



E84DGFCxxx

CANopen

Inverter Drives 8400 motec-----

Communication Manual

EN



13564905

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	16
4	Technical data	18
4.1	General data and operating conditions of the CANopen	18
4.2	Supported protocols	19
4.3	Communication time	20
5	Installation	21
5.1	Mechanical installation	22
5.2	Electrical installation	23
5.2.1	Network topology	23
5.2.2	Bus termination	24
5.2.3	Specification of the bus cable	25
5.2.4	Bus cable length	25
5.2.5	CANopen connection	28
6	Commissioning	29
6.1	Before initial switch-on	29
6.2	How to configure the host (master)	30
6.3	Possible settings via DIP switch	31
6.3.1	Setting the baud rate	31
6.3.2	Setting the CAN node address	32
6.4	Settings in the Lenze »Engineer«	33
6.5	Initial switch-on	34

Contents

7	Data transfer	35
7.1	Structure of the CAN data telegram	35
7.1.1	Identifier	36
7.1.2	User data	37
7.2	Communication phases/network management	37
7.2.1	State transitions	39
7.2.2	Network management telegram (NMT)	40
7.2.3	Parameterising the Inverter Drives 8400 motec as CAN master	41
8	Process data transfer	42
8.1	Accessing process data / PDO mapping	44
8.2	Port interconnection of process data objects (PDO)	45
8.3	Identifiers of the process data objects	49
8.4	Transmission type	50
8.5	PDO synchronisation via sync telegram	52
9	Parameter data transfer	53
9.1	Identifiers of the parameter data objects	54
9.2	User data	55
9.2.1	Command	55
9.2.2	Addressing by means of index and subindex	56
9.2.3	Data 1 ... data 4	57
9.2.4	Error messages	58
9.3	Parameter data telegram examples	60
9.3.1	Read parameters	60
9.3.2	Write parameters	61
9.3.3	Reading block parameters	62
10	Monitoring	65
10.1	Monitoring of the RPDOs for data reception	65
10.2	Integrated error detection	66
10.3	Heartbeat protocol	67
10.3.1	Telegram structure	67
10.3.2	Parameter setting	68
10.3.3	Commissioning example	70
10.4	Emergency telegram	71
11	Diagnostics	72

Contents

12	Parameter reference	73
12.1	Communication-relevant parameters of the operating system	73
12.2	Parameters relevant for CANopen communication	74
12.3	Table of attributes	85
13	Implemented CANopen objects	87
14	DIP switch positions for setting the CAN node address	102
15	Index	104

1 About this documentation

Contents

This documentation exclusively describes the system bus (CAN) and the CANopen-specific functions of 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 features of the system bus (CAN) and CANopen-specific functions for the Inverter Drive 8400 motec are described in detail.

Examples illustrate typical applications.

This documentation contains as well ...

- the most important technical data for CAN communication;
- information on the installation and commissioning of the CAN network;
- information on CAN data transfer, CAN monitoring functions, communication-relevant parameters and implemented CAN objects.

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

Depending on the software version of the controller and the version of the »Engineer« software installed, the screenshots in this documentation may deviate from the »Engineer« representation.

This documentation does not describe the software of other manufacturers. No responsibility is taken for corresponding information given in this documentation. Information on how to use the software can be obtained from the documents of the master computer.

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



Tip!

Detailed information about the system bus (CAN) can be found on the website of the CAN user organisation CiA (CAN in Automation):

www.can-cia.org

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 CANopen communication unit	E84DGFCCxNx	CANopen
	E84DGFCCxJx	CANopen + safety

► [Features and variants](#) (13)

About this documentation

Document history

1.1 Document history

Version			Description
4.0	02/2019	TD23	General revision
3.0	11/2011	TD17	General revision
2.0	01/2011	TD17	Update of the ... • Parameters relevant for CANopen communication (📖 73) (version 02.00) • »Engineer« screenshots
1.0	09/2010	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«
Control element	Bold	The OK button... / The Copy command... / The Properties tab... / The Name input field...
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
CAN	CAN (Controller Area Network) is an asynchronous, serial fieldbus system.
CANopen®	CANopen® is a CAN-based communication protocol and a registered trademark, licensed by the CiA® (CAN in Automation e. V.) CAN user organisation, www.can-cia.org .
Inverter	Lenze frequency inverter 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.
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



Note!

Some of the terms used originate from the CANopen protocol.

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](#) (📖 12)
 - 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](#) (📖 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.
- 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.
- In case of external voltage supply, each control cabinet must be provided with a safely separated power supply unit ("SELV"/"PELV") according to EN 61800-5-1.
- Only use cables that meet the listed specifications.
 - ▶ [Specification of the bus cable](#) (📖 24)



Documentation of "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](#) (📖 20)

3 Product description

3.1 Application as directed

The CANopen 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 CANopen networks.
- can also be used without being connected to the CANopen network.

Any other use shall be deemed inappropriate!

Product description

Features and variants

3.2 Features and variants

The CANopen communication unit is available in the following versions:

Product series	Type designation	Product features				
		Enclosure	Connection CANopen	I/O: Connection via terminal	I/O: Connection via M12	Safety
Inverter Drives 8400 motec CANopen communication unit	E84DGFCCFNP	IP 65	M12	3× DI 1× DO	2× DI	
	E84DGFCCENP	IP 65	M12	2× DI	3× DI 1× DO	
	E84DGFCCFJP	IP 65	M12	3× DI 1× DO 1× AI	2× DI	●
	E84DGFCEJP	IP 65	M12	3× DI	2× DI 1× DO 1× AI	●

- The CANopen communication unit ...
 - mounted on top of the Wiring Unit (E84DGVNxx);
 - is exclusively supplied internally by the drive unit (E84DGDVxxxxxxxx).
- The I/O connections can be brought into the device via M12 connectors or cable glands.
- In the E84DGFCC9xx version, a maximum of four digital inputs is conducted on M12 connectors (see "Inverter Drives 8400 motec" hardware manual).
- Devices without an integrated safety system (safety option) have no analog input and no relay output.
- In the case of the E84DGFCCxJx communication units, the integrated safety system can be used for the protection of persons on machines.
- Setting of the CAN node address and baud rate is possible via DIP switch or code.
- Communication with the Lenze »Engineer« (access to all Lenze parameters) is preferably carried out via the CAN bus. Furthermore communication can be effected via the diagnostic interface of the drive unit.



"Inverter Drives 8400 motec" hardware manual

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

Software 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

Features and variants

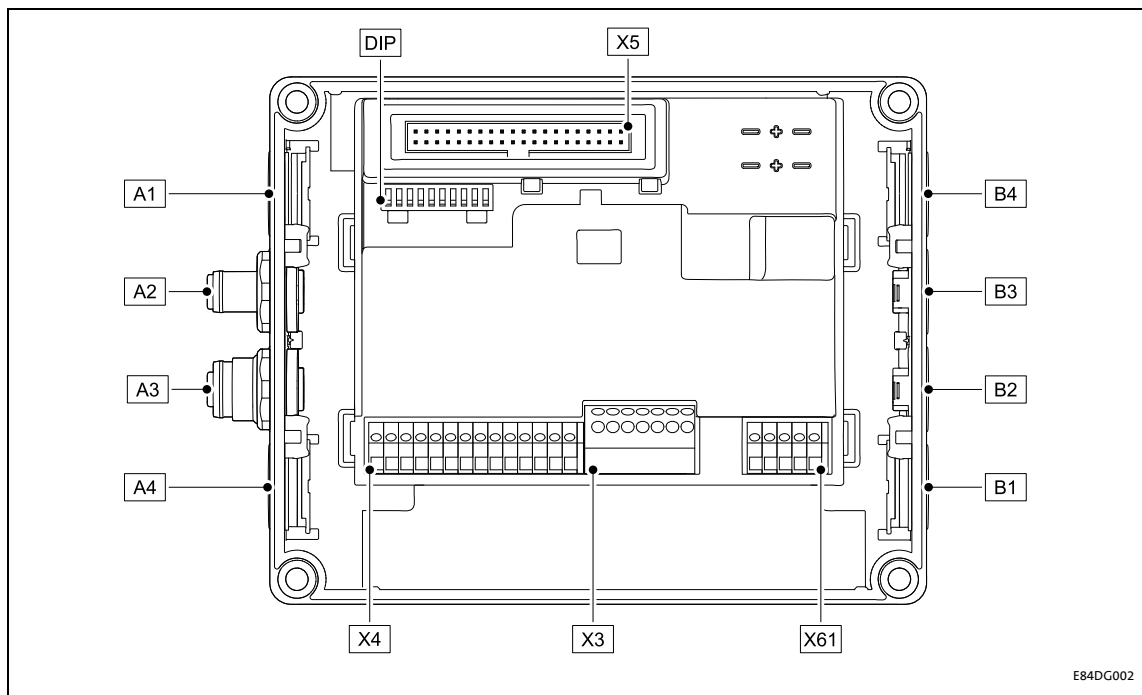
The system bus (CANopen) of the Inverter Drives 8400 motec is the advanced version of the system bus (CAN) and includes the following features:

- Full compatibility according to CANopen DS301, V4.02.
- Support of the "Heartbeat" NMT slave function (DS301, V4.02).
- Number of parameterisable server SDO channels:
 - Max. 2 channels with 1 ... 8 bytes
 - Because of the 2 server SDO channels, the address range from 1 ... 63 is available.
- Number of parameterisable PDO channels:
 - Max. 2 transmit PDOs (TPDOs) with 1 ... 8 bytes (adjustable)
 - Max. 2 receive PDOs (RPDOs) with 1 ... 8 bytes (adjustable)
- All PDO channels are functionally equivalent.
- Monitoring of the RPDOs for data reception
- Adjustable error response to ...
 - physical CAN errors (frame, bit, ACK error)
 - bus-stop, bus-working
 - missing PDOs
- Telegram counters for SDOs and PDOs
- Bus status diagnostics
- Boot-up telegram generation
- Emergency telegram generation
- Reset node telegram generation (in the case of master configuration)
- Sync telegram generation and response to sync telegrams:
 - Data transmission/reception
 - Device-internal time base synchronisation
- Abort codes
- Object directory (all mandatory functions, optional functions, indexes)

Product description

Connections and interfaces

3.3 Connections and interfaces



[3-1] CANopen communication unit

Pos.	Description
DIP	DIP switch ▶ Possible settings via DIP switch (□ 30)
A2	CANopen input (M12 pins, 5-pole) ▶ CANopen connection (□ 27)
A3	CANopen output (M12 socket, 5-pole) ▶ CANopen connection (□ 27)
A1 / A4 B1 ... B4	Positions for further freely designable inputs and outputs: • Digital inputs • Digital output • Analog input (only for E84DGFCCxJx) • Relay output (only for E84DGFCCxJx) • Connection of safety system "Safety Option" (only for E84DGFCCxJx)
X3 / X4 / X61	Terminal strips for wiring the connectors at A1 ... A4 and B1 ... B4
X5	Plug connector for connection to the Drive Unit

Product description

Connections and interfaces

- By default, the CANopen connectors are already pre-assembled and wired with the terminal strip X3.
- The CANopen connections and further connections (e.g. digital inputs) can be freely designed at the positions A1 ... A4 and B1 ... B4..
- 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 plugs or cable glands used with the corresponding contacts of the terminal strips X3, X4 and X61.



"Inverter Drives 8400 motec" hardware manual

Observe the notes and wiring instructions contained in this documentation.

Technical data

General data and operating conditions of the CANopen

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 of the CANopen

Range	Values
Order designation	• E84DGFCxNx (CANopen) • E84DGFCxJx (CANopen + Safety)
Communication profile	CANopen, DS301 V4.02
Communication medium	DIN ISO 11898
Interface	• CANopen input: M12 pins, 5-pole, A-coded • CANopen output: M12 socket, 5-pole, A-coded
Network topology	Line terminated on both sides
Adjustable node address	1 ... 63 (can be set via DIP switch or code C00350)
Max. number of nodes	63
Baud rate [kbps]	20, 50, 125, 250, 500, 800, 1000 kbps, adjustable via DIP switches or code C00351
Process data	• Max. 2 transmit PDOs (TPDOs) with 1 ... 8 bytes (adjustable) • Max. 2 receive PDOs (RPDOs) with 1 ... 8 bytes (adjustable)
Parameter data	Max. 2 server SDO channels with 1 ... 8 bytes
Transmission mode for TPDOs	• With change of data • Time-controlled, 1 to x ms • After the reception of 1 to 240 sync telegrams
Conformities, approvals	• CE • UR / cUR (see also hardware manual)

4.2 Supported protocols

Protocols	
Standard PDO protocols	PDO write PDO read
SDO protocols	SDO download SDO download initiate SDO download segment
	SDO upload SDO upload initiate SDO upload segment
	SDO abort transfer
	SDO block download SDO block download initiate SDO block download end
	SDO block upload SDO block upload initiate SDO block upload end
NMT protocols	Start remote node (master and slave)
	Stop remote node (slave)
	Enter pre-operational (slave)
	Reset node (slave and local device)
	Reset communication protocol (slave)
Monitoring protocols	Heartbeat (heartbeat producer and heartbeat consumer) • 1 Heartbeat Producer can be monitored.
	Emergency telegram (to master)

4.3 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 a CANopen network depend on ...

- processing time in the inverter;
- frame runtime (baud rate / frame length);
- nesting depth of the network.

Processing time inside the inverter

Data	Processing time	
Process data	Approx. 2 ms + 0 ... 1 ms + 1 ... x ms	Update cycle Processing time in the module Runtime of the application task of the technology application used (tolerance)
Parameter data	Approx. 30 ms + a tolerance of 20 ms (typically) • For some codes, the processing time may be longer (see software manual/»Engineer« online help "Inverter Drives 8400 motec").	

There are no interdependencies between parameter data and process data.

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

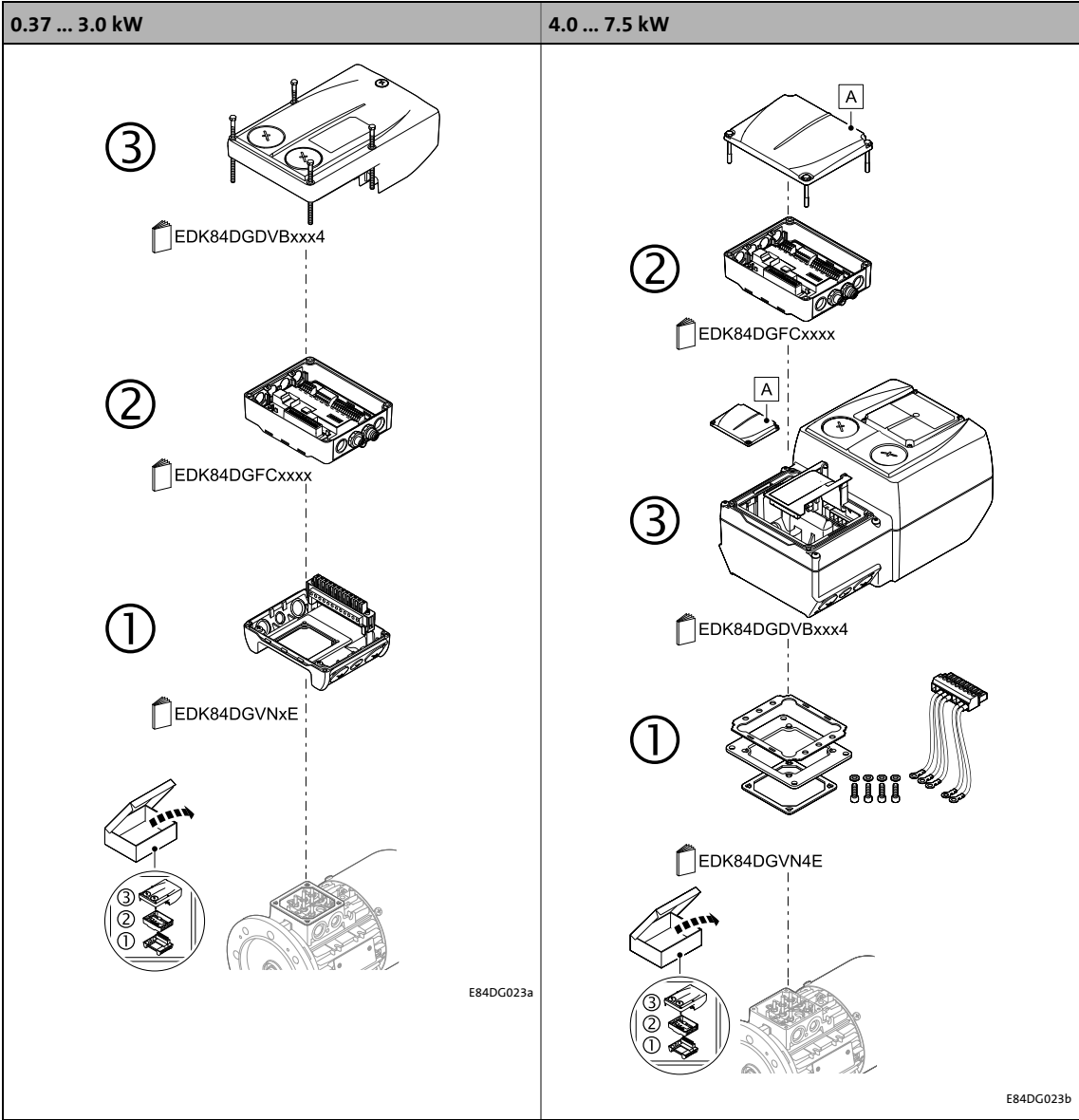
- Discharge electrostatic charges before touching the Communication Unit.

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.

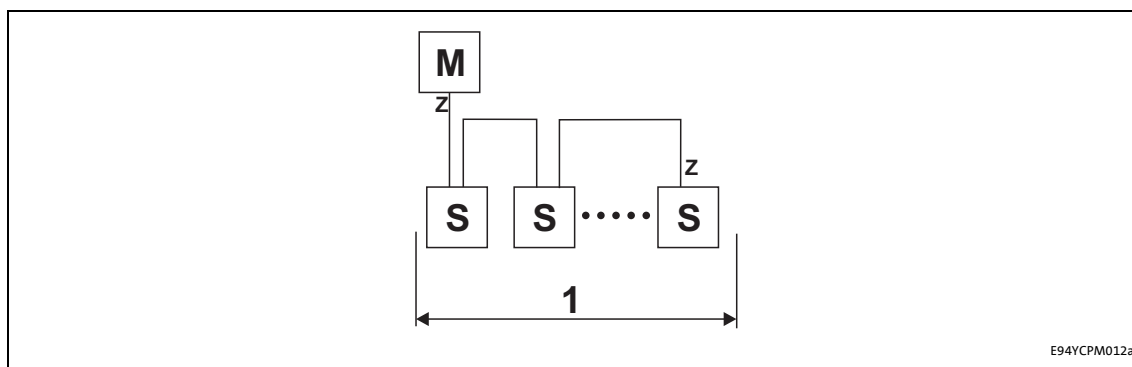
5.2.1 Network topology

The following examples show two simple CAN networks.

Each segment of the network must be terminated at both ends by resistors ($120\ \Omega$) between CAN-Low and CAN-High. The bus terminators of the system bus (CAN) are marked with a "Z" in the following examples.

A CAN network consisting of only one segment starts with the CAN master (M) with integrated bus termination, whereas the last CAN node (S) has to be terminated by a bus terminating resistor.

► [Bus termination](#) (□ 23)



E94YCPM012a

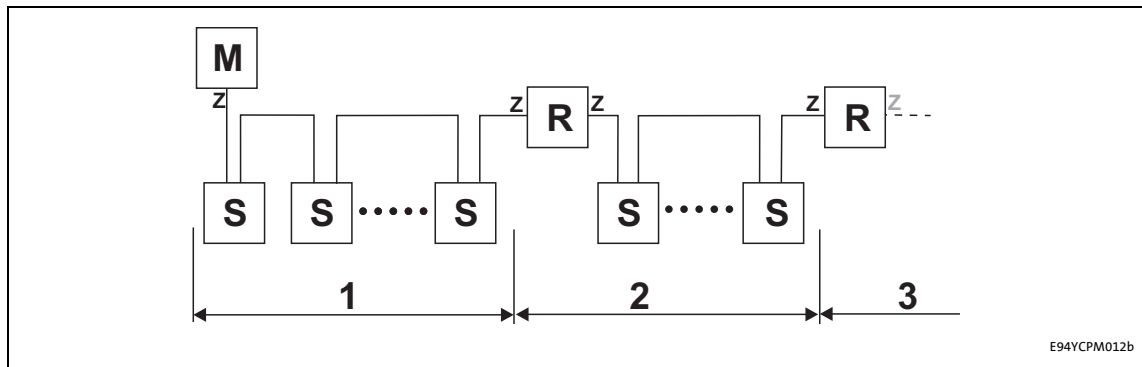
[5-2] CAN network with one segment

Installation

Electrical installation

A CAN network consisting of several segments contains repeaters (R) for connecting the segments. The repeaters are provided with integrated bus terminations.

► [Checking the use of repeaters](#) (25)



[5-3] CAN network with repeaters

If no repeater is to be used at the end of the segment, the bus must be terminated by means of a bus terminating resistor at the last station (S). The bus termination is supplied by this station.

5.2.2 Bus termination

The system bus (CANopen) must be terminated through a bus terminating resistor at the first and last physical node (120 Ω).

In the case of the communication unit, the bus terminating resistor can only be installed externally at the M12 connector. This has the advantage that an installed resistor is visible when the device is closed.



Note!

- The CANopen terminals (input and output) must be installed so that they are closed. For this purpose either use a connecting cable, a closed terminating resistor plug (M12 pins, 5-pole, A-coded), or a cap.
- The connecting cable and terminating resistor plug can be procured freely from various cable manufacturers (e.g. Lapp or Turck).
- If you want to disconnect individual bus stations, ensure that the bus terminators at the cable ends remain active. Otherwise, the bus may become unstable.
- Observe that the bus terminator is no longer active when the terminating resistor plug has been removed.

5.2.3 Specification of the bus cable

We recommend to use CAN cables in accordance with ISO 11898-2:

CAN cables according to ISO 11898-2	
Cable type	Twisted in pairs with shield
Impedance	120 Ω (95 ... 140 Ω)
Cable resistance/cross-section	
Cable length ≤ 300 m	≤ 70 m Ω /m / 0.25... 0.34 mm ² (AWG22)
Cable length 301 ... 1000 m	≤ 40 m Ω /m / 0.5 mm ² (AWG20)
Signal propagation delay	≤ 5 ns/m

Observe the notes provided on the [Bus cable length](#) (□ 24)!

5.2.4 Bus cable length



Note!

- It is absolutely necessary to comply with the permissible cable lengths.
- Observe the reduction of the total cable length due to the signal delay of the repeater.
 ▶ [Checking the use of repeaters](#) (□ 25)
- Mixed operation refers to different nodes being connected to the same network.
- If the total cable lengths of the nodes are different at the same baud rate, the smaller value must be used to determine the maximum cable length.

Total cable length

1. Check that the total cable length is not exceeded.

The total cable length is defined by the baud rate.

Baud rate [kbps]	Max. bus length [m]
20	4013
50	1575
125	600
250	275
500	113
800	38
1000	13

[5-1] Total cable length

Segment cable length

2. Check that the segment cable length is not exceeded

The segment cable length is defined by the used cable cross-section and by the number of nodes. Without repeaters the segment cable length corresponds to the total cable length.

Maximum number of nodes per segment	Cable cross-section			
	0.25 mm ²	0.5 mm ²	0.75 mm ²	1 mm ²
2	240 m	430 m	650 m	940 m
5	230 m	420 m	640 m	920 m
10	230 m	410 m	620 m	900 m
20	210 m	390 m	580 m	850 m
32	200 m	360 m	550 m	800 m
63	170 m	310 m	470 m	690 m

[5-2] Segment cable length

3. Compare both values.

If the value determined from the [Segment cable length \[5-2\]](#) table is smaller than the required total cable length [Total cable length \[5-1\]](#), repeaters must be used. Repeater divide the total cable length into segments.

Example: Selection help

Given	
• Cable cross-section:	0.5 mm ² , according to Specification of the bus cable (EN 24)
• Number of nodes:	63
• Repeater:	Lenze repeater, type 2176 (cable reduction: 30 m)

Based on the given specifications, the following cable lengths/number of repeaters result for a maximum of 63 nodes:

Baud rate [kbps]	20	50	125	250	500	800	1000
Max. cable length [m]	4013	1575	600	275	113	38	13
Segment cable length [m]	270	270	270	270	113	38	13
Number of repeaters	15	6	2	1	-	-	-

Checking the use of repeaters



Note!

The use of an additional repeater is recommended as:

- Service interface
 - Advantage: Trouble-free connecting during ongoing bus operation is possible.
- Calibration interface
 - Advantage: Calibration/programming units remain electrically isolated.

Installation

Electrical installation

Given	
• Baud rate:	125 kbps
• Cable cross-section:	0.5 mm ²
• Number of nodes:	28
• Cable length:	450 m

Test step		Cable length	See
1	Total cable length at 125 kbps:	600 m	Table Total cable length [5-1] (□ 24)
2	Segment cable length for 28 nodes and a cable cross-section of 0.5 mm ² :	360 m	Table Segment cable length [5-2] (□ 25)
3	Comparison: The value determined in step 2 is smaller than the required cable length of 450 m.		

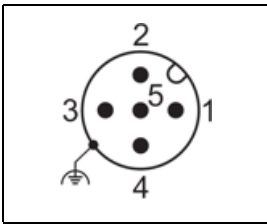
Conclusion:

- It is not possible to use a cable length of 450 m without using a repeater.
- After 360 m (test step 2) a repeater has to be used.

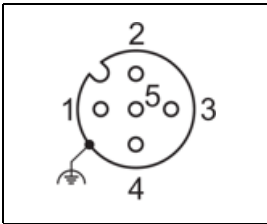
Result:

- The Lenze repeater, type 2176 (cable reduction: 30 m), is used
- Calculation of the maximum cable length:
 - First segment: 360 m
 - Second segment: 360 m (according to the table [Segment cable length \[5-2\]](#) (□ 25)) *minus* 30 m (cable reduction when a repeater is used)
- Max. achievable cable length with a repeater: 690 m
 - Now it is possible to use the required cable length.

5.2.5 CANopen connection



- Input: M12 pins, 5-pole, A-coded
- Wiring of terminal strip X3



- Output: M12 socket, 5-pole, A-coded
- Wiring of terminal strip X3

CANopen connection		
Pin	Signal	Description
1	-	Not assigned
2	-	Not assigned
3	CG	CAN GND potential
4	CH	CAN-High data line
5	CL	CAN-Low data line

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-volatilely as a data set in the memory module.

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

▶ [Parameter reference](#) (□ 72)

6.1 Before initial switch-on



Stop!

Before switching on the inverter for the first time, check ...

- the entire wiring for completeness, short circuit, and earth fault.
- whether the bus system is terminated by means of a bus terminating resistor at the first and last physical bus station.

▶ [Bus termination](#) (□ 23)

6.2 How to configure the host (master)

Communication with the inverter first requires configuration of the host (master).

Defining the user data length

- The CANopen communication unit supports the configuration of max. 8 process data words (max. 64 bytes).
- The user data length is defined during the initialisation phase of the master.
- The user data lengths for process input data and process output data are identical.



Note!

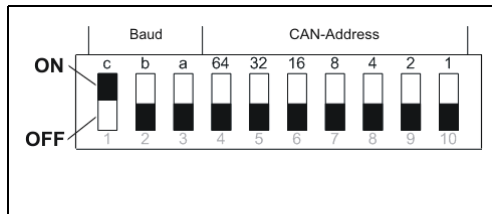
The CANopen process data objects are designated as seen from the node's view:

- Receive PDO (RPDOx): Process data object received by a node
- Transmit PDO (TPDOx): Process data object sent by a node

Commissioning

Possible settings via DIP switch

6.3 Possible settings via DIP switch



[6-1] DIP switch

The DIP switches serve to ...

- [Setting the baud rate](#) (📖 30) (switches: a ... c)
- [Setting the CAN node address](#) (📖 31) (switches: 1 ... 64)

Lenze setting: all switches in OFF position



Note!

- The DIP switches can only be accessed when the drive unit is detached from the communication unit. Loosen the four fixing screws at the drive unit. **Observe the notes in the mounting instructions.**
- Switch off the voltage supply of the inverter and the external supply of the communication unit before you start disassembling the drive unit.
- The DIP switches are only read in when the device is switched on.

6.3.1 Setting the baud rate

The baud rate ...

- must be the same for all networked CANopen nodes;
- can be set via the DIP switches a ... c or via the »Engineer« (code [C00351](#)).

DIP switch position			Baud rate
c	b	a	
ON	OFF	ON	20 kbps
OFF	ON	ON	50 kbps
OFF	ON	OFF	125 kbps
OFF	OFF	ON	250 kbps
OFF	OFF	OFF	500 kbps
ON	ON	OFF	800 kbps
ON	OFF	OFF	1000 kbps

► [Settings in the Lenze »Engineer«](#) (📖 32)

Commissioning

Possible settings via DIP switch

6.3.2 Setting the CAN node address

The node addresses must differ from each other in the case of several networked CANopen nodes.

The node address can be set via DIP switches 1 ... 64 or via the »Engineer« with code [C00350](#).

For the setting with [C00350](#) DIP switches 1 ... 64 must be set to OFF.



Note!

- The valid address range is 0 ... 63.
- If DIP switch 64 = ON (node address > 63), always node address 63 is used.

DIP switch							Node address
64	32	16	8	4	2	1	
OFF	OFF	OFF	OFF	OFF	OFF	OFF	Value from C00350
OFF	OFF	OFF	OFF	OFF	OFF	ON	1
OFF	...	...	...	...	...	...	...
OFF	ON	ON	ON	ON	ON	ON	63
ON	...	...	...	...	...	...	

The labelling on the housing corresponds to the values of the individual DIP switches for determining the node address.

DIP switch	64	32	16	8	4	2	1
Switch position	OFF	OFF	ON	OFF	ON	ON	ON
Value	0	0	16	0	4	2	1
Node address	= sum of the valencies = 16 + 4 + 2 + 1 = 23						

The current address setting of the DIP switches is displayed in [C00349](#).

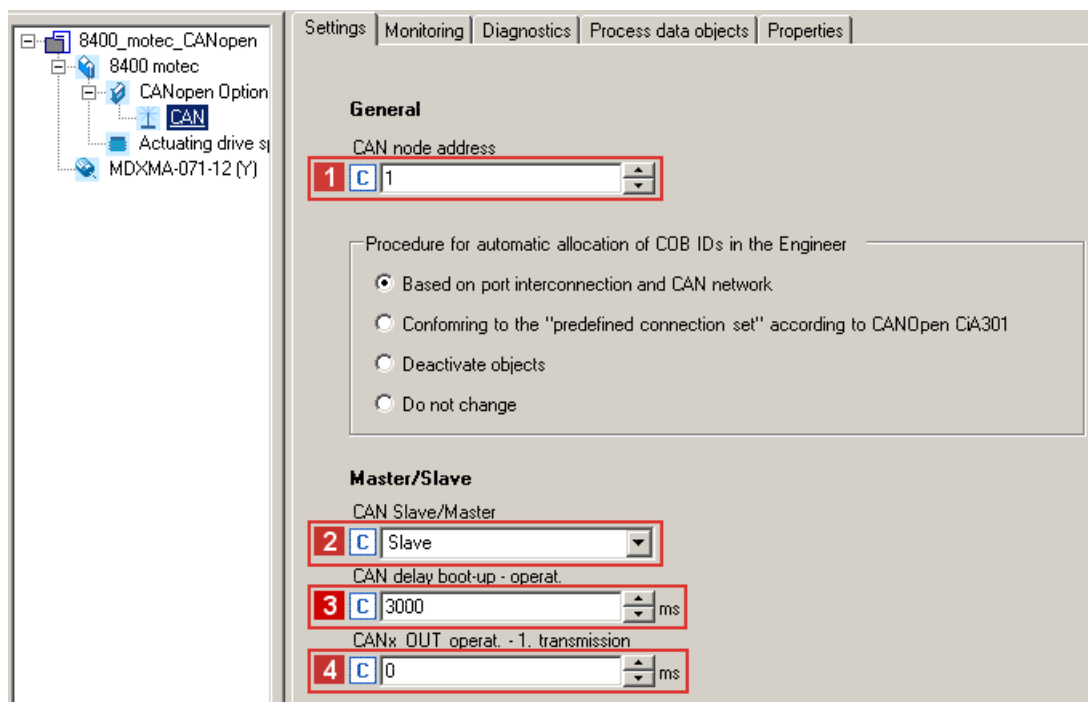
► [DIP switch positions for setting the CAN node address](#) (101)

► [Settings in the Lenze »Engineer«](#) (32)

6.4 Settings in the Lenze »Engineer«

The following settings can be made in the »Engineer« under the **Settings** tab:

- CAN node address **1** ([C00350](#))
 - The node address can only be parameterised if the node address "0" is set via the DIP switches.
 - A change of the node address will only become effective after a CAN reset node.
- CAN node is slave or master **2** ([C00352](#))
- Deceleration during status change from "Boot-up" to "Operational" **3** ([C00356/1](#))
- Time to the first transmission of CANx_OUT in the "Operational" state **4** ([C00356/4](#))



Save changed settings with the device command **C00002/11** (save all parameter sets).

6.5 Initial switch-on

Establishing communication

- To establish communication, the inverter drive must be supplied with mains voltage.
- CANopen 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 Drive, so that CANopen communication can be established.
- During mains connection, all parameters (codes) and the DIP switch settings are read.
- The positions of the DIP switches define whether the CAN node address and the baud rate are selected via the DIP switches or via codes [C00350](#) and [C00351](#).
 - ▶ [Possible settings via DIP switch](#) (📖 30)

Data transfer

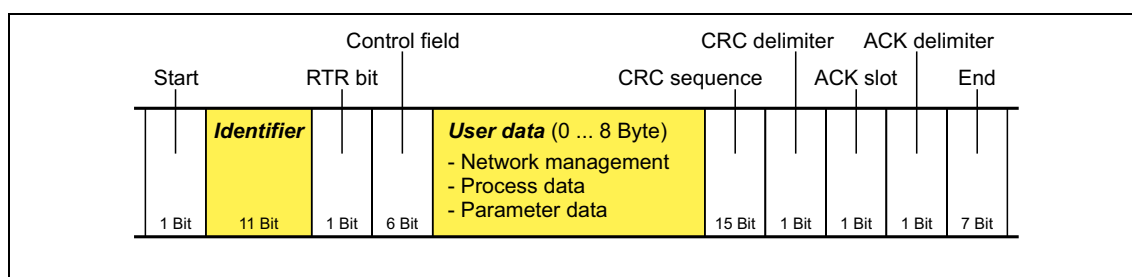
Structure of the CAN data telegram

7 Data transfer

Via the system bus interface, for instance process data and parameter values can be exchanged between the nodes. In addition, the interface enables the connection of additional modules such as distributed terminals, keypads and input devices or external control systems and hosts (masters).

The system bus interface transfers CAN objects following the CANopen communication profile (CiA DS301, version 4.02) developed by the umbrella organisation of **CiA** (CAN in Automation) in conformity with the **CAL** (CAN Application Layer).

7.1 Structure of the CAN data telegram



[7-1] Basic structure of the CAN telegram

The following subchapters provide a detailed description of the identifier and the user data. The other signals refer to the transfer characteristics of the CAN telegram the description of which is not included in the scope of this documentation.



Tip!

Please visit the homepage of the CAN user organisation CiA (CAN in automation) for further information:

<http://www.can-cia.org>

Data transfer

Structure of the CAN data telegram

7.1.1 Identifier

The principle of CAN communication is based on a message-oriented data exchange between a transmitter and many receivers. All nodes can virtually transmit and receive simultaneously.

The identifier, also called COB-ID (abbr. for communication object identifier), is used to control which node is to receive a transmitted message. In addition to the addressing, the identifier contains information on the priority of the message and the type of user data.

The identifier consists of a basic identifier and the node address of the node to be addressed:

Identifier (COB-ID) = basic identifier + node address (node ID)

Exception: The identifier for process data/heartbeat/emergency objects as well as network management and sync telegrams is freely assigned by the user (either manually or automatically by the network configurator), or is permanently assigned.

Node address (node ID)

For unambiguous identification, a node address (also called node ID) within the valid address range (1 ... 63) must be assigned to every node of the system bus network.

- A node address may not be assigned more than once within a network.
- The own node address can be configured via the DIP switches or via code [C00350](#).

► [Setting the CAN node address](#) (□ 31)

Identifier assignment

The system bus is message-oriented instead of node-oriented. Every message has an unambiguous identification, the identifier. For CANopen, node-oriented transfer is achieved by the fact that every message has only one transmitter.

- The basic identifiers for network management (NMT) and sync as well as the basic SDO channel (SDO1) are defined in the CANopen protocol and cannot be changed.
- In the Lenze setting, the basic identifiers of the PDOs are preset according to the "Predefined connection set" of DS301, V4.02 and can be changed via parameters/indexes, if required.

► [Identifiers of the process data objects](#) (□ 48)

Object		Direction		Lenze-Base-ID		CANopen-Base-ID	
		from device	to device	dec	Hex	dec	Hex
Network management (NMT)				0	0	0	0
Sync ¹⁾				128	80	128	80
Emergency ¹⁾		●		128	80	128	80
PDO1 (Process data channel 1)	TPDO1	●		384	180	384	180
	RPDO1		●	512	200	512	200
PDO2 (Process data channel 2)	TPDO2	●		640	280	640	280
	RPDO2		●	641	281	768	300
SDO1 (Parameter data channel 1)	TSDO1	●		1408	580	1408	580
	RSDO1		●	1536	600	1536	600
SDO2 (Parameter data channel 2)	TSDO2	●		1472	5C0	1472	5C0
	RSDO2		●	1600	640	1600	640
Heartbeat		●		1792	700	1792	700
Boot-up		●		1792	700	1792	700

1) If you set the sync transmit/receive identifier manually, observe the use of the emergency telegram, since it has the same COB-ID.

Data transfer

Communication phases/network management

7.1.2 User data

All nodes communicate by exchanging data telegrams via the system bus. The user data area of the CAN telegram either contains network management data, or parameter data, or process data:

Network management data

(NMT data)

- Control information on start, stop, reset, etc. of communication to specific nodes or to all nodes of the CAN network.

Process data

(PDOs – process data objects)

- Process data are transferred via the process data channel.
- You can control the inverter by means of the process data.
- Process data are not saved in the inverter.
- Process data are transmitted between the host (master) and the inverters (slaves), providing for a continuous exchange of current input and output data.
- Process data usually are unscaled/scalable raw data.
- Process data are, for instance, setpoints and actual values.
- The exact meaning of the PDO file contents is determined via the function block editor (FB Editor) in the I/O level or via the PDO mapping.

Parameter data

(SDOs – service data objects)

- Parameter data are the CANopen indexes or, in the case of Lenze devices, the codes.
- The parameters are, for instance, set for the initial system set-up during commissioning or when material is changed on the production machine.
- Parameter data are transmitted as SDOs via the parameter data channel. They are acknowledged by the receiver, i.e. the sender receives a feedback about the transmission being successful or not.
- The parameter data channel enables access to all Lenze codes and CANopen indexes.
- Parameter changes are saved automatically in the inverter until mains switching.
- Generally the parameter transfer is not time-critical.
- Parameter data are, for instance, operating parameters, motor data and diagnostic information.

7.2 Communication phases/network management

With regard to communication via the system bus, the inverter knows the following states:

Data transfer

Communication phases/network management

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

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

Code [C00359](#) serves to display the status of the CAN bus.



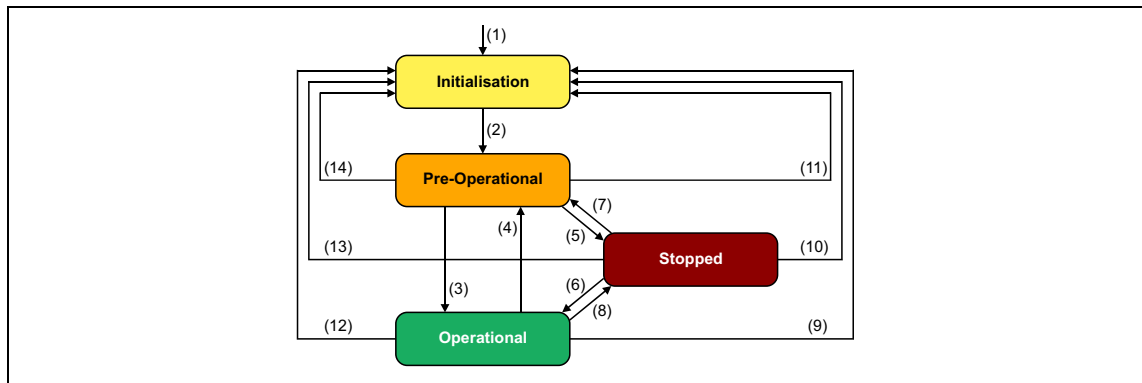
Tip!

Part of the initialisation or the entire initialisation can be carried out again in every status by transmitting the corresponding network management telegrams.

Data transfer

Communication phases/network management

7.2.1 State transitions



[7-2] NMT state transitions in the CAN network

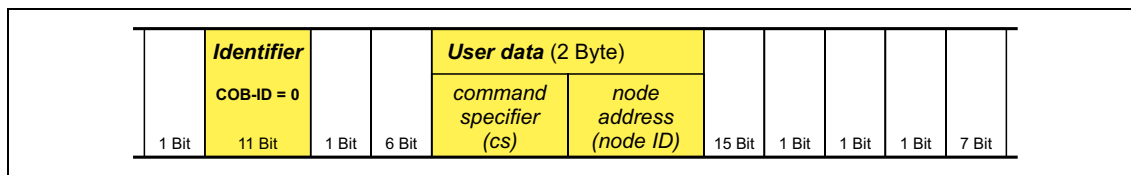
Transition	NMT command	Status after change	Effects on process/parameter data after status change
(1)	-	Initialisation	Initialisation starts automatically when the mains is switched on. - During the initialisation phase, the inverter does not participate in the data exchange. - After the initialisation is completed, the node sends a boot-up message with an individual identifier and automatically changes to the "Pre-operational" status.
(2)	-	Pre-Operational	In this phase, the master determines the way in which the node(s) takes/take part in communication.
	From here, the master changes the states for the entire network. - A target address included in the NMT command defines the receiver(s). - If the inverter has been configured as CAN master, the state change to "Operational" takes place automatically after a waiting time has elapsed (C00356/1), and the NMT command 0x0100 ("Start Remote Node") is sent to all nodes. - Data can only be exchanged via process data objects if the status is "Operational"!		
(3), (6)	0x01 xx Start remote node	Operational	Network management/sync/emergency telegrams as well as process data (PDO) and parameter data (SDO) are active. Optional: When the status is changed, event- and time-controlled process data (PDOs) are transmitted once.
(4), (7)	0x80 xx Enter Pre-operational	Pre-Operational	Network management/sync/emergency telegrams and parameter data (SDO) are active.
(5), (8)	0x02 xx Stop remote node	Stopped	Only network management telegrams can be received.
(9), (10), (11)	0x81 xx Reset node	Initialisation	All CAN-relevant parameters (CiA DS 301) are initialised with the saved values.
(12), (13), (14)	0x82 xx Reset communication		All CAN-relevant parameters (CiA DS 301) are initialised with the saved values.
	Meaning of the node address in the NMT command: - xx = 0x00: With this assignment, all nodes are addressed by the telegram (broadcast telegram). The state can be changed for all nodes at the same time. - xx = Node ID: If a node address is specified, only the status of the node with the corresponding address changes.		

Data transfer

Communication phases/network management

7.2.2 Network management telegram (NMT)

The telegram for the network management contains the identifier "0" and the command included in the user data, which consists of the command byte and the node address:



[7-3] Network management telegram for changing over the communication phases

Command specifier (cs)		NMT command
dec	hex	
1	0x01	Start remote node
2	0x02	Stop remote node
128	0x80	Enter Pre-operational
129	0x81	Reset node
130	0x82	Reset communication

One node, the CAN master, is responsible for switching over the communication phases in the entire network. The role of the CAN master can also be taken over by the inverter.

► [Parameterising the Inverter Drives 8400 motec as CAN master](#) (40)

Example :

Data can only be exchanged via process data objects if the status is "Operational". If the CAN master is supposed to switch all nodes connected to the bus from the "Pre-operational" communication status to the "Operational" communication status, the identifier and user data in the transmission telegram must be set as follows:

- Identifier: 0x00 (network management)
- User data: 0x0100 ("Start remote node" NMT command to all nodes)

7.2.3 Parameterising the Inverter Drives 8400 motec as CAN master

If the initialisation of the system bus and the associated status change from "Pre-operational" to "Operational" is not effected by a higher-level host, the Inverter Drive 8400 motec can instead be defined to be a "quasi" master to execute this task.

Configuration of the inverter as CAN master is carried out in [C00352](#).

- As CAN master, the inverter sets all nodes connected to the bus (broadcast telegram) to the "Operational" communication status by means of the NMT telegram "Start Remote Node". This is the only communication status allowing for data exchange via the process data objects.
- [C00356/1](#) can be used to set a delay time, which has to pass after mains switching before the inverter applies the "Start Remote Node" NMT telegram on the bus.

Parameters	Info	Lenze setting	
		Value	Unit
C00352	CAN slave/master	Slave	
C00356/1	CAN delay boot-up - Operational	3000	ms



Note!

The changes of the master/slave operation in [C00352](#) will only be activated

- when mains switching of the inverter takes place again

or

- by transmission of the "Reset Node" NMT telegram or "Reset Communication" to the inverter.

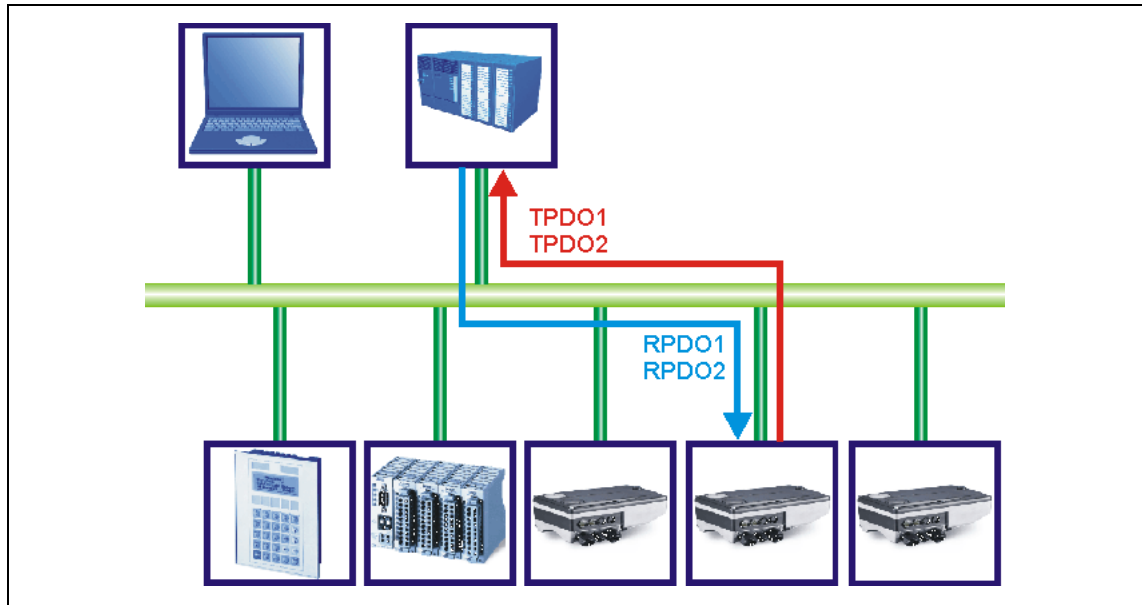
The "CAN reset node" device command (C00002/26) is provided as an alternative to the "Reset node" NMT telegram for the reinitialisation of the CAN-specific device parameters.



Tip!

Master functionality is only required during the initialisation phase of the drive system.

8 Process data transfer



[8-1] PDO data transfer from / to the higher-level host (master)

The CANopen communication unit is provided with two separate process data channels (PDO1 and PDO2) for transmitting process data. Each process data channel can transmit up to four words (8 bytes) at a maximum.

The system bus (CANopen) transmits parameter data, configuration data, diagnostic data, alarm messages and process data between the host (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.

- Process data are transmitted via the process data channel.
- The process data serve to control the inverter.
- The transmission of process data is time-critical.
- Process data are cyclically transferred between the master and the slaves participating in the fieldbus (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 assigned to the I/O area.
- Process data are not saved in the inverter.
- Process data are e.g. setpoints, actual values, control words, and status words.

Agreements

- The following distinction for process data telegrams between the host (master) and the inverter (slaves) is made with regard to their direction:
 - Process data telegrams to the device (RPDO)
 - Process data telegrams from the device (TPDO)
- The CANopen process data objects are designated as seen from the node's view:
 - Receive PDOs (RPDOx): Process data object received by a node
 - Transmit PDOs (TPDOx): Process data object sent by a node



Note!

Data can only be exchanged via process data objects if the status is "Operational"!

