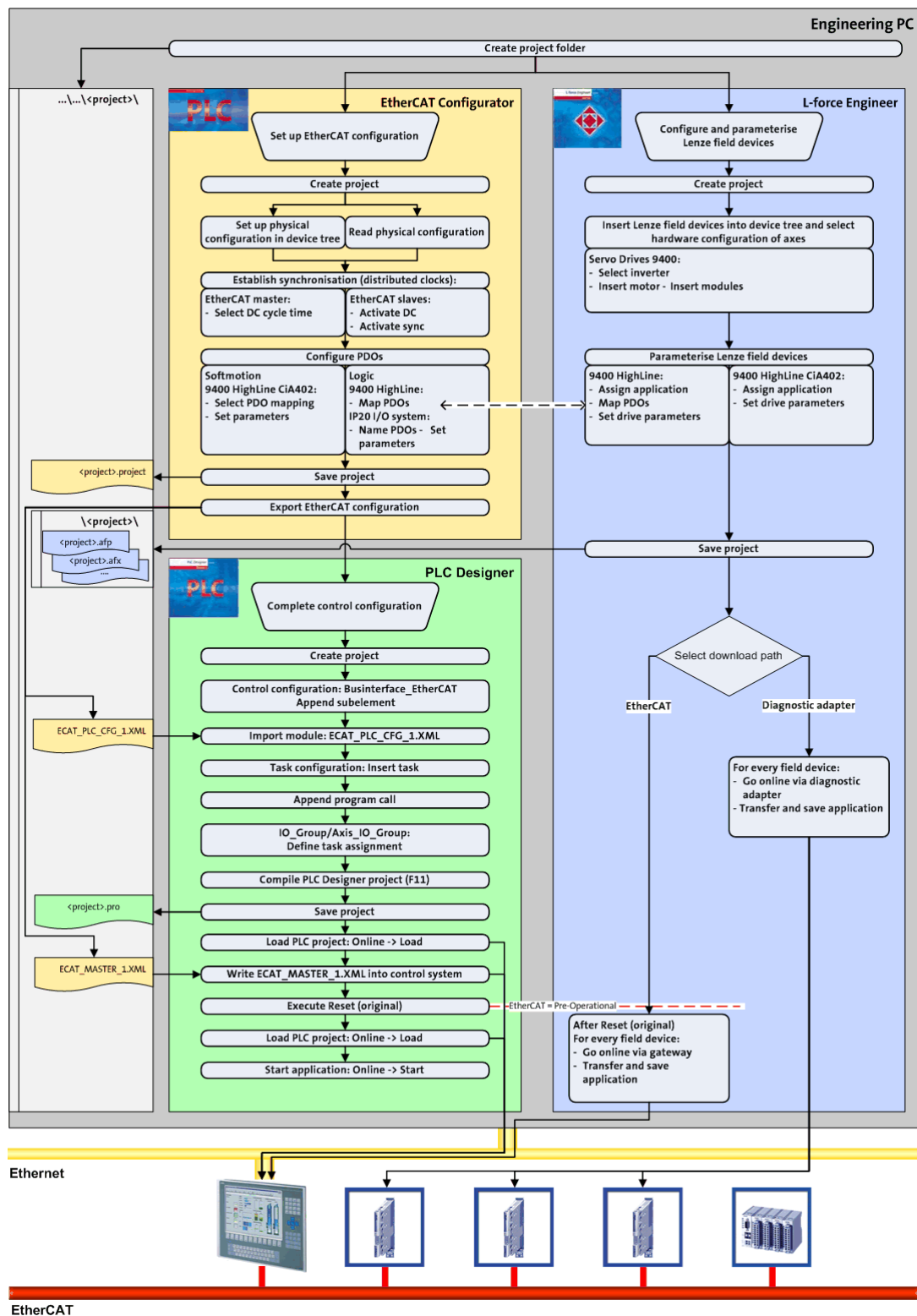


Legend

PLC: EtherCAT:	Transient state of PLC and EtherCAT Automatic change to next state	Load PLC	Load PLC application
PLC: EtherCAT:	Steady state of PLC and EtherCAT Change to next by means of an external action	Start PLC	Start PLC application
Reset (origin)	Reset the PLC. The PLC application is deleted.	Start	Beginning of the scenario

7.5 Detailed overview of the commissioning steps

In the following diagram the individual commissioning steps and their processing order are summarised once again. Detailed information on the individual processing steps can be found in the chapter [Detailed commissioning steps](#) (45).



8 EtherCAT with CANopen or PROFIBUS



Note!

A mixed operation is only possible with industrial PCs which have two additional slots for communication cards. A mixed operation is not possible with the "Command Station".

The EtherCAT bus system can be combined with CANopen or PROFIBUS. This makes sense if not all field devices are available for the same bus system or a motion bus is required parallelly to the PROFIBUS. The bus systems are synchronised in the control.

The following combinations are allowed:

- ▶ CANopen and EtherCAT
 - CANopen configuration in the »PLC Designer«
 - EtherCAT configuration in the »EtherCAT Configurator«
- ▶ PROFIBUS (as logic bus) and EtherCAT (as motion bus)
 - PROFIBUS configuration in the »PLC Designer«
 - EtherCAT configuration in the »EtherCAT Configurator«



Communication manual "CANopen control technology"

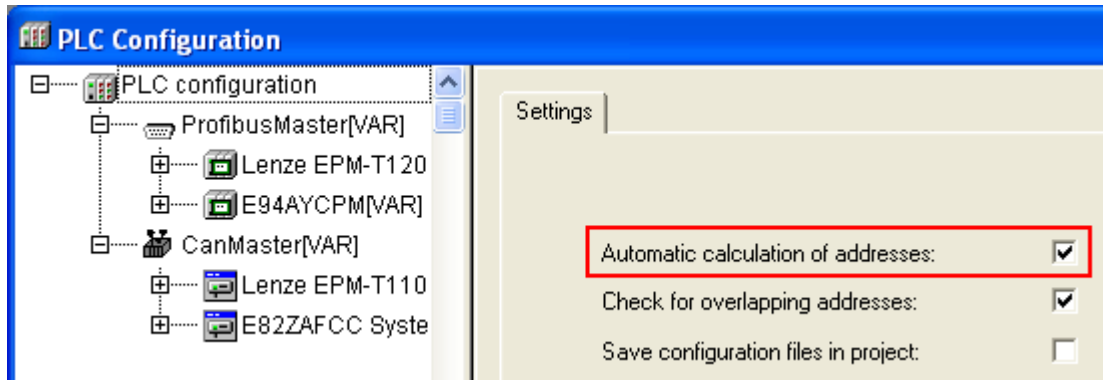
Here you can find detailed information on how to commission CANopen components.

Communication manual "PROFIBUS control technology"

Here you can find detailed information on how to commission PROFIBUS components.

8.1 Addressing the PROFIBUS and CANopen stations

The addresses for input and output objects of the PROFIBUS and CANopen stations are automatically allocated in the »PLC Designer« (standard setting):



Note!

We recommend to keep the standard setting. In case of a manual address allocation, you must ensure that each object address is **non-ambiguous** in the entire control configuration.

Detailed information on this can be found in the documentation/online help of the »PLC Designer«.

8.2 Addressing EtherCAT nodes using CANopen/PROFIBUS nodes

The addresses for input and output objects of the EtherCAT stations are allocated in the »EtherCAT Configurator«. Afterwards, the EtherCAT configuration is imported into the »PLC Designer« control configuration.

There are two cases for the control configuration in the »PLC Designer«:

1. PROFIBUS/CANopen before EtherCAT:

- Keep the standard settings for address allocation for the PROFIBUS/CANopen stations in the »PLC Designer« control configuration. (The addressing of the PROFIBUS/CAN stations starts with '0'.)
- Define a sufficient address offset for the first input and output object in the »EtherCAT Configurator« at the first EtherCAT slave (> PROFIBUS/CAN addresses). The addresses of the other input and output objects are updated automatically after the entry.
Thus, address conflicts can be prevented after the import of the EtherCAT configuration into the »PLC Designer« control configuration.

2. EtherCAT before PROFIBUS/CANopen:

- Do not define any address offsets in the »EtherCAT Configurator«. (The addressing of the EtherCAT stations starts with '0'.)
- For PROFIBUS/CANopen stations, no address offsets must be defined in the »PLC Designer« control configuration. If address offsets are defined, the offsets must be greater than the EtherCAT addresses.

► [Settings in the »EtherCAT Configurator«](#) (📖 94)

Settings in the »EtherCAT Configurator«

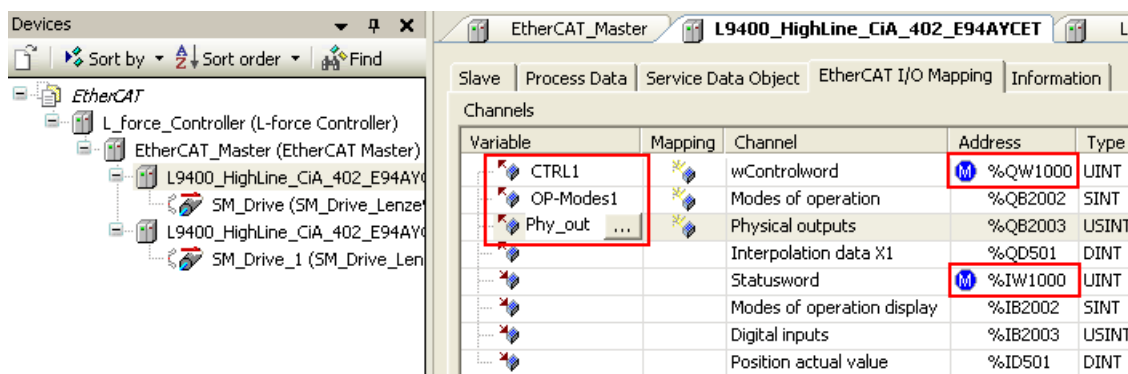
- Assign **non-ambiguous** variable names to the input and output objects according to IEC 61131 syntax (no blanks and leading digits in the variable name). After the import of the EtherCAT configuration into the »PLC Designer« control configuration, corresponding system variables are available for the PLC program.



Note!

Always use the system variables within the PLC program in order to access the input and output objects or assign values to them.

- By double-clicking the corresponding fields, you can make the adjustments:

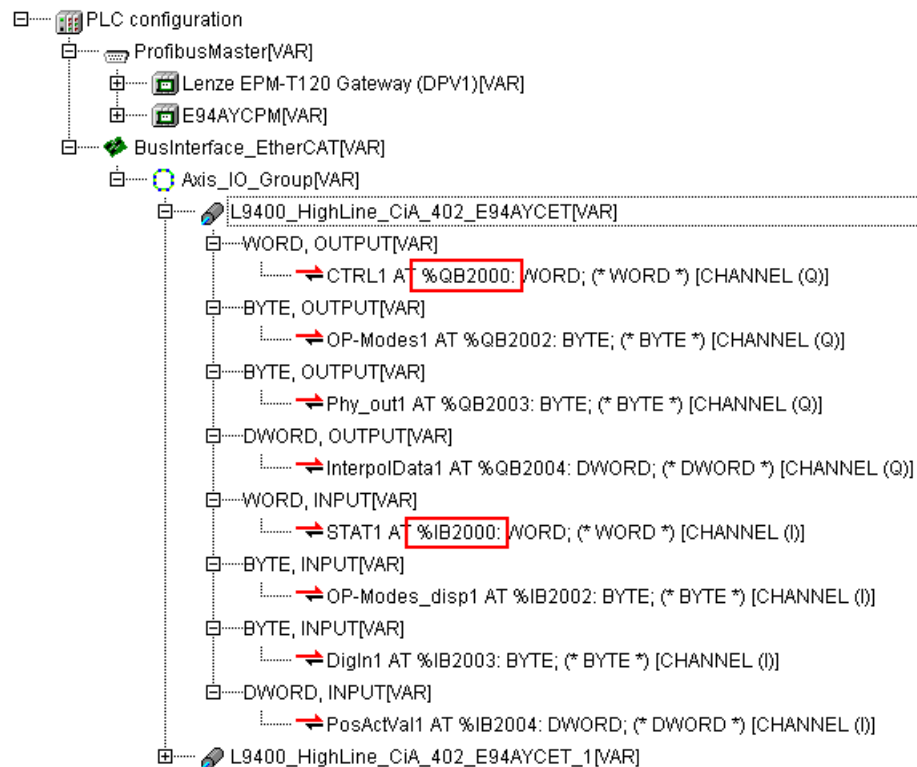


In the example, variable names have been assigned to the first three output objects. Moreover, an offset of '1000' has been entered for the first output object (%QW...) and the first input object (%IW...).

- The »EtherCAT Configurator« does not work with byte addresses. The objects are addressed with the byte address in the control configuration:

Objects	Addresses in the »EtherCAT Configurator«	Byte addresses
Output objects	%QX1000.0	%QB1000
	%QB1000	%QB1000
	%QW1000	%QB500
	%QD1000	%QB250
Input objects	%IX1000.0	%IB1000
	%IB1000	%IB1000
	%IW1000	%IB500
	%ID1000	%IB250

In the »PLC Designer«, the output and input objects are "byte-addressed". The original address offset of '1000' (word level) for the first output and input objects is set accordingly to '2000' after the import of the EtherCAT configuration. The addresses of the other input and output objects are also updated.



Note!

- Changes of the EtherCAT configuration in the »PLC Designer« are overwritten when the EtherCAT configuration is re-imported. Always adapt the EtherCAT configuration to the »EtherCAT Configurator«.
- In a mixed operation, it must always be ensured that the CAN-Motion task has the highest priority. The task assigned to the EtherCAT bus should have the second-highest priority. The tasks assigned to the Logic bus systems should be configured with a lower priority.

9 EtherCAT function libraries

The **SM_Ethercat.lib** and **AtEm.LIB** function libraries contain function blocks and functions that are required for the creation of your »PLC Designer« project.

These function blocks and functions support:

- ▶ The setting and reading of the master/slave states;
- ▶ The network management;
- ▶ The network diagnostics;
- ▶ The upload and download of CoE parameters.

9.1 Usability

The **SM_Ethercat.lib** and **AtEm.LIB** function libraries are integrated in the following target systems:

- ▶ L-force Logic x700 from version 6.x
- ▶ L-force Motion x700 from version 6.x

Via the **Library manager** of the »PLC Designer« you can integrate the function libraries into the »PLC Designer« project. They can be found in the library directory of the corresponding target system in the subdirectory "EtherCAT".



Note!

The integration of the **SM_EthercatDrive.lib** function library into a »PLC Designer« project is not required, as the function blocks and the functions of this library are not relevant.

In the course of this chapter only the function blocks and functions required for the creation of a »PLC Designer« project are described.

9.2 Function blocks/functions required for a »PLC Designer« project (overview)

**Note!**

Use the function blocks and functions of the function library **AtEm.lib** only if the master is configured completely. Otherwise it can lead to instabilities of the control system. A check of the master configuration can be executed with the [ecatMasterIsConfigured \(FUN\)](#) (108) function.

Function library	Function block (FB) / function (FUN)
SM_Ethercat.lib	ECATDiagnostic (FB) (113)
	ResetMasterStatus (FB) (114)
	SMC_ETCErrorString (FUN) (115)
	L_ECAT_ReadErrCnt (FB) (116)
	L_ECAT_ResetErrCnt (FB) (117)
AtEm.lib	ecatCoeSdoDownloadReq (FB) (122)
	ecatCoeSdoUploadReq (FB) (123)
	ecatGetMasterState (FUN) (105)
	ecatGetNumConfiguredSlaves (FUN) (112)
	ecatGetNumConnectedSlaves (FUN) (113)
	ecatGetSlaveId (FUN) (109)
	ecatGetSlaveIdAtPosition (FUN) (110)
	ecatGetSlaveProp (FUN) (111)
	ecatGetSlaveState (FUN) (106)
	ecatGetSlaveStateAsync (FB) (107)
	ecatMasterIsConfigured (FUN) (108)
	ecatSetMasterStateAsync (FB) (102)
	ecatSetSlaveStateAsync (FB) (103)
	ecatStartAsync (FB) (100)
	ecatStopAsync (FB) (101)
	The global EtherCAT master structure ECAT_MASTER (118)

9.3 Properties of function blocks

The function blocks are equipped with inputs and outputs for the activation of the POU, the display of the current POU status, and for the output of error messages:

Input/output	Data type	Action
bExecute / bEnable	BOOL	<i>bExecute</i> and <i>bEnable</i> are edge-controlled: In the case of a positive edge, the function block is executed.
bBusy	BOOL	As long as a function block is executed, <i>bBusy</i> is TRUE and <i>bDone</i> is FALSE.
bDone	BOOL	If a function block has been executed, <i>bDone</i> is set to TRUE and <i>bBusy</i> is set to FALSE. If <i>bExecute/bEnable</i> has been reset, <i>bDone</i> is only active during the function block call. If <i>bExecute/bEnable</i> is TRUE, then <i>bDone</i> remains TRUE as long as <i>bExecute/bEnable</i> is reset.
bError	BOOL	If an error has occurred, <i>bError</i> is set to TRUE.
dwErrorCode	DWORD	If an error has occurred (output <i>bError</i> = TRUE), a hexadecimal error code (see also System error messages (157)) is displayed at the output <i>dwErrorCode</i> . The error code is only available during the function block call (if <i>bDone</i> = TRUE).

9.4 The EC_T_STATE structure



Note!

The identifiers itemised here cannot be found in IEC 61131. For all state inputs and outputs, always the numerical values are indicated.

The **ECAT_STATE** structure describes all possible states of the EtherCAT bus system:

Status	Identifier	Value (DINT)
Unknown	eEcatState_UNKNOWN	0
Initialization	eEcatState_INIT	1
Pre-Operational	eEcatState_PREOP	2
Bootstrap Mode (Is currently not supported.)	eEcatState_BOOT	3
Safe-Operational	eEcatState_SAFEOP	4
Operational	eEcatState_OP	8

9.5 Function blocks and functions for master/slave states

The function blocks and functions described in the following support the setting and reading of the master/slave states.

- ▶ [ecatStartAsync \(FB\)](#) (100)
- ▶ [ecatStopAsync \(FB\)](#) (101)
- ▶ [ecatSetMasterStateAsync \(FB\)](#) (102)
- ▶ [ecatSetSlaveStateAsync \(FB\)](#) (103)
- ▶ [ecatGetMasterState \(FUN\)](#) (105)
- ▶ [ecatGetSlaveState \(FUN\)](#) (106)
- ▶ [ecatGetSlaveStateAsync \(FB\)](#) (107)

9.5.1 ecatStartAsync (FB)

Function: This function block sets the master and all slaves connected to the *Operational* status.

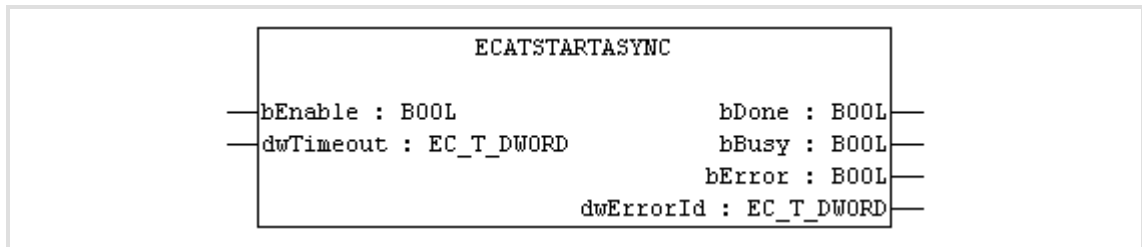
Library: AtEm.lib

Visualisation: VISU_ecatStartAsync



Note!

State changes are usually carried out automatically by the control system. By means of this function block you can change the state manually. Please note that the bus system may become unstable if the state is changed manually.



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
bEnable BOOL	The function block is activated in an edge-controlled manner: Positive edge (TRUE) = function block is executed.
dwTimeout DWORD	Time-out in milliseconds - If the action could not be carried out successfully after time-out, <i>bError</i> is set to TRUE. - The time required for the action depends on the structure of the EtherCAT network.

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Function block is active or has not been called TRUE: Function block has been executed.
bBusy BOOL	FALSE: Function block is not active. TRUE: Function block is active.
bError BOOL	FALSE: No error TRUE: An error has occurred.
dwErrorCode DWORD	Display of a hexadecimal error code (see also System error messages (157)) if an error has occurred (<i>bError</i> = TRUE).

9.5.2 ecatStopAsync (FB)

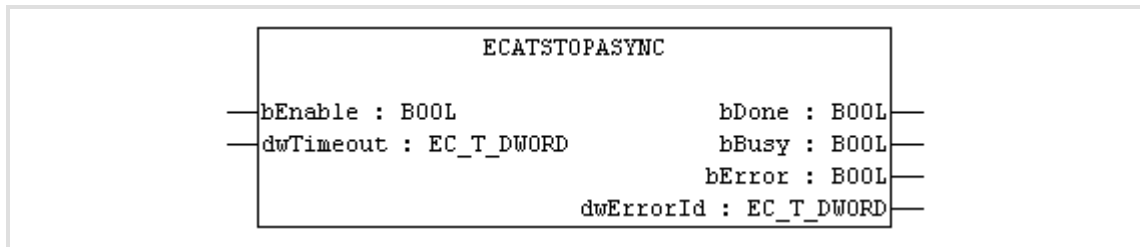
Function: This function block sets the master and all slaves connected to the "Initialization" state.

Library: AtEm.lib

Visualisation: VISU_ecatStopAsync

**Note!**

State changes are usually carried out automatically by the control system. By means of this function block you can change the state manually. Please note that the bus system may become unstable if the state is changed manually.

**Inputs (VAR_INPUT)**

Identifier/data type	Meaning/possible settings
bEnable BOOL	The function block is activated in an edge-controlled manner: Positive edge (TRUE) = function block is executed.
dwTimeout DWORD	Time-out in milliseconds - If the action could not be carried out successfully after time-out, <i>bError</i> is set to TRUE. - The time required for the action depends on the structure of the EtherCAT network.

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Function block is active or has not been called TRUE: Function block has been executed.
bBusy BOOL	FALSE: Function block is not active. TRUE: Function block is active.
bError BOOL	FALSE: No error TRUE: An error has occurred.
dwErrorCode DWORD	Display of a hexadecimal error code (see also System error messages (157)) if an error has occurred (<i>bError</i> = TRUE).

9.5.3 ecatSetMasterStateAsync (FB)

Function: This function block sets the master and all slaves connected to the state requested at the function block (see chapter "[The EC T STATE structure](#)" (198)).

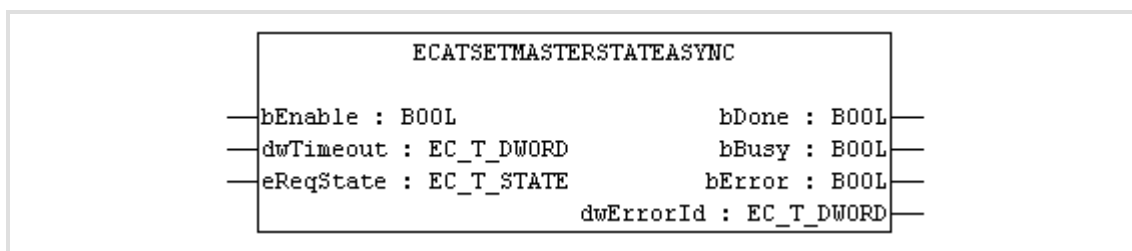
Library: AtEm.lib

Visualisation: VISU_ecatSetMasterStateAsync



Note!

- State changes are usually carried out automatically by the control system. By means of this function block you can change the state manually. Please note that the bus system may become unstable if the state is changed manually.
- The ecatSetMasterStateAsync function block can be used to set the master and the slaves to the *Operational* state. In this case, however, no re-synchronisation of the distributed clocks initiated. Use the [ecatStartAsync \(FB\)](#) (100) function block for this purpose.



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
bEnable BOOL	The function block is activated in an edge-controlled manner: • Positive edge (TRUE) = function block is executed.
dwTimeout DWORD	Time-out in milliseconds • If the action could not be carried out successfully after time-out, <i>bError</i> is set to TRUE. • The time required for the action depends on the structure of the EtherCAT network.
eReqState STATE	Requested state

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Function block is active or has not been called TRUE: Function block has been executed.
bBusy BOOL	FALSE: Function block is not active. TRUE: Function block is active.
bError BOOL	FALSE: No error TRUE: An error has occurred.
dwErrorCode DWORD	Display of a hexadecimal error code (see also System error messages (157)) if an error has occurred (<i>bError</i> = TRUE).

9.5.4 ecatSetSlaveStateAsync (FB)

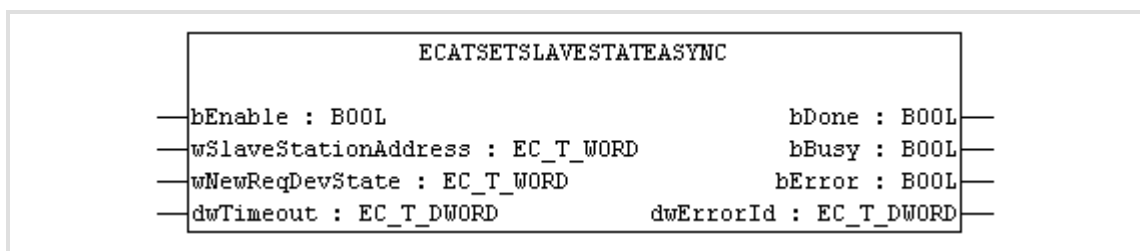
Function: This function block sets a specific slave to the state requested at the block (see chapter "[The EC T STATE structure](#)" (98)). The function block can only be executed if the master is in the *Operational* state.

Library: AtEm.lib

Visualisation: VISU_ecatSetSlaveStateAsync

**Note!**

State changes are usually carried out automatically by the control system. By means of this function block you can change the state manually. Please note that the bus system may become unstable if the state is changed manually.



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
bEnable BOOL	The function block is activated in an edge-controlled manner: • Positive edge (TRUE) = function block is executed.
wSlaveStationAddress WORD	Station address of the slave • Designation in the EtherCAT configurator: "EtherCAT address" • Designation in the EtherCAT specification and parameter reference: "Physical address".
pwNewReqDevState WORD	Requested state
dwTimeout DWORD	Time-out in milliseconds • If the action could not be carried out successfully after time-out, <i>bError</i> is set to TRUE. • The time required for the action depends on the structure of the EtherCAT network.

Outputs (VAR_OUTPUT)

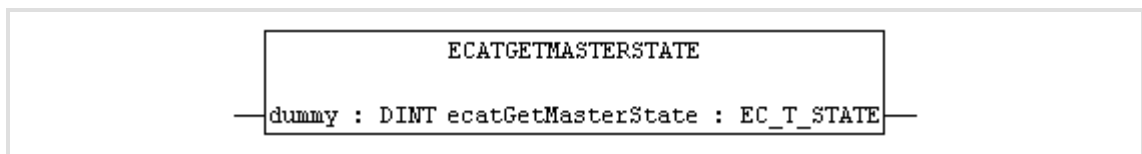
Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Function block is active or has not been called TRUE: Function block has been executed.
bBusy BOOL	FALSE: Function block is not active. TRUE: Function block is active.
bError BOOL	FALSE: No error TRUE: An error has occurred.
dwErrorCode DWORD	Display of a hexadecimal error code (see also System error messages (157)) if an error has occurred (<i>bError</i> = TRUE).

