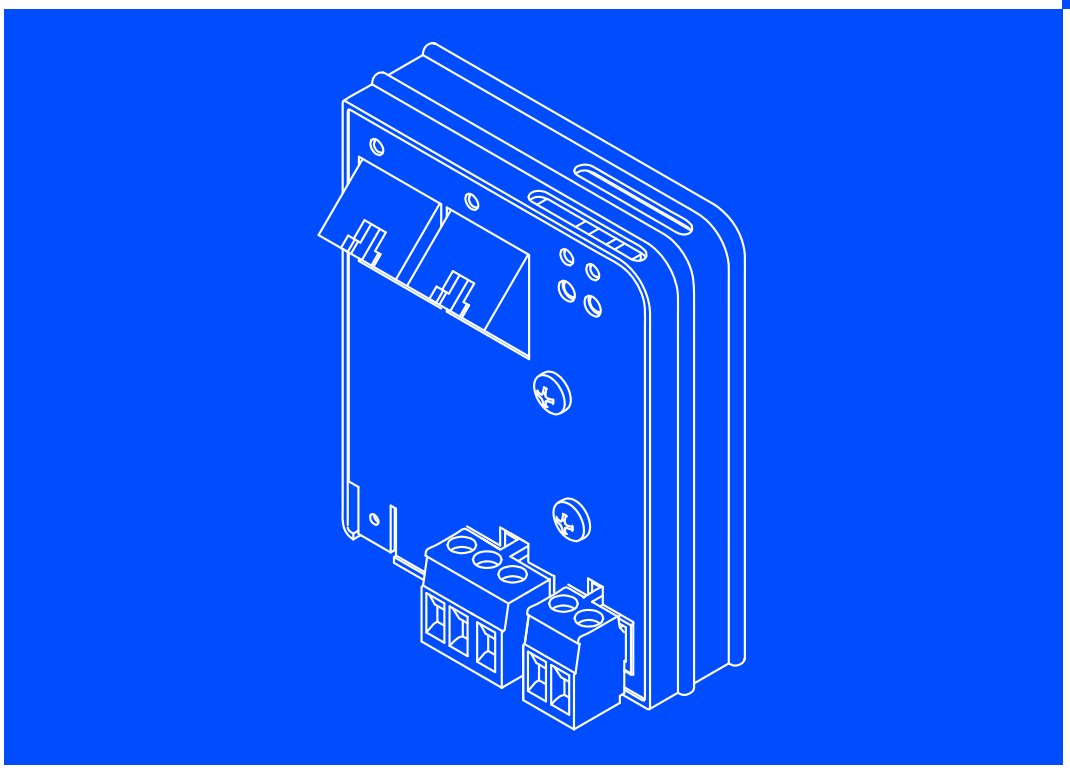
EDSMF2192IB
13508452

L-force Communication

Communication Manual

EtherCAT®



EMF2192IB

Communication module

Lenze

i Contents

1 About this documentation

Contents

This documentation only contains descriptions for the EMF2192IB communication module (EtherCAT).

Note!

This documentation supplements the **mounting instructions** supplied with the function/communication module and the **documentation of the used standard device**.

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.

Furthermore this documentation contains the following:

- Safety instructions that must be observed.
- Key technical data relating to the communication module
- Information on versions of Lenze standard devices to be used.
- Notes on troubleshooting and fault elimination

The theoretical correlations are only explained in so far as they are necessary for comprehending the function of the communication module.

This documentation does not describe the software of an original equipment manufacturer. No responsibility is taken for corresponding information given in this manual. Information on how to use the software can be obtained from the documents of the host system (master).

All brand names mentioned in this manual are trademarks of their respective companies.

Tip!

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

http://www.EtherCAT.org

Target group

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

Tip!

Information and tools concerning the Lenze products can be found in the download area at

www.lenze.com

Validity information

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

Extension module	Type designation	From hardware version upwards	From software version upwards
EtherCAT communication module	EMF2192IB	VA	1.0

1.1 Document history

Version			Description
4.0	03/2016	TD17	• Information for EAC supplemented. • General update and correction
3.0	06/2011	TD17	EtherCAT® is a registered trademark by Beckhoff Automation GmbH, Germany.
3.0	06/2011	TD17	General revision
2.0	09/2010	TD14	General revision
1.0	10/2009	TD17	First edition

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	Identification	Examples/notes
Spelling of numbers		
Decimal separator	Point	In general, the decimal point is used. For instance: 1234.56
Decimal	Standard notation	Example: 1234
Hexadecimal	0x[0 ... 9, A ... F]	Example: 0x60F4
Binary ● Nibble	0b[0, 1] Point	Example: '0b0110' Example: '0b0110.0100'
Text		
Program name	» «	PC software For example: »Engineer«, »Global Drive Control« (GDC)
Icons		
Page reference	🕮	Reference to another page with additional information For instance: 🕮 16 = see page 16
Documentation reference	🌍	Reference to another documentation with additional information For example: 🌍 EDKxxx = see documentation EDKxxx

1.3 Terminology used

Term	Meaning
EtherCAT®	EtherCAT® is a real-time capable Ethernet system with top performance. EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.
Standard device	Lenze controllers with which the communication module can be used. 📖 12
Controller	
»Global Drive Control« / »GDC«	Lenze PC software which supports you in ”engineering” (parameterisation, diagnostics and configuration) throughout the whole life cycle, i.e. from planning to maintenance of the commissioned machine.
»Engineer«	
»PLC Designer«	
»TwinCAT«	EtherCAT configuration software by Beckhoff Automation GmbH, Germany
Code	Parameter used for controller parameterisation or monitoring. The term is usually called ”index”.
Subcode	If a code contains several parameters, the individual parameters are stored under ”subcodes”. This manual uses a slash ”/” as a separator between code and subcode (e.g. ”C118/3”). The term is usually called ”subindex”.
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-1600.20	CoE index (hexadecimal representation) ● In the example: Index 0x1600, subindex 0x20
DC	”Distributed clocks” for EtherCAT synchronisation
PDO	Process data object
SDO	Service data object
”Hot connect”	This feature enables the slave nodes to be coupled/decoupled during operation.

1.4 Notes used

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

Safety instructions

Structure of safety instructions:

Danger!
(characterises the type and severity of danger)
Note
(describes the danger and gives information about how to prevent dangerous situations)

Pictograph and 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 and signal word	Meaning
 **Note!**	Important note to ensure troublefree operation
 **Tip!**	Useful tip for simple 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 information

Danger!

Disregarding the following basic safety measures may lead to severe personal injury and damage to material assets!

- Lenze drive and automation components ...
  - ... must only be used for the intended purpose.
  - ... must never be operated if damaged.
  - ... must never be subjected to technical modifications.
  - ... must never be operated unless completely assembled.
  - ... must never be operated without the covers/guards.
  - ... can - depending on their degree of protection - have live, movable or rotating parts during or after operation. Surfaces can be hot.
- For Lenze drive components ...
  - ... only use permitted accessories.
  - ... only use original manufacturer spare parts.
- All specifications of the corresponding enclosed documentation must be observed.

  This is vital for safe and trouble-free operation and for achieving the specified product features.

  The procedural notes and circuit details provided in this document are proposals which the user must check for suitability for his application. The manufacturer does not accept any liability for the suitability of the specified procedures and circuit proposals.
- Only qualified skilled personnel are permitted to work with or on Lenze drive and automation components.

  According to IEC 60364 or CENELEC HD 384, these are persons ...

  ... who are familiar with the installation, assembly, commissioning and operation of the product,

  ... possess the appropriate qualifications for their work,

  ... and are acquainted with and can apply all the accident prevent regulations, 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 in accordance with EN 61800-5-1 in every control cabinet (SELV/PELV).
- Only use cables that meet the given specifications. (📖 25)

Documentation of the standard device, control system, and 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

Protection of persons

- If inverters are connected to phase-earthed system with a rated mains voltage ≥ 400 V, external measures need to be implemented to provide reliable protection against accidental contact. (see chapter ”4.2”, 📖 17)

Device protection

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

3 Product description

3.1 Application as directed

The communication module ...