► [Communication phases/network management](#) (📖 36)

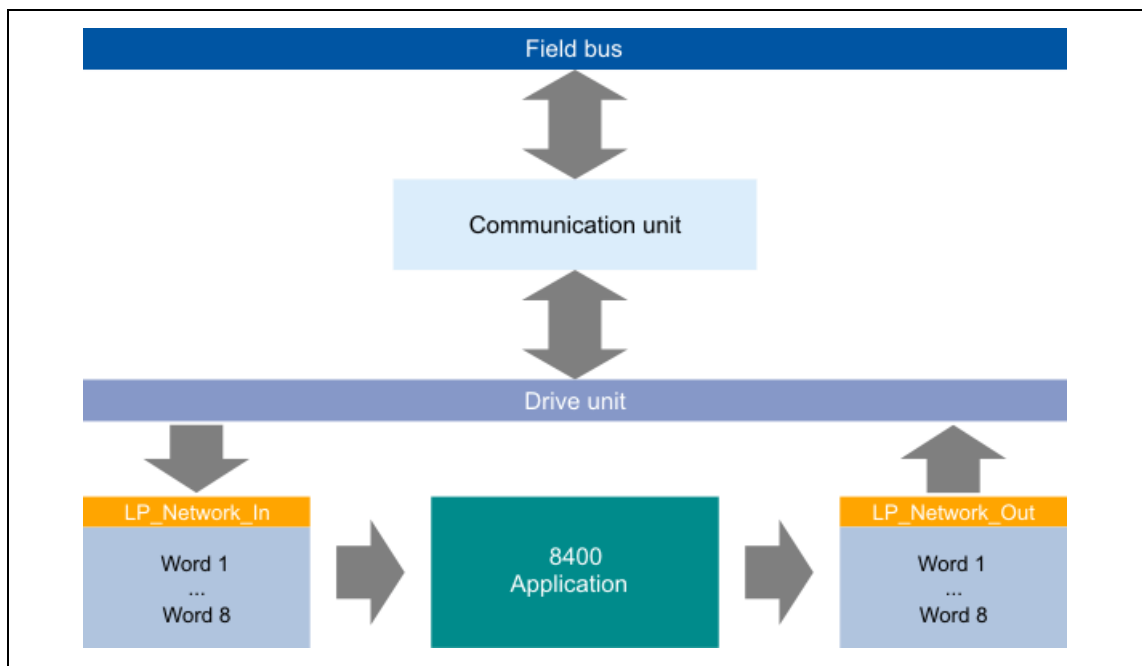
Process data transfer

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.
 - 2 x 4 words via the input ports CAN1_IN and CAN2_IN
 - 2 x 4 words via the output ports CAN1_OUT and CAN2_OUT
- 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.
- The port/function block interconnection of the process data objects (PDO) takes place via the Lenze »Engineer«.



[8-2] External and internal data transfer between the bus system, inverter, and application



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

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

Process data transfer

Port interconnection of process data objects (PDO)

8.2 Port interconnection of process data objects (PDO)



Note!

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

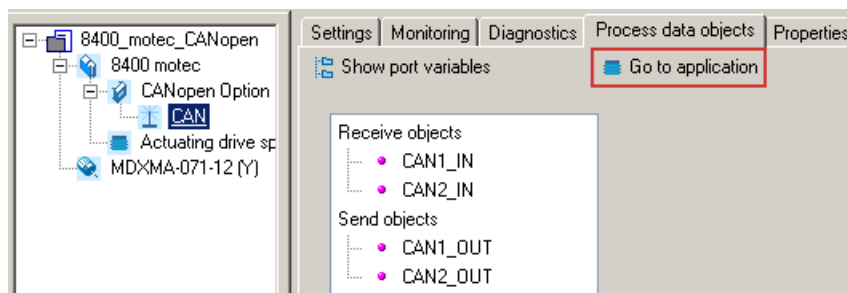
Depending on the software version of the inverter and the version of the »Engineer« software installed, the screenshots in this documentation may differ from the actual »Engineer« screens.

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

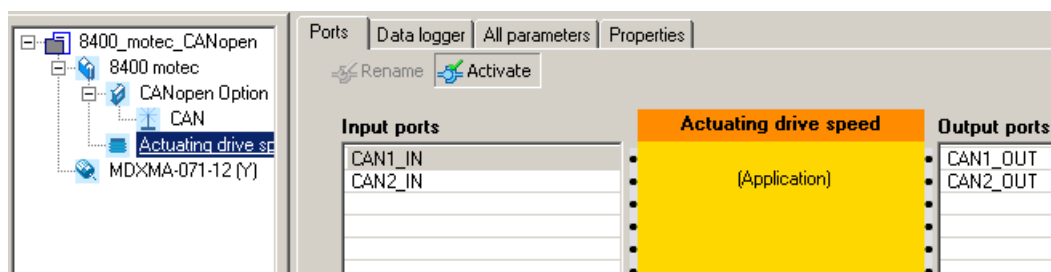


How to freely configure the port interconnection in the »Engineer«:

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



2. The **Ports** tab displays the port blocks CAN1_IN/CAN2_IN and CAN1_OUT/CAN2_OUT.



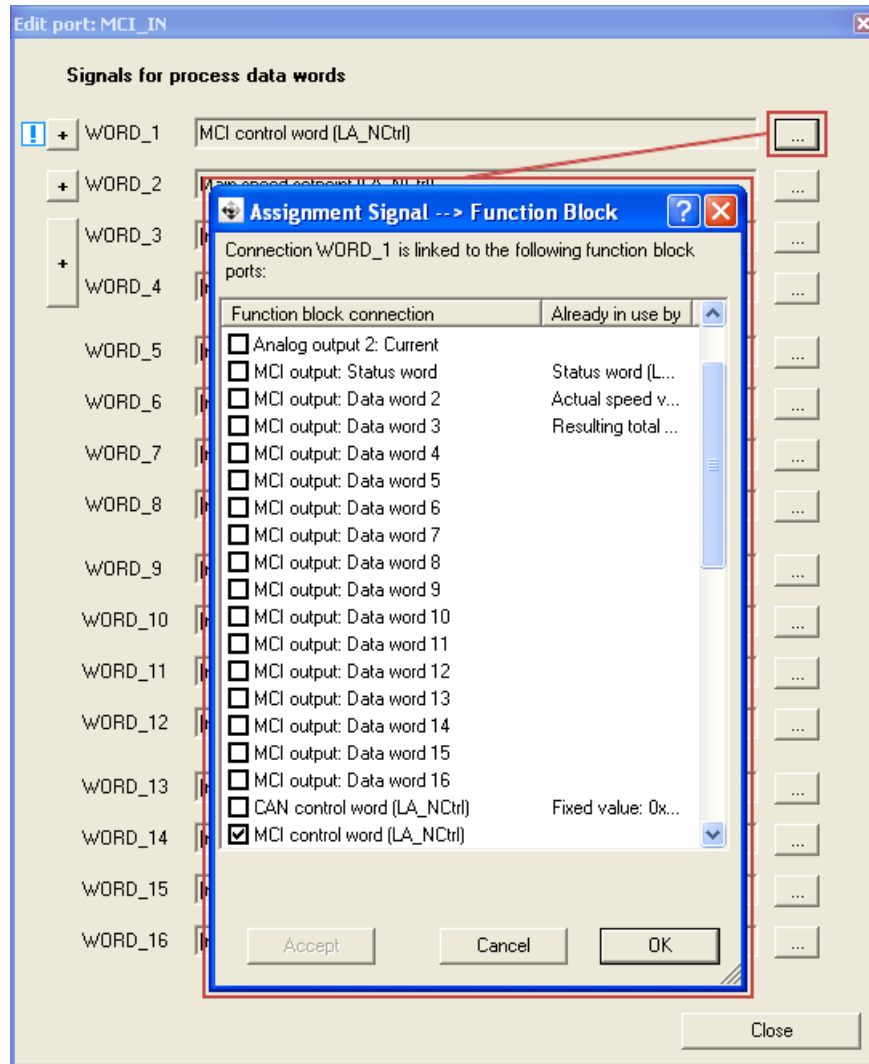
Port interconnection of process data objects (PDO)

Lenze · Decentralised frequency inverter 8400 motec (CANopen option) · EDS84DMOTCAN EN 4.0 - 02/2019

Process data transfer

Port interconnection of 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**.

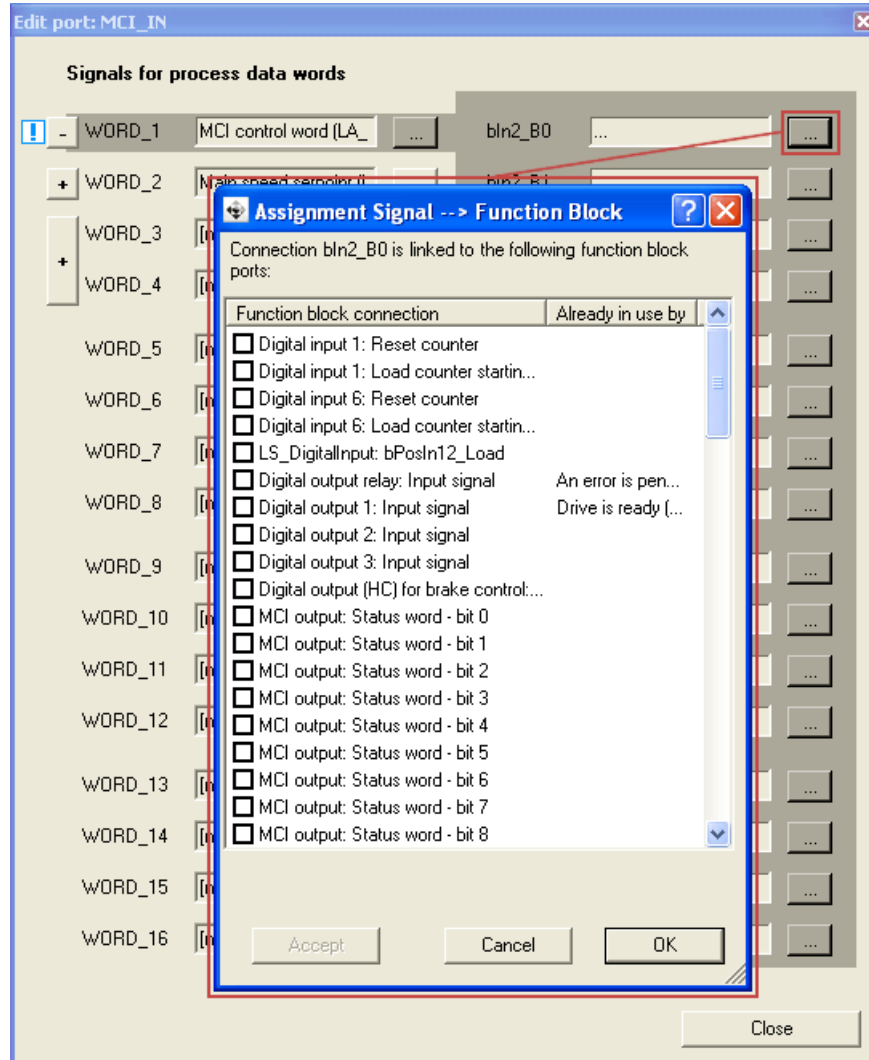


Process data transfer

Port interconnection of 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)**.

Process data transfer

Identifiers of the process data objects

8.3 Identifiers of the process data objects

In the Lenze setting, the identifier for the process data objects PDO1 and PDO2 consists of a basic identifier (CANBaseID) and the node address set in [C00350](#):

Identifier (COB-ID) = basic identifier + node address (node ID)

- The basic identifiers of the PDOs comply with the "Predefined connection set" of DS301, V4.02.
- Alternatively, define via code [C00353](#) that the identifiers of the PDOs are to be assigned according to Lenze definition or that individual settings are to be made.
 - If [C00353](#) = "2: COBID = C0354/x", the identifiers of the PDOs can be set individually via the Lenze codes and CANopen indexes listed in the table below. That way, identifiers independent of the node address can be set for specific PDOs.
 - If identifiers are assigned individually, all PDOs must have basic identifier values in the range of 385 ... 1407.

Process data object	Basic identifier		Individual setting	
	dec	hex	Lenze code	CANopen index
PDO1				
RPDO1	512	0x200	C00354/1	I-1400/1
TPDO1	384	0x180	C00354/2	I-1800/1
PDO2				
RPDO2	768	0x300	C00354/3	I-1401/1
TPDO2	640	0x280	C00354/4	I-1801/1



Note!

After a node address change ([C00350](#)) and a subsequent CAN reset node, the subcodes of [C00354](#) automatically resume the values which result from the respective basic identifier and the node address set.

Short overview: Parameters for setting the identifiers

Parameters	Info	Lenze setting	
		Value	Unit
C00353/1	COBID source CAN1_IN/OUT	0: COBID = C0350 + CANBaseID	
C00353/2	COBID source CAN2_IN/OUT	0: COBID = C0350 + CANBaseID	
C00354/1	COBID CAN1_IN	0x00000201	
C00354/2	COBID CAN1_OUT	0x00000181	
C00354/3	COBID CAN2_IN	0x00000301	
C00354/4	COBID CAN2_OUT	0x00000281	

Process data transfer

Transmission type

8.4 Transmission type

Process data objects can be transmitted in an event-controlled or time-controlled manner. The below table shows that it is possible to combine the different methods by means of logic operations (AND, OR):

- Event-controlled
The PDO is sent when a certain device-internal event has occurred, e.g. when the data contents of the TPDO have changed or when a transmission cycle time has elapsed
- Synchronous transmission
A TPDO (or RPDO) is sent (or received) after the device has received a sync telegram (COB-ID 0x80).
- Cyclic transmission
The cyclic transmission of PDOs takes place when the transmission cycle time has elapsed.
- Polled via RTR
A TPDO is sent when another device requests it by means of a data request telegram (RTR remote transmit request). For this purpose, the data requester (e.g. the master) sends the data request telegram with the COB-ID of the TPDO requested to be sent. The receiver recognises the RTR and transmits the corresponding PDO.

Transmission type	cyclic	PDO transmission synchronous	event-controlled	Logic combination of different transmission types
0		•	•	AND
1 ... 240	•	•		AND
254	•		•	OR

Transmission type	Description
0	Synchronous and acyclic: The PDO is transmitted on an event-controlled basis with every sync (e.g. when a bit change occurs in the PDO).
1 ... 240	Synchronous and cyclic (sync-controlled with response): • Selection n = 1: The PDO is transmitted with <u>every</u> sync. • Selection $1 < n \leq 240$: The PDO is transmitted with <u>every n-th</u> sync.
241 ... 251	Reserved
252, 253	RTR-controlled manner is not permissible.
254	Event-controlled with cyclic transmission: If this value is entered, the PDO is transferred in an event-controlled or cyclic manner. (The values "254" and "255" are equivalent). For a cyclic transmission, a cycle time must be set for the respective PDO. In this case, cyclic transmission takes place in addition to event-controlled transmission.
255	Not permissible

Process data transfer

Transmission type

The communication parameters such as the transmission mode and cycle time can be set freely for every PDO and independently of the settings of other PDOs:

Parameters	Info	Lenze setting	
		Value	Unit
C00322/1	Transmission mode CAN1 OUT	254	
C00322/2	Transmission mode CAN2 OUT	254	
C00323/1	Transmission mode CAN1 IN	254	
C00323/2	Transmission mode CAN2 IN	254	
C00324/1	Inhibit time for emergency telegrams	0	ms
C00324/2	CAN1_OUT blocking time	0	ms
C00324/3	CAN2_OUT inhibit time	0	ms
C00356/5	CAN1_OUT cycle time	0	ms
C00356/2	CAN2_OUT cycle time	0	ms



Tip!

The setting can also be carried out via the following CANopen objects:

- [I-1400](#) / [I-1401](#): Communication parameter for RPDO1 and RPDO2
- [I-1800](#) / [I-1801](#): Communication parameter for TPDO1 and TPDO2

Process data transfer

PDO synchronisation via sync telegram

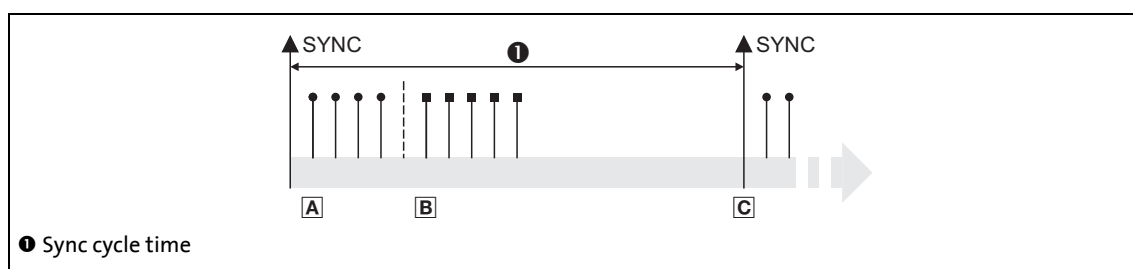
8.5 PDO synchronisation via sync telegram

In case of cyclic transmission, one or several PDOs are transmitted or received at fixed time intervals. For synchronising the cyclic process data, an additional special telegram, the sync telegram, is used.

- The sync telegram is the trigger point for the transmission of process data from the slaves to the master and for the acceptance of process data from the master in the slaves.
- For sync-controlled process data processing, the sync telegram must be generated accordingly.
- The response to a sync telegram is determined by the selected transmission type.

► [Transmission type](#) (49)

General procedure



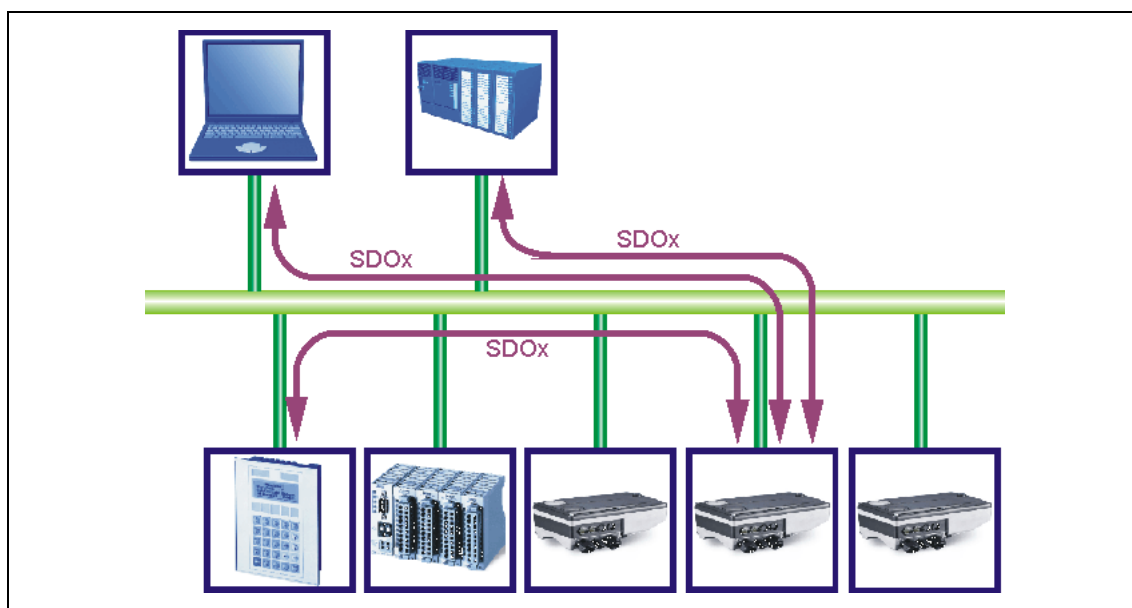
[8-3] Sync telegram

- After the sync telegram has been received, the slaves transmit the synchronous process data to the master (TPDOs). The master reads them as process input data.
- When the transmission process is completed, the slaves receive (RPDOs) the process output data (of the master).
 - All other telegrams (e.g. parameters or event-controlled process data) are accepted acyclically by the slaves after the transmission is completed.
 - The acyclic data are not shown in figure [8-3]. They must be considered when the cycle time is dimensioned.
- The data are accepted in the slave with the next sync telegram if the Rx mode is set to 1 ... 240. If the Rx mode is 254 or 255, the data are accepted in the next device cycle, irrespective of the sync telegram.

Short overview: Parameters for the synchronisation via sync telegram

Parameters	Info	Lenze setting		Assignment	
		Value	Unit	Sync master	Sync slave
C00367	CAN SYNC Rx identifier	128			●
C00368	CAN SYNC Tx identifier	128		●	
C00369	CAN SYNC transmit cycle time	0	ms	●	

9 Parameter data transfer



[9-1] Parameter data transfer via the available parameter data channels

Parameters are values stored in codes on Lenze controllers.

Two parameter data channels are available for parameter setting, enabling the simultaneous connection of different devices for configuration purposes.

Parameter data are transmitted via the system bus as SDOs (*Service Data Objects*) via the system bus (CANopen) and are acknowledged by the receiver. The SDO enables read and write access to all device parameters and to the CANopen object directory integrated in the device. Indexes (e.g. 0x1000) enable access to device parameters and functions included in the object directory. To transfer SDOs, the information contained in the user data must comply with the CAN-SDO protocol.

Parameter data transfer

Identifiers of the parameter data objects

9.1 Identifiers of the parameter data objects

In the Lenze setting, the basic identifiers of the SDOs are preset according to the "Predefined Connection Set".

The identifiers for the parameter data objects SDO1 and SDO2 are generated from the basic identifier and the node address set in code [C00350](#):

Identifier = basic identifier + node address

Object		Direction		Lenze-Base-ID		CANopen-Base-ID	
		from device	to device	dec	Hex	dec	Hex
SDO1 (Parameter data channel 1)	TSDO1	●		1408	580	1408	580
	RSDO1		●	1536	600	1536	600
SDO2 (Parameter data channel 2)	TSDO2	●		1472	5C0	1472	5C0
	RSDO2		●	1600	640	1600	640
Heartbeat		●		1792	700	1792	700
Boot-up		●		1792	700	1792	700



Note!

Please observe that the parameter data channels 1 and 2 are active in the factory setting.

Parameter data transfer

User data

9.2 User data

Structure of the user data of the parameter data telegram

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte



Note!

For the user data, the Motorola format is used.

► [Parameter data telegram examples](#) (📖 59)

The following subchapters provide detailed information on user data.

9.2.1 Command

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte

The following commands can be transmitted or received for writing and reading the parameters:

Command	1st byte		Data length	Info
	hex	dec		
Write request	0x23	35	4 bytes	Writing a parameter to the inverter.
	0x2B	43	2 bytes	
	0x2F	47	1 byte	
	0x21	33	Block	
Write response	0x60	96	4 bytes	Acknowledgement from the inverter to a write request.
Read request	0x40	64	4 bytes	Reading a parameter from the inverter.
Read response	0x43	67	4 bytes	Response by the inverter to a read request with the current parameter value.
	0x4B	75	2 bytes	
	0x4F	79	1 byte	
	0x41	65	Block	
Error response	0x80	128	4 bytes	Response by the inverter if the write/read request could not be executed correctly. ► Error messages (📖 57)

Parameter data transfer

User data

More precisely, the command byte comprises the following information:

Command	1st byte							
	Command specifier (cs)			Toggle (t)	Length*		e	s
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write request	0	0	1	0	0/1	0/1	1	1
Write response	0	1	1	0	0	0	0	0
Read request	0	1	0	0	0	0	0	0
Read response	0	1	0	0	0/1	0/1	1	1
Error response	1	0	0	0	0	0	0	0

*Bit coding of the length: 00 = 4 bytes, 01 = 3 bytes, 10 = 2 bytes, 11 = 1 byte
e: expedited (shortened block service)
s: segmented (normal block service)



Tip!

More commands are defined in CANopen specification DS301, V4.02 (e.g. segmented transfer).

9.2.2 Addressing by means of index and subindex

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte

A parameter (Lenze code) is addressed according to the following formula:

Index = 24575 - (Lenze code number)

Example

The C00011 parameter (motor reference speed) is to be addressed.

Calculation:

- Index:
 - Decimal: $24575 - 11 = 24564$
 - Hexadecimal: $0x5FFF - 0xB = 0x5FF4$
- Subindex: 0x00 (subindex 0 since the parameter does not have any subcodes)

Entries:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	0xF4	0x5F	0x00				

Parameter data transfer

User data

9.2.3 Data 1 ... data 4

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte

Maximally 4 bytes are available for parameter value entries. Depending on the data format, they are assigned as follows:

5th byte	6th byte	7th byte	8th byte
Parameter value (1 byte)	0x00	0x00	0x00
Parameter value (2 bytes)		0x00	0x00
Low byte	High byte		
Parameter value (4 bytes)			
Low word		High word	
Low byte	High byte	Low byte	High byte



Note!

The [Table of attributes](#) (84) contains a scaling factor for Lenze parameters in the "factor" column. The scaling factor is important for the transfer of parameter values which are represented with one or several decimal positions in the parameter list.

If the scaling factor is > 1 , the value must be multiplied by the indicated scaling factor prior to transmission to be able to transfer the value as an integer. At the SDO client end, the integer must be divided by the scaling factor to obtain the original value including decimal positions again.

Example

A value of "123.45" is to be transmitted for a code, unit: "%" (e.g. C00039/1: "Fixed setpoint-JOG1"). Parameters with the "%" unit have two decimal positions and hence a scaling factor of "100".

Calculation:

- Value to be transmitted = scaling factor x value
- Data (1 ... 4) = $100 \times 123.45 = 12345$ (0x00 00 30 39)

Entries:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
				0x39	0x30	0x00	0x00

Parameter data transfer

User data

9.2.4 Error messages

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Error code			
0x80 (128)	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte

In the event of an error, the node addressed generates a telegram with the "Error response" (0x80) command.

- The telegram includes the index and subindex of the code where the error occurred.
- The error code is entered in bytes 5 ... 8.
 - The error codes are standardised according to DS301, V4.02.
 - The representation of the error codes is provided in reverse read direction (see example below).

Example

Representation of error code "0x06 04 00 41" in bytes 5 ... 8:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Error code			
				0x41	0x00	0x04	0x06

Parameter data transfer

User data

Meaning of the error codes

The error codes are standardised according to DS301, V4.02.

Error code	Explanation
0x0503 0000	Toggle bit not changed
0x0504 0000	SDO protocol expired
0x0504 0001	Invalid or unknown client/server command specifier
0x0504 0002	Invalid block size (only block mode)
0x0504 0003	Invalid sequence number (only block mode)
0x0504 0004	CRC error (only block mode)
0x0504 0005	Not sufficient memory
0x0601 0000	Object access not supported
0x0601 0001	Attempt to read a write-only object
0x0601 0002	Attempt to write to a read-only object
0x0602 0000	Object not listed in object directory
0x0604 0041	Object not mapped to PDO
0x0604 0042	Number and length of objects to be transferred longer than PDO length.
0x0604 0043	General parameter incompatibility
0x0604 0047	General internal device incompatibility
0x0606 0000	Access denied because of hardware error
0x0607 0010	Unsuitable data type, unsuitable service parameter length
0x0607 0012	Unsuitable data type, service parameter length exceeded
0x0607 0013	Unsuitable data type, service parameter length not long enough
0x0609 0011	Subindex does not exist
0x0609 0030	Parameter value range exceeded
0x0609 0031	Parameter values too high
0x0609 0032	Parameter values too low
0x0609 0036	Maximum value falls below minimum value
0x0800 0000	General error
0x0800 0020	Data cannot be transferred/stored for application.
0x0800 0021	Data cannot be transferred/stored for application due to local control.
0x0800 0022	Data cannot be transferred/stored for application due to current device status.
0x0800 0023	Dynamic generation of object directory not successful or no object directory available (e.g. object directory generated from file, generation not possible because of a file error).

Parameter data transfer

Parameter data telegram examples

9.3 Parameter data telegram examples

9.3.1 Read parameters

Task: The heatsink temperature of 43 °C (code C00061, data format INTEGER32, scaling factor 1) of the inverter with the node address "5" is to be read.

Telegram to the drive

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x0605	0x40	0xC2	0x5F	0x00	0x00	0x00	0x00	0x00

Explanations on the telegram to the drive	
Identifier	= 1536 + node address = 1536 + 5 = 1541 = 0x0605 (1536 = basic identifier SDO1 to the inverter)
Command	= 0x40 = "Read Request" (read request of a parameter from the inverter)
Index	= 24575 - code number = 24575 - 61 = 24514 = 0x5FC2
Subindex	= 0 (code C00061 does not have any subcodes)

Response telegram from drive (if data have been transmitted correctly)

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x0585	0x43	0xC2	0x5F	0x00	0x2B	0x00	0x00	0x00

Explanations on the telegram from the drive	
Identifier	= 1408 + node address = 1408 + 5 = 1413 = 0x0585 (1408 = basic identifier SDO1 from the inverter)
Command	= 0x43 = "Read response" (response to the read request with current value)
Index	As in telegram to the drive
Subindex	
Data 1 ... 4	= 0x0000002B = 43 [°C]

Parameter data transfer

Parameter data telegram examples

9.3.2 Write parameters

Task: The rated current of the motor connected with $I_{rated} = 10.20 \text{ A}$ (code C00088) is to be entered into the inverter with the node address "2".

Data 1 ... 4	Calculation
Value for motor current, (data type U16; display factor 1/100)	$10.20 \times 100 = 1020 \text{ (0x03 FC)}$

Telegram to the drive

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x0602	0x2B	0xA7	0x5F	0x00	0xFC	0x03	0x00	0x00

Explanations on the telegram to the drive	
Identifier	= $1536 + \text{node address} = 1536 + 2 = 1538 = 0x0602$ (1536 = basic identifier SDO1 to the inverter)
Command	= $0x23 = \text{"Write Request"}$ (write request of a parameter to the inverter)
Index	= $24575 - \text{code number} = 24575 - 88 = 24487 = 0x5FA7$
Subindex	= 0 (code C00088 does not have any subcodes)
Data 1 ... 4	= $10.20 \times 100 = 1020 = 0x000003FC$ (motor current value; data type U32; display factor 1/100)

Response telegram from drive (if data have been transmitted correctly)

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x0582	0x60	0xA7	0x5F	0x00	0x00	0x00	0x00	0x00

Explanations on the telegram from the drive	
Identifier	= $1408 + \text{node address} = 1408 + 2 = 1410 = 0x0582$ (1408 = basic identifier SDO1 from the inverter)
Command	= $0x60 = \text{"Write Response"}$ (acknowledgement of the write access from the inverter)
Index	As in telegram to the drive
Subindex	

Parameter data transfer

Parameter data telegram examples

9.3.3 Reading block parameters

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

Telegram 1 to the drive: Read request

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x060C	0x40	0x9C	0x5F	0x00	0x00	0x00	0x00	0x00

Explanations on the telegram to the drive	
Identifier	= 1536 + node address = 1536 + 12 = 1548 = 0x060C (1536 = basic identifier SDO1 to the inverter)
Command	= 0x40 = "Read Request" (read request of a parameter from the inverter)
Index	= 24575 - code number = 24575 - 99 = 24476 = 0x5F9C
Subindex	= 0 (code C00099 does not have any subcodes)

Response message 1 from the drive: Indication of the block length (11 characters)

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x058C	0x41	0x9C	0x5F	0x00	0x0B	0x00	0x00	0x00

Explanations on the telegram from the drive	
Identifier	= 1408 + node address = 1408 + 12 = 1420 = 0x058C (1408 = basic identifier SDO1 from the inverter)
Command	= 0x41 = "Read response" (response is block telegram)
Index	As in telegram to the drive
Subindex	
Data 1 ... 4	= 0x0000000B = data length of 11 characters in the ASCII format

Parameter data transfer

Parameter data telegram examples

Telegram 2 to the drive: Request of the 1st data block

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x060C	0x60	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Explanations on the telegram to the drive	
Command	= 0x60 = "Read segment request" (request: read data block) • Bit 4 = 0 (toggle bit) Influence of the toggle bit on the request command The individual blocks are toggled successively, i.e. first the request with command "0x60" is effected (= 0110*0000_{bin}), then command "0x70" (= 0111*0000_{bin}), then "0x60" again, etc. * Toggle bit

Response message 2 from the drive: Transmission of the 1st data block

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x058C	0x00	0x30	0x31	0x2E	0x30	0x30	0x2E	0x30
		0 _{asc}	1 _{asc}	· _{asc}	0 _{asc}	0 _{asc}	· _{asc}	0 _{asc}

Explanations on the telegram to the drive	
Command	= 0x00 = 00000000_{bin} • Bit 4 = 0 (toggle bit) Influence of the toggle bit on the transmission command • The 1st response of the inverter in the command byte is "0x0000*0000_{bin}" if bytes 2 ... 8 are fully occupied with data and further telegrams are following. • The 2nd response of the inverter in the command byte is "0x0001*0000_{bin}" if bytes 2 ... 8 are fully occupied with data and further telegrams are following, etc. * Toggle bit
Data 1 ... 7	= "01.00.0" (ASCII representation)

Parameter data transfer

Parameter data telegram examples

Telegram 3 to the drive: Request of the 2nd data block

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x060C	0x70	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Explanations on telegram 3 to the drive

Command	= 0x70 = "Read segment request" (request: read data block) • Bit 4 = 1 (toggle bit)
---------	--

Response message 3 from the drive: Transmission of the 2nd data block including end identifier

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x058C	0x17	0x30	0x2E	0x30	0x30	0x00	0x00	0x00
		0 _{asc}	·asc	0 _{asc}	0 _{asc}	-	-	-

Explanations on telegram 3 from the drive

Command	= 0x17 = 00010111 _{bin} : • Bit 0 = 1 (end of transmission) • Bit 1 ... bit 3 = 011 _{bin} (3 bytes do not contain any data) • Bit 4 = 1 (toggle bit) Influence of the final bit and the residual data length on the transmission command • The end of transmission is signalled via the set final bit 0. • Bits 1 ... 3 reveal the number of bytes that do not contain data anymore. * Toggle bit
Data 1 ... 7	= "0.00" (ASCII representation) The result of the data block transmission is: "01.00.00.00"

Monitoring

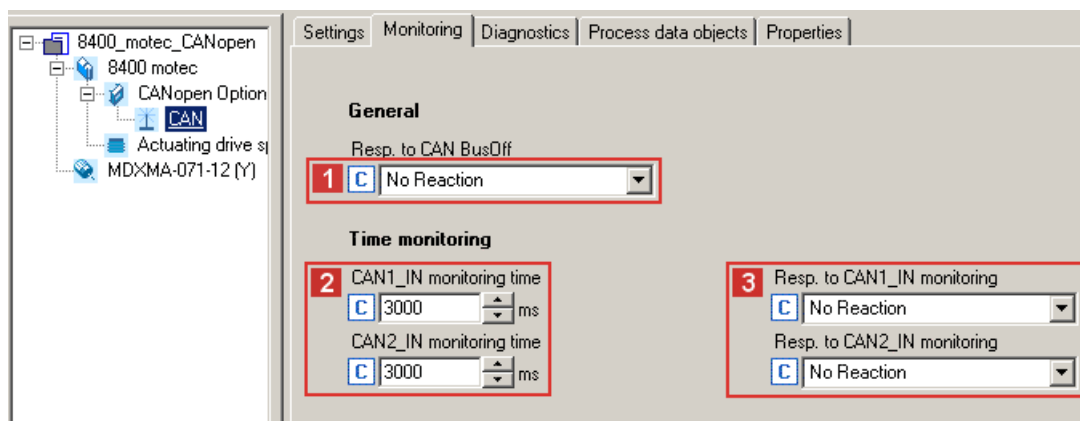
Monitoring of the RPDOs for data reception

10 Monitoring

10.1 Monitoring of the RPDOs for data reception

RPDO1 and RPDO2 have a parameterisable monitoring time within which the RPDO must arrive. The following settings can be made in the »Engineer« under the **Monitoring** tab:

- Response to "BusOff" (bus system switched off), **1** [C00592/2](#)
- CAN1_IN monitoring time, **2** [C00357/1](#)
- CAN2_IN monitoring time, **2** [C00357/2](#)
- Response to CAN1_IN monitoring, **3** [C00593/1](#)
- Response to CAN2_IN monitoring, **3** [C00593/2](#)



If a monitoring time > 0 ms **2** ([C00357/1...2](#)) is entered for CAN1_IN/CAN2_IN, the RPDO is expected after the set time has expired.

If the RPDO is not received within the monitoring time or with the configured sync, the response set for the respective RPDO is effected **3** ([C00593/1...2](#)).

A monitoring time = 0 ms deactivates the monitoring function.

10.2 Integrated error detection

If a node detects an error, it rejects the CAN telegram bits received so far and transmits an error flag. The error flag consists of 6 consecutive bits with the same logic value.

The following errors are detected:

Bit error

The sending node monitors the bus and interrupts the transmission if it receives a different logic value than the value transmitted. With the next bit, the sending node starts the transmission of an error flag.

In the arbitration phase, the sender only detects a bit error if a dominantly sent bit is received as a recessive bit. In the ACK slot as well, the dominant overwriting of a recessive bit is not indicated as a bit error.

Stuff-bit error

If more than 5 consecutive bits before the ACK delimiter in the CAN telegram have the same logic value, the previously transmitted telegram will be rejected and an error flag will be sent with the next bit.

CRC error

If the CRC checksum received does not correspond to the checksum calculated in the CAN chip, the CAN controller sends an error flag after the ACK delimiter, and the previously transmitted telegram is invalidated.

Acknowledgement error

If the ACK slot which is sent recessively by the transmitting node is not overwritten dominantly by a receiver, the transmitting node aborts the transmission. The transmitting node invalidates the telegram transmitted and sends an error flag with the next bit.

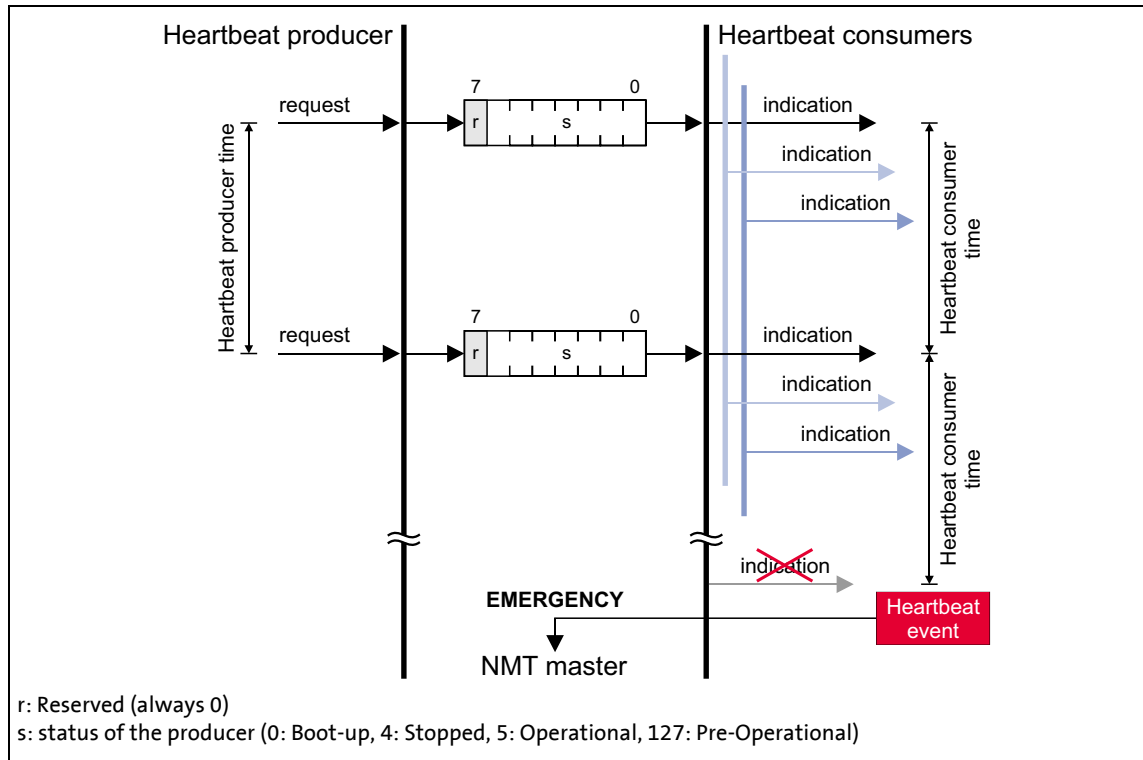
Format error

If a dominant bit is detected in the CRC delimiter, in the ACK delimiter or in the first 6 bits of the EOF field, the telegram received is rejected and an error flag is sent with the next bit.

10.3 Heartbeat protocol

The heartbeat protocol can be used for node monitoring purposes within a CAN network.

General procedure



[10-1] Heartbeat protocol

1. A heartbeat producer cyclically transmits a so-called heartbeat telegram to one or more consumers.
2. The consumer(s) monitor(s) the heartbeat telegram for arrival on a regular basis.

10.3.1 Telegram structure

- The heartbeat telegram of the producer has the following identifier:
Identifier (COB-ID) = 1792 + producer's node address
- The user data (1 byte) contain the status (s) of the producer:

Heartbeat producer status		Data							
Communication status	Decimal value (s)	(r)	Producer status (s)						
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Boot-up	0	0	0	0	0	0	0	0	0
Stopped	4	0	0	0	0	0	1	0	0
Operational	5	0	0	0	0	0	1	0	1
Pre-Operational	127	0	1	1	1	1	1	1	1

10.3.2 Parameter setting

Short overview of the parameters for the "Heartbeat" monitoring function:

Parameters	Info	Lenze setting		Assignment	
		Value	Unit	Consumer	Producer
C00347/1...n	CAN status of heartbeat producer 1	-		●	
C00381	Heartbeat producer time	0	ms		●
C00385/1...n	CAN node address of heartbeat producer 1	0		●	
C00386/1...n	Heartbeat consumer time for heartbeat producer 1	0	ms	●	
C00592/5	Resp. to heartbeat event	No response		●	
Greyed out = display parameter					

Heartbeat producer time

Time interval for the transmission of the heartbeat telegram to the consumer(s).

- Parameterisable in [C00381](#) or via object [I-1017](#). The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is sent automatically as soon as a time > 0 ms is set.

Heartbeat consumer time

Monitoring time for the nodes (producers) to be monitored.

- Parameterisable in [C00386/1...n](#) or via object [I-1016](#).
- The parameterised time is rounded down to an integer multiple of 5 ms and must have a greater value than the heartbeat producer time of the node to be monitored.
- 1 Heartbeat Producer can be monitored.
- The node address(es) of the nodes to be monitored is/are set in [C00385/1...n](#) or via object [I-1016](#), too.

Heartbeat event

The "heartbeat event" is triggered in the consumer if the consumer has not received a heartbeat telegram from the producer within the heartbeat consumer time:

- The consumer changes from the "Operational" communication status to the "Pre-operational" communication status.
- The NMT master receives an emergency telegram containing emergency error code 0x8130.
- The response parameterised in [C00592/5](#) is activated (Lenze setting: "No response").



Note!

The heartbeat monitoring will not start until the first heartbeat telegram of a monitored producer has been received successfully and the "Pre-operational" NMT status has been achieved.

The boot-up telegram counts as the first heartbeat telegram.

10.3.3 Commissioning example

Task

An inverter configured as heartbeat consumer (node 2) is to monitor another inverter (heartbeat producer; node 1).

- The heartbeat producer is to transmit a heartbeat telegram to the heartbeat consumer every 10 ms.
- The heartbeat consumer monitors the heartbeat telegram for arrival. A response is to be activated in the event of an error.

Parameterising the heartbeat producer (node 1)

1. Set the heartbeat producer time ([C00381](#)) to 10 ms.

Parameterising the heartbeat consumer (node 2)

1. Set the CAN node address of the producer in [C00385/1](#).
2. Set the heartbeat consumer time in [C00386/1](#).
 - Note: The heartbeat consumer time must be greater than the heartbeat producer time of the node to be monitored set in [C00381](#).
3. Set the desired response in [C00592/5](#) which is to be activated if a heartbeat event in the consumer occurs.



Tip!

[C00347/1...n](#) displays the heartbeat status of the nodes monitored.

Heartbeat telegram

- The heartbeat telegram from the producer has the following identifier:
Identifier (COB-ID) = 1792 + producer node address = 1792 + 1 = 1793 = 0x701

10.4 Emergency telegram

If the error status changes because an internal device error occurs or has been eliminated, the NMT master once receives an emergency telegram with the following structure:

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

Emergency error code	Error register	Cause
0x0000	0xXX	One of several errors eliminated
	0x00	Single error eliminated (no more errors)
0x3100	0x01	Supply voltage of standard device faulty or failed
0x8100	0x11	Communication error (warning)
0x8130	0x11	Life guarding error or heartbeat error
0x8150	0x11	Collision of identifiers (COB-IDs): An identifier parameterised for reception is also used for transmission.
0x8210	0x11	PDO length shorter than expected
0x8220	0x11	PDO length greater than expected
0x8700	0x11	Sync telegram monitoring
0xF000	0x01	Generic error - An error with a "Fault", "Trouble", "TroubleQSP", "Warning", or "SystemFault" error response occurred in the standard device. - Error message is the Lenze error number (C00168).

More emergency error codes are listed in the short overview of the error messages of the operating system in the software manual/»Engineer« online help "Inverter Drives 8400 motec".

Example

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Emergency error code		Error register	Manufacturer-specific error message				
0x00	0xF0	0x01	0x00 Reserved	Lenze error number			
Generic error				Corresponding error-free message: Value "0x00000000"			



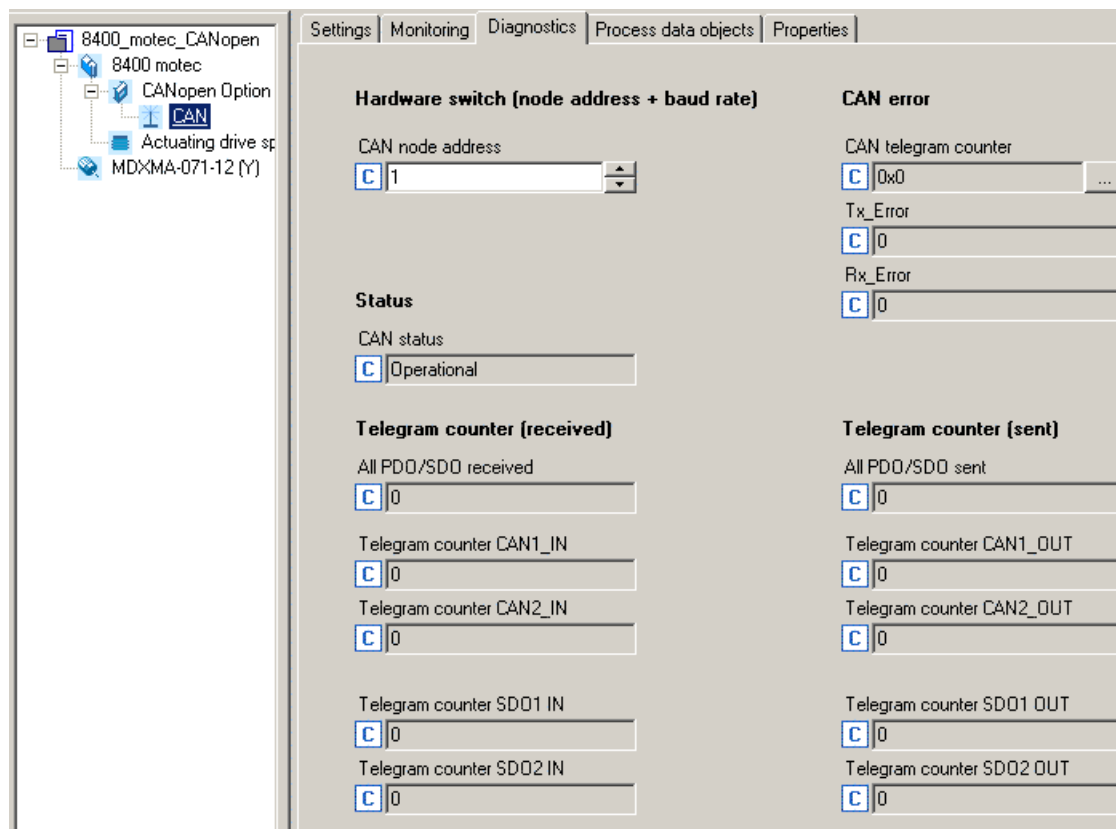
Tip!

A detailed description can be found in CAN specification DS301, V4.02.

11 Diagnostics

Diagnostics with the »Engineer«

In the »Engineer« under the **Diagnostics** tab, various system bus (CANopen) diagnostics information is displayed.



Parameter reference

Communication-relevant parameters of the operating system

12 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 motec by the parameters for CANopen communication.



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

Here you will find general information on parameters.

12.1 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 Resp. to communication error with MCI		
Configuration of monitoring functions for the Communication Unit		
Selection list		
0	No Reaction	
1	Fault	
4	WarningLocked	
Subcodes	Lenze setting	Info
C01501/1	1: Fault	Resp. to MCI fault 1 • Response to a communication fault.
C01501/2	1: Fault	Resp. to MCI fault 2 • Response to an incompatible 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 relevant for CANopen communication

12.2 Parameters relevant for CANopen communication

This chapter lists the CANopen parameters of the communication unit in numerically ascending order.