9.5.5 ecatGetMasterState (FUN)

Function: This function shows the current state of the master (see chapter "[The EC_T_STATE structure](#)" (98)).

Library: AtEm.lib

Visualisation: -



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
dummy DINT	No function

Return value

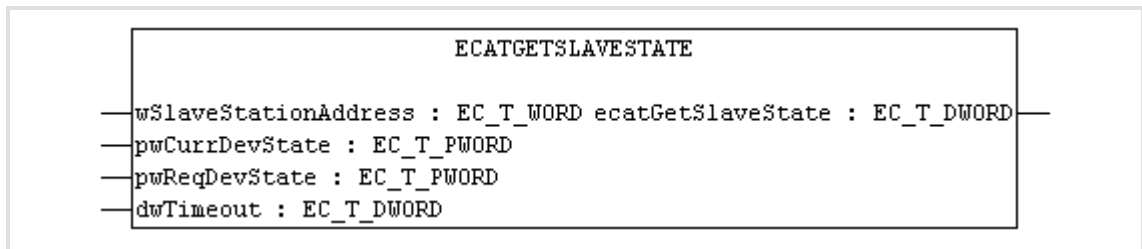
Identifier/data type	Meaning/possible settings
ecatGetMasterState STATE	Current state of the master

9.5.6 ecatGetSlaveState (FUN)

Function: This function shows the current state of the slave (see chapter "[The EC T STATE structure](#)" (98)).

Library: AtEm.lib

Visualisation: -



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
wSlaveStationAddress WORD	Station address of the slave - Designation in the EtherCAT configurator: "EtherCAT address" - Designation in the EtherCAT specification and parameter reference: "Physical address".
pwCurrDevState POINTER TO WORD	Current state of the slave
pwReqDevState POINTER TO WORD	Set value of the current slave state
dwTimeout DWORD	Time-out in milliseconds - If the action could not be carried out successfully after time-out, <i>bError</i> is set to TRUE. - The time required for the action depends on the structure of the EtherCAT network.

Return value

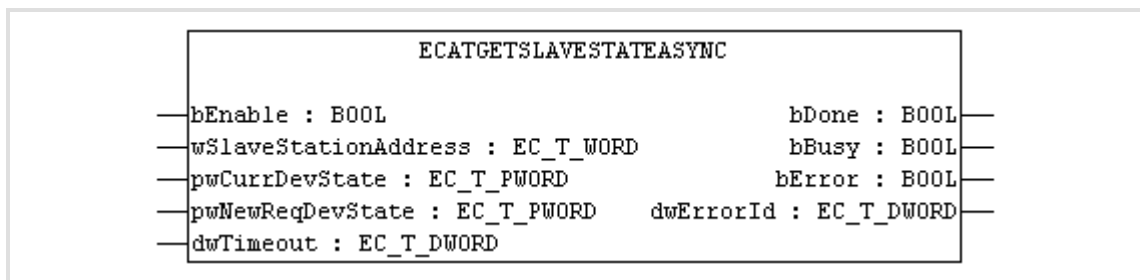
Identifier/data type	Meaning/possible settings
ecatGetSlaveState DWORD	Current state of the slave

9.5.7 ecatGetSlaveStateAsync (FB)

Function: This function block shows the current state of the slave (see chapter ["The EC T STATE structure"](#) (□ 98)).

Library: AtEm.lib

Visualisation: VISU_ecatGetSlaveStateAsync

**Inputs (VAR_INPUT)**

Identifier/data type	Meaning/possible settings
bEnable BOOL	The function block is activated in an edge-controlled manner: Positive edge (TRUE) = function block is executed.
wSlaveStationAddress WORD	Station address of the slave - Designation in the EtherCAT configurator: "EtherCAT address" - Designation in the EtherCAT specification and parameter reference: "Physical address".
pwCurrDevState POINTER TO WORD	Current state of the slave
pwNewReqDevState POINTER TO WORD	State last requested at the slave.
dwTimeout DWORD	Time-out in milliseconds - If the action could not be carried out successfully after time-out, <i>bError</i> is set to TRUE. - The time required for the action depends on the structure of the EtherCAT network.

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Function block is active or has not been called TRUE: Function block has been executed.
bBusy BOOL	FALSE: Function block is not active. TRUE: Function block is active.
bError BOOL	FALSE: No error TRUE: An error has occurred.
dwErrorCode DWORD	Display of a hexadecimal error code (see also System error messages (□ 157)) if an error has occurred (<i>bError</i> = TRUE).

9.6 Functions for the network management

The functions described in the following support the network management:

- ▶ [ecatMasterIsConfigured \(FUN\)](#) (108)
- ▶ [ecatGetSlaveId \(FUN\)](#) (109)
- ▶ [ecatGetSlaveIdAtPosition \(FUN\)](#) (110)
- ▶ [ecatGetSlaveProp \(FUN\)](#) (111)

9.6.1 ecatMasterIsConfigured (FUN)

Function: This function supplies TRUE if the master has been fully configured, and FALSE, if not.

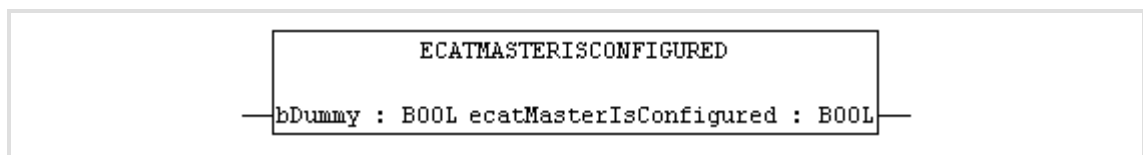
Library: AtEm.lib

Visualisation: -



Note!

This function should be evaluated at the start of every cycle requiring EtherCAT functions. If the master has not been fully configured and functions/function blocks of an EtherCAT function library are called, the control system may become unstable.



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
bDummy BOOL	No function

Return value

Identifier/data type	Meaning/possible settings
ecatMasterIsConfigured BOOL	TRUE: The master was configured correctly/successfully. FALSE: The master was not configured correctly/successfully.

Example

```

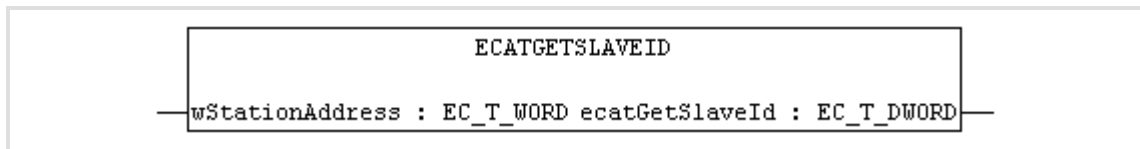
IF (ecatMasterIsConfigured(TRUE) = FALSE) THEN
RETURN;
END_IF
  
```

9.6.2 ecatGetSlaveId (FUN)

Function: This function shows the slave ID of the slave the station address of which (EtherCAT address/physical address) is configured in the control configuration file (XML). The slave ID is used by the [ecatGetSlaveProp \(FUN\)](#) (111) function.

Library: AtEm.lib

Visualisation: -

**Inputs (VAR_INPUT)**

Identifier/data type	Meaning/possible settings
wSlaveStationAddress WORD	Station address of the slave • Designation in the EtherCAT configurator: "EtherCAT address" • Designation in the EtherCAT specification and the parameter reference: "Physical address".

Return value

Identifier/data type	Meaning/possible settings
ecatGetSlaveId DWORD	Slave ID of the slave specified under <i>wSlaveStationAddress</i> .

Example

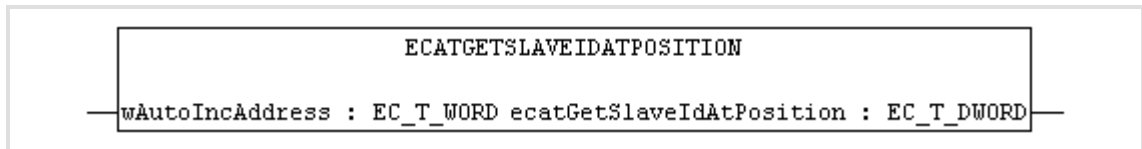
```
dwSlaveId := ecatGetSlaveId (1005);
```


9.6.3 ecatGetSlaveIdAtPosition (FUN)

Function: This function shows the slave ID of the slave the auto-increment address of which is configured in the control configuration file (XML). The slave ID is used by the [ecatGetSlaveProp \(FUN\)](#) (111) function.

Library: AtEm.lib

Visualisation: -



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
wAutoIncAddress WORD	Auto-increment address of the slave The auto-increment address is the logic position of the slave in the bus, starting with 16#0000, then descending with 16#FFFF, 16#FFFD, 16#FFFC, etc.

Return value

Identifier/data type	Meaning/possible settings
ecatGetSlaveIdAtPosition DWORD	Slave ID of the slave specified under <i>wAutoIncAddress</i> .

Example

```
dwSlaveId := ecatGetSlaveAtPosition(16#FFFC);
```


9.7 Function blocks and functions for diagnosing the network

The function blocks and functions described in the following support the diagnostics of the network:

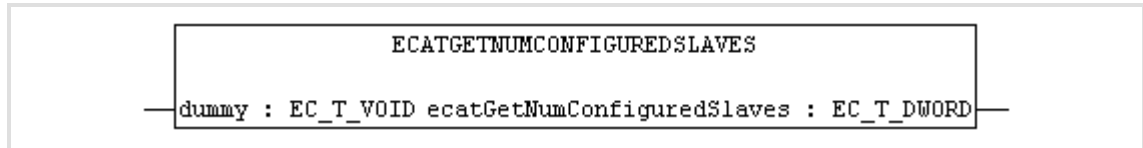
- ▶ [ecatGetNumConfiguredSlaves \(FUN\)](#) (112)
- ▶ [ecatGetNumConnectedSlaves \(FUN\)](#) (113)
- ▶ [ECATDiagnostic \(FB\)](#) (113)
- ▶ [ResetMasterStatus \(FB\)](#) (114)
- ▶ [SMC_ETCErrorString \(FUN\)](#) (115)
- ▶ [L_ECAT_ReadErrCnt \(FB\)](#) (116)
- ▶ [L_ECAT_ResetErrCnt \(FB\)](#) (117)
- ▶ [The global EtherCAT master structure ECAT_MASTER](#) (118)

9.7.1 ecatGetNumConfiguredSlaves (FUN)

Function: This function shows the number of slaves that are configured in the control configuration file (XML).

Library: AtEm.lib

Visualisation: -



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
dummy VOID	No function

Return value

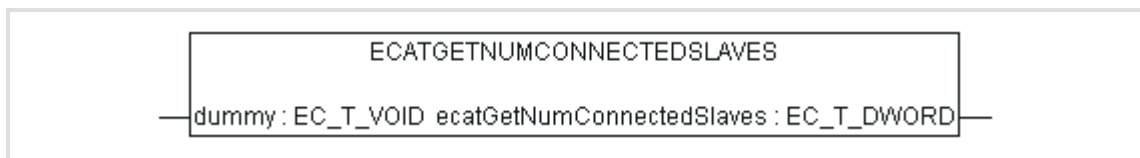
Identifier/data type	Meaning/possible settings
ecatGetNumConfigured Slaves DWORD	Number of slaves configured in the control configuration file (XML)

9.7.2 ecatGetNumConnectedSlaves (FUN)

Function: This function shows the number of slaves that are physically connected in the control system.

Library: AtEm.lib

Visualisation: -

**Inputs (VAR_INPUT)**

Identifier/data type	Meaning/possible settings
dummy VOID	No function

Return value

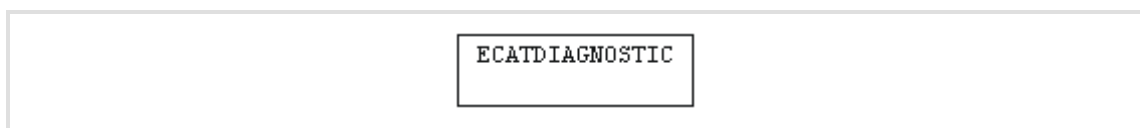
Identifier/data type	Meaning/possible settings
ecatGetNumConnected Slaves DWORD	Number of slaves that are physically connected in the control system

9.7.3 ECATDiagnostic (FB)

Function: This function block is the collection of all FBs of the **AtEm.lib** function library.

Library: SM_Ethercat.lib

Visualisation: VISU_ECATDiagnostic

**Tip!**

Using the [VISU ECATDiagnostic visualisation template](#) (136) you can operate all relevant EtherCAT functions and view all [Notifications](#) (120) in the "ETHERCATMaster status" template.

9.7.4 **ResetMasterStatus (FB)**

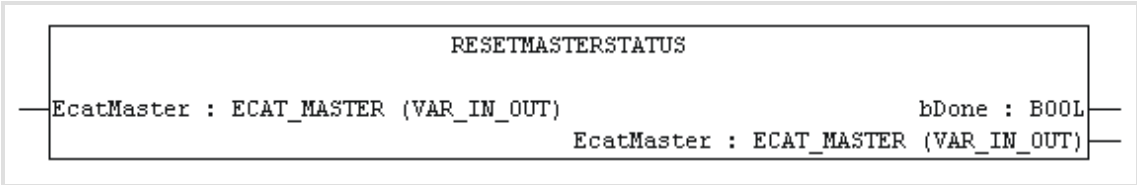
Function: This function block resets the [Notifications](#) (120). It is typically used after a bus restart. A bus restart is initiated by executing the following function blocks in the listed order:

- 1. [ecatStopAsync \(FB\)](#) (101)
- 2. [ecatStartAsync \(FB\)](#) (100)
- 3. ResetMasterStatus (FB)

(After the bus has been restarted, the distributed clocks are synchronised and the EtherCAT bus is in the *Operational* state.)

Library: SM_Ethercat.lib

Visualisation: -



Inputs and outputs (VAR_IN_OUT)

Identifier/data type	Meaning/possible settings
EcatMaster ECAT_MASTER	Input: Master state before the reset Output: Master state <i>Operational</i> (The global EtherCAT master structure ECAT_MASTER (118) contains the master state for the ECAT1 interface.)

Outputs (VAR_OUTPUT)

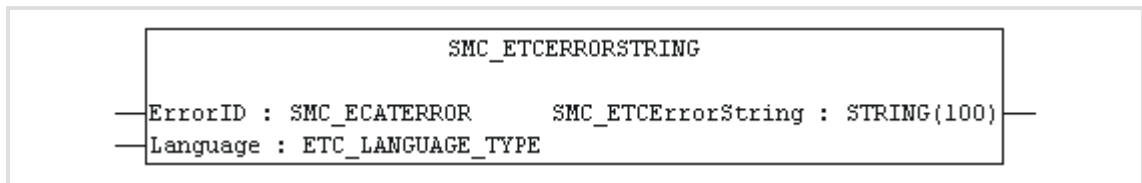
Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Function block is active or has not been called TRUE: Function block has been executed.

9.7.5 SMC_ETCErrorString (FUN)

Function: On the basis of the error number (ErrorID), this function returns the corresponding error description as a string with max. 100 characters.

Library: SM_Ethercat.lib

Visualisation: -



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
ErrorID SMC_ECATERORR	Number of the error message (see also System error messages (157))
Language ETC_LANGUAGE_TYPE	Language selection: • 0: English • 1: German

Return value

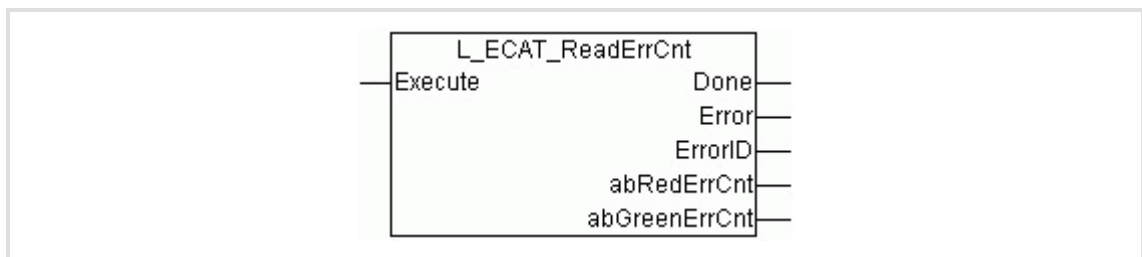
Identifier/data type	Meaning/possible settings
SMC_ECErrorString STRING[100]	Error description as a string with max. 100 characters

9.7.6 L_ECANT_ReadErrCnt (FB)

Function: This function block reads the telegram error counters of all connected slaves (see [Error counters of the EtherCAT slaves](#) (143)). This block should be executed cyclically at longer intervals (e.g. every 10 minutes) to evaluate the state of the bus. The values in the abRedErrCnt array serve to evaluate the cabling quality/EMC sensitivity of the EtherCAT bus.

Library: SM_Ethercat.lib

Visualisation: VISU_L_ECANT_ReadErrCnt



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
bExecute BOOL	The function block is activated in an edge-controlled manner: Positive edge (TRUE) = function block is executed.

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Function block is active or has not been called TRUE: Function block has been executed.
bError BOOL	FALSE: No error TRUE: An error has occurred.
dwErrorID DWORD	Display of a hexadecimal error code (see also System error messages (157)) if an error has occurred (bError = TRUE).
abRedErrCnt Array [n] of L_ECANT_ErrCnt	This array contains the values of the error counters of the telegram errors occurred for the first time of all connected slaves. - Each slave has 4 telegram error counters. 'n' corresponds to the number of connected slaves.
abGreenErrCnt Array [n] of L_ECANT_ErrCnt	This array contains the values of the error counters of the forwarded telegram errors of all connected slaves. - Each slave has 4 telegram error counters. 'n' corresponds to the number of connected slaves.

The L_ECAT_ErrCnt structure

The L_ECAT_ErrCnt structure contains the error counters of the single ports of a connected device.

```

TYPE L_ECAT_ErrCnt :
STRUCT
  Port0 : BYTE;
  Port1 : BYTE;
  Port2 : BYTE;
  Port3 : BYTE;
END_STRUCT
END_TYPE

```

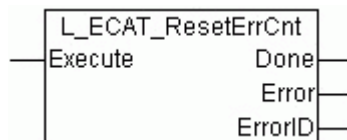
Identifier/data type	Description
Port[n] BYTE	Error counter value of the errors recorded at the port[n] n: 0 ... 3

9.7.7 L_ECAT_ResetErrCnt (FB)

Function: This function block resets the telegram error counters of all connected slaves (see [Error counters of the EtherCAT slaves](#) (□ 143)). This block should be executed before one of the telegram error counters of the slaves has reached the maximum value of 255.

Library: SM_Ethercat.lib

Visualisation: VISU_L_ECAT_ResetErrCnt



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
bExecute BOOL	The function block is activated in an edge-controlled manner: • Positive edge (TRUE) = function block is executed.

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Function block is active or has not been called TRUE: Function block has been executed.
bError BOOL	FALSE: No error TRUE: An error has occurred.
dwErrorID DWORD	Display of a hexadecimal error code (see also System error messages (□ 157)) if an error has occurred (<i>bError</i> = TRUE).

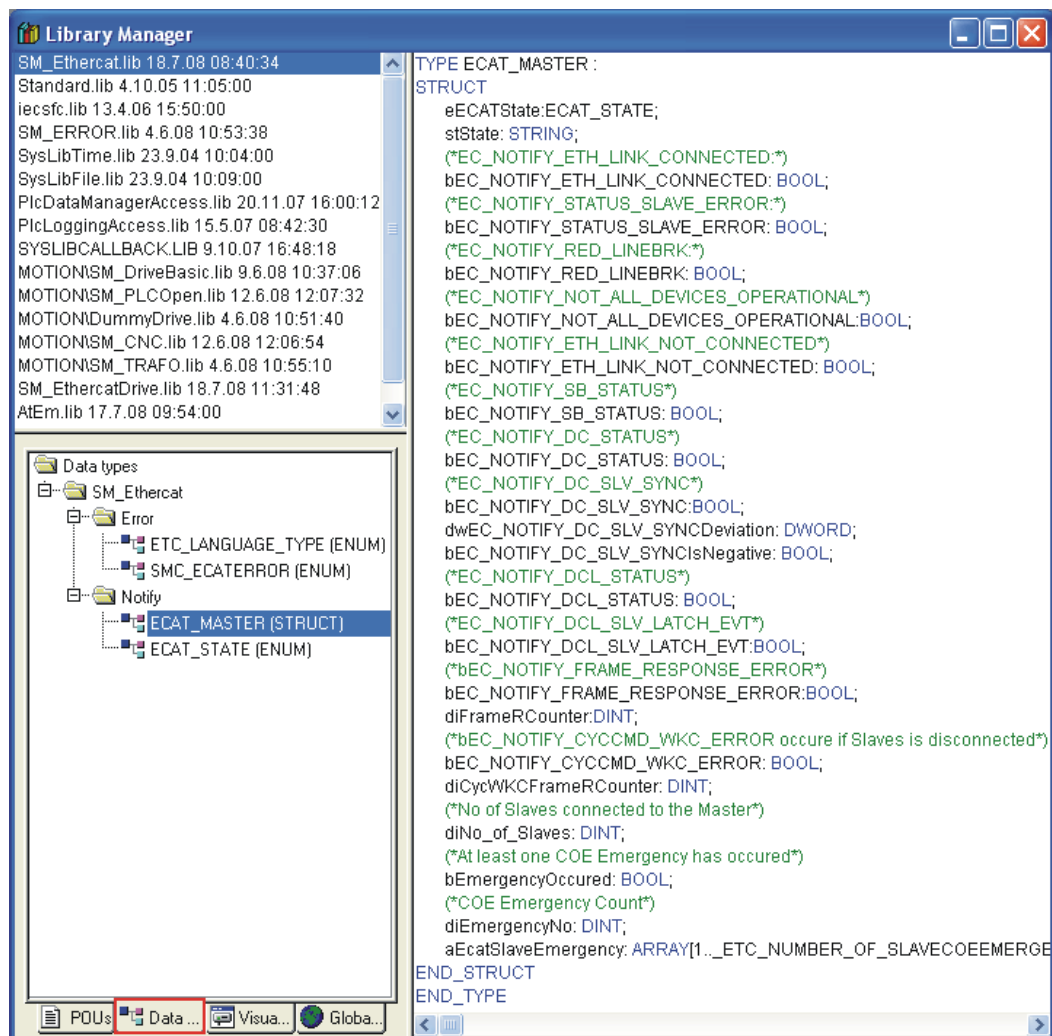
9.7.8 The global EtherCAT master structure ECAT_MASTER

Function: The global EtherCAT master structure is contained in the **SM_EthercatDrive.lib** function library. The structure comprises variables and "[Notifications](#)" (120) which indicate the state of the EtherCAT bus system.

Library: SM_Ethercat.lib

Visualisation: VISU_ETHERCATMaster

How to find the variable in the **Library Manager** of the »PLC Designer«:



9.7.8.1 Definition in the SM_EtherCAT.lib library



Note!

Only one EtherCAT master instance can be executed on an IPC. (The global variable `g_ecatMaster[1]` contains the master state for the ECAT1 interface.)

```
g_EcatMaster: ARRAY[1.._ETC_NUMBER_OF_MASTERS] OF ECAT_MASTER;
```

9.7.8.2 The data type ECAT_STATE

The data type **ECAT_STATE** describes all possible states of the EtherCAT bus system:

Status	Identifier
Unknown	SMC_ECSTATE_UNKNOWN
Initialization	SMC_ECSTATE_INIT
Pre-Operational	SMC_ECSTATE_PREOP
Bootstrap Mode (Is currently not supported.)	SMC_ECSTATE_BOOT
Safe-Operational	SMC_ECSTATE_SAFEOP
Operational	SMC_ECSTATE_OP

9.7.8.3 Variables

The variables of the EtherCAT master structure **ECAT_MASTER** provide additional information besides the state of the EtherCAT bus system:

Identifier/data type	Meaning/possible settings
eCATState ECAT_STATE	Current state of the EtherCAT bus system as an ECAT_STATE value (see chapter " The data type ECAT_STATE " (119))
stState STRING	Current state of the EtherCAT bus system as a character string Contains the types of ECAT_STATE as a character string. (E.g. "SMC_ECSTATE_INIT")
diFrameRCounter DINT	Number of response frames received by the master
diCycWKCFramerCounter DINT	Number of frames with WKC error
diNo_of_Slaves DINT	Number of slaves participating in the communication via the EtherCAT bus
bEmergencyOccured BOOL	FALSE: No emergency message has occurred. TRUE: At least one emergency message has occurred.
diEmergencyNo DINT	Number of emergency messages
aEcatSlaveEmergency ARRAY	Numbers of the emergency messages

9.7.8.4 Notifications

If an event (information, warning, error) occurs on the EtherCAT bus system, the user or the PLC program receives a "notification".

The notifications only provide information about the fact that an event of a certain type has occurred.

In the »PLC Designer«, the notifications are implemented as boolean elements in the global EtherCAT master structure **ECAT_MASTER** ([118](#)).

Overview of notifications

Notification	Description
bEC_NOTIFY_ETH_LINK_CONNECTED	This notification is set when the interrupted connection between the EtherCAT master and the first slave (see also bEC_NOTIFY_ETH_LINK_NOT_CONNECTED) is restored.
bEC_NOTIFY_STATUS_SLAVE_ERROR	When cyclic frames are processed, the EtherCAT master checks whether the ERROR bit has been set in the AL-STATUS register of at least one of the slaves. If this is the case, the error is indicated.
bEC_NOTIFY_RED_LINEBRK	Redundancy support (is currently not supported): This notification is set for every bus/communication interruption detected by the master (see also bEC_NOTIFY_ETH_LINK_NOT_CONNECTED).
bEC_NOTIFY_NOT_ALL_DEVICES_OPERATIONAL	When cyclic frames are processed, the EtherCAT master checks that all slaves are still in the <i>Operational</i> state. If at least one slave is not in the <i>Operational</i> state, this this notification is set.
bEC_NOTIFY_ETH_LINK_NOT_CONNECTED	This notification is set when the connection between the EtherCAT master and the first slave is interrupted. Note: A missing connection can only be detected directly at the master. If, for instance, a switch or a multi-channel probe is connected downstream of the master and the connection is interrupted downstream of these devices, missing frames and/or the error message EC_NOTIFY_CYCCMD_WKC_ERROR are indicated. (This only applies to the cabling directly at the master.)
bEC_NOTIFY_SB_STATUS	Bus state notification: This notification is set when the EtherCAT bus is scanned.
bEC_NOTIFY_DC_STATUS	DC state notification: This notification is either received after starting the master (ecatStart) or after loading and then starting the PLC program, if all slave clocks have successfully been updated with the compensation values for the transmission delay.
bEC_NOTIFY_DC_SLV_SYNC	This notification is set when the maximum permissible DC deviation for the connected slaves (code C1082/2 / C1582/2) has been exceeded.
dwEC_NOTIFY_DC_SLV_SYNCDeviation	Indicates the limit value of the DC deviation (in nanoseconds): 0: 1 ns 1: 3 ns 2: 7 ns 3: 15 ns 4: 31 ns 5: 63 ns 6: 127 ns 7: 255 ns ... 31: 2147483647 ns 32: reserved
bEC_NOTIFY_DCL_STATUS	DC latching state notification: This notification is set after the master has been started (ecatStart) and the DC latch instance has been initialised correctly.
bEC_NOTIFY_DCL_SLV_LATCH_EVT	DC latch notification (single mode): This notification is set when a slave generates a latch event in "single latch mode".