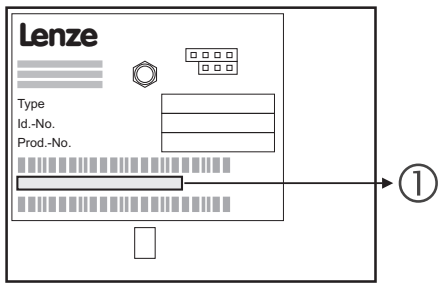
- ► is a device intended for use in industrial power systems.
- ► is only to be used in EtherCAT networks.
- ► can be used in connection with the following standard devices (nameplate data):

Device type	Design	Version		Variant	Explanation
		HW	SW [1)]		
82EVxxxxxBxxxXX		≥ Vx	≥ 1x		8200 vector
82CVxxxxxBxxxXX		≥ Vx	≥ 1x		8200 vector, cold plate
82DVxxxKxBxxxXX		≥ Vx	≥ 1x		8200 vector, thermally separated
EPL 10200	E	≥ 1x	≥ 1x		Drive PLC
33.93xx	xE.	≥ 2x	≥ 1x	Vxxx	9321 - 9332 vector
33.938X	xE.	≥ 1x	≥ 0x		9381 - 9383 vector
33.93xx	xC.	≥ 2x	≥ 1x	Vxxx	9321 - 9332, vector with cold plate design
33.93xx	EI / ET	≥ 2x	≥ 1x	Vxxx	9300 Servo PLC
33.93xx	CI / CT	≥ 2x	≥ 1x	Vxxx	9300 Servo PLC, cold plate
ECSxSxxxx4xxxxXX		≥ 1A	≥ 7.0		ECSxS "Speed & Torque"
ECSxPxxxx4xxxxXX		≥ 1A	≥ 7.0		ECSxP "Posi & Shaft"
ECSxMxxxx4xxxxXX		≥ 1A	≥ 7.0		ECSxM "Motion"
ECSxAxxxx4xxxxXX		≥ 1A	≥ 7.0		ECSxA "Application"
ECSxExxxx4xxxxXX		≥ VA	≥ 5.0		ECSxE power supply module

1) Operating system software versions of the controllers

Any other use shall be deemed inappropriate!

3.2 Identification



Type code	① →	33.2192IB	VA	1.0
Device series				
Hardware version				
Software version				

3.3 Product features

- Interface module for the EtherCAT communication system to the AIF slots of the Lenze device series 8200 vector, 9300, and ECS (📖 12)
- Support of the EtherCAT slave functionality
- External 24V supply for maintaining the EtherCAT network if the standard device fails
- Support of the ”Distributed clocks” (DC) functionality for synchronisation via the fieldbus
- PDO transfer with CoE (CANopen over EtherCAT)
- Access to all Lenze parameters with CoE (CANopen over EtherCAT)

Front panel connector elements

- Two sockets (RJ45) for the connection to EtherCAT
- 2-pin plug connector for the external supply of the communication module
- 3-pin plug connector (electrically isolated) for the synchronisation of the standard device

Front panel LED status displays

- Voltage supply of the communication module
- Connection from the communication module to the EtherCAT bus system
- Connection from the communication module to the standard device
- Bus state according to EtherCAT specification

3.4 Connections and interfaces

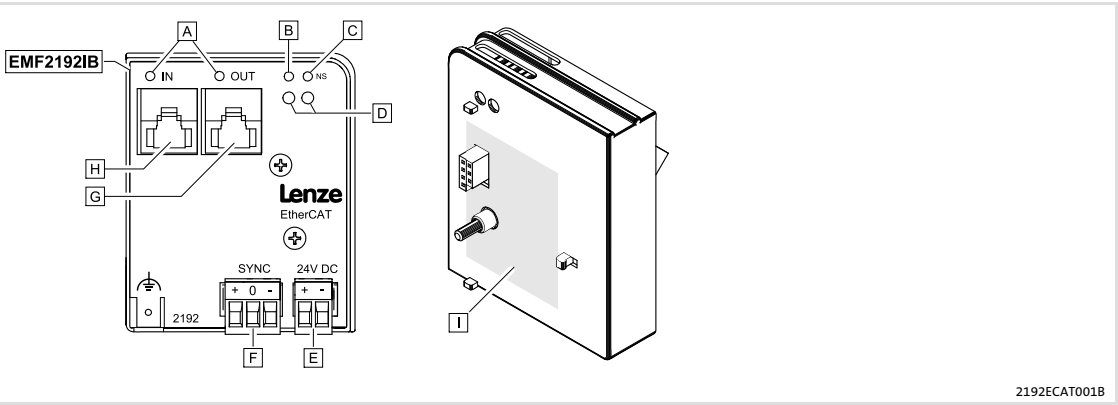


Fig. 3-1 EMF2192IB (EtherCAT) communication module

Legend for the illustration	
Pos.	**Description**
A ... D	LED status displays 📖 53
E	Connection to the external voltage supply (24 V) of the communication module ● Plug connector with screw connection, 2-pin 📖 27
F	Connection to EtherCAT synchronisation ● Plug connector with screw connection, 3-pin 📖 29
G	EtherCAT output (OUT) ● RJ45 socket in accordance with IEC 60603-7 📖 24
H	EtherCAT input (IN) ● RJ45 socket in accordance with IEC 60603-7 📖 24
I	Nameplate 📖 13

4 Technical data

4.1 General data and operating conditions

Area	Values
Order designation	EMF2192IB
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 (acc. to IEEE 802.3), 100Base-TX (Fast Ethernet)
Network topology	Line, switch
Node type	EtherCAT slave
Number of nodes	max. 65535 (in the entire network)
Cable length between two EtherCAT nodes	max. 100 m (typical)
Cycle times	1 ms or an integer multiple of 1 ms, max. 15 ms when ”Distributed clocks” (DC) are used
Vendor-ID	0x3B
Product-ID	depending on the standard device used
Revision-ID	depending on the main software version of the EtherCAT module
Baud rate	100 Mbps, full duplex
Voltage supply	External supply via separate external power supply unit ● Terminal ”+”: U = 24 V DC (20.4 V - 0 % ... 28.8 V + 0 %) I = 140 mA ● Terminal ”-”: Reference potential for external voltage supply
Conformities, approvals	CE

EAC	TP TC 020/2011 (TR CU 020/2011)	Electromagnetic compatibility of technical means	Eurasian Conformity TR CU: Technical Regulation of Customs Union
EAC	TP TC 004/2011 (TR CU 004/2011)	On safety of low voltage equipment	Eurasian Conformity TR CU: Technical Regulation of Customs Union

Documentation for Lenze series of devices 8200 vector, 9300 and ECS

Here you can find the **ambient conditions** and the **electromagnetic compatibility (EMC)** specifications applying to the communication module.

4.2 Protective insulation

Danger!

Dangerous electrical voltage

If Lenze controllers are used on a phase earthed mains with a rated mains voltage ≥ 400 V, protection against accidental contact is not ensured without implementing external measures.

Possible consequences:

► Death or serious injury

Protective measures:

► If protection against accidental contact is required for the control terminals of the controller 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.