C00322

Parameter Name: C00322 Transmission mode CAN TxPDOs		Data type: UNSIGNED_8 Index: 24253 _d = 5EBD _h
TPDO transmission type according to DS301 V4.02 • The following transfer modes are supported: • 0: synchronous and acyclic • 1 ... 240: synchronous and cyclic • 252: synchronous - only RTR • 253: asynchronous - only RTR • 254: send PDOs in an event-controlled fashion as long as C00324 = 0 • 255: asynchronous - device profile-specific • The basic setting for all PDOs is "Send PDOs in event-controlled fashion as long as C00324 = 0" (254). • Representation of the CANopen objects I-1800/2 and I-1801/2 (see DS301 V4.02).		
Setting range (min. value unit max. value)		
0		255
Subcodes	Lenze setting	Info
C00322/1	254	Transmission mode CAN1 OUT
C00322/2	254	Transmission mode CAN2 OUT
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C00323

Parameter Name: C00323 Transmission mode CAN Rx PDOs		Data type: UNSIGNED_8 Index: 24252 _d = 5EBD _h
RPDO transmission type according to DS301 V4.02 • For the RPDO, it serves as monitoring setting in the case of sync-controlled PDOs. • The following transfer modes are supported: • 0: synchronous and acyclic • 1 ... 240: synchronous and cyclic • 252: synchronous - only RTR • 253: asynchronous - only RTR • 254: asynchronous - manufacturer-specific • 255: asynchronous - device profile-specific • The basic setting for all PDOs is "Asynchronous - manufacturer-specific" (254). • Representation of the CANopen objects I-1400/2 and I-1401/2 (see DS301 V4.02).		
Setting range (min. value unit max. value)		
0		255
Subcodes	Lenze setting	Info
C00323/1	254	Transmission mode CAN1 IN
C00323/2	254	Transmission mode CAN2 IN
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

Parameter reference

Parameters relevant for CANopen communication

C00324

Parameter Name: C00324 CAN transmission blocking time		Data type: UNSIGNED_16 Index: 24251 _d = 5EBB _h	
Blocking time for the transmission of the emergency telegram and the process data			
Note: If the "Asynchronous - manufacturer-specific/device profile-specific" transmission type is set, the transmission cycle timer is reset to 0 if event-controlled transmission has been triggered. Example: cycle time (C00356/x) = 500 ms, blocking time = 100 ms, sporadic data change: • In the case of a sporadic data change < 500 ms, due to the blocking time set, the quickest transmission cycle is 100 ms (event-controlled transmission). • In the case of a sporadic data change > 500 ms, due to the cycle time set, the transmission cycle is 500 ms (cyclic transmission). • Representation of the CANopen objects I-1800/3 and I-1801/3 (see DS301 V4.02).			
Setting range (min. value unit max. value)			
0	ms	6500	
Subcodes	Lenze setting	Info	
C00324/1	0 ms	Inhibit time for emergency telegrams	
C00324/2	0 ms	CAN1_OUT blocking time	
C00324/3	0 ms	CAN2_OUT inhibit time	
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT			

C00345

Parameter Name: C00345 CAN error status		Data type: UNSIGNED_8 Index: 24230 _d = 5EA6 _h	
Display of the CAN error status			
Selection list (read only)			
0	No Error		
1	Warning ErrActive		
2	Warning ErrPassive		
3	Bus off		
4	Reserved		
5	Reserved		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☒ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

Parameter reference

Parameters relevant for CANopen communication

C00347

Parameter Name: C00347 CAN status HeartBeat producer		Data type: UNSIGNED_8 Index: 24228 _d = 5EA4 _h
Display of the heartbeat producer's CAN status ▶ Heartbeat protocol (📖 66)		
Selection list		
0	Boot-up	
4	Stopped	
5	Operational	
127	Pre-Operational	
250	Failed	
255	NoResponse	
Subcodes		Info
C00347/1		Status node 1
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☒ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C00349

Parameter Name: C00349 CAN setting DIP switch		Data type: UNSIGNED_16 Index: 24226 _d = 5EA2 _h
Display of the DIP switch setting at the last time the mains connection was established ▶ Possible settings via DIP switch (📖 30)		
Display area (min. hex value max. hex value)		
0x0000	0xFFFF	
Value is bit-coded:		
Bit 0	Node address 1	
Bit 1	Node address 2	
Bit 2	Node address 4	
Bit 3	Node address 8	
Bit 4	Node address 16	
Bit 5	Node address 32	
Bit 6	Node address 64	
Bit 7	Baud rate 1	
Bit 8	Baud rate 2	
Bit 9	Baud rate 4	
Bit 10	Reserved	
...	...	
Bit 14	Reserved	
Bit 15	Accept DIP switch 24V ON	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☒ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

Parameter reference

Parameters relevant for CANopen communication

C00350

Parameter Name: C00350 CAN node address			Data type: UNSIGNED_8 Index: 24225 _d = 5EA1 _h
Setting of the node address via parameters • The node address can only be parameterised if the node address "0" is set via the DIP switches. • A change of the node address will only become effective after a CAN reset node. ▶ Setting the CAN node address (📖 31)			
Setting range (min. value unit max. value)			Lenze setting
1		63	1
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT			

C00351

Parameter Name: C00351 CAN baud rate		Data type: UNSIGNED_8 Index: 24224 _d = 5EA0 _h	
Setting of the baud rate via parameters • The baud rate can only be parameterised if the baud rate "0" is set via the DIP switches. • A change of the baud rate will only become effective after a CAN reset node. ▶ Setting the baud rate (□ 30)			
Selection list (Lenze setting printed in bold)			
0	500 kbps		
1	250 kbps		
2	125 kbps		
3	50 kbps		
4	1000 kbps		
5	20 kbps		
14	800 kbps		
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT			

C00352

Parameter Name: C00352 CAN slave/master		Data type: UNSIGNED_8 Index: 24223 _d = 5E9F _h	
If "1" is entered and saved, the drive will start as a CAN master after mains switching.			
Selection list (Lenze setting printed in bold)			
0	Slave		
1	Master		
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT			

Parameter reference

Parameters relevant for CANopen communication

C00353

Parameter Name: C00353 CAN IN/OUT COBID source		Data type: UNSIGNED_8 Index: 24222 _d = 5E9E _h
Identifier allocation procedure for the CANx_IN/OUT process data		
Selection list		Info
0	COBID = C0350 + LenzeBaseID	COBID = device address + LenzeBaseID
1	COBID = C0350 + CANBaseID	COBID = device address + CANBaseID (C00354/x)
2	COBID = C0354/x	COBID = direct setting from C00354/x
Subcodes	Lenze setting	Info
C00353/1	1	COBID source CAN1_IN/OUT
C00353/2	1	COBID source CAN2_IN/OUT
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C00354

Parameter Name: C00354 COBID		Data type: UNSIGNED_32 Index: 24221 _d = 5E9D _h
Setting of the default COBID according to CANopen A change of the COBID will only become effective after a CAN reset node. ► Identifiers of the process data objects (48)		
Value is bit-coded:		
Bit 0	COBID Bit0	
...	...	
Bit 10	COBID Bit10	
Bit 11	Reserved	
...	...	
Bit 30	Reserved	
Bit 31	PDO invalid	
Subcodes	Lenze setting	Info
C00354/1	513 (0x00000201)	COBID CAN1_IN
C00354/2	385 (0x00000181)	COBID CAN1_OUT
C00354/3	769 (0x00000301)	COBID CAN2_IN
C00354/4	641 (0x00000281)	COBID CAN2_OUT
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C00355

Parameter Name: C00355 Active COBID			Data type: UNSIGNED_16 Index: 24220 _d = 5E9C _h		
Display of the COBID of the PDOs that is active in the CAN stack ▶ Identifiers of the process data objects (📖 48)					
Display range (min. value unit max. value)					
0		2047			
Subcodes			Info		
C00355/1			Active COBID CAN1_IN		
C00355/2			Active COBID CAN1_OUT		
C00355/3			Active COBID CAN2_IN		
C00355/4			Active COBID CAN2_OUT		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☒ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT					

Parameter reference

Parameters relevant for CANopen communication

C00356

Parameter Name: C00356 CAN time settings			Data type: UNSIGNED_16 Index: 24219 _d = 5E9B _h
Different time settings for the CAN interface			
Setting range (min. value unit max. value)			
0	ms		
Subcodes	Lenze setting	Info	
C00356/1	3000 ms	CAN delay during status change from "Boot-up" to "Operational"	
C00356/2	0 ms	CAN2_OUT cycle time	
C00356/3	0 ms	Reserved	
C00356/4	0 ms	CANx_OUT time "Operational" to "First transmission"	
C00356/5	0 ms	CAN1_OUT cycle time	
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT			

C00357

Parameter Name: C00357 CAN monitoring times			Data type: UNSIGNED_16 Index: 24218 _d = 5E9A _h		
Mapping of the RPDO event time (see DS301 V4.02) • If a different value than "0" is entered, the RPDO is expected after the time set has elapsed. • If the RPDO is not received within the expected time, the response set in C00593/1...2 is effected.					
Setting range (min. value unit max. value)					
0	ms	65000			
Subcodes	Lenze setting		Info		
C00357/1	3000 ms		CAN1_IN monitoring time		
C00357/2	3000 ms		CAN2_IN monitoring time		
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT					

C00359

Parameter Name: C00359 CAN status		Data type: UNSIGNED_8 Index: 24216 _d = 5E98 _h	
Display of the CAN status ▶ Communication phases/network management (📖 36)			
Selection list (read only)			
0	Operational		
1	Pre-Operational		
2	Reserved		
3	Reserved		
4	BootUp		
5	Stopped		
6	Reserved		
7	Reset		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☒ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT			

Parameter reference

Parameters relevant for CANopen communication

C00360

Parameter Name: C00360 CAN telegram counter			Data type: UNSIGNED_16 Index: 24215 _d = 5E97 _h	
Number of received and sent CAN telegrams				
Display range (min. value unit max. value)				
0		65535		
Subcodes			Info	
C00360/1			All PDOs/SDOs sent	
C00360/2			All PDOs/SDOs received	
C00360/3			Telegram counter CAN1_OUT	
C00360/4			Telegram counter CAN2_OUT	
C00360/5			Reserved	
C00360/6			Telegram counter SDO1 OUT	
C00360/7			Telegram counter SDO2 OUT	
C00360/8			Telegram counter CAN1_IN	
C00360/9			Telegram counter CAN2_IN	
C00360/10			Reserved	
C00360/11			Telegram counter SDO1 IN	
C00360/12			Telegram counter SDO2 IN	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☒ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT				

C00364

Parameter Name: C00364 CAN MessageError		Data type: UNSIGNED_8 Index: 24211 _d = 5E93 _h	
Value is bit-coded:			
Bit 0	No Error		
Bit 1	StuffError		
Bit 2	FormError		
Bit 3	AckError		
Bit 4	Bit1Error		
Bit 5	Bit0Error		
Bit 6	CRCError		
Bit 7	Reserved		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☒ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT			

Parameter reference

Parameters relevant for CANopen communication

C00366

Parameter Name: C00366 Number of CAN SDO channels		Data type: UNSIGNED_8 Index: 24209 _d = 5E91 _h
Available from firmware version 02.00 onwards. Selection of the number of active parameter data channels • In the Lenze setting complying with CANopen, only parameter data channel 1 is active. In order to activate both parameter data channels, set the selection "2 SDO Lenze". • Representation of the CANopen objects I-1201 (see DS301 V4.02)		
Selection list (Lenze setting printed in bold)		Info
0	1 SDO CANopen	I-1201 • Subindex1.Bit31 = 1 (client -> server (rx)) • Subindex2.Bit31 = 1 (server -> client (tx)) • Bit 31 = 1 (SDO invalid/not available)
1	2 SDO Lenze	I-1201 • Subindex1.Bit31 = 0 (client -> server (rx)) • Subindex2.Bit31 = 0 (server -> client (tx)) • Bit 31 = 1 (SDO valid/available)
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C00367

Parameter Name: C00367 CAN sync Rx identifier		Data type: UNSIGNED_16 Index: 24208 _d = 5E90 _h
Identifier by means of which the sync slave is to receive sync telegrams. • Mapping of the CANopen object I-1005 (see DS301 V4.02). ► PDO synchronisation via sync telegram (📘 51)		
Setting range (min. value unit max. value)		Lenze setting
128		255 128
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C00368

Parameter Name: C00368 CAN sync-Tx identifier		Data type: UNSIGNED_16 Index: 24207 _d = 5E8F _h
Identifier by means of which the sync master is to transmit sync telegrams. • Mapping of the CANopen object I-1005 (see DS301 V4.02). ► PDO synchronisation via sync telegram (📘 51)		
Setting range (min. value unit max. value)		Lenze setting
128		255 128
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C00369

Parameter Name: C00369 CAN sync transmission cycle time		Data type: UNSIGNED_16 Index: 24206 _d = 5E8E _h
Cycle during which the sync master is to transmit sync telegrams. • If "0 ms" is set (Lenze setting), no sync telegrams are generated. • Mapping of the CANopen object I-1006 (see DS301 V4.02). ► PDO synchronisation via sync telegram (📘 51)		
Setting range (min. value unit max. value)		Lenze setting
0	ms	65000 0 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

Parameter reference

Parameters relevant for CANopen communication

C00372

Parameter Name: C00372 CAN_Tx_Rx_Error			Data type: UNSIGNED_8 Index: 24203 _d = 5E8B _h		
Display of CAN transmission and reception errors					
Display range (min. value unit max. value)					
0	ms	255			
Subcodes			Info		
C00372/1			Transmission error (Tx_Error)		
C00372/2			Receipt error (Rx_Error)		
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☒ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT					

C00381

Parameter Name: C00381 CAN heartbeat producer time			Data type: UNSIGNED_16 Index: 24194 _d = 5E82 _h		
Time interval for the transmission of the heartbeat telegram to the consumer(s). • The heartbeat telegram is sent automatically as soon as a time > 0 ms is set. • Mapping of the CANopen object I-1017 (see DS301 V4.02). ▶ Heartbeat protocol (📖 66)					
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					

C00385

Parameter Name: C00385 CAN NodeID heartbeat producer			Data type: UNSIGNED_8 Index: 24190 _d = 5E7E _h		
Subcode 1 represents the node which is to be monitored via heartbeat. ▶ Heartbeat protocol (📘 66)					
Setting range (min. value unit max. value)					
0		127			
Subcodes		Lenze setting		Info	
C00385/1	0		CAN NodeID heartbeat producer 1		
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☒ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT					

C00386

Parameter Name: C00386 ConsumerTime HeartBeat producer			Data type: UNSIGNED_16 Index: 24189 _d = 5E7D _h		
Monitoring time for the nodes to be monitored • Mapping of the CANopen object I-1016 (see DS301 V4.02). ▶ Heartbeat protocol (📖 66)					
Setting range (min. value unit max. value)					
0	ms	60000			
Subcodes		Lenze setting		Info	
C00386/1	0 ms		ConsumerTime HeartBeat Producer 1		
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT					

Parameter reference

Parameters relevant for CANopen communication

C00389

Parameter Name: C00389 PDO valid / invalid		Data type: UNSIGNED_8 Index: 24186 _d = 5E7A _h
Validity of the PDOs		
Selection list (Lenze setting printed in bold)		
0	PDO available/valid	
1	PDO not available/invalid	
Subcodes	Lenze setting	Info
C00389/1	0	PDO valid / invalid CAN1_IN
C00389/2	0	PDO valid / invalid CAN1_OUT
C00389/3	0	PDO valid / invalid CAN2_IN
C00389/4	0	PDO valid / invalid CAN2_OUT
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☒ COM ☐ MOT		

C00409

Parameter Name: C00409 LP_CanIn mapping		Data type: UNSIGNED_16 Index: 24166 _d = 5E66 _h
Mapping for the port blocks LP_CanIn1...2 • Representation of the CANopen objects I-1600 ... I-1601 (see DS301 V4.02)		
Setting range (min. value unit max. value)		
0		65535
Subcodes	Lenze setting	Info
C00409/1	0	LP_CanIn1_wIn1(wCtrl)
C00409/2	0	LP_CanIn1_wIn2
C00409/3	0	LP_CanIn1_wIn3
C00409/4	0	LP_CanIn1_wIn4
C00409/5	0	LP_CanIn2_wIn1
C00409/6	0	LP_CanIn2_wIn2
C00409/7	0	LP_CanIn2_wIn3
C00409/8	0	LP_CanIn2_wIn4
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☒ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

Parameter reference

Parameters relevant for CANopen communication

C00592

Parameter Name: C00592 Resp. to CAN bus connection		Data type: UNSIGNED_8 Index: 23983 _d = 5DAF _h
Configuration of monitoring of the CAN interface		
Selection list		
0	No Reaction	
1	Fault	
2	Trouble	
4	WarningLocked	
Subcodes	Lenze setting	Info
C00592/1	0: No Reaction	Response to an erroneous telegram in the CAN communication
C00592/2	0: No Reaction	Response to "BusOff" (bus system switched off)
C00592/3	0: No Reaction	Response to warnings of the CAN controller
C00592/4	0: No Reaction	Response to communication stop of a CAN bus node
C00592/5	0: No Reaction	Response to an event in the case of monitoring via heartbeat protocol
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C00593

Parameter Name: C00593 Resp. to CANx_IN monitoring		Data type: UNSIGNED_8 Index: 23982 _d = 5DAE _h
Configuration of monitoring for the reception of PDOs CAN1_IN and CAN2_IN		
Selection list		
0	No Reaction	
1	Fault	
2	Trouble	
4	WarningLocked	
Subcodes	Lenze setting	Info
C00593/1	0: No Reaction	Response if the monitoring time set in C00357/1 for the reception of the PDO CAN1_IN is exceeded.
C00593/2	0: No Reaction	Response if the monitoring time set in C00357/2 for the reception of the PDO CAN2_IN is exceeded.
☒ Read access ☒ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

Parameter reference

Table of attributes

12.3 Table of attributes

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
			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	

Parameter reference

Table of attributes

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	Data type	Factor	R	W	CINH
C00322	Transmission mode CAN TxPDOs	24253	5EBD	A	2	UNSIGNED_8	1	☒	☒	
C00323	Transmission mode CAN Rx PDOs	24252	5EBC	A	2	UNSIGNED_8	1	☒	☒	
C00324	CAN transmission blocking time	24251	5EBB	A	3	UNSIGNED_16	1	☒	☒	
C00345	CAN error status	24230	5EA6	E	1	UNSIGNED_8	1	☒		
C00347	CAN status HeartBeat producer	24228	5EA4	A	1	UNSIGNED_8	1	☒		
C00349	CAN setting of DIP switch	24226	5EA2	E	1	UNSIGNED_16		☒		
C00350	CAN node address	24225	5EA1	E	1	UNSIGNED_8	1	☒	☒	
C00351	CAN baud rate	24224	5EA0	E	1	UNSIGNED_8	1	☒	☒	
C00352	CAN slave/master	24223	5E9F	E	1	UNSIGNED_8	1	☒	☒	
C00353	CAN IN/OUT COBID source	24222	5E9E	A	2	UNSIGNED_8	1	☒	☒	
C00354	COBID	24221	5E9D	A	4	UNSIGNED_32		☒	☒	
C00355	Active COBID	24220	5E9C	A	4	UNSIGNED_16	1	☒		
C00356	CAN time settings	24219	5E9B	A	5	UNSIGNED_16	1	☒	☒	
C00357	CAN monitoring times	24218	5E9A	A	2	UNSIGNED_16	1	☒	☒	
C00359	CAN status	24216	5E98	E	1	UNSIGNED_8	1	☒		
C00360	CAN telegram counter	24215	5E97	A	12	UNSIGNED_16	1	☒		
C00364	CAN MessageError	24211	5E93	E	1	UNSIGNED_8		☒		
C00366	Number of CAN SDO channels	24209	5E91	E	1	UNSIGNED_8	1	☒	☒	
C00367	CAN SYNC Rx identifier	24208	5E90	E	1	UNSIGNED_16	1	☒	☒	
C00368	CAN sync Tx identifier	24207	5E8F	E	1	UNSIGNED_16	1	☒	☒	
C00369	CAN SYNC transmit cycle time	24206	5E8E	E	1	UNSIGNED_16	1	☒	☒	
C00372	CAN_Tx_Rx_Error	24203	5E8B	A	2	UNSIGNED_8	1	☒		
C00381	CAN heartbeat producer time	24194	5E82	E	1	UNSIGNED_16	1	☒	☒	
C00385	CAN NodeID Heartbeat producer	24190	5E7E	A	1	UNSIGNED_8	1	☒	☒	
C00386	ConsumerTime HeartBeat Producer	24189	5E7D	A	1	UNSIGNED_16	1	☒	☒	
C00389	PDO valid / invalid	24186	5E7A	A	4	UNSIGNED_8	1	☒	☒	
C00409	LP_CanIn mapping	24166	5E66	A	8	UNSIGNED_16	1	☒	☒	
C00592	Resp. to CAN bus connection	23983	5DAF	A	5	UNSIGNED_8	1	☒	☒	
C00593	Resp. to CANx_IN monitoring	23982	5DAE	A	2	UNSIGNED_8	1	☒	☒	

13 Implemented CANopen objects

Lenze devices can be parameterised with both Lenze codes and manufacturer-independent "CANopen objects". Completely CANopen-compliant communication can only be achieved by solely using CANopen objects for parameter setting. The CANopen objects described in this chapter are defined in the CAN specification DS301 V4.02.

Many CANopen objects can be mapped to Lenze codes. The "Relationship to Lenze code" column of the following table lists the Lenze codes used.



Note!

Some of the terms used here derive from the CANopen protocol.

Overview of CANopen indexes and their relationship to Lenze codes

CANopen object			Relationship to Lenze code
Index	Subindex	Name	
I-1000	0	Device type	-
I-1001	0	Error register	-
I-1003	Predefined error field		
	0	Number of errors	-
	1 ... 10	Standard error field	-
I-1005	0	COB-ID SYNC message	C00367
			C00368
I-1006	0	Communication cycle period	C00369
I-1014	0	COB-ID EMCY	-
I-1016	Consumer heartbeat time		
	0	Highest subindex supported	-
	1	Consumer heartbeat time	C00385/1...n
			C00386/1...n
I-1017	0	Producer heartbeat time	C00381
I-1018	Identity object		
	0	Highest subindex supported	-
	1	Vendor ID	-
	2	Product code	-
	3	Revision number	-
	4	Serial number	-
I-1200	SDO1 server parameter		
	0	Highest subindex supported	-
	1	COB-ID client → server (rx)	-
	2	COB-ID server → client (tx)	-
I-1201	SDO2 server parameter		C00366
	0	Highest subindex supported	
	1	COB-ID client → server (rx)	
	2	COB-ID server → client (tx)	

Implemented CANopen objects

CANopen object			Relationship to Lenze code
Index	Subindex	Name	
I-1400	RPDO1 communication parameter		
	0	Highest subindex supported	-
	1	COB-ID used by RPDO	C00355/1
	2	Transmission type	C00323/1
I-1401	RPDO2 communication parameter		
	0	Highest subindex supported	-
	1	COB-ID used by RPDO	C00355/3
	2	Transmission type	C00323/2
I-1600	RPDO1 mapping parameter		
	0	Number of mapped application objects in PDO	-
	1 ... 4	Application object 1 ... 4	C00409/1...4
I-1601	RPDO2 mapping parameter		
	0	Number of mapped application objects in PDO	-
	1 ... 4	Application object 1 ... 4	C00409/5...8
I-1800	TPDO1 communication parameter		
	0	Highest subindex supported	-
	1	COB-ID used by TPDO	C00355/2
	2	Transmission type	C00322/1
	3	Inhibit time	-
	5	Event timer	C00356/5 C00369
I-1801	TPDO2 communication parameter		
	0	Highest subindex supported	-
	1	COB-ID used by TPDO	C00355/4
	2	Transmission type	C00322/2
	3	Inhibit time	-
	5	Event timer	C00356/2 C00369
I-1A00	TPDO1 mapping parameter		
	0	Number of mapped application objects in PDO	-
	1 ... 4	Application object 1 ... 4	-
I-1A01	TPDO2 mapping parameter		
	0	Number of mapped application objects in PDO	-
	1 ... 4	Application object 1 ... 4	-

I-1000 - Device type

Index I-1000	Name: Device type					
Subindex	Default setting	Display range (min. value unit max. value)			Access	Data type
0: Device type	0	0		4294967295	ro	U32

The CANopen index I-1000 specifies the profile for this device. Furthermore, additional information defined in the device profile itself can be stored here.

Implemented CANopen objects

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
High word		Low word	
High byte	Low byte	High byte	Low byte
Additional information		Device profile number	

[13-1] Data telegram assignment

The following values are included in the 4 bytes for inverters of the 8400 series:

- 5th and 6th byte: The data contents are 0x0000, i.e. no profile definition.
- 7th byte: The data content indicates the device type: Here, the value 0x00 is included for inverters.
- 8th byte: The data contents are 0x00.

The data content for the 8400 inverter is therefore: 00 00 00 00

I-1001 - Error register

Index: I-1001	Name: Error register					
Subindex	Default setting	Display range (min. value unit max. value)			Access	Data type
0: Error register	-	0		255	ro	U8

Error register

The error status in the data byte (U8) is bit-coded. The following error states are coded in the data byte (U8):

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Error status
0	0	0	0	0	0	0	0	No error
0	0	0	0	0	0	0	1	Device error message
0	0	0	1	0	0	0	1	Communication error

I-1003 - Pre-defined error field

Index: I-1003	Name: Predefined error field					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Number of errors	0	0		255	rw	U8
1 ... 10: Standard error field	-	0		4294967295	ro	U32

Error history

This object indicates that an error has occurred in the module and in the standard device.

Subindex	Meaning
0	Number of saved error messages
1 ... 10	Display of the error list The error messages (U32) consist of a 16-bit error code and a manufacturer-specific information field comprising 16 bits.

Implemented CANopen objects



Note!

The values of the "Standard error field" in subindex 1 ... 10 will be deleted if the "Number of recorded errors" subindex is overwritten with a value of "0".

Emergency error code	Cause	Entry in the Error register (I-1001)
0x0000	One of several errors eliminated	0xXX
	Elimination of one single error (afterwards no more errors)	0x00
0x1000	Standard device is in error status (error response "fault", "message", "warning", "error", "quick stop by trouble", or "system error")	0x01
0x3100	Supply voltage of standard device faulty or failed	0x01
0x8100	Communication error (warning)	0x11
0x8130	Life guard error or heartbeat error	0x11
0x8150	Collision of COB IDs: An ID parameterised for reception is also used for transmission.	0x11
0x8210	PDO length shorter than expected	0x11
0x8220	PDO length greater than expected	0x11
0x8700	Sync telegram monitoring	0x11

I-1005 - COB-ID SYNC message

Index: I-1005	Name: COB-ID SYNC message					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: COB-ID SYNC message	0x0000 0080 or 0x8000 0080	0		4294967295	rw	U32

This object can be used to activate the generation of sync telegrams and to write the identifier value.

This object relates to codes [C00367](#) and [C00368](#).

Creating sync telegrams

To create sync telegrams, bit 30 (see below) must be set to "1". The interval of the sync telegrams can be set with the object [I-1006](#).

Writing identifiers

For the reception of PDOs, the value 0x80 is entered in the Lenze setting (and according to CANopen specification) into the 11 bit identifier. This means that all modules are set to the same sync telegram by default.

- If sync telegrams are only to be received by specific communication modules, their identifiers can be entered with a value of up to and including 0x07FF.
- The identifier may only be changed when the communication module does not send any sync telegrams (bit 30 = "0").

Implemented CANopen objects

- How to change the identifier:
 - Deactivate identifier (set bit 30 to "0").
 - Change identifier.
 - Activate identifier (set bit 30 to "1").

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
x	0/1	Extended identifier*				11-bit identifier	

* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".

[13-2] Data telegram assignment

I-1006 - Communication cycle period

Index: I-1006	Name: Communication cycle period					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Communication cycle period	0 µs	0	µs	65535000	rw	U32

Setting the sync telegram cycle time.

- The cycle time can be selected as "1000" or as an integer multiple of it.
- With "0 µs" (Lenze setting), no sync telegrams are created.
- This object relates to code [C00369](#).

I-1014 - COB-ID EMCY

Index: I-1014	Name: COB-ID EMCY					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: COB-ID EMCY	0x80 + node ID	0		4294967295	rw	U32

When a communication error or an internal error in the communication module or the inverter occurs or is acknowledged (e.g. "Fault"), an error message is sent via the system bus. The telegram is sent once in the event of each error. With bit 31, this function can be activated or deactivated.

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0/1	0	Extended identifier*				11-bit identifier	
* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".							

[13-3] Data telegram assignment

Bit	setting	
Bit 31	0	Emergency object is valid.
	1	Emergency object is invalid.



Note!

The identifier can only be changed in the "emergency object invalid" status (bit 31 = 1).

I-1016 - Consumer heartbeat time

Index: I-1016	Name: Consumer heartbeat time					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Highest subindex supported	1	- (read access only)			ro	U16
1 ... n: Consumer heartbeat time	0	0		65535	rw	U16

Monitoring time for the nodes to be monitored via heartbeat.

► [Heartbeat protocol](#) (66)

The parameterised time is rounded down to an integer multiple of 5 ms and must have a greater value than the heartbeat producer time of the node to be monitored.

Subindex	Meaning	Lenze code
0	Number of nodes to be monitored	
1 ... n	Node ID and heartbeat time of the node to be monitored	Node ID: C00385/x Heartbeat time: C00386/x

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
Bit 31 ... bit 24	Bit 23 ... bit 16	Bit 15 ... bit 0	
0 Reserved	Node ID	Heartbeat time in [ms]	

[13-4] Data telegram assignment

I-1017 - Producer heartbeat time

Index: I-1017	Name: Producer heartbeat time					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Producer heartbeat time	0	0	ms	65535	rw	U16

Time interval for the transmission of the heartbeat telegram to the consumer(s).

► [Heartbeat protocol](#) (66)

- The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is transmitted automatically as soon as a time > 0 ms is set. In this case, the node guarding monitoring function is deactivated.
- This object relates to code [C00381](#).

Implemented CANopen objects

I-1018 - Identity object

Index: I-1018	Name: Identity object					
Subindex	Default setting	Display range (min. value unit max. value)			Access	Data type
0: Highest subindex supported	See below	0		4294967295	ro	U32
1: Vendor ID						
2: Product code						
3: Revision number						
4: Serial number						

Subindex	Meaning
1	Manufacturer's identification number • The identification number allocated to Lenze by the organisation "CAN in Automation e. V." is "0x0000003B".
2	Product code
3	Main version and subversion of firmware
4	Serial number

I-1200 - SDO1 server parameter

Index: I-1200	Name: SDO1 server parameter					
Subindex	Default setting	Display range (min. value unit max. value)			Access	Data type
0: Highest subindex supported	2	2		2	ro	U8
1: COB-ID client -> server (rx)	Node ID + 0x600	0		4294967295	ro	U32
2: COB-ID server -> client (tx)	Node ID + 0x580	0		4294967295	ro	U32

Identifiers for SDO server channel 1 (basic SDO channel).

According to DS301 V4.02, the basic SDO channel can neither be changed nor deactivated.

Subindex	Meaning
1	Specification of the receive identifier • For SDO server channel 1: node address (C00350) + 0x600
2	Specification of the transmit identifier • For SDO server channel 1: node address (C00350) + 0x580

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0	0	Extended identifier*				11-bit identifier	
* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".							

[13-5] Data telegram assignment

Implemented CANopen objects

I-1201 - SDO2 server parameter

Index: I-1201	Name: SDO2 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Highest subindex supported	3	- (read access only)			ro	U8
1: COB-ID client -> server (rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (tx)	0x80000000	0		4294967295	rw	U32

Setting of the identifiers for SDO server channel 2.

- The server SDO parameter is only valid if bit 31 is set to "0" for both transmission directions (subindex 1 and 2).
- In the Lenze setting, SDO server channel 2 is deactivated (bit 31 = "1").
- The identifier may only be changed when the SDO is invalid (bit 31 = "1").

Subindex	Meaning
1	Specification of the receive identifier
2	Specification of the transmit identifier

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0/1	0	Extended identifier*				11-bit identifier	
* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".							

* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".

[13-6] Data telegram assignment

Bit	setting	
Bit 31	0	SDO is valid.
	1	SDO is invalid.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

Implemented CANopen objects

Example

Parameter data channel 2 of the inverter with the node address 4 is to be activated.

- For this, bit 31 must be set to the value "0" (SDO is valid) in subindexes 1 and 2 of object [I-1201](#).
- The master must send the two "write request" commands to the nodes via the basic SDO channel.

Identifier calculation

- Identifier (COB-ID) = basic identifier + node address (node ID)
- Basic identifier SDO2 from the master to the drive: 1600 (0x640)
→ identifier = 0x640 + 0x4 = 0x644
- Basic identifier SDO2 from the drive to the master: 1472 (0x5C0)
→ identifier = 0x5C0 + 0x4 = 0x5C4

Resulting data (data 1 ... data 4)

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0	0	Extended identifier = 0				11-bit identifier = 0x644	
0x00		0x00		0x06		0x44	

[13-7] Data telegram assignment for subindex 1

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0	0	Extended identifier = 0				11-bit identifier = 0x5C4	
0x00		0x00		0x05		0xC4	

[13-8] Data telegram assignment for subindex 2

User data assignment

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x23	0x01	0x12	0x01	0x44	0x06	0x00	0x00

[13-9] User data assignment for writing to subindex 1

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x23	0x01	0x12	0x02	0xC4	0x05	0x00	0x00

[13-10] User data assignment for writing to subindex 2

Implemented CANopen objects

I-1400 - RPDO1 communication parameter

Index: I-1400	Name: RPDO1 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Highest subindex supported	5	- (read access only)			ro	U8
1: COB-ID used by RPDO	0x200 + node ID	0		4294967295	rw	U32
2: Transmission type	254	0		255	rw	U8
3: Inhibit time	-	- (not used for RPDOs)			rw	U16
4: Compatibility entry	-	- (reserved, read or write access results in error message 0x06090011)			rw	U8
5: Event timer	-	- (not used for RPDOs)			rw	U16

Communication parameters for receiving process data via RPDO1

Subindex	Meaning	Code
0	The value 5 is permanently set. • Max. 5 subindexes are supported.	-
1	RPDO1 identifier • According to the "Predefined Connection Set", the basic setting is: identifier = 0x200 + node ID	C00354/1
2	RPDO transmission type according to DS301 V4.02 ► Transmission type (49)	C00323/1

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0/1	0/1	Extended identifier*				11-bit identifier	
* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".							

[13-11] Data telegram assignment

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

Description of subindex 1

Bit no.	Value	Explanation
0 ... 10	0/1	11-bit identifier
(11 ... 28)*	0	*) The extended identifier (29 bits) is not supported. Any of these bits must be "0".
29*	0	
30	0	RTR to this PDO permissible (cannot be set)
	1	RTR to this PDO not permissible (Lenze)
31	0	PDO active
	1	PDO not active

[13-12] I-1400 / I-1401, subindex 1

Implemented CANopen objects

Description of subindex 2

PDO transmission			Transmission type	Explanation
cyclic	synchronous	event-controlled		
●	●		n = 1 ... 240	When a value n is entered, this PDO will be accepted with every nth sync.
		●	n = 254	PDO will be accepted immediately.

[13-13] I-1400 / I-1401, subindex 2

I-1401 - RPDO2 communication parameter

Index: I-1401	Name: RPDO2 communication parameter				
Subindex	Default setting	Setting range (min. value unit max. value)			Access Data type
0: Highest subindex supported	5	- (read access only)			ro U8
1: COB-ID used by RPDO	0x300 + node ID	0		4294967295	rw U32
2: Transmission type	254	0		255	rw U8
3: Inhibit time	-	- (not used for RPDOs)			rw U16
4: Compatibility entry	-	- (reserved, read or write access results in error message 0x06090011)			rw U8
5: Event timer	-	- (not used for RPDOs)			rw U16

Communication parameters for receiving process data via RPDO2

Subindex	Meaning	Code
0	The value 5 is permanently set. • Max. 5 subindexes are supported.	-
1	RPDO2 identifier • According to the "Predefined Connection Set", the basic setting is: identifier = 0x300 + node ID	C00354/3
2	RPDO transmission type according to DS301 V4.02 ▶ Transmission type (□ 49)	C00323/2

For data telegram assignment and description of subindexes 1 and 2, see object [I-1400](#).

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

I-1600 - RPDO1 mapping parameter

Index: I-1600	Name: RPDO1 mapping parameter				
Subindex	Default setting	Setting range (min. value unit max. value)			Access Data type
0: Number of mapped application objects in PDO	0	0		8	rw U8
1 ... 4: Application object 1 ... 4	0	0		4294967295	rw U32

Object I-1600 serves to receive parameter data as RPDO1.

This object relates to code [C00409/1...4](#).

Implemented CANOpen objects

Subindex	Meaning
0	Number of mapped objects
1 ... 4	Mapping entries 1 ... 4 for RPDO1 The 4th mapping entry is used for the static mapping. For this, no value is available.

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
Bit 31 ... bit 16		Bit 15 ... bit 8	Bit 7 ... bit 0
Index		Subindex	Length

[13-14] Data telegram assignment

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

I-1601 - RPDO2 mapping parameter

Index: I-1601	Name: RPDO2 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 4: Application object 1 ... 4	0	0		4294967295	rw	U32

Object I-1601 serves to receive parameter data as RPDO2.

This object relates to code [C00409/5...8](#).

Subindex	Meaning
0	Number of mapped objects
1 ... 4	Mapping entries 1 ... 4 for RPDO2 The 4th mapping entry is used for the static mapping. For this, no value is available.

For assignment of the data telegram see object [I-1600](#).

I-1800 - TPDO1 communication parameter

Index: I-1800	Name: TPDO1 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Highest subindex supported	5	- (read access only)			ro	U8
1: COB-ID used by TPDO	0x180 + node ID	0		4294967295	rw	U32
2: Transmission type	254	0		255	rw	U8
3: Inhibit time	0 ms	0	0.1 ms	65535	rw	U16
4: Reserved	-	- (reserved, read or write access results in error message 0x06090011)			rw	U8
5: Event timer	0 ms	0	ms	65535	rw	U16

Communication parameters for sending process data via TPDO1

Implemented CANopen objects

Subindex	Meaning	Code
0	The value 5 is permanently set. • Max. 5 subindexes are supported.	-
1	TPDO1 identifier • According to the "Predefined Connection Set", the basic setting is: identifier = 0x180 + node ID	C00354/2
2	TPDO transmission type according to DS301 V4.02 ► Transmission type (□ 49)	C00322/1
3	Minimum time between sending two identical TPDOs (see DS301 V4.02).	-
5	Cycle time for PDO transmission with transmission type "254".	C00356/5 C00369

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0/1	0/1	Extended identifier*				11-bit identifier	
* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".							

[13-15] Data telegram assignment

Bit	setting	
Bit 30	0	RTR to this PDO permissible (Lenze).
	1	RTR to this PDO not permissible (cannot be set).
Bit 31	0	PDO active
	1	PDO inactive

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

Subindex 2 - transmission type

PDO transmission			Transmission type	Explanation
cyclic	synchronous	event-controlled		
•	•		n = 1 ... 240	When a value n is entered, this PDO will be accepted with every nth sync.
•		•	n = 254	Event-controlled or cyclic

Subindex 3 - inhibit time



Note!

The delay time can only be changed when the PDO is not active (see subindex 1, bit 31 = 1).

Implemented CANopen objects

The entered value multiplied by 0.1 gives the delay time in [ms]. Only integers will be considered, i.e. fractional numbers will be **rounded down** to integers.

Example :

- Value entered: 26
- Calculated time = 26×0.1 [ms] = 2.6 [ms] → delay time = 2 [ms]

Subindex 5 - event timer

For cyclic operation (transmission type 254), the cycle time for sending the process data object on the CAN bus can be set here:

The value entered corresponds to the time in [ms].

I-1801 - TPDO2 communication parameter

Index: I-1801	Name: TPDO2 communication parameter				
Subindex	Default setting	Setting range (min. value unit max. value)			Access Data type
0: Highest subindex supported	5	- (read access only)			ro U8
1: COB-ID used by TPDO	0x280 + node ID	0		4294967295	rw U32
2: Transmission type	254	0		255	rw U8
3: Inhibit time	0 ms	0	0.1 ms	65535	rw U16
4: Reserved	-	- (reserved, read or write access results in error message 0x06090011)			rw U8
5: Event timer	0 ms	0	ms	65535	rw U16

Communication parameters for sending process data via TPDO2

Subindex	Meaning	Code
0	The value 5 is permanently set. • Max. 5 subindexes are supported.	-
1	TPDO2 identifier • According to the "Predefined Connection Set", the basic setting is: identifier = 0x280 + node ID	C00354/4
2	TPDO transmission type according to DS301 V4.02 ▶ Transmission type (□ 49)	C00322/2
3	Minimum time between sending two identical TPDOs (see DS301 V4.02).	-
5	Cycle time for PDO transmission with transmission type "254".	C00356/2 C00369

For data telegram assignment and description of subindexes, see object [I-1800](#).

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

Implemented CANOpen objects

I-1A00 - TPDO1 mapping parameter

Index: I-1A00	Name: TPDO1 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 4: Application object 1 ... 4	0	0		4294967295	rw	U32

Object I-1A00 serves to send parameter data as TPDO1.

Subindex	Meaning
0	Number of mapped objects
1 ... 4	Mapping entries 1 ... 4 for TPDO1 The 4th mapping entry is used for the static mapping. For this, no value is available.

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
Bit 31 ... bit 16		Bit 15 ... bit 8	Bit 7 ... bit 0
Index		Subindex	Length

[13-16] Data telegram assignment

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

I-1A01 - TPDO2 mapping parameter

Index: I-1A01	Name: TPDO2 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 4: Application object 1 ... 4	0	0		4294967295	rw	U32

Object I-1A01 serves to send parameter data as TPDO2.

Subindex	Meaning
0	Number of mapped objects
1 ... 4	Mapping entries 1 ... 4 for TPDO2 The 4th mapping entry is used for the static mapping. For this, no value is available.

For assignment of the data telegram see object [I-1A00](#).

14 DIP switch positions for setting the CAN node address

The node address results from the sum of the binary values of switches 1 ... 64.

The following table shows the switch positions for the valid address range of 1 ... 63.

► [Setting the CAN node address](#) (31)

Station address	DIP switch						
	64	32	16	8	4	2	1
1	OFF	OFF	OFF	OFF	OFF	OFF	ON
2	OFF	OFF	OFF	OFF	OFF	ON	OFF
3	OFF	OFF	OFF	OFF	OFF	ON	ON
4	OFF	OFF	OFF	OFF	ON	OFF	OFF
5	OFF	OFF	OFF	OFF	ON	OFF	ON
6	OFF	OFF	OFF	OFF	ON	ON	OFF
7	OFF	OFF	OFF	OFF	ON	ON	ON
8	OFF	OFF	OFF	ON	OFF	OFF	OFF
9	OFF	OFF	OFF	ON	OFF	OFF	ON
10	OFF	OFF	OFF	ON	OFF	ON	OFF
11	OFF	OFF	OFF	ON	OFF	ON	ON
12	OFF	OFF	OFF	ON	ON	OFF	OFF
13	OFF	OFF	OFF	ON	ON	OFF	ON
14	OFF	OFF	OFF	ON	ON	ON	OFF
15	OFF	OFF	OFF	ON	ON	ON	ON
16	OFF	OFF	ON	OFF	OFF	OFF	OFF
17	OFF	OFF	ON	OFF	OFF	OFF	ON
18	OFF	OFF	ON	OFF	OFF	ON	OFF
19	OFF	OFF	ON	OFF	OFF	ON	ON
20	OFF	OFF	ON	OFF	ON	OFF	OFF
21	OFF	OFF	ON	OFF	ON	OFF	ON
22	OFF	OFF	ON	OFF	ON	ON	OFF
23	OFF	OFF	ON	OFF	ON	ON	ON
24	OFF	OFF	ON	ON	OFF	OFF	OFF
25	OFF	OFF	ON	ON	OFF	OFF	ON
26	OFF	OFF	ON	ON	OFF	ON	OFF
27	OFF	OFF	ON	ON	OFF	ON	ON
28	OFF	OFF	ON	ON	ON	OFF	OFF
29	OFF	OFF	ON	ON	ON	OFF	ON
30	OFF	OFF	ON	ON	ON	ON	OFF
31	OFF	OFF	ON	ON	ON	ON	ON
32	OFF	ON	OFF	OFF	OFF	OFF	OFF
33	OFF	ON	OFF	OFF	OFF	OFF	ON
34	OFF	ON	OFF	OFF	OFF	ON	OFF
35	OFF	ON	OFF	OFF	OFF	ON	ON
36	OFF	ON	OFF	OFF	ON	OFF	OFF
37	OFF	ON	OFF	OFF	ON	OFF	ON
38	OFF	ON	OFF	OFF	ON	ON	OFF

DIP switch positions for setting the CAN node address

Station address	DIP switch						
	64	32	16	8	4	2	1
39	OFF	ON	OFF	OFF	ON	ON	ON
40	OFF	ON	OFF	ON	OFF	OFF	OFF
41	OFF	ON	OFF	ON	OFF	OFF	ON
42	OFF	ON	OFF	ON	OFF	ON	OFF
43	OFF	ON	OFF	ON	OFF	ON	ON
44	OFF	ON	OFF	ON	ON	OFF	OFF
45	OFF	ON	OFF	ON	ON	OFF	ON
46	OFF	ON	OFF	ON	ON	ON	OFF
47	OFF	ON	OFF	ON	ON	ON	ON
48	OFF	ON	ON	OFF	OFF	OFF	OFF
49	OFF	ON	ON	OFF	OFF	OFF	ON
50	OFF	ON	ON	OFF	OFF	ON	OFF
51	OFF	ON	ON	OFF	OFF	ON	ON
52	OFF	ON	ON	OFF	ON	OFF	OFF
53	OFF	ON	ON	OFF	ON	OFF	ON
54	OFF	ON	ON	OFF	ON	ON	OFF
55	OFF	ON	ON	OFF	ON	ON	ON
56	OFF	ON	ON	ON	OFF	OFF	OFF
57	OFF	ON	ON	ON	OFF	OFF	ON
58	OFF	ON	ON	ON	OFF	ON	OFF
59	OFF	ON	ON	ON	OFF	ON	ON
60	OFF	ON	ON	ON	ON	OFF	OFF
61	OFF	ON	ON	ON	ON	OFF	ON
62	OFF	ON	ON	ON	ON	ON	OFF
63	OFF	ON	ON	ON	ON	ON	ON

15 Index

A

Accessing process data [44](#)
Acknowledgement error [66](#)
Active COBID (C00355) [78](#)
Application as directed [13](#)
Application notes (representation) [10](#)
Application of the Communication Unit [13](#)
Approvals [18](#)

B

Baud rate [18](#)
Before initial switch-on [29](#)
Bit error [66](#)
Bus cable [25](#)
Bus cable length [25](#)
Bus cable specification [25](#)
Bus termination [24](#)

C

C00322 | Transmission mode CAN TxPDOs [74](#)
C00323 | Transmission mode CAN Rx PDOs [74](#)
C00324 | CAN transmission blocking time [75](#)
C00345 | CAN error status [75](#)
C00347 | CAN status HeartBeat producer [76](#)
C00349 | CAN setting, DIP switch [76](#)
C00350 | CAN node address [77](#)
C00351 | CAN baud rate [77](#)
C00352 | CAN slave/master [77](#)
C00353 | CAN IN/OUT COBID source [78](#)
C00354 | COBID [78](#)
C00355 | Active COBID [78](#)
C00356 | CAN time settings [79](#)
C00357 | CAN monitoring times [79](#)
C00359 | CAN status [79](#)
C00360 | CAN telegram counter [80](#)
C00364 | CAN MessageError [80](#)
C00366 | Number of CAN SDO channels [81](#)
C00367 | CAN sync Rx identifier [81](#)
C00368 | CAN sync Tx identifier [81](#)
C00369 | CAN sync transmission cycle time [81](#)
C00372 | CAN_Tx_Rx_Error [82](#)
C00381 | CAN Heartbeat Producer Time [82](#)
C00385 | CAN NodeID heartbeat producer [82](#)
C00386 | ConsumerTime HeartBeat producer [82](#)
C00389 | PDO valid / invalid [83](#)
C00409 | LP_CanIn Mapping [83](#)
C00592 | Resp. to CAN bus connection [84](#)
C00593 | Resp. to CANx_IN monitoring [84](#)
C01501 | Resp. to communication error with MCI [73](#)
C01503 | MCI timeout [73](#)
CAN baud rate (C00351) [77](#)

CAN cable according to ISO 11898-2 [25](#)
CAN data telegram [35](#)
CAN error status (C00345) [75](#)
CAN heartbeat producer time (C00381) [82](#)
CAN IN/OUT COBID source (C00353) [78](#)
CAN MessageError (C00364) [80](#)
CAN monitoring times (C00357) [79](#)
CAN node address (C00350) [77](#)
CAN NodeID heartbeat producer (C00385) [82](#)
CAN setting - DIP switch (C00349) [76](#)
CAN slave/master (C00352) [77](#)
CAN start remote node [41](#)
CAN status (C00359) [79](#)
CAN status HeartBeat producer (C00347) [76](#)
CAN sync Rx identifier (C00367) [81](#)
CAN sync transmission cycle time (C00369) [81](#)
CAN sync Tx identifier (C00368) [81](#)
CAN telegram counter (C00360) [80](#)
CAN time settings (C00356) [79](#)
CAN transmission blocking time (C00324) [75](#)
CAN_Tx_Rx_Error (C00372) [82](#)
CANopen connection [28](#)
CANopen objects (indexes) [87](#)
Checking the use of repeaters [26](#)
COB-ID [36](#)
COBID (C00354) [78](#)
COB-ID EMCY (I-1014) [91](#)
COB-ID SYNC message (I-1005) [90](#)
Codes [73](#)
Commissioning [29](#)
Communication cycle period (I-1006) [91](#)
Communication medium [18](#)
Communication profile [18](#)
Communication time [20](#)
Communication-relevant parameters of the operating system [73](#)
Configuration of the master [30](#)
Conformities [18](#)
Connections [16](#)
Consumer heartbeat time (I-1016) [92](#)
ConsumerTime HeartBeat producer (C00386) [82](#)
Conventions [8](#)
Conventions used [8](#)
CRC error [66](#)

D

Data transfer [35](#)
Device and application-specific safety instructions [12](#)
Device protection [12](#)
Device type (I-1000) [88](#)
Diagnostics [72](#)
Diagnostics with the »Engineer« [72](#)
DIP switch positions for setting the CAN node address [102](#)

DIP switch settings [31](#), [102](#)
Document history [7](#)

E

Electrical installation [23](#)
E-mail to Lenze [110](#)
Emergency [71](#)
Error detection [66](#)
Error messages (system bus) [58](#)
Error register (I-1001) [89](#)
Establishing communication [34](#)

F

Feedback to Lenze [110](#)
Format error [66](#)

G

General data [18](#)
General safety and application notes [11](#)

H

Heartbeat protocol [67](#)
How to configure the host (master) [30](#)
How to configure the port interconnection in the »Engineer« [45](#)

I

I-1000 [88](#)
I-1001 [89](#)
I-1003 [89](#)
I-1005 [90](#)
I-1006 [91](#)
I-1014 [91](#)
I-1016 (Consumer heartbeat time) [92](#)
I-1017 [92](#)
I-1018 [93](#)
I-1200 (SDO1 server parameter) [93](#)
I-1201 (SDO2 server parameter) [94](#)
I-1400 (RPDO1 communication parameter) [96](#)
I-1401 (RPDO2 communication parameter) [97](#)
I-1600 (RPDO1 mapping parameter) [97](#)
I-1601 (RPDO2 mapping parameter) [98](#)
I-1800 (TPDO1 communication parameter) [98](#)
I-1801 (TPDO2 communication parameter) [100](#)
I-1A00 (TPDO1 mapping parameter) [101](#)
I-1A01 (TPDO2 mapping parameter) [101](#)
Identifier (CAN) [36](#)
Identifiers of the parameter data objects [54](#)
Identifiers of the process data objects [49](#)
Identity object (I-1018) [93](#)
Implemented CANopen objects [87](#)
Initial switch-on [34](#)
Installation [21](#)
Integrated error detection [66](#)

Interface [18](#)
Interfaces [16](#)

L

LP_CanIn mapping (C00409) [83](#)

M

Master functionality (CAN) [41](#)
MCI timeout (C01503) [73](#)
Mechanical installation [22](#)

N

Network management data [37](#)
Network management telegram (NMT) [40](#)
Network topology [18](#), [23](#)
NMT (network management) [40](#)
Node address [18](#), [36](#)
Node ID [36](#)
Notes used [10](#)
Number of CAN SDO channels (C00366) [81](#)
Number of nodes [18](#)

O

Operating conditions [18](#)
Overview [87](#)

P

Parameter [73](#)
Parameter data [18](#), [37](#)
Parameter data transfer [53](#)
Parameters for CANopen communication [74](#)
PDO mapping [44](#)
PDO synchronisation [52](#)
PDO valid / invalid (C00389) [83](#)
Pre-defined error field (I-1003) [89](#)
Process data [18](#), [37](#), [42](#)
Process data objects, identifiers [49](#)
Process data transfer [42](#)
Processing time [20](#)
Producer heartbeat time (I-1017) [92](#)
Product description [13](#)
Product features [14](#)

R

Residual hazards [12](#)
Resp. to CAN bus connection (C00592) [84](#)
Resp. to CANx_IN monitoring (C00593) [84](#)
Resp. to communication error with MCI (C01501) [73](#)
RPDO1 communication parameter (I-1400) [96](#)
RPDO1 mapping parameter (I-1600) [97](#)
RPDO2 communication parameter (I-1401) [97](#)
RPDO2 mapping parameter (I-1601) [98](#)

S

Safety instructions [11](#)
Safety instructions (representation) [10](#)
SDO1 server parameter (I-1200) [93](#)
SDO2 server parameter (I-1201) [94](#)
Segment cable length [26](#)
Setting the baud rate [31](#)
Setting the CAN node address [32](#)
Settings in the Lenze »Engineer« [33](#)
Stuff-bit error [66](#)
Supported protocols [19](#)
Sync telegram [52](#)

T

Table of attributes [85](#)
Target group [6](#)
Technical data [18](#)
Terminology used [9](#)
Terms [9](#)
Total cable length [25](#)
TPDO1 communication parameter (I-1800) [98](#)
TPDO1 mapping parameter (I-1A00) [101](#)
TPDO2 communication parameter (I-1801) [100](#)
TPDO2 mapping parameter (I-1A01) [101](#)
Transmission mode CAN Rx PDOs (C00323) [74](#)
Transmission mode CAN TxPDOs (C00322) [74](#)
Transmission mode for TPDOs [18](#)
Transmission type [50](#)

U

User data [37](#), [55](#)
User data length [30](#)

V

Validity of the documentation [6](#)
Versions [14](#)



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:

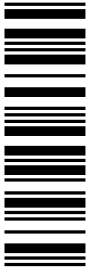
feedback-docu@lenze.com

Thank you very much for your support.