Notification	Description
bEC_NOTIFY_FRAME_RESPONSE_ERROR	This notification is set when the response frame currently received by the master does not correspond with the expected frame or when the master has received no response at all.
bEC_NOTIFY_CYCCMD_WKC_ERROR	This notification is set if within a cycle the current value of the working counter in the control/PLC does not correspond with the value read back via the fieldbus. (Not all of the slaves addressed have reached.)

9.7.8.5 Setting and resetting the master structure

The notifications and variables are set by the EtherCAT master stack when the event occurs. If a notification/variable has been set, it remains set until the master structure is explicitly reset.

When an error has been eliminated, the notifications/variables must be reset. This can be done with the following actions:

- ▶ Calling the [ResetMasterStatus \(FB\)](#) function block of the **SM_EtherCAT.lib** function library (typically used in Logic control systems).
- ▶ Calling the **SMC_ResetAxisGroup** PLCopen function block of the **SM_DriveBasic.lib** function library.
- ▶ Reset PLC (cold and original)

9.8 Function blocks for CANopen over EtherCAT (CoE)

The function blocks described in the following support the upload and download of CoE parameters:

▸ [ecatCoeSdoDownloadReq \(FB\)](#) (122)

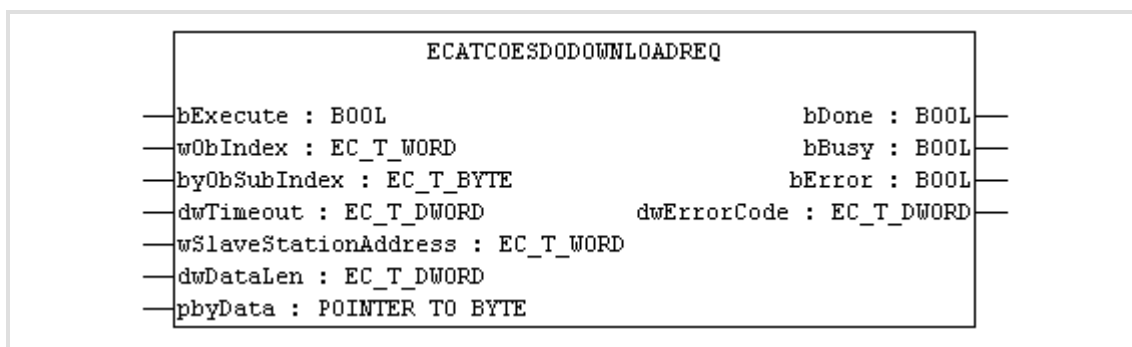
▸ [ecatCoeSdoUploadReq \(FB\)](#) (123)

9.8.1 ecatCoeSdoDownloadReq (FB)

Function: This function block activates the download of a CoE parameter to the slave.

Library: AtEm.lib

Visualisation: VISU_ecatCoeSdoDownloadReq



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
bExecute BOOL	The function block is activated in an edge-controlled manner: Positive edge (TRUE) = function block is executed.
wObIndex WORD	CANopen index The indexes for codes can be calculated with $16\#5FFF_{\text{hex}} - \text{code number}$.
byObSubIndex BYTE	Object subindex
dwTimeout DWORD	Time-out in milliseconds - If the action could not be carried out successfully after time-out, <i>bError</i> is set to TRUE. - The time required for the action depends on the structure of the EtherCAT network.
wSlaveStationAddress WORD	Station address of the slave - Designation in the EtherCAT configurator: "EtherCAT address" - Designation in the EtherCAT specification and parameter reference: "Physical address".
dwDataLen DWORD	Number of the data bytes to be written
pbyData POINTER TO BYTE	Pointer to the service data object (SDO) to be written

Outputs (VAR_OUTPUT)

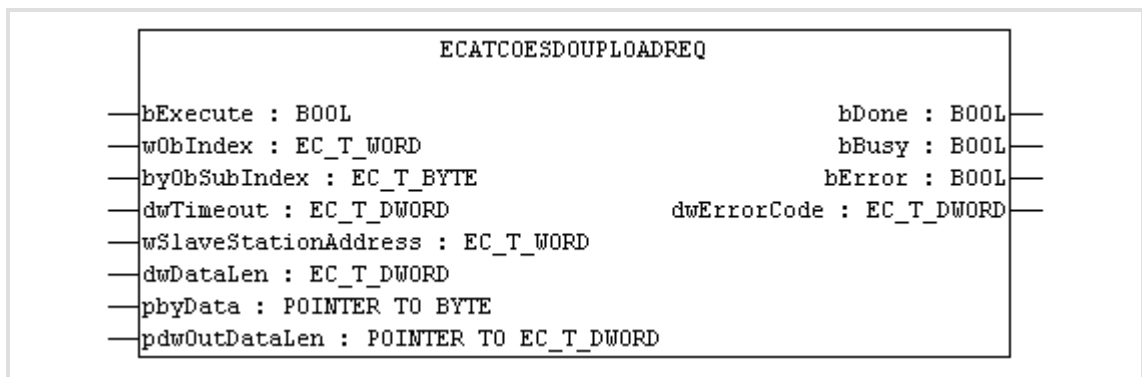
Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Download is active. TRUE: Download has been executed.
bBusy BOOL	FALSE: Download is not active. TRUE: Download is active.
bError BOOL	FALSE: No error TRUE: An error has occurred.
dwErrorCode DWORD	Display of a hexadecimal error code (see also System error messages (□ 157)) if the download has failed (<i>bError</i> = TRUE).

9.8.2 ecatCoeSdoUploadReq (FB)

Function: This function block activates the upload of a CoE parameter from the slave to the master.

Library: AtEm.lib

Visualisation: VISU_ecatCoeSdoUploadReq



Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
bExecute BOOL	The function block is activated in an edge-controlled manner: Positive edge (TRUE) = function block is executed.
wObIndex WORD	CANopen index The indexes for codes can be calculated with $16\#5FFF_{\text{hex}}$ - code number.
byObSubIndex BYTE	Object subindex
dwTimeout DWORD	Time-out in milliseconds - If the action could not be carried out successfully after time-out, <i>bError</i> is set to TRUE. - The time required for the action depends on the structure of the EtherCAT network.
wSlaveStationAddress WORD	Station address of the slave - Designation in the EtherCAT configurator: "EtherCAT address" - Designation in the EtherCAT specification and parameter reference: "Physical address".

Identifier/data type	Meaning/possible settings
dwDataLen DWORD	Number of the data bytes to be read
pbyData POINTER TO BYTE	Pointer to the service data object (SDO) to be written
pdwOutDataLen POINTER TO DWORD	Size of the memory buffer transmitted under <i>pByData</i> . The memory buffer has to be great enough to contain the object read.

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings
bDone BOOL	FALSE: Upload is active. TRUE: Upload has been executed.
bBusy BOOL	FALSE: Upload is not active. TRUE: Upload is active.
bError BOOL	FALSE: No error TRUE: An error has occurred.
dwErrorCode DWORD	Display of a hexadecimal error code (see also System error messages (□ 157)) if the upload has failed (<i>bError</i> = TRUE).

10 Defining the minimum cycle time of the PLC project

This chapter will inform you on how the minimum cycle time of the PLC project can be defined.

The minimum cycle time is calculated in three steps:

1. Calculating the total access time $T_{\text{Correction}}$ to the peripheral devices.
 - ▶ [Calculating the total access time to the peripheral devices \(\$T_{\text{Correction}}\$ \)](#) (125)
2. Detecting the task utilisation $T_{\text{Task utilisation}}$ of the application during operation.
 - ▶ [Detecting the task utilisation of the application \(\$T_{\text{Task utilisation}}\$ \)](#) (126)
3. Calculating the minimum cycle time.
 - ▶ [Calculating the minimum cycle time](#) (128)
4. Optimising the system.
 - ▶ [Optimising the system](#) (129)

10.1 Calculating the total access time to the peripheral devices ($T_{\text{Correction}}$)

The cycle times depend on the number of configured field devices and the IPC hardware used.

Configuration	Access time with processor			
	Intel Celeron 600 MHz	Intel Celeron 1 GHz	Intel Celeron 1.5 GHz	Intel Pentium M 1.8 GHz
EtherCAT bus: Axis_IO_Group and IO_Group	160 μs	130 μs	100 μs	90 μs
EtherCAT bus: per motion axis or logic field device	10 μs	5 μs	5 μs	5 μs



Note!

The LX 800 system is not supported with EtherCAT.

Example

Configuration with Intel Celeron 1 GHz processor	Access time	Total access time
Axis_IO_Group	130 μs	
six Motion axes	+ 6 x 5 μs	= 160 μs

10.2 Detecting the task utilisation of the application ($T_{\text{Task utilisation}}$)

The time $T_{\text{Task utilisation}}$ cannot be calculated. It is determined in the running system. For this the system is commissioned on the basis of cycle times that are sufficiently long, and afterwards it is optimised.

In order to detect the task utilisation, use the task editor in the »PLC Designer«.

10.2.1 Display of the system utilisation in the »PLC Designer« with the task editor



Note!

In order to be able to display the utilisation for all tasks, the IEC61131 *SysTaskInfo* library has to be included in the project.

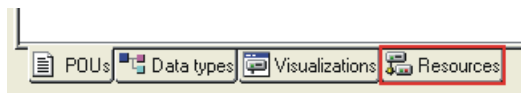
The task editor contains a dialog window consisting of two parts.

- ▶ The left part represents the tasks in a configuration tree.
- ▶ If the *Task configuration* entry is highlighted, the utilisation for all tasks is shown in bar diagrams in the right dialog window.

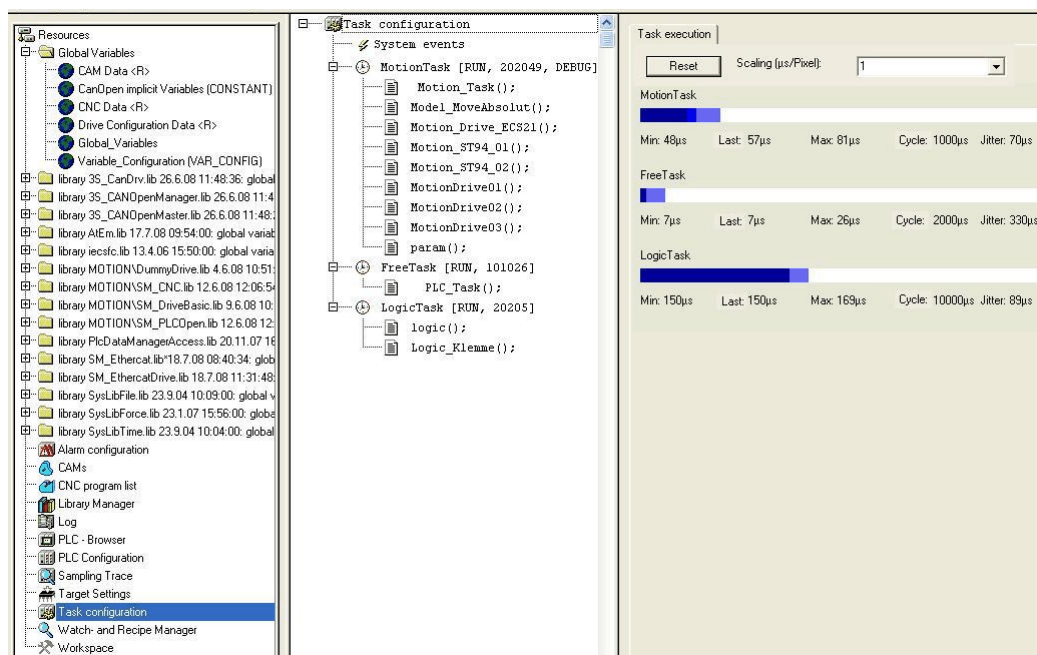


How to display the system utilisation:

1. Select the *Resources* tab:



2. Open the *task configuration* in the online mode of the »PLC Designer«.



10.2.2 Detecting the task utilisation

Initial situation

A project has been completely created with, for instance, a Motion task and two tasks of lower priority.



How to detect the task utilisation $T_{\text{Task utilisation}}$:

1. For a first measurement of $T_{\text{Task utilisation}}$ the cycle times of all cyclic tasks in the PLC system are set to 'long'.
 - Example: Motion task = 10 ms, all other cyclic tasks = 20 ms
2. Log in and load project.
3. After the system has started up completely, press the **Reset** button on the *Task processing* tab.
 - The displayed task runtimes are reset.
4. Read the maximum computing time of the task with the highest priority that is shown in the task configuration ($T_{\text{Task utilisation}}$).

10.3 Calculating the minimum cycle time



Note!

The safety factor 1.5 is included in the calculation.

The minimum cycle time $T_{\min}$ for a system results from the sum of the times detected before, multiplied by the safety factor:

$$T_{\min} > \text{Safety factor} * (T_{\text{Task utilisation}} + T_{\text{Correction}})$$

Example

Configuration: System with Intel Celeron 1 GHz processor and six motion axes

Detected access time		Result
Calculated correction value	$T_{\text{Correction}}$	160 μs (= 130 μs + 6 * 5 μs)
Value read from task configuration:	$T_{\text{Task utilisation}}$	500 μs
Actual required computing time		660 μs
Minimum cycle time including the safety factor 1.5	$T_{\min}$	990 μs
Actual cycle time		1000 μs

10.4 Optimising the system



How to optimise the system:

1. Log in and load project.
2. Check the task processing times.
3. Optimising the cycle times:
 - If required technologically, the cycle times of the remaining tasks with lower priorities can be decreased.
 - Condition: No task with a low priority may assign more than 60 percent of the corresponding cycle time in its task utilisation.

11 Diagnostics

This chapter provides information about diagnostic tools, error scenarios for the most frequent user errors and system error messages.

Use the following tools for diagnostics:

- ▶ [Diagnostics with the »EtherCAT Configurator«](#) (📖 131)
- ▶ [Diagnostics with the »PLC Designer«](#) (📖 134)
- ▶ [Diagnostic codes](#) (📖 140) in the »Engineer« and the »WebConfig«
- ▶ [Logbook of the IPC](#) (📖 140)

11.1 Diagnostics with the »EtherCAT Configurator«

The »EtherCAT Configurator« provides the following tools for system diagnostics:

- ▶ ["Diagnostics" tab](#) (131)
- ▶ [Representation in the online mode](#) (133)

11.1.1 "Diagnostics" tab

When an online connection has been established to the IPC, the **Diagnostics** tab of the EtherCAT master displays different status information on the master, distributed clocks and the EtherCAT nodes.

- ▶ The checkmarks (☑) are set at the corresponding status information if this applies.

The screenshot shows the 'Diagnostic' tab with the following status information:

Master	
Configurator equal Stack Configuration	☐
Master ok	☐
Master State	OP
Master in requested Mode	☒
Bus Scan Match	☒
Link Up	☒

Distributed Clocks	
DC is enabled	☒
DC in Sync	☐
DC busy	☐

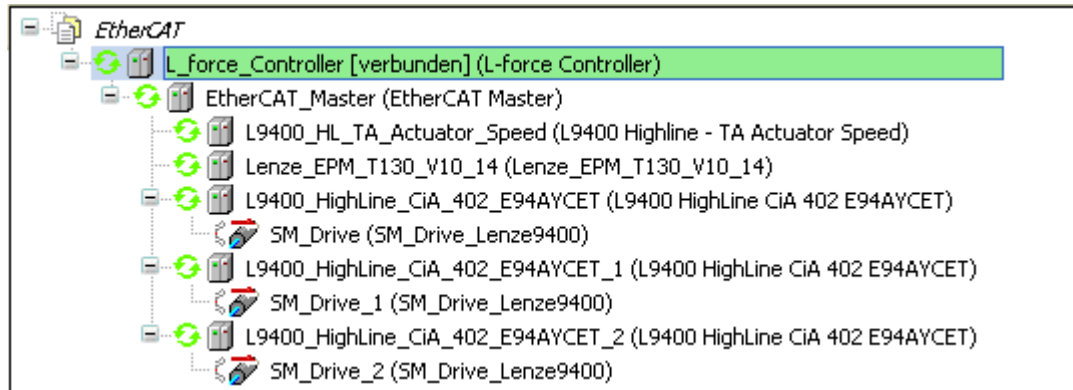
Slaves	
Slaves in requested Mode	☒

Dialog box/section	Function
Master	Status information on the EtherCAT master: - Configurator corresponds to the stack configuration: - Information whether the configuration file exported from the »Engineer« matches the configuration that is loaded in the master stack. - Master OK (active if the value from code C1081.5/C1581.5 = "Master OK") - Current master status: OK - Master status (value from code C1081.2/C1581.2*) - Possible states are: - Unknown - INIT (initialisation) - PREOP (<i>Pre-Operational</i>) - SAFEOP (<i>Safe-Operational</i>) - OP (<i>Operational</i>) - Master in the requested mode (active if value from code C1081.3/C1581.3* = "1") - Master status is identical to the requested mode (by PLC). - Bus scan compliance (active if value from code C1080.4/C1584.4* = "1") - Information whether the master configuration corresponds to the physical bus structure. The master configuration of the stack is compared with the actual bus structure. - Network link available: - Information on the active Ethernet connection of the EtherCAT card.
Distributed clocks	Status information on the distributed clocks (DC) settings (value from code C1082.1/C1582.1*). Possible states are: - active - DC synchronisation is activated. - in sync - Distributed clocks in the EtherCAT system are synchronised - busy - DC synchronisation adjustment takes place. More information on how to configure the settings of the distributed clocks: ► Synchronisation with "Distributed clocks" (36)
Slaves	Status information on the EtherCAT nodes (active if value from code C1081.4/C1581.4* = "1") Slaves are in the status required by the master

* Depending on the IPC slot used (1/2) of the MC-ETC communication card, the first or second code is relevant.

11.1.2 Representation in the online mode

When an online connection has been established to the Industrial PC, the icon in front of the single entries in the configuration tree provides the information on the status of the corresponding EtherCAT node (in the example: two green arrows each):



Icon	Meaning
	Device is online - Successful online connection to the <i>L-force controller</i> - Successful online connection to the EtherCAT node. Status of the node: <i>Operational (OP)</i>
	Device status is unknown Status: <i>Unknown</i>
	Device is online Possible states of the EtherCAT node: - INIT (initialisation) - PREOP (<i>Pre-Operational</i>) - SAFEOP (<i>Safe-Operational</i>)
no icon	Device is offline No connection to the Industrial PC (<i>L-force controller</i>)

11.2 Diagnostics with the »PLC Designer«

The »PLC Designer« provides the following tools for system diagnostics:

- ▶ [The global EtherCAT master structure ECAT_MASTER](#) (📖 118)
- ▶ [VISU ETHERCATMaster visualisation template](#) (📖 135)
- ▶ [VISU ECATDiagnostic visualisation template](#) (📖 136)
- ▶ [Function blocks and functions for diagnosing the network](#) (📖 112)
- ▶ [The global variable wState](#) (📖 137)

11.2.1 VISU_ETHERCATMaster visualisation template

The **VISU_ETHERCATMaster** visualisation template of the »PLC Designer« displays the [Notifications](#) (▢ 120) of the global EtherCAT master variable (function library SM_Ethercat.lib).



Note!

- When the target system is set to **L-force_Motion_x700_Vx.xx.xx**, the displays of the "EtherCAT master status" are always correct even if no instance of [ECATDiagnostic \(FB\)](#) (▢ 113) is available in the »PLC Designer«.
- When the target system is set to **L-force_Logic_x700_Vx.xx.xx**, the displays of the "EtherCAT master status" are only updated if an instance of [ECATDiagnostic \(FB\)](#) (▢ 113) is called in the »PLC Designer« project. Without such an instance, there will be no display.

First, all notification fields are white. If a notification is set or active (TRUE), the corresponding field is displayed in green or red:

- Red fields represents an "error".
- Green fields display an "information".

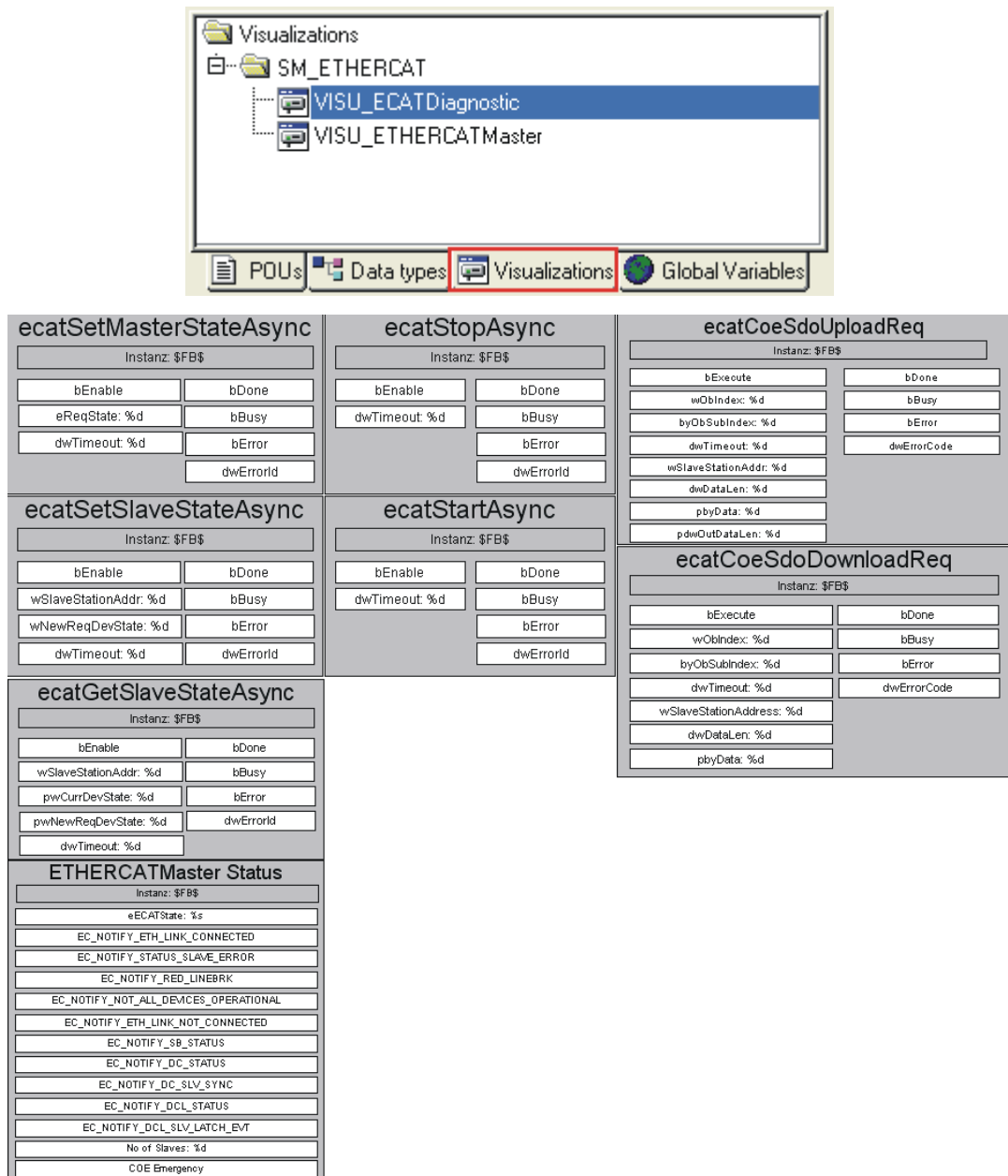
The screenshot shows the **VISU_ETHERCATMaster** visualization template. On the left, a tree view shows the project structure: Visualizations > SM_EtherCAT > VISU_ETHERCATMaster. The main area displays the **ETHERCATMaster Status** panel. This panel contains a list of status indicators, each with a label and a corresponding color-coded bar (green for information, red for error). The indicators are as follows:

Label	Status Color
Instanz: \$FB\$	White
eECATState: %s	White
EC_NOTIFY_ETH_LINK_CONNECTED	Green
EC_NOTIFY_STATUS_SLAVE_ERROR	Red
EC_NOTIFY_RED_LINEBRK	Red
EC_NOTIFY_NOT_ALL_DEVICES_OPERATIONAL	Red
EC_NOTIFY_ETH_LINK_NOT_CONNECTED	Red
EC_NOTIFY_SB_STATUS	Green
EC_NOTIFY_DC_STATUS	Green
EC_NOTIFY_DC_SLV_SYNC	Red
EC_NOTIFY_DCL_STATUS	Green
EC_NOTIFY_DCL_SLV_LATCH_EVT	Red
EC_NOTIFY_FRAME_RESPONSE_ERROR	Red
EC_NOTIFY_CYCCMD_WKC_ERROR	Red
No of Slaves: %d	White
COE Emergency	Red

The bottom of the window shows a toolbar with icons for POU, Data types, Visualizations, and Resources.

11.2.2 VISU_ECATDiagnostic visualisation template

The **VISU_ECATDiagnostic** visualisation template of the »PLC Designer« can be used to integrate the function blocks of the function libraries **SM_Ethercat.lib** and **AtEm.lib**. Then you can operate all relevant EtherCAT functions and see all notifications in the "ETHERCATMaster Status" template (Fig. below on the left):



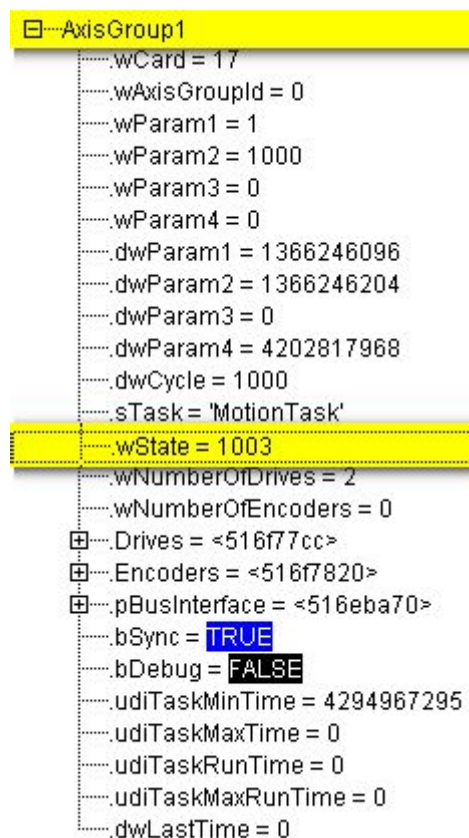
11.2.3 The global variable *wState*

When Motion nodes are used for control, »PLC Designer« displays the current status of the control acceleration in the global variable *wState* of the "AxisGroup" structure.