Insulation between bus and ...	Type of insulation (in accordance with EN 61800-5-1)
● Earth reference / PE	Functional insulation
● With external supply	Functional insulation
● Power stage	
– 8200 vector	Reinforced insulation
– 9300 servo inverter	Reinforced insulation
– 93xx servo position controller	Reinforced insulation
– 93xx servo register control	Reinforced insulation
– 93xx servo cam profiler	Reinforced insulation
– 9300 vector / Servo PLC	Reinforced insulation
– ECS devices	Reinforced insulation
● Control terminals	
– 8200 vector	Functional insulation
– 9300 servo inverter	Basic insulation
– 93xx servo position controller	Basic insulation
– 93xx servo register control	Basic insulation
– 93xx servo cam profiler	Basic insulation
– 9300 vector / Servo PLC	Basic insulation
– ECS devices	Basic insulation

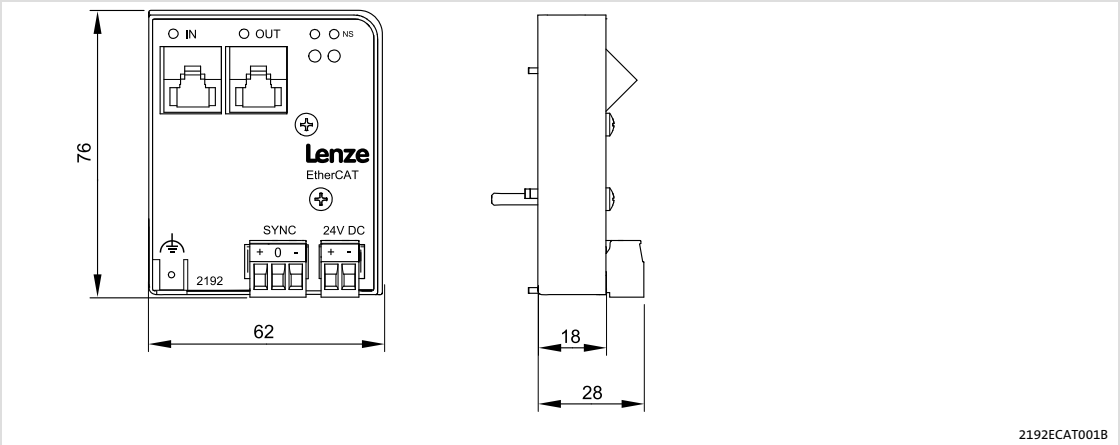
4.3 Communication time

Processing times in the controller

The parameter data and process data are independent of each other.

Processing times	Parameter data	Process data
Processing time within the controller	● Approx. 30 ms + a tolerance of 20 ms for parameters within the controller ● In the case of some codes the processing time can be longer. (See documentation for the controller) ● In the case of ECS devices the processing time depends on the application loaded (duration of the system task).	● Time for 8200 vector: approx. 3 ms + tolerance of 2 ms ● Time for devices of the 9300 series: approx. 2 ms + tolerance of 1 ms (depending on the basic cycle time in each case) ● Time for ECS devices: – In synchronous operation a minimum of 1 ms (AIF communication) or according to the fastest task – Otherwise 1 ms + task cycle time ● A synchronisation depends on the controller used (61) and has to be configured accordingly (33 et seqq.).
Additional times outside the controller	● Communication transmission times ● Communication processing times of the transmitting node	

4.4 Dimensions



All dimensions in mm

5 Installation

Danger!

Inappropriate handling of the communication module and the standard device can cause serious personal injury and material damage.

Observe the safety instructions and residual hazards described in the documentation for the standard device.

Stop!

The device contains components that can be destroyed by electrostatic discharge!

Before working on the device, the personnel must ensure that they are free of electrostatic charge by using appropriate measures.

5.1 Mechanical installation

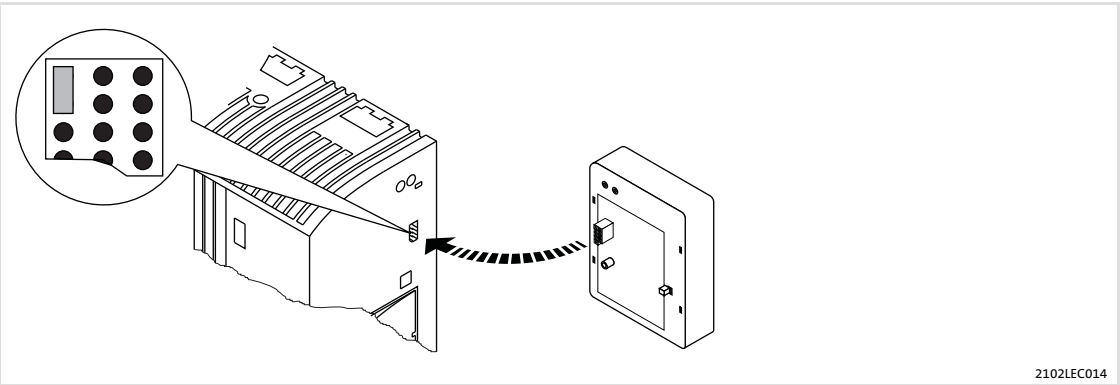


Fig. 5-1 Attaching the communication module

- ► Plug the communication module onto the standard device (here: 8200 vector).
- ► Tighten the communication module to the standard device using the fixing screw in order to ensure a good PE connection.

Note!

For the internal supply of the communication module by the 8200 vector frequency inverter the jumper has to be adjusted within the interface opening (see illustration above).

Observe the notes (📖 27).

5.2 Electrical installation

5.2.1 Wiring according to EMC (CE-typical drive system)

For wiring according to EMC requirements observe the following points:

Note!

- ► Separate control cables/data lines from motor cables.
- ► Connect the shields of control cables/data lines *at both ends* in the case of digital signals.
- ► Use an equalizing conductor with a cross-section of at least 16 mm^2 (reference: PE) to avoid potential differences between the bus nodes.
- ► Observe the other notes concerning EMC-compliant wiring given in the documentation for the standard device.

Wiring procedure

1. Comply with bus topology, thus do not use stubs.
2. Observe notes and wiring instructions in the documents for the control system.
3. Only use cables that comply with the given specifications (📖 25).
4. Observe notes for the voltage supply of the module (📖 27).

5.2.2 Network topology

An EtherCAT frame is transmitted by a pair of conductors from the master to the slaves. The frame is forwarded from slave to slave until it has passed through all devices. Finally the last slave sends the frame back to the master by a second pair of conductors. Thus, EtherCAT always creates a logic ring topology, irrespective of the topology selected.

Line topology

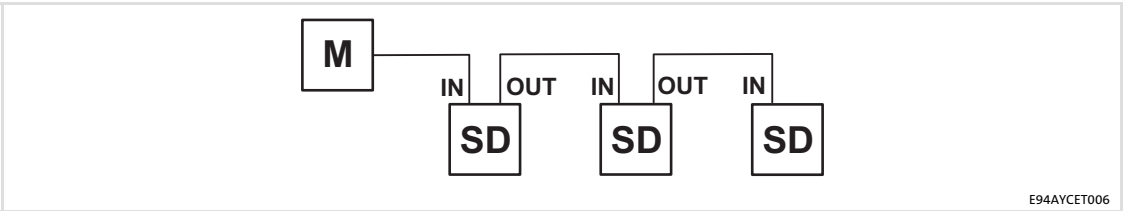


Fig. 5-2 Line topology

M Master
SD Slave Device

- The devices are interconnected successively.
- For correct operation it is necessary that the Ethernet sockets IN and OUT are assigned correctly.
  Plug the incoming cable into the IN socket and the ongoing cable into the OUT socket.
- The direction of data transmission is from the master to the slaves.

Tip!