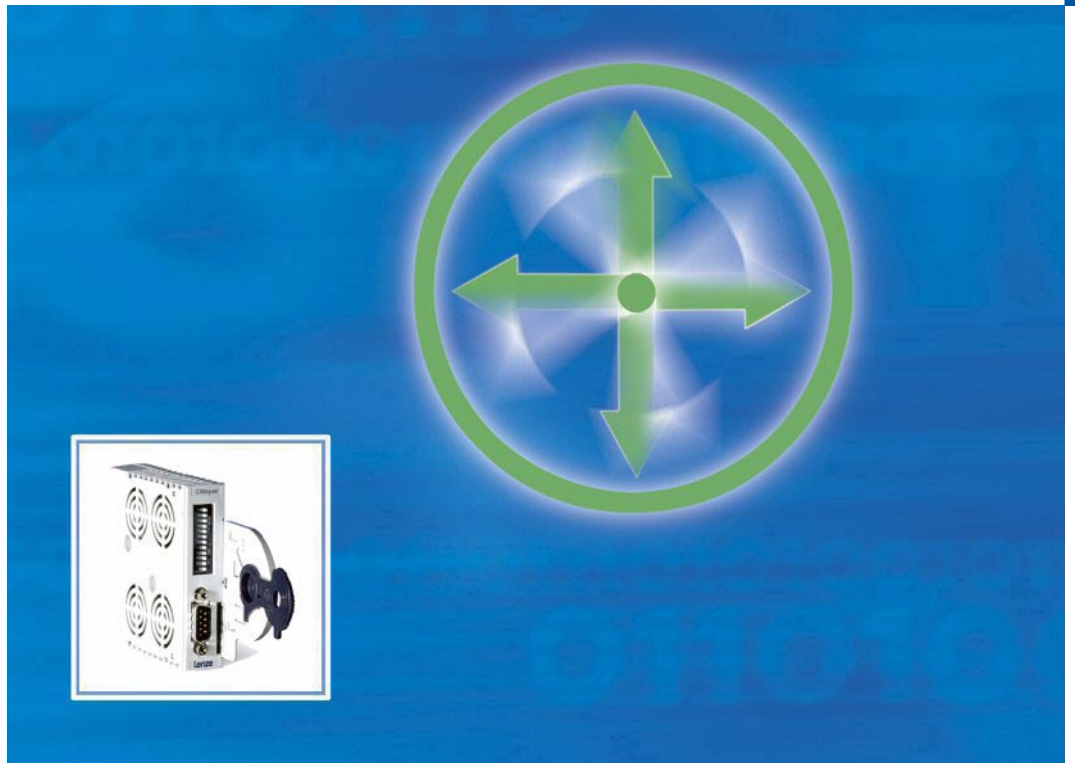
Your Lenze documentation team

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



9400



E94AYCCA

CANopen® communication module

Contents

1	About this documentation	6
1.1	Document history	8
1.2	Conventions used	9
1.3	Terminology used	10
1.4	Notes used	11
2	Safety instructions	12
2.1	General safety and application notes	12
2.2	Device and application-specific safety instructions	13
2.3	Residual hazards	13
3	Product description	14
3.1	Application as directed	14
3.2	Identification	14
3.3	Features	15
3.4	Terminals and interfaces	16
4	Technical data	17
4.1	General data and operating conditions	17
4.2	Supported protocols	18
4.3	Communication time	19
4.4	Protective insulation	20
4.5	Dimensions	23
5	Installation	24
5.1	Mechanical installation	25
5.1.1	Mounting	25
5.1.2	Dismounting	25
5.2	Electrical installation	26
5.2.1	System bus (CANopen) connection	26
5.2.2	Specification of the bus cable	27
5.2.3	Bus cable length	28
5.2.4	Voltage supply	31

6	<u>Commissioning</u>	32
6.1	<u>Before initial switch-on</u>	32
6.2	<u>Possible settings via DIP switches</u>	33
6.2.1	<u>Setting the node address</u>	33
6.2.2	<u>Setting the baud rate</u>	34
6.3	<u>Settings in the »Engineer«</u>	35
6.4	<u>Initial switch-on</u>	36
7	<u>Data transfer</u>	37
7.1	<u>Structure of the CAN data telegram</u>	37
7.1.1	<u>Identifier</u>	38
7.1.2	<u>User data</u>	40
7.2	<u>Communication phases / network management</u>	41
7.2.1	<u>State transitions</u>	42
7.2.2	<u>Network management telegram (NMT)</u>	43
7.2.3	<u>Parameterising the controller as a CAN master</u>	44
8	<u>Process data transfer</u>	45
8.1	<u>Identifiers of the process data objects</u>	46
8.2	<u>Transmission type</u>	47
8.3	<u>Masking of the TPDOs for event control</u>	49
8.4	<u>Synchronisation of PDOs via sync telegram</u>	50
8.4.1	<u>Parameter setting</u>	51
8.4.2	<u>Effect of C01130 on the sync phase position</u>	54
8.5	<u>PDO mapping</u>	56
9	<u>Parameter data transfer</u>	61
9.1	<u>Identifiers of the parameter data objects</u>	62
9.2	<u>User data</u>	63
9.2.1	<u>Command</u>	64
9.2.2	<u>Addressing through index and subindex</u>	65
9.2.3	<u>Data 1 ... data 4</u>	66
9.2.4	<u>Error messages</u>	67
9.3	<u>Examples for a parameter data telegram</u>	69
9.3.1	<u>Reading parameters</u>	69
9.3.2	<u>Writing parameters</u>	70
9.3.3	<u>Reading block parameters</u>	71

10	<u>Monitoring</u>	74
10.1	<u>Node guarding protocol</u>	74
10.1.1	<u>Telegram structure</u>	75
10.1.2	<u>Parameter setting</u>	76
10.1.3	<u>Commissioning example</u>	78
10.2	<u>Heartbeat protocol</u>	80
10.2.1	<u>Telegram structure</u>	81
10.2.2	<u>Parameter setting</u>	81
10.2.3	<u>Commissioning example</u>	83
10.3	<u>Emergency telegram</u>	84
10.4	<u>Settings in the »Engineer«</u>	85
11	<u>Diagnostics</u>	86
11.1	<u>LED status displays</u>	86
11.2	<u>Diagnostics with the »Engineer«</u>	88
12	<u>Error messages</u>	89
12.1	<u>Short overview of the CANopen error messages</u>	89
12.2	<u>Possible causes and remedies</u>	91
13	<u>Parameter reference</u>	101
13.1	<u>Communication-relevant parameters of the standard device</u>	101
13.2	<u>Parameters of the communication module for slot MXI1</u>	107
13.3	<u>Parameters of the communication module for slot MXI2</u>	128
13.4	<u>Table of attributes</u>	149
14	<u>Implemented CANopen objects</u>	153
15	<u>DIP switch positions for setting the CAN node address</u>	178
16	<u>Index</u>	182

1 About this documentation

Contents

This documentation solely contains descriptions for the E94AYCCA communication module (CANopen®).



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 product features and functions of the communication module are described in detail. Examples illustrate typical applications.

This documentation also contains ...

- ▶ safety instructions that must be observed;
- ▶ key technical data relating to the communication module;
- ▶ information about the versions of the Lenze standard devices to be used;
- ▶ notes on troubleshooting and fault elimination.

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 about how to use the software, please refer to the host (PLC, master) documents.

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



Tip!

Detailed information about the CAN/CANopen bus system can be found on the website of the CAN user organisation CiA® (CAN in Automation):

www.can-cia.org

Screenshots/application examples

All screenshots in this documentation are application examples. Depending on the firmware version of the field devices and the software version of the Engineering tools installed (e.g. »Engineer«), the screenshots in this documentation may deviate from the actual screens shown.

Target group

This documentation is intended for people involved in configuring, installing, commissioning, and maintaining the networking and remote maintenance of a machine.



Tip!

Current documentation and software updates for 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
CANopen communication manual	E94AYCCA	VA	01.00

1.1 Document history

Version			Description
1.0	08/2006	TD17	First edition
2.0	04/2008	TD17	General revision
3.0	02/2011	TD17	General revision
4.0	11/2011	TD17	Assignment of the 9-pin Sub-D plug connector corrected. ▶ System bus (CANopen) connection (📖 26)
5.0	06/2012	TD17	General revision

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 different types of information:

Information type	Display	Examples/notes
Spelling of numbers		
Decimal	Standard 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 generally used. For example: 1234.56
Text		
Program name	» «	PC software Example: Lenze »Engineer«
Control element	Bold	The OK button... / the copy command... / the Properties tab... / the Name input field...
Sequence of menu commands		If the execution of a command requires several commands in succession, the individual commands are separated by an arrow: Select the File→Open command to...
Hyperlink	<u>Underlined</u>	Optically highlighted reference to another subject which is activated with a mouse-click.
Symbols		
Page reference	 9	Optically highlighted reference to another page which is activated with a mouse-click.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

1.3 Terminology used

Term	Meaning
CAN	CAN (Controller Area Network) is an asynchronous, serial fieldbus system.
CANopen®	CANopen® is a communication protocol based on CAN. CANopen® is a trademark and patented technology, licensed by the CAN user organisation CiA® (CAN in Automation e. V.), www.can-cia.org .
System bus (CAN/CANopen)	The Servo Drives 9400 system bus (CANopen) is an advancement of the 9300 controller series' system bus (CAN). ▶ Features (15)
Standard device	Lenze controller of the "Servo Drives 9400" product series, with which the communication module can be used. ▶ Application as directed (14)
Controller	
Module	Accessories and extension modules such as communication modules, distributed terminals, operator and input devices (HMI) as well as external controls and control systems
Interface module	
Communication module	
»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«	
Code	Parameter by means of which you can parameterise or monitor the controller. In everyday language, the term is also 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".
Lenze setting	Default settings for the device.
Basic setting	
PDO	Process data object (process data communication)
SDO	Service data object (parameter data communication)
HW	Hardware
SW	Software
PLC	Programmable Logic Controller
Host	CAN master
Control	
Node	Components of a CAN network (PLC, controller, etc.)
Bus node	
Node address	Unique ID for addressing the individual components of a CAN network
I-1600.20	CANopen index (hexadecimal representation) • In the example: index 0x1600, subindex 0x20



Note!

Some of the terms used derive from the CANopen protocol. These terms are not listed here.

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 the safety instructions:



Pictograph and signal word!

(characterise the type and severity of danger)

Note

(describes the danger and provides 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 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 and application notes



Danger!

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

- ▶ Lenze drive and automation components ...
 - may only be used as directed.
 - ▶ [Application as directed](#) (14)
 - must never be commissioned if they display any signs of damage.
 - must never be modified technically.
 - must never be commissioned if they are not fully mounted.
 - must never be operated without the required covers.
 - can have live, moving and rotating parts during and after operation, depending on their degree of protection. Surfaces can be hot.
- ▶ For Lenze drive components ...
 - only use approved accessories.
 - only use genuine spare parts supplied by the manufacturer of the product.
- ▶ Observe all the specifications contained in the enclosed and corresponding documentation.
 - This is a precondition for ensuring safe, trouble-free operation and for making use of the stated product features.
 - ▶ [Features](#) (15)
 - 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 may only be carried out by qualified personnel. According to IEC 60364 and CENELEC HD 384 these are persons who ...
 - are familiar with installing, mounting, commissioning, and operating the product.
 - have the qualifications necessary for their occupation.
 - know and are able to apply all regulations for the prevention of accidents, directives and laws that apply to the location 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.
 - ▶ [Specification of the bus cable](#) (📖 27)



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 stated in this manual.

2.3 Residual hazards

Protection of persons

- ▶ If Servo Drives 9400 are used on a corner grounded system with a rated mains voltage $\geq 400\text{V}$, external measures need to be implemented to provide reliable protection against accidental contact.
 - ▶ [Protective insulation](#) (📖 20)

Device protection

- ▶ The communication module contains electronic components that can be damaged or destroyed by electrostatic discharge.
 - ▶ [Installation](#) (📖 24)

3 Product description

3.1 Application as directed

The communication module ...

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

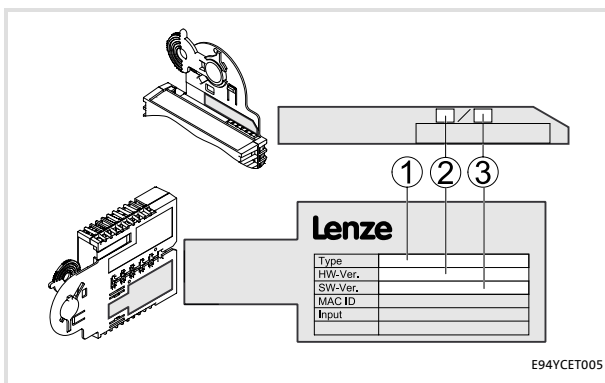
Product series	Type designation	From hardware version	From software version
Servo Drives 9400 HighLine	E94AxHExxx	VA	01.30
Servo Drives 9400 PLC	E94AxPExxx	VA	01.00
Servo Drives 9400 regenerative power supply module	E94ARNxxxx	VA	01.00

- 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 CAN networks.

Any other use shall be deemed inappropriate!

3.2 Identification

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



1 Type designation (type)

E94 Product series

A Version

Y Module identification: extension module

C Module type: communication module

CA CANopen

2 Hardware version (HW)

3 Software version (SW)

[3-1] Identification data

3.3 Features

For many years the system bus (CAN) based on the CANopen communication profile has been integrated in Lenze controllers. Due to the lower number of data objects available, the functionality and compatibility of the old system bus are lower as compared with CANopen. For parameter setting, two parameter data channels are always available to the user while CANopen provides only one active parameter data channel (along with the possibility to establish further channels).

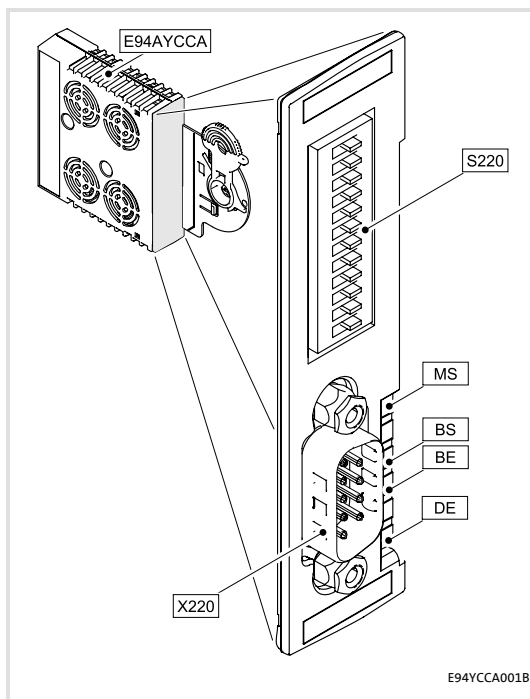
The system bus (CANopen) of the Servo Drives 9400 is an advancement of the 9300 controller series' system bus (CAN).

The communication module E94AYCCA (CANopen) provides the following features:

- ▶ Interface module for the system bus (CANopen), connectable to the expansion slots of Servo Drives 9400
- ▶ Internal voltage supply of the communication module via the Servo Drive 9400.
- ▶ Full compatibility according to CANopen DS301, V4.02
- ▶ Support of the NMT master/slave function "node guarding" (DS301,V4.02)
- ▶ Support of the NMT slave function "heartbeat" (DS301,V4.02)
- ▶ No restrictions regarding the selection of node addresses
- ▶ Number of parameterisable server and client SDO channels:
 - max. 10 channels with 1 ... 8 bytes
- ▶ Number of parameterisable PDO channels:
 - max. 4 Transmit-PDOs (TPDOs) with 1 ... 8 bytes
 - max. 4 Receive-PDOs (RPDOs) with 1 ... 8 bytes
- ▶ All PDO channels are functionally equivalent.
- ▶ Monitoring of the RPDOs for data reception
- ▶ Telegram counters for SDOs and PDOs
- ▶ Bus status diagnostics
- ▶ Boot-up telegram generation
- ▶ Emergency telegram generation
- ▶ Reset node telegram generation (with master configuration)
- ▶ Sync telegram generation and response to sync telegrams:
 - Transmit/receive data
 - Synchronisation of the device-internal time base
- ▶ Abort codes
- ▶ All CAN functions parameterisable via codes
- ▶ Object directory (all mandatory functions, optional functions, indexes)

3.4 Terminals and interfaces

- ▶ 9-pin Sub-D plug connector for system bus connection (CANopen)
- ▶ DIP switches for ...
 - setting the CAN node address
 - setting the baud rate
- ▶ Front LEDs for diagnosing the ...
 - voltage supply of the communication module;
 - connection to the standard device;
 - CANopen state machine and error states;
 - physical CAN connection.



X220 9-pin Sub-D plug connector for the ...
▶ [System bus \(CANopen\) connection](#) (26)

S220 DIP switches for ...

- setting the CAN node address
- setting the baud rate

▶ [Possible settings via DIP switches](#) (33)

MS LED status displays for diagnostics

BS ▶ [LED status displays](#) (86)

BE

DE

[3-2] E94AYCCA communication module (CANopen)

4 Technical data

4.1 General data and operating conditions

Field	Values
Order designation	E94AYCCA
Communication profile	CANopen (DS301, V4.02)
Communication medium	CAN cable according to ISO 11898-2
Network topology	Line terminated on both sides (e.g. termination with Lenze system connector EWZ0046)
Adjustable node addresses	1 ... 127 Adjustable via DIP switches or code C13350 / C14350 .
Max. number of nodes	127
Baud rates [kbps]	• 10 • 20 • 50 • 125 • 250 • 500 • 800 • 1000 Adjustable via DIP switches or code C13351 / C14351 .
Process data	• max. 4 TPDOs with 1 ... 8 bytes • max. 4 RPDOs with 1 ... 8 bytes
Parameter data	Max. 10 client and server SDO channels with 1 ... 8 bytes
Transfer mode for TPDOs	• With change of data • Time-controlled, 1 to x ms • After the reception of 1 to 240 sync telegrams
Voltage supply	Internal voltage supply of the communication module via the Servo Drive 9400.
Conformities, approvals	• CE • UL



"Servo Drives 9400" hardware manual

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

4.2 Supported protocols

Category	Protocol
Standard PDO protocols	PDO write PDO read
SDO protocols	SDO download SDO download initiate SDO download segment SDO upload SDO upload initiate SDO upload segment SDO abort transfer SDO block download SDO block download initiate SDO block download end SDO block upload SDO block upload initiate SDO block upload end
NMT protocols	Start remote node (master and slave) Stop remote node (slave) Enter pre-operational (slave) Reset node (slave and local device) Reset communication (slave)
Monitoring protocols	Node guarding (master and slave) Heartbeat (heartbeat producer and heartbeat consumer)

4.3 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 CAN network depend on the ...

- ▶ processing time in the device;
- ▶ telegram runtime (baud rate/telegram length);
- ▶ bus load (especially if the bus is loaded with PDOs and SDOs at a low baud rate).

Servo Drives 9400 processing time

There are no interdependencies between parameter data and process data.

- ▶ Parameter data:
 - For controller-internal parameters: approx. 30 ms ± 20 ms tolerance (typically)
 - For some codes the processing time can be longer.
- ▶ Process data are transported in real time.

4.4 Protective insulation



Danger!

Dangerous electrical voltage

If Servo Drives 9400 are used on a corner grounded system with a rated mains voltage ≥ 400 V, external measures need to be implemented to provide reliable protection against accidental contact.

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

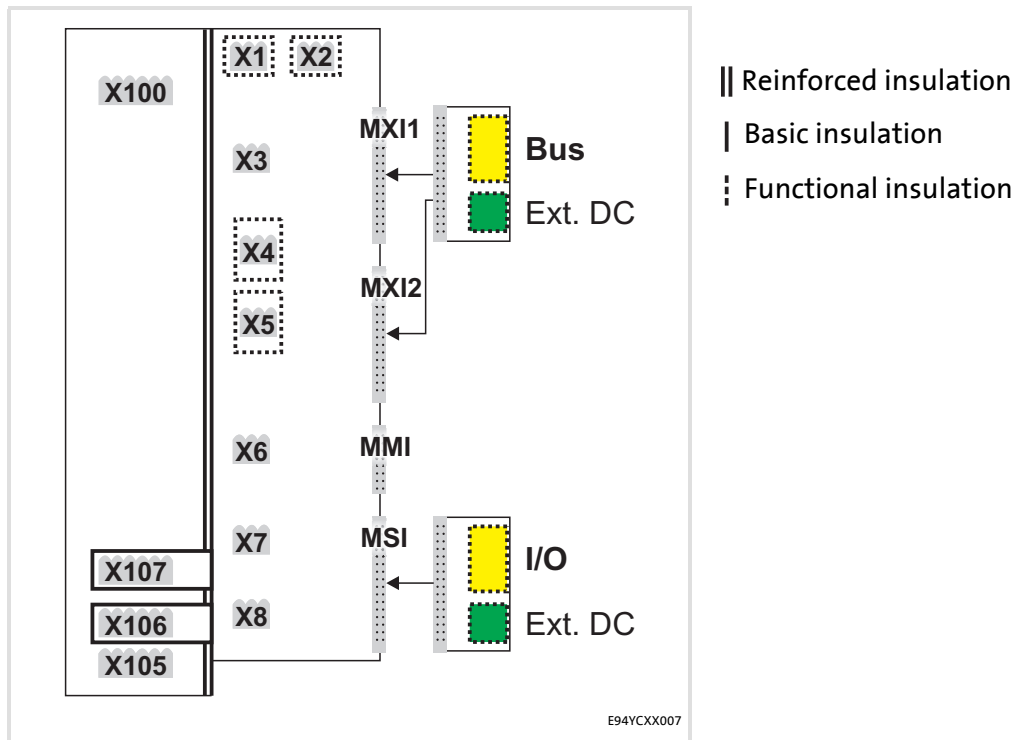


Note!

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

The illustration below ...

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



[4-1] Protective insulation according to EN61800-5-1

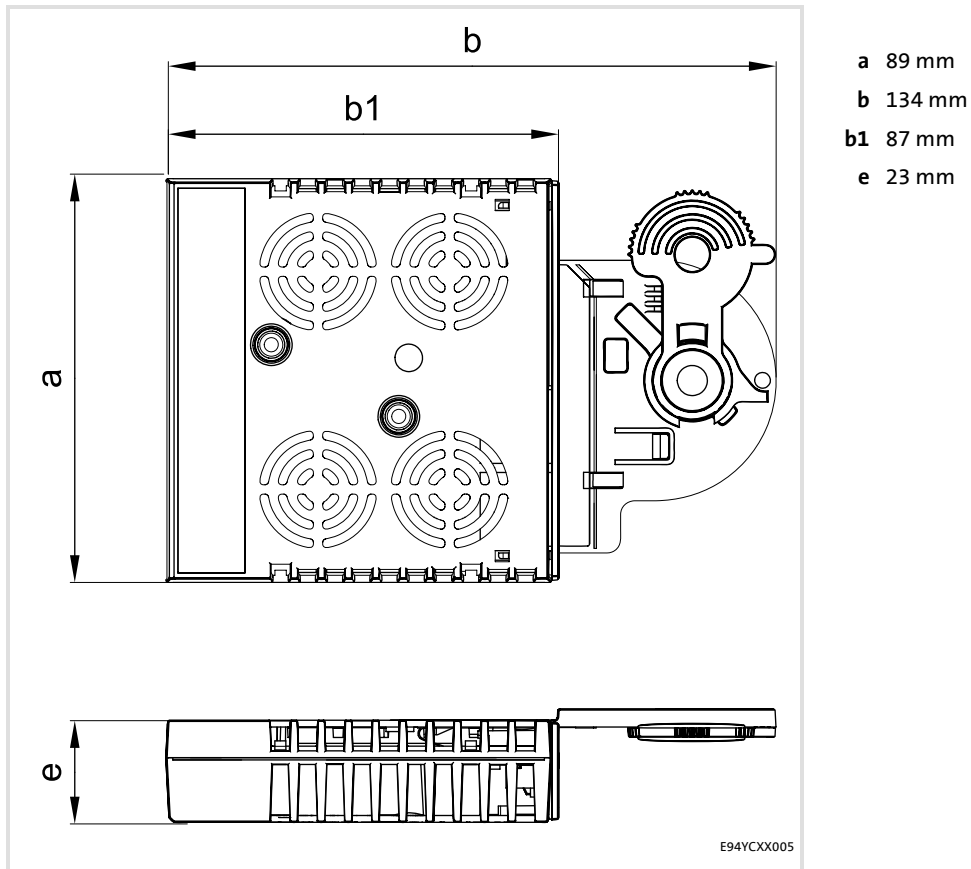
Terminal strip	Connection	Terminal strip	Connection
X100	L1, L2, L3 (only Single Drive)	X1	CAN on board 9400
	+UG, -UG	X2	State bus
X105	U, V, W		24 V (ext.)
	Rb1, Rb2 (Single Drive only)	X3	Analog inputs/outputs
X106	Motor PTC	X4	Digital outputs
X107	Control of the motor holding brake	X5	Digital inputs
		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 "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.

4.5 Dimensions



[4-2] Dimensions

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.

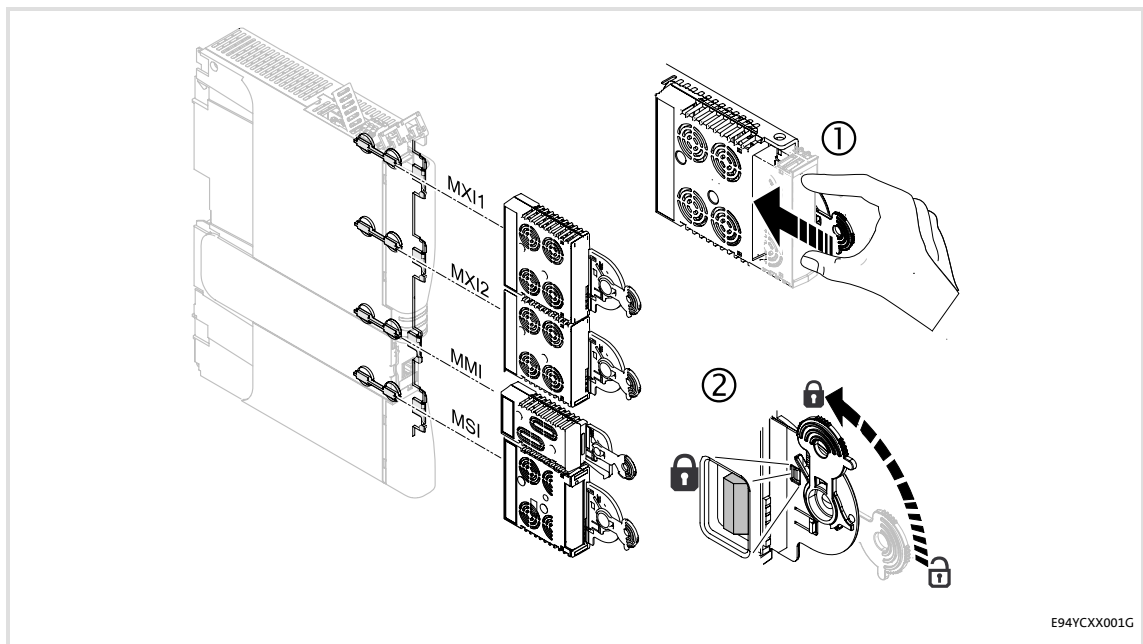
5.1 Mechanical installation



Note!

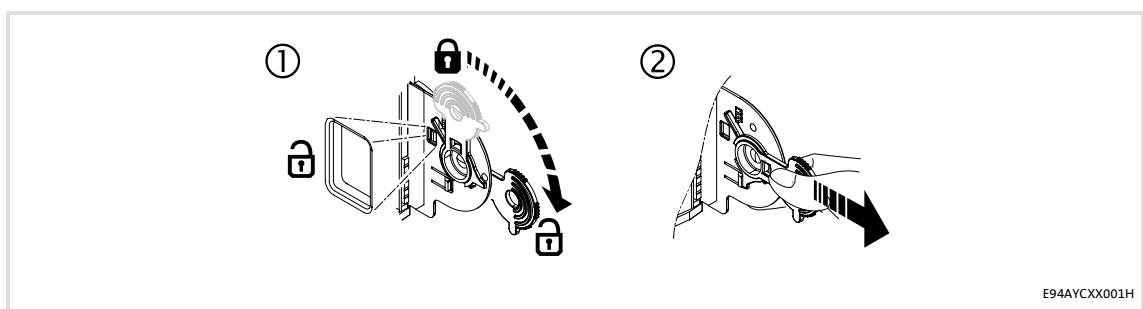
- Only one communication module E94AYCCA (CANopen) may be inserted per Servo Drive 9400.
- The slot used (MX1 or MX2) can be selected freely.

5.1.1 Mounting



[5-1] Mounting

5.1.2 Dismounting



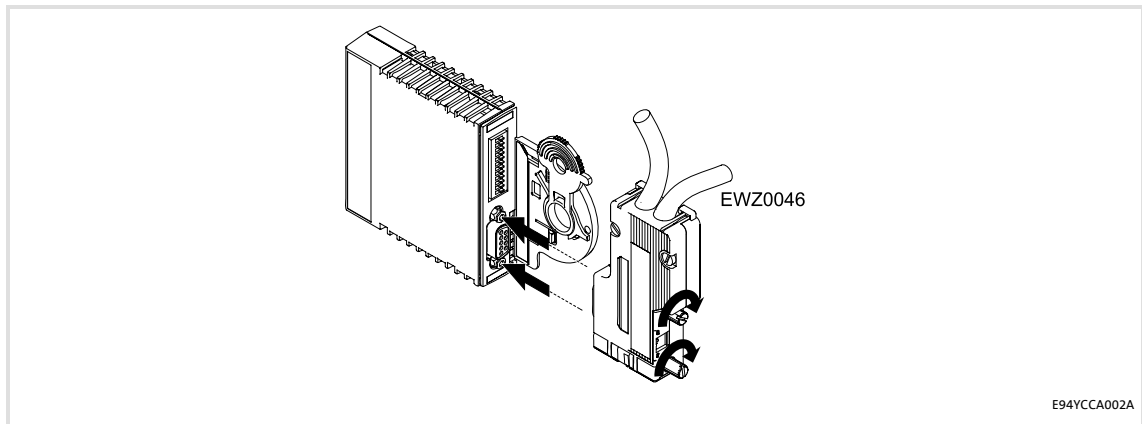
[5-2] Dismounting

5.2 Electrical installation

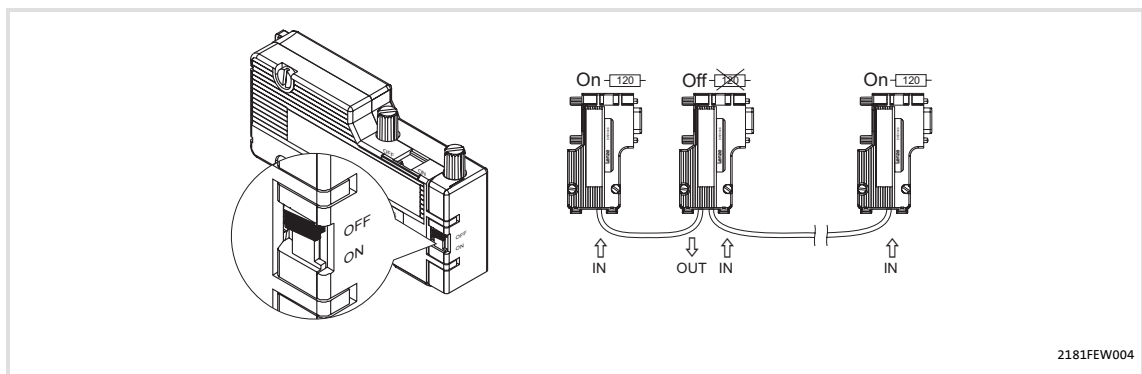


Documentation of the standard device, control system, and plant/machine
Observe the notes and wiring instructions stated.

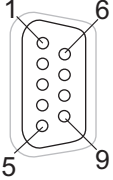
5.2.1 System bus (CANopen) connection



- The system bus (CANopen) must be terminated with resistors ($120\ \Omega$) between CAN-low and CAN-high.
- The Lenze system connector EWZ0046 with integrated terminating resistor complies with the DS102-1 recommendation of the CAN user organisation CiA. The system connector is not contained in the scope of supply of the communication module.



Assignment of the 9-pin Sub-D plug connector

View	Pin	Assignment
	1	-
	2	CAN-LOW
	3	CAN-GND
	4	-
	5	-
	6	-
	7	CAN-HIGH
	8	-
	9	-

5.2.2 Specification of the bus cable

We recommend the use of CAN cables according to ISO 11898-2:

CAN cable according to ISO 11898-2	
Cable type	Twisted in pairs with shield
Impedance	120 Ω (95 ... 140 Ω)
Cable resistance/cross-section	
Cable length $\leq$ 300 m:	$\leq$ 70 m Ω /m / 0.25 ... 0.34 mm ² (AWG22)
Cable length 301 ... 1000 m:	$\leq$ 40 m Ω /m / 0.5 mm ² (AWG20)
Signal propagation delay	$\leq$ 5 ns/m

5.2.3 Bus cable length



Note!

- It is absolutely necessary to comply with the permissible cable lengths.
- If the total cable lengths of the CAN nodes differ for the same baud rate, the smaller value must be used to determine the max. cable length.
- Observe the reduction of the total cable length due to the signal delay of the repeater. ▶ [Checking the use of repeaters](#) (30)

5.2.3.1 Total cable length

The baud rate also determines the total cable length.

Baud rate [kbps]	Max. bus length [m]
10	8075
20	4012
50	1575
125	600
250	275
500	112
800	38
1000	12

5.2.3.2 Segment cable length

The segment cable length is determined by the cable cross-section used and by the number of nodes. Repeaters divide the total cable length into segments. If no repeaters are used, the segment cable length is identical to the total cable length.

Max. number of nodes per segment	Cable cross-section (can be interpolated)			
	0.25 mm ² (AWG 24)	0.50 mm ² (AWG 21)	0.75 mm ² (AWG 19)	1.00 mm ² (AWG 18)
2	240 m	430 m	650 m	940 m
5	230 m	420 m	640 m	920 m
10	230 m	410 m	620 m	900 m
20	210 m	390 m	580 m	850 m
32	200 m	360 m	550 m	800 m
63	170 m	310 m	470 m	690 m
100	150 m	270 m	410 m	600 m

Example: selection help

Given	
Total cable length to be implemented	200 m
Number of nodes	63

Results	
Max. possible baud rate	250 kbps (derived from table Total cable length (□ 28))
Cable cross-section required (interpolated)	0.30 mm ² (AWG23) (derived from the table Segment cable length (□ 29))
Cable cross-section of standard CAN cable	0.34 mm ² (AWG22) ▶ Specification of the bus cable (□ 27)

5.2.3.3 Checking the use of repeaters

Compare the values derived from tables [Total cable length](#) (□ 28) and [Segment cable length](#) (□ 29).

- ▶ If the sum of the segment cable lengths is smaller than the total cable length to be implemented, either repeaters must be used or the cable cross-section must be increased.
- ▶ If the use of repeaters reduces the max. possible total cable length so much that it is smaller than the total cable length to be implemented, then the cable cross-section must be increased or less repeaters must be used or the baud rate must be decreased.
- ▶ The use of a further repeater is recommended as ...
 - service interface
Advantage: trouble-free connection during bus operation is possible.
 - calibration interface
Advantage: the calibration/programming unit remains electrically isolated.

Example

Given	
Total cable length to be implemented	450 m
Number of nodes	32
Cable cross-section	0.50 mm ² (AWG 20)
Baud rate	125 kbps
Repeater used	Lenze repeater EMF2176IB
Reduction of the max. total cable length per repeater (EMF2176IB)	30 m

Results	
Max. possible total cable length	600 m (see table Total cable length (□ 28))
Max. segment cable length	360 m (see table Segment cable length (□ 29))
Comparison	The max. segment cable length is smaller than the total cable length to be implemented.
Conclusion	A repeater must be installed at the determined max. segment cable length of 360 m.

Results with 1 repeater	
Max. possible total cable length	570 m (Reduction of the Total cable length (□ 28) by 30 m)
Sum of the segment cable lengths	720 m
Comparison	Both the possible total cable length and the segment cable lengths are larger than the total cable length to be implemented.
Conclusion	1 repeater is sufficient to implement the total cable length of 450 m.

5.2.4 Voltage supply

- ▶ The communication module is only internally supplied with voltage via the standard device.
- ▶ If the standard device fails, data transfer between the communication module and other CAN nodes is interrupted.

During commissioning, the controller receives system-specific data, e.g. motor parameters, operating parameters, responses, and parameters for the fieldbus communication. In case of Lenze devices, this is done via codes.

In addition to codes for the configuration, there are codes for diagnosing and monitoring the nodes.

Note!

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

- **C13nnn** for slot MXI1
 - ▶ [Parameters of the communication module for slot MXI1](#) (📖 107)
- **C14nnn** for slot MXI2
 - ▶ [Parameters of the communication module for slot MXI2](#) (📖 128)

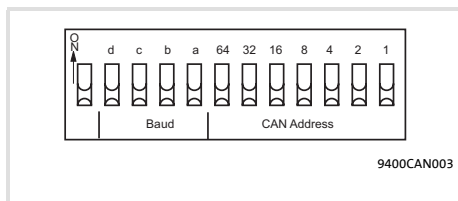
Furthermore, the [Communication-relevant parameters of the standard device](#) (📖 101) must be set.

Stop!

Before switching on the controller for the first time, check ...

- the entire wiring with regard to completeness, earth fault and short circuit.
- whether the bus system is terminated by a bus terminating resistor at the physically first and last node.
 - ▶ **System bus (CANopen) connection** (26)

6.2 Possible settings via DIP switches



[6-1] DIP switch

The front DIP switches can be used to set:

- ▶ address (switches 1 ... 64)
- ▶ baud rate (switches a ... d)

Lenze setting: all switches OFF

6.2.1 Setting the node address

- ▶ If several CAN nodes are interconnected, their node addresses must differ from each other.
- ▶ The node address can be set with the DIP switches 1 ... 64 or via the »Engineer« (code [C13350](#) / [C14350](#)).
- ▶ Valid address range: 1 ... 127

	Node address determined by ...	
	DIP switch	C13350 / 14350
Prerequisite	At least one switch 1 ... 64 = ON	Switches 1 ... 64 = OFF

The labelling on the package corresponds to the values of the individual DIP switches for determining the node address.

DIP switch	64	32	16	8	4	2	1
Switch position	OFF	OFF	ON	OFF	ON	ON	ON
Value	0	0	16	0	4	2	1
Node address	= Sum of the values = 16 + 4 + 2 + 1 = 23 ▶ DIP switch positions for setting the CAN node address (p. 178)						



Note!

Switch the voltage supply of the communication module off and then on again to activate altered settings.



Tip!

The node address resulting from the DIP switch setting applied at last mains power-up is indicated under [C13349/1](#) / [C14349/1](#).

6.2.2 Setting the baud rate

- The baud rate can be set with the DIP switches a ... d or via the »Engineer« (code [C13351](#) / [C14351](#)).
- If several CAN nodes are interconnected, their baud rates must be identical.

Switch positions				Baud rate
d	c	b	a	
OFF	ON	ON	OFF	10 kbps
OFF	ON	OFF	ON	20 kbps
OFF	OFF	ON	ON	50 kbps
OFF	OFF	ON	OFF	125 kbps
OFF	OFF	OFF	ON	250 kbps
OFF	OFF	OFF	OFF	500 kbps
ON	ON	ON	OFF	800 kbps
OFF	ON	OFF	OFF	1000 kbps
OFF	ON	ON	ON	Automatic detection



Note!

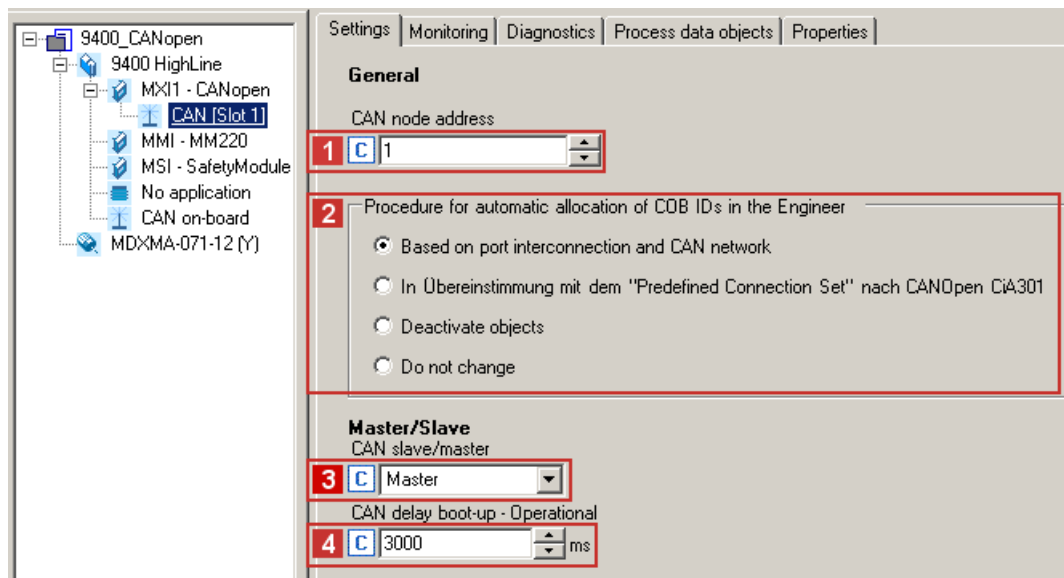
Switch the voltage supply of the standard device off and then on again to activate altered settings.



Tip!

The baud rate resulting from the DIP switch setting applied at last mains power-up is indicated under [C13349/2](#) / [C14349/2](#).

6.3 Settings in the »Engineer«



Go to the **Settings** tab to ...

- ▶ set the **1** node address ([C13350](#) / [C14350](#)).
 - ▶ [Setting the node address](#) ([book 33](#))
- ▶ select the **2** procedure for automatic allocation of the COB-IDs in the »Engineer«.
- ▶ configure the CAN node as a **3** master or slave ([C13352](#) / [C14352](#)).
- ▶ display **4** the time that must elapse after mains power-up before the CAN NMT master transmits the "Start remote node" telegram via the system bus (CANopen) ([C13378](#) / [C14378](#)).

6.4 Initial switch-on



Documentation of Servo Drive 9400

Observe the safety instructions and residual hazards stated.



Note!

Establishing communication

When the communication module is externally supplied, the standard device must also be switched on for establishing communication.

After communication has been established, communication of the externally supplied module is independent of the on/off state of the standard device.

Activating changed settings

In order to activate changed settings ...

- execute device command "11: Save start parameters" via standard device code **C00002** and ...
- then execute a "CAN reset node" on the node using **C00002 = 92** 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 not wanted or even impermissible that the drive restarts.

In the Lenze setting of the 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 controller:

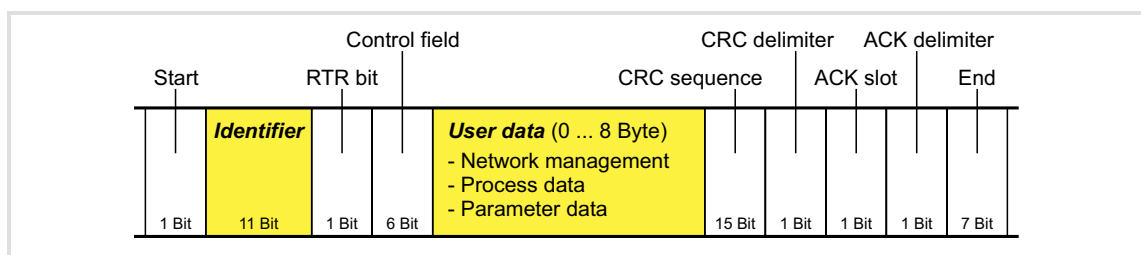
- **C00142 = "0: Inhibited"** (Lenze setting)
 - The controller remains inhibited (even if the fault is no longer active).
 - The drive starts up in a controlled manner by explicit controller enable: LOW-HIGH edge at digital input X5/RFR.
- **C00142 = "1: Enabled"**
 - An uncontrolled restart of the drive is possible.

7 Data transfer

Via the CANopen interface, process data and parameter values can be exchanged between the CAN nodes. In addition, the interface enables the connection of further modules such as distributed terminals, operator and input devices (HMI), or external controls and control systems.

The interface transfers CAN objects following the CANopen communication profile (DS301, V4.02) developed by the umbrella organisation of CiA (CAN in Automation) in conformity with the CAL (CAN Application Layer).

7.1 Structure of the CAN data telegram



[7-1] Basic structure of the CAN telegram

The identifier and the user data are described in detail in the following subchapters. The other signals refer to the transfer characteristics of the CAN telegram that are not described in this documentation.



Tip!

For further information please refer to the website of the CiA (CAN in Automation) user organisation:

www.can-cia.org

7.1.1 Identifier

The principle of the CAN communication is based on a message-oriented data exchange between a transmitter and many receivers. All nodes can transmit and receive quasi-simultaneously.

The identifier, also called "COB-ID" (Communication Object Identifier), is used to control which node is to receive a transmitted message. In addition to the addressing, the identifier contains information on the priority of the message and the type of the user data.

The identifier consists of a basic identifier and the node address of the node to be addressed:

Identifier (COB-ID) = basic identifier + node address (node ID)

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

Node address (node ID)

For the purpose of unique identification, a node address, also referred to as node ID, in the valid address range (1 ... 127) is to be assigned to each node within the CAN network.

- ▶ A node address may not be assigned more than once within a network.
- ▶ The node address can be configured with the DIP switches of the communication module or with code [C13350](#) / [C14350](#).
 - ▶ [Setting the node address](#) (📖 33)

Identifier assignment

The system bus (CANopen) is message-oriented and not node-oriented. Each message has a unique identification, the identifier. In the case of CANopen, a node-orientation is achieved by the fact that for each message there is only one sender.