The value of the *wState* variable has the following meaning:

Status of the AxisGroup	State of the system
<i>wState</i> = 0	- Initial state - Project loaded - PLC in stop
<i>wState</i> = 1...99	- System is starting up - Project loaded - PLC started
<i>wState</i> = 100	System has started up successfully
<i>wState</i> > 1000	Error occurred during startup, compare error message in <i>g_strBootupError</i>

Example for a faulty acceleration:



Additional information on the type of error occurred are provided in the global variables of the **SM_DriveBasic.lib** function library.

The *g_strBootupError* variable, for instance, contains an error text:

```
⊞---g_DRIVESTRUCT
  g_strBootupError = 'SDO 40 00 10 00 00 00 00 00 :request got no reply.'
  g_bSMCMotorolaByteOrder = FALSE
  gc_SIZEOFPARAMQUEUE = 20
  g_SMC_NUMBER_FB_ERRORS = 5
  gc_SMC_MAX_NO_ENCODERS = 7
  gc_SMC_FollowPosition = 1
  gc_SMC_FollowVelocity = 2
  gc_SMC_FollowAcceleration = 4
  gc_SMC_FollowTorque = 16
  gc_SMC_FollowCurrent = 32
⊞---g_PARAMQUEUE
```

Here, an SDO access of the control has not been responded by the slave.

11.2.4 Error scenario (example)

A slave is switched off or removed from the bus system while the system is in operation.

During operation, the master cyclically sends EtherCAT commands to read the process data from the slave and to write process data to the slave. The working counter (WKC) is incremented by the corresponding value through the slaves.

When a slave is switched off or removed, the master generates the EC_NOTIFY_CYCCMD_WKC_ERROR notification to indicate such an event.

The slave can be re-commissioned by calling the function block **SMC_ResetAxisGroup** (SM_DriveBasic.lib function library) or by a **bus restart** within the PLC program. (The distributed clocks are synchronised and the EtherCAT bus is in the *Operational* state.)

Bus restart

Since the **SMC_ResetAxisGroup** function block cannot be used in logic systems, you must program a **bus restart** in your PLC program. A **bus restart** is executed through call sequence of the following function blocks:

1. [ecatStopAsync \(FB\)](#) (📘 101)
2. [ecatStartAsync \(FB\)](#) (📘 100)
3. [ResetMasterStatus \(FB\)](#) (📘 114)

For a more detailed analysis of a problem, function blocks (see [EtherCAT function libraries](#) (📘 96)), EtherCAT parameters (see [Parameter reference](#) (📘 165)) and the IPC logbook (📘 140) are available.

11.3 Diagnostic codes

The diagnostic codes can be displayed in the »Engineer« and »WebConfig«.

- ▶ [Interface parameters of the MC-ETC communication card in slot 1](#) (167)
- ▶ [Interface parameters of the MC-ETC communication card in slot 2](#) (183)

11.4 Logbook of the IPC

»Engineer« and »WebConfig« provide access to the IPC logbook.



Note!

The **ClearLog** button deletes the contents of the logbook on the IPC without a prior confirmation prompt.

Logging

LogViewer

Device Page 1

Device Page 2

Command

Ethernet 1

Panel

PLC

User 1-5

User 6-10

McETC 1

McETC1 If

McETC1 Master

McETC1 Statistics

McETC1 Slaves

Services

Lenze

Apply & Refresh

ClearLog

LogBook

00007	12-06-2008 11:42:11	NEW application attached. (1 applications active) CAN Modules or components are initialised Information
00006	12-06-2008 11:41:40	Persist all finished: HR=0 DataManager Continuous processing Information
00005	12-06-2008 11:41:24	Starting command START_PLC DataManager A module is activated Information
00004	12-06-2008 11:40:59	No response for SDO request ID:1567, index:24564, subindex:0. CAN Busserver Communication on bus systems Warning
00003	12-06-2008 11:40:57	Starting command RELOAD_PLC_BOOT_PROJECT DataManager A module is activated Information
00002	12-06-2008 11:39:44	Persist all finished: HR=0 DataManager Continuous processing Information

11.4.1 Displaying the EtherCAT entries of the logbook

Enable the "EtherCAT" checkbox to display only the EtherCAT entries of the logbook.

Logging	Time Period Selection	All
LogViewer	- From	yyyy-mm-dd hh:mm:ss 2008 04 16 02 18 54
Device Page 1	- To	yyyy-mm-dd hh:mm:ss 2008 04 16 02 18 54
Device Page 2	- Last	1 Years
Command	Applications	All ☒
Ethernet 1		WindowsCE ☐
Panel		DeviceScanner ☐
PLC		DataManager ☐
User 1-5		AutomationShell ☒
User 6-10		Update/Restore ☐
McETC 1		CAN ☐
McETC1 If		EPL ☐
McETC1 Master		EtherCAT ☒
McETC1 Statistics		SoftPLC ☐
McETC1 Slaves		MotionKernel ☐
Services		OPC Tunnel ☐
		DMBusserver ☐
		CANBusserver ☐
		Transferservice ☐
	SoftPLC USER FB ☐	
	EtherCAT bus server ☐	
	EtherCAT Master Stack ☐	
	LogService ☐	
	WebConfig ☐	
	Unknown ☐	
	Severity	All ☐
		Information ☐
		Warning ☒
		Error ☒
		Fatal error ☒
	Facility	All ☒
		A module is activated ☐
		Modules or components are initialised ☐
		A module was disconnected ☐
		A module is deactivated ☐
		Continuous processing ☐
		Communication on bus systems ☐
		Reserved 1 ☐
		Reserved 2 ☐
		Reserved 3 ☐
		PLC_USER_LOG_LEVEL1 ☐
		PLC_USER_LOG_LEVEL2 ☐
		PLC_USER_LOG_LEVEL3 ☐
Apply & Refresh ClearLog		

11.4.2 Messages in the logbook of the industrial PC

- ▶ Errors, warnings or information are indicated as messages in the logbook.
- ▶ Read the messages in the logbook from bottom to top. The most recent message always appears at the top of the logbook.

Structure of the messages in the logbook:

- Consecutive number
- Date/time of occurrence
- Location of occurrence
- Type of EtherCAT message: CTRL, PRJ, CFG, SLV, PRJ, LLA, RAP, ISW (see [System error messages \(157\)](#))
- Error description
- Message type: error / warning / information

Examples of logbook messages

...

00007 14-06-2008 10:48:11 EtherCAT Master Stack	CTRL: Cannot set EtherCAT-Master to init! (Errorcode = 0x98110010), retry... Communication on bus systems Error
00006 14-06-2008 10:47:56 EtherCAT Master Stack	CTRL: Master state change from <init> to <unknown> Communication on bus systems Information
00005 14-06-2008 10:47:55 EtherCAT Master Stack	CTRL: Cannot set EtherCAT-Master to preop! (Errorcode = 0x98110010), retry... Communication on bus systems Error
00004 14-06-2008 10:47:50 EtherCAT Master Stack	CTRL: Master state change from <unknown>to <init> Communication on bus systems Information

...

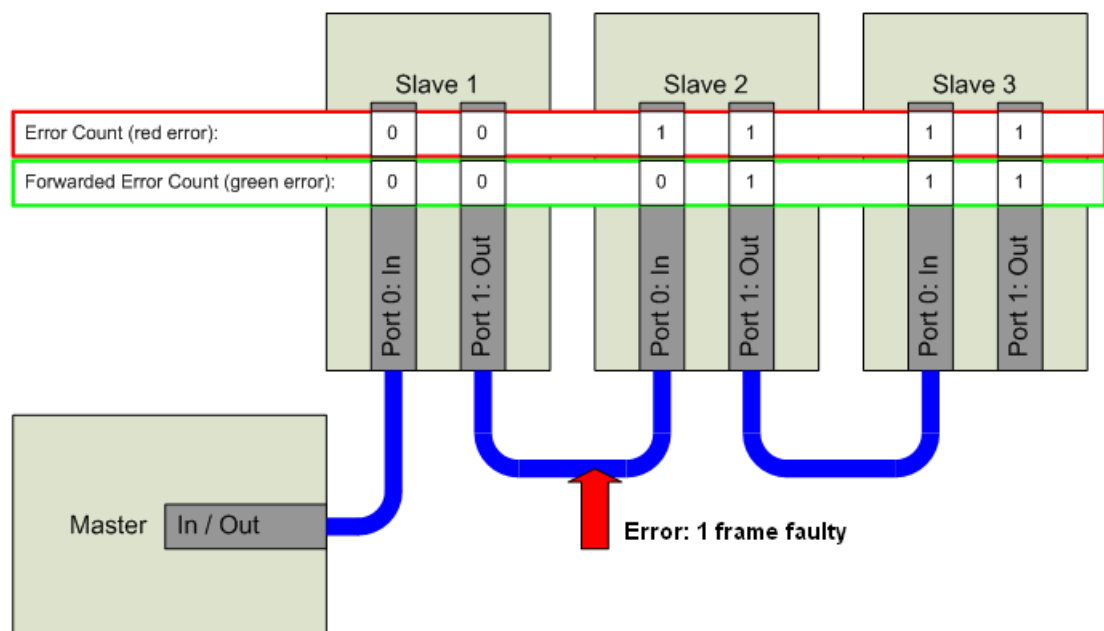
11.5 Error counters of the EtherCAT slaves

The EtherCAT slaves have numerical error counters for detecting and analysing error states. All error counters have a limited counting range of 0 ... 255. After the maximum value of 255 is reached, no "wrap-around" takes place. If the PLC application is to make a sensible evaluation of the error counters, these must be deleted by the application after the evaluation via a write access.

11.5.1 Error types: "Errors" and "Forwarded Errors"

The EtherCAT differ between errors detected in the slave for the first time (red error) and forwarded errors, i.e. errors that have already been detected in a previous slave (green error).

When the corresponding error counters are evaluated, an error in the EtherCAT network can be clearly assigned in a bus segment or a slave.



The different error counters are assigned to the following IPC parameters:

Error counters	MC-ETC in slot 1		MC-ETC in slot 2	
	Port 1	Port 2	Port 1	Port 2
Errors detected for the first time (red error)	C1096/25	C1096/26	C1096/25	C1096/26
Forwarded errors (green error)	C1096/29	C1096/30	C1096/29	C1096/30

11.5.2 Error counter reset from the application

The [L_ECAT_ReadErrCnt \(FB\)](#) (📖 116) function block enables the PLC application to access the telegram failure error counter by reading.

The [L_ECAT_ResetErrCnt \(FB\)](#) (📖 117) function block resets the error counters to the initial value 0.

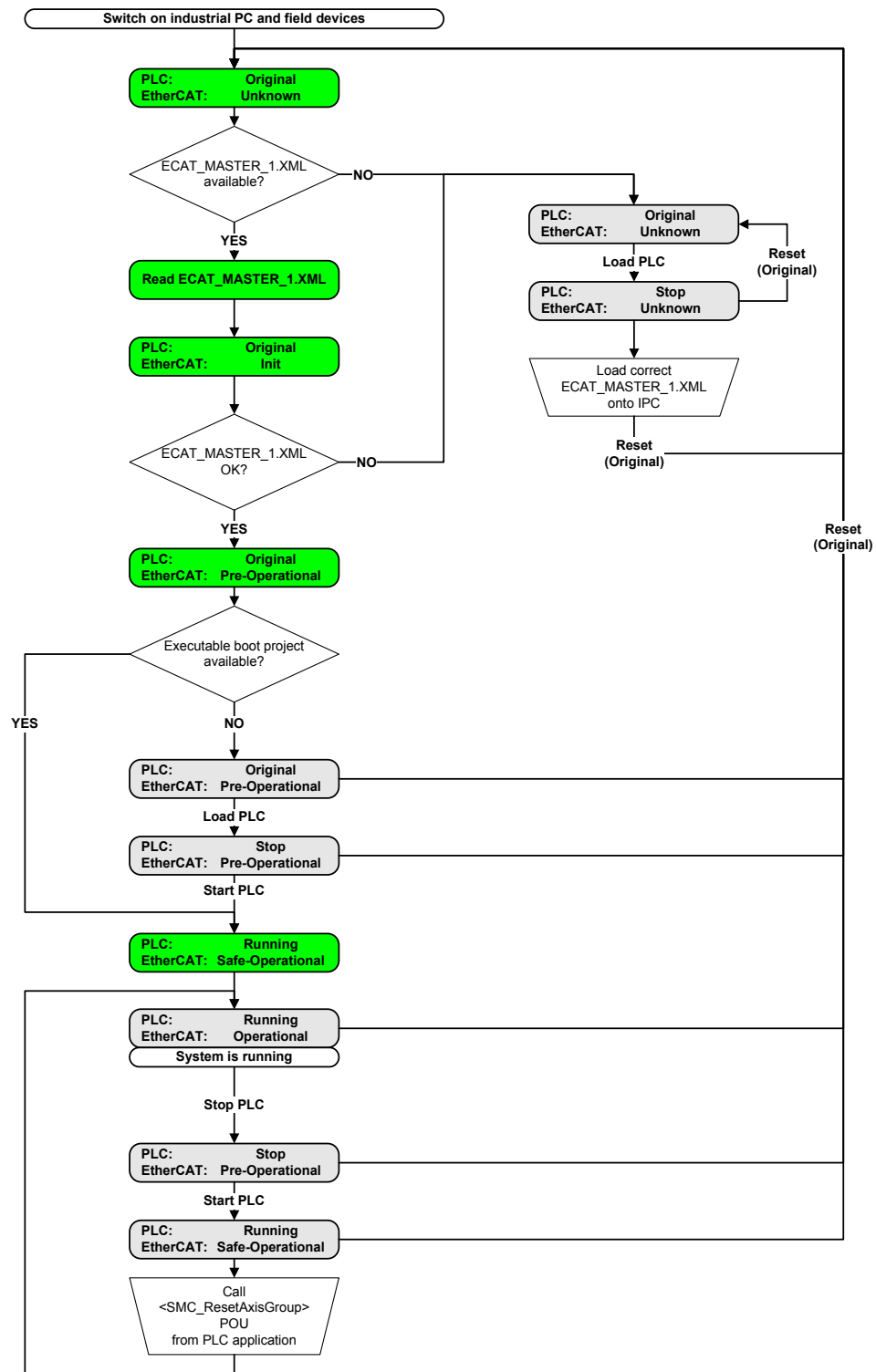
Example

Once per minute the PLC reads the error counters and evaluates the contents. If it recognises a error counter value of 250, the application will reset the error counters. Depending on the EMC load of the network environment, approx. 1 to 2 frame errors per day in the bus are normal. The error counters should not reach the value of 255.

11.6 Error scenarios

In the following sections, the causes and remedies for the most frequent user errors are described.

The state diagram and the table on the next page serve to localise an error.



Expected state changes	Current states		Error
	PLC	EtherCAT bus	
- PLC: Origin → RUN - Bus: Unknown → Init → Pre-Operational	Origin	Unknown	EtherCAT bus does not enter the Pre-Operational state (147)
- PLC: Origin → RUN - Bus: Pre-Operational → Safe-Operational → Operational	Origin	Pre-Operational	Compilation error in »PLC Designer« (147) Control unit/PLC does not enter the RUN state (147)
- PLC: Origin → RUN - Bus: Pre-Operational → Safe-Operational → Operational	RUN	Pre-Operational	EtherCAT bus does not enter the Operational state (148)
- PLC: Origin → RUN - Bus: Safe-Operational → Operational	RUN	Operational	Shafts make clicking noises (149) Shafts do not rotate (150) Messages in the logbook of the IPC: Error during EtherCAT data transmission (148) Logbook message: "Cannot spawn Remote API Server" (151) Logbook message: "Ethernet cable not connected" (152) Logbook message: "Ethernet cable connected" (153) Logbook messages: "Slave at index X missing" with "Cyclic command WKC error ..." (154) Logbook message: "Cyclic command WKC error ..." (156)

11.6.1 Compilation error in »PLC Designer«

Causes	Remedies
Logic system: No update task has been created for the IO_Group.	Create an update task.
SoftMotion system: No update task has been created for the Axis_IO_Group.	

11.6.2 EtherCAT bus does not enter the Pre-Operational state

During the start-up of the EtherCAT bus, a check is carried out at the transition *Init*→*Pre-Operational* to determine if the physical bus configuration corresponds to the configured bus configuration. If these configurations are different the master does not enter the *Pre-Operational* state.

Furthermore, the slaves are reset during the transition from initialization in *Pre-Operational*. If this fails because, for instance, a slave denies the configuration, the master does not enter the *Pre-Operational* state.

Causes	Remedies
The current master configuration has not been loaded on the control system or imported into the »PLC Designer« project (Axi-IO_Group or IO_Group are missing)	Load the current master configuration onto the control system or import it into the »PLC Designer« project.
Faulty master configuration • Slaves are missing. • Slaves have been interchanged or their configuration is faulty. • A wrong slave type has been configured.	Correct the master configuration with the »EtherCAT Configurator« and load it onto the control system or import it into the »PLC Designer« project.
Wiring error • Cabling is incomplete or there is a cable break. • Faulty cabling of the slave • The inputs and outputs of the EtherCAT communication module are reversed (IN/OUT connections).	Check the wiring and correct it. 1. Stopping the PLC program: Execute the menu command Online→Stop. 2. Plug in the bus cable. 3. Execute the SMC_ResetAxisGroup function block or a Bus restart (□ 139).

11.6.3 Control unit/PLC does not enter the RUN state

Causes	Remedies
Boot project is missing or faulty.	• Start the control unit/PLC manually. • Create a boot project. • Correct the boot project. Note: Only save a tested and consistent boot project to the control unit.
The task and DC-cycle times set in the logic/motion system differ.	Set identical task and DC cycle times. Note: Unlike loading a project and starting the PLC program via the »PLC Designer«, starting via a boot project also starts the PLC - even if there is a DC/task cycle difference!

11.6.4 EtherCAT bus does not enter the Operational state

The control unit/PLC transfers the EtherCAT bus to the *Operational* state if it set to the RUN mode.

The EtherCAT bus can only reach the *Operational* state if the bus has already been set to the *Pre-Operational* state.

Causes	Remedies
The current master configuration has not been loaded on the control system or imported into the »PLC Designer« project.	Load the current master configuration onto the control system or import it into the »PLC Designer« project.
The DC cycle time of the master configuration is not identical with the task cycle time of the Axis_IO_Group/IO_Group.	Set identical task and DC cycle times. Note: Unlike loading a project and starting the PLC program via the »PLC Designer«, starting via a boot project also starts the PLC - even if there is a DC/task cycle difference!

11.6.5 Error during EtherCAT data transmission

Causes	Remedies
The »PLC Designer« project does not access the EtherCAT inputs and outputs via symbolic addresses, but via logical addresses. This causes problems if the bus structure, the PDO selection, etc. have been changed.	Check and correct the mapping settings in the master configuration. ▶ Executing PDO mapping (□ 65)
For the Servo Drive 9400 HighLine, the ports are not mapped correctly or not mapped at all in the »Engineer«, and parameters have therefore not been transferred to the device.	During start-up of the PLC, the complete configuration/ PDO mapping is written into the EtherCAT slaves. Mapping entries, e.g. from the »Engineer«, are not overwritten.

11.6.6 Shafts make clicking noises

If the shafts make clicking noises, this is often caused by faulty synchronisation or a shift of data in the process image.

Causes	Remedies
The current master configuration has not been loaded on the control system or imported into the »PLC Designer« project.	Load the current master configuration onto the control system or import it into the »PLC Designer« project.
Faulty settings in master configuration	Carry out the following settings for the master configuration on the Slave device tab of the »EtherCAT Configurator«: • Select "DC". • Enable "Sync0". (Standard setting) • Set Sync Unit Cycle "x 1". (Standard setting)
The DC cycle time of the master configuration is not identical with the task cycle time of the Axis_IO_Group/IO_Group.	Set identical task and DC cycle times. Note: Unlike loading a project and starting the PLC program via the »PLC Designer«, starting via a boot project also starts the PLC - even if there is a DC/task cycle difference!
For a Servo Drive 9400 HighLine, the synchronisation source of the EtherCAT communication module has not been entered correctly in C01120. During the initialisation, the Softmotion Kernel sets C01120.	Correct the setting in C01120 (synchronisation source).
If the last 9400 SoftMotion drive makes clicking noises, the wiring of in- and outputs of the communication module (IN/OUT ports) may have been reversed. Note: A bus scan does not indicate this error!	Check the wiring and correct it. Especially check the wiring at the input and output of the EtherCAT communication module (IN/OUT ports).

11.6.7 Shafts do not rotate

Causes	Remedies
The EtherCAT bus could not be set to the <i>Operational</i> state	See: EtherCAT bus does not enter the Operational state (148)
The current master configuration has not been loaded on the control system or imported into the »PLC Designer« project.	Load the current master configuration onto the control system or import it into the »PLC Designer« project.
The process image on the EtherCAT bus does not comply with the one in the control.	Correct the current master configuration and load it onto the control system or import it into the »PLC Designer« project.
In the EtherCAT Configurator, the increments per revolution are not set for the SoftMotion scaling/mapping.	Check the following settings and correct them if required: • Gearbox ratio in the »PLC Designer« project • Mapping settings in the master configuration ▶ Executing PDO mapping (65) • During start-up of the PLC, the complete configuration/PDO mapping is written into the EtherCAT slaves. Mapping entries, e.g. from the »Engineer«, are not overwritten. • Tip: In case of the Servo Drive 9400 HighLine Cia402, 65536 increments per revolution are correct.
The »PLC Designer« project does not access the EtherCAT inputs and outputs via symbolic addresses, but via logical addresses. This causes problems if the bus structure, the PDO selection, etc. have been changed.	Check and correct the mapping settings in the master configuration. ▶ Executing PDO mapping (65) During start-up of the PLC, the complete configuration/PDO mapping is written into the EtherCAT slaves. Mapping entries, e.g. from the »Engineer«, are not overwritten.
For the Servo Drive 9400 HighLine, the ports are not mapped correctly or not mapped at all in the »Engineer«, and parameters have therefore not been transferred to the device.	

11.6.8 Logbook message: "Cannot spawn Remote API Server"

...

```
xxxxx DD-MM-YYYY hh:mm:ss   CTRL: Cannot spawn Remote API Server Communication on bus  
EtherCAT Master Stack       systems  
                             Error
```

...

The remote API server enables the access from an engineering tool (e.g. the »Engineer«) which superimposes the IPC to an EtherCAT slave parameter. The access is made via CoE (CANopen over EtherCAT).

Causes	Remedies
No master configuration or a master configuration not matching the physical bus has been loaded on the control system. SDO communication with the slave field devices is not possible.	• Create a master configuration with the »EtherCAT Configurator« and load it onto the control system. • Load the current master configuration onto the control system.

11.6.9 Logbook message: "Ethernet cable not connected"

...

```
xxxxx DD-MM-YYYY hh:mm:ss    LLA: Ethernet cable not connected
EtherCAT Master Stack        Communication on bus systems
                             Error
```

...

This error message is entered into the logbook when the error is detected for the first time and then cyclically every 5 seconds until 65535 messages have occurred or the error is not active anymore.

Causes	Remedies
The bus cable between the IPC and the first node has been unplugged.	1. Stopping the PLC program: Execute the menu command Online→Stop. 2. Plug in the bus cable. 3. Execute the SMC_ResetAxisGroup function block or a Bus restart (139). (For this, see also Logbook message: "Ethernet cable connected" (153).)

Error notifications in the »PLC Designer«

In the "ETHERCATMaster Status" visualisation template of the »PLC Designer«, the following error notifications are highlighted in red (fault):

ETHERCATMaster Status	
Instanz: \$FB\$	
eECATState: %s	
EC_NOTIFY_ETH_LINK_CONNECTED	
EC_NOTIFY_STATUS_SLAVE_ERROR	
EC_NOTIFY_RED_LINEBRK	
EC_NOTIFY_NOT_ALL_DEVICES_OPERATIONAL	
EC_NOTIFY_ETH_LINK_NOT_CONNECTED	
EC_NOTIFY_SB_STATUS	
EC_NOTIFY_DC_STATUS	
EC_NOTIFY_DC_SLV_SYNC	
EC_NOTIFY_DCL_STATUS	
EC_NOTIFY_DCL_SLV_LATCH_EVT	
EC_NOTIFY_FRAME_RESPONSE_ERROR	
EC_NOTIFY_CYCCMD_WKC_ERROR	
No of Slaves: %d	
COE Emergency	

11.6.10 Logbook message: "Ethernet cable connected"

...

```
xxxxx DD-MM-YYYY hh:mm:ss    LLA: Ethernet cable connected
EtherCAT Master Stack        Communication on bus systems
                             Error
```

...

When an unplugged bus cable is plugged in again into the first EtherCAT node, this message is entered into the logbook of the industrial PC. The EtherCAT connection has been re-established.

As the EtherCAT slave sync managers do not receive any messages, a time-out expires. This information is sent to the master.

Error notifications in the »PLC Designer«

In the "ETHERCATMaster Status" visualisation template of the »PLC Designer«, the following error notifications are highlighted in green (information) or red (fault):

ETHERCATMaster Status	
Instanz: \$FB\$	
eECATState: %s	
EC_NOTIFY_ETH_LINK_CONNECTED	
EC_NOTIFY_STATUS_SLAVE_ERROR	
EC_NOTIFY_RED_LINEBRK	
EC_NOTIFY_NOT_ALL_DEVICES_OPERATIONAL	
EC_NOTIFY_ETH_LINK_NOT_CONNECTED	
EC_NOTIFY_SB_STATUS	
EC_NOTIFY_DC_STATUS	
EC_NOTIFY_DC_SLV_SYNC	
EC_NOTIFY_DCL_STATUS	
EC_NOTIFY_DCL_SLV_LATCH_EVT	
EC_NOTIFY_FRAME_RESPONSE_ERROR	
EC_NOTIFY_CYCCMD_WKC_ERROR	
No of Slaves: %d	
COE Emergency	

11.6.11 Logbook messages: "Slave at index X missing" with "Cyclic command WKC error ..."

...

```
xxxxx DD-MM-YYYY hh:mm:ss    SLV: Cyclic command WKC error on LWR - Address: 0x10000 -
EtherCAT Master Stack         WKC act/set=2/7
                               Communication on bus systems
                               Error
```

...

```
xxxxx DD-MM-YYYY hh:mm:ss    SLV: Slave at index X missing. State req/act=Operational/
EtherCAT Master Stack         UNKNOWN
                               Communication on bus systems
                               Error
```

...

The "Cyclic command WKC error" message is entered cyclically every 5 seconds into the logbook of the industrial PC until the error is not active anymore.

Causes	Remedies
The bus cable between two EtherCAT nodes has been unplugged.	1. Stopping the PLC program: Execute the menu command Online→Stop. 2. Plug in the bus cable. 3. Execute the SMC_ResetAxisGroup function block or a Bus restart (□ 139). (For this, see also Logbook message: "Cyclic command WKC error ..." (□ 156).)
The node at position X is deenergised.	Switch-on the node and execute the SMC_ResetAxisGroup function block or a Bus restart (□ 139).