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

Switch topology

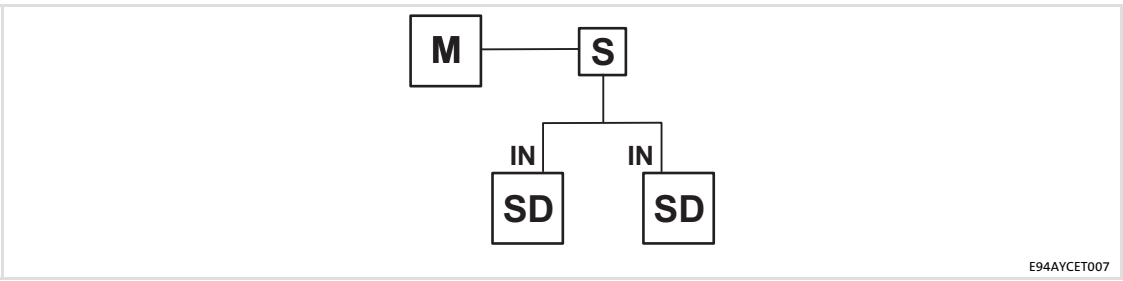


Fig. 5-3 Switch topology

M Master
S Switch
SD Slave Device

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

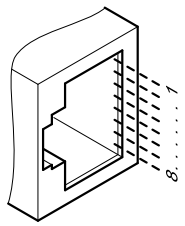
5.2.3 EtherCAT connection

You can use a standard Ethernet patch cable for connecting the communication module to the fieldbus (see ”Ethernet cable specifications” (📖 25)).

Note!

Plug/remove the Ethernet cable plug in a straight manner (at right angles) into/from the socket to make sure that the RJ45 socket will not be damaged.

Pin assignment

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

Tip!

The EtherCAT interfaces are equipped with an auto-MDIX function. This function adapts the polarity of the RJ45 interfaces such that independently of the polarity of the opposite EtherCAT interface and the cable type used (standard patch cable or crossover cable) a connection is established.

5.2.4 Specification of the Ethernet cable

Note!
Only use cables complying with the below specifications.

Ethernet cable specifications	
Ethernet standard	Standard Ethernet (according 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 (at 100 MHz and per 100 m)
Crosstalk damping	24 dB (at 100 MHz and per 100 m)
Return loss	10 dB (per 100 m)
Surge impedance	100 Ω

Design of the Ethernet cable

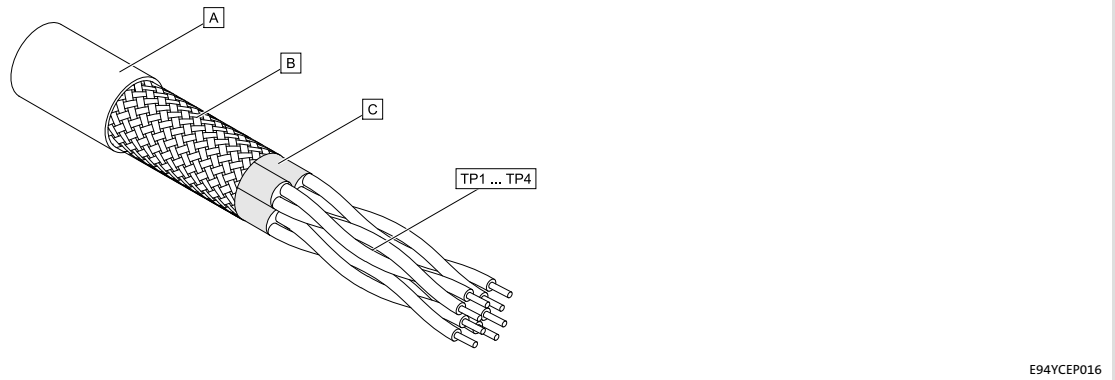


Fig. 5-4 Design of the Ethernet cable (S/FTP, CAT 5e)

A	Cable insulation
B	Braid
C	Foil shielding of the core pairs
TP1 ... TP4	Twisted core pairs 1 ... 4

Colour code of Ethernet cable

Note!

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

You can use 4-pin Ethernet cables in accordance with the industrial standard. The cable type only connects the assigned pins 1, 2, 3 and 6 with each other.

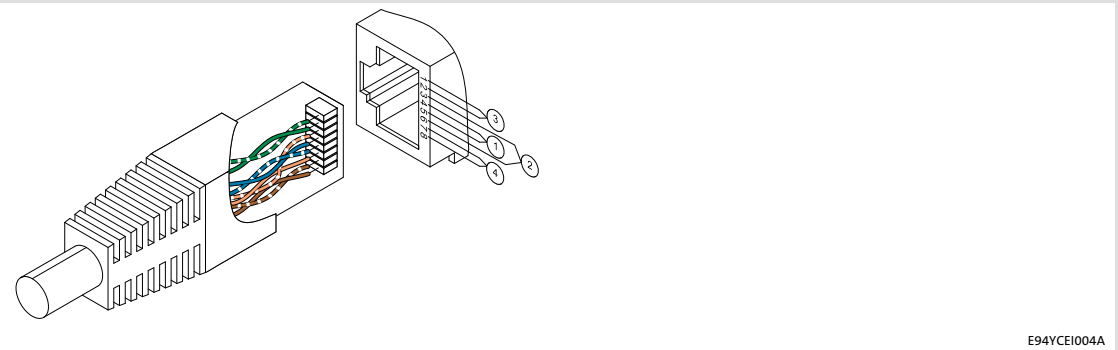


Fig. 5-5 Ethernet plug in accordance with EIA/TIA 568A/568B

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	Not assigned	Blue	Blue
	5	Not assigned	White/blue	Blue/white
2	6	Rx -	Orange	Green
4	7	Not assigned	White/brown	White/brown
	8	Not assigned	Brown	Brown

5.2.5 Voltage supply

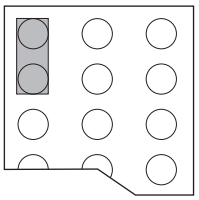
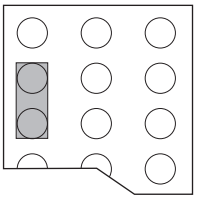
Internal voltage supply

Note!

Internal voltage supply has been selected in the case of standard devices with an extended AIF interface opening (e.g. front of 8200 vector). The area shown on a grey background in the graphic marks the jumper position.

- ► By default, this is **not** supplied internally in the standard device.
- ► For internal voltage supply place the jumper on the position indicated below.

In the case of all other device series (9300, ECS), voltage is always supplied from the standard device.

Lenze setting (Only external voltage supply possible.)	**Internal voltage supply**
	

External voltage supply

Note!

In the case of an external voltage supply and for greater distances between the control cabinets, always use a separate power supply unit (SELV/PELV) that is safely separated in accordance with EN 61800-5-1 in each control cabinet.

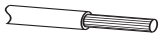
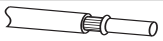
The external voltage supply of the communication module ...

- ► is required if communication via the fieldbus is to be continued when the supply of the device fails.
- ► is provided via the 2-pin terminal strip with screw-type connection (24 V DC):

Terminal	Description
+	External voltage supply U = 24 V DC (20.4 V - 0 % ... 28.8 V + 0 %) I = 85 mA
-	Reference potential for external voltage supply

- ► The parameters of a standard device disconnected from the mains cannot be accessed.

Terminal data

Area	Values	
Electrical connection	Plug connector with screw connection	
Possible connections	rigid:	
		1.5 mm² (AWG 16)
	flexible:	
		without wire end ferrule 1.5 mm² (AWG 16)
		with wire end ferrule, without plastic sleeve 1.5 mm² (AWG 16)
		with wire end ferrule, with plastic sleeve 1.5 mm² (AWG 16)
Tightening torque	0.5 ... 0.6 Nm (4.4 ... 5.3 lb-in)	
Stripping length	6 mm	

5.2.6 Synchronisation of the standard device

The synchronisation of the standard device via the EtherCAT fieldbus – if it is supported – can be carried out via the 3-pin plug connector with screw connection (sync).

Note!

ECS servo system

- ► For the ECS axis modules, a synchronisation with operating system software version ≥ 8.3 is possible.
- ► For the ECS power supply module a synchronisation is not supported.

Wire ...

- ► terminal ”0” to the corresponding sync input of the standard device (see documentation of the standard device).
- ► the sync supply to the 24V supply of the communication module or the standard device.