- The basic identifiers for network management (NMT) and sync as well as the basic SDO channel (SDO1) are specified in the CANopen protocol and cannot be changed.
- In the Lenze setting, the basic identifiers of the PDOs are preset according to the "Predefined Connection Set" of DS301, V4.02. They can be changed via parameters/ indexes if required.

► [Identifiers of the process data objects](#) (46)

Object		Direction		Basic identifier	
		from the device	to the device	dec	hex
Network management (NMT)				0	0
Sync				128	80
Emergency		●		128	80
PDO1 (Process data channel 1)	TPDO1	●		384	180
	RPDO1		●	512	200
PDO2 (Process data channel 2)	TPDO2	●		640	280
	RPDO2		●	768	300
PDO3 (Process data channel 3)	TPDO3	●		896	380
	RPDO3		●	1024	400
PDO4 (Process data channel 4)	TPDO4	●		1152	480
	RPDO4		●	1280	500
SDO1 (Basic SDO channel)		●		1408	580
			●	1536	600
SDO2 ... SDO10 (Parameter data channel 2 ... 10)		●		1472	5C0
			●	1600	640
Node guarding, heartbeat		●		1792	700

7.1.2 User data

All nodes communicate with each other by exchanging data telegrams via the system bus (CANopen). The user data area of the CAN telegram contains network management data, parameter data or process data:

Network management data

(NMT data)

- ▶ Control information on start, stop, reset, etc. of communication to certain or all nodes of the CAN network.

Process data

(PDOs – Process Data Objects)

- ▶ Process data are transferred via the process data channel.
- ▶ The controller can be controlled using process data.
- ▶ Process data are not saved in the controller.
- ▶ Process data are transferred between the host and the nodes to ensure a continuous exchange of current input and output data.
- ▶ Process data usually are unscaled/scalable raw data
- ▶ Process data are, for instance, setpoints and actual values.

Parameter data

(SDOs – Service Data Objects)

- ▶ Parameter data are the CANopen indexes or, in the case of Lenze devices, the codes.
- ▶ The parameters are, for instance, set for the initial system set-up during commissioning or when material is changed on the production machine.
- ▶ Parameter data are transmitted as SDOs via the parameter data channel. They are acknowledged by the receiver, i.e. the sender receives a feedback about the transmission being successful or not.
- ▶ The parameter data channel enables access to all Lenze parameters (codes) and CANopen indexes.
- ▶ Parameter changes are automatically saved in the controller until the mains is switched.
- ▶ In general, the parameter transfer is not time-critical.
- ▶ Parameter data are, for instance, operating parameters, diagnostic information and motor data.

7.2 Communication phases / network management

Regarding communication via the system bus (CANopen), the drive distinguishes between the following states:

Status	Explanation
"Initialisation" (Initialisation)	After power-up, initialisation is executed. • During this phase, the controller does not take part in the data transfer on the system bus (CANopen). • All CAN-relevant parameters are written with their standard values again. • After initialisation has been completed, the controller is automatically set to the "pre-operational" status.
"Pre-operational" (before being ready for operation)	Parameter data can be received, process data is ignored.
"Operational" (ready for operation)	Parameter data and process data can be received!
"Stopped" (Stopped)	Only network management telegrams can be received.

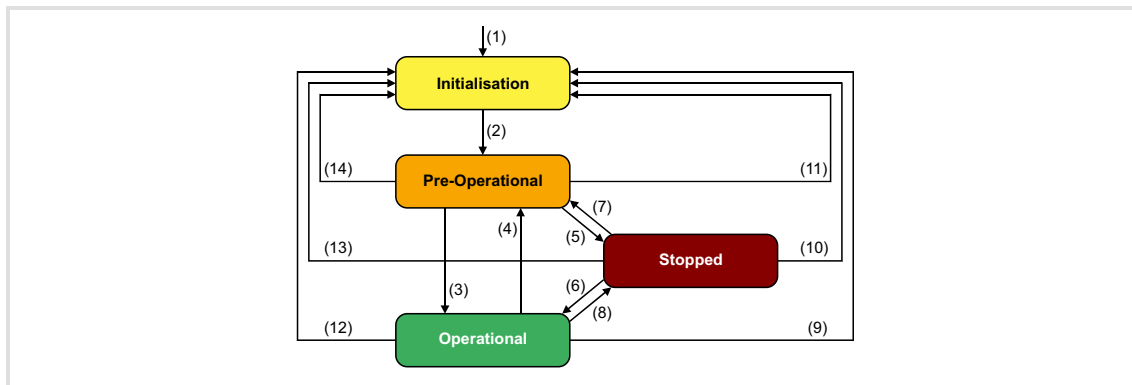
Communication object	Initialisation	Pre-operational	Operational	Stopped
PDO			•	
SDO		•	•	
Sync		•	•	
Emergency		•	•	
Boot-up	•			
Network management (NMT)		•	•	•



Tip!

In every state, the initialisation can be re-executed partly or completely by transmitting appropriate network management telegrams.

7.2.1 State transitions

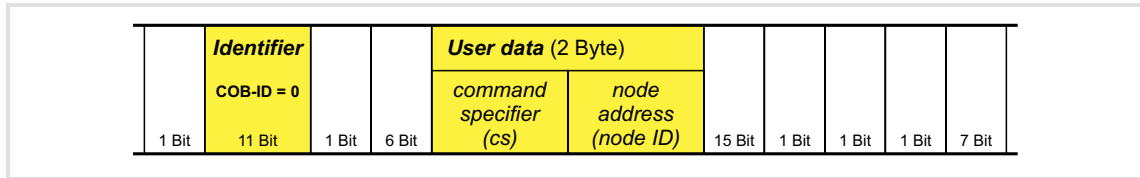


[7-2] NMT state transitions in the CAN network

Transition	NMT command	State after change	Effect on process and parameter data after state change
(1)	-	Initialisation	Initialisation starts automatically when the mains is switched on. - During initialisation, the controller does not take part in the data transfer. - After initialisation has been completed, the node sends a boot-up message with an individual identifier and automatically changes to the "pre-operational" status.
(2)	-	Pre-operational	In this phase, the master determines the way in which the node(s) takes/take part in the communication.
	From here, the states are changed over by the master for the entire network. - A target address contained in the NMT command specifies the receiver(s). - If the 9400 controller has been configured as a CAN master, the state automatically changes to "operational" after a waiting time (C13378 / C14378) has expired, and the NMT command 0x0100 ("Start remote node") is sent to all nodes. - Data can only be exchanged via process data objects if the state is "Operational".		
(3), (6)	0x01 xx Start Remote Node	Operational	Network management, sync and emergency telegrams as well as process data (PDO) and parameter data (SDO) are active. Optional: When the state is changed, event and time-controlled process data (PDOs) will be sent once.
(4), (7)	0x80 xx Enter Pre-Operational	Pre-operational	Network management, sync and emergency telegrams as well as parameter data (SDO) are active.
(5), (8)	0x02 xx Stop Remote Node	Stopped	Only network management telegrams can be received.
(9), (10), (11)	0x81 xx Reset node	Initialisation	Initialisation of all CAN-relevant parameters (DS301, V4.02) with the stored values.
(12), (13), (14)	0x82 xx Reset communication		Initialisation of all CAN-relevant parameters (DS301, V4.02) with the stored values.
	Meaning of the node address in the NMT command: - xx = 0x00: With this assignment, all nodes are addressed by the telegram (broadcast telegram). The state can be changed for all nodes at the same time. - xx = Node-ID: If a node address is indicated, the state will only be changed for the node addressed.		

7.2.2 Network management telegram (NMT)

The telegram for the network management contains the identifier "0" and the command included in the user data, which consists of the command byte and the node address.



[7-3] Network management telegram for changing the communication phases

Command specifier (cs)		NMT command
dec	hex	
1	0x01	Start Remote Node
2	0x02	Stop Remote Node
128	0x80	Enter Pre-Operational
129	0x81	Reset node
130	0x82	Reset communication

The changeover of the communication phases is carried out by one node, the CAN master, for the entire network. The role of the CAN master can also be assumed by the drive controller.

► [Parameterising the controller as a CAN master](#) (44)

Example:

Data can only be exchanged via the process data objects in the "Operational" state. To change all nodes on the bus from "Pre-Operational" to "Operational" via the CAN master, the following identifiers and user data must be set as follows in the transmit telegram:

- Identifier: 0x00 (network management)
- User data: 0x0100 (NMT command "Start Remote Node" to all nodes)

7.2.3 Parameterising the controller as a CAN master

If the initialisation of the system bus (CANopen) and the state change from "pre-operational" to "operational" is not carried out by a higher-level control system, the controller can be defined to be a "quasi" master to take over this task.

The controller can be configured to be a CAN master in [C13352](#) / [C14352](#).

- ▶ As a CAN master, the controller sets all nodes connected to the bus (broadcast telegram) to the "operational" communication state using the "Start remote node" NMT telegram. Only this communication state enables a data exchange via the process data objects.
- ▶ In [C13378](#) / [C14378](#), a delay time can be set. This delay time must elapse after mains switching before the controller transmits the "Start remote node" NMT telegram on the system bus (CANopen).

Parameter	Information	Lenze setting	
		Value	Unit
C13352 / C14352	CAN slave/master	Slave	
C13378 / C14378	CAN delay boot-up - Operational	3000	ms



Note!

Changing the master/slave operation in [C13352](#) / [C14352](#) will only be effective

- after switching the mains power of the controller off and then on again

or

- by sending the NMT telegram "Reset Node" or "Reset Communication" to the controller.

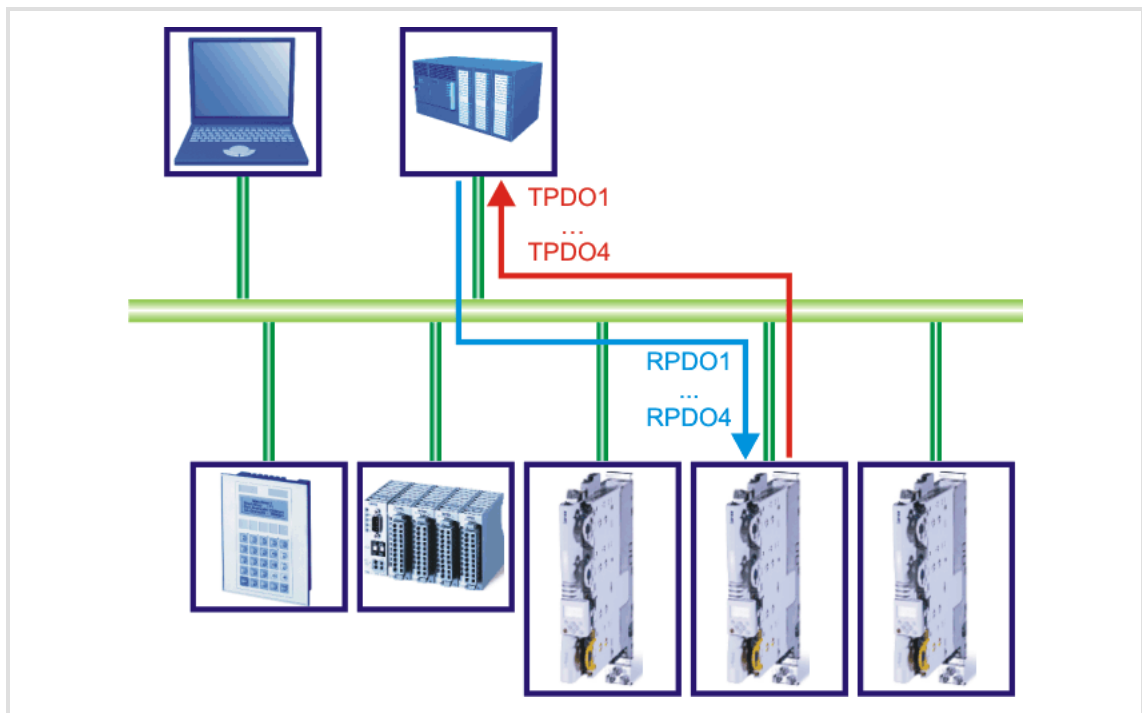
As an alternative to the "Reset node" NMT telegram, the device command **C00002 = "92: CAN module: reset node"** can be used to reinitialise the CAN-specific device parameters.



Tip!

The master functionality is only required for the initialisation phase of the drive system.

8 Process data transfer



[8-1] PDO data transfer from / to the higher-level host system

For the transfer of process data, four separated process data channels (PDO1 ... PDO4) are available.

Definitions

- ▶ Process data telegrams between the host and the devices are distinguished as follows:
 - Process data telegrams to the device (RPDO)
 - Process data telegrams from the device (TPDO)
- ▶ The CANopen process data objects are designated as seen from the node's view:
 - Receive PDO (RPDOx): process data object received by a node
 - Transmit PDO (TPDOx): process data object transmitted by a node



Note!

Data can only be exchanged via the process data objects if the state is "Operational".

▶ [Communication phases / network management](#) (41)

8.1 Identifiers of the process data objects

In the Lenze setting, the identifier for the process data objects PDO1 ... PDO4 results from a basic identifier and the node address set in [C13350](#) / [C14350](#):

Identifier (COB-ID) = basic identifier + node address (node ID)

- In the Lenze setting, the basic identifiers of the PDOs are preset according to the "Predefined Connection Set" of DS301, V4.02.
- The identifiers for the PDOs can be set individually via the Lenze codes and CANopen indexes listed in the following table. Thus, you can also set an identifier independently of the node address for certain PDOs.

Process data object		Basic identifier		Individual setting	
		dec	hex	Lenze code	CANopen index
PDO1	RPDO1	512	0x200	C13321/1 C14321/1	I-1400/1
	TPDO1	384	0x180	C13320/1 C14320/1	I-1800/1
PDO2	RPDO2	768	0x300	C13321/2 C14321/2	I-1401/1
	TPDO2	640	0x280	C13320/2 C14320/2	I-1801/1
PDO3	RPDO3	1024	0x400	C13321/3 C14321/3	I-1402/1
	TPDO3	896	0x380	C13320/3 C14320/3	I-1802/1
PDO4	RPDO4	1280	0x500	C13321/4 C14321/4	I-1403/1
	TPDO4	1152	0x480	C13320/4 C14320/4	I-1803/1



Note!

When the node address has been changed ([C13350](#) / [C14350](#)) and then "Reset node" (C00002 = 92) has been executed, the subcodes of [C13320](#) / [C14320](#) and [C13321](#) / [C14321](#) are automatically reset to the identifiers resulting from the respective basic identifier and the node address set.



Tip!

The "Predefined Connection Set" can be re-established anytime using the following device commands of **C00002**:

- "93: CAN on board: Pred.Connect.Set" for CAN on board
- "94: CAN module: Pred.Connect.Set" for communication module E94AYCCA

8.2 Transmission type

The process data objects are transmitted in an event-controlled or time-controlled way.

- ▶ **Event-controlled:** The PDO is sent if a special device-internal event has occurred, for instance, when the data contents of the TPDO have changed or when a transmission cycle time has elapsed.
- ▶ **Synchronous:** A TPDO (or RPDO) is transmitted (or received) after the device has received a sync telegram (with identifier 0x80).
- ▶ **Cyclically:** The PDOs are transmitted in fixed time intervals after the transmission cycle time has elapsed.

The table shows that combinations of logic operations (AND, OR) are also possible between the different transmission modes:

Transmission type	PDO transmission			Logic operation
	cyclic	synchronous	event-controlled	
0		●	●	AND
1 ... 240		●		-
254, 255	●		●	OR

Transmission type	Description
0	The PDO is transmitted at every sync in an event-controlled manner (e. g. by means of a bit change within the PDO).
1 ... 240	SYNC (with response) - Selection n = 1: The PDO is transmitted with <u>every</u> sync. - Selection 1 < n ≤ 240: The PDO is transmitted with <u>every n-th</u> sync.
254, 255	Event-controlled (with mask) with cyclic overlay If this value is entered, the PDO transmission is event-controlled <u>or</u> cyclic. (Note: The values "254" and "255" have the same meaning). For cyclic overlay, a cycle time must be set for the respective PDO. In this case, cyclic transmission takes place in addition to event-controlled transmission (e.g. through a bit change in the PDO).

The communication parameters (as e.g. transmission mode and cycle time) can be freely adjusted for any PDO and independent of the settings of other PDOs:

Parameter	Information	Lenze setting	
		Value	Unit
C13322/1...4 C14322/1...4	CAN TPDOx Tx mode	254	
C13323/1...4 C14323/1...4	CAN RPDOx Rx mode	254	
C13324/1...4 C14324/1...4	CAN TPDOx delay time	0	1/10 ms
C13356/1...4 C14356/1...4	CAN TPDOx cycle time	0	ms



Tip!

The setting can also be made via the following CANopen objects:

- [I-1400](#) ... [I-1403](#): Communication parameters for RPDO1 ... RPDO4
- [I-1800](#) ... [I-1803](#): Communication parameters for TPDO1 ... TPDO4

8.3 Masking of the TPDOs for event control

For TPDO1 ... TPDO4, a mask can be parameterised for every byte. In case of event-controlled transmission of a PDO, only the masked bits are used for the event control.

- ▶ Mask "0x0" means that no bit of the corresponding byte triggers the transmission.
- ▶ Mask "0xff" means that each bit of the corresponding byte can trigger the transmission.

Short overview: Parameters for masking the TPDOs

Parameter	Information	Lenze setting
C13311/1...8 C14311/1...8	CAN TPDO1 mask byte x	0x00
C13312/1...8 C14312/1...8	CAN TPDO2 mask byte x	0x00
C13313/1...8 C14313/1...8	CAN TPDO3 mask byte x	0x00
C13314/1...8 C14314/1...8	CAN TPDO4 mask byte x	0x00

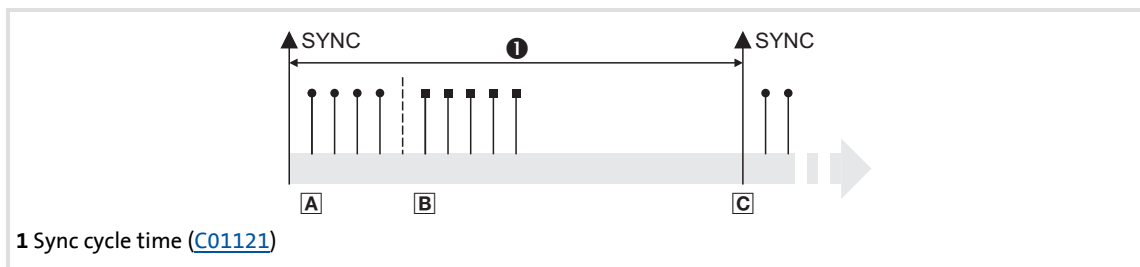
8.4 Synchronisation of PDOs via sync telegram

In case of cyclic transmission, one or several PDOs are transmitted or received at fixed time intervals. For synchronising the cyclic process data, an additional special telegram, the sync telegram, is used.

- ▶ The sync telegram is the trigger point for the transmission of process data from the frequency inverters to the master and for the acceptance of process data from the master by the slaves.
- ▶ For sync-controlled process data processing, the sync telegram must be generated accordingly.
- ▶ The response to a sync telegram is determined by the transmission type selected.

▶ [Transmission type](#) (47)

General procedure



[8-2] Sync telegram

- After the sync telegram has been received, the slaves send the synchronous process data to the master (TPDOs). The master reads them as process input data.
- When the sending process has been completed, the slaves (RPDOs) receive the process output data (from the master).
 - All other telegrams (e.g. parameters or the event-controlled process data) are accepted acyclically by the slaves after the transmission has been completed successfully.
 - The acyclic data are not shown in figure [8-2]. They must be considered when the cycle time is dimensioned.
- The data in the slave is accepted with the next sync telegram if the Rx mode is set to 1 ... 240. When the Rx mode is set to 254 or 255 the data is accepted in the next device cycle independent of the sync telegram.

8.4.1 Parameter setting

Short overview: Parameters for synchronisation via sync telegram

Parameter	Information	Lenze setting		Assignment	
		Value	Unit	Sync Master	Sync Slave
C13367 C14367	CAN SYNC Rx identifier	128			●
C13368 C14368	CAN SYNC Tx identifier	128		●	
C00369/1	CAN SYNC transmit cycle time	0	ms	●	
C01120	Sync source	Off			●
C01121	Sync cycle time	1000	µs		●
C01122	Sync phase position	400	µs		●
C01123	Sync tolerance	0	µs		●
C01124	Sync-PLL increment	109	ns		●
C01130	Sync application cycle	1000	µs		●

Sync source

[C01120](#) is used to select the source of the synchronisation signals. Basically, only one source can synchronise the node.

Sync cycle time

Time with which the internal phase-locking loop (PLL) expects the synchronisation signals. The time must be set in [C01121](#) in accordance with the cycle of the synchronisation source selected in [C01120](#).



Note!

For synchronisation via the system bus (CANopen), only integer multiples of 1000 µs should be set in [C01121](#).

Example: For the system bus (CANopen), the interval between two synchronisation signals has been set to 2 ms. If the system bus (CANopen) is to be used as synchronisation source, a sync cycle time of 2000 µs must be set in [C01121](#).

Sync phase position

The phase position defines the zero point of time for the application relating to the synchronisation signal (bus cycle). Since PDO processing is integrated in the system part of the application, the instant of the PDO acceptance also changes if the phase position is changed.

- ▶ If 0 is set, the application is started together with the synchronisation signal.
- ▶ If a value > 0 is set, the application starts by the set time interval before the synchronisation signal arrives (the phase position acts negatively).

Example: If the phase position is set to 400 µs, the system part of the application starts 400 µs before the synchronisation signal arrives.



Note!

From Servo Drive 9400 software version V3.0:

The effect of the sync phase position can be influenced by the application cycle set in [C01130](#). For the Lenze setting of [C01130](#), the behaviour remains as before.

Sync tolerance

Time slot for monitoring the synchronisation signal via the **LS_SyncInput** system block.

- ▶ If the last synchronisation signal has been within this time slot around the expected value, the *SYNC_bSyncInsideWindow* output of the **LS_SyncInput** system block is set to TRUE.
- ▶ This setting does not affect the synchronisation process.

Sync-PLL increment

If the cycle times of the synchronisation signal and the phase-locking loop (PLL) differ from each other, the setting in [C01124](#) defines the increment with which the phase-locking loop can be readjusted.

- ▶ If the system bus (CANopen) is used as synchronisation source, the recommended time increment for deviations is 109 ns (Lenze setting).

Sync application cycle

This parameter influences the effect of the sync phase position ([C01122](#)) with regard to the instant of acceptance of the synchronous PDOs by the application or the instant of transmission of the synchronous PDOs to the system bus (CANopen).

For Servo Drive 9400 software versions lower than V3.0 the following applies:

- ▶ The sync application cycle is permanently set to 1000 µs.
- ▶ The resulting PDO delay can be calculated with the following formula taking into consideration an internal processing time of 150 s:
$$\text{PDO delay} = (\text{sync cycle time} - \text{sync phase position} + 150 \mu\text{s}) \text{ modulo } 1000$$

From Servo Drive 9400 software version V3.0 the following applies:

- ▶ The sync application cycle can be set in [C01130](#). The value set is automatically rounded down to full 1000 µs.
- ▶ The resulting PDO delay can be calculated with the following formula, taking an internal processing time of 150 s into consideration:
$$\text{PDO delay} = (\text{sync cycle time} - \text{sync phase position} + 150 \mu\text{s}) \text{ modulo } \text{C01130}$$



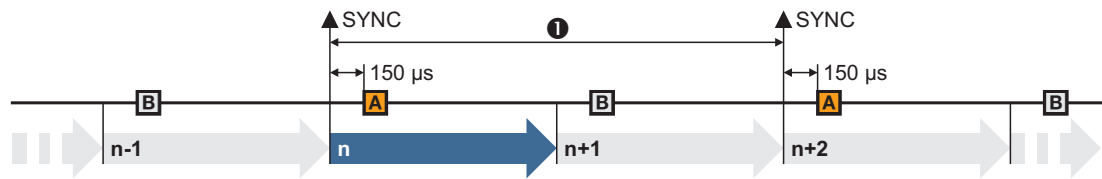
Note!

If the sync application cycle in [C01130](#) is set higher than the sync cycle time ([C01121](#)), the response is undefined. The same applies if the sync phase position ([C01122](#)) is set higher than the sync cycle time ([C01121](#)).

In these cases, usually no synchronous PDOs are transmitted via the system bus (CANopen) anymore.

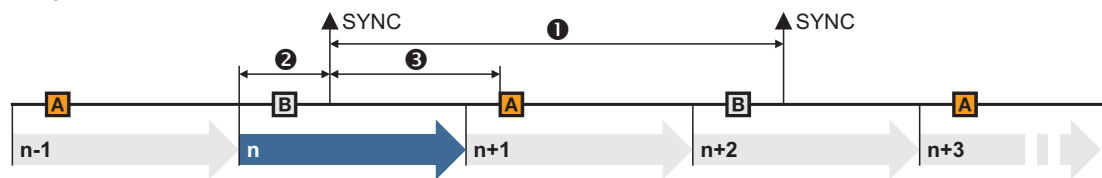
8.4.2 Effect of C01130 on the sync phase position

Example 1:



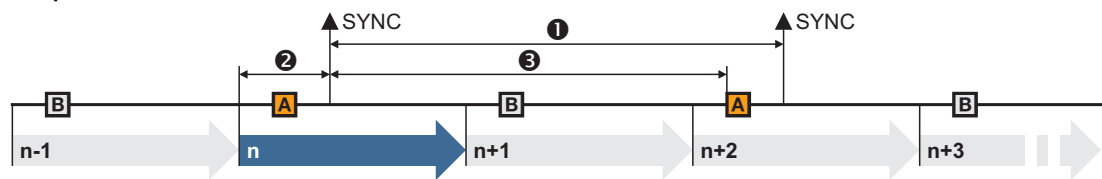
- 1 Sync cycle time (C01121) = 2000 µs
- Sync phase position (C01122) = 0 µs
- Sync application cycle (C01130) = 1000 µs
- A Instant of acceptance/transmission of synchronous PDOs
- B Inactive instant of acceptance
- n Device cycle

Example 2:



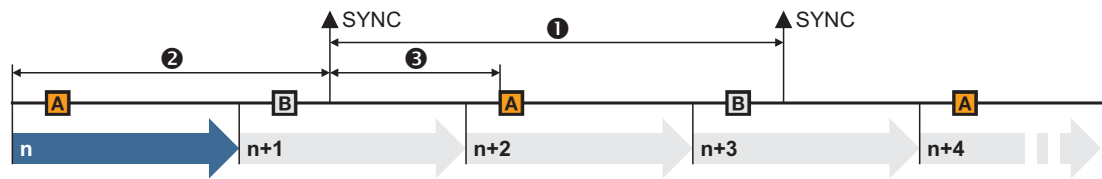
- 1 Sync cycle time (C01121) = 2000 µs
- 2 Sync phase position (C01122) = 400 µs
- Sync application cycle (C01130) = 1000 µs
- 3 PDO delay = (C01121 - C01122 + 150 µs) modulo C01130 = 750 µs
- A Instant of acceptance/transmission of synchronous PDOs
- B Inactive instant of acceptance
- n Device cycle

Example 3:



- 1 Sync cycle time (C01121) = 2000 µs
- 2 Sync phase position (C01122) = 400 µs
- Sync application cycle (C01130) = 2000 µs
- 3 PDO delay = (C01121 - C01122 + 150 µs) modulo C01130 = 1750 µs
- A Instant of acceptance/transmission of synchronous PDOs
- B Inactive instant of acceptance
- n Device cycle

Example 4:



- 1 Sync cycle time ([C01121](#)) = 2000 μ s
- 2 Sync phase position ([C01122](#)) = 1400 μ s
Sync application cycle ([C01130](#)) = 1000 μ s
- 3 PDO delay = ([C01121](#) - [C01122](#) + 150 μ s) modulo [C01130](#) = 750 μ s
- A Instant of acceptance/transmission of synchronous PDOs
- B Inactive instant of acceptance
- n Device cycle

8.5 PDO mapping

With Servo Drives 9400 HighLine, you can map the process data individually. For this purpose, the »Engineer« provides a port configurator.



Note!

The port mapping is no configuration that can be carried out online for the Servo Drive 9400. For this purpose, an update of the »Engineer« project and a subsequent download of the application is always required.

The »Engineer« screenshots illustrated in the following are only examples of the setting sequence and the resulting screens displayed.

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



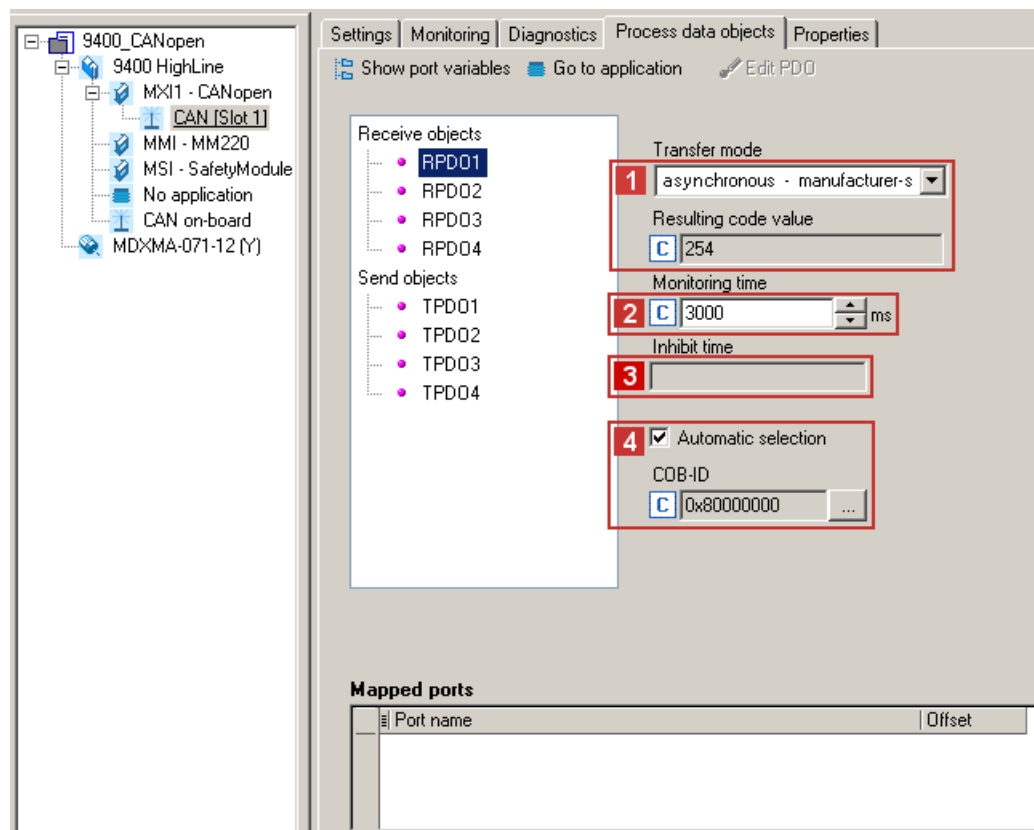
Implementing PDO mapping with the »Engineer«

Example

A project to be created in the »Engineer« consists of a master drive (master) and a slave drive (slave). Among other things, for instance PDOs are to be exchanged between the two drives via the system bus (CANopen). In the following, the parameterisation of the master drive is described.

1. On the **Process data objects** tab, the parameters for the receive objects (RPDO1 ... RPDO4) and transmit objects (TPDO1 ... TPDO4) can be set.

Settings for receive objects (RPDO1 ... RPDO4)



1 Transfer mode of RPDOx ([C13323/1...4](#) / [C14323/1...4](#))

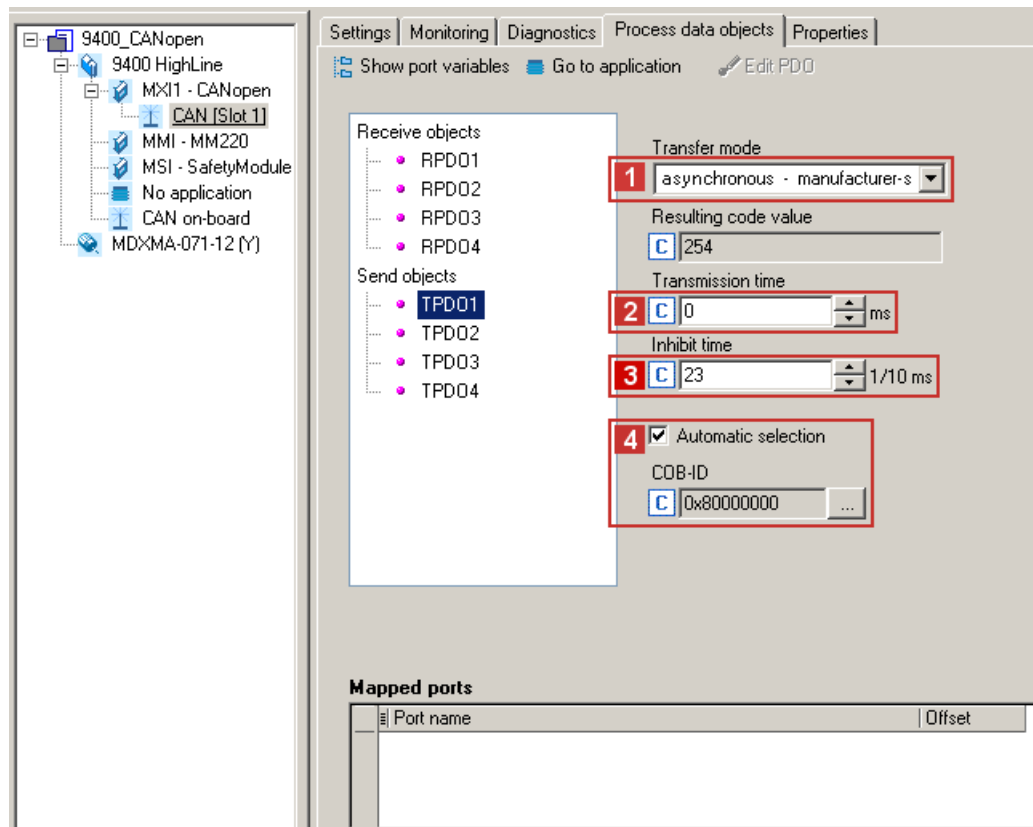
2 Monitoring time for RPDOx ([C13357/1...4](#) / [C14357/1...4](#))

- If a different value than "0" is entered, the RPDO is expected after the time set has elapsed.
- If the RPDO is not received within the monitoring time, an error response parameterised under [C13591/1...4](#) / [C14591/1...4](#) is initiated.

3 The inhibit time is not relevant for receive objects.

4 Display/setting of the RPDOx COB-ID ([C13321/1...4](#) / [C14321/1...4](#))

Settings for transmit objects (TPDO1 ... TPDO4)



1 Transfer mode of TPDOx ([C13322/1...4](#) / [C14322/1...4](#))

2 Transfer time of TPDOx ([C13356/1...4](#) / [C14356/1...4](#))

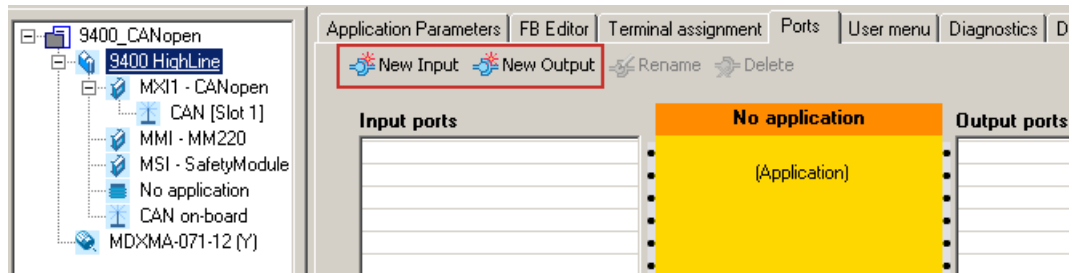
- If a different value than "0" is entered, the TPDO is transmitted without further consideration of the transport type after the time set has elapsed.

3 Inhibit time/delay time for TPDOx ([C13324/1...4](#) / [C14324/1...4](#))

- Minimum time between the transmission of two identical TPDOs.

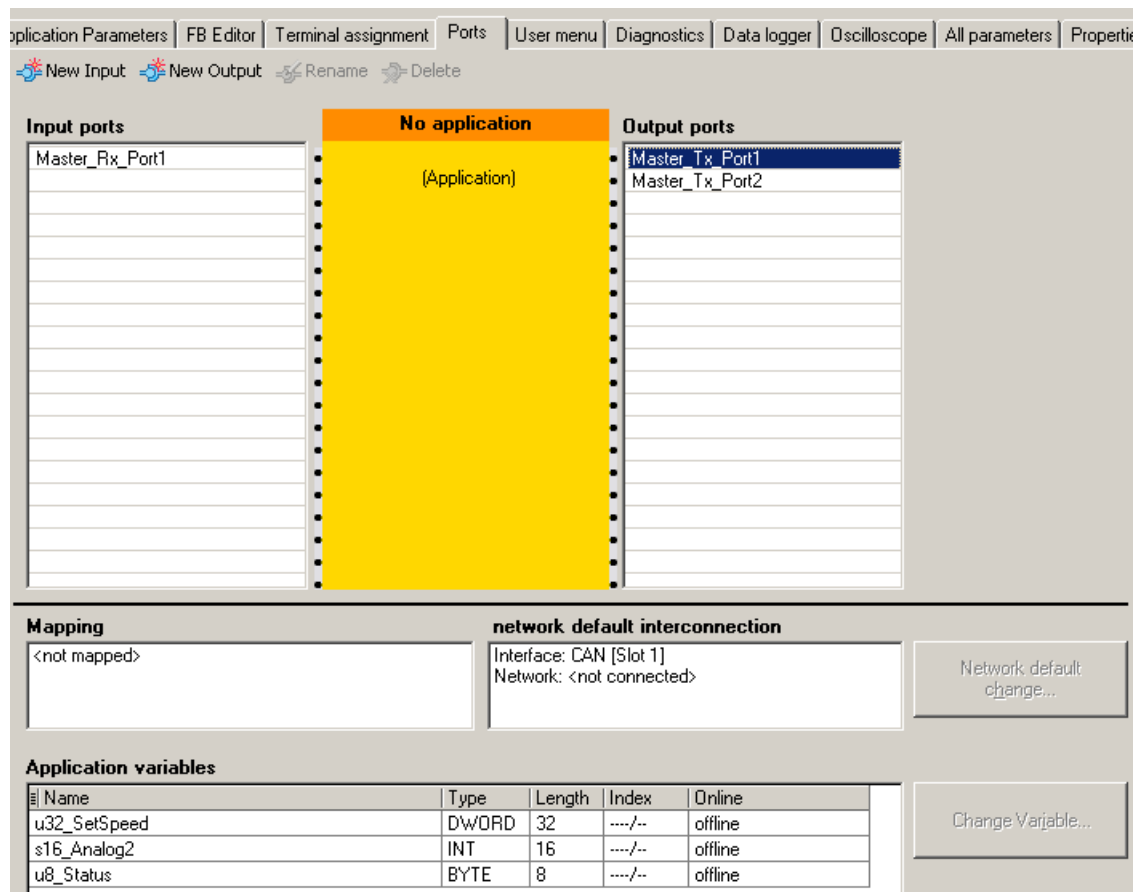
4 Display/setting of the TPDOx COB-ID ([C13320/1...4](#) / [C14320/1...4](#))

2. On the **Ports** tab, *input ports* and *output ports* can be created using the **New Input** and **New Output** buttons.



In the example, two *output ports* and one *input port* have been created for the master drive (master).

- A port exactly corresponds to a PDO object.
- For each port, *element variables* have been defined.



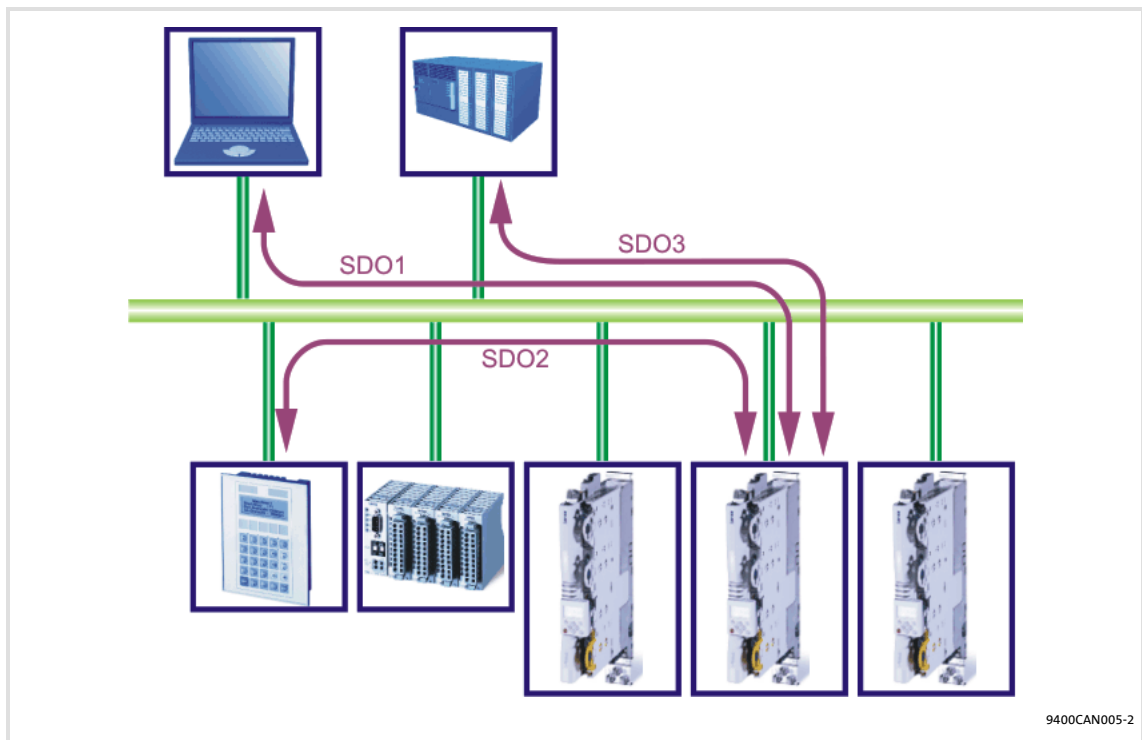
The following *element variables* are transmitted via TPDO1 (*Master_Tx_Port1*):

- u32_SetSpeed, type DWORD, 32 bits
- s16_Analog2, type INTEGER, 16 bits
- u8_Status, type BYTE, 8 bits

3. The application can be downloaded to the Servo Drive 9400 with the menu command **Online→Transfer application to device**.

When all PDOs have been programmed and the application has been completed, it can be downloaded to the Servo Drive 9400. All CAN parameters will be transferred according to the settings made in the »Engineer« project.

9 Parameter data transfer



[9-1] Parameter data transfer via available parameter data channels

Parameters are values stored in codes on Lenze controllers.

Ten separate parameter data channels are available for parameter setting, enabling the simultaneous connection of several devices for configuration.

Parameter data are transmitted via the system bus (CANopen) as SDOs ("Service Data Objects") and acknowledged by the receiver. The SDO enables read and write access to the object directory. Indexes (e.g. [I-1000](#)) ensure access to device parameters and functions included in the object directory. To transfer SDOs, the information contained in the user data must comply with the CAN-SDO protocol.

9.1 Identifiers of the parameter data objects

In the Lenze setting, the identifier for the parameter data objects SDO1 ... SDO10 results from a basic identifier and the node address set in [C13350](#) / [C14350](#):

Identifier (COB-ID) = basic identifier + node address (node ID)

- In the Lenze setting, the basic identifiers of the SDOs are preset according to the "Predefined Connection Set" of DS301, V4.02:

Parameter data object	Direction		Basic identifier	
	from the device	to the device	dec	hex
SDO1 (Parameter data channel 1)	●		1408	580
		●	1536	600
SDO2 ... 10 (Parameter data channel 2 ... 10)	●	●	Deactivated	
Node guarding, heartbeat	●		1792	700
Boot-up	●		1792	700



Note!

Please observe that the parameter data channels 2 ... 10 are deactivated in the Lenze setting.

The procedure for activating these parameter data channels is explained in the description for parameters [C13372](#) / [C14372](#) to [C13374](#) / [C14374](#) or in the description for the implemented CANopen object [I-120x](#).

► [Example](#) (166)

9.2 User data

Structure of the user data of the parameter data telegram

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte



Note!

User data are displayed in the Motorola format.

► [Examples for a parameter data telegram](#) (📖 69)

The following subchapters describe the user data in detail.

9.2.1 Command

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte

The following commands can be transmitted or received for writing and reading the parameters:

Command	1st byte		Data length	Information
	hex	dec		
Write request	0x23	35	4 bytes	Writing a parameter to the controller.
	0x2B	43	2 bytes	
	0x2F	47	1 byte	
	0x21	33	Block	
Write response	0x60	96	4 bytes	Acknowledgement by the controller regarding a write request.
Read request	0x40	64	4 bytes	Reading a parameter from the controller.
Read response	0x43	67	4 bytes	Response from the controller to a read request with the current parameter value.
	0x4B	75	2 bytes	
	0x4F	79	1 byte	
	0x41	65	Block	
Error response	0x80	128	4 bytes	Response from the controller when the read/write request could not be executed correctly. ► Error messages (67)

In detail, the command byte contains the following information:

Command	1st byte							
	Command specifier (cs)			Toggle (t)	Length*		e	s
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write request	0	0	1	0	0/1	0/1	1	1
Write response	0	1	1	0	0	0	0	0
Read request	0	1	0	0	0	0	0	0
Read response	0	1	0	0	0/1	0/1	1	1
Error response	1	0	0	0	0	0	0	0

*Bit coding of the length: 00 = 4 bytes, 01 = 3 bytes, 10 = 2 bytes, 11 = 1 bytes
e: expedited (shortened block service)
s: segmented (normal block service)



Tip!

Further commands are defined in the CANopen specification DS301, V4.02 (e.g. segmented transfer).

9.2.2 Addressing through index and subindex

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte

A parameter (Lenze code) is addressed according to the following formula:

Index = 24575 - (Lenze code number)

Example

The parameter **C00011** (motor reference speed) is to be addressed.

Calculation:

- ▶ Index:
 - Decimal: $24575 - 11 = 24564$
 - Hexadecimal: $0x5FFF - 0xB = 0x5FF4$
- ▶ Subindex: 0x00 (subindex 0, since the parameter has no subcodes.)

Entries:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	0xFF4	0x5F	0x00				

9.2.3 Data 1 ... data 4

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte

A maximum of 4 bytes is available for parameter value entries. The bytes are assigned as follows, depending on the data format:

5th byte	6th byte	7th byte	8th byte
Parameter value (1 byte)	0x00	0x00	0x00
Parameter value (2 bytes)		0x00	0x00
Low byte	High byte		
Parameter value (4 bytes)			
Low word		High word	
Low byte	High byte	Low byte	High byte



Note!

The [Table of attributes](#) (149) contains a scaling factor for Lenze parameters in the "factor" column. The scaling factor is important for the transfer of parameter values which are represented with one or several decimal positions in the parameter list.

When the scaling factor is > 1, the value must be multiplied by the scaling factor before being transmitted to be able to transmit the value as an integer. On the SDO client side, the integer must be divided by the scaling factor again to obtain the original value with decimal positions.

Example

For a code with the scaling factor "100" and the data format U32 the value "123.45" is to be transmitted.

Calculation:

- Value to be transmitted = scaling factor x value
- Data (1 ... 4) = 100 x 123.45 = 12345 (0x00 00 30 39)

Entries:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
				0x39	0x30	0x00	0x00

9.2.4 Error messages

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Error code			
0x80 (128)	Low byte	High byte		Low word		High word	
				Low byte	High byte	Low byte	High byte

In case of an error, the addressed node generates a telegram with the "Error Response" (0x80) command.

- The telegram contains the index and subindex of the code in which an error has occurred.
- Bytes 5 ... 8 contain the error code.
 - The error codes are standardised in accordance with DS301, V4.02.
 - The representation of the error codes is reversed to the read direction (see the following example).

Example

The representation of the error code "0x06040041" in bytes 5 ... 8:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Error code			
				0x41	0x00	0x04	0x06

Error codes

Error code	Explanation
0x0503 0000	Toggle bit not changed.
0x0504 0000	SDO protocol expired.
0x0504 0001	Invalid or unknown client/server command specifier.
0x0504 0002	Invalid block size (block mode only).
0x0504 0003	Invalid processing number (block mode only).
0x0504 0004	CRC error (block mode only).
0x0504 0005	Memory does not suffice.
0x0601 0000	Object access not supported.
0x0601 0001	Attempted read access to a writable only object.
0x0601 0002	Attempted write access to a readable only object.
0x0602 0000	Object not listed in object directory.
0x0604 0041	Object not mapped to PDO.
0x0604 0042	Number and length of objects to be transferred longer than PDO length.
0x0604 0043	General parameter incompatibility.
0x0604 0047	General internal device incompatibility.
0x0606 0000	Access denied because of hardware error.
0x0607 0010	Unsuitable data type (unsuitable service parameter length).
0x0607 0012	Unsuitable data type (service parameter length exceeded).
0x0607 0013	Unsuitable data type (service parameter length too short).
0x0609 0011	Subindex does not exist.
0x0609 0030	Parameter value range exceeded.
0x0609 0031	Parameter values too high.
0x0609 0032	Parameter values too low.
0x0609 0036	Maximum value falls below minimum value.
0x0800 0000	General error.
0x0800 0020	Data cannot be transferred/stored for application.
0x0800 0021	Data cannot be transferred/stored for application due to local control.
0x0800 0022	Data cannot be transferred/stored for application due to current device status.
0x0800 0023	Dynamic generation of object directory not successful or no object directory available (e.g. object directory generated from file, generation not possible because of a file error).