Error notifications in the »PLC Designer«

In the "ETHERCATMaster Status" visualisation template of the »PLC Designer«, the following error notifications are highlighted in red (fault):

ETHERCATMaster Status	
Instanz: \$FB\$	
eECATState: %s	
EC_NOTIFY_ETH_LINK_CONNECTED	
EC_NOTIFY_STATUS_SLAVE_ERROR	
EC_NOTIFY_RED_LINEBRK	
EC_NOTIFY_NOT_ALL_DEVICES_OPERATIONAL	
EC_NOTIFY_ETH_LINK_NOT_CONNECTED	
EC_NOTIFY_SB_STATUS	
EC_NOTIFY_DC_STATUS	
EC_NOTIFY_DC_SLV_SYNC	
EC_NOTIFY_DCL_STATUS	
EC_NOTIFY_DCL_SLV_LATCH_EVT	
EC_NOTIFY_FRAME_RESPONSE_ERROR	
EC_NOTIFY_CYCCMD_WKC_ERROR	
No of Slaves: %d	
COE Emergency	

11.6.12 Logbook message: "Cyclic command WKC error ..."

...

```
xxxxx DD-MM-YYYY hh:mm:ss    SLV: Cyclic command WKC error on LWR - Address: 0x10000 -
EtherCAT Master Stack        WKC act/set=2/7
                             Communication on bus systems
                             Error
```

...

The "Cyclic command WKC error" message is entered cyclically every 5 seconds into the logbook of the industrial PC until the error is not active anymore.

Causes	Remedies
The EtherCAT slave sync managers do not receive any messages and a time-out expires. Example: An unplugged bus cable between two EtherCAT nodes has been replugged. Afterwards, however, neither the SMC_ResetAxisGroup function block nor a Bus restart (139) have been executed.	1. Stopping the PLC program: Execute the menu command Online→Stop. 2. Plug in the bus cable. 3. Execute the SMC_ResetAxisGroup function block or a Bus restart (139).

Error notifications in the »PLC Designer«

In the "ETHERCATMaster Status" visualisation template of the »PLC Designer«, the following error notifications are highlighted in red (fault):

ETHERCATMaster Status
Instanz: \$FB\$
eECATState: %s
EC_NOTIFY_ETH_LINK_CONNECTED
EC_NOTIFY_STATUS_SLAVE_ERROR
EC_NOTIFY_RED_LINEBRK
EC_NOTIFY_NOT_ALL_DEVICES_OPERATIONAL
EC_NOTIFY_ETH_LINK_NOT_CONNECTED
EC_NOTIFY_SB_STATUS
EC_NOTIFY_DC_STATUS
EC_NOTIFY_DC_SLV_SYNC
EC_NOTIFY_DCL_STATUS
EC_NOTIFY_DCL_SLV_LATCH_EVT
EC_NOTIFY_FRAME_RESPONSE_ERROR
EC_NOTIFY_CYCCMD_WKC_ERROR
No of Slaves: %d
COE Emergency

11.7 System error messages

For system error messages, the following error types are distinguished:

Error type	Abbr.	Description
Application error	CTRL	Internal error in the application operating the master. • E.g. an API function has been called with invalid parameters.
Project error	PRJ	The XML file of the master does not correspond to the slaves. • E.g. not all slaves included in the XML file have been detected during the bus scan.
Configuration error (master)	CFG	Faulty or insufficient configuration of the master. • E.g. mailbox command queue too small.
Bus/slave error	SLV	Error cause by slave • E.g. "Working Counter Error"
Link layer error	LLA	Error in the link layer (network driver). • E.g. the Intel Pro 1000 card could not be found. • These are mostly internal errors.
Remote API error	RAP	Error in remote API (The applications and engineering tools »EtherCAT OPC server«, »EtherCAT Configurator«, »Engineer« use the remote API.) • E.g. a connection between slave and master is not possible.
Internal software error	ISW	Internal master error • E.g. the master state machine is in an invalid state.

11.7.1 IPC logbook messages

IPC logbook messages are indicated as errors, warnings, or information.



Note!

The WKC error messages (highlighted in blue in the table) are entered into the logbook when the error is detected for the first time and then cyclically every 5 seconds until 65535 messages have occurred or the error is not active anymore.

In the following table, "[...]" serves as a wildcard for addresses, indexes, message types, error numbers, etc.

Error type	Error text in the IPC logbook	Description
CTRL	Master state change from [...] to [...]	State change of the master from [...] to [...]
CTRL	Cyclic command time-out: Time between sending cyclic commands too high	Time-out of cyclic commands: The time between the transmission processes is too long.
LLA	Retry sending a [...] frame due to [...]	Retry to send a data frame
LLA	[...] response on [...] Ethernet frame	Response to an Ethernet frame
LLA	Ethernet cable connected	Ethernet cable is connected.
LLA	Ethernet cable not connected	Ethernet cable is not connected.
PRJ	Cyclic command wrong size (too long)	The process image seize in the EtherCAT master configuration (ECAT_MASTER_1.XML) is too large (Example: 35 MB).
PRJ	Invalid input offset in cyc cmd, please check InputOffs	
PRJ	Invalid output offset in cyc cmd, please check OutputOffs	
SLV	Scan Bus Succeeded, found [...] slaves	Bus scan has been completed successfully. [...] slaves were found.
SLV	Scan Bus Error [...], found [...] slaves	Bus scan error [...] [...] slaves were found.
SLV	Distributed Clocks status [...] ([...])	Distributed clocks status

Error type	Error text in the IPC logbook	Description
SLV	Distributed Clocks latching status [...] ([...])	Distributed clocks - latching status
SLV	CoE: SDO Download failure, statVal=[...], errCode=[...] ([...])	CoE: SDO download has failed.
SLV	CoE: SDO Upload failure, statVal=[...], errCode=[...] ([...])	CoE: SDO upload has failed.
SLV	CoE: OD-List Upload failure, statVal=[...], errCode=[...] ([...])	CoE: OD list upload has failed.
SLV	CoE: Object Description Upload failure, statVal=[...], errCode=[...] ([...])	CoE: Object description upload has failed.
SLV	CoE: Object Entry Description Upload failure, statVal=[...], errCode=[...] ([...])	CoE: Object entry description upload has failed.
SLV	CoE: Emergency transfer failure, statVal=[...], errCode=[...] ([...])	CoE: Emergency transfer has failed
SLV	CoE: Emergency request, id=[...], len=[...] ==> slave address=[...], ErrCode=[...], ErrReg=[...], data: '[...] [...] [...] [...] [...]']	CoE: Emergency request to slave [...]
SLV	Cyclic command working counter error - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	Cyclic command - WKC error at the node with the logic/physical address [...]
SLV	Master init command working counter error - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	Master init command - WKC error at the node with the logic/physical address [...]
SLV	Slave init command working counter error - Properties for slave \"[...]\" - EtherCAT address=[...] - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	Slave init command - WKC error at the node with the logic/physical address [...]
SLV	EoE receive working counter error - Properties for slave \"[...]\" - EtherCAT address=[...] - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	EoE receive - WKC error at the slave with the logic/physical address [...]
SLV	CoE receive working counter error - Properties for slave \"%1!hs! - EtherCAT address=%2!lu! - Command: %3!hs! - Logical/Physical address: 0x%4!xl, WKC act/set=[...]/[...]	CoE receive - WKC error at the slave with the logic/physical address [...]
SLV	FoE receive working counter error - Properties for slave \"[...]\" - EtherCAT address=[...] - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	FoE receive - WKC error at the slave with the logic/physical address [...]
SLV	SoE receive working counter error - Properties for slave \"[...]\" - EtherCAT address=[...] - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	SoE receive - WKC error at the slave with the logic/physical address [...]
SLV	EoE send working counter error - Properties for slave \"[...]\" - EtherCAT address=[...] - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	EoE send - WKC error at the slave with the logical/physical address [...]
SLV	CoE send working counter error - Properties for slave \"[...]\" - EtherCAT address=[...] - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	CoE send - WKC error at the slave with the logical/physical address [...]
SLV	FoE send working counter error - Properties for slave \"[...]\" - EtherCAT address=[...] - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	FoE send - WKC error at the slave with the logical/physical address [...]
SLV	SoE send working counter error - Properties for slave \"[...]\" - EtherCAT address=[...] - Command: [...] - Logical/Physical address: [...], WKC act/set=[...]/[...]	SoE send - WKC error at the slave with the logical/physical address [...]
SLV	Additional error information: [...]	Additional error information [...]
SLV	Ecat command IDX act value=[...]	EtherCAT command - IDX actual value
SLV	Ecat command IDX set value=[...]	EtherCAT command - IDX setpoint
SLV	Init command response error – Properties for slave \"[...]\" - EtherCAT address=[...] – Current State change of slave= \"[...]\" No Response, is there a slave at this position?	Response error to Init command to slave [...]: No response: Is there a slave at this position?
SLV	Init command response error – Properties for slave \"[...]\" - EtherCAT address=[...] – Current State change of slave= \"[...]\" Validation error, is the correct slave at this position?	Response error to Init command to slave [...]: Validation error: Is the correct slave at this position?
SLV	Init command response error – Properties for slave \"[...]\" - EtherCAT address=[...] – Current State change of slave= \"[...]\" target state could not be reached, is the correct slave at this position?	Response error to Init command to slave [...]: Target state has not been reached. Is the correct slave at this position?
SLV	Master init command response error – Current State change of master= \"[...]\" No Response, is there anything connected?	Response error to master init command: State change of the master No response: Is there anything connected at all?
SLV	Master init command response error – Current State change of master= \"[...]\" Validation error, are the correct slaves connected?	Response error to master init command: State change of the master Validation error, are the correct slaves connected?
SLV	Missing EtherCAT [...] command in Ethernet frame – Properties for slave \"[...]\" - EtherCAT address=[...] – Index of missing command in the Ethernet frame=[...]	EtherCAT command [...] is missing in the Ethernet frame to slave [...].
SLV	Mailbox Slave init command timeout – Properties for slave \"[...]\" - EtherCAT address=[...] – Current State change of mailbox slave= \"[...]\"	Time-out "Mailbox init command" to slave [...]

Error type	Error text in the IPC logbook	Description
SLV	Not all EtherCAT slave devices are in Operational state	Some EtherCAT slaves are not in Operational state
SLV	Redundancy: Line Break	Redundant operation: Bus/communication interruption (Ethernet cable is not connected.)
SLV	At least one slave signals error status!	At least one slave has an active error.
SLV	Slave error status info – Properties for slave \"[...]\": - EtherCAT address=[...] – slave status <[...]>([...]), control status <[...]>([...])	Error status information of the slave [...]
SLV	Slave [...] not Addressable - Properties for slave \"[...]\": - EtherCAT address=[...] -	The slave [...] cannot be addressed.
SLV	SDO Abort - Properties for slave \"[...]\": - EtherCAT address=[...] - [...] ([...]) - Index=[...] SubIndex=[...]	SDO abort at slave [...]
SLV	DC Slaves In-Sync. Deviation : [...] nanosec	DC slaves are "in-sync": DC deviation: [...] nanoseconds
SLV	DC Slaves Out-Of-Sync. Deviation : [...] nanosec	DC slaves are "out-of-sync": DC deviation: [...] nanoseconds
SLV	DC Single Latch on Slave [...] Ident [...]	DC Single Latch: Transfer of the I/O data to slave [...]

11.7.2 General error codes (0x00000000_{hex}, 0x98110001 ... 0x98110038_{hex})

- General error codes are provided at the *dwErrorCode* output of the EtherCAT function blocks (see chapter "[EtherCAT function libraries](#)" (96)).
- In the [Logbook of the IPC](#) (140) the error messages are output as additional error information:

...

```
00007 14-06-2008 10:48:11    CTRL: Cannot set EtherCAT-Master to init! (Errorcode =
EtherCAT Master Stack      0x98110010), retry...
                           Communication on bus systems
                           Error
```

...

Error number [hex] Designation	Error type	Description
0x00000000 EC_E_NOERROR	-	Function completed successfully.
0x98110001 EC_E_NOTSUPPORTED	CTRL	Function or feature not available.
0x98110002 EC_E_INVALIDINDEX	CTRL	CoE: invalid SDO index
0x98110003 EC_E_INVALIDOFFSET	ISW	Invalid offset value during access to process data image
0x98110005 EC_E_INVALIDSIZE	CTRL	Invalid offset value - during access to process data image - during storage of data in a data area
0x98110006 EC_E_INVALIDDATA	ISW	Invalid data
0x98110007 EC_E_NOTREADY	ISW	Internal software error (numerous possible causes)
0x98110008 EC_E_BUSY	CTRL	The master is busy at the moment and cannot process the API function. The function should be repeated at a later time.
0x98110009 EC_E_ACYC_FRM_FREEQ_EMPTY	ISW	The queue for acyclic commands is full.
0x9811000A EC_E_NOMEMORY	CFG	Not enough application memory available.

Error number [hex] Designation	Error type	Description
0x9811000B EC_E_INVALIDPARM	CTRL	An API function has been called with incorrect parameters.
0x9811000C EC_E_NOTFOUND	CTRL	An API function has been called with an invalid slave ID.
0x9811000E EC_E_INVALIDSTATE	ISW	Invalid state
0x9811000F EC_E_TIMER_LIST_FULL	ISW	Not enough application memory available.
0x98110010 EC_E_TIMEOUT	All	A time-out is active.
0x98110011 EC_E_OPENFAILED	ISW	Internal software error (numerous possible causes)
0x98110012 EC_E_SENDFAILED	LLA	The transmission of the frame has failed.
0x98110013 EC_E_INSERTMAILBOX	CFG	The mailbox command cannot be enqueued on the internal queue.
0x98110014 EC_E_INVALIDDCMD	ISW	Unknown mailbox command code
0x98110015 EC_E_UNKNOWN_MBX_PROTOCOL	ISW	Unknown mailbox protocol Mailbox command ID with unknown protocol assignment
0x98110016 EC_E_ACCESSDENIED	ISW	Access denied (internal software error on the master)
0x9811001A EC_E_PRODKEY_INVALID	CFG	The evaluation version of the master is used. The master enters the "Stop" mode after 30 minutes.
0x9811001B EC_E_WRONG_FORMAT	PRJ	The XML file contains no or faulty content.
0x9811001C EC_E_FEATURE_DISABLED	CTRL	Attempt to execute a non-existing or deactivated function.
0x9811001E EC_E_BUSCONFIG_MISMATCH	PRJ	The bus configuration of the master and the connected slaves in the XML file does not comply with the physical bus structure.
0x9811001F EC_E_CONFIGDATAREAD	PRJ	The XML file cannot be read.
0x98110021 EC_E_XML_CYCCMDS_MISSING	PRJ	The XML file of the master does not contain any cyclic commands.
0x98110022 EC_E_XML_ALSTATUS_READ_MISSING	PRJ	The XML file of the master does not contain the command for reading the AL Status Register.
0x98110023 EC_E_MCSM_FATAL_ERROR	ISW	The master state machine is in an invalid state.
0x98110024 EC_E_SLAVE_ERROR	SLV	The slave cannot be addressed.
0x98110025 EC_E_FRAME_LOST	SLV	An EtherCAT frame was lost on the bus, i.e. it has not been received. If this error occurred frequently, this indicates the wiring may be faulty.
0x98110026 EC_E_CMD_MISSING	SLV	The received EtherCAT frame is not complete.
0x98110028 EC_E_INVALID_DCL_MODE	CTRL	This function cannot be used when DC latching is in the "Auto Read" operating mode.
0x98110029 EC_E_AI_ADDRESS	SLV	The bus configuration (XML file of the master) does not correspond with the connected slaves. This error only occurs if a previously existing slave disappears.
0x9811002A EC_E_INVALID_SLAVE_STATE	CTRL	The mailbox commands are not permissible in the current slave state.
0x9811002B EC_E_SLAVE_NOT_ADDRESSABLE	SLV	The slave has been switched on/off.
0x9811002C EC_E_CYC_CMDS_OVERFLOW	PRJ	Error during XML file creation by the configurator
0x9811002D EC_E_LINK_DISCONNECTED	SLV	The EtherCAT cable is not connected to the control unit/is not plugged in.
0x9811002E EC_E_MASTERCORE_INACCESSIBLE	RAP	The connection to the master (server) is interrupted or the master has been stopped.

Error number [hex] Designation	Error type	Description
0x9811002F EC_E_COE_MBXSEND_WKC_ERROR	SLV	It is not possible to write to the CoE mailbox in the slave. The slave has not yet read the mailbox.
0x98110030 EC_E_COE_MBXRCV_WKC_Error	SLV	It is not possible to read the CoE mailbox in the slave.
0x98110031 EC_E_NO_MBX_SUPPORT	CTRL	The slave does not support mailbox transfer.
0x98110032 EC_E_NO_COE_SUPPORT	PRJ	Configurator error or the slave description file does not correspond with the slave firmware.
0x98110033 EC_E_NO_EOE_SUPPORT	PRJ	Configurator error or the slave description file does not correspond with the slave firmware.
0x98110034 EC_E_NO_FOE_SUPPORT	PRJ	Configurator error or the slave description file does not correspond with the slave firmware.
0x98110035 EC_E_NO_SOE_SUPPORT	PRJ	Configurator error or the slave description file does not correspond with the slave firmware.
0x98110036 EC_E_NO_VOE_SUPPORT	PRJ	Configurator error or the slave description file does not correspond with the slave firmware.
0x98110037 EC_E_EVAL_VIOLATION	PRJ	The number of slaves indicated in the XML file is too large for the evaluation version of the master.
0x98110038 EC_E_EVAL_EXPIRED	CFG	The evaluation time has expired. The bus is stopped.

11.7.3 CANOpen over EtherCAT (CoE) SDO error codes (0x98110040 ... 0x9811005D_{hex})

The CoE SDO error codes are relevant for:

- The function blocks `ecatSdoUploadReq` and `ecatSdoDownloadReq`.
- SDO read/write error messages caused by SDO requests from the system (e.g. initialisation code or SDO requests from the engineering tool).

The CoE SDO error codes present the [SDO abort codes](#) (164).

Error number [hex] Designation	Error type	Description
0x98110040 EC_E_SDO_ABORTCODE_TOGGLE	SLV	The status of the toggle bit has not changed. Abort code 0x05030000
0x98110041 EC_E_SDO_ABORTCODE_TIMEOUT	SLV	SDO protocol time-out Abort code 0x05040000
0x98110042 EC_E_SDO_ABORTCODE_CCS_SCS	SLV	Invalid or unknown specification symbol for the client/server command Abort code 0x05040001
0x98110043 EC_E_SDO_ABORTCODE_BLK_SIZE	SLV	Invalid block size (only in "Block mode") Abort code 0x05040002
0x98110044 EC_E_SDO_ABORTCODE_SEQNO	SLV	Invalid sequence number (only in "Block mode") Abort code 0x05040003
0x98110045 EC_E_SDO_ABORTCODE_CRC	SLV	CRC error (only in "Block mode") Abort code 0x05040004
0x98110046 EC_E_SDO_ABORTCODE_MEMORY	SLV	There is not sufficient space in the main memory. Abort code 0x05040005
0x98110047 EC_E_SDO_ABORTCODE_ACCESS	SLV	Access to object not supported Abort code 0x06010000
0x98110048 EC_E_SDO_ABORTCODE_WRITEONLY	SLV	Read access to a write-protected object Abort code 0x06010001

Error number [hex] Designation	Error type	Description
0x98110049 EC_E_SDO_ABORTCODE_ READONLY	SLV	Write access to a write-protected object Abort code 0x06010002
0x9811004A EC_E_SDO_ABORTCODE_ INDEX	SLV	Object is not listed in the object directory. Abort code 0x06020000
0x9811004B EC_E_SDO_ABORTCODE_ PDO_MAP	SLV	Object cannot be mapped into the PDO. Abort code 0x06040041
0x9811004C EC_E_SDO_ABORTCODE_ PDO_LEN	SLV	The number and/or length of the mapped objects would exceed the PDO length. Abort code 0x06040042
0x9811004D EC_E_SDO_ABORTCODE_P_ INCOMP	SLV	General parameter incompatibility Abort code 0x06040043
0x9811004E EC_E_SDO_ABORTCODE_I_ INCOMP	SLV	General internal device incompatibility Abort code 0x06040047
0x9811004F EC_E_SDO_ABORTCODE_ HARDWARE	SLV	Access has failed because of hardware errors. Abort code 0x06060000
0x98110050 EC_E_SDO_ABORTCODE_ DATA_SIZE	SLV	Wrong data type or parameter length. Abort code 0x06070010
0x98110051 EC_E_SDO_ABORTCODE_ DATA_SIZE1	SLV	Wrong data type (parameter length is too large). Abort code 0x06070012
0x98110052 EC_E_SDO_ABORTCODE_ DATA_SIZE2	SLV	Wrong data type (parameter length is too small). Abort code 0x06070013
0x98110053 EC_E_SDO_ABORTCODE_ OFFSET	SLV	Subindex does not exist. Abort code 0x06090011
0x98110054 EC_E_SDO_ABORTCODE_ DATA_RANGE	SLV	The value range for parameters is too large (only for write access). Abort code 0x06090030
0x98110055 EC_E_SDO_ABORTCODE_ DATA_RANGE1	SLV	The parameter value is too high. Abort code 0x06090031
0x98110056 EC_E_SDO_ABORTCODE_ DATA_RANGE2	SLV	The parameter value is too low. Abort code 0x06090032
0x98110057 EC_E_SDO_ABORTCODE_ MINMAX	SLV	The maximum value is smaller than the minimum value. Abort code 0x06090036
0x98110058 EC_E_SDO_ABORTCODE_ GENERAL	SLV	General error Abort code 0x08000000
0x98110059 EC_E_SDO_ABORTCODE_ TRANSFER	SLV	Data cannot be transferred/saved to the application. Abort code 0x08000020
0x9811005A EC_E_SDO_ABORTCODE_ TRANSFER1	SLV	Data cannot be transferred/saved to the application because of local control. Abort code 0x08000021
0x9811005B EC_E_SDO_ABORTCODE_ TRANSFER2	SLV	Data cannot be transferred/saved to the application because of current device state. Abort code 0x08000022
0x9811005C EC_E_SDO_ABORTCODE_ DICTIONARY	SLV	Dynamic object directory generation has failed or no object directory available. Abort code 0x08000023
0x9811005D EC_E_SDO_ABORTCODE_ UNKNOWN	SLV	Unknown internal slave error

11.7.4 Remote API error codes (0x98110181 ... 0x98110196_{hex})

Remote API error codes are software-internal errors when the EtherCAT master stack is accessed. The applications and engineering tools »EtherCAT OPC server«, »EtherCAT Configurator«, »Engineer« use the remote API.

Error number [hex] Designation	Error type	Description
0x98110181 EMRAS_E_INVALIDCOOKIE	RAP	Reconnecting using the old cookie has failed. A new reconnection attempt is made automatically.
0x98110182 EMRAS_E_WDEXPIRED	RAP	Connection abort because of missing keep-alive messages. (Server or client frozen/does not respond anymore.)
0x98110183 EMRAS_E_MULSRVDISMULCON	RAP	Attempt to connect to another remote server has been rejected because the multi-instance API has not been used for establishing an already existing connection.
0x98110184 EMRAS_E_LOGONCANCELLED	RAP	Server aborts connection during client logon.
0x98110186 EMRAS_E_INVALIDVERSION	RAP	Server and client version are not identical (different protocol versions). Therefore, connecting has been rejected.
0x98110191 EMRAS_EVT_SERVERSTOPPED	RAP	Detailed description for connection abort/termination if connection to server has been closed due to "API call (local)".
0x98110192 EMRAS_EVT_WDEXPIRED	RAP	Detailed description for connection abort/termination if connection has been closed due to missing keep-alive messages.
0x98110193 EMRAS_EVT_RECONEXPIRED	RAP	Client attempts to reopen an old connection (after the connection has been aborted), but the server has already cleared the session. A new connection must be established (register client and mailbox objects must be created again)
0x98110194 EMRAS_EVT_CLIENTLOGON	RAP	Server message when a new client has connected.
0x98110195 EMRAS_EVT_RECONNECT	RAP	Server message when a client has successfully reopened an old connection.
0x98110196 EMRAS_EVT_SOCKCHANGE	RAP	Detailed description (event) which marks the successful socket transfer of a new connection to an already existing session object (reconnect).

11.8 SDO abort codes

The abort codes are relevant for:

- ▶ The function blocks **ecatSdoUploadReq** and **ecatSdoDownloadReq**.
- ▶ SDO read/write error messages caused by SDO requests from the system (e.g. initialisation code or SDO requests from the engineering tool).

Error number [hex]	Description
0x05030000	The status of the toggle bit has not changed.
0x05040000	SDO protocol time-out
0x05040001	Invalid or unknown specification symbol for the client/server command
0x05040002	Invalid block size (only in "Block mode")
0x05040003	Invalid sequence number (only in "Block mode")
0x05040004	CRC error (only in "Block mode")
0x05040005	There is not sufficient space in the main memory.
0x06010000	Access to object not supported
0x06010001	Read access to a write-protected object
0x06010002	Write access to a write-protected object
0x06020000	Object is not listed in the object directory.
0x06040041	Object cannot be mapped into the PDO.
0x06040042	The number and/or length of the mapped objects would exceed the PDO length.
0x06040043	General parameter incompatibility
0x06040047	General internal device incompatibility
0x06060000	Access has failed because of hardware errors.
0x06070010	Wrong data type or parameter length.
0x06070012	Wrong data type (parameter length is too large).
0x06070013	Wrong data type (parameter length is too small).
0x06090011	Subindex does not exist.
0x06090030	The value range for parameters is too large (only for write access).
0x06090031	The parameter value is too high.
0x06090032	The parameter value is too low.
0x06090036	The maximum value is smaller than the minimum value.
0x08000000	General error
0x08000020	Data cannot be transferred/saved to the application.
0x08000021	Data cannot be transferred/saved to the application because of local control.
0x08000022	Data cannot be transferred/saved to the application because of current device state.
0x08000023	Dynamic object directory generation has failed or no object directory available.

12 Parameter reference

This chapter complements the parameter list of the online documentation for the industrial PC by the parameters of the **Communication card MC-ETC**:

- ▶ [Parameters of the MC-ETC communication card in slot 1](#) (📖 166)
- ▶ [Interface parameters of the MC-ETC communication card in slot 1](#) (📖 167)
- ▶ [Parameters of the MC-ETC communication card in slot 2](#) (📖 182)
- ▶ [Interface parameters of the MC-ETC communication card in slot 2](#) (📖 183)



Note!

- Only one MC-ETC communication card can be installed per industrial PC. The »WebConfig« designation of the card is *MC-ETC1*, independently of the slot in which the communication card has been inserted.
- Depending on the slot used, the code numbers differ by an offset of '500'. To obtain the code numbers for a communication card inserted in slot 2, an offset of '500' must be added to the code numbers of a card inserted in slot 1.



Tip!

For general information about parameters please see the online documentation for the industrial PC.

12.1 Parameters of the MC-ETC communication card in slot 1

The parameters are listed in numerically ascending order.