Terminal	Description
+	External sync supply (SELV/PELV) U = 24 V DC (20.4 V - 0 % ... 28.8 V + 0 %)
0	Sync output (t = 150 µs, I_{max} = 10 mA at 24 V)
-	Reference potential for external sync supply

On 🕮 61 you’ll find an overview of the Lenze standard devices which support a synchronisation.

6 Commissioning

During commissioning, system-dependent data as e.g. motor parameters, operating parameters, responses and parameters for fieldbus communication are selected for the controller.

In Lenze devices, this is done via codes. The codes are stored in numerically ascending order in the Lenze controllers and in the plugged-in communication/function modules.

In addition to these configuration codes, there are codes for diagnosing and monitoring the bus devices.

6.1 Before switching on

Stop!

Before switching on the standard device with the communication module for the first time, check the entire wiring for completeness, short circuit and earth fault.

6.2 Configuring the host system (master)

For communication with the communication module, first the host system (master) must be configured.

For configuring EtherCAT networks. a configuration software is always required for the host system (master), e.g.:

- Lenze »PLC Designer«
- »TwinCAT« by the company Beckhoff

These are software systems for programming 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 detection.
- During the search for nodes (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 **Lenze_AIF-Vxzz-ddmmyy.xml**required for the EMF2192IB (EtherCAT) communication module can be found on the in the download area under:

http://www.Lenze.com

Wildcard in the file name "Lenze_AIF-Vxzz-ddmmyy.xml"	
x	Major version of the XML device description file used
zz	Minor version of the XML device description file used
dd	Day
mm	Month
yy	Year

6.2.2 Automatic device detection

- For an error-free integration of the EtherCAT slaves into a master configuration it is required to select the correct Lenze device in the EtherCAT configuration software.
- An EtherCAT node is clearly identified via the configuration software by the product code (identical with the CoE object I-1018.2), the manufacturer's identification mark (0x3B) and the main software version of the communication module.
- In order that the configuration software selects the configuration from the device description file specific to the Ether-CAT nodes, the product code is automatically set in the identity object and updated after switch-on or each application download.
- During the initialisation, the product code is transmitted to the EtherCAT master. With this identification, the master can adopt the corresponding settings from the device description.

6.2.3 Configuring process data

- The process data configuration is defined during the initialisation phase of the master (PDO mapping).
- The process data configuration predefined application-specifically in the device description files and can be adapted by the user if required.

6.2.4 Defining the cycle time

The process data objects (PDO) are transmitted cyclically between the EtherCAT master and the slaves (controllers). The cycle time is set using the EtherCAT configuration software.

6.2.5 Address allocation

Usually, the EtherCAT nodes are addressed via a permanent 16-bit address defined by the EtherCAT master. At the start, this address is assigned to each node by the master depending on the physical sequence in the EtherCAT network. The address is not saved and gets lost after the device is switched off.

Additionally there is the possibility of allocating a fixed station alias address (chapter 6.2.6).

6.2.6 Specifying the station alias

By means of the station alias, a permanent address is assigned to the EtherCAT slave.

- For this, carry out the setting via the 0x58C5 object or code C1850 > 0.
- In addition, specify the use of the fixed addressing on the master.

Note!

- The station alias must only be set if the node is a member of a ”hot connect” group.
- The station alias must be non-ambiguous and may only be assigned once in the EtherCAT network.
- Use the same station alias in the EtherCAT master and the slave.

6.3 Synchronisation with ”Distributed clocks” (DC)

The ”Distributed clocks” (DC) functionality 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, the so-called ”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.
  - – In order to be able to use the DC functionality, the first slave connected to the EtherCAT master (e.g. L-force Controller) 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 after the EtherCAT master must be the **DC master** which provides the exact time to the other EtherCAT nodes (incl. EtherCAT master).

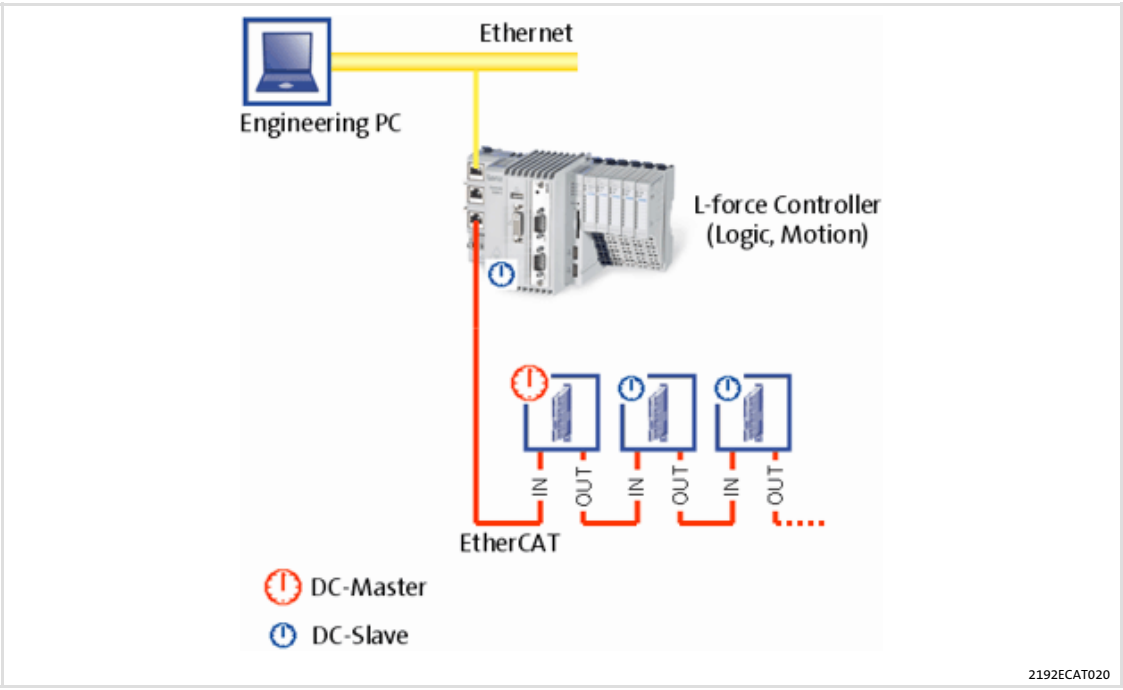


Fig. 6-1 Distributed clocks (DC)

The settings for DC synchronisation are made using the EtherCAT configuration software. (📖 31).

” EtherCAT control technology” communication manual

Here you’ll find detailed information on the EtherCAT configuration and commissioning of Lenze devices in the EtherCAT network.

6.3.1 Preparation / installation

Note!

Lenze devices without an external sync terminal input do not support DC synchronisation.

Connect the sync terminal block of the EtherCAT communication module to the voltage supply and the corresponding input terminal of the standard device (see 📖 29 and the documentation for the standard device).

6.3.2 DC configuration in the master

- The use of the DC synchronisation is deactivated in the device description (📖 31) by default.
- Parameterise the DC synchronisation in the EtherCAT configuration software.
- Specify the synchronisation cycle time in the master. It significantly complies with the processing time of the master and slaves.

6.3.3 DC configuration in the standard device (slave)

- In order to be able to use the DC synchronisation via EtherCAT in the standard device, select the ”AIF” sync source with standard device code C1120.
- Set the cycle time of the standard device in milliseconds with code C1121.
- Depending on the standard device, it may be required to also select a corresponding operating mode and control interface for EtherCAT communication via code. Information on this can be found in the documentation for the standard device.

6.3.4 Behaviour of the Lenze EtherCAT nodes during start-up

If the DC synchronisation is used, the communication module first changes to the ”Operational” state if the standard device has adapted its phase position to the DC signal. This process can take several seconds.

Note!