9.3 Examples for a parameter data telegram

9.3.1 Reading parameters

Task: The heatsink temperature of 43 °C (code **C00061**, data format INTEGER32, scaling factor 1) is to be read from the controller with node address "5".

Telegram to drive

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x0605	0x40	0xC2	0x5F	0x00	0x00	0x00	0x00	0x00

Explanations of the telegram to the drive

Identifier	= 1536 + node address = 1536 + 5 = 1541 = 0x0605 (1536 = basic identifier SDO1 to the controller)
Command	= 0x40 = "Read Request" (read request of a parameter from the controller)
Index	= 24575 - code number = 24575 - 61 = 24514 = 0x5FC2
Subindex	= 0 (code C00061 has no subcodes)

Response telegram from drive (if data have been transmitted correctly)

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x0585	0x43	0xC2	0x5F	0x00	0x2B	0x00	0x00	0x00

Explanations of the telegram from the drive

Identifier	= 1408 + node address = 1408 + 5 = 1413 = 0x0585 (1408 = basic identifier SDO1 from the controller)
Command	= 0x43 = "Read Response" (response to read request with current value)
Index	as in telegram to the drive
Subindex	
Data 1 ... 4	= 0x0000002B = 43 [°C]

9.3.2 Writing parameters

Task: The rated current of the connected motor with $I_{\text{rated}} = 10.2 \text{ A}$ (code **C00088**) is to be entered in the controller with node address "2".

Telegram to drive

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x0602	0x23	0xA7	0x5F	0x00	0x66	0x00	0x00	0x00

Explanations of the telegram to the drive

Identifier	= 1536 + node address = 1536 + 2 = 1538 = 0x0602 (1536 = basic identifier SDO1 to the controller)
Command	= 0x23 = "Write Request" (write request of a parameter to the controller)
Index	= 24575 - code number = 24575 - 88 = 24487 = 0x5FA7
Subindex	= 0 (code C00088 has no subcodes)
Data 1 ... 4	= $10.2 \times 10 = 102 = 0x00000066$ (Value for motor current, data type U32; display factor 1/10)

Response telegram from drive (if data have been transmitted correctly)

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x0582	0x60	0xA7	0x5F	0x00	0x00	0x00	0x00	0x00

Explanations of the telegram from the drive

Identifier	= 1408 + node address = 1408 + 2 = 1410 = 0x0582 (1408 = basic identifier SDO1 from the controller)
Command	= 0x60 = "Write Response" (acknowledgement of the write access from the controller)
Index	as in telegram to the drive
Subindex	

9.3.3 Reading block parameters

Task: The firmware version (code **C00099**) is to be read from the parameter set of the controller with node address "12". The firmware version has a length of 11 ASCII characters and is transmitted as a block parameter. In each block, the data width from the 2nd to the 8th byte within the user data is occupied.

Telegram 1 to the drive: read request

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x060C	0x40	0x9C	0x5F	0x00	0x00	0x00	0x00	0x00

Explanations of the telegram to the drive

Identifier	= 1536 + node address = 1536 + 12 = 1548 = 0x060C (1536 = basic identifier SDO1 to the controller)
Command	= 0x40 = "Read Request" (read request of a parameter from the controller)
Index	= 24575 - code number = 24575 - 99 = 24476 = 0x5F9C
Subindex	= 0 (code C00099 has no subcodes)

Response telegram 1 from the drive: Data of the block length (11 characters)

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x058C	0x41	0x9C	0x5F	0x00	0x0B	0x00	0x00	0x00

Explanations of the telegram from the drive

Identifier	= 1408 + node address = 1408 + 12 = 1420 = 0x058C (1408 = basic identifier SDO1 from the controller)
Command	= 0x41 = "Read Response" (response is a block telegram)
Index	as in telegram to the drive
Subindex	
Data 1 ... 4	= 0x0000000B = data length of 11 characters in ASCII format

Telegram 2 to the drive: Request of the 1st data block

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x060C	0x60	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Explanations of the telegram to the drive

Command = 0x60 = "Read Segment Request" (request: read data block)
 • Bit 4 = 0 (toggle bit)

Influence of the toggle bit to the request command

The individual blocks are toggled successively, i.e. first the request with command "0x60" is effected (= 0110*0000_{bin}), then command "0x70" (= 0111*0000_{bin}), then "0x60" again, etc.

* Toggle bit

Response telegram 2 from the drive: Send 1st data block

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x058C	0x00	0x30	0x31	0x2E	0x30	0x30	0x2E	0x30
		0 _{asc}	1 _{asc}	· _{asc}	0 _{asc}	0 _{asc}	· _{asc}	0 _{asc}

Explanations of the telegram to the drive

Command = 0x00 = 00000000_{bin}
 • Bit 4 = 0 (toggle bit)

Influence of the toggle bit to the transmission command

• The first response of the controller in the command byte is "0x0000*0000_{bin}" if bytes 2 ... 8 are completely occupied by data and further telegrams are following.

• The second response of the controller in the command byte is "0x0001*0000_{bin}" if bytes 2 ... 8 are completely occupied by data and further telegrams are following, etc.

* Toggle bit

Data 1 ... 7 = "01.00.0" (ASCII representation)

Telegram 3 to the drive: Request of the 2nd data block

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x060C	0x70	0x00	0x00	0x00	0x00	0x00	0x00	0x00

Explanations of the telegram 3 to the drive

Command	= 0x70 = "Read Segment Request" (request: read data block) • Bit 4 = 1 (toggle bit)
---------	--

Response telegram 3 from the drive: Send 2nd data block with end identifier

Identifier	User data							
	1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
	Command	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7
0x058C	0x17	0x30	0x2E	0x30	0x30	0x00	0x00	0x00
		0 _{asc}	·asc	0 _{asc}	0 _{asc}	-	-	-

Explanations of the telegram 3 from the drive

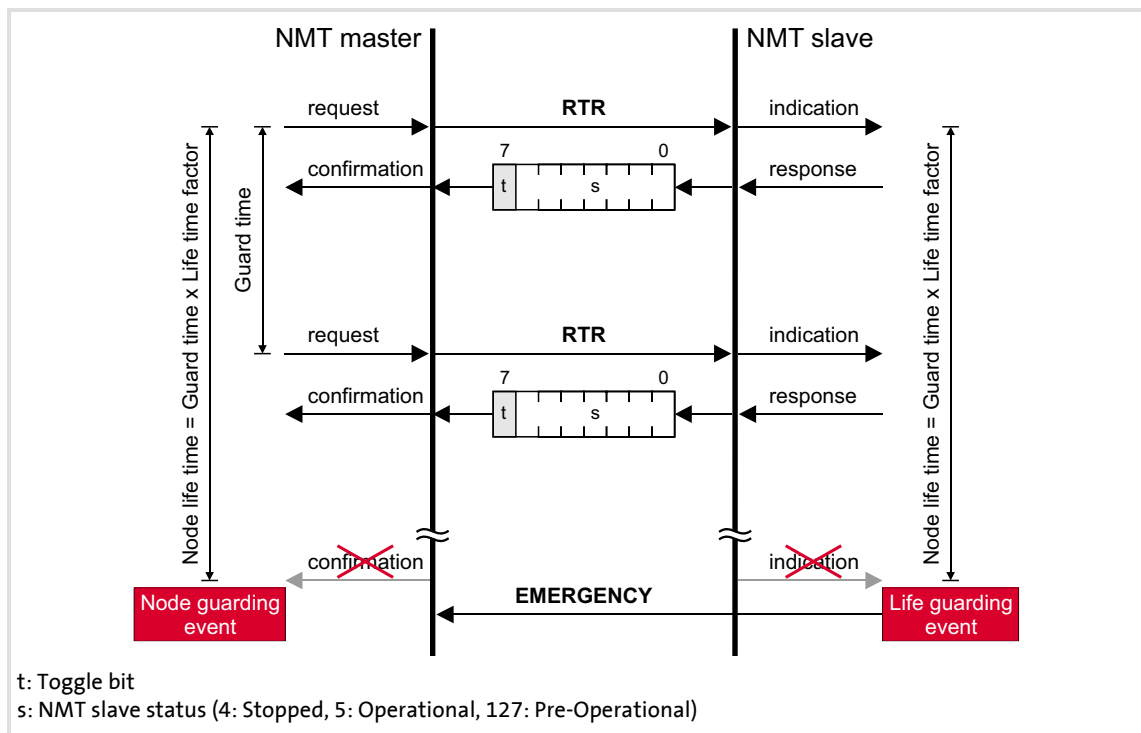
Command	= 0x17 = 00010111 _{bin} : • Bit 0 = 1 (end of transmission) • Bit 1 ... bit 3 = 011 _{bin} (3 bytes do not contain any data) • Bit 4 = 1 (toggle bit)
	Influence of the end bit and the residual data length on the transmission command • The end of transmission is reported by the set end bit 0. • By means of bit 1 ... 3, the number of bytes which do not contain data anymore is announced. * Toggle bit
Data 1 ... 7	= "0.00" (ASCII representation) The result of the data block transfer is: "01.00.00.00"

10 Monitoring

10.1 Node guarding protocol

In a CAN network, the node guarding protocol serves to monitor the connection between the NMT master and the NMT slave(s). If the controller was parameterised as NMT master, it can monitor up to 32 NMT slaves.

General procedure



[10-1] Node guarding protocol

1. The NMT master within cyclic time intervals sends a data telegram to the NMT slave, which is referred to as "Remote Transmission Request" (RTR).
2. The NMT slave then returns a response telegram ("Response") to the NMT master.

10.1.1 Telegram structure

RTR telegram

- ▶ The RTR telegram from the NMT master has the following identifiers:
Identifier (COB-ID) = 1792 + node address of the NMT slave
- ▶ The RTR telegram does not contain any user data.
- ▶ The RTR bit in the arbitration field of the RTR telegram is set to the valency LOW (dominant level).

Response telegram

- ▶ The response telegram from the requested NMT slave has the same identifier as the RTR telegram received by the NMT master.
- ▶ The user data (1 byte) contains the NMT slave status and the toggle bit (see the following description).

NMT slave state (s)

NMT slave status		Data							
Communication status	Decimal value (s)	(t)	NMT slave state (s)						
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Stopped	4	0/1	0	0	0	0	1	0	0
Operational	5	0/1	0	0	0	0	1	0	1
Pre-operational	127	0/1	1	1	1	1	1	1	1

Toggle bit (t)

- ▶ The toggle bit (t) in the response telegram has the value "0" when the node guarding protocol is activated for the first time.
- ▶ The toggle bit (t) must change its value with each response.



Note!

The toggle bit is monitored by the NMT master.

If a telegram is received with a toggle bit value that has not changed compared to the previously received telegram, it will be treated as if it were not received, i.e. the monitoring time is not reset and elapses further.

The toggle bit can only be reset to the value "0" by the "Reset Communication" telegram of the NMT master.

10.1.2 Parameter setting

Short overview of parameters for the "Node Guarding" monitoring function:

Parameter	Information	Lenze setting		Assignment	
		Value	Unit	Master	Slave
C13382 / C14382	CAN guard time	0	ms		●
C13383 / C14383	CAN life time factor	0			●
C13386 / C14386	CAN node guarding	0x00000000		●	
C13387 / C14387	CAN node guarding activity	-		●	
C13388 / C14388	CAN node guarding status	-		●	
C13612 / C14612	Resp. CAN node guarding error	No response		●	
C13614 / C14614	Resp. CAN life guarding error	No response			●
C13625 / C14625	CAN behaviour in the case of error	Pre-operational state		●	●

Highlighted in grey = display parameter

Guard time

The time interval with which the NMT master transmits the RTR telegram is the guard time.

- For each NMT slave to be monitored an individual time interval can be set.
- The RTR telegram prompts the NMT slave to send its current status.

Node life time

The node life time is the product of the guard time and the life time factor:

node life time = guard time x life time factor

- "Life time factor" and "Guard time" have to be known to the NMT master. For this, the values from the NMT slave are read at each reboot, or defined values are sent to the NMT slave at each reboot.
- It is possible to select a different "node life time" for each NMT slave to be monitored.

OK status

The status of the connection is ok (OK status) if within the "Node life time"

- the NMT slave has received an RTR telegram from the NMT master and
- the NMT master has received a correct response from the requested NMT slave.

In the OK status the monitoring times for the NMT master and the NMT slave are reset and the node guarding protocol is continued.

Life guarding event

The "life guarding event" is triggered in the NMT slave if the slave has not received an RTR telegram from the NMT master within the node life time:

- ▶ In the Lenze setting, the NMT slave changes from the "Operational" communication status into the "Pre-Operational" communication status.
 - [C13625](#) / [C14625](#) or the [I-1029](#) object can be used to set another status change.
- ▶ An emergency telegram with the emergency error code 0x8130 is transmitted to the NMT master.
- ▶ The response parameterised in [C13614](#) / [C14614](#) is executed (Lenze setting: "No response").



Note!

The "Life Guarding Event" can only be triggered in the NMT slave if at least one RTR telegram has been received successfully from the NMT master.

Node guarding event

The "node guarding event" is triggered in the NMT master if the master has not received any response to its RTR telegram from the requested NMT slave within the node life time or if the toggle bit in the response telegram has not changed within the node life time.

- ▶ In the Lenze setting, the NMT master changes from the "Operational" communication status into the "Pre-Operational" communication status.
 - [C13625](#) / [C14625](#) or the [I-1029](#) object can be used to set another status change.
- ▶ The response parameterised in [C13612/1...32](#) / [C14612/1...32](#) is executed (Lenze setting: "No response"). For each monitored node, an individual NMT master response can be set.



Note!

The "Node Guarding Event" can only be triggered in the NMT master if at least one response has been received successfully from the requested NMT slave.

10.1.3 Commissioning example

Task

A Servo Drive 9400 configured as NMT master (node 1) is to monitor another Servo Drive 9400 (node 2).

- ▶ The node guarding telegram is to be transmitted from the NMT master to the NMT slave in intervals of 1 s:
 - Guard time = 1000 ms
- ▶ The node life time is to amount to 5 seconds:
 - Node life time = guard time (1000 ms) x life time factor (5)
- ▶ If an error occurs, an error response is to be activated both in the NMT master and the NMT slave.

Parameter setting of the NMT master (node 1)

1. Set heartbeat producer time ([C13381](#) / [C14381](#)) to 0 ms to deactivate the heartbeat monitoring (node guarding and heartbeat must not be used simultaneously on a CANopen device).
2. Configure controller as NMT master: Set [C13352](#) / [C14352](#) = "1: Master".
3. Set guard time ([C13382](#) / [C14382](#)) to 0 ms (slave parameter).
4. Set life time factor ([C13383](#) / [C14383](#)) to 0 (slave parameter).
5. Configure monitoring for node guarding in [C13386](#) / [C14386](#).
 - The value to be entered into a free subcode (1 ... 32) is "0x050203E8". It consists of the following:

Bit 31 ... bit 24 Life time factor	Bit 23 ... bit 16 Node address of slave	Bit 15 ... bit 0 Guard time
0x05	0x02	1000 [ms] = 0x03E8

6. Go to [C13612/1...32](#) / [C14612/1...32](#) and set the respective response required for the monitoring functions parameterised under [C13386/1...32](#) / [C14386/1...32](#). This response will be executed if a node guarding event occurs in the NMT master.



Tip!

- [C13387](#) / [C14387](#) displays the activity of each monitoring function parameterised under [C13386/1...32](#) / [C14386/1...32](#) in a bit-coded manner.
- [C13388/1...32](#) / [C14388/1...32](#) display the node guarding status of the monitored NMT slaves.
- [C13625](#) / [C14625](#) serves to set which state change is to occur in the NMT master in the case of a node guarding event.

Parameterise NMT slave (node 2)

1. Adapt the life time factor and guard time settings of the NMT slave to the settings applied to the NMT master under [C13386](#) / [C14386](#):
 - Set guard time ([C13382](#) / [C14382](#)) to 1000 ms.
 - Set life time factor ([C13383](#) / [C14383](#)) to 5.
2. Go to [C13614](#) / [C14614](#) and set the response to be executed in the NMT slave in the case of a life guarding event.

**Tip!**

[C13625](#) / [C14625](#) serves to set which state change is to occur in the NMT slave in the case of a life guarding event.

Node guarding telegrams

- ▶ Remote Transmission Request:
The RTR telegram from the NMT master has the following identifiers:
Identifier (COB-ID) = 1792 + node address of slave = 1792 + 2 = 1794 = 0x702
 - ▶ Remote Transmission Response:
The response telegram from the NMT slave has the same identifier and the "operational" NMT slave state in the user data (s = 5).
- ▶ [Telegram structure](#) (📖 75)

10.2 Heartbeat protocol

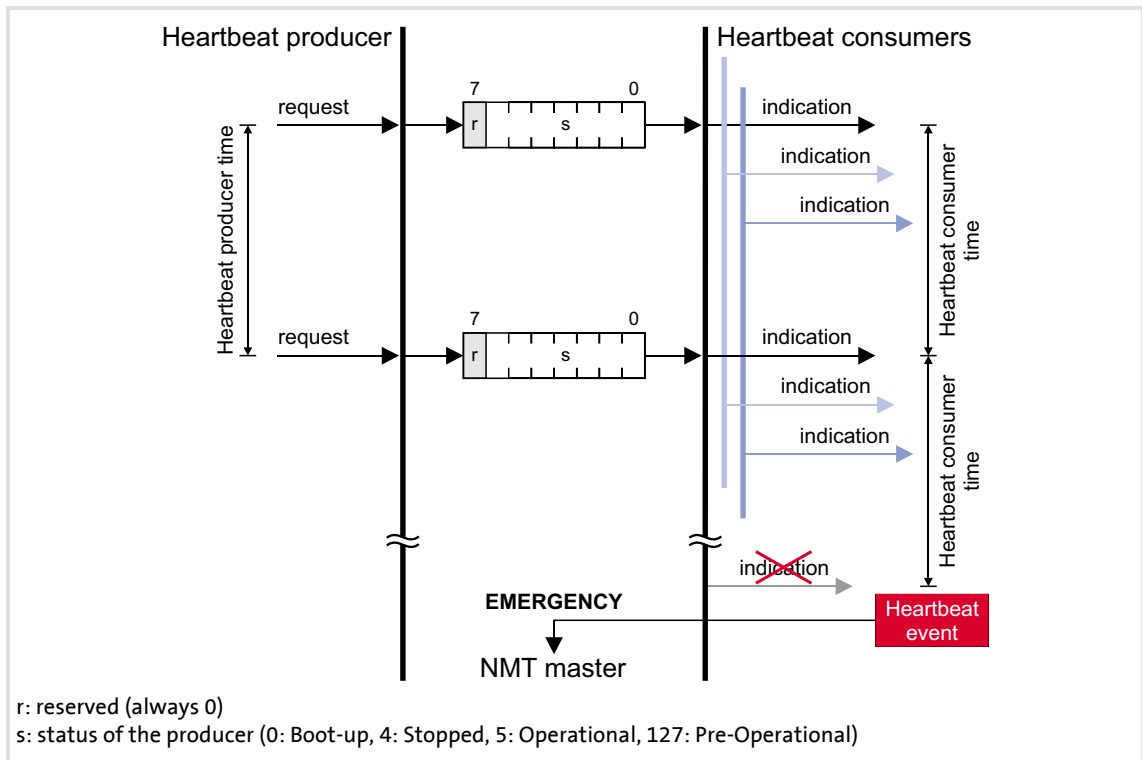
The heartbeat protocol can be used as an alternative to the node guarding protocol for monitoring nodes within a CAN network. Unlike node guarding, this monitoring does not require a polling by means of an RTR telegram (Remote Transmission Request) from the NMT master.



Note!

Heartbeat and node guarding protocols must not be used simultaneously on a CANopen device. If the heartbeat producer time is set > 0 ms, the heartbeat protocol is used.

General procedure



[10-2] Heartbeat protocol

1. A heartbeat producer cyclically sends a heartbeat telegram to one or several heartbeat consumers.
2. One or several consumers monitor the regular arrival of the heartbeat telegram.

10.2.1 Telegram structure

- The heartbeat telegram from the producer has the following identifier:
Identifier (COB-ID) = 1792 + node address of the producer
- The user data (1 byte) contain the status (s) of the producer:

Heartbeat producer status		Data							
Communication status	Decimal value (s)	(r)	Producer status (s)						
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Boot-up	0	0	0	0	0	0	0	0	0
Stopped	4	0	0	0	0	0	1	0	0
Operational	5	0	0	0	0	0	1	0	1
Pre-operational	127	0	1	1	1	1	1	1	1

10.2.2 Parameter setting

Short overview of the parameters for the "Heartbeat" monitoring function:

Parameter	Information	Lenze setting		Assignment	
		Value	Unit	Consumer	Producer
C13346 / C14346	CAN heartbeat activity	-		●	
C13347 / C14347	CAN heartbeat status	-		●	
C13381 / C14381	CAN heartbeat producer time	0	ms		●
C13385 / C14385	CAN heartbeat consumer time	0x00000000		●	
C13613 / C14613	Resp. to CAN heartbeat error	No response		●	
C13625 / C14625	CAN behaviour in the case of error	Pre-operational state		●	●

Highlighted in grey = display parameter

Heartbeat producer time

Time interval for the transmission of the heartbeat telegram to one or several consumers.

- Can be parameterised in [C13381](#) / [C14381](#) or via the [I-1017](#) object. The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is automatically transmitted as soon as a time > 0 ms is set.



Note!

Heartbeat and node guarding protocols must not be used simultaneously on a CANopen device. If the heartbeat producer time is set > 0 ms, the heartbeat protocol is used.

Heartbeat consumer time

Monitoring time for the node (producer) to be monitored.

- ▶ Can be parameterised in [C13385/1...32](#) / [C14385/1...32](#) or via the [I-1016](#) object.
- ▶ The parameterised time is rounded down to an integer multiple of 5 ms and must have a higher value than the heartbeat producer time of the node to be monitored.
- ▶ A consumer can monitor up to 32 producers.

Heartbeat event

The "heartbeat event" is triggered in the consumer if the consumer has not received a heartbeat telegram from the producer within the heartbeat consumer time:

- ▶ In the Lenze setting, the consumer changes from the "Operational" communication status into the "Pre-Operational" communication status.
 - [C13625](#) / [C14625](#) or the [I-1029](#) object can be used to set another status change.
- ▶ An emergency telegram with the emergency error code 0x8130 is transmitted to the NMT master.
- ▶ The response parameterised for the respective producer under [C13613/1...32](#) / [C14613/1...32](#) is executed (Lenze setting: "No response").



Note!

The heartbeat monitoring starts when the first heartbeat telegram of a monitored producer has been successfully received and the "pre-operational" NMT status has been assumed.

The boot-up telegram counts as the first heartbeat telegram!

10.2.3 Commissioning example

Task

A Servo Drive 9400 configured as heartbeat consumer (node 2) is to monitor another Servo Drive 9400 (heartbeat producer; node 1).

- ▶ The heartbeat producer is to transmit every 10 seconds a heartbeat telegram to the heartbeat consumer.
- ▶ The heartbeat consumer monitors the arrival of the heartbeat telegram. In case of an error, a response is to take place.

Parameter setting of the heartbeat producer (node 1)

1. Set heartbeat producer time ([C13381](#) / [C14381](#)) to 10 ms.

Parameter setting of the heartbeat consumer (node 2)

1. Configure heartbeat monitoring in [C13385](#) / [C14385](#).
 - Note: The heartbeat consumer time must be higher than the heartbeat producer time set in [C13381](#) / [C14381](#) for the node to be monitored.
 - The value to be entered into a free subcode (1 ... 32) is "0x0001000F". It consists of the following:

Bit 31 ... bit 24 Reserved	Bit 23 ... bit 16 Node address of the producer	Bit 15 ... bit 0 Heartbeat consumer time (integer multiple of 5 ms)
0x00	0x01	15 [ms] = 0x000F

2. Go to [C13613/1...32](#) / [C14613/1...32](#) and set the respective response required for the monitoring functions parameterised under [C13385/1...32](#) / [C14385/1...32](#). This response will be executed if a heartbeat event occurs in the consumer.



Tip!

- [C13346](#) / [C14346](#) displays the activity of each monitoring function parameterised under [C13385/1...32](#) / [C14385/1...32](#) in a bit-coded manner.
- [C13347/1...32](#) / [C14347/1...32](#) display the heartbeat state of the monitored nodes.
- [C13625](#) / [C14625](#) serves to set which state change is to occur in the case of a heartbeat event.

Heartbeat telegram

The heartbeat telegram from the producer has the following identifier:

Identifier (COB-ID) = 1792 + producer node address = 1792 + 1 = 1793 = 0x701

10.3 Emergency telegram

If the error status changes due to the occurrence or elimination of an internal device error, an emergency telegram with the following structure is sent once to the NMT master:

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Emergency error code		Error register	Manufacturer-specific error message				
Low byte	High byte	I-1001	0x00 (reserved)	Low word		High word	
				Low byte	High byte	Low byte	High byte
See table below			- With emergency error code "0x1000: Lenze error number" (value displayed in C00168) - With all other emergency error codes, the value "0" is entered here.				

Emergency error code	Error register	Cause
0x0000	0xxx	One of several errors eliminated
	0x00	Single error eliminated (no more errors)
0x1000	0x01	Generic error - In the standard device, an error has occurred with the error response "Fault", "Trouble", "Quick stop by trouble", "Warning", "Warning locked" or "System fault". - Error message is the Lenze error number (C00168). - For error cause see error description (C00166).
0x3100	0x01	Supply voltage of standard device faulty or failed
0x8100	0x11	Communication error (warning)
0x8130	0x11	Life guarding error or heartbeat error
0x8150	0x11	Collision of identifiers (COB-IDs): An identifier parameterised for reception is also used for transmission.
0x8210	0x11	PDO length is shorter than the expected PDO length
0x8220	0x11	PDO length is longer than the expected PDO length
0x8700	0x11	Sync telegram monitoring

Example

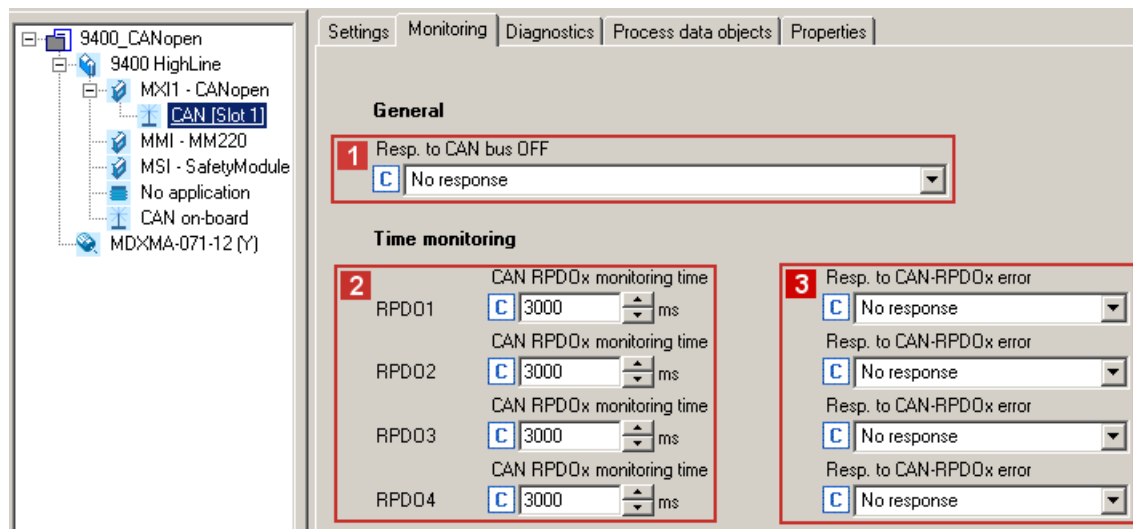
1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Emergency error code		Error register	Manufacturer-specific error message				
0x00	0x10	0x01	0x00 (reserved)	0x1B	0x00	0x7B	0x00
Generic error			Lenze error message "0x007B001B: Encoder wire breakage". Corresponding "error-free" message: value "0x00000000"				



Tip!

A detailed description can be gathered from the CANopen specification DS301, V4.02.

10.4 Settings in the »Engineer«



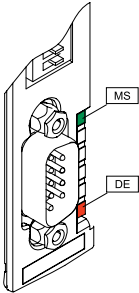
Go to the **Monitoring** tab to ...

- ▶ set the **1** response for the CAN node changing to the "bus OFF" state ([C13959](#) / [C14595](#)).
- ▶ set the **2** monitoring time for RPDOx ([C13357/1...4](#) / [C14357/1...4](#)).
 - If a different value than "0" is entered, the RPDO is expected after the time set has elapsed.
 - If the RPDO is not received within the monitoring time, an error response parameterised under **3** [C13591/1...4](#) / [C14591/1...4](#) is initiated.

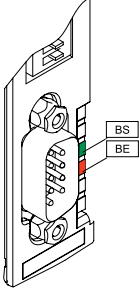
11 **Diagnostics**

11.1 **LED status displays**

Status displays MS and DE

LEDs	Status	Description
 E94YCCA001E	MS (green) on	The communication module is supplied with voltage.
	DE (red) on	The communication module is not accepted by the standard device (see notes in the documentation for the standard device).

Status displays BS and BE (connection status to the system bus (CANopen))

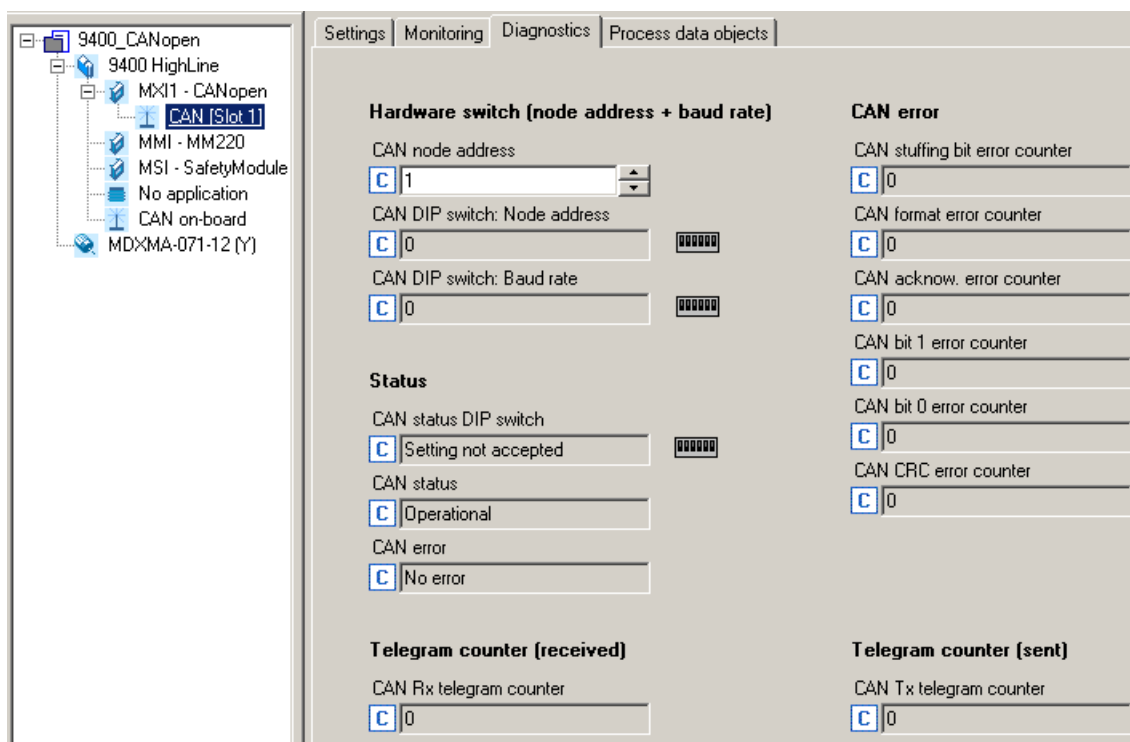
LEDs	Description	
	BS (green)	CANopen status
	BE (red)	CANopen error

E94YCCA001D

LED status display		Meaning	
BS (green)	BE (red)	CANopen status	CANopen error
BE lit only 		-	Bus off
BS and BE flickering 		Automatic detection of the baud rate is active.	
BS blinking every 0.2 s BE is off 		Pre-operational	-
BS blinking every 0.2 s BE blinking once, 1 s off 			Warning limit reached
BS blinking every 0.2 s BE blinking twice, 1 s off 			Node guard event
BS lit only 		Operational	-
BS lit BE blinking once, 1 s off 			Warning limit reached
BS lit BE blinking twice, 1 s off 			Node guard event
BS lit BE blinking three times, 1 s off 			Sync message error
BS blinking every 1 s BE is off 		Stopped	-
BS blinking every 1 s BE blinking once, 1 s off 			Warning limit reached
BS blinking every 1 s BE blinking twice, 1 s off 			Node guard event

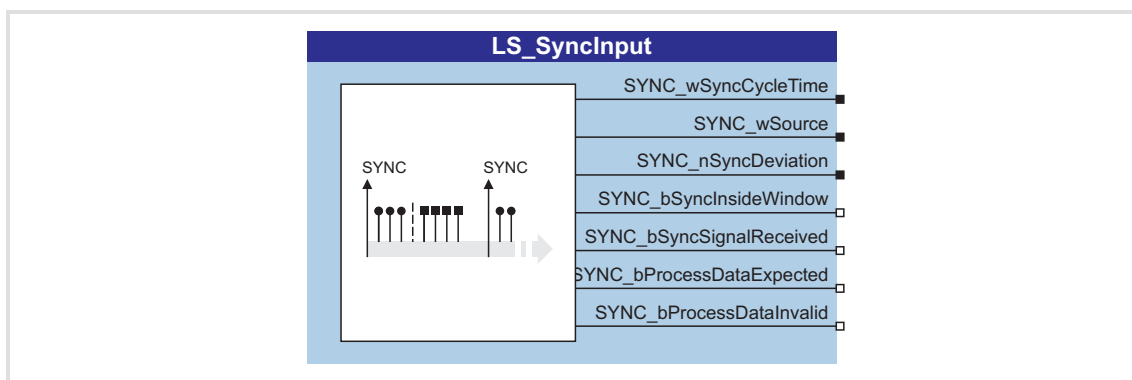
11.2 Diagnostics with the »Engineer«

In the »Engineer«, diagnostic information for CAN is provided on the **Diagnostics** tab.



LS_SyncInput system block

In the »FB Editor«, the **LS_SyncInput** system block provides status information for the received sync telegram.



Software manual/»Engineer« online help for Servo Drive 9400

Here you can find detailed information on the **LS_SyncInput** system block.

12 Error messages

This chapter supplements the error list in the software manual and the »Engineer« online help for the Servo Drive 9400 with the error messages of the communication module.



Software manual/»Engineer« online help for Servo Drive 9400

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

12.1 Short overview of the CANopen error messages

The following table lists all error messages of the communication module in the numerical order of the error numbers. In addition, the preset error response and – where available – the parameter for setting the error response are indicated.



Tip!

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

Error number hex	dec	Designation	Response (Lenze setting)	Can be set in
0x008c0017	9175063	MXI1: CAN module is missing or is incompatible	Fault	-
0x008c0018	9175064	MXI2: CAN module is missing or is incompatible	Fault	-
0x009d0000	10289152	CAN module (MXI1): bus-off	Information	C13595
0x009d0001	10289153	CAN module (MXI1): invalid node address 0	Warning	-
0x009d0002	10289154	CAN module (MXI1): basic configuration invalid	Warning locked	-
0x009e0000	10354688	CAN module (MXI1): heartbeat error index 1 ... 32	No response	C13613/1...32
0x009e0020	10354720	CAN module (MXI1): life guarding error	No response	C13614
0x009e0021	10354721	CAN module (MXI1): NMT slave configuration incomplete	Warning locked	-
0x009f0000	10420224	CAN module (MXI1): emergency configuration incomplete	Warning locked	-
0x00a00000	10485760	CAN module (MXI1): node guarding error 1 ... 32	No response	C13612/1...32
0x00a00020	10485792	CAN module (MXI1): NMT master configuration incomplete	Warning locked	-
0x00a10000	10551296	CAN module (MXI1) RPDO1: telegram n. received or faulty	No response	C13591/1
0x00a10001	10551297	CAN module (MXI1) RPDO2: telegram n. received or faulty	No response	C13591/2
0x00a10002	10551298	CAN module (MXI1) RPDO3: telegram n. received or faulty	No response	C13591/3
0x00a10003	10551299	CAN module (MXI1) RPDO4: telegram n. received or faulty	No response	C13591/4
0x00a10004	10551300	CAN module (MXI1) RPDO5: telegram n. received or faulty	No response	C13591/5
0x00a10005	10551301	CAN module (MXI1) RPDO6: telegram n. received or faulty	No response	C13591/6
0x00a10006	10551302	CAN module (MXI1) RPDO7: telegram n. received or faulty	No response	C13591/7
0x00a10007	10551303	CAN module (MXI1) RPDO8: telegram n. received or faulty	No response	C13591/8
0x00a10008	10551304	CAN module (MXI1) PDO manager: configuration incomplete	Warning locked	-
0x00a20000	10616832	CAN module (MXI1) SDO server: configuration incomplete	Warning locked	-
0x00a30000	10682368	CAN module (MXI1) SDO client: configuration incomplete	Warning locked	-
0x00ac0000	11272192	CAN module (MXI2): bus-off	Information	C14595
0x00ac0001	11272193	CAN module (MXI2): invalid node address 0	Warning	-
0x00ac0002	11272194	CAN module (MXI2): basic configuration invalid	Warning locked	-

E94AYCCA communication manual (CANopen®)

Error messages

Short overview of the CANopen error messages

Error number hex	dec	Designation	Response (Lenze setting)	Can be set in
0x00ad0000 ...	11337728	CAN module (MXI2): heartbeat error index 1 ... 32	No response	C14613/1...32
0x00ad001f	11337759			
0x00ad0020	11337760	CAN module (MXI2): life guarding error	No response	C14614
0x00ad0021	11337761	CAN module (MXI2): NMT slave configuration incomplete	Warning locked	-
0x00ae0000	11403264	CAN module (MXI2): emergency configuration incomplete	Warning locked	-
0x00af0000 ...	11468800	CAN module (MXI2): node guarding error 1 ... 32	No response	C14612/1...32
0x00af001f	11468831			
0x00af0020	11468832	CAN module (MXI2): NMT master configuration incomplete	Warning locked	-
0x00b00000	11534336	CAN module (MXI2) RPDO1: telegram n. received or faulty	No response	C14591/1
0x00b00001	11534337	CAN module (MXI2) RPDO2: telegram n. received or faulty	No response	C14591/2
0x00b00002	11534338	CAN module (MXI2) RPDO3: telegram n. received or faulty	No response	C14591/3
0x00b00003	11534339	CAN module (MXI2) RPDO4: telegram n. received or faulty	No response	C14591/4
0x00b00004	11534340	CAN module (MXI2) RPDO5: telegram n. received or faulty	No response	C14591/5
0x00b00005	11534341	CAN module (MXI2) RPDO6: telegram n. received or faulty	No response	C14591/6
0x00b00006	11534342	CAN module (MXI2) RPDO7: telegram n. received or faulty	No response	C14591/7
0x00b00007	11534343	CAN module (MXI2) RPDO8: telegram n. received or faulty	No response	C14591/8
0x00b00008	11534344	CAN module (MXI2) PDO manager: configuration incomplete	Warning locked	-
0x00b10000	11599872	CAN module (MXI2) SDO server: configuration incomplete	Warning locked	-
0x00b20000	11665408	CAN module (MXI2) SDO client: configuration incomplete	Warning locked	-

12.2 Possible causes and remedies

This chapter includes a list of all error messages of the communication module in numerically ascending order of the error number. Possible causes and remedies as well as responses to error messages are described in detail.

► [Short overview of the CANopen error messages](#) (📖 89)

MXI1: CAN module is missing or is incompatible [0x008c0017]

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 CANopen communication module in module slot MXI1 is incompatible with the application.	Use a communication module supported by the application.	

MXI2: CAN module is missing or is incompatible [0x008c0018]

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 CANopen communication module in module slot MXI2 is incompatible with the application.	Use a communication module supported by the application.	

CAN module (MXI1): bus-off [0x009d0000]

Response (Lenze setting printed in bold)		Setting: C13595 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI1: "bus-off" status • Cable defect (e. g. loose contact) • Too many faulty telegrams received. • Two CAN nodes have the same node address.	• Eliminate fault in the ambience (e. g. EMC). • Eliminate loose contact, tighten adapter. • Assign different node addresses. ► Setting the node address (📖 33)	

CAN module (MXI1): invalid node address 0 [0x009d0001]

Response (Lenze setting printed in bold)		Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ Warning ☐ Information		
Cause	Remedy	
CANopen communication module in MXI1: initialisation error • The hardware allocation of the node address was selected via DIP switches and the DIP switches of the node address are all set to "0". • Instead of the impermissible node address 0, node address 1 is used.	• Set a different node address than "0" and carry out mains switching. ► Setting the node address (📖 33)	

CAN module (MX11): basic configuration invalid [0x009d0002]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information			
Cause		Remedy	
CANopen communication module in MX11: configuration error - Faulty download of an »Engineer« or »PLC Designer« project. - Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project.		- Repeat download - Correct CAN settings in the project and have project regenerated.	

CAN module (MX11): heartbeat error index 1 ... 32 [0x009e0000 ... 0x009e001f]

Response (Lenze setting printed in bold)		Setting: C13613/1...32 (☒ Adjustable response)	
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information			
Cause		Remedy	
CANopen communication module in MX11: cyclic node monitoring The CAN node has not received a heartbeat telegram from node 1 ... 32 within the time defined.		- Reactivate CAN node by mains switching, restart of the controller (C00002 = 11000) or "Reset node" (C00002 = 92). - Parameterise heartbeat producer monitoring time again or switch off monitoring. - Reset possibly active error status. Tip: Before switching the mains and restarting the controller, save the current parameter set (C00002 = 11). ▶ Heartbeat protocol (□ 80)	

CAN module (MX11): life guarding error [0x009e0020]

Response (Lenze setting printed in bold)		Setting: C13614 (☒ Adjustable response)	
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information			
Cause		Remedy	
CANopen communication module in MX11: cyclic node monitoring Slave response: The maximum time between two node guarding telegrams (RTR) from the master has been exceeded.		Parameterise life guarding monitoring time again or switch it off. ▶ Node guarding protocol (□ 74)	

CAN module (MX11): NMT slave configuration faulty [0x009e0021]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information			
Cause		Remedy	
CANopen communication module in MX11: on the CAN slave a configuration error has occurred in the network management. - Faulty download of an »Engineer« or »PLC Designer« project. - Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project. - Node guarding or heartbeat parameterised incorrectly.		- Repeat download - Correct CAN settings in the project and have project regenerated.	

CAN module (MX11): emergency configuration faulty [0x009f0000]

Response (Lenze setting printed in bold) ☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information	
Setting: not possible	
Cause	Remedy
CANopen communication module in MX11: CAN emergency configuration is faulty. - Faulty download of an »Engineer« or »PLC Designer« project. - Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project.	- Repeat download - Correct CAN settings in the project and have project regenerated.

CAN module (MX11): node guarding error 1 ... 32 [0x00a00000 ... 0x00a0001f]

Response (Lenze setting printed in bold) ☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information	
Setting: C13612/1...32 (☒ Adjustable response)	
Cause	Remedy
CANopen communication module in MX11: cyclic node monitoring The CAN master has not received a response to a node guarding telegram (RTR) by the node 1 ... 32 within the time defined.	- Reactivate CAN node by mains switching, restart of the controller (C00002 = 11000) or "Reset node" (C00002 = 92). - Parameterise node guarding monitoring time again or switch off the monitoring. - Reset possibly active error status. Tip: Before switching the mains and restarting the controller, save the current parameter set (C00002 = 11). ► Node guarding protocol (□ 74)

CAN module (MX11): NMT master configuration faulty [0x00a00020]

Response (Lenze setting printed in bold) ☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information	
Setting: not possible	
Cause	Remedy
CANopen communication module in MX11: on the CAN master a configuration error has occurred in the network management. - Faulty download of an »Engineer« or »PLC Designer« project. - Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project. - Node guarding or heartbeat parameterised incorrectly.	- Repeat download - Correct CAN settings in the project and have project regenerated.

CAN module (MX11) RPDO1: telegram not received or faulty [0x00a10000]

Response (Lenze setting printed in bold) ☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information	
Setting: C13591/1 / C14591/1 (☒ Adjustable response)	
Cause	Remedy
CANopen communication module in MX11: CAN-IN1 error - PDO telegram length incorrect. - Error during transmission. - PDO time monitoring has been triggered.	- Set correct telegram length for the CAN master (transmitter). - Eliminate fault in the ambience (e. g. EMC). - Set different time monitoring or switch off time monitoring.

E94AYCCA communication manual (CANopen®)

Error messages

Possible causes and remedies

CAN module (MXI1) RPDO2: telegram not received or faulty [0x00a10001]

Response (Lenze setting printed in bold)		Setting: C13591/2 / C14591/2 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI1: CAN-IN2 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI1) RPDO3: telegram not received or faulty [0x00a10002]

Response (Lenze setting printed in bold)		Setting: C13591/3 / C14591/3 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI1: CAN-IN3 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI1) RPDO4: telegram not received or faulty [0x00a10003]

Response (Lenze setting printed in bold)		Setting: C13591/4 / C14591/4 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI1: CAN-IN4 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI1) RPDO5: telegram not received or faulty [0x00a10004]

Response (Lenze setting printed in bold)		Setting: C13591/5 / C14591/5 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI1: CAN-IN5 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI1) RPDO6: telegram not received or faulty [0x00a10005]

Response (Lenze setting printed in bold)		Setting: C13591/6 / C14591/6 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI1: CAN-IN6 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI1) RPDO7: telegram not received or faulty [0x00a10006]

Response (Lenze setting printed in bold)	Setting: C13591/7 / C14591/7 (☑ Adjustable response)
---	---

☑ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information
--

Cause	Remedy
CANopen communication module in MXI1: CAN-IN7 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.

CAN module (MXI1) RPDO8: telegram not received or faulty [0x00a10007]

Response (Lenze setting printed in bold)	Setting: C13591/8 / C14591/8 (☑ Adjustable response)
---	---

☑ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information
--

Cause	Remedy
CANopen communication module in MXI1: CAN-IN8 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.

CAN module (MXI1) PDO manager: configuration faulty [0x00a10008]

Response (Lenze setting printed in bold)	Setting: not possible
---	------------------------------

☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information
--

Cause	Remedy
CANopen communication module in MXI1: CAN-PDO configuration error • Erroneous project download. • Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project. • Mapping variables have incorrect CANopen indexes according to DS405.	• Repeat download. • Correct CAN settings in the project and have project regenerated.

CAN module (MXI1) SDO server: configuration faulty [0x00a20000]

Response (Lenze setting printed in bold)	Setting: not possible
---	------------------------------

☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information
--

Cause	Remedy
CANopen communication module in MXI1: a configuration error has occurred in the CAN-SDO server. • Erroneous project download. • Invalid SDO server settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project.	• Repeat download. • Correct CAN settings in the project and have project regenerated.

E94AYCCA communication manual (CANopen®)

Error messages

Possible causes and remedies

CAN module (MXI1) SDO client: configuration faulty [0x00a30000]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information			
Cause	Remedy		
CANopen communication module in MXI1: a configuration error has occurred in the CAN-SDO client. • Erroneous project download. • Invalid SDO client settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project.	• Repeat download • Correct CAN settings in the project and have project regenerated.		

CAN module (MXI2): bus-off [0x00ac0000]

Response (Lenze setting printed in bold)		Setting: C14595 ☒ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI2: "bus-off" status • Cable defect (e. g. loose contact) • Too many faulty telegrams received. • Two CAN nodes have the same node address.	• Eliminate fault in the ambience (e. g. EMC). • Eliminate loose contact, tighten adapter. • Assign different node addresses. ▶ Setting the node address (📖 33)	

CAN module (MXI2): invalid node address 0 [0x00ac0001]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☒ Warning ☐ Information			
Cause	Remedy		
CANopen communication module in MXI2: initialisation error • The hardware allocation of the node address was selected via DIP switches and the DIP switches of the node address are all set to "0". • Instead of the impermissible node address 0, node address 1 is used.	• Set a different node address than "0" and carry out mains switching. ▶ Setting the node address (📖 33)		

CAN module (MXI2): basic configuration invalid [0x00ac0002]

Response (Lenze setting printed in bold)		Setting: not possible	
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information			
Cause	Remedy		
CANopen communication module in MXI2: configuration error • Faulty download of an »Engineer« or »PLC Designer« project. • Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project.	• Repeat download • Correct CAN settings in the project and have project regenerated.		

CAN module (MX12): heartbeat error index 1 ... 32 [0x00ad0000 ... 0x00ad001f]

Response (Lenze setting printed in bold)		Setting: C14613/1...32 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MX12: cyclic node monitoring The CAN node has not received a heartbeat telegram from node 1 ... 32 within the time defined.	- Reactivate CAN node by mains switching, restart of the controller (C00002 = 11000) or "Reset node" (C00002 = 92). - Parameterise heartbeat producer monitoring time again or switch off monitoring. - Reset possibly active error status. Tip: Before switching the mains and restarting the controller, save the current parameter set (C00002 = 11). ▶ Heartbeat protocol (□ 80)	

CAN module (MX12): life guarding error [0x00ad0020]

Response (Lenze setting printed in bold)		Setting: C14614 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MX12: cyclic node monitoring Slave response: The maximum time between two node guarding telegrams (RTR) from the master has been exceeded.	Parameterise life guarding monitoring time again or switch it off. ▶ Node guarding protocol (□ 74)	

CAN module (MX12): NMT slave configuration faulty [0x00ad0021]

Response (Lenze setting printed in bold)		Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information		
Cause	Remedy	
CANopen communication module in MX12: on the CAN slave a configuration error has occurred in the network management. - Faulty download of an »Engineer« or »PLC Designer« project. - Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project. - Node guarding or heartbeat parameterised incorrectly.	- Repeat download - Correct CAN settings in the project and have project regenerated.	