C1031

Parameter Name:	Data type: VISIBLE_STRING
C1031 Device: type key	Index: 23544 _d = 5BF8 _h
Identification of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1032

Parameter Name:	Data type: VISIBLE_STRING
C1032 Device: type version	Index: 23543 _d = 5BF7 _h
Version number of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1033

Parameter Name:	Data type: VISIBLE_STRING
C1033 Device: name	Index: 23542 _d = 5BF6 _h
Device name of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1034

Parameter Name:	Data type: VISIBLE_STRING
C1034 Device: software revision	Index: 23541 _d = 5BF5 _h
Software version of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1035

Parameter Name:	Data type: VISIBLE_STRING
C1035 Device: hardware revision	Index: 23540 _d = 5BF4 _h
Hardware version of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1036

Parameter Name:	Data type: VISIBLE_STRING
C1036 Device: serial number	Index: 23539 _d = 5BF3 _h
Serial number of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1037

Parameter Name:	Data type: VISIBLE_STRING
C1037 Device: manufacturer	Index: 23538 _d = 5BF2 _h
Manufacturer of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1038

Parameter Name:	Data type: VISIBLE_STRING
C1038 Device: manufacturing date	Index: 23537 _d = 5BF1 _h
Manufacturing date of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

12.2 Interface parameters of the MC-ETC communication card in slot 1

The parameters are listed in numerically ascending order.

C1074

Parameter Name:	Data type: VISIBLE_STRING Index: 23501 _d = 5BCD _h
C1074 ECAT MAC address	
MAC address of the interface	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1080/1

Parameter Name:	Data type: DATE Index: 23495.1 _d = 0x5BC7.0x01 _h
C1080/1 ECAT Master Configuration Date	
Date of the master configuration file in the file system of the IPC The file manager determines the date of the master configuration file in the index ...\\storage\\IPC\\PLC. If this file does not exist, the date "01/01/1980 00:00:00" is displayed.	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1080/2

Parameter Name:	Data type: UNSIGNED_32 Index: 23495.2 _d = 0x5BC7.0x02 _h
C1080/2 ECAT Master Config Checksum	
Checksum of the master configuration file in the file system of the IPC The data manager determines the checksum of the master configuration file in the ...\\storage\\IPC\\PLC directory. If this file does not exist, the value "0x00000000" is displayed.	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1080/3

Parameter Name:	Data type: UNSIGNED_32 Index: 23495.3 _d = 0x5BC7.0x03 _h
C1080/3 ECAT Stack Master Checksum	
The stack determines the checksum of the loaded/active master configuration file. If there is no loaded/active master configuration file, the value "0x00000000" is displayed.	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1080/4

Parameter Name:	Data type: UNSIGNED_8 Index: 23495.4 _d = 0x5BC7.0x04 _h
C1080/4 ECAT Bus Scan Match	
Brief information whether the master configuration corresponds to the physical bus structure. The master configuration of the stack is compared with the actual bus structure.	
Selection list (Lenze setting printed in bold)	
0	No match
1	OK
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

Information

The master configuration does not correspond to the bus structure.

The master configuration corresponds to the bus structure.

C1081/1

Parameter | Name:

C1081/1 | ECAT State Change Command

Data type: UNSIGNED_32

Index: 23494.1_d =

0x5BC6.0x01_h

This code can be used to change the master state.

Note: The parameter should only be set to analyse start problems (e.g. master/bus does not reach the Operational state). Setting this parameter only sets the state of the master stack. All other nodes remain in their previous states. Changing the master state via this code may result in an unstable system state.

- A higher bus state than the one initiated by the PLC can never be reached.
- A reload of the master configuration cannot be initiated.
- Loading the PLC program normally stops the bus and sets it to the *Pre-Operational* state. If this is not possible, the bus remains in the Unknown or Initialisation state.
- Starting the PLC program normally sets the bus to the *Operational* state. If this is not possible, the bus remains in the *Pre-Operational* or *Safe Operational* state.

Selection list (Lenze setting printed in bold)

0	No Operation
1	Init
2	Pre-Operational
4	Safe-Operational
8	Operational

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer

C1081/2

Parameter | Name:

C1081/2 | ECAT Master State

Data type: UNSIGNED_8

Index: 23494.2_d =

0x5BC6.0x02_h

Display of the current master state

Selection list(read only)

0	Unknown
1	Init
2	Pre-Operational
3	Bootstrap Mode
4	Safe-Operational
8	Operational

Bootstrap Mode is not supported.

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1081/3

Parameter | Name:

C1081/3 | ECAT Master in requested Mode

Data type: UNSIGNED_8

Index: 23494.3_d =

0x5BC6.0x03_h

The state is usually requested by the PLC. For possible requested states see [C1080/1](#) (168).

- Value "0": Master is not in the requested state.
- Value "1": Master is in the requested state.

Display range (min. value | unit | max. value)

0		1
---	--	---

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1081/4

Parameter | Name:

C1081/4 | ECAT Slaves in requested Mode

Data type: UNSIGNED_8

Index: 23494.4_d =0x5BC6.0x04_h

- Value "0": One or more slaves are not in the state requested by the master.
- Value "1": All slaves are in the state requested by the master.

Display range (min. value | unit | max. value)

0

1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer**C1081/5**

Parameter | Name:

C1081/5 | ECAT Master State Summary

Data type: UNSIGNED_32

Index: 23494.5_d =0x5BC6.0x05_h

Display of additional information about the current master state

The bits are set to the value 1 when the respective states are reached.

Value is bit-coded:

Bit 0	Master ok
Bit 1	Reserved 1
Bit 2	Reserved 2
Bit 3	Reserved 3
Bit 4	Init
Bit 5	Pre-Operational
Bit 6	Safe Operational
Bit 7	Operational
Bit 8	Slaves in Requested State
Bit 9	Master in Requested State
Bit 10	Bus Scan Match
Bit 11	Reserved 4
Bit 12	DC: Activated
Bit 13	DC: Synchronised
Bit 14	DC: Busy
Bit 15	Reserved 5
Bit 16	Link Up
Bit 17	Reserved 6
...	...
Bit 31	Reserved 20

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer**C1081/6**

Parameter | Name:

C1081/6 | ECAT BusScan

Data type: UNSIGNED_8

Index: 23494.6_d =0x5BC6.0x06_h

Activation of the bus scan

The bus scan updates all EtherCAT codes.

Selection list (Lenze setting printed in bold)

0	No Operation
1	Scan Bus

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer

C1082/1

Parameter Name:	Data type: BITFIELD_32
C1082/1 ECAT Distributed Clocks State	Index: 23493.1 _d = 0x5BC5.0x01 _h

Display of the state of the distributed clocks
The bits are set to the value 1 when the respective states are reached.

Value is bit-coded:		
Bit 0	DC: Activated	
Bit 1	DC: Synchronised	
Bit 2	DC: Busy	

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1082/2

Parameter Name:	Data type: UNSIGNED_32
C1082/2 ECAT DC Slave Sync Deviation Limit	Index: 23493.2 _d = 0x5BC5.0x02 _h

Permitted deviation of the distributed clocks of all devices in nanoseconds. If the permitted deviation is exceeded, the master initiates a resynchronisation of the distributed clocks.

Display range (min. value unit max. value)		
0	ns	4294967295

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1082/3

Parameter Name:	Data type: INTEGER_32
C1082/3 ECAT DC Current Deviation	Index: 23493.3 _d = 0x5BC5.0x03 _h

Current maximum deviation of the distributed clocks of all devices in nanoseconds.

Display range (min. value unit max. value)		
0	ns	2147483647

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1086/2

Parameter Name:	Data type: UNSIGNED_32
C1086/2 ECAT Configuration Checksum CRC-32	Index: 23489.2 _d = 0x5BC1.0x02 _h

Checksum of the master configuration file (loaded from stack/active). The stack determines the checksum of the loaded/active master configuration file. If this file does not exist, the value "0x00000000" is displayed.

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1086/3

Parameter Name:	Data type: UNSIGNED_32
C1086/3 ECAT Number of found Slaves	Index: 23489.3 _d = 0x5BC1.0x03 _h

Number of slaves connected to the fieldbus

Display range (min. value unit max. value)		
0		4294967295

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1086/4

Parameter Name:	Data type: UNSIGNED_32
C1086/4 ECAT Number of found DC-Slaves	Index: 23489.4 _d = 0x5BC1.0x04 _h

Number of slaves connected to the fieldbus supporting distributed clocks

Display range (min. value unit max. value)		
0		4294967295

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1086/5

Parameter Name:	Data type: UNSIGNED_32
C1086/5 ECAT Number Slaves in Configuration	Index: 23489.5 _d = 0x5BC1.0x05 _h
Number of slaves configured in the master configuration file	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1086/6

Parameter Name:	Data type: UNSIGNED_32
C1086/6 ECAT Number Mailbox Slaves in Configuration	Index: 23489.6 _d = 0x5BC1.0x06 _h
Number of mailbox slaves configured in the master configuration file	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1086/7

Parameter Name:	Data type: UNSIGNED_32
C1086/7 ECAT TX-Frames Counter	Index: 23493.7 _d = 0x5BC5.0x07 _h
Number of sent frames	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1086/8

Parameter Name:	Data type: UNSIGNED_32
C1086/8 ECAT RX-Frames Counter	Index: 23489.8 _d = 0x5BC1.0x08 _h
Number of received frames	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1086/9

Parameter Name:	Data type: UNSIGNED_32
C1086/9 ECAT Lost Frames Counter	Index: 23489.9 _d = 0x5BC1.0x09 _h
Number of lost frames	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1086/10

Parameter Name:	Data type: UNSIGNED_32
C1086/10 ECAT Cyclic Frames Counter	Index: 23489.10 _d = 0x5BC1.0x0A _h
Number of cyclic frames	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1086/11

Parameter Name:	C1086/11 ECAT Cyclic Datagrams Counter	Data type: UNSIGNED_32 Index: 23489.11 _d = 0x5BC1.0x0B _h
Number of cyclic datagrams		
Display range (min. value unit max. value)		
0		4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1086/12

Parameter Name:	C1086/12 ECAT Acyclic Frames Counter	Data type: UNSIGNED_32 Index: 23489.12 _d = 0x5BC1.0x0C _h
Number of acyclic frames		
Display range (min. value unit max. value)		
0		4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1086/13

Parameter Name:	C1086/13 ECAT Acyclic Datagrams Counter	Data type: UNSIGNED_32 Index: 23489.13 _d = 0x5BC1.0x0D _h
Number of acyclic datagrams		
Display range (min. value unit max. value)		
0		4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1086/14

Parameter Name:	C1086/14 ECAT Clear specific Counters	Data type: UNSIGNED_32 Index: 23489.14 _d = 0x5BC1.0x0E _h
Reset of frame and datagram counters (C1086/7 ... 13)		
Selection list (Lenze setting printed in bold)		
0	No Operation	
1	Clear all Counters	
2	Clear TX-Frame Counter	
4	Clear RX-Frame Counter	
8	Clear Lost Frame Counter	
16	Clear Cyclic Frame Counter	
32	Clear Cyclic Datagram Counter	
64	Clear Acyclic Frame Counter	
128	Clear Acyclic Datagram Counter	
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer		

C1095/1

Parameter Name: C1095/1 ECAT Slave-Address-Mode	Data type: UNSIGNED_8 Index: 23480.1 _d = 0x5BB8.0x01 _h
---	--

The address mode refers to the slave address ([C1095/2](#)).

Selection list (Lenze setting printed in bold)		
0	Position Addressing	Auto-increment addressing: The addresses depend on the position of the respective slave in the EtherCAT bus (i.e. 0, -1, -2, ...)
1	Node Addressing	Physical addresses: The addresses are consecutively assigned by the EtherCAT configurator (i.e. 1001, 1002, ...)

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer

C1095/2

Parameter Name: C1095/2 ECAT Slave-Address	Data type: INTEGER_32 Index: 23480.2 _d = 0x5BB8.0x02 _h
--	--

This code can be used to select a slave and display its parameters.

The addressing mode can be selected in code [C1095/1](#). A maximum of 4096 slave addresses can follow after the master address.

Setting range (min. value unit max. value)	Lenze setting
-65536 65536	0

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/1

Parameter Name: C1096/1 ECAT Is Entry Valid	Data type: UNSIGNED_8 Index: 23479.1 _d = 0x5BB7.0x01 _h
---	--

This code indicates whether the entries/values of the subcodes C1096/2 ... /38 are valid.

- Value "0": Invalid entry/value
- Value "1": Valid entry/value (The parameters for an existing slave are displayed.)

Display range (min. value unit max. value)
0 1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/2

Parameter Name: C1096/2 ECAT Slave Vendor-ID	Data type: UNSIGNED_32 Index: 23479.2 _d = 0x5BB7.0x02 _h
--	---

ID number of the slave manufacturer (e.g. 0x0000003B for Lenze)

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/3

Parameter Name: C1096/3 ECAT Product Code	Data type: UNSIGNED_32 Index: 23479.3 _d = 0x5BB7.0x03 _h
---	---

Product code of the slave

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/4

Parameter Name: C1096/4 ECAT Revision Number	Data type: UNSIGNED_32 Index: 23479.4 _d = 0x5BB7.0x04 _h
--	---

Revision number of the slave

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/5

Parameter Name: C1096/5 ECAT Serial Number	Data type: UNSIGNED_32 Index: 23479.5 _d = 0x5BB7.0x05 _h
Serial number of the slave	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/6

Parameter Name: C1096/6 ECAT Slave Device Name	Data type: VISIBLE_STRING Index: 23479.6 _d = 0x5BB7.0x06 _h
Device name of the slave The name is stored in the master configuration file.	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/7

Parameter Name: C1096/7 ECAT Auto Increment Address	Data type: INTEGER_32 Index: 23479.7 _d = 0x5BB7.0x07 _h
The auto-increment address is determined by the slave position. The first node obtains the address "0". Consecutive descending numbers: 0, -1, -2 ...	
Display range (min. value unit max. value)	
-65536 0	
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/8

Parameter Name: C1096/8 ECAT Physical Address	Data type: UNSIGNED_16 Index: 23479.8 _d = 0x5BB7.0x08 _h
Physical address of the slave Consecutive ascending numbers, starting with 1001.	
Display range (min. value unit max. value)	
0 65536	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/9

Parameter Name: C1096/9 ECAT Config Physical Address	Data type: UNSIGNED_16 Index: 23479.9 _d = 0x5BB7.0x09 _h
The physical slave address set in the master configuration file This address is assigned by the configurator; consecutive ascending numbers starting with 1001.	
Display range (min. value unit max. value)	
0 65536	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/10

Parameter Name: C1096/10 ECAT Alias Address	Data type: UNSIGNED_16 Index: 23479.10 _d = 0x5BB7.0x0A _h
Alias addresses are addresses which are set directly on the device (e.g. via DIP switches). At the moment, our system does not support alias addresses.	
Display range (min. value unit max. value)	
0 65536	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/11

Parameter | Name:

C1096/11 | ECAT Port State

Data type: BITFIELD_16

Index: 23479.11_d =0x5BB7.0x0B_h

Display of the port states/connections (ports 0 ... 3) of a slave.
The states are bit-coded (WORD).

Value is bit-coded:

Bit 0	Port 0	1 (TRUE) = slave connected (logical result of bits 0 ... 11)
Bit 1	Port 1	
Bit 2	Port 2	
Bit 3	Port 3	
Bit 4	Port 0	1 (TRUE) = link known
Bit 5	Port 1	
Bit 6	Port 2	
Bit 7	Port 3	
Bit 8	Port 0	1 (TRUE) = loop closed
Bit 9	Port 1	
Bit 10	Port 2	
Bit 11	Port 3	
Bit 12	Port 0	1 (TRUE) = signal received
Bit 13	Port 1	
Bit 14	Port 2	
Bit 15	Port 3	

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer
C1096/12

Parameter | Name:

C1096/12 | ECAT DC Support

Data type: UNSIGNED_8

Index: 23479.12_d =0x5BB7.0x0C_h

Support of distributed clocks by the slave:

- Value "0": Distributed clocks are not supported.
- Value "1": Support of distributed clocks.

Display range (min. value | unit | max. value)

0 | | 1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer
C1096/13

Parameter | Name:

C1096/13 | ECAT DC Support 64-Bit

Data type: UNSIGNED_8

Index: 23479.13_d =0x5BB7.0x0D_h

Support of distributed clocks (64 bits) by the slave:

- Value "0": Distributed clocks (64 bits) are not supported.
- Value "1": Support of distributed clocks (64 bits).

Display range (min. value | unit | max. value)

0 | | 1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/14

Parameter Name: C1096/14 ECAT Mailbox Support	Data type: UNSIGNED_8 Index: 23479.14 _d = 0x5BB7.0x0E _h
Mailbox support by the slave: - Value "0": Mailbox is not supported. - Value "1": Support of mailbox.	
Display range (min. value unit max. value)	
0	1
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/15

Parameter Name: C1096/15 ECAT Requested State	Data type: UNSIGNED_32 Index: 23479.15 _d = 0x5BB7.0x0F _h
This code displays the state specified by the master. At the same time, this code can also be used by the user to request a state. Note: This is a debug code. Setting this parameter only sets the state of the currently selected slave. All other nodes remains in their states. Changing the slave state via this code may result in an unstable system state.	
Selection list (Lenze setting printed in bold)	
0	Invalid
1	Init
2	Pre-Operational
4	Safe-Operational
8	Operational
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer	

C1096/16

Parameter Name: C1096/16 ECAT Current State	Data type: UNSIGNED_32 Index: 23479.16 _d = 0x5BB7.0x10 _h
Display of the current slave state	
Selection list(read only)	
0	Unknown
1	Init
2	Pre-Operational
4	Safe-Operational
8	Operational
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/17

Parameter Name: C1096/17 ECAT Is Error Flag set	Data type: UNSIGNED_8 Index: 23479.17 _d = 0x5BB7.0x11 _h
- Value "0": No error - Value "1": Active error	
Display range (min. value unit max. value)	
0	1
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/18

Parameter Name: C1096/18 ECAT Enable Linkmessages	Data type: UNSIGNED_8 Index: 23479.18 _d = 0x5BB7.0x12 _h
---	---

Debug parameter:

If the parameter is set to the value "1", all state changes of the slave are output as diagnostic frames via the Ethernet interface. In the event of a fault, this enables error diagnosing by means of a Wireshark recording.

(»Wireshark«: Program for analysing network communication links)

- Value "0": No link messages
- Value "1": Enable link messages

Setting range (min. value unit max. value)	Lenze setting
0	1 0
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/19

Parameter Name: C1096/19 ECAT Error Code	Data type: UNSIGNED_32 Index: 23479.19 _d = 0x5BB7.0x13 _h
--	--

Display of the AL status code (slave register "0x0134 : 0x0135")

☒ Read access
☐ Write access
☐ CINH
☐ PLC STOP
☐ No transfer

AL status codes

Code (hex)	Description	Current state or state change	Resulting state
0x0000	No error	Any	Current state
0x0001	Unspecified error	Any	Any + E
0x0011	Invalid requested state	I→S, I→O, P→O, O→B, S→B, P→B	Current state + E
0x0012	Unknown requested state	Any	Current state + E
0x0013	Bootstrap not supported	I→B	I + E
0x0014	No valid firmware	I→P	I + E
0x0015	Invalid mailbox configuration	I→B	I + E
0x0016	Invalid mailbox configuration	I→P	I + E
0x0017	Invalid sync manager configuration	P→S, S→O	Current state + E
0x0018	No valid inputs available	O, S, P→S	P + E
0x0019	No valid outputs	O, S→O	S + E
0x001A	Synchronisation error	O, S→O	S + E
0x001B	Sync manager watchdog	O, S	S + E
0x001C	Invalid sync manager types	O, S P→S	S + E P + E
0x001D	Invalid output configuration	O, S P→S	S + E P + E
0x001E	Invalid input configuration	O, S, P→S	P + E
0x001F	Invalid watchdog configuration	O, S, P→S	P + E

Legend:

I: Init (initialisation)

B: Bootstrap (not supported)

P: Pre-Operational

S: Safe Operational

O: Operational

E: Error Flag

Code (hex)	Description	Current state or state change	Resulting state
0x0020	Slave needs cold start	Any	Current state + E
0x0021	Slave requires <i>Init</i> state	B, P, S, O	Current state + E
0x0022	Slave requires <i>Pre-Operational</i> state	s	S + E
		O	O + E
0x0023	Slave requires <i>Safe-Operational</i> state	O	O + E
0x0030	Invalid DC configuration	O, S	S + E
0x0031	Invalid DC Latch configuration	O, S	S + E
0x0032	PLL error	O, S	S + E
0x0033	DC I/O error	O, S	S + E
0x0034	DC time-out error	O, S	S + E
0x0042	MBX_EOE	B, P, S, O	Current state + E
0x0043	MBX_COE	B, P, S, O	Current state + E
0x0044	MBX_FOE	B, P, S, O	Current state + E
0x0045	MBX_SOE	B, P, S, O	Current state + E
0x004F	MBX_VOE	B, P, S, O	Current state + E

Legend:

I: Init (initialisation)

B: Bootstrap (not supported)

P: Pre-Operational

S: Safe Operational

O: Operational

E: Error Flag

C1096/20

Parameter | Name:

C1096/20 | ECAT Is Sync Pulse active

Data type: UNSIGNED_8
Index: 23479.20_d =
0x5BB7.0x14_h

- Value "0": Distributed clocks sync pulse on slave not activated.
- Value "1": Distributed clocks sync pulse on slave activated.

Display range (min. value | unit | max. value)

0 1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/21

Parameter | Name:

C1096/21 | ECAT DC Sync 0 Period

Data type: UNSIGNED_32
Index: 23479.21_d =
0x5BB7.0x15_h

Period within which setpoints from the control system are expected if distributed clock support is activated.

Display range (min. value | unit | max. value)

0 4294967295

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/22

Parameter Name:		Data type: UNSIGNED_32
C1096/22 ECAT DC Sync 1 Period		Index: 23479.22 _d = 0x5BB7.0x16 _h
Distributed clocks sync 1 period		
Note: At the moment, Lenze slaves do not support distributed clocks on sync 1.		
Display range (min. value unit max. value)		
0		4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1096/23

Parameter Name:		Data type: UNSIGNED_32
C1096/23 ECAT Bus Scan Error Code		Index: 23479.23 _d = 0x5BB7.0x17 _h
Error code after bus scan		
The parameter has the value "0" if the slave corresponds to the configuration. In the event of an error, the error code, e.g. 0x9811001E _{hex} (Bus Scan Mismatch), is stored here.		
See also System error messages (157).		
Display range (min. value unit max. value)		
0		4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1096/24

Parameter Name:		Data type: UNSIGNED_16
C1096/24 ECAT RX Error Counter Port 0		Index: 23479.24 _d = 0x5BB7.0x18 _h
Number of errors that have occurred during data reception via port 0.		
Display range (min. value unit max. value)		
0		65535
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1096/25

Parameter Name:		Data type: UNSIGNED_16
C1096/25 ECAT RX Error Counter Port 1		Index: 23479.25 _d = 0x5BB7.0x19 _h
Number of errors that have occurred during data reception via port 1.		
Display range (min. value unit max. value)		
0		65535
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1096/26

Parameter Name:		Data type: UNSIGNED_16
C1096/26 ECAT RX Error Counter Port 2		Index: 23479.26 _d = 0x5BB7.0x1A _h
Number of errors that have occurred during data reception via port 2.		
Display range (min. value unit max. value)		
0		65535
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1096/27

Parameter | Name:

C1096/27 | ECAT RX Error Counter Port 3

Data type: UNSIGNED_16

Index: 23479.27_d =

0x5BB7.0x1B_h

Number of errors that have occurred during data reception via port 3.

Display range (min. value | unit | max. value)

0 65535

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/28

Parameter | Name:

C1096/28 | Forwarded ECAT RX Error Counter Port 0

Data type: UNSIGNED_8

Index: 23479.28_d =

0x5BB7.0x1C_h

Number of errors which have occurred at upstream slave nodes (referred to the position of the current slave in the bus topology) during data reception via port 0.

Display range (min. value | unit | max. value)

0 255

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/29

Parameter | Name:

C1096/29 | Forwarded ECAT RX Error Counter Port 1

Data type: UNSIGNED_8

Index: 23479.29_d =

0x5BB7.0x1D_h

Number of errors which have occurred at upstream slave nodes (referred to the position of the current slave in the bus topology) during data reception via port 1.

Display range (min. value | unit | max. value)

0 255

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/30

Parameter | Name:

C1096/30 | Forwarded ECAT RX Error Counter Port 2

Data type: UNSIGNED_8

Index: 23479.30_d =

0x5BB7.0x1E_h

Number of errors which have occurred at upstream slave nodes (referred to the position of the current slave in the bus topology) during data reception via port 2.

Display range (min. value | unit | max. value)

0 255

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/31

Parameter | Name:

C1096/31 | Forwarded ECAT RX Error Counter Port 3

Data type: UNSIGNED_8

Index: 23479.31_d =

0x5BB7.0x1F_h

Number of errors which have occurred at upstream slave nodes (referred to the position of the current slave in the bus topology) during data reception via port 3.

Display range (min. value | unit | max. value)

0 255

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1096/32

Parameter Name:	Data type: UNSIGNED_8
C1096/32 ECAT Processing Unit Error Counter	Index: 23479.32 _d = 0x5BB7.0x20 _h
Number of errors which have occurred in the processing unit. EtherCAT slave controller error counter register "0x030C": Internal slave error	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/33

Parameter Name:	Data type: UNSIGNED_8
C1096/33 ECAT PDI Error Counter	Index: 23479.33 _d = 0x5BB7.0x21 _h
Number of internal errors of the slave indicated by the process data interface.	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/35

Parameter Name:	Data type: UNSIGNED_8
C1096/35 ECAT Lost Link Counter Port 0	Index: 23479.35 _d = 0x5BB7.0x23 _h
Number of connection interruptions on port 0	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/36

Parameter Name:	Data type: UNSIGNED_8
C1096/36 ECAT Lost Link Counter Port 1	Index: 23479.36 _d = 0x5BB7.0x24 _h
Number of connection interruptions on port 1	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/37

Parameter Name:	Data type: UNSIGNED_8
C1096/37 ECAT Lost Link Counter Port 2	Index: 23479.37 _d = 0x5BB7.0x25 _h
Number of connection interruptions on port 2	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1096/38

Parameter Name:	Data type: UNSIGNED_8
C1096/38 ECAT Lost Link Counter Port 3	Index: 23479.38 _d = 0x5BB7.0x26 _h
Number of connection interruptions on port 3	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

12.3 Parameters of the MC-ETC communication card in slot 2

The parameters are listed in numerically ascending order.