- If the communication module does not change to ”Operational”, there possibly is an error in the configuration or the EtherCAT wiring.
- The communication module compares the cycle time specified by the EtherCAT master to the cycle time of the standard device set in C1121. The synchronisation cycle time in the master has to be equal to the cycle time of the standard device.
- Moreover it is checked whether the selection of the sync source in the standard device code C01120 is correct.
- Further information can be gathered from the master as status information or an emergency message.

6.4 Initial switch-on

Switch on the inverter and check whether it is ready for operation using the diagnostic LEDs at the front of the communication module.

► Red diagnostic LEDs must not be on.

► The following signalling should be visible:

LED			Description
Pos.	Colour	Status	
A	green	blinking	
		on	 ● The EtherCAT connection has been established. ● Data communication of the EtherCAT connection is active.
B	green	on	 The communication module is supplied with voltage and is connected to the standard device.
C	green	The EtherCAT state machine controls the LED.	
		blinking	 ”Pre-operational” or ”Safe-operational” state active.
		on	 The communication module is in the ”Operational” status.

7 Data transfer

With EtherCAT, data is transmitted in ”EtherCAT frames”. The EtherCAT nodes only take the data determined for them while the EtherCAT frame passes through the device. Output data are entered in the frame the same way during the passage. Read and write accesses are only executed in a small section of the total EtherCAT frame, the datagrams. Thus, a frame does not need to be received completely before being processed. Processing starts as early as possible.

EtherCAT transmits process data, parameter data, configuration data and diagnostic data between the host system (master) and the controllers connected to the fieldbus (slaves). The data is transmitted via corresponding communication channels depending on their time-critical behaviour
(see chapter ”Process data transfer” (📖 41) and chapter ”Parameter data transfer” (📖 42).

7.1 EtherCAT frame structure

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 EtherCAT frame (EtherType = 0x88A4)

Ethernet data

The Ethernet data contain the following information:

- Length of the datagrams within the EtherCAT frame (length)
- A reserved bit (reserved)
- Type of datagrams within the EtherCAT frames (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 via EtherCAT is possible, the fieldbus passes through the EtherCAT status machine during power-up. The following illustration shows the possible state changes from an EtherCAT slave view:

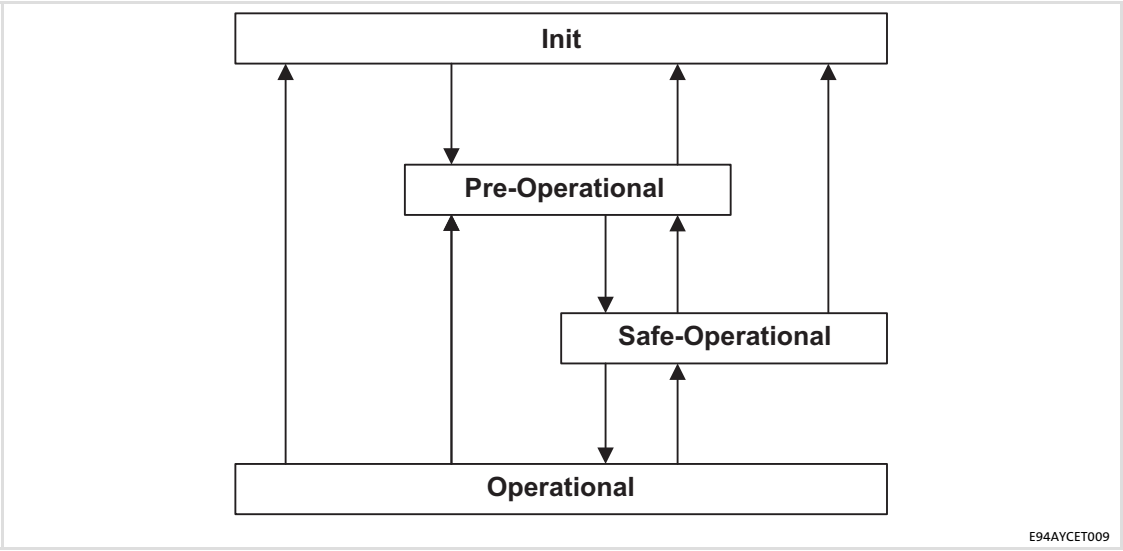


Fig. 7-1 EtherCAT state machine

Status	Description
Init	● Initialisation phase ● No SDO/PDO communication with the slaves ● The device detection is provided by a fieldbus scan.
Pre-operational	● The fieldbus is active. ● The SDO communication (mailbox communication) is possible. ● No PDO communication
Safe-operational	● The SDO communication (mailbox communication) is possible. ● PDO communication: – The input data is transmitted to the master and evaluated. – The output data is the ”Safe” status. They will not be transmitted to the standard device.
Operational	● Normal operation: – SDO communication – PDO communication – Fieldbus synchronisation successful (if used)

8 Process data transfer

- The process data are transmitted by means of ”EtherCAT datagrams” (📖 39) via the CoE process data channel.
- By means of the process data the controller is operated.
- The transmission of process data is time-critical.
- Process data are transmitted cyclically between the host system (master) and the controllers (slaves) (permanent 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.
- Process data are not stored in the controller.
- Process data, for instance, are setpoints, actual values, control words and status words.

9 Parameter data transfer

Parameter data are transmitted via the fieldbus as SDOs (Service Data Objects). The SDO services allow for the writing and reading access to the object directory.

- Via the SDO channel, access to all implemented CoE objects (📖 55) and Lenze codes (📖 57) is enabled with the CoE protocol.
- When a ”CiA402” technology application is used in the controller, the access to all implemented CANopen CiA402 objects is enabled.
- The transmission of parameter data usually is not time-critical.
- Parameter data for instance are operating parameters, diagnostic information, motor data.

9.1 Connection establishment between master and slave

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

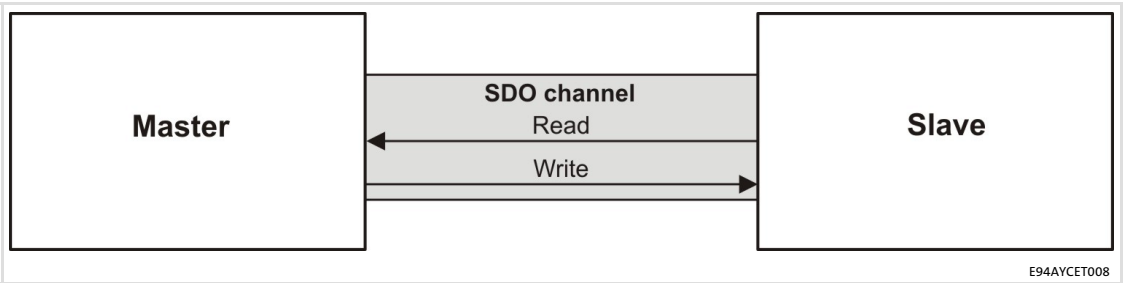


Fig. 9-1 Data communication via the SDO channel

9.2 Reading and writing parameters

Parameters ...

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

In the case of Lenze controllers, 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 controller are addressed by the index.

The index for Lenze codes is settled in the manufacturer-specific range of the object directory between 8192 (0x2000) and 24575 (0x5FFF).

The index number for a code results as follows:

Conversion formula	
Index (dec)	**Index (hex)**
24575 - Lenze code	$0x5FFF - (Lenze\ code)_{hex}$

Example for C0001 (operating mode):

Index (dec)	Index (hex)
24575 - 1 = 24574	0x5FFF - 0x0001 = 0x5FFE

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

9.2.1 Reading parameters (expedited upload)

1. The master transmits ”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 controller, please make sure that you convert the code into an index (📖 43).

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	0x0: 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.
	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” takes place if the data length of the parameter data to be read amounts to 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 ”data set size”
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes data 0x01: 3 bytes data 0x02: 2 bytes data 0x03: 1 byte data
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is read. 0x01: The complete object is read.
	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” takes place if the data length of the parameter data to be read amounts to ≥ 4 bytes.