CAN module (MX12): emergency configuration faulty [0x00ae0000]

Response (Lenze setting printed in bold)		Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information		
Cause	Remedy	
CANopen communication module in MX12: CAN emergency configuration is faulty. - Faulty download of an »Engineer« or »PLC Designer« project. - Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project.	- Repeat download - Correct CAN settings in the project and have project regenerated.	

E94AYCCA communication manual (CANopen®)

Error messages

Possible causes and remedies

CAN module (MX12): node guarding error 1 ... 32 [0x00af0000 ... 0x00af001f]

Response (Lenze setting printed in bold)		Setting: C14612/1...32 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MX12: cyclic node monitoring The CAN master has not received a response to a node guarding telegram (RTR) by the node 1 ... 32 within the time defined.	- Reactivate CAN node by mains switching, restart of the controller (C00002 = 11000) or "Reset node" (C00002 = 92). - Parameterise node guarding monitoring time again or switch off the monitoring. - Reset possibly active error status. Tip: Before switching the mains and restarting the controller, save the current parameter set (C00002 = 11). ▶ Node guarding protocol (☐ 74)	

CAN module (MX12): NMT master configuration faulty [0x00af0020]

Response (Lenze setting printed in bold)		Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information		
Cause	Remedy	
CANopen communication module in MX12: on the CAN master a configuration error has occurred in the network management. - Faulty download of an »Engineer« or »PLC Designer« project. - Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project. - Node guarding or heartbeat parameterised incorrectly.	- Repeat download - Correct CAN settings in the project and have project regenerated.	

CAN module (MX12) RPDO1: telegram not received or faulty [0x00b00000]

Response (Lenze setting printed in bold)		Setting: C14591/1 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MX12: CAN-IN1 error - PDO telegram length incorrect. - Error during transmission. - PDO time monitoring has been triggered.	- Set correct telegram length for the CAN master (transmitter). - Eliminate fault in the ambience (e. g. EMC). - Set different time monitoring or switch off time monitoring.	

CAN module (MX12) RPDO2: telegram not received or faulty [0x00b00001]

Response (Lenze setting printed in bold)		Setting: C14591/2 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MX12: CAN-IN2 error - PDO telegram length incorrect. - Error during transmission. - PDO time monitoring has been triggered.	- Set correct telegram length for the CAN master (transmitter). - Eliminate fault in the ambience (e. g. EMC). - Set different time monitoring or switch off time monitoring.	

CAN module (MXI2) RPDO3: telegram not received or faulty [0x00b00002]

Response (Lenze setting printed in bold)		Setting: C14591/3 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI2: CAN-IN3 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI2) RPDO4: telegram not received or faulty [0x00b00003]

Response (Lenze setting printed in bold)		Setting: C14591/4 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI2: CAN-IN4 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI2) RPDO5: telegram not received or faulty [0x00b00004]

Response (Lenze setting printed in bold)		Setting: C14591/5 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI2: CAN-IN5 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI2) RPDO6: telegram not received or faulty [0x00b00005]

Response (Lenze setting printed in bold)		Setting: C14591/6 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI2: CAN-IN6 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI2) RPDO7: telegram not received or faulty [0x00b00006]

Response (Lenze setting printed in bold)		Setting: C14591/7 (☑ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI2: CAN-IN7 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI2) RPDO8: telegram not received or faulty [0x00b00007]

Response (Lenze setting printed in bold)		Setting: C14591/8 (☒ Adjustable response)
☒ None ☐ System fault ☒ Fault ☒ Trouble ☒ Quick stop by trouble ☒ Warning locked ☒ Warning ☒ Information		
Cause	Remedy	
CANopen communication module in MXI2: CAN-IN8 error • PDO telegram length incorrect. • Error during transmission. • PDO time monitoring has been triggered.	• Set correct telegram length for the CAN master (transmitter). • Eliminate fault in the ambience (e. g. EMC). • Set different time monitoring or switch off time monitoring.	

CAN module (MXI2) PDO manager: configuration faulty [0x00b00008]

Response (Lenze setting printed in bold)		Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information		
Cause	Remedy	
CANopen communication module in MXI2: CAN-PDO configuration error • Erroneous project download. • Invalid CAN settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project. • Mapping variables have incorrect CANopen indexes according to DS405.	• Repeat download. • Correct CAN settings in the project and have project regenerated.	

CAN module (MXI2) SDO server: configuration faulty [0x00b10000]

Response (Lenze setting printed in bold)		Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information		
Cause	Remedy	
CANopen communication module in MXI2: a configuration error has occurred in the CAN-SDO server. • Erroneous project download. • Invalid SDO server settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project.	• Repeat download. • Correct CAN settings in the project and have project regenerated.	

CAN module (MXI2) SDO client: configuration faulty [0x00b20000]

Response (Lenze setting printed in bold)		Setting: not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ Warning locked ☐ Warning ☐ Information		
Cause	Remedy	
CANopen communication module in MXI2: a configuration error has occurred in the CAN-SDO client. • Erroneous project download. • Invalid SDO client settings according to DS301, V4.02 carried out in the »Engineer« or »PLC Designer« project.	• Repeat download • Correct CAN settings in the project and have project regenerated.	

13 Parameter reference

This chapter supplements the parameter list and the table of attributes in the software manual and the »Engineer« online help for the Servo Drive 9400 by the parameters of the E94AYCCA communication module (CANopen).



Software manual/»Engineer« online help for Servo Drive 9400

Here you will find general information about parameters.

13.1 Communication-relevant parameters of the standard device

In this chapter communication-relevant parameters of the Servo Drive 9400 are listed in numerically ascending order.

C00369

Parameter | Name:

C00369 | CAN sync transmission cycle time

Data type: UNSIGNED_16
Index: 11206_d = 2BC6_h

Cycle within which the sync master is to send sync telegrams.

- With "0 ms" (Lenze setting), no sync telegrams are created.
- Mapping of the CANopen object [I-1006](#) (see DS301, V4.02).

► [Synchronisation of PDOs via sync telegram](#) (50)

Setting range (min. value | unit | max. value)

0	ms	65535
---	----	-------

Subcodes	Lenze setting	Information
C00369/1	0 ms	Transmission cycle time for CAN on board
C00369/2	0 ms	Transmission cycle time for CAN module in MXI1/MXI2
C00369/3	0 ms	No function

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C00615

Parameter | Name:

C00615 | Resp. to imp. device config.

Data type: UNSIGNED_32
Index: 23960_d = 5D98_h

Selection of 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	Information
C00615/1	0: No response	Reserved
C00615/2	1: Fault	Resp. to imperm. module in MXI1
C00615/3	1: Fault	Resp. to imperm. module in MXI2
C00615/4	0: No response	Reserved
C00615/5	0: No response	Reserved

☒ Read access

☒ Write access

☐ CINH

☐ PLC STOP

☐ No transfer

Scaling factor: 1

C00636

Parameter Name:		Data type: UNSIGNED_32 Index: 23939 _d = 5D83 _h	
C00636 Resp. to new module in MXI1			
Selection of the response to a new extension module being inserted into module slot MXI1.			
Selection list (Lenze setting printed in bold)			
1	Fault		
6	Information		
5	Warning		
4	Warning locked		
3	Quick stop by trouble		
0	No response		
☒ Read access		☒ Write access	
☐ CINH		☐ PLC STOP	
☐ No transfer		Scaling factor: 1	

C00637

Parameter Name:		Data type: UNSIGNED_32 Index: 23939 _d = 5D83 _h	
C00637 Resp. to new module in MXI2			
Selection of the response to a new extension module being inserted into module slot MXI2.			
Selection list (Lenze setting printed in bold)			
1	Fault		
6	Information		
5	Warning		
4	Warning locked		
3	Quick stop by trouble		
0	No response		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1	

C01120

Parameter | Name:

C01120 | Sync sourceData type: UNSIGNED_8
Index: 23455_d = 5B9F_h

Selection of the source for the synchronisation signals.

- The drive can only be synchronised by one source.

Note:

Select "2: CAN module" for the CANopen communication module (E94AYCCA).

► [Synchronisation of PDOs via sync telegram](#) (50)

Selection list (Lenze setting printed in bold)

0	Off
1	CAN on board
2	CAN module
4	Module in MXI1
5	Module in MXI2
6	Digital input 1
7	Digital input 2
8	Digital input 3
9	Digital input 4
10	Digital input 5
11	Digital input 6
12	Digital input 7
13	Digital input 8

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01121

Parameter | Name:

C01121 | Sync cycle timeData type: UNSIGNED_32
Index: 23454_d = 5B9E_h

Time interval at which the phase control loop (PLL) in the controller expects the synchronisation signals.

- The time interval set must correspond to the cycle of the synchronisation source.

Note:

For synchronisation via the system bus (CANopen), only integer multiples of 1000 µs should be set.

Example: For the system bus, the interval between two synchronisation signals is set to 2 ms. If the system bus is to be used as the synchronisation source, a synchronisation cycle of 2000 µs must be set in C01121.

► [Synchronisation of PDOs via sync telegram](#) (50)

Setting range (min. value unit max. value)			Lenze setting
250	µs	20000	1000 µs

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01122

Parameter | Name:

C01122 | Sync phase position

Data type: UNSIGNED_32
Index: 23453_d = 5B9D_h

The phase position defines the zero point of time for the application relating to the synchronisation signal (bus cycle). Since PDO processing is integrated in the system part of the application, the instant of the PDO acceptance also changes if the phase position is changed.

- If 0 is set, the application is started together with the synchronisation signal.
- If a value > 0 is set, the application starts by the set time interval before the synchronisation signal arrives (the phase position acts negatively).

Example:

If the phase position is set to 400 µs, the system part of the application starts 400 µs before the synchronisation signal arrives.

From software version V3.0:

The effect of the sync phase position can be influenced by the application cycle set in [C01130](#). For the Lenze setting of [C01130](#), the behaviour remains as before.

► [Synchronisation of PDOs via sync telegram](#) (50)

Setting range (min. value unit max. value)			Lenze setting
0	µs	64000	400 µs
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01123

Parameter | Name:

C01123 | Sync tolerance

Data type: UNSIGNED_32
Index: 23452_d = 5B9C_h

Time slot for monitoring the synchronisation signal via the **LS_SyncInput** system block

- If the last synchronisation signal has been within this time slot around the expected value, the **SYNC_bSyncInsideWindow** output of the **LS_SyncInput** system block is set to TRUE.
- This setting does not affect the synchronisation process.

► [Synchronisation of PDOs via sync telegram](#) (50)

Setting range (min. value unit max. value)			Lenze setting
0	µs	1000	0 µs
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C01124

Parameter | Name:

C01124 | Sync-PLL increment

Data type: UNSIGNED_8

Index: 23451_d = 5B9B_h

When the cycle times of the synchronisation signal and the phase control loop (PLL) differ, this setting defines the increment to be used to readjust the phase control loop.

- If the system bus (CANopen) is used as synchronisation source, the recommended time increment for deviations is 109 ns.

► [Synchronisation of PDOs via sync telegram](#) (□ 50)

Selection list (Lenze setting printed in bold)

1	7 ns
2	15 ns
3	23 ns
4	31 ns
5	39 ns
6	46 ns
7	54 ns
8	62 ns
9	70 ns
10	78 ns
11	85 ns
12	93 ns
13	101 ns
14	109 ns
15	117 ns
16	125 ns
17	132 ns
18	140 ns
19	148 ns
20	156 ns
21	164 ns
22	171 ns
23	179 ns
24	187 ns
25	195 ns
26	203 ns
27	210 ns

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C01130

Parameter | Name:

C01130 | CAN SYNC application cycle

Data type: UNSIGNED_16
Index: 23445_d = 5B95_h

[From software version V3.0](#)

This parameter influences the effect of the sync phase position ([C01122](#)) with regard to the instant of acceptance of the synchronous PDOs by the application or the instant of transmission of the synchronous PDOs to the system bus (CANopen).

- The resulting PDO delay can be calculated with the following formula, taking an internal processing time of 150 µs into consideration: PDO delay = ([C01121](#) - [C01122](#) + 150 µs) modulo C01130
- For the Lenze setting, the behaviour remains as before; the sync phase position ([C01122](#)) is always calculated "modulo 1000".
- The set value is automatically rounded down to multiples of 1000 µs.

► [Effect of C01130 on the sync phase position](#) (📖 54)

Note: Setting the application cycle to a higher value than the sync cycle time ([C01121](#)) results in an undefined response. The same applies if the value set for the sync phase position ([C01122](#)) is higher than the sync cycle time ([C01121](#)). In this case, the drive usually cannot send synchronous PDOs on the system bus anymore.

► [Synchronisation of PDOs via sync telegram](#) (📖 50)

Setting range (min. value unit max. value)			Lenze setting
1000	µs	65000	1000 µs
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1



Note!

The standard device codes **C01501** and **C01502** are not effective when using the CANopen communication module.

13.2 Parameters of the communication module for slot MXI1

In this chapter, the parameters of the E94AYCCA communication module (CANopen) for slot MXI1 of the Servo Drive 9400 are listed in numerically ascending order.

C13311

Parameter Name: C13311 CAN TPDO1 mask byte x		Data type: BITFIELD_8 Index: 11264 _d = 2C00 _h
For each byte of the TPDO1 a mask can be parameterised in the corresponding subcode.		
• In case of event-controlled transmission of a PDO, only the masked bits are used for the event control. • Mask "0x0" means that no bit of the corresponding byte triggers the transmission. • Mask "0xff" means that each bit of the corresponding byte can trigger the transmission.		
▶ Masking of the TPDOs for event control (49)		
Setting range		
0x00		0xFF
Value is bit-coded:		
Bit 0	Mask bit 0	
...	...	
Bit 7	Mask bit 7	
Subcodes	Lenze setting	Information
C13311/1	0x00	Mask for byte 1 ... byte 8 of TPDO1
...		
C13311/8		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C13312

Parameter Name: C13312 CAN TPDO2 mask byte x		Data type: BITFIELD_8 Index: 11263 _d = 2BFF _h
For each byte of the TPDO2 a mask can be parameterised in the corresponding subcode.		
• In case of event-controlled transmission of a PDO, only the masked bits are used for the event control. • Mask "0x0" means that no bit of the corresponding byte triggers the transmission. • Mask "0xff" means that each bit of the corresponding byte can trigger the transmission.		
▶ Masking of the TPDOs for event control (49)		
Setting range		
0x00		0xFF
Value is bit-coded:		
Bit 0	Mask bit 0	
...	...	
Bit 7	Mask bit 7	
Subcodes	Lenze setting	Information
C13312/1	0x00	Mask for byte 1 ... byte 8 of TPDO2
...		
C13312/8		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C13313

Parameter | Name:

C13313 | CAN TPDO3 mask byte x

Data type: BITFIELD_8
Index: 11262_d = 2BFE_h

For each byte of the TPDO3 a mask can be parameterised in the corresponding subcode.

- In case of event-controlled transmission of a PDO, only the masked bits are used for the event control.
- Mask "0x0" means that no bit of the corresponding byte triggers the transmission.
- Mask "0xff" means that each bit of the corresponding byte can trigger the transmission.

[▶ Masking of the TPDOs for event control](#) (49)

Setting range		
0x00		0xFF
Value is bit-coded:		
Bit 0	Mask bit 0	
...	...	
Bit 7	Mask bit 7	
Subcodes	Lenze setting	Information
C13313/1	0x00	Mask for byte 1 ... byte 8 of TPDO3
...		
C13313/8		

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C13314

Parameter | Name:

C13314 | CAN TPDO4 mask byte x

Data type: BITFIELD_8

Index: 11261_d = 2BFD_h

For each byte of the TPDO4 a mask can be parameterised in the corresponding subcode.

- In case of event-controlled transmission of a PDO, only the masked bits are used for the event control.
- Mask "0x0" means that no bit of the corresponding byte triggers the transmission.
- Mask "0xff" means that each bit of the corresponding byte can trigger the transmission.

▶ [Masking of the TPDOs for event control](#) (📖 49)

Setting range		
0x00		0xFF
Value is bit-coded:		
Bit 0	Mask bit 0	
...	...	
Bit 7	Mask bit 7	
Subcodes	Lenze setting	Information
C13314/1	0x00	Mask for byte 1 ... byte 8 of TPDO4
...		
C13314/8		

☒ Read access
 ☒ Write access
 ☐ CINH
 ☐ PLC STOP
 ☐ No transfer

Scaling factor: 1

C13320

Parameter | Name:

C13320 | CAN TPDOx identifier

Data type: BITFIELD_32

Index: 11255_d = 2BF7_h

Identifier for TPDO1 ... TPDO4

- If bit 31 is set (0x8nnnnnnn), the TPDO is deactivated.
- The basic setting is according to the "Predefined Connection Set".
- Mapping of the CANopen objects [I-1800/1](#) ... [I-1803/1](#) (see DS301, V4.02).

▶ [Identifiers of the process data objects](#) (46)

Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • Bit 31: PDO invalid
...	...	
Bit 31	PDO invalid	
Subcodes	Lenze setting	Information
C13320/1	0x00000181	Identifier TPDO1 • After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x180 + node address is set by default.
C13320/2	0x00000281	Identifier TPDO2 • After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x280 + node address is set by default.
C13320/3	0x00000381	Identifier TPDO3 • After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x380 + node address is set by default.
C13320/4	0x00000481	Identifier TPDO4 • After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x480 + node address is set by default.
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C13321

Parameter Name: C13321 CAN RPDOx identifier		Data type: BITFIELD_32 Index: 11254 _d = 2BF6 _h
Identifier for RPDO1 ... RPDO4		
- If bit 31 is set (0x8nnnnnnn), the RPDO is deactivated. - The basic setting is according to the "Predefined Connection Set". - Mapping of the CANopen objects I-1400/1 ... I-1403/1 (see DS301, V4.02).		
▶ Identifiers of the process data objects (46)		
Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	- Bit 0 ... 10: COB-ID - Bit 11 ... 30: Reserved - Bit 31: PDO invalid
...	...	
Bit 31	PDO invalid	
Subcodes	Lenze setting	Information
C13321/1	0x00000201	Identifier RPDO1 After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x200 + node address is set by default.
C13321/2	0x00000301	Identifier RPDO2 After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x300 + node address is set by default.
C13321/3	0x00000401	Identifier RPDO3 After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x400 + node address is set by default.
C13321/4	0x00000501	Identifier RPDO4 After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x500 + node address is set by default.
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C13322

Parameter Name: C13322 CAN TPDOx Tx mode		Data type: UNSIGNED_8 Index: 11253 _d = 2BF5 _h
TPDO transmission type according to DS301, V4.02		
- The types 0 (acyclic sync), 1-240 (cyclic sync), 254 (event-controlled manufacturer-specific), 255 (event-controlled device-profile-specific) are supported. - The basic setting for all PDOs is the event-controlled setting "254". - Mapping of the CANopen objects I-1800/2 ... I-1803/2 (see DS301, V4.02).		
▶ Transmission type (47)		
Setting range (min. value unit max. value)		
0		255
Subcodes	Lenze setting	Information
C13322/1	254	Transmission mode for TPDO1 ... TPDO4
...		
C13322/4		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C13323

Parameter | Name:

C13323 | CAN RPDOx Rx mode

Data type: UNSIGNED_8
Index: 11252_d = 2BF4_h

RPDO transmission type according to DS301, V4.02

- For the RPDO serves as monitoring setting in the case of sync-controlled PDOs.
- The types 0 (acyclic sync), 1-240 (cyclic sync), 254 (event-controlled manufacturer-specific), 255 (event-controlled device-profile-specific) are supported.
- The basic setting for all PDOs is the event-controlled setting "254".
- Mapping of the CANopen objects [I-1400/2](#) ... [I-1403/2](#) (see DS301, V4.02).

[Transmission type](#) (📖 47)

Setting range (min. value unit max. value)		
0		255
Subcodes	Lenze setting	Information
C13323/1	254	Transmission mode for RPDO1 ... RPDO4
...		
C13323/4		

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Scaling factor: 1

C13324

Parameter Name:			Data type: UNSIGNED_16 Index: 11251 _d = 2BF3 _h		
C13324 CAN TPDOx delay time					
TPDO inhibit time according to DS301, V4.02					
• Minimum time between the transmission of two identical TPDOs. • The time is entered in 1/10 ms and automatically rounded to full milliseconds by the code. • Mapping of the CANopen objects I-1800/3 ... I-1803/3 (see DS301, V4.02).					
Setting range (min. value unit max. value)					
0	1/10 ms	65535			
Subcodes		Lenze setting	Information		
C13324/1	0 1/10 ms		Delay time for TPDO1 ... TPDO4		
...					
C13324/4					
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1		

C13343

Parameter Name:			Data type: UNSIGNED_32 Index: 11232 _d = 2BE0 _h					
C13343 CAN TPDO counter								
Counter for TPDO1 ... TPDO4								
Display range (min. value unit max. value)								
0		4294967295						
Subcodes						Information		
C13343/1								
...								
C13343/4								
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1					

C13344

Parameter Name: C13344 CAN RPDO counter		Data type: UNSIGNED_32 Index: 11231 _d = 2BDF _h
Counter for RPDO1 ... RPDO4		
Display range (min. value unit max. value)		
0		4294967295
Subcodes		Information
C13344/1		
...		
C13344/4		
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C13345

Parameter Name: C13345 CAN error		Data type: UNSIGNED_8 Index: 11230 _d = 2BDE _h
Display of the active CAN network error		
Selection list (read only)		
0	No error	
1	Guard Event	
2	Warning	
3	Bus off	
4	Sync telegram error	
6	CAN controller overflow	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C13346

Parameter Name: C13346 CAN heartbeat activity		Data type: BITFIELD_32 Index: 11229 _d = 2BDD _h
▶ Heartbeat protocol (80)		
Display range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		
Bit 0	Heartbeat node 1	
...	...	
Bit 31	Heartbeat node 32	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C13347

Parameter Name: C13347 CAN heartbeat status		Data type: UNSIGNED_8 Index: 11228 _d = 2BDC _h
Status of node 1 ... 32 ▶ Heartbeat protocol (□ 80)		
Selection list (read only)		
0	Unknown	
4	Stopped	
5	Operational	
127	Pre-operational	
Subcodes		Information
C13347/1		
...		
C13347/32		
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C13348

Parameter Name: C13348 CAN status DIP switch		Data type: UNSIGNED_8 Index: 11227 _d = 2BDB _h
- The value "1" means that the CAN DIP switches have been recognised during power-on and a valid baud rate and node address have been set. - The value "0" means that either no CAN DIP switch or no valid setting has been recognised or that the setting has been overwritten by code C13350 or C13351.		
Selection list (read only)		Information
0	Setting not accepted	
1	Setting accepted	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C13349

Parameter Name: C13349 CAN setting of DIP switch		Data type: UNSIGNED_8 Index: 11226 _d = 2BDA _h
CAN DIP switch setting during last power-on ▶ Possible settings via DIP switches (□ 33)		
Display range (min. value unit max. value)		
0		255
Subcodes		Information
C13349/1		Node address
C13349/2		Baud rate: 0: 500 kbps 1: 250 kbps 2: 125 kbps 3: 50 kbps 4: 1 Mbps 5: 20 kbps 6: 10 kbit/s 14: 800 kbps 255: Automatic detection
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

E94AYCCA communication manual (CANopen®)

Parameter reference

Parameters of the communication module for slot MX11

C13350

Parameter | Name:

C13350 | CAN node address

Data type: UNSIGNED_8
Index: 11225_d = 2BD9_h

- A change in the node address is only effective after a "Reset node" (C00002 = 92).
- The basic server channel RX/TX is automatically provided by the node address ([C13372](#) and [C13373](#); both subcode 1).

▶ [Setting the node address](#) ([33](#))

Setting range (min. value unit max. value)			Lenze setting
1		127	1
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C13351

Parameter | Name:

C13351 | CAN baud rate

Data type: UNSIGNED_8
Index: 11224_d = 2BD8_h

A change in the baud rate is only effective after a "Reset node" (C00002 = 92).

▶ [Setting the baud rate](#) ([34](#))

Selection list (Lenze setting printed in bold)	
0	500 kbps
1	250 kbps
2	125 kbps
3	50 kbps
4	1 Mbps
5	20 kbps
6	10 kbps
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	800 kbps
15	Reserved
255	Automatic detection
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	
Scaling factor: 1	

C13352

Parameter | Name:

C13352 | CAN slave/master

Data type: UNSIGNED_8
Index: 11223_d = 2BD7_h

If "1" is entered and saved, the drive will start as a CAN master after mains switching.

Selection list (Lenze setting printed in bold)	
0	Slave
1	Master
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	
Scaling factor: 1	

C13356

Parameter | Name:

C13356 | CAN TPDOx cycle time

Data type: UNSIGNED_16
Index: 11219_d = 2BD3_h

TPDO event time according to DS301, V4.02

- If a different value than "0" is entered, the TPDO is transmitted without further consideration of the transport type after the time set has elapsed.
- Mapping of the CANopen objects [I-1800/5](#) ... [I-1803/5](#) (see DS301, 4.02).

Setting range (min. value unit max. value)		
0	ms	65535
Subcodes	Lenze setting	Information
C13356/1	0 ms	Cycle time for TPDO1 ... TPDO4
...		
C13356/4		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C13357

Parameter | Name:

C13357 | CAN RPDOx monitoring time

Data type: UNSIGNED_16
Index: 11218_d = 2BD2_h

Mapping of the RPDO event time (see DS301, V4.02)

- If a different value than "0" is entered, the RPDO is expected after the time set has elapsed.
- If the RPDO is not received within the expected time, an error message parameterisable under [C13591/1...4](#) can be triggered.

Setting range (min. value unit max. value)		
0	ms	65535
Subcodes	Lenze setting	Information
C13357/1	3000 ms	Monitoring time for RPDO1 ... RPDO4
...		
C13357/4		

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Scaling factor: 1

C13359

Parameter Name:		Data type: UNSIGNED_8 Index: 11216 _d = 2BD0 _h	
C13359 CAN status			
Display of the current CAN network status			
Selection list (read only)			
0	Operational		
1	Pre-operational		
4	Boot-up		
5	Stopped		
7	Reset		
8	Initialisation		
9	Unknown		
10	Baud rate autom. detected		
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1	

C13360

Parameter | Name: **C13360 | CAN telegram and error counter**

Data type: UNSIGNED_16
Index: 11215_d = 2BCF_h

- After mains connection, all counters start with "0".
- After the maximum value has been reached, it is started with "0" again.

Display range (min. value unit max. value)		
0		65535

Subcodes	Information
C13360/1	Stuffing bit error counter More than five identical bits have been detected.
C13360/2	Format error counter CAN frame has not been observed.
C13360/3	Acknowledge error counter No node has confirmed the telegram.
C13360/4	Bit1 error counter "1" should be sent after bus arbitration, but "0" was read.
C13360/5	Bit0 error counter "0" should be sent after bus arbitration, but "1" was read.
C13360/6	CRC error counter Checksum check has indicated an error.
C13360/7	Tx telegram counter Telegrams received without errors.
C13360/8	Rx telegram counter Telegrams sent without errors.

☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Scaling factor: 1

C13361

Parameter | Name: **C13361 | CAN bus load**

Data type: UNSIGNED_32
Index: 11214_d = 2BCE_h

The display of the node peak load (subcodes 4 ... 6) can be reset by mains switching or via a "Reset node" command (C00002 = 92).

Display range (min. value unit max. value)		
0	%	100
Subcodes	Information	
C13361/1	Current node load in Tx direction	
C13361/2	Current node load in Rx direction	
C13361/3	Current node load by faulty telegrams	
C13361/4	Node peak load in Tx direction	
C13361/5	Node peak load in Rx direction	
C13361/6	Node peak load by faulty telegrams	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C13367

Parameter | Name:

C13367 | CAN sync Rx identifier

Data type: UNSIGNED_32
Index: 11208_d = 2BC8_h

Identifier with which the sync slave is to receive sync telegrams.

- Mapping of the CANopen object [I-1005](#) (see DS301, V4.02).

[Synchronisation of PDOs via sync telegram](#) (📖 50)

Setting range (min. value unit max. value)			Lenze setting
0		2047	128
☒ Read access	☒ Write access	☐ CINH	☐ PLC STOP
☐ No transfer			Scaling factor: 1

C13368

Parameter | Name:

C13368 | CAN sync Tx identifier

Data type: UNSIGNED_32
Index: 11207_d = 2BC7_h

Identifier with which the sync master is to send sync telegrams.

- Mapping of the CANopen object [I-1005](#) (see DS301, V4.02).

[Synchronisation of PDOs via sync telegram](#) ( 50)

Setting range (min. value unit max. value)			Lenze setting
0		2047	128
☒ Read access	☒ Write access	☐ CINH ☐ PLC STOP ☐ No transfer	Scaling factor: 1

C13372

Parameter | Name:

C13372 | CAN SDO server Rx identifier

Data type: BITFIELD_32
Index: 11203_d = 2BC3_h

Identifier with which the corresponding SDO server can be reached.

- If bit 31 is set (0x8nnnnnnn), the corresponding SDO server is deactivated.
- Mapping of the CANopen objects [I-1200/1](#) ... [I-1209/1](#) (see DS301, V4.02).

[Identifiers of the parameter data objects](#) ([□ 62](#))

Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • Bit 31: SDO invalid
...	...	
Bit 31	SDO invalid	
Subcodes	Lenze setting	Information
C13372/1	0x00000601	SDO server channel 1 RX • Subcode 1 contains the basic SDO channel. According to DS301, V4.02 it can neither be changed nor deactivated. Writing this subcode has no effect. • The value in subcode 1 results from the node address (C13350) + 0x600.
C13372/2	0x80000000	SDO server channel 2 RX
...	...	...
C13372/10	0x80000000	SDO server channel 10 RX

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Scaling factor: 1

C13373

Parameter Name: C13373 CAN SDO server Tx identifier		Data type: BITFIELD_32 Index: 11202 _d = 2BC2 _h
Identifier with which the corresponding SDO server can carry out transmissions. - If bit 31 is set (0x8nnnnnnn), the corresponding SDO server is deactivated. - Mapping of the CANopen objects I-1200/2 ... I-1209/2 (see DS301, V4.02). Identifiers of the parameter data objects (62)		
Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	- Bit 0 ... 10: COB-ID - Bit 11 ... 30: Reserved - Bit 31: SDO invalid
...	...	
Bit 31	SDO invalid	
Subcodes	Lenze setting	Information
C13373/1	0x00000581	SDO server channel 1 TX - Subcode 1 contains the basic SDO channel. According to DS301, V4.02 it can neither be changed nor deactivated. Writing this subcode has no effect. - The value in subcode 1 results from the node address (C13350) + 0x580.
C13373/2	0x80000000	SDO server channel 2 TX
...	...	...
C13373/10	0x80000000	SDO server channel 10 TX
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C13374

Parameter Name: C13374 CAN SDO client node address		Data type: UNSIGNED_8 Index: 11201 _d = 2BC1 _h
Node address of the client assigned to this server (see DS301, V4.02).		
Setting range (min. value unit max. value)		
1		127
Subcodes	Lenze setting	Information
C13374/1	1	SDO server channel 1 remote client node address - Subcode 1 contains the basic SDO channel. According to DS301, V4.02 it does not have this entry. Writing this subcode has no effect. - The value of subindex 1 is 0.
C13374/2	1	SDO server channel 2 remote client node address
...	...	...
C13374/10	1	SDO server channel 10 remote client node address
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C13375

Parameter | Name:

C13375 | CAN SDO client Rx identifierData type: BITFIELD_32
Index: 11200_d = 2B0_h

Identifier with which the corresponding SDO client can be reached.

- If bit 31 is set (0x8nnnnnnn), the corresponding SDO client channel is deactivated (see DS301, V4.02).
- The client channels need not be parameterised right now. Their functionality will only be required when using the gateway services.

► [Identifiers of the parameter data objects](#) (62)

Setting range			Information
0x00000000		0xFFFFFFFF	
Value is bit-coded:			Information
Bit 0	COB-ID bit 0		
...	...		
Bit 31	SDO invalid		
Subcodes	Lenze setting	Information	
C13375/1	0x80000000	SDO client channel 1 RX ... 10 RX	
...			
C13375/10			
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1			

C13376

Parameter | Name:

C13376 | CAN SDO client Tx identifierData type: BITFIELD_32
Index: 11199_d = 2BBF_h

Identifier with which the corresponding SDO client can carry out transmissions.

- If bit 31 is set (0x8nnnnnnn), the corresponding SDO client channel is deactivated (see DS301, V4.02).
- The client channels need not be parameterised right now. Their functionality will only be required when using the gateway services.

► [Identifiers of the parameter data objects](#) (62)

Setting range			Information
0x00000000		0xFFFFFFFF	
Value is bit-coded:			Information
Bit 0	COB-ID bit 0		
...	...		
Bit 31	SDO invalid		
Subcodes	Lenze setting	Information	
C13376/1	0x80000000	SDO client channel 1 TX ... 10 TX	
...			
C13376/10			
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1			

C13377

Parameter Name:			Data type: UNSIGNED_8 Index: 11198 _d = 2BBE _h
C13377 CAN SDO server node address			
Node address of the server with which this client communicates via the client channel selected.			
- An activation of the client functionality is not required. - An entry is required so that the CAN-SDO client channel can be activated (see DS301, V4.02).			
Setting range (min. value unit max. value)			
1		127	
Subcodes	Lenze setting	Information	
C13377/1	1	Remote server node address for SDO client channel 1 ...	
...		10	
C13377/10			
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C13378

Parameter Name:			Data type: UNSIGNED_16 Index: 11197 _d = 2BBD _h
C13378 CAN delay boot-up - Operational			
Time that has to elapse after mains switching before the CAN NMT master transmits the "Start remote node" telegram on the bus.			
This time is only used after mains switching if the master bit is activated (C13352).			
Setting range (min. value unit max. value)			Lenze setting
0	ms	65535	3000 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C13381

Parameter Name:			Data type: UNSIGNED_16 Index: 11194 _d = 2BBA _h
C13381 CAN Heartbeat Producer Time			
Time interval for the transmission of the heartbeat telegram to one or several consumers.			
- The parameterised time is rounded down to an integer multiple of 5 ms. - The heartbeat telegram is transmitted automatically as soon as a time > 0 ms is set. The monitoring function "Node Guarding" is deactivated in this case. - Mapping of the CANopen object I-1017 (see DS301, V4.02).			
▶ Heartbeat protocol (□ 80)			
Setting range (min. value unit max. value)			Lenze setting
0	ms	65535	0 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C13382

Parameter Name:			Data type: UNSIGNED_16 Index: 11193 _d = 2BB9 _h
C13382 CAN Guard Time			
- After the guard time (set here) multiplied with the life time factor (C13383), a node guarding telegram must have arrived. - Mapping of the CANopen object I-100C (see DS301, V4.02).			
▶ Node guarding protocol (□ 74)			
Setting range (min. value unit max. value)			Lenze setting
0	ms	65535	0 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C13383

Parameter | Name:

C13383 | CAN Life Time Factor

Data type: UNSIGNED_8

Index: 11192_d = 2BB8_h

- The life time factor multiplied with the guard time ([C13382](#)) is the time within which a node guarding telegram must have arrived.
 - Mapping of the CANopen object [I-100D](#) (see DS301, V4.02).
- [Node guarding protocol](#) ([80](#))

Setting range (min. value unit max. value)			Lenze setting
0		255	0
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C13385

Parameter | Name:

C13385 | CAN Heartbeat Consumer Time

Data type: BITFIELD_32

Index: 11190_d = 2BB6_h

The 32 subcodes represent the nodes which are to be monitored via heartbeat.

- Each bit-coded subcode entry contains the expected heartbeat time and the node ID (node address) from which the heartbeat telegram is expected.
 - The parameterised time is rounded down to an integer multiple of 5 ms and must have a higher value than the heartbeat producer time ([C13381](#)) of the node to be monitored.
 - The response to a missing heartbeat telegram can be parameterised in [C13613](#).
 - Mapping of the CANopen object [I-1016](#) (see DS301, V4.02).
- [Heartbeat protocol](#) ([80](#))

Setting range		
0x00000000	0xFFFFFFFF	
Value is bit-coded:		
Bit 0	Heartbeat time bit 0	• Bit 0 ... 15: Heartbeat time • Bit 16 ... 23: Node address • Bit 24 ... 31: Reserved
...	...	
Bit 31	Reserved	
Subcodes	Lenze setting	Information
C13385/1	0x00000000	Monitoring entry 1 ... 32
...		
C13385/32		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C13386

Parameter | Name:

Data type: BITFIELD_32
Index: 11189_d = 2BB5_h

C13386 | CAN Node Guarding

These 32 subcodes represent the nodes which are to be monitored by the master via node guarding.

- Each bit-coded subcode entry contains the guard time ([C13382](#)), the life time factor ([C13383](#)) and the node address ([C13350](#)) from which the node guarding telegram is expected.
- The response to a missing node guarding response can be parameterised in [C13612](#).

[Node guarding protocol](#) (74)

Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	Guard time bit 0	- Bit 0 ... 15: Guard time - Bit 16 ... 23: Node address - Bit 24 ... 31: Life time factor
...	...	
Bit 31	Lifetime factor bit 7	
Subcodes	Lenze setting	Information
C13386/1	0x00000000	Monitoring entry 1 ... 32
...		
C13386/32		

☒ Read access

☒ Write access

☐ CINH

☐ PLC STOP

☐ No transfer

Scaling factor: 1

C13387

Parameter | Name:

C13387 | CAN Node Guarding Activity

Data type: BITFIELD_32

Index: 11188_d = 2BB4_h

▶ Node guarding protocol (📖 74)

Display range

0x00000000

0xFFFFFFFF

Value is bit-coded:

Bit 0

Node guarding of node 1

...

...

Bit 31

Node guarding of node 32

☒ Read access
☐ Write access
☐ CINH
☐ PLC STOP
☐ No transfer

Scaling factor: 1

C13388

Parameter Name:		Data type: UNSIGNED_8 Index: 11187 _d = 2BB3 _h	
C13388 CAN node guarding status			
Node guarding status of node 1 ... 32			
▶ Node guarding protocol (📖 74)			
Selection list (read only)			
0 Unknown			
4 Stopped			
5 Operational			
127 Pre-operational			
Subcodes		Information	
C13388/1			
...			
C13388/32			
☒ Read access		Scaling factor: 1	
☐ Write access			
☐ CINH			
☐ PLC STOP			
☐ No transfer			

C13390

Parameter Name:		Data type: BITFIELD_8 Index: 11185 _d = 2BB1 _h	
C13390 CAN Error Register (DS301V402)			
Mapping of the CANopen object I-1001 (see DS301, V4.02)			
Display range			
0x00			0xFF
Value is bit-coded:		Information	
Bit 0	Generic error	Currently only bits 0 and 4 contain the corresponding information.	
Bit 1	Current error (not used)		
Bit 2	Voltage error (not used)		
Bit 3	Temperature error (not used)		
Bit 4	Communication error		
Bit 5	Dev. prof. spec. err (not used)		
Bit 6	Reserved		
Bit 7	Manuf. spec. error (not used)		
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1	

C13391

Parameter | Name: **C13391 | CAN Emergency Object**

Data type: BITFIELD_32
Index: 11184_d = 2BB0_h

Identifier of the emergency telegram

- If bit 31 of this code is set (0x8nnnnnnn), the generation of emergency telegrams is deactivated.
- Mapping of the CANopen object [I-1014](#) (see DS301, V4.02).

[Emergency telegram](#) (📖 84)

Setting range			Lenze setting
0x00000000		0xFFFFFFFF	0x00000081 (decimal: 129)
Value is bit-coded: (☑ = bit set)			Information
Bit 0	☑	COB-ID bit 0	
Bit 1	☐	COB-ID bit 1	
Bit 2	☐	COB-ID bit 2	
Bit 3	☐	COB-ID bit 3	
Bit 4	☐	COB-ID bit 4	
Bit 5	☐	COB-ID bit 5	
Bit 6	☐	COB-ID bit 6	
Bit 7	☑	COB-ID bit 7	
Bit 8	☐	COB-ID bit 8	
Bit 9	☐	COB-ID bit 9	
Bit 10	☐	COB-ID bit 10	
Bit 11	☐	Reserved	
...	...		
Bit 30	☐	Reserved	
Bit 31	☐	Emergency inactive/active	

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Scaling factor: 1

C13392

Parameter | Name:

C13392 | CAN emergency delay time

Data type: UNSIGNED_16
Index: 11183_d = 2BAF_h

Minimum time that has to elapse between two successive emergency telegrams.

- If "0" is set, checking of the inhibit time is deactivated.
- The time is entered in 1/10 ms. The code automatically rounds the entries down to the preceding full millisecond.
- Mapping of the CANopen object [I-1015](#) (see DS301, V4.02).

▶ [Emergency telegram](#) (📖 84)

Setting range (min. value unit max. value)			Lenze setting
0		65535	0
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C13393

Parameter Name:		Data type: UNSIGNED_8 Index: 11182 _d = 2BAE _h	
C13393 CAN result - bus scan			
Result of a CAN bus scan previously carried out (C00002 = 96).			
• The subcode number 1 ... 128 corresponds to the node address 1 ... 128.			
Display range (min. value unit max. value)			
0 1			
Subcodes			
C13393/1		Information Result of the CAN bus scan for node address 1 ... 128 • The value "1" means that a device with the corresponding node address has been found.	
...			
C13393/128			
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1	

C13394

Parameter Name:		Data type: UNSIGNED_32 Index: 24181 _d = 5E75 _h	
C13394 CAN predefined error field (DS301V402)			
Predefined error field according to DS301, V4.02			
Display range (min. value unit max. value)			
0	4294967295		
Subcodes		Information	
C13394/1			
...			
C13394/10			
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1	

C13591

Parameter | Name:

C13591 | Resp. to CAN-RPDOx error

Data type: UNSIGNED_8
Index: 10984_d = 2AE8_h

Response if the corresponding CAN RPDO has not been received within the configured time ([C13357/1...4](#) / [C14357/1...4](#)) or with the configured sync.

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
3	Quick stop by trouble
4	Warning locked
5	Warning
6	Information
0	No response

Subcodes	Lenze setting	Information
C13591/1	0: No response	Response to the absence of RPDO1 ... RPDO4
...		
C13591/4		

☒ Read access

☒ Write access

☐ CINH

☐ PLC STOP

☐ No transfer

Scaling factor: 1

C13595

Parameter Name:		Data type: UNSIGNED_8 Index: 10980 _d = 2AE4 _h	
C13595 Resp. to CAN bus OFF			
Response if CAN node switches to the bus-off state.			
Selection list (Lenze setting printed in bold)			
1	Fault		
2	Trouble		
3	Quick stop by trouble		
4	Warning locked		
5	Warning		
6	Information		
0	No response		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1	

C13612

Parameter Name:		Data type: UNSIGNED_8 Index: 10963 _d = 2AD3 _h
C13612 Resp. to CAN node guarding error		
Response of master if the corresponding node guarding response is missing. ▶ Node guarding protocol (📖 74)		
Selection list (Lenze setting printed in bold)		
1	Fault	
2	Trouble	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
6	Information	
0	No response	
Subcodes	Lenze setting	Information
C13612/1	0: No response	Response to the absence of the node guarding response for monitoring entry 1 ... 32
...		
C13612/32		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C13613

Parameter Name:		Data type: UNSIGNED_8 Index: 10962 _d = 2AD2 _h
C13613 Resp. to CAN heartbeat error		
Response if the corresponding heartbeat telegram is missing. ▶ Heartbeat protocol (📖 80)		
Selection list (Lenze setting printed in bold)		
1	Fault	
2	Trouble	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
6	Information	
0	No response	
Subcodes	Lenze setting	Information
C13613/1	0: No response	Response to the absence of the heartbeat telegram for monitoring entry 1 ... 32
...		
C13613/32		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C13614

Parameter | Name:

C13614 | Resp. to CAN life guarding errorData type: UNSIGNED_8
Index: 10961_d = 2AD1_h

Response of slave if node guarding request is missing.

▶ [Node guarding protocol](#) (□ 74)**Selection list** (Lenze setting printed in bold)

1	Fault
2	Trouble
3	Quick stop by trouble
4	Warning locked
5	Warning
6	Information
0	No response

☒ Read access
 ☒ Write access
 ☐ CINH
 ☐ PLC STOP
 ☐ No transfer
 Scaling factor: 1
C13625

Parameter | Name:

C13625 | CAN behaviour in case of faultData type: UNSIGNED_8
Index: 10950_d = 2AC6_hMapping of the CANopen object [I-1029](#) (see DS301, V4.02)**Selection list** (Lenze setting printed in bold)

0	Pre-operational state
1	No state change
2	Stopped state

☒ Read access
 ☒ Write access
 ☐ CINH
 ☐ PLC STOP
 ☐ No transfer
 Scaling factor: 1

13.3 Parameters of the communication module for slot MXI2

In this chapter, the parameters of the E94AYCCA communication module (CANopen) for slot MXI2 of the Servo Drive 9400 are listed in numerically ascending order.

C14311

Parameter | Name:

C14311 | CAN TPDO1 mask byte x

Data type: BITFIELD_8
Index: 11264_d = 2C00_h

For each byte of the TPDO1 a mask can be parameterised in the corresponding subcode.

- In case of event-controlled transmission of a PDO, only the masked bits are used for the event control.
- Mask "0x0" means that no bit of the corresponding byte triggers the transmission.
- Mask "0xff" means that each bit of the corresponding byte can trigger the transmission.

[Masking of the TPDOs for event control](#) (49)

Setting range	
0x00	0xFF
Value is bit-coded:	
Bit 0	Mask bit 0
...	...
Bit 7	Mask bit 7
Subcodes	Lenze setting
C14311/1	0x00
...	
C14311/8	

Subcodes	Lenze setting	Information
C14311/1	0x00	Mask for byte 1 ... byte 8 of TPDO1
...		
C14311/8		

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C14312

Parameter | Name:

C14312 | CAN TPDO2 mask byte x

Data type: BITFIELD_8
Index: 11263_d = 2BFF_h

For each byte of the TPDO2 a mask can be parameterised in the corresponding subcode.

- In case of event-controlled transmission of a PDO, only the masked bits are used for the event control.
- Mask "0x0" means that no bit of the corresponding byte triggers the transmission.
- Mask "0xff" means that each bit of the corresponding byte can trigger the transmission.

[Masking of the TPDOs for event control](#) (49)

Setting range		
0x00	0xFF	
Value is bit-coded:		
Bit 0	Mask bit 0	
...	...	
Bit 7	Mask bit 7	
Subcodes	Lenze setting	Information
C14312/1	0x00	Mask for byte 1 ... byte 8 of TPDO2
...		
C14312/8		

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1

C14313

Parameter | Name:

C14313 | CAN TPDO3 mask byte x

Data type: BITFIELD_8

Index: 11262_d = 2BFE_h

For each byte of the TPDO3 a mask can be parameterised in the corresponding subcode.

- In case of event-controlled transmission of a PDO, only the masked bits are used for the event control.
- Mask "0x0" means that no bit of the corresponding byte triggers the transmission.
- Mask "0xff" means that each bit of the corresponding byte can trigger the transmission.

▶ [Masking of the TPDOs for event control](#) (49)

Setting range		
0x00		0xFF
Value is bit-coded:		
Bit 0	Mask bit 0	
...	...	
Bit 7	Mask bit 7	
Subcodes	Lenze setting	Information
C14313/1	0x00	Mask for byte 1 ... byte 8 of TPDO3
...		
C14313/8		

☒ Read access
 ☒ Write access
 ☐ CINH
 ☐ PLC STOP
 ☐ No transfer

Scaling factor: 1

C14314

Parameter | Name:

C14314 | CAN TPDO4 mask byte x

Data type: BITFIELD_8
Index: 11261_d = 2BFD_h

For each byte of the TPDO4 a mask can be parameterised in the corresponding subcode.

- In case of event-controlled transmission of a PDO, only the masked bits are used for the event control.
- Mask "0x0" means that no bit of the corresponding byte triggers the transmission.
- Mask "0xff" means that each bit of the corresponding byte can trigger the transmission.

[▶ Masking of the TPDOs for event control](#) (49)

Setting range		
0x00	0xFF	
Value is bit-coded:		
Bit 0	Mask bit 0	
...	...	
Bit 7	Mask bit 7	
Subcodes	Lenze setting	Information
C14314/1	0x00	Mask for byte 1 ... byte 8 of TPDO4
...		
C14314/8		

☒ Read access

☒ Write access

☐ CINH

☐ PLC STOP

☐ No transfer

Scaling factor: 1

C14320

Parameter | Name:

C14320 | CAN TPDOx identifier

Data type: BITFIELD_32
Index: 11255_d = 2BF7_h

Identifier for TPDO1 ... TPDO4

- If bit 31 is set (0x8nnnnnnn), the TPDO is deactivated.
- The basic setting is according to the "Predefined Connection Set".
- Mapping of the CANopen objects [I-1800/1](#) ... [I-1803/1](#) (see DS301, V4.02).