C1531

Parameter Name:	Data type: VISIBLE_STRING Index: 23044 _d = 5A04 _h
C1531 Device: type key	
Identification of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1532

Parameter Name:	Data type: VISIBLE_STRING Index: 23043 _d = 5A03 _h
C1532 Device: type version	
Version number of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1533

Parameter Name:	Data type: VISIBLE_STRING Index: 23042 _d = 5A02 _h
C1533 Device: name	
Device name of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1534

Parameter Name:	Data type: VISIBLE_STRING Index: 23041 _d = 5A01 _h
C1534 Device: software revision	
Software version of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1535

Parameter Name:	Data type: VISIBLE_STRING Index: 23040 _d = 5A00 _h
C1535 Device: hardware revision	
Hardware version of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1536

Parameter Name:	Data type: VISIBLE_STRING Index: 23039 _d = 59FF _h
C1536 Device: serial number	
Serial number of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1537

Parameter Name:	Data type: VISIBLE_STRING Index: 23038 _d = 59FE _h
C1537 Device: manufacturer	
Manufacturer of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1538

Parameter Name:	Data type: VISIBLE_STRING Index: 23037 _d = 59FD _h
C1538 Device: manufacturing date	
Manufacturing date of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

12.4 Interface parameters of the MC-ETC communication card in slot 2

The parameters are listed in numerically ascending order.

C1574

Parameter Name:	Data type: VISIBLE_STRING Index: 23001 _d = 59D9 _h
C1574 ECAT: MAC address	
MAC address of the interface	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1580/1

Parameter Name:	Data type: DATE Index: 22995.1 _d = 0x59D3.0x01 _h
C1580/1 ECAT Master Configuration Date	
Date of the master configuration file in the file system of the IPC The data manager determines the date of the master configuration file in the ...\\storage\\IPC\\PLC directory. If this file does not exist, the date "01.01.1900" is displayed.	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1580/2

Parameter Name:	Data type: UNSIGNED_32 Index: 22995.2 _d = 0x59D3.0x02 _h
C1580/2 ECAT Master Config Checksum	
Checksum of the master configuration file in the file system of the IPC The data manager determines the checksum of the master configuration file in the ...\\storage\\IPC\\PLC directory. If this file does not exist, the value "0x00000000" is displayed.	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1580/3

Parameter Name:	Data type: UNSIGNED_32 Index: 22995.3 _d = 0x59D3.0x03 _h
C1580/3 ECAT Stack Master Checksum	
The stack determines the checksum of the loaded/active master configuration file. If there is no loaded/active master configuration file, the value "0x00000000" is displayed.	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1580/4

Parameter Name:	Data type: UNSIGNED_8 Index: 22995.4 _d = 0x59D3.0x04 _h
C1580/4 ECAT Bus Scan Match	
Brief information whether the master configuration corresponds to the physical bus structure. The master configuration of the stack is compared with the actual bus structure.	
Selection list (Lenze setting printed in bold)	
0	No match
1	OK
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

Info	
	The master configuration does not correspond to the bus structure.
	The master configuration corresponds to the bus structure.

C1581/1

Parameter Name:	Data type: UNSIGNED_32
C1581/1 ECAT State Change Command	Index: 22994.1 _d = 0x59D2.0x01 _h

This code can be used to change the master state.

Note: The parameter should only be set to analyse start problems (e.g. master/bus does not reach the Operational state). Setting this parameter only sets the state of the master stack. All other nodes remain in their previous states. Changing the master state via this code may result in an unstable system state.

- A higher bus state than the one initiated by the PLC can never be reached.
- A reload of the master configuration cannot be initiated.
- Loading the PLC program normally stops the bus and sets it to the *Pre-Operational* state. If this is not possible, the bus remains in the Unknown or Initialisation state.
- Starting the PLC program normally sets the bus to the Operational state. If this is not possible, the bus remains in the *Pre-Operational* or *Safe Operational* state.

Selection list (Lenze setting printed in bold)	
0	No Operation
1	Init
2	Pre-Operational
4	Safe-Operational
8	Operational

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer

C1581/2

Parameter Name:	Data type: UNSIGNED_8
C1581/2 ECAT Master State	Index: 22994.2 _d = 0x59D2.0x02 _h

Display of the current master state

Selection list(read only)	
0	Unknown
1	Init
2	Pre-Operational
3	Bootstrap Mode
4	Safe-Operational
8	Operational

Bootstrap Mode is not supported.

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1581/3

Parameter Name:	Data type: UNSIGNED_8
C1581/3 ECAT Master in requested Mode	Index: 22994.3 _d = 0x59D2.0x03 _h

The state is usually requested by the PLC. For possible requested states see [C1581/1](#) (184).

- Value "0": Master is not in the requested state.
- Value "1": Master is in the requested state.

Display range (min. value unit max. value)	
0	1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1581/4

Parameter | Name:

C1581/4 | ECAT Slaves in requested Mode

Data type: UNSIGNED_8

Index: 22994.4_d =0x59D2.0x04_h

- Value "0": One or more slaves are not in the state requested by the master.
- Value "1": All slaves are in the state requested by the master.

Display range (min. value | unit | max. value)

0

1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer**C1581/5**

Parameter | Name:

C1581/5 | ECAT Master State Summary

Data type: BITFIELD_32

Index: 22994.5_d =0x59D2.0x05_h

Display of additional information about the current master state

The bits are set to the value 1 when the respective states are reached.

Value is bit-coded:

Bit 0	Master ok
Bit 1	Reserved 1
Bit 2	Reserved 2
Bit 3	Reserved 3
Bit 4	Init
Bit 5	Pre-Operational
Bit 6	Safe Operational
Bit 7	Operational
Bit 8	Slaves in Requested State
Bit 9	Master in Requested State
Bit 10	Bus Scan Match
Bit 11	Reserved 4
Bit 12	DC: Activated
Bit 13	DC: Synchronised
Bit 14	DC: Busy
Bit 15	Reserved 5
Bit 16	Link Up
Bit 17	Reserved 6
...	...
Bit 31	Reserved 20

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer**C1581/6**

Parameter | Name:

C1581/6 | ECAT BusScan

Data type: UNSIGNED_8

Index: 22994.6_d =0x59D2.0x06_h

Activation of the bus scan

The bus scan updates all EtherCAT codes.

Selection list (Lenze setting printed in bold)

0	No Operation
1	Scan Bus

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer

C1582/1

Parameter Name:	Data type: BITFIELD_32
C1582/1 ECAT Distributed Clocks State	Index: 22993.1 _d = 0x59D1.0x01 _h

Display of the state of the distributed clocks
The bits are set to the value 1 when the respective states are reached.

Value is bit-coded:	
Bit 0	DC: Activated
Bit 1	DC: Synchronised
Bit 2	DC: Busy

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1582/2

Parameter Name:	Data type: UNSIGNED_32
C1582/2 ECAT DC Slave Sync Deviation Limit	Index: 22993.2 _d = 0x59D1.0x02 _h

Permitted deviation of the distributed clocks of all devices in nanoseconds. If the permitted deviation is exceeded, the master initiates a resynchronisation of the distributed clocks.

Display range (min. value unit max. value)		
0	ns	4294967295

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1582/3

Parameter Name:	Data type: INTEGER_32
C1582/3 ECAT DC Current Deviation	Index: 22993.3 _d = 0x59D1.0x03 _h

Current maximum deviation of the distributed clocks of all devices in nanoseconds.

Display range (min. value unit max. value)		
0	ns	2147483647

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1586/2

Parameter Name:	Data type: UNSIGNED_32
C1586/2 ECAT Configuration Checksum CRC-32	Index: 22989.2 _d = 0x59CD.0x02 _h

Checksum of the master configuration file (loaded from stack/active). The stack determines the checksum of the loaded/active master configuration file. If this file does not exist, the value "0x00000000" is displayed.

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1586/3

Parameter Name:	Data type: UNSIGNED_32
C1586/3 ECAT Number of found Slaves	Index: 22989.3 _d = 0x59CD.0x03 _h

Number of slaves connected to the fieldbus

Display range (min. value unit max. value)		
0		4294967295

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1586/4

Parameter Name:	Data type: UNSIGNED_32
C1586/4 ECAT Number of found DC-Slaves	Index: 22989.4 _d = 0x59CD.0x04 _h

Number of slaves connected to the fieldbus supporting distributed clocks

Display range (min. value unit max. value)		
0		4294967295

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1586/5

Parameter Name:	Data type: UNSIGNED_32
C1586/5 ECAT Number Slaves in Configuration	Index: 22989.5 _d = 0x59CD.0x05 _h
Number of slaves configured in the master configuration file	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1586/6

Parameter Name:	Data type: UNSIGNED_32
C1586/6 ECAT Number Mailbox Slaves in Configuration	Index: 22989.6 _d = 0x59CD.0x06 _h
Number of mailbox slaves configured in the master configuration file	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1586/7

Parameter Name:	Data type: UNSIGNED_32
C1586/7 ECAT TX-Frames Counter	Index: 22989.7 _d = 0x59CD.0x07 _h
Number of sent frames	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1586/8

Parameter Name:	Data type: UNSIGNED_32
C1586/8 ECAT RX-Frames Counter	Index: 22989.8 _d = 0x59CD.0x08 _h
Number of received frames	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1586/9

Parameter Name:	Data type: UNSIGNED_32
C1586/9 ECAT Lost Frames Counter	Index: 22989.9 _d = 0x59CD.0x09 _h
Number of lost frames	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1586/10

Parameter Name:	Data type: UNSIGNED_32
C1586/10 ECAT Cyclic Frames Counter	Index: 22989.10 _d = 0x59CD.0x0A _h
Number of cyclic frames	
Display range (min. value unit max. value)	
0	4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1586/11

Parameter Name:	C1586/11 ECAT Cyclic Datagrams Counter	Data type: UNSIGNED_32 Index: 22989.11 _d = 0x59CD.0x0B _h
Number of cyclic datagrams		
Display range (min. value unit max. value)		
0		4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1586/12

Parameter Name:	C1586/12 ECAT Acyclic Frames Counter	Data type: UNSIGNED_32 Index: 22989.12 _d = 0x5BC1.0x0C _h
Number of acyclic frames		
Display range (min. value unit max. value)		
0		4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1586/13

Parameter Name:	C1586/13 ECAT Acyclic Datagrams Counter	Data type: UNSIGNED_32 Index: 22989.13 _d = 0x59CD.0x0D _h
Number of acyclic datagrams		
Display range (min. value unit max. value)		
0		4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1586/14

Parameter Name:	C1586/14 ECAT Clear specific Counters	Data type: UNSIGNED_32 Index: 22989.14 _d = 0x59CD.0x0E _h
Reset of frame and datagram counters (C1586/7 ... 13)		
Selection list (Lenze setting printed in bold)		
0	No Operation	
1	Clear all Counters	
2	Clear TX-Frame Counter	
4	Clear RX-Frame Counter	
8	Clear Lost Frame Counter	
16	Clear Cyclic Frame Counter	
32	Clear Cyclic Datagram Counter	
64	Clear Acyclic Frame Counter	
128	Clear Acyclic Datagram Counter	
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer		

C1595/1

Parameter Name: C1595/1 ECAT Slave-Address-Mode	Data type: UNSIGNED_8 Index: 22980.1 _d = 0x59C4.0x01 _h
---	--

The address mode refers to the slave address ([C1595/2](#)).

Selection list (Lenze setting printed in bold)		
0	Position Addressing	Auto-increment addressing: The addresses depend on the position of the respective slave in the EtherCAT bus (i.e. 0, -1, -2, ...)
1	Node Addressing	Physical addresses: The addresses are consecutively assigned by the EtherCAT configurator (i.e. 1001, 1002, ...)

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer

C1595/2

Parameter Name: C1595/2 ECAT Slave-Address	Data type: INTEGER_32 Index: 22980.2 _d = 0x59C4.0x02 _h
--	--

This code can be used to select a slave and display its parameters.

The addressing mode can be selected in code [C1595/1](#). A maximum of 4096 slave addresses can follow after the master address.

Setting range (min. value unit max. value)		
-65536		65536

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1596/1

Parameter Name: C1596/1 ECAT Is Entry Valid	Data type: UNSIGNED_8 Index: 22979.1 _d = 0x59C3.0x01 _h
---	--

This code indicates whether the entries/values of the subcodes C1596/2 ... /38 are valid.

- Value "0": Invalid entry/value
- Value "1": Valid entry/value (The parameters for an existing slave are displayed.)

Display range (min. value unit max. value)		
0		1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1596/2

Parameter Name: C1596/2 ECAT Slave Vendor-ID	Data type: UNSIGNED_32 Index: 22979.2 _d = 0x59C3.0x02 _h
--	---

ID number of the slave manufacturer (e.g. 0x0000003B for Lenze)

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1596/3

Parameter Name: C1596/3 ECAT Product Code	Data type: UNSIGNED_32 Index: 22979.3 _d = 0x59C3.0x03 _h
---	---

Product code of the slave

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1596/4

Parameter Name: C1596/4 ECAT Revision Number	Data type: UNSIGNED_32 Index: 22979.4 _d = 0x59C3.0x04 _h
--	---

Revision number of the slave

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1596/5

Parameter Name: C1596/5 ECAT Serial Number	Data type: UNSIGNED_32 Index: 22979.5 _d = 0x59C3.0x05 _h
Serial number of the slave	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/6

Parameter Name: C1596/6 ECAT Slave Device Name	Data type: VISIBLE_STRING Index: 22979.6 _d = 0x59C3.0x06 _h
Device name of the slave The name is stored in the master configuration file.	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/7

Parameter Name: C1596/7 ECAT Auto Increment Address	Data type: INTEGER_32 Index: 22979.7 _d = 0x59C3.0x07 _h
The auto-increment address is determined by the slave position. The first node obtains the address "0". Consecutive descending numbers: 0, -1, -2 ...	
Display range (min. value unit max. value)	
-65536 0	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/8

Parameter Name: C1596/8 ECAT Physical Address	Data type: UNSIGNED_16 Index: 22979.8 _d = 0x59C3.0x08 _h
Physical address of the slave Consecutive ascending numbers, starting with 1001.	
Display range (min. value unit max. value)	
0 65536	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/9

Parameter Name: C1596/9 ECAT Config Physical Address	Data type: UNSIGNED_16 Index: 22979.9 _d = 0x59C3.0x09 _h
The physical slave address set in the master configuration file This address is assigned by the configurator; consecutive ascending numbers starting with 1001.	
Display range (min. value unit max. value)	
0 65536	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/10

Parameter Name: C1596/10 ECAT Alias Address	Data type: UNSIGNED_16 Index: 22979.10 _d = 0x59C3.0x0A _h
Alias addresses are addresses which are set directly on the device (e.g. via DIP switches). At the moment, our system does not support alias addresses.	
Display range (min. value unit max. value)	
0 65536	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/11

Parameter | Name:

C1596/11 | ECAT Port State

Data type: BITFIELD_16

Index: 22979.11_d =0x59C3.0x0B_h

Display of the port states/connections (ports 0 ... 3) of a slave.
The states are bit-coded (WORD).

Value is bit-coded:

Bit 0	Port 0	1 (TRUE) = slave connected (logical result of bits 0 ... 11)
Bit 1	Port 1	
Bit 2	Port 2	
Bit 3	Port 3	
Bit 4	Port 0	1 (TRUE) = link known
Bit 5	Port 1	
Bit 6	Port 2	
Bit 7	Port 3	
Bit 8	Port 0	1 (TRUE) = loop closed
Bit 9	Port 1	
Bit 10	Port 2	
Bit 11	Port 3	
Bit 12	Port 0	1 (TRUE) = signal received
Bit 13	Port 1	
Bit 14	Port 2	
Bit 15	Port 3	

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer
C1596/12

Parameter | Name:

C1596/12 | ECAT DC Support

Data type: UNSIGNED_8

Index: 22979.12_d =0x59C3.0x0C_h

Support of distributed clocks by the slave:

- Value "0": Distributed clocks are not supported.
- Value "1": Support of distributed clocks.

Display range (min. value | unit | max. value)

0 | | 1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer
C1596/13

Parameter | Name:

C1596/13 | ECAT: DC Support 64-Bit

Data type: UNSIGNED_8

Index: 22979.13_d =0x59C3.0x0D_h

Support of distributed clocks (64 bits) by the slave:

- Value "0": Distributed clocks (64 bits) are not supported.
- Value "1": Support of distributed clocks (64 bits).

Display range (min. value | unit | max. value)

0 | | 1

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

C1596/14

Parameter Name: C1596/14 ECAT Mailbox Support	Data type: UNSIGNED_8 Index: 22979.14 _d = 0x59C3.0x0E _h
Mailbox support by the slave: - Value "0": Mailbox is not supported. - Value "1": Support of mailbox.	
Display range (min. value unit max. value)	
0	1
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/15

Parameter Name: C1596/15 ECAT Requested State	Data type: UNSIGNED_32 Index: 22979.15 _d = 0x59C3.0x0F _h
This code displays the state specified by the master. At the same time, this code can also be used by the user to request a state. Note: This is a debug code. Setting this parameter only sets the state of the currently selected slave. All other nodes remains in their states. Changing the slave state via this code may result in an unstable system state.	
Selection list (Lenze setting printed in bold)	
0	Invalid
1	Init
2	Pre-Operational
4	Safe-Operational
8	Operational
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☒ No transfer	

C1596/16

Parameter Name: C1596/16 ECAT Current State	Data type: UNSIGNED_32 Index: 22979.16 _d = 0x59C3.0x10 _h
Display of the current slave state	
Selection list (read only)	
0	Unknown
1	Init
2	Pre-Operational
4	Safe-Operational
8	Operational
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/17

Parameter Name: C1596/17 ECAT Is Error Flag set	Data type: UNSIGNED_8 Index: 22979.17 _d = 0x59C3.0x11 _h
- Value "0": No error - Value "1": Active error	
Display range (min. value unit max. value)	
0	1
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/18

Parameter Name:	Data type: UNSIGNED_8
C1596/18 ECAT Enable Linkmessages	Index: 22979.18 _d = 0x59C3.0x12 _h

Debug parameter:

If the parameter is set to the value "1", all state changes of the slave are output as diagnostic frames via the Ethernet interface. In the event of a fault, this enables error diagnosing by means of a Wireshark recording.

(»Wireshark«: Program for analysing network communication links)

- Value "0": No link messages
- Value "1": Enable link messages

Setting range (min. value unit max. value)	Lenze setting
0	1 0
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/19

Parameter Name:	Data type: UNSIGNED_32
C1596/19 ECAT Error Code	Index: 22979.19 _d = 0x59C3.0x13 _h

Display of the AL status code (slave register "0x0134 : 0x0135")

☒ Read access
☐ Write access
☐ CINH
☐ PLC STOP
☐ No transfer

AL status codes

Code (hex)	Description	Current state or state change	Resulting state
0x0000	No error	Any	Current state
0x0001	Unspecified error	Any	Any + E
0x0011	Invalid requested state	I→S, I→O, P→O, O→B, S→B, P→B	Current state + E
0x0012	Unknown requested state	Any	Current state + E
0x0013	Bootstrap not supported	I→B	I + E
0x0014	No valid firmware	I→P	I + E
0x0015	Invalid mailbox configuration	I→B	I + E
0x0016	Invalid mailbox configuration	I→P	I + E
0x0017	Invalid sync manager configuration	P→S, S→O	Current state + E
0x0018	No valid inputs available	O, S, P→S	P + E
0x0019	No valid outputs	O, S→O	S + E
0x001A	Synchronisation error	O, S→O	S + E
0x001B	Sync manager watchdog	O, S	S + E
0x001C	Invalid sync manager types	O, S P→S	S + E P + E
0x001D	Invalid output configuration	O, S P→S	S + E P + E
0x001E	Invalid input configuration	O, S, P→S	P + E
0x001F	Invalid watchdog configuration	O, S, P→S	P + E

Legend:

I: Init (initialisation)

B: Bootstrap (not supported)

P: Pre-Operational

S: Safe Operational

O: Operational

E: Error Flag

Code (hex)	Description	Current state or state change	Resulting state
0x0020	Slave needs cold start	Any	Current state + E
0x0021	Slave requires <i>Init</i> state	B, P, S, O	Current state + E
0x0022	Slave requires <i>Pre-Operational</i> state	s	S + E
		O	O + E
0x0023	Slave requires <i>Safe-Operational</i> state	O	O + E
0x0030	Invalid DC configuration	O, S	S + E
0x0031	Invalid DC Latch configuration	O, S	S + E
0x0032	PLL error	O, S	S + E
0x0033	DC I/O error	O, S	S + E
0x0034	DC time-out error	O, S	S + E
0x0042	MBX_EOE	B, P, S, O	Current state + E
0x0043	MBX_COE	B, P, S, O	Current state + E
0x0044	MBX_FOE	B, P, S, O	Current state + E
0x0045	MBX_SOE	B, P, S, O	Current state + E
0x004F	MBX_VOE	B, P, S, O	Current state + E

Legend:

I: Init (initialisation)

B: Bootstrap (not supported)

P: Pre-Operational

S: Safe Operational

O: Operational

E: Error Flag

C1596/20

Parameter Name:	Data type: UNSIGNED_8
C1596/20 ECAT Is Sync Pulse active	Index: 22979.20 _d = 0x59C3.0x14 _h
- Value "0": Distributed clocks sync pulse on slave not activated. - Value "1": Distributed clocks sync pulse on slave activated.	
Display range (min. value unit max. value) 0 1	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/21

Parameter Name:	Data type: UNSIGNED_32
C1596/21 ECAT DC Sync 0 Period	Index: 22979.21 _d = 0x59C3.0x15 _h
Period within which setpoints from the control system are expected if distributed clock support is activated.	
Display range (min. value unit max. value) 0 4294967295	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/22

Parameter Name:		Data type: UNSIGNED_32	
C1596/22 ECAT DC Sync 1 Period		Index: 22979.22 _d = 0x59C3.0x16 _h	
Distributed clocks sync 1 period			
Note: At the moment, Lenze slaves do not support distributed clocks on sync 1.			
Display range (min. value unit max. value)			
0		4294967295	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			

C1596/23

Parameter Name:	Data type: UNSIGNED_32	
C1596/23 ECAT Bus Scan Error Code	Index: 22979.23 _d =	
	0x59C3.0x17 _h	
Error code after bus scan		
The parameter has the value "0" if the slave corresponds to the configuration. In the event of an error, the error code, e.g. 0x9811001E _{hex} (Bus Scan Mismatch), is stored here.		
See also System error messages (157).		
Display range (min. value unit max. value)		
0		4294967295
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		

C1596/24

Parameter Name:	Data type: UNSIGNED_16		
C1596/24 ECAT RX Error Counter Port 0	Index: 22979.24 _d = 0x59C3.0x18 _h		
Number of errors that have occurred during data reception via port 0.			
Display range (min. value unit max. value)			
0		65535	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			

C1596/25

Parameter Name:	Data type: UNSIGNED_16		
C1596/25 ECAT RX Error Counter Port 1	Index: 22979.25 _d = 0x59C3.0x19 _h		
Number of errors that have occurred during data reception via port 1.			
Display range (min. value unit max. value)			
0		65535	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			

C1596/26

Parameter Name:		Data type: UNSIGNED_16	
C1596/26 ECAT RX Error Counter Port 2		Index: 22979.26 _d = 0x59C3.0x1A _h	
Number of errors that have occurred during data reception via port 2.			
Display range (min. value unit max. value)			
0		65535	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			

C1596/27

Parameter Name:	Data type: UNSIGNED_16
C1596/27 ECAT RX Error Counter Port 3	Index: 22979.27 _d = 0x59C3.0x1B _h
Number of errors that have occurred during data reception via port 3.	
Display range (min. value unit max. value)	
0	65535
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/28

Parameter Name:	Data type: UNSIGNED_8
C1596/28 Forwarded ECAT RX Error Counter Port 0	Index: 22979.28 _d = 0x59C3.0x1C _h
Number of errors which have occurred at upstream slave nodes (referred to the position of the current slave in the bus topology) during data reception via port 0.	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/29

Parameter Name:	Data type: UNSIGNED_8
C1596/29 Forwarded ECAT RX Error Counter Port 1	Index: 22979.29 _d = 0x59C3.0x1D _h
Number of errors which have occurred at upstream slave nodes (referred to the position of the current slave in the bus topology) during data reception via port 1.	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/30

Parameter Name:	Data type: UNSIGNED_8
C1596/30 Forwarded ECAT RX Error Counter Port 2	Index: 22979.30 _d = 0x59C3.0x1E _h
Number of errors which have occurred at upstream slave nodes (referred to the position of the current slave in the bus topology) during data reception via port 2.	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/31

Parameter Name:	Data type: UNSIGNED_8
C1596/31 Forwarded ECAT RX Error Counter Port 3	Index: 22979.31 _d = 0x59C3.0x1F _h
Number of errors which have occurred at upstream slave nodes (referred to the position of the current slave in the bus topology) during data reception via port 3.	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/32

Parameter Name:	Data type: UNSIGNED_8
C1596/32 ECAT Processing Unit Error Counter	Index: 22979.32 _d = 0x59C3.0x20 _h
Number of errors which have occurred in the processing unit. EtherCAT slave controller error counter register "0x030C": Internal slave error	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/33

Parameter Name:	Data type: UNSIGNED_8
C1596/33 ECAT PDI Error Counter	Index: 22979.33 _d = 0x59C3.0x21 _h
Number of internal errors of the slave indicated by the process data interface.	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/35

Parameter Name:	Data type: UNSIGNED_8
C1596/35 ECAT Lost Link Counter Port 0	Index: 22979.35 _d = 0x59C3.0x23 _h
Number of connection interruptions on port 0	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/36

Parameter Name:	Data type: UNSIGNED_8
C1596/36 ECAT Lost Link Counter Port 1	Index: 22979.36 _d = 0x59C3.0x24 _h
Number of connection interruptions on port 1	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/37

Parameter Name:	Data type: UNSIGNED_8
C1596/37 ECAT Lost Link Counter Port 2	Index: 22979.37 _d = 0x59C3.0x25 _h
Number of connection interruptions on port 2	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1596/38

Parameter Name:	Data type: UNSIGNED_8
C1596/38 ECAT Lost Link Counter Port 3	Index: 22979.38 _d = 0x59C3.0x26 _h
Number of connection interruptions on port 3	
Display range (min. value unit max. value)	
0	255
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

13 Index

A

Abort codes (SDO) [164](#)
 Addressing EtherCAT nodes using CANopen/PROFIBUS nodes [93](#)
 Addressing of the slaves [20](#)
 Addressing the PROFIBUS and CANopen stations [92](#)
 Adjusting task cycle time and DC cycle time [38](#)
 AL status codes [177](#), [193](#)
 Application notes (representation) [12](#)
 AtEm.lib [96](#)
 Auto-increment addressing [20](#)

B

Baud rate [34](#)
 bEC_NOTIFY_CYCCMD_WKC_ERROR [121](#)
 bEC_NOTIFY_DC_SLV_SYNC [120](#)
 bEC_NOTIFY_DC_STATUS [120](#)
 bEC_NOTIFY_DCL_SLV_LATCH_EVT [120](#)
 bEC_NOTIFY_DCL_STATUS [120](#)
 bEC_NOTIFY_ETH_LINK_CONNECTED [120](#)
 bEC_NOTIFY_ETH_LINK_NOT_CONNECTED [120](#)
 bEC_NOTIFY_NOT_ALL_DEVICES_OPERATIONAL [120](#)
 bEC_NOTIFY_RED_LINEBRK [120](#)
 bEC_NOTIFY_SB_STATUS [120](#)
 bEC_NOTIFY_STATUS_SLAVE_ERROR [120](#)
 Breakpoints [75](#)
 Brief description of EtherCAT [16](#)
 Bus restart [114](#), [139](#)