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

SDO frame area	Data field	Data type / length		Value [hex] / description
Mailbox Header	Length	WORD	2 bytes	n ≥ 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.
	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 transmitted response structure in case of an **upload** to the index 0x5FD8 (standard value of C00039/1 = 0x0FA0) contains the following data:

SDO frame area	Data field	Data type / length		Value [hex] / 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 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

9.2.2 Writing parameters (expedited download)

1. The master transmits ”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 controller, please make sure that you convert the code into an index (📖 43).

SDO Download Expedited Request

A ”SDO Download Expedited Request” takes place if the data length of the parameter data to be written amounts to 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 ”data set size”
	Transfer type		1 bit (1)	0x01: Expedited transfer
	Data set size		2 bits (2, 3)	0x00: 4 bytes data 0x01: 3 bytes data 0x02: 2 bytes data 0x03: 1 byte data
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is written. 0x01: The complete object is written.
	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” takes place if the data length of the parameter data to be written amounts to ≥ 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	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)	0x0
	Complete access		1 bit (4)	0x00: The entry addressed with index and subindex is written. 0x01: The complete object is written.
	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 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)	0x00
	Transfer type		1 bit (1)	0x00
	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.
	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 transmitted request structure in case of a **download** from the index 0x1600 contains the following data:

SDO frame area	Data field	Data type / length		Value [hex] / description
Mailbox Header	Length	WORD	2 bytes	0xA: Length of the mailbox service data
	Address	WORD	2 bytes	0x0
	Channel	WORD	6 bits (0 ... 5)	0x0: Reserved
	Priority		2 bits (6, 7)	0x0: Lowest priority
	Type		4 bits (8 ... 11)	0x3: CANopen over EtherCAT (CoE)
	Reserved		4 bits (12 ... 15)	0x0
CANopen Header	Number	WORD	9 bits (0 ...8)	0x0
	Reserved		3 bits (9... 11)	0x0
	Service		4 bits (12 ... 15)	0x2: 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 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

9.3 SDO 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
0x05040002	The data block length is too great.
0x05040005	The space in the main memory is not sufficient.
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	An 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	Incorrect data type (The parameter length is too big)
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 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	Data cannot be transferred to or saved in the application because of current device state.
0x08000023	Dynamic object directory generation has failed or no object directory available.

10 Diagnostics

The LEDs on the front are provided to the communication module for the purpose of fault diagnostics.

10.1 LED status displays

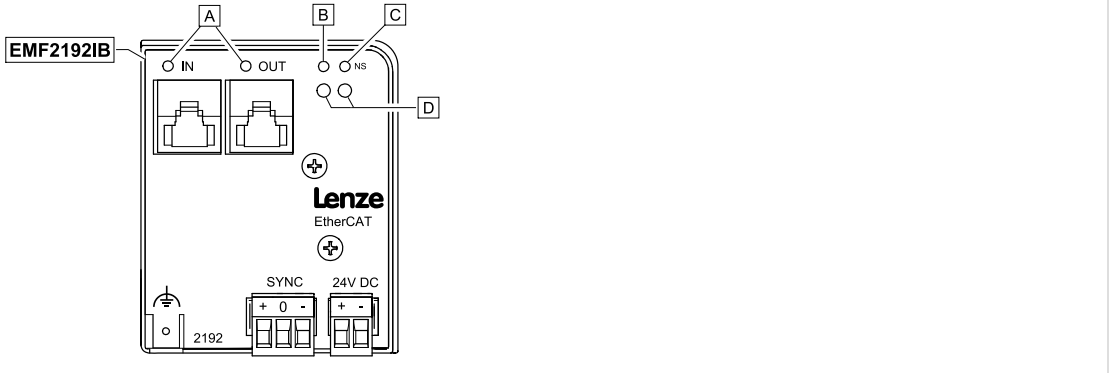


Fig. 10-1 LEDs of the communication module

LED			Description
Pos.	Colour	Status	
A	green	blinking	
		on	● The EtherCAT connection has been established. ● Data communication of the EtherCAT connection is active.
B	green	off	The communication module is not supplied with voltage.
		blinking	The communication module is supplied with voltage, but has no connection to the standard device (standard device is switched off, in the initialisation phase, or not available.)
		on	The communication module is supplied with voltage and is connected to the standard device.
C	green	The EtherCAT state machine controls the two-colored LED (red/green): ● Status messages are shown in green. ● Error messages are shown in red.	
		off	The communication module is not active on the fieldbus or is in the ”Init” status.
		blinking	”Pre-operational” or ”Safe-operational” state active.
		on	The communication module is in the ”Operational” status.
	red	on	An error has occurred in the communication module.
D	red		The red and green drive LED indicates the operating status of the standard device (see operating instructions of the standard device).
	green		

10.2 Emergency requests / emergency messages

Emergency messages are sent once to the EtherCAT master if the error status of the controller/communication module changes, i.e. ...

- if an error of the controller/communication module occurs;
- if an error of the controller/communication module is omitted.

An **"Emergency Request"** on the fieldbus consists of the "Mailbox Header", "CANopen Header" and the Emergency message:

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

Structure of the emergency message

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 (Device)			
Low Byte	High Byte	Low Byte	High Byte	Low Word		High Word	
				Low Byte	High Byte	Low Byte	High Byte

Example: The AIF connection to the standard device has been lost (error code "0x31").

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 (Device)			
0x00	**0x10**	**0x01**	**0x00**	**0x00**	**0x00**	**0x00**	**0x31**

- Bytes 1 and 2 indicate that an error has occurred.
- Byte 3 indicates the contents of the error register (I-1001).
- Bytes 5 ... 8 indicate the corresponding error code.

Possible error codes (overview)

No. (Byte 8)	Designation	Meaning
0x10	EMCY_BAD_SYNC_INPUT	The sync source specified in code C1120 of the standard device is incorrect.
0x11	EMCY_BAD_SYNC_CYCLETIME	The sync cycle time specified by the master cannot be used.
0x12	EMCY_BAD_SYNC_CYCLE_GG	The specified sync cycle time from code C1121 of the standard device cannot be used.
0x13	EMCY_CANT_SYNC	Synchronisation of the standard device is not possible.
0x14	EMCY_SYNC_LOST	EtherCAT has lost the synchronisation.
0x31	EMCY_AIF_LOST	The AIF connection to the standard device has been lost.
0x32	EMCY_AIF_UNKNOWN_GG	The standard device is unknown.

11 Appendix

11.1 Implemented CoE objects

Lenze devices can be parameterised with Lenze codes and with the manufacturer-independent ”CoE objects”. In order to obtain a complete EtherCAT-compliant communication, only the CoE objects may be used for parameterisation. The CoE objects described in this documentation are defined in the ”EtherCAT Specification, Part 6 – Application Layer Protocol Specification”.

Index	Index name	Subindex	Subindex name	Data type	Bits	Access
0x1000	Device type	-	-	UDINT	32	R
0x1008	Device name	-	-	STRING(30)	240	R
0x1009	Hardware version	-	-	STRING(2)	16	R
0x100A	Software version	-	-	STRING(30)	240	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	IO Outputs	0	Number of elements	USINT	8	RW
		1 … 32	Output Object 1 … 32	UDINT	32	RW
0x1800	IO Inputs	0	Number of elements	USINT	8	RW
		7	TxPDO-State	BOOL	1	R
		9	TxPDO-Toggle	BOOL	1	R
0x1A00	IO Inputs	0	Number of elements	USINT	8	RW
		1 … 12	Input Object 1 … 12	UDINT	32	RW
0x1C00	Sync Man Communication type	0	Number of elements	USINT	8	R
		1	Elements	UDINT	32	R
0x1C12	RxPDO Assignment	0	Number of elements	USINT	8	R
		1	Elements	UDINT	32	R
0x1C13	TxPDO Assignment	0	Number of elements	USINT	8	R
		1	Elements	UDINT	32	R
0x1C32	SM output parameter	0	Number of elements	USINT	8	RW
		1	Synchronization type	UINT	16	RW
		2	Cycle time / ns	UDINT	32	RW
		3	Shift time / ns	UDINT	32	RW
		4	Sync types supported	UINT	16	R
		5	Minimum cycle time / ns	UDINT	32	R
		6	Minimum shift time / ns	UDINT	32	R
0x1C33	SM input parameter	0	Number of elements	USINT	8	RW
		1	Synchronization type	UINT	16	RW
		2	Cycle time / ns	UDINT	32	RW
		3	Shift time / ns	UDINT	32	RW
		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

11.2 Codes

The objects specified in the table can be accessed via EtherCAT fieldbus. The objects are implemented in the Lenze code structure. Writable codes are stored permanently and are maintained after the communication module is switched off.

Tip!

The codes are visible in the object directory of the EtherCAT configuration tool.

Object		Code	Subcode	Designation	Access	Information
Index [hex]	Subindex					
0x58ED	-	C1810	-	Software identification of the module	R	58
0x58EC		C1811	-	Software creation date	R	58
0x58E1	1 ... n	C1822	1 ... n	AIF input words (to the standard device)	R	58
0x58E0	1 ... n	C1823	1 ... n	AIF output words (from the standard device)	R	58
0x58DF	1 ... n	C1824	1 ... n	AIF input double words (to the standard device)	R	58
0x58DE	1 ... n	C1825	1 ... n	AIF output double words (from the standard device)	R	59
0x58D9	-	C1830	-	Bus status	R	59
0x58C5	-	C1850	-	Station alias address	RW	59
0x58A5	-	C1882	-	Response when exiting "Operational"	RW	59
0x58A4	-	C1883	-	Monitoring time when exiting "Operational"	RW	60

R: Read access only
RW: Read and write access

Code description

Parameter	Name	Data type: STRING (30)
C1810	**Software identification of the module**	Index: $22765_{dec} = 58ED_{hex}$

The software ID of the communication module is shown here.
Display: ”33S2192I_xy000” (xy = version x.y)

Parameter	Name	Data type: STRING (30)
C1811	**Software creation date**	Index: $22764_{dec} = 58EC_{ex}$

The software creation date (”mm dd yyyy”) and the time (”hh:mm:ss”) are shown here.
Example: ”FEB 06 2008 09:23”

Parameter	Name	Data type: UINT16
C1822	**AIF input words (to the standard device)**	Index: $22753_{dec} = 58E1_{hex}$

Display of the process input data

Display area (min. value | unit | max. value)			
0x0000		0xFFFF	
Subcodes			**Information**
C1822/1			
...			
C1822/n			

Parameter	Name	Data type: UINT16
C1823	**AIF output words (from the standard device)**	Index: $22752_{dec} = 58E0_{hex}$

Display of the process output data

Display area (min. value | unit | max. value)			
0x0000		0xFFFF	
Subcodes			**Information**
C1823/1			
...			
C1823/n			

Parameter	Name	Data type: UINT32
C1824	**AIF input double words (to the standard device)**	Index: $22751_{dec} = 58DF_{hex}$

Display of the process input data

Display area (min. value | unit | max. value)			
0x00000000		0xFFFFFFFF	
Subcodes			**Information**
C1824/1			
...			
C1824/n			

Parameter	Name	Data type: UINT32
C1825	**AIF output double words (from the standard device)**	Index: $22750_{dec} = 58DE_{hex}$

Display of the process output data

Display area (min. value | unit | max. value)			
0x00000000		0xFFFFFFFF	
Subcodes			**Information**
C1825/1			
...			
C1825/n			

Parameter	Name	Data type: FIX32
C1830	**Bus status**	Index: $22745_{dec} = 58D9_{hex}$

Bit-coded display of the current bus status
📖 40

Display area (min. value | unit | max. value)			
0x00000000		0xFFFFFFFF	
Bits (read only)			**Information**
Bit 0	EtherCAT bus status in accordance with CiA DS301		0: Unknown 1: Init 2: Pre-operational 3: Bootstrap 4: Safe-operational 8: Operational
...			
Bit 7			
Bit 8	Not assigned		
...			
Bit 12			
Bit 13	SYNC_bProcessDataExpected		New process data were expected during the application cycle.
Bit 14	SYNC_bProcessDataInvalid		New/last process data are invalid. ● Checksum error ● No telegram received
Bit 15	EtherCAT error flag		

Parameter	Name	Data type: FIX32
C1850	**Station alias address**	Index: $22725_{dec} = 58C5_{hex}$

Specification of the station address if the master is used to address the alias
- 0: no alias address
- 65535: Alias address preselection in the slave

Setting range (min. value | unit | max. value)			**Lenze setting**
0		65535	**0**

Parameter	Name	Data type: FIX32
C1882	**Response when exiting "Operational"**	Index: $22693_{dec} = 58A5_{hex}$

Adjustable response for the process data monitoring

Selection list (Lenze setting printed in bold)		**Information**
0	**No response**	
1	Error (TRIP)	
2	Controller inhibit (CINH)	
3	Quick stop (QSP)	

Parameter	Name	Data type: FIX32
C1883	**Monitoring time when exiting "Operational"**	Index: $22692_{dec} = 58A4_{hex}$

If the "Operational" status is exited, the response parameterised with C1882 occurs after the time set here has elapsed.

- With the value = 65535 monitoring is deactivated.
- With the value = 0 the immediate response is effected after the internal bus status watchdog time has elapsed
- A change in monitoring is effective immediately.

Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	65535	**65535 ms**

11.3 Product codes of the Lenze standard devices

Product code [decimal]	Meaning	Sync support	Number Process data words	AIF status/ control word
21920000	Generic	-	-	-
21920100	8200 vector	-	3	-
21920101	8200 vector in combination with an application I/O function module	-	3	-
21920102	8200 vector in combination with a DeviceNet/CANopen function module	-	3	-
21920103	8200 vector in combination with an INTERBUS function module	-	3	-
21920104	8200 vector in combination with a LECOM-B function module	-	3	-
21920105	8200 vector in combination with a PROFIBUS I/O function module	-	3	-
21920106	8200 vector in combination with a PROFIBUS function module	-	3	-
21920107	8200 vector in combination with a CANopen function module	-	3	-
21920108	8200 vector in combination with a DeviceNet function module	-	3	-
21920200	9300 servo inverter	✓	4	-
21920202	93xx servo position controller	✓	4	-
21920204	93xx servo register control	✓	4	-
21920206	93xx servo cam profiler	✓	4	-
21920301	9300 hoist	-	4	-
21920400	9300 vector	-	4	-
21920500	9300 Servo PLC	✓	12	✓
21920600	Drive PLC	✓	12	✓
21920700	ECSxA axis module ”Application”	✓	12	✓
21920701	ECSxM axis module ”Motion”	✓	12	✓
21920702	ECSxP axis module ”Posi & Shaft”	✓	12	✓
21920703	ECSxS axis module ”Speed & Torque”	✓	12	✓
21920711	ECSxE power supply module	-	3	-

Note!

ECS servo system

From operating system software version 8.3 onwards, a synchronisation is possible for the ECS axis modules.

12 Index

Lenze Automation GmbH
Postfach 10 13 52, 31763 Hameln
Hans-Lenze-Str. 1, 31855 Aerzen
GERMANY
HR Hannover B 205381

+49 5154 82-0

+49 5154 82-2800

lenze@lenze.com

www.lenze.com

Service Lenze Service GmbH
Breslauer Straße 3, 32699 Extertal

GERMANY

008000 2446877 (24 h helpline)

+49 5154 82-1112

service@lenze.com

EDSMF2192IB ■ 13508452 ■ EN ■ 4.0 ■ TD17

10 9 8 7 6 5 4 3 2 1