C

C1031 | Device - type key [166](#)
 C1032 | Device - type version [166](#)
 C1033 | Device - name [166](#)
 C1034 | Device - software version [166](#)
 C1035 | Device - hardware version [166](#)
 C1036 | Device - serial number [166](#)
 C1037 | Device - manufacturer [166](#)
 C1038 | Device - manufacturing date [166](#)
 C1074 | ECAT MAC address [167](#)
 C1080/1 | ECAT Master Configuration Date [167](#)
 C1080/2 | ECAT Master Config Checksum [167](#)
 C1080/3 | ECAT Stack Master Checksum [167](#)
 C1080/4 | ECAT Bus Scan Match [167](#)
 C1081/1 | ECAT State Change Command [168](#)
 C1081/2 | ECAT Master State [168](#)
 C1081/3 | ECAT Master in requested Mode [168](#)
 C1081/4 | ECAT Slaves in requested Mode [169](#)
 C1081/5 | ECAT Master State Summary [169](#)
 C1081/6 | ECAT BusScan [169](#)
 C1082/1 | ECAT Distributed Clocks State [170](#)
 C1082/2 | ECAT DC Slave Sync Deviation Limit [170](#)

C1082/3 | ECAT DC Current Deviation [170](#)
 C1086/10 | ECAT Cyclic Frames Counter [171](#)
 C1086/11 | ECAT Cyclic Datagrams Counter [172](#)
 C1086/12 | ECAT Acyclic Frames Counter [172](#)
 C1086/13 | ECAT Acyclic Datagrams Counter [172](#)
 C1086/14 | ECAT Clear specific Counters [172](#)
 C1086/2 | ECAT Configuration Checksum CRC-32 [170](#)
 C1086/3 | ECAT Number of found Slaves [170](#)
 C1086/4 | ECAT Number of found DC-Slaves [170](#)
 C1086/5 | ECAT Number Slaves in Configuration [171](#)
 C1086/6 | ECAT Number Mailbox Slaves in Configuration [171](#)
 C1086/7 | ECAT TX-Frames Counter [171](#)
 C1086/8 | ECAT RX-Frames Counter [171](#)
 C1086/9 | ECAT Lost Frames Counter [171](#)
 C1095/1 | ECAT Slave-Address-Mode [173](#)
 C1095/2 | ECAT Slave-Address [173](#)
 C1096/1 | ECAT Is Entry Valid [173](#)
 C1096/10 | ECAT Alias Address [174](#)
 C1096/11 | ECAT Port State [175](#)
 C1096/12 | ECAT DC Support [175](#)
 C1096/13 | ECAT DC Support 64-Bit [175](#)
 C1096/14 | ECAT Mailbox Support [176](#)
 C1096/15 | ECAT Requested State [176](#)
 C1096/16 | ECAT Current State [176](#)
 C1096/17 | ECAT Is Error Flag set [176](#)
 C1096/18 | ECAT Enable Linkmessages [177](#)
 C1096/19 | ECAT Error Code [177](#)
 C1096/2 | ECAT Slave Vendor-ID [173](#)
 C1096/20 | ECAT Is Sync Pulse active [178](#)
 C1096/21 | ECAT DC Sync 0 Period [178](#)
 C1096/22 | ECAT DC Sync 1 Period [179](#)
 C1096/23 | ECAT Bus Scan Error Code [179](#)
 C1096/24 | ECAT RX Error Counter Port 0 [179](#)
 C1096/25 | ECAT RX Error Counter Port 1 [179](#)
 C1096/26 | ECAT RX Error Counter Port 2 [179](#)
 C1096/27 | ECAT RX Error Counter Port 3 [180](#)
 C1096/28 | Forwarded ECAT RX Error Counter Port 0 [180](#)
 C1096/29 | Forwarded ECAT RX Error Counter Port 1 [180](#)
 C1096/3 | ECAT Product Code [173](#)
 C1096/30 | Forwarded ECAT RX Error Counter Port 2 [180](#)
 C1096/31 | Forwarded ECAT RX Error Counter Port 3 [180](#)
 C1096/32 | ECAT Processing Unit Error Counter [181](#)
 C1096/33 | ECAT PDI Error Counter [181](#)
 C1096/35 | ECAT Lost Link Counter Port 0 [181](#)
 C1096/36 | ECAT Lost Link Counter Port 1 [181](#)
 C1096/37 | ECAT Lost Link Counter Port 2 [181](#)
 C1096/38 | ECAT Lost Link Counter Port 3 [181](#)

- C1096/4 | ECAT Revision Number [173](#)
- C1096/5 | ECAT Serial Number [174](#)
- C1096/6 | ECAT Slave Device Name [174](#)
- C1096/7 | ECAT Auto Increment Address [174](#)
- C1096/8 | ECAT Physical Address [174](#)
- C1096/9 | ECAT Config Physical Address [174](#)
- C1531 | Device - type key [182](#)
- C1532 | Device - type version [182](#)
- C1533 | Device - name [182](#)
- C1534 | Device - software version [182](#)
- C1535 | Device - hardware version [182](#)
- C1536 | Device - serial number [182](#)
- C1537 | Device - manufacturer [182](#)
- C1538 | Device - manufacturing date [182](#)
- C1574 | ECAT MAC address [183](#)
- C1580/1 | ECAT Master Configuration Date [183](#)
- C1580/2 | ECAT Master Config Checksum [183](#)
- C1580/3 | ECAT Stack Master Checksum [183](#)
- C1580/4 | ECAT Bus Scan Match [183](#)
- C1581/1 | ECAT State Change Command [184](#)
- C1581/2 | ECAT Master State [184](#)
- C1581/3 | ECAT Master in requested Mode [184](#)
- C1581/4 | ECAT Slaves in requested Mode [185](#)
- C1581/5 | ECAT Master State Summary [185](#)
- C1581/6 | ECAT BusScan [185](#)
- C1582/1 | ECAT Distributed Clocks State [186](#)
- C1582/2 | ECAT DC Slave Sync Deviation Limit [186](#)
- C1582/3 | ECAT DC Current Deviation [186](#)
- C1586/10 | ECAT Cyclic Frames Counter [187](#)
- C1586/11 | ECAT Cyclic Datagrams Counter [188](#)
- C1586/12 | ECAT Acyclic Frames Counter [188](#)
- C1586/13 | ECAT Acyclic Datagrams Counter [188](#)
- C1586/14 | ECAT Clear specific Counters [188](#)
- C1586/2 | ECAT Configuration Checksum CRC-32 [186](#)
- C1586/3 | ECAT Number of found Slaves [186](#)
- C1586/4 | ECAT Number of found DC-Slaves [186](#)
- C1586/5 | ECAT Number Slaves in Configuration [187](#)
- C1586/6 | ECAT Number Mailbox Slaves in Configuration [187](#)
- C1586/7 | ECAT TX-Frames Counter [187](#)
- C1586/8 | ECAT RX-Frames Counter [187](#)
- C1586/9 | ECAT Lost Frames Counter [187](#)
- C1595/1 | ECAT Slave-Address-Mode [189](#)
- C1595/2 | ECAT Slave-Address [189](#)
- C1596/1 | ECAT Is Entry Valid [189](#)
- C1596/10 | ECAT Alias Address [190](#)
- C1596/11 | ECAT Port State [191](#)
- C1596/12 | ECAT DC Support [191](#)
- C1596/13 | ECAT
 - DC Support 64-Bit [191](#)
- C1596/14 | ECAT Mailbox Support [192](#)
- C1596/15 | ECAT Requested State [192](#)
- C1596/16 | ECAT Current State [192](#)
- C1596/17 | ECAT Is Error Flag set [192](#)
- C1596/18 | ECAT Enable Linkmessages [193](#)
- C1596/19 | ECAT Error Code [193](#)
- C1596/2 | ECAT Slave Vendor-ID [189](#)
- C1596/20 | ECAT Is Sync Pulse active [194](#)
- C1596/21 | ECAT DC Sync 0 Period [194](#)
- C1596/22 | ECAT DC Sync 1 Period [195](#)
- C1596/23 | ECAT Bus Scan Error Code [195](#)
- C1596/24 | ECAT RX Error Counter Port 0 [195](#)
- C1596/25 | ECAT RX Error Counter Port 1 [195](#)
- C1596/26 | ECAT RX Error Counter Port 2 [195](#)
- C1596/27 | ECAT RX Error Counter Port 3 [196](#)
- C1596/28 | Forwarded ECAT RX Error Counter Port 0 [196](#)
- C1596/29 | Forwarded ECAT RX Error Counter Port 1 [196](#)
- C1596/3 | ECAT Product Code [189](#)
- C1596/30 | Forwarded ECAT RX Error Counter Port 2 [196](#)
- C1596/31 | Forwarded ECAT RX Error Counter Port 3 [196](#)
- C1596/32 | ECAT Processing Unit Error Counter [197](#)
- C1596/33 | ECAT PDI Error Counter [197](#)
- C1596/35 | ECAT Lost Link Counter Port 0 [197](#)
- C1596/36 | ECAT Lost Link Counter Port 1 [197](#)
- C1596/37 | ECAT Lost Link Counter Port 2 [197](#)
- C1596/38 | ECAT Lost Link Counter Port 3 [197](#)
- C1596/4 | ECAT Revision Number [189](#)
- C1596/5 | ECAT Serial Number [190](#)
- C1596/6 | ECAT Slave Device Name [190](#)
- C1596/7 | ECAT Auto Increment Address [190](#)
- C1596/8 | ECAT Physical Address [190](#)
- C1596/9 | ECAT Config Physical Address [190](#)
- Cable length (max.) [34](#)
- Cable type [34](#)
- Calculating the minimum cycle time [128](#)
- Calculating the total access time to the peripheral devices [125](#)
- CANopen [91](#)
- Check of the DC synchronicity [41](#)
- Checking the system startup [80](#)
- Checking with »WebConfig« or »Engineer« [79](#)
- Codes [165](#)
- CoE/SDO error codes (0x98110040 ... 0x9811005Dhex) [161](#)
- Commissioning of the system [42](#)
- Commissioning steps (detailed overview) [90](#)
- Commissioning steps (short overview) [43](#)
- Communication [18](#)
- Communication between engineering PC and field devices [32](#)
- Communication medium [34](#)
- Communication profile [34](#)
- Communication times and drive-specific data [35](#)
- Compiling project data [77](#)

Configuring the communication parameters [76](#)

Conventions used [10](#)

Copyright [2](#)

Creating a project folder [46](#)

Creating a task [73](#)

Creating manual configurations in the »EtherCAT Configurator« [56](#)

Cross communication [35](#)

Cycle synchronisation [35](#)

D

Data type ECAT_STATE [119](#)

DC master [36](#), [60](#)

Defining the minimum cycle time of the PLC project [125](#)

Deleting the error counter from the application [144](#)

Detecting the task utilisation [127](#)

Detecting the task utilisation of the application [126](#)

Determining the physical EtherCAT configuration [47](#)

Device

Hardware revision (C1035) [166](#)

hardware version (C1535) [182](#)

identification (C1531) [182](#)

Manufacturer (C1037) [166](#)

manufacturer (C1537) [182](#)

Manufacturing date (C1038) [166](#)

manufacturing date (C1538) [182](#)

Name (C1033) [166](#)

name (C1533) [182](#)

Serial number (C1036) [166](#)

serial number (C1536) [182](#)

Software revision (C1034) [166](#)

software version (C1534) [182](#)

Type key (C1031) [166](#)

Type version (C1032) [166](#)

version (C1532) [182](#)

Device ID [24](#)

Diagnostic codes [140](#)

Diagnostics [130](#)

Diagnostics tab [131](#)

Diagnostics with the »EtherCAT Configurator« [131](#)

Diagnostics with the »PLC Designer« [134](#)

Display of the system utilisation in the »PLC Designer« with the task editor [126](#)

Displaying the EtherCAT entries of the logbook [141](#)

Distributed clocks [36](#)

dwEC_NOTIFY_DC_SLV_SYNCDeviation [120](#)

E

EC_NOTIFY_FRAME_RESPONSE_ERROR [121](#)

EC_T_SLAVE_PROP [111](#)

EC_T_SLAVE_PROP structure [111](#)

EC_T_STATE [98](#)

EC_T_STATE structure [98](#)

ECAT

Acyclic Datagrams Counter (C1086/13) [172](#)

Acyclic Datagrams Counter (C1586/13) [188](#)

Acyclic Frames Counter (C1086/12) [172](#)

Acyclic Frames Counter (C1586/12) [188](#)

Alias Address (C1096/10) [174](#)

Alias Address (C1596/10) [190](#)

Auto Increment Address (C1096/7) [174](#)

Auto Increment Address (C1596/7) [190](#)

Bus Scan Error Code (C1096/23) [179](#)

Bus Scan Error Code (C1596/23) [195](#)

Clear specific Counters (C1086/14) [172](#)

Clear specific Counters (C1586/14) [188](#)

Current State (C1096/16) [176](#)

Current State (C1596/16) [192](#)

Cyclic Datagrams Counter (C1086/11) [172](#)

Cyclic Datagrams Counter (C1586/11) [188](#)

Cyclic Frames Counter (C1086/10) [171](#)

Cyclic Frames Counter (C1586/10) [187](#)

DC Support (C1096/12) [175](#)

DC Support (C1596/12) [191](#)

DC Support 64-Bit (C1096/13) [175](#)

DC Support 64-Bit (C1596/13) [191](#)

DC Sync 0 Period (C1096/21) [178](#)

DC Sync 0 Period (C1596/21) [194](#)

DC Sync 1 Period (C1096/22) [179](#)

DC Sync 1 Period (C1596/22) [195](#)

Enable Linkmessages (C1096/18) [177](#)

Enable Linkmessages (C1596/18) [193](#)

Error Code (C1096/19) [177](#)

Error Code (C1596/19) [193](#)

Forwarded RX Error Counter Port 0 (C1096/28) [180](#)

Forwarded RX Error Counter Port 0 (C1596/28) [196](#)

Forwarded RX Error Counter Port 1 (C1096/29) [180](#)

Forwarded RX Error Counter Port 1 (C1596/29) [196](#)

Forwarded RX Error Counter Port 2 (C1096/30) [180](#)

Forwarded RX Error Counter Port 2 (C1596/30) [196](#)

Forwarded RX Error Counter Port 3 (C1096/31) [180](#)

Forwarded RX Error Counter Port 3 (C1596/31) [196](#)

Is Entry Valid (C1096/1) [173](#)

Is Entry Valid (C1596/1) [189](#)

Is Error Flag set (C1096/17) [176](#)

Is Error Flag set (C1596/17) [192](#)

Is Sync Pulse active (C1096/20) [178](#)

Is Sync Pulse active (C1596/20) [194](#)

Lost Frames Counter (C1086/9) [171](#)

Lost Frames Counter (C1586/9) [187](#)

Lost Link Counter Port 0 (C1096/35) [181](#)

Lost Link Counter Port 0 (C1596/35) [197](#)

Lost Link Counter Port 1 (C1096/36) [181](#)

Lost Link Counter Port 1 (C1596/36) [197](#)

Lost Link Counter Port 2 (C1096/37) [181](#)

Lost Link Counter Port 2 (C1596/37) [197](#)

Lost Link Counter Port 3 (C1096/38) [181](#)

Lost Link Counter Port 3 (C1596/38) [197](#)

MAC address (C1074) [167](#)

MAC address (C1574) [183](#)

Mailbox Support (C1096/14) [176](#)

Mailbox Support (C1596/14) [192](#)

- Number of found DC-Slaves (C1086/4) [170](#)
- Number of found DC-Slaves (C1586/4) [186](#)
- Number of found Slaves (C1086/3) [170](#)
- Number of found Slaves (C1586/3) [186](#)
- PDI Error Counter (C1096/33) [181](#)
- PDI Error Counter (C1596/33) [197](#)
- Physical Address (C1096/8) [174](#)
- Physical Address (C1596/8) [190](#)
- Port State (C1096/11) [175](#)
- Port State (C1596/11) [191](#)
- Processing Unit Error Counter (C1096/32) [181](#)
- Processing Unit Error Counter (C1596/32) [197](#)
- Product Code (C1096/3) [173](#)
- Product Code (C1596/3) [189](#)
- Requested State (C1096/15) [176](#)
- Requested State (C1596/15) [192](#)
- Revision Number (C1096/4) [173](#)
- Revision Number (C1596/4) [189](#)
- RX Error Counter Port 0 (C1096/24) [179](#)
- RX Error Counter Port 0 (C1596/24) [195](#)
- RX Error Counter Port 1 (C1096/25) [179](#)
- RX Error Counter Port 1 (C1596/25) [195](#)
- RX Error Counter Port 2 (C1096/26) [179](#)
- RX Error Counter Port 2 (C1596/26) [195](#)
- RX Error Counter Port 3 (C1096/27) [180](#)
- RX Error Counter Port 3 (C1596/27) [196](#)
- RX-Frames Counter (C1086/8) [171](#)
- RX-Frames Counter (C1586/8) [187](#)
- Serial Number (C1096/5) [174](#)
- Serial Number (C1596/5) [190](#)
- Slave Device Name (C1096/6) [174](#)
- Slave Device Name (C1596/6) [190](#)
- Slave Vendor-ID (C1096/2) [173](#)
- Slave Vendor-ID (C1596/2) [189](#)
- Slave-Address (C1095/2) [173](#)
- Slave-Address (C1595/2) [189](#)
- Slave-Address-Mode (C1095/1) [173](#)
- Slave-Address-Mode (C1595/1) [189](#)
- TX-Frames Counter (C1086/7) [171](#)
- TX-Frames Counter (C1586/7) [187](#)
- ECAT Bus Scan
 - Match (C1080/4) [167](#)
 - Match (C1580/4) [183](#)
- ECAT BusScan (C1081/6) [169](#)
- ECAT BusScan (C1581/6) [185](#)
- ECAT Config.
 - Checksum CRC-32 (C1086/2) [170](#)
 - Checksum CRC-32 (C1586/2) [186](#)
 - Number Mailbox Slaves in Configuration (C1086/6) [171](#)
 - Number Mailbox Slaves in Configuration (C1586/6) [187](#)
 - Number Slaves in Configuration (C1086/5) [171](#)
 - Number Slaves in Configuration (C1586/5) [187](#)
 - Physical Address (C1096/9) [174](#)
 - Physical Address (C1596/9) [190](#)
- ECAT DC
 - Current Deviation (C1082/3) [170](#)
 - Current Deviation (C1582/3) [186](#)
 - Slave Sync Deviation Limit (C1082/2) [170](#)
 - Slave Sync Deviation Limit (C1582/2) [186](#)
 - State (C1082/1) [170](#)
 - State (C1582/1) [186](#)
- ECAT Master
 - State (C1081/2) [168](#)
 - State (C1581/2) [184](#)
 - State Summary (C1081/5) [169](#)
 - State Summary (C1581/5) [185](#)
- ECAT Master Config
 - Checksum (C1080/2) [167](#)
- ECAT Master Configuration
 - Checksum (C1580/2) [183](#)
 - Date (C1080/1) [167](#)
 - Date (C1580/1) [183](#)
- ECAT Master in requested Mode (C1081/3) [168](#)
- ECAT Master in requested Mode (C1581/3) [184](#)
- ECAT Slaves in requested Mode (C1081/4) [169](#)
- ECAT Slaves in requested Mode (C1581/4) [185](#)
- ECAT Stack
 - Master Checksum (C1080/3) [167](#)
 - Master Checksum (C1580/3) [183](#)
- ECAT State
 - Change Command (C1081/1) [168](#)
 - Change Command (C1581/1) [184](#)
- ECAT_MASTER [118](#)
- ECAT_STATE [119](#)
- ecatCoeSdoDownloadReq (FB) [122](#)
- ecatCoeSdoUploadReq (FB) [123](#)
- ECATDiagnostic (FB) [113](#)
- ecatGetMasterState (FUN) [105](#)
- ecatGetNumConfiguredSlaves (FUN) [112](#)
- ecatGetNumConnectedSlaves (FUN) [113](#)
- ecatGetSlaveId (FUN) [109](#)
- ecatGetSlaveIdAtPosition (FUN) [110](#)
- ecatGetSlaveProp (FUN) [111](#)
- ecatGetSlaveState (FUN) [106](#)
- ecatGetSlaveStateAsync (FB) [107](#)
- ecatMasterIsConfigured (FUN) [108](#)
- ecatSetMasterStateAsync (FB) [102](#)
- ecatSetSlaveStateAsync (FB) [103](#)
- ecatStartAsync (FB) [100](#)
- ecatStopAsync (FB) [101](#)
- Editing the EtherCAT I/O image [66](#)
- Engineering tools [28](#), [42](#)
- Error codes (CoE/SDO, 0x98110040 ... 0x9811005Dhex) [161](#)
- Error codes (general, 0x00000000hex, 0x98110001 ... 0x98110038hex) [159](#)
- Error codes (remote API, 0x98110181 ... 0x98110196hex) [163](#)
- Error counters of the EtherCAT slaves [143](#)
- Error scenario (example) [139](#)
- Error scenarios [145](#)

- Error types [157](#)
 - "Errors" and "Forwarded Errors" [143](#)
- EtherCAT [16](#)
- EtherCAT connection [34](#)
- EtherCAT connection (SUB-D, 9-pole socket) [34](#)
- EtherCAT cycle times [35](#)
- EtherCAT hardware for the industrial PC [26](#)
- EtherCAT master structure ECAT_MASTER [118](#)
- EtherCAT module [34](#)
- EtherCAT product codes [24](#)
- EtherCAT state machine [19](#)
- EtherCAT with CANopen or PROFIBUS [91](#)
- Evaluation of the axis state [82](#)
- Evaluation of the Axis_IO_Group state [81](#)
- Evaluation of the boot-up error message of the SM_DriveBasic.lib library [80](#)
- Executing PDO mapping [65](#)
- F**
- Field devices [23](#)
- Fieldbus scan with the »EtherCAT Configurator« [47](#)
- Fieldbus scan with the »scandf« command line tool [51](#)
- Fieldbus scanning command [51](#)
- Fixed-address addressing [20](#)
- Function block libraries [96](#)
- Function blocks and functions for diagnosing the network [112](#)
- Function blocks and functions for master/slave states [99](#)
- Function blocks for CANopen over EtherCAT (CoE) [122](#)
- Functions for the network management [108](#)
- G**
- g_EcatMaster [119](#)
- General data of the EtherCAT bus [34](#)
- General error codes (0x00000000hex, 0x98110001 ... 0x98110038hex) [159](#)
- Global EtherCAT master structure ECAT_MASTER [118](#)
- H**
- Hardware components [22](#)
- I**
- Importing missing field devices [59](#)
- Inserting devices available on the fieldbus into the »EtherCAT Configurator« project [55](#)
- Installing field devices [45](#)
- Interface parameters of the MC-ETC communication card in slot 1 [167](#)
- Interface parameters of the MC-ETC communication card in slot 2 [183](#)
- IPC logbook messages [157](#)
- L**
- L_ECANT_ErrCnt [117](#)
- L_ECANT_ErrCnt structure [117](#)
- L_ECANT_ReadErrCnt [116](#)
- L_ECANT_ResetErrCnt [117](#)
- LEDs on RJ45 socket [34](#)
- Loading and starting the PLC program [79](#)
- Loading the control configuration onto the IPC [78](#)
- Loading the master configuration onto the EtherCAT master [79](#)
- Logbook message
 - "Cannot spawn Remote API Server" [151](#)
 - "Cyclic command WKC error ..." [156](#)
 - "Ethernet cable connected" [153](#)
 - "Ethernet cable not connected" [152](#)
- Logbook messages [157](#)
 - "Slave at index X missing" with "Cyclic command WKC error ..." [154](#)
- Logbook of the IPC [140](#)
- Logging on to the control system with the »PLC Designer« [77](#)
- M**
- Max. number of Servo Drives 9400 HighLine per frame [35](#)
- MC-ETC communication card [26](#)
- Messages in the logbook of the industrial PC [142](#)
- Mixed operation (EtherCAT with CANopen or PROFIBUS) [91](#)
- N**
- Network topology [34](#)
- Notes used [12](#)
- Notifications [120](#)
- Number of nodes [34](#)
- O**
- Optimising the system [129](#)
- Overview of notifications [120](#)
- P**
- Parameter data (SDO) [35](#)
- Parameter reference [165](#)
- Parameters of the MC-ETC communication card in slot 1 [166](#)
- Parameters of the MC-ETC communication card in slot 2 [182](#)
- Permissible EtherCAT cycle times [35](#)
- Pin assignment [34](#)
- Process data words (PZD) for Servo Drives 9400 HighLine [35](#)
- Product codes for Inverter Drives 8400 [25](#)
- Product codes for Servo Drives 9400 [24](#)
- Product codes for the ECS servo system [25](#)
- Product codes for the I/O system 1000 [25](#)
- PROFIBUS [91](#)
- Properties of function blocks [98](#)

R

Remote API error codes (0x98110181 ...
0x98110196hex) [163](#)

Representation in the online mode [133](#)

Required engineering tools [28](#)

Required hardware components [22](#)

ResetMasterStatus (FB) [114](#)

Resetting the master structure [121](#)

Runtime of the actual values [35](#)

Runtime of the setpoints [35](#)

S

Safety [13](#)

Safety instructions (representation) [12](#)

scanfd (command line tool) [51](#)

SDO abort codes [164](#)

Set DC synchronisation at the field devices [39](#)

Setting and resetting the master structure [121](#)

Setting up a DC synchronisation [60](#)

Setting up the EtherCAT configuration in the device
tree [57](#)

SM_Ethercat.lib [96](#)

SMC_ETCErrorString (FUN) [115](#)

State machine [19](#)

State machine of the Lenze control technology [30](#)

Structure of the EtherCAT bus system [17](#)

Structure of the messages in the logbook [142](#)

Synchronisation [34](#)

Synchronisation with "distributed clocks" [36](#)

Synchronous communication [37](#)

System error messages [157](#)

T

Target group [9](#)

Technical data [34](#)

Technical data of MC-ETC [34](#)

Terms [11](#)

Total signal runtime for a cycle time of 1ms [35](#)

U

Usability of the EtherCAT function libraries [96](#)

User data per frame [35](#)

V

Variable wState [137](#)

VISU_ECATCHDiagnostic [136](#)

VISU_ECATCHDiagnostic visualisation template [136](#)

VISU_ETHERCATMaster visualisation template [135](#)

W

Working counter [21](#)



© 01/2011

Lenze Automation GmbH
Grünstraße 36
D-40667 Meerbusch
Germany



+49 (0)21 32 / 99 04-0



+49 (0)21 32 / 7 21 90



Lenze@Lenze.de



www.Lenze.com

Service Lenze Service GmbH
Breslauer Straße 3
D-32699 Extertal
Germany



00 80 00 / 24 4 68 77 (24 h helpline)



+49 (0)51 54 / 82-11 12



Service@Lenze.de

KHBETPCBAUTO ■ 13369406 ■ EN 3.1 ■ TD17

10 9 8 7 6 5 4 3 2 1