► [Identifiers of the process data objects](#) (46)

Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	• Bit 0 ... 10: COB-ID • Bit 11 ... 30: Reserved • Bit 31: PDO invalid
...	...	
Bit 31	PDO invalid	
Subcodes	Lenze setting	Information
C14320/1	0x00000181	Identifier TPDO1 • After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x180 + node address is set by default.
C14320/2	0x00000281	Identifier TPDO2 • After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x280 + node address is set by default.
C14320/3	0x00000381	Identifier TPDO3 • After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x380 + node address is set by default.
C14320/4	0x00000481	Identifier TPDO4 • After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x480 + node address is set by default.
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C14321

Parameter Name: C14321 CAN RPDOx identifier		Data type: BITFIELD_32 Index: 11254 _d = 2BF6 _h
Identifier for RPDO1 ... RPDO4		
- If bit 31 is set (0x8nnnnnnn), the RPDO is deactivated. - The basic setting is according to the "Predefined Connection Set". - Mapping of the CANopen objects I-1400/1 ... I-1403/1 (see DS301, V4.02).		
▶ Identifiers of the process data objects (46)		
Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	- Bit 0 ... 10: COB-ID - Bit 11 ... 30: Reserved - Bit 31: PDO invalid
...	...	
Bit 31	PDO invalid	
Subcodes	Lenze setting	Information
C14321/1	0x00000201	Identifier RPDO1 After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x200 + node address is set by default.
C14321/2	0x00000301	Identifier RPDO2 After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x300 + node address is set by default.
C14321/3	0x00000401	Identifier RPDO3 After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x400 + node address is set by default.
C14321/4	0x00000501	Identifier RPDO4 After changing the node address and executing "Reset node" (C00002 = 92), here the value 0x500 + node address is set by default.
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14322

Parameter Name: C14322 CAN TPDOx Tx mode		Data type: UNSIGNED_8 Index: 11253 _d = 2BF5 _h
TPDO transmission type according to DS301, V4.02		
- Types 0 (acyclic sync), 1-240 (cyclic sync), 254 (event-controlled manufacturer-specific), 255 (event-controlled device-profile-specific) are supported. - The basic PDO setting is "254" (event-controlled). - Mapping of the CANopen objects I-1800/2 ... I-1803/2 (see DS301, V4.02).		
▶ Transmission type (47)		
Setting range (min. value unit max. value)		
0		255
Subcodes	Lenze setting	Information
C14322/1	254	Transmission mode for TPDO1 ... TPDO4
...		
C14322/4		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14323

Parameter | Name:

C14323 | CAN RPDOx Rx mode

Data type: UNSIGNED_8
Index: 11252_d = 2BF4_h

RPDO transmission type according to DS301, V4.02

- For the RPDO serves as monitoring setting in the case of sync-controlled PDOs.
- Types 0 (acyclic sync), 1-240 (cyclic sync), 254 (event-controlled manufacturer-specific), 255 (event-controlled device-profile-specific) are supported.
- The basic PDO setting is "254" (event-controlled).
- Mapping of the CANopen objects [I-1400/2](#) ... [I-1403/2](#) (see DS301, V4.02).

[Transmission type](#) (📖 47)

Setting range (min. value unit max. value)		
0		255
Subcodes	Lenze setting	Information
C14323/1	254	Transmission mode for RPDO1 ... RPDO4
...		
C14323/4		

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Scaling factor: 1

C14324

Parameter | Name:

C14324 | CAN TPDOx delay time

Data type: UNSIGNED_16
Index: 11251_d = 2BF3_h

TPDO inhibit time according to DS301, V4.02

- Minimum time between the transmission of two identical TPDOs.
- The time is entered in 1/10 ms and automatically rounded to full milliseconds by the code.
- Mapping of the CANopen objects [I-1800/3](#) ... [I-1803/3](#) (see DS301, V4.02).

Setting range (min. value | unit | max. value)

0

1/10 ms

65535

Subcodes

Lenze setting

C14324/1

0 1/10 ms

...

C14324/4

Information

Delay time for TPDO1 ... TPDO4

☒ Read access

☒ Write access

☐ CINH

☐ PLC STOP

☐ No transfer

Scaling factor: 1

C14343

Parameter Name:			Data type: UNSIGNED_32 Index: 11232 _d = 2BE0 _h		
C14343 CAN TPDO counter					
Counter for TPDO1 ... TPDO4					
Display range (min. value unit max. value)			Information		
0		4294967295			
Subcodes					
C14343/1					
...					
C14343/4					
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1		

C14344

Parameter Name: C14344 CAN RPDO counter		Data type: UNSIGNED_32 Index: 11231 _d = 2BDF _h
Counter for RPDO1 ... RPDO4		
Display range (min. value unit max. value)		
0		4294967295
Subcodes		Information
C14344/1		
...		
C14344/4		
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14345

Parameter Name: C14345 CAN error		Data type: UNSIGNED_8 Index: 11230 _d = 2BDE _h
Display of the active CAN network error		
Selection list (read only)		
0	No error	
1	Guard Event	
2	Warning	
3	Bus off	
4	Sync telegram error	
6	CAN controller overflow	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14346

Parameter Name: C14346 CAN heartbeat activity		Data type: BITFIELD_32 Index: 11229 _d = 2BDD _h
▶ Heartbeat protocol (80)		
Display range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		
Bit 0	Heartbeat node 1	
...	...	
Bit 31	Heartbeat node 32	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14347

Parameter Name: C14347 CAN heartbeat status		Data type: UNSIGNED_8 Index: 11228 _d = 2BDC _h
Status of node 1 ... 32 ▶ Heartbeat protocol (□ 80)		
Selection list (read only)		
0	Unknown	
4	Stopped	
5	Operational	
127	Pre-operational	
Subcodes		Information
C14347/1		
...		
C14347/32		
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14348

Parameter Name: C14348 CAN status DIP switch		Data type: UNSIGNED_8 Index: 11227 _d = 2BDB _h
- The value "1" means that the CAN DIP switches have been recognised during power-on and a valid baud rate and node address have been set. - The value "0" means that either no CAN DIP switch or no valid setting has been recognised or that the setting has been overwritten by code C14350 or C14351.		
Selection list (read only)		Information
0	Setting not accepted	
1	Setting accepted	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14349

Parameter Name: C14349 CAN setting of DIP switch		Data type: UNSIGNED_8 Index: 11226 _d = 2BDA _h
CAN DIP switch setting during last power-on ▶ Possible settings via DIP switches (□ 33)		
Display range (min. value unit max. value)		
0		255
Subcodes		Information
C14349/1		Node address
C14349/2		Baud rate: 0: 500 kbps 1: 250 kbps 2: 125 kbps 3: 50 kbps 4: 1 Mbps 5: 20 kbps 6: 10 kbit/s 14: 800 kbps 255: Automatic detection
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14350

Parameter | Name:

C14350 | CAN node addressData type: UNSIGNED_8
Index: 11225_d = 2BD9_h

- A change in the node address is only effective after a "Reset node" (C00002 = 92).
- The basic server channel RX/TX is automatically provided by the node address ([C14372](#) and [C14373](#); both subcode 1).

▶ [Setting the node address](#) ([33](#))

Setting range (min. value unit max. value)			Lenze setting
1		127	1
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C14351

Parameter | Name:

C14351 | CAN baud rateData type: UNSIGNED_8
Index: 11224_d = 2BD8_h

A change in the baud rate is only effective after a "Reset node" (C00002 = 92).

▶ [Setting the baud rate](#) ([34](#))

Selection list (Lenze setting printed in bold)	
0	500 kbps
1	250 kbps
2	125 kbps
3	50 kbps
4	1 Mbps
5	20 kbps
6	10 kbps
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	800 kbps
15	Reserved
255	Automatic detection
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	
Scaling factor: 1	

C14352

Parameter | Name:

C14352 | CAN slave/masterData type: UNSIGNED_8
Index: 11223_d = 2BD7_h

If "1" is entered and saved, the drive will start as a CAN master after mains switching.

Selection list (Lenze setting printed in bold)	
0	Slave
1	Master
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	
Scaling factor: 1	

C14356

Parameter | Name:

C14356 | CAN TPDOx cycle time

Data type: UNSIGNED_16
Index: 11219_d = 2BD3_h

TPDO event time according to DS301, V4.02

- If a different value than "0" is entered, the TPDO is transmitted without further consideration of the transport type after the time set has elapsed.
- Mapping of the CANopen objects [I-1800/5](#) ... [I-1803/5](#) (see DS301, 4.02).

Setting range (min. value unit max. value)		
0	ms	65535
Subcodes	Lenze setting	Information
C14356/1	0 ms	Cycle time for TPDO1 ... TPDO4
...		
C14356/4		

☒ Read access

☒ Write access

☐ CINH

☐ PLC STOP

☐ No transfer

Scaling factor: 1

C14357

Parameter | Name:

C14357 | CAN RPDOx monitoring time

Data type: UNSIGNED_16
Index: 11218_d = 2BD2_h

Mapping of the RPDO event time (see DS301, V4.02)

- If a different value than "0" is entered, the RPDO is expected after the time set has elapsed.
- If the RPDO is not received within the expected time, an error message parameterisable under [C14591/1...4](#) can be triggered.

Setting range (min. value unit max. value)		
0	ms	65535
Subcodes	Lenze setting	Information
C14357/1	3000 ms	Monitoring time for RPDO1 ... RPDO4
...		
C14357/4		

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Scaling factor: 1

C14359

Parameter Name:		Data type: UNSIGNED_8	
C14359 CAN status		Index: 11216 _d = 2BD0 _h	
Display of the current CAN network status			
Selection list (read only)			
0	Operational		
1	Pre-operational		
4	Boot-up		
5	Stopped		
7	Reset		
8	Initialisation		
9	Unknown		
10	Baud rate autom. detected		
☒ Read access		☐ Write access	
☐ CINH		☐ PLC STOP	
☐ No transfer		Scaling factor: 1	

C14360

Parameter | Name:

C14360 | CAN telegram and error counter

Data type: UNSIGNED_16
Index: 11215_d = 2BCF_h

- After mains connection, all counters start with "0".
- After the maximum value has been reached, it is started with "0" again.

Display range (min. value unit max. value)		
0		65535

Subcodes	Information
C14360/1	Stuffing bit error counter More than five identical bits have been detected.
C14360/2	Format error counter CAN frame has not been observed.
C14360/3	Acknowledge error counter No node has confirmed the telegram.
C14360/4	Bit1 error counter "1" should be sent after bus arbitration, but "0" was read.
C14360/5	Bit0 error counter "0" should be sent after bus arbitration, but "1" was read.
C14360/6	CRC error counter Checksum check has indicated an error.
C14360/7	Tx telegram counter Telegrams received without errors.
C14360/8	Rx telegram counter Telegrams sent without errors.

☒ Read access☐ Write access☐ CINH☐ PLC STOP☐ No transfer

Scaling factor: 1

C14361

Parameter | Name:

C14361 | CAN bus load

Data type: UNSIGNED_32
Index: 11214_d = 2BCE_h

The display of the node peak load (subcodes 4 ... 6) can be reset by mains switching or via a "Reset node" command (C00002 = 92).

Display range (min. value unit max. value)		
0	%	100
Subcodes		Information
C14361/1		Current node load in Tx direction
C14361/2		Current node load in Rx direction
C14361/3		Current node load by faulty telegrams
C14361/4		Node peak load in Tx direction
C14361/5		Node peak load in Rx direction
C14361/6		Node peak load by faulty telegrams
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14367

Parameter Name: C14367 CAN SYNC Rx identifier		Data type: UNSIGNED_32 Index: 11208 _d = 2BC8 _h
Identifier with which the sync slave is to receive sync telegrams. Mapping of the CANopen object I-1005 (see DS301, V4.02). Synchronisation of PDOs via sync telegram (65 50)		
Setting range (min. value unit max. value)		Lenze setting
0		2047 128
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14368

Parameter Name: C14368 CAN SYNC Tx identifier		Data type: UNSIGNED_32 Index: 11207 _d = 2BC7 _h
Identifier with which the sync master is to send sync telegrams. Mapping of the CANopen object I-1005 (see DS301, V4.02). Synchronisation of PDOs via sync telegram (65 50)		
Setting range (min. value unit max. value)		Lenze setting
0		2047 128
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14372

Parameter Name: C14372 CAN SDO server Rx identifier		Data type: BITFIELD_32 Index: 11203 _d = 2BC3 _h
Identifier with which the corresponding SDO server can be reached. - If bit 31 is set (0x8nnnnnnn), the corresponding SDO server is deactivated. - Mapping of the CANopen objects I-1200/1 ... I-1209/1 (see DS301, V4.02). Identifiers of the parameter data objects (62 62)		
Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	- Bit 0 ... 10: COB-ID - Bit 11 ... 30: Reserved - Bit 31: SDO invalid
...	...	
Bit 31	SDO invalid	
Subcodes	Lenze setting	Information
C14372/1	0x00000601	SDO server channel 1 RX - Subcode 1 contains the basic SDO channel. According to DS301, V4.02 it can neither be changed nor deactivated. Writing this subcode has no effect. - The value in subcode 1 results from the node address (C14350) + 0x600.
C14372/2	0x80000000	SDO server channel 2 RX
...	...	...
C14372/10	0x80000000	SDO server channel 10 RX
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14373

Parameter Name: C14373 CAN SDO server Tx identifier		Data type: BITFIELD_32 Index: 11202 _d = 2BC2 _h
Identifier with which the corresponding SDO server can carry out transmissions. - If bit 31 is set (0x8nnnnnnn), the corresponding SDO server is deactivated. - Mapping of the CANopen objects I-1200/2 ... I-1209/2 (see DS301, V4.02). Identifiers of the parameter data objects (62)		
Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	- Bit 0 ... 10: COB-ID - Bit 11 ... 30: Reserved - Bit 31: SDO invalid
...	...	
Bit 31	SDO invalid	
Subcodes	Lenze setting	Information
C14373/1	0x00000581	SDO server channel 1 TX - Subcode 1 contains the basic SDO channel. According to DS301, V4.02 it can neither be changed nor deactivated. Writing this subcode has no effect. - The value in subcode 1 results from the node address (C14350) + 0x580.
C14373/2	0x80000000	SDO server channel 2 TX
...	...	...
C14373/10	0x80000000	SDO server channel 10 TX
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14374

Parameter Name: C14374 CAN SDO client node address		Data type: UNSIGNED_8 Index: 11201 _d = 2BC1 _h
Node address of the client assigned to this server (see DS301, V4.02).		
Setting range (min. value unit max. value)		
1		127
Subcodes	Lenze setting	Information
C14374/1	1	SDO server channel 1 remote client node address - Subcode 1 contains the basic SDO channel. According to DS301, V4.02 it does not have this entry. Writing this subcode has no effect. - The value of subindex 1 is 0.
C14374/2	1	SDO server channel 2 remote client node address
...	...	...
C14374/10	1	SDO server channel 10 remote client node address
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14375

Parameter | Name:

C14375 | CAN SDO client Rx identifier

Data type: BITFIELD_32
Index: 11200_d = 2B0C_h

Identifier with which the corresponding SDO client can be reached.

- If bit 31 is set (0x8nnnnnnn), the corresponding SDO client channel is deactivated (see DS301, V4.02).
- The client channels need not be parameterised right now. Their functionality will only be required when using the gateway services.

[Identifiers of the parameter data objects](#) (62)

Setting range		
0x00000000	0xFFFFFFFF	
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	- Bit 0 ... 10: COB-ID - Bit 11 ... 30: Reserved - Bit 31: SDO invalid
...	...	
Bit 31	SDO invalid	
Subcodes	Lenze setting	Information
C14375/1	0x80000000	SDO client channel 1 RX ... 10 RX
...		
C14375/10		

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Scaling factor: 1

C14376

Parameter | Name:

C14376 | CAN SDO client Tx identifier

Data type: BITFIELD_32
Index: 11199_d = 2BBF_h

Identifier with which the corresponding SDO client can carry out transmissions.

- If bit 31 is set (0x8nnnnnnn), the corresponding SDO client channel is deactivated (see DS301, V4.02).
- The client channels need not be parameterised right now. Their functionality will only be required when using the gateway services.

[Identifiers of the parameter data objects](#) (62)

Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	COB-ID bit 0	- Bit 0 ... 10: COB-ID - Bit 11 ... 30: Reserved - Bit 31: SDO invalid
...	...	
Bit 31	SDO invalid	
Subcodes	Lenze setting	Information
C14376/1	0x80000000	SDO client channel 1 TX ... 10 TX
...		
C14376/10		

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ Scaling factor: 1

C14377

Parameter Name:		Data type: UNSIGNED_8 Index: 11198 _d = 2BBE _h	
C14377 CAN SDO server node address			
Node address of the server with which this client communicates via the client channel selected.			
• An activation of the client functionality is not required. • An entry is required so that the CAN-SDO client channel can be activated (see DS301, V4.02).			
Setting range (min. value unit max. value)			
1		127	
Subcodes	Lenze setting	Information	
C14377/1	1	Remote server node address for SDO client channel 1 ...	
...		10	
C14377/10			
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1	

C14378

Parameter | Name:

C14378 | CAN delay boot-up - Operational

Data type: UNSIGNED_16
Index: 11197_d = 2BBD_h

Time that has to elapse after mains switching before the CAN NMT master transmits the "Start remote node" telegram on the bus.

- This time is only used after mains switching if the master bit is activated ([C14352](#)).

Setting range (min. value unit max. value)			Lenze setting
0	ms	65535	3000 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C14381

Parameter | Name:

C14381 | CAN Heartbeat Producer Time

Data type: UNSIGNED_16
Index: 11194_d = 2BBA_h

Time interval for the transmission of the heartbeat telegram to one or several consumers.

- The parameterised time is rounded down to an integer multiple of 5 ms.
- The heartbeat telegram is transmitted automatically as soon as a time > 0 ms is set. The monitoring function "Node Guarding" is deactivated in this case.
- Mapping of the CANopen object [I-1017](#) (see DS301, V4.02).

[Heartbeat protocol](#) [\(80\)](#)

Setting range (min. value unit max. value)			Lenze setting
0	ms	65535	0 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C14382

Parameter | Name:

C14382 | CAN Guard Time

Data type: UNSIGNED_16
Index: 11193_d = 2BB9_h

- After the guard time (set here) multiplied with the life time factor (C14383), a node guarding telegram must have arrived.
- Mapping of the CANopen object I-100C (see DS301, V4.02).

▶ [Node guarding protocol](#) (📖 74)

Setting range (min. value unit max. value)			Lenze setting
0	ms	65535	0 ms
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C14383

Parameter Name: C14383 CAN Life Time Factor			Data type: UNSIGNED_8 Index: 11192 _d = 2BB8 _h
- The life time factor multiplied with the guard time (C14382) is the time within which a node guarding telegram must have arrived. - Mapping of the CANopen object I-100D (see DS301, V4.02). ▶ Node guarding protocol (80)			
Setting range (min. value unit max. value)			Lenze setting
0		255	0
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C14385

Parameter | Name: **C14385 | CAN Heartbeat Consumer Time**

Data type: BITFIELD_32
Index: 11190_d = 2BB6_h

The 32 subcodes represent the nodes which are to be monitored via heartbeat.

- Each bit-coded subcode entry contains the expected heartbeat time and the node ID (node address) from which the heartbeat telegram is expected.
- The parameterised time is rounded down to an integer multiple of 5 ms and must have a higher value than the heartbeat producer time ([C14381](#)) of the node to be monitored.
- The response to a missing heartbeat telegram can be parameterised in [C14613](#).
- Mapping of the CANopen object [I-1016](#) (see DS301, V4.02).

[▶ Heartbeat protocol](#) (80)

Setting range	
0x00000000	0xFFFFFFFF
Value is bit-coded:	
Bit 0	Heartbeat time bit 0
...	...
Bit 31	Reserved
Subcodes	Lenze setting
C14385/1	0x00000000
...	
C14385/32	

Information	
- Bit 0 ... 15: Heartbeat time - Bit 16 ... 23: Node address - Bit 24 ... 31: Reserved	

Information	
Monitoring entry 1 ... 32	

☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer

Scaling factor: 1

C14386

Parameter Name: C14386 CAN Node Guarding		Data type: BITFIELD_32 Index: 11189 _d = 2BB5 _h
These 32 subcodes represent the nodes which are to be monitored by the master via node guarding. - Each bit-coded subcode entry contains the guard time (C14382), the life time factor (C14383), and the node address (C14350) from which the node guarding telegram is expected. - The response to a missing node guarding response can be parameterised in C14612. ► Node guarding protocol (74)		
Setting range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		Information
Bit 0	Guard time bit 0	- Bit 0 ... 15: Guard time - Bit 16 ... 23: Node address - Bit 24 ... 31: Life time factor
...	...	
Bit 31	Lifetime factor bit 7	
Subcodes	Lenze setting	Information
C14386/1	0x00000000	Monitoring entry 1 ... 32
...		
C14386/32		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14387

Parameter Name: C14387 CAN Node Guarding Activity		Data type: BITFIELD_32 Index: 11188 _d = 2BB4 _h
► Node guarding protocol (74)		
Display range		
0x00000000		0xFFFFFFFF
Value is bit-coded:		
Bit 0	Node guarding of node 1	
...	...	
Bit 31	Node guarding of node 32	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14388

Parameter Name: C14388 CAN node guarding status		Data type: UNSIGNED_8 Index: 11187 _d = 2BB3 _h
Node guarding status of node 1 ... 32 ► Node guarding protocol (74)		
Selection list (read only)		
0	Unknown	
4	Stopped	
5	Operational	
127	Pre-operational	
Subcodes	Information	
C14388/1		
...		
C14388/32		
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14390

Parameter Name: C14390 CAN Error Register (DS301V402)		Data type: BITFIELD_8 Index: 11185 _d = 2BB1 _h
Mapping of the CANopen object I-1001 (see DS301, V4.02)		
Display range		
0x00		0xFF
Value is bit-coded:		Information
Bit 0	Generic error	Currently only bits 0 and 4 contain the corresponding information.
Bit 1	Current error (not used)	
Bit 2	Voltage error (not used)	
Bit 3	Temperature error (not used)	
Bit 4	Communication error	
Bit 5	Dev. prof. spec. err (not used)	
Bit 6	Reserved	
Bit 7	Manuf. spec. error (not used)	
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14391

Parameter Name: C14391 CAN Emergency Object		Data type: BITFIELD_32 Index: 11184 _d = 2BB0 _h
Identifier of the emergency telegram		
- If bit 31 of this code is set (0x8nnnnnnn), the generation of emergency telegrams is deactivated. - Mapping of the CANopen object I-1014 (see DS301, V4.02).		
▶ Emergency telegram (84)		
Setting range		Lenze setting
0x00000000	0xFFFFFFFF	0x00000081 (decimal: 129)
Value is bit-coded: (☒ = bit set)		Information
Bit 0 ☒	COB-ID bit 0	
Bit 1 ☐	COB-ID bit 1	
Bit 2 ☐	COB-ID bit 2	
Bit 3 ☐	COB-ID bit 3	
Bit 4 ☐	COB-ID bit 4	
Bit 5 ☐	COB-ID bit 5	
Bit 6 ☐	COB-ID bit 6	
Bit 7 ☒	COB-ID bit 7	
Bit 8 ☐	COB-ID bit 8	
Bit 9 ☐	COB-ID bit 9	
Bit 10 ☐	COB-ID bit 10	
Bit 11 ☐	Reserved	
...	...	
Bit 30 ☐	Reserved	
Bit 31 ☐	Emergency inactive/active	
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14392

Parameter | Name:

C14392 | CAN emergency delay timeData type: UNSIGNED_16
Index: 11183_d = 2BAF_h

Minimum time that has to elapse between two successive emergency telegrams.

- If "0" is set, checking of the inhibit time is deactivated.
- The time is entered in 1/10 ms. The code automatically rounds the entries down to the preceding full millisecond.
- Mapping of the CANopen object [I-1015](#) (see DS301, V4.02).

▶ [Emergency telegram](#) (📘 84)

Setting range (min. value unit max. value)			Lenze setting
0		65535	0
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C14393

Parameter | Name:

C14393 | CAN result - bus scanData type: UNSIGNED_8
Index: 11182_d = 2BAE_h

Result of a CAN bus scan previously carried out (C00002 = 96).

- The subcode number 1 ... 128 corresponds to the node address 1 ... 128.

Display range (min. value unit max. value)			Information Result of the CAN bus scan for node address 1 ... 128 • The value "1" means that a device with the corresponding node address has been found.
0		1	
Subcodes			
C14393/1			
...			
C14393/128			
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C14394

Parameter | Name:

C14394 | CAN predefined error field (DS301V402)Data type: UNSIGNED_32
Index: 24181_d = 5E75_h

Predefined error field according to DS301, V4.02

Display range (min. value unit max. value)			
0		4294967295	
Subcodes			Information
C14394/1			
...			
C14394/10			
☒ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer			Scaling factor: 1

C14591

Parameter | Name:

C14591 | Resp. to CAN-RPDOx error

Data type: UNSIGNED_8
Index: 10984_d = 2AE8_h

Response if the corresponding CAN RPDO has not been received within the configured time ([C13357/1...4](#) / [C14357/1...4](#)) or with the configured sync.

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
3	Quick stop by trouble
4	Warning locked
5	Warning
6	Information
0	No response

Subcodes	Lenze setting	Information
C14591/1	0: No response	Response to the absence of RPDO1 ... RPDO4
...		
C14591/4		

☒ Read access

☒ Write access

☐ CINH

☐ PLC STOP

☐ No transfer

Scaling factor: 1

C14595

Parameter Name:		Data type: UNSIGNED_8 Index: 10980 _d = 2AE4 _h	
C14595 Resp. to CAN bus OFF			
Response if CAN node switches to the bus-off state.			
Selection list (Lenze setting printed in bold)			
1	Fault		
2	Trouble		
3	Quick stop by trouble		
4	Warning locked		
5	Warning		
6	Information		
0	No response		
☒ Read access		☒ Write access	
☐ CINH		☐ PLC STOP	
☐ No transfer		Scaling factor: 1	

C14612

Parameter | Name:

C14612 | Resp. to CAN node guarding errorData type: UNSIGNED_8
Index: 10963_d = 2AD3_h

Response of master if the corresponding node guarding response is missing.

▶ [Node guarding protocol](#) (📖 74)

Selection list (Lenze setting printed in bold)		
1	Fault	
2	Trouble	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
6	Information	
0	No response	
Subcodes	Lenze setting	Information
C14612/1	0: No response	Response to the absence of the node guarding response for monitoring entry 1 ... 32
...		
C14612/32		
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer Scaling factor: 1		

C14613

Parameter | Name:

C14613 | Resp. to CAN heartbeat errorData type: UNSIGNED_8
Index: 10962_d = 2AD2_h

Response if the corresponding heartbeat telegram is missing.

▶ [Heartbeat protocol](#) (📖 80)

Selection list (Lenze setting printed in bold)	
1	Fault
2	Trouble
3	Quick stop by trouble
4	Warning locked
5	Warning
6	Information
0	No response
Subcodes	Lenze setting
C14613/1	0: No response
...	
C14613/32	

Information	
Response to the absence of the heartbeat telegram for monitoring entry 1 ... 32	

☒ Read access

☒ Write access

☐ CINH

☐ PLC STOP

☐ No transfer

Scaling factor: 1

C14614

Parameter Name:		Data type: UNSIGNED_8 Index: 10961 _d = 2AD1 _h
C14614 Resp. to CAN life guarding error		
Response of slave if node guarding request is missing. ▶ Node guarding protocol (□ 74)		
Selection list (Lenze setting printed in bold)		
1	Fault	
2	Trouble	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
6	Information	
0	No response	
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

C14625

Parameter Name:		Data type: UNSIGNED_8 Index: 10950 _d = 2AC6 _h
C14625 CAN behaviour in case of fault		
Mapping of the CANopen object I-1029 (see DS301, V4.02)		
Selection list (Lenze setting printed in bold)		
0	Pre-operational state	
1	No state change	
2	Stopped state	
☒ Read access ☒ Write access ☐ CINH ☐ PLC STOP ☐ No transfer		Scaling factor: 1

13.4 Table of attributes

The table of attributes contains information required for communicating with the controller via parameters.

How to read the table of attributes:

Column		Meaning	Entry	
Code		Parameter designation	Cxxxxx	
Name		Parameter short text (display text)	Text	
Index	dec	Index by which the parameter is addressed. The subindex of array variables corresponds to the Lenze subcode number.	24575 - Lenze code number	Only required for access via 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
			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 possible in the case of controller inhibit (CINH) only	

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C13311	CAN TPDO1 mask byte x	24264	5EC8	A	8	BITFIELD_8	1	☑	☑	
C13312	CAN TPDO2 mask byte x	24263	5EC7	A	8	BITFIELD_8	1	☑	☑	
C13313	CAN TPDO3 mask byte x	24262	5EC6	A	8	BITFIELD_8	1	☑	☑	
C13314	CAN TPDO4 mask byte x	24261	5EC5	A	8	BITFIELD_8	1	☑	☑	
C13320	CAN TPDOx identifier	24255	5EBF	A	4	BITFIELD_32	1	☑	☑	
C13321	CAN RPDOx identifier	24254	5EBE	A	4	BITFIELD_32	1	☑	☑	
C13322	CAN TPDOx Tx mode	24253	5EBD	A	4	UNSIGNED_8	1	☑	☑	
C13323	CAN RPDOx Rx mode	24252	5EBC	A	4	UNSIGNED_8	1	☑	☑	
C13324	CAN TPDOx delay time	24251	5EBB	A	4	UNSIGNED_16	1	☑	☑	
C13343	CAN TPDO counter	24232	5EA8	A	4	UNSIGNED_32	1	☑		
C13344	CAN RPDO counter	24231	5EA7	A	4	UNSIGNED_32	1	☑		
C13345	CAN error	24230	5EA6	E	1	UNSIGNED_8	1	☑		
C13346	CAN heartbeat activity	24229	5EA5	E	1	BITFIELD_32	1	☑		
C13347	CAN heartbeat status	24228	5EA4	A	32	UNSIGNED_8	1	☑		
C13348	CAN status DIP switch	24227	5EA3	E	1	UNSIGNED_8	1	☑		
C13349	CAN setting of DIP switch	24226	5EA2	A	2	UNSIGNED_8	1	☑		
C13350	CAN node address	24225	5EA1	E	1	UNSIGNED_8	1	☑	☑	
C13351	CAN baud rate	24224	5EA0	E	1	UNSIGNED_8	1	☑	☑	
C13352	CAN slave/master	24223	5E9F	E	1	UNSIGNED_8	1	☑	☑	
C13356	CAN TPDOx cycle time	24219	5E9B	A	4	UNSIGNED_16	1	☑	☑	
C13357	CAN RPDOx monitoring time	24218	5E9A	A	4	UNSIGNED_16	1	☑	☑	
C13359	CAN status	24216	5E98	E	1	UNSIGNED_8	1	☑		
C13360	CAN telegram and error counter	24215	5E97	A	8	UNSIGNED_16	1	☑		
C13361	CAN bus load	24214	5E96	A	6	UNSIGNED_32	1	☑		
C13367	CAN SYNC Rx identifier	24208	5E90	E	1	UNSIGNED_32	1	☑	☑	
C13368	CAN SYNC Tx identifier	24207	5E8F	E	1	UNSIGNED_32	1	☑	☑	
C13372	CAN SDO server Rx identifier	24203	5E8B	A	10	BITFIELD_32	1	☑	☑	
C13373	CAN SDO server Tx identifier	24202	5E8A	A	10	BITFIELD_32	1	☑	☑	
C13374	CAN SDO client node address	24201	5E89	A	10	UNSIGNED_8	1	☑	☑	
C13375	CAN SDO client Rx identifier	24200	5E88	A	10	BITFIELD_32	1	☑	☑	
C13376	CAN SDO client Tx identifier	24199	5E87	A	10	BITFIELD_32	1	☑	☑	
C13377	CAN SDO server node address	24198	5E86	A	10	UNSIGNED_8	1	☑	☑	
C13378	CAN delay boot-up - Operational	24197	5E85	E	1	UNSIGNED_16	1	☑	☑	
C13381	CAN heartbeat producer time	24194	5E82	E	1	UNSIGNED_16	1	☑	☑	
C13382	CAN guard time	24193	5E81	E	1	UNSIGNED_16	1	☑	☑	
C13383	CAN life time factor	24192	5E80	E	1	UNSIGNED_8	1	☑	☑	
C13385	CAN heartbeat consumer time	24190	5E7E	A	32	BITFIELD_32	1	☑	☑	
C13386	CAN node guarding	24189	5E7D	A	32	BITFIELD_32	1	☑	☑	
C13387	CAN node guarding activity	24188	5E7C	E	1	BITFIELD_32	1	☑		
C13388	CAN node guarding status	24187	5E7B	A	32	UNSIGNED_8	1	☑		
C13390	CAN Error Register (DS301V402)	24185	5E79	E	1	BITFIELD_8	1	☑		
C13391	CAN emergency object	24184	5E78	E	1	BITFIELD_32	1	☑	☑	
C13392	CAN emergency delay time	24183	5E77	E	1	UNSIGNED_16	1	☑	☑	
C13393	CAN bus scan result	24182	5E76	A	128	UNSIGNED_8	1	☑		
C13394	CAN predefined error field (DS301V402)	24181	5E75	A	10	UNSIGNED_32	1	☑		
C13591	Resp. CAN RPDOx error	23984	5DB0	A	4	UNSIGNED_8	1	☑	☑	
C13595	Resp. to CAN bus OFF	23980	5DAC	E	1	UNSIGNED_8	1	☑	☑	
C13612	Resp. CAN node guarding error	23963	5D9B	A	32	UNSIGNED_8	1	☑	☑	

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C13613	Resp. to CAN heartbeat error	23962	5D9A	A	32	UNSIGNED_8	1	☒	☒	
C13614	Resp. to CAN life guarding error	23961	5D99	E	1	UNSIGNED_8	1	☒	☒	
C13625	CAN behaviour in the case of error	23950	5D8E	E	1	UNSIGNED_8	1	☒	☒	
C14311	CAN TPDO1 mask byte x	24264	5EC8	A	8	BITFIELD_8	1	☒	☒	
C14312	CAN TPDO2 mask byte x	24263	5EC7	A	8	BITFIELD_8	1	☒	☒	
C14313	CAN TPDO3 mask byte x	24262	5EC6	A	8	BITFIELD_8	1	☒	☒	
C14314	CAN TPDO4 mask byte x	24261	5EC5	A	8	BITFIELD_8	1	☒	☒	
C14320	CAN TPDOx identifier	24255	5EBF	A	4	BITFIELD_32	1	☒	☒	
C14321	CAN RPDOx identifier	24254	5EBE	A	4	BITFIELD_32	1	☒	☒	
C14322	CAN TPDOx Tx mode	24253	5EBD	A	4	UNSIGNED_8	1	☒	☒	
C14323	CAN RPDOx Rx mode	24252	5EBC	A	4	UNSIGNED_8	1	☒	☒	
C14324	CAN TPDOx delay time	24251	5EBB	A	4	UNSIGNED_16	1	☒	☒	
C14343	CAN TPDO counter	24232	5EA8	A	4	UNSIGNED_32	1	☒		
C14344	CAN RPDO counter	24231	5EA7	A	4	UNSIGNED_32	1	☒		
C14345	CAN error	24230	5EA6	E	1	UNSIGNED_8	1	☒		
C14346	CAN heartbeat activity	24229	5EA5	E	1	BITFIELD_32	1	☒		
C14347	CAN heartbeat status	24228	5EA4	A	32	UNSIGNED_8	1	☒		
C14348	CAN status DIP switch	24227	5EA3	E	1	UNSIGNED_8	1	☒		
C14349	CAN setting of DIP switch	24226	5EA2	A	2	UNSIGNED_8	1	☒		
C14350	CAN node address	24225	5EA1	E	1	UNSIGNED_8	1	☒	☒	
C14351	CAN baud rate	24224	5EA0	E	1	UNSIGNED_8	1	☒	☒	
C14352	CAN slave/master	24223	5E9F	E	1	UNSIGNED_8	1	☒	☒	
C14356	CAN TPDOx cycle time	24219	5E9B	A	4	UNSIGNED_16	1	☒	☒	
C14357	CAN RPDOx monitoring time	24218	5E9A	A	4	UNSIGNED_16	1	☒	☒	
C14359	CAN status	24216	5E98	E	1	UNSIGNED_8	1	☒		
C14360	CAN telegram and error counter	24215	5E97	A	8	UNSIGNED_16	1	☒		
C14361	CAN bus load	24214	5E96	A	6	UNSIGNED_32	1	☒		
C14367	CAN SYNC Rx identifier	24208	5E90	E	1	UNSIGNED_32	1	☒	☒	
C14368	CAN SYNC Tx identifier	24207	5E8F	E	1	UNSIGNED_32	1	☒	☒	
C14372	CAN SDO server Rx identifier	24203	5E8B	A	10	BITFIELD_32	1	☒	☒	
C14373	CAN SDO server Tx identifier	24202	5E8A	A	10	BITFIELD_32	1	☒	☒	
C14374	CAN SDO client node address	24201	5E89	A	10	UNSIGNED_8	1	☒	☒	
C14375	CAN SDO client Rx identifier	24200	5E88	A	10	BITFIELD_32	1	☒	☒	
C14376	CAN SDO client Tx identifier	24199	5E87	A	10	BITFIELD_32	1	☒	☒	
C14377	CAN SDO server node address	24198	5E86	A	10	UNSIGNED_8	1	☒	☒	
C14378	CAN delay boot-up - Operational	24197	5E85	E	1	UNSIGNED_16	1	☒	☒	
C14381	CAN heartbeat producer time	24194	5E82	E	1	UNSIGNED_16	1	☒	☒	
C14382	CAN guard time	24193	5E81	E	1	UNSIGNED_16	1	☒	☒	
C14383	CAN life time factor	24192	5E80	E	1	UNSIGNED_8	1	☒	☒	
C14385	CAN heartbeat consumer time	24190	5E7E	A	32	BITFIELD_32	1	☒	☒	
C14386	CAN node guarding	24189	5E7D	A	32	BITFIELD_32	1	☒	☒	
C14387	CAN node guarding activity	24188	5E7C	E	1	BITFIELD_32	1	☒		
C14388	CAN node guarding status	24187	5E7B	A	32	UNSIGNED_8	1	☒		
C14390	CAN Error Register (DS301V402)	24185	5E79	E	1	BITFIELD_8	1	☒		
C14391	CAN emergency object	24184	5E78	E	1	BITFIELD_32	1	☒	☒	
C14392	CAN emergency delay time	24183	5E77	E	1	UNSIGNED_16	1	☒	☒	
C14393	CAN bus scan result	24182	5E76	A	128	UNSIGNED_8	1	☒		
C14394	CAN predefined error field (DS301V402)	24181	5E75	A	10	UNSIGNED_32	1	☒		
C14591	Resp. CAN RPDOx error	23984	5DB0	A	4	UNSIGNED_8	1	☒	☒	
C14595	Resp. to CAN bus OFF	23980	5DAC	E	1	UNSIGNED_8	1	☒	☒	

E94AYCCA communication manual (CANopen®)

Parameter reference

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C14612	Resp. CAN node guarding error	23963	5D9B	A	32	UNSIGNED_8	1	☒	☒	
C14613	Resp. to CAN heartbeat error	23962	5D9A	A	32	UNSIGNED_8	1	☒	☒	
C14614	Resp. to CAN life guarding error	23961	5D99	E	1	UNSIGNED_8	1	☒	☒	
C14625	CAN behaviour in the case of error	23950	5D8E	E	1	UNSIGNED_8	1	☒	☒	

14 Implemented CANopen objects

Lenze devices can be parameterised with Lenze codes and with manufacturer-independent "CANopen objects". For a completely CANopen-conform communication only the CANopen objects may be used for parameter setting. The CANopen objects described in this chapter are defined in the CANopen specification DS301, V4.02.

Many CANopen objects can be mapped to Lenze codes. In the following table, the related Lenze codes are listed in the column "Relationship to Lenze code".



Note!

Some of the terms used here derive from the CANopen protocol.

Overview of the CANopen indexes and their relationship to Lenze codes

CANopen object			Relationship to Lenze code
Index	Subindex	Name	
I-1000	0	Device type	-
I-1001	0	Error register	C13390 / C14390
I-1003	Pre-defined error field		
	0	Number of errors	-
	1 ... 10	Standard error field	-
I-1005	0	COB-ID SYNC message	C13367 / C14367 C13368 / C13368
I-1006	0	Communication cycle period	C00369
I-100C	0	Guard time	C13382 / C14382
I-100D	0	Life time factor	C13383 / C14383
I-1010	Store parameters		
	0	Highest subindex supported	-
	1	Save all parameters	-
I-1011	Restore default parameters		
	0	Highest subindex supported	-
	1	restore all default parameters	-
I-1014	0	COB-ID EMCY	C13391 / C14391
I-1015	0	Inhibit time EMCY	C13392 / C14392
I-1016	Consumer heartbeat time		
	0	Highest subindex supported	-
	1 ... 32	Consumer heartbeat time	C13385/1...32 / C14385/1...32
I-1017	0	Producer heartbeat time	C13381 / C14381
I-1018	Identity object		
	0	Highest subindex supported	-
	1	Vendor ID	-
	2	Product code	-
	3	Revision number	-
	4	Serial number	-

CANopen object			Relationship to Lenze code
Index	Subindex	Name	
I-1029	Error behaviour		
	0	Highest subindex supported	-
	1	Communication error	C13625 / C14625
I-1200	SDO1 server parameter		
	0	Highest subindex supported	-
	1	COB-ID client → server (rx)	C13372/1 / C14372/1
	2	COB-ID server → client (tx)	C13373/1 / C14373/1
I-1201 ... I-1209	SDO2 ... SDO10 server parameter		
	0	Highest subindex supported	-
	1	COB-ID client → server (rx)	C13372/2...10 / C14372/2...10
	2	COB-ID server → client (tx)	C13373/2...10 / C14373/2...10
	3	Node-ID of the SDO client	-
I-1400 ... I-1403	RPDO1 ... RPDO4 communication parameter		
	0	Highest subindex supported	-
	1	COB-ID used by RPDO	C13321/1...4 / C14321/1...4
	2	Transmission type	C13323/1...4 / C14323/1...4
	3	Inhibit time	-
	4	Compatibility entry	-
	5	Event timer	-
I-1600 ... I-1603	RPDO1 ... RPDO4 mapping parameter		
	0	Number of mapped application objects in PDO	-
	1 ... 8	Application object 1 ... 8	-
I-1800 ... I-1803	TPDO1 ... TDDO4 communication parameter		
	0	Highest subindex supported	-
	1	COB-ID used by TPDO	C13320/1...4 / C14320/1...4
	2	Transmission type	C13322/1...4 / C14322/1...4
	3	Inhibit time	C13324/1...4 / C14324/1...4
	4	Reserved	-
	5	Event timer	C13356/1...4 / C14356/1...4
I-1A00 ... I-1A03	TPDO1 ... TDDO4 mapping parameter		
	0	Number of mapped application objects in PDO	-
	1 ... 8	Application object 1 ... 8	-

I-1000 - Device type

Index I-1000	Name: Device type				
Subindex	Default setting	Display range (min. value unit max. value)		Access	Data type
0: Device type	0	0		4294967295	ro U32

This object specifies the profile for this device. Furthermore, additional information defined in the device profile itself can be stored here.

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
High word		Low word	
High byte	Low byte	High byte	Low byte
Additional information		Device profile number	

[14-1] Data telegram assignment

I-1001 - Error register

Index: I-1001	Name: Error register				
Subindex	Default setting	Display range (min. value unit max. value)		Access	Data type
0: Error register	-	0		255	ro U8

Error register

- This object is related to the Lenze code [C13390](#) / [C14390](#).
- The error status in the data byte (U8) is bit coded (see the following table). Currently only bit 0 and bit 4 in the data byte contain the corresponding information.

Bit	Meaning if bit is set:
Bit 0	Generic error
Bit 1	Current error (not used)
Bit 2	Voltage error (not used)
Bit 3	Temperature error (not used)
Bit 4	Communication error
Bit 5	Device profile spec. error (not used)
Bit 6	Reserved
Bit 7	Manufacturer-specific error (not used)

I-1003 - Pre-defined error field

Index: I-1003	Name: Pre-defined error field					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Number of errors	0	0		255	rw	U8
1 ... 10: Standard error field	-	0		4294967295	ro	U32

Error history

This object indicates that an error has occurred in the communication module and in the standard device.

Subindex	Meaning
0	Number of the error messages stored
1 ... 10	Display of the error list The error messages (U32) consist of a 16-bit error code and a 16-bit manufacturer-specific information field.



Note!

The values in the "standard error field" under subindexes 1 ... 10 will be deleted if the subindex "number of recorded errors" is overwritten with the value "0".

Emergency error code	Cause	Entry in the error register (I-1001)
0x0000	One of several errors eliminated	0xxx
	Elimination of an individual error (afterwards error-free state)	0x00
0x1000	Standard device is in error status (error response "Trouble", "Message", "Warning", "Fault", "Quick stop by trouble" or "System fault")	0x01
0x3100	Supply voltage of standard device faulty or failed	0x01
0x8100	Communication error (warning)	0x11
0x8130	Life guard error or heartbeat error	0x11
0x8150	Collision of COB-IDs: an ID parameterised for reception also is used for transmission.	0x11
0x8210	PDO length is shorter than the expected PDO length	0x11
0x8220	PDO length is longer than the expected PDO length	0x11
0x8700	Sync telegram monitoring	0x11

I-1005 - COB-ID SYNC message

Index: I-1005	Name: COB-ID SYNC message					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: COB-ID SYNC message	0x0000 0080 or 0x8000 0080	0		4294967295	rw	U32

This object serves to activate the creation of sync telegrams and write the value of the identifier.

- This object is related to the Lenze codes [C13367](#) / [C14367](#) and [C13368](#) / [C14368](#).

Creating sync telegrams

To create sync telegrams, bit 30 (see below) must be set to "1". The interval of the sync telegrams can be set with the object [I-1006](#).

Writing identifiers

For the reception of PDOs, the value 0x80 is entered in the 11-bit identifier in the Lenze setting (and according to DS301, V4.02). This means that all modules are by default set to the same sync telegram.

- If sync telegrams are only to be received by specific communication modules, their identifiers can be entered with a value of up to and including 0x07FF.
- The identifier may only be changed if the communication module does not transmit a sync telegram (bit 30 = "0").
- How to change the identifier:
 - Deactivate identifier (set bit 30 to "0").
 - Change identifier.
 - Activate identifier (set bit 30 to "1").

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
x	0/1	Extended identifier*				11-bit identifier	

* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".

* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".

[14-2] Data telegram assignment

I-1006 - Communication cycle period

Index: I-1006	Name: Communication cycle period					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Communication cycle period	0 µs	0	µs	65535000	rw	U32

Setting the sync telegram cycle time.

- ▶ A cycle time can be defined with the entry "1000" or an integer multiple of it.
- ▶ With "0 µs" (Lenze setting), no sync telegrams are created.
- ▶ This object is related to the Lenze code [C00369](#).

I-100C - Guard time

Index: I-100C	Name: Guard time					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Guard time	0 ms	0	ms	65535	rw	U16

Monitoring time for node guarding. ▶ [Node guarding protocol](#) (74)

- ▶ Time within the NMT slave expects the RTRs from the NMTmaster.
- ▶ The node life time is the product of the guard time and the life time factor:
Node Life Time = Guard Time (I-100C) x Life Time Factor (I-100D)
- ▶ The "life guarding event" occurs in the NMT slave if the slave has not been triggered by the NMT master through an RTR within the node life time.
- ▶ With "0 ms" (Lenze setting), monitoring is not supported by the slave.
- ▶ This object is related to the Lenze code [C13382](#) / [C14382](#).

I-100D

Index: I-100D	Name: Life time factor					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Life time factor	0	0		255	rw	U8

Life Time Factor for node guarding. ▶ [Node guarding protocol](#) (74)

- ▶ The node life time is the product of the guard time and the life time factor:
Node Life Time = Guard Time (I-100C) x Life Time Factor (I-100D)
- ▶ The "life guarding event" occurs in the NMT slave if the slave has not been triggered by the NMT master through an RTR within the node life time.
- ▶ With "0" (Lenze setting), monitoring is not supported by the slave.
- ▶ This object is related to the Lenze code [C13383](#) / [C14383](#).

I-1010 - Store parameters

Index: I-1010	Name: Store parameters					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	1	- (only read access)			ro	U32
1: Save all parameters	-	0		4294967295	rw	U32

Save parameters with mains failure protection.

- Corresponds to the device command **C00002** = "**11: Save start parameters**".
- This command serves to save the current parameter settings of the active application with mains failure protection in the memory module of the controller.

Subindex	Meaning	
	Read	Write
0	Max. supported subindex: 1	- (A write attempt triggers the error message 0x06010002.)
1	Reading memory functions of all parameters.	Save parameters with mains failure protection.

Read subindex 1

8th byte	7th byte	6th byte	5th byte	
Data 4	Data 3	Data 2	Data 1	
Bit 31 ... bit 2			Bit 1	Bit 0
0			0/1	0/1

[14-3] Assignment of the data telegram (read access)

Bit	Meaning
Bit 0	0 No saving of parameters on command.
	1 Saving of parameters on command (Lenze).
Bit 1	0 No automatic saving of parameters (Lenze).
	1 Automatic saving of parameters.

Write subindex 1

In addition to the index and subindex, the telegram data must also include the "save" signature (ASCII characters; ISO 8859) so that the parameters are stored.

- Writing with a wrong identifier causes a response according to CANopen specification DS301, V4.02.

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
"e" = 0x65	"v" = 0x76	"a" = 0x61	"s" = 0x73

[14-4] Assignment of the data telegram (write access)

Index: I-1011	Name: Restore default parameters					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	1	- (only read access)			ro	U32
1: restore all default parameters	-	0		4294967295	rw	U32

- ▶ Corresponds to the device command **C00002 = "0: Load Lenze setting"**.
- ▶ This command serves to reset the parameters of the active application to the Lenze setting which is stored in the firmware.

Read subindex 1

[14-5] Assignment of the data telegram (read)

Write subindex 1

- ▶ Writing with a wrong identifier causes a response according to the CANopen specification DS301, V4.02.

[14-6] Assignment of the data telegram (write)

I-1014 - COB-ID EMCY

Index: I-1014	Name: COB-ID EMCY				
Subindex	Default setting	Setting range (min. value unit max. value)		Access	Data type
0: COB-ID EMCY	0x80+Node-ID	0		4294967295 rw	U32

If a communication error or an internal error of the communication module or the controller occurs or is acknowledged (e. g. "trouble"), an error message is sent via the system bus (CANopen). For each error, the telegram is transmitted once. This function can be activated or deactivated by means of bit 31.

► This object is related to the Lenze code [C13391](#) / [C14391](#).

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0/1	0	Extended identifier*				11-bit identifier	

* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".

[14-7] Data telegram assignment

Bit	Setting
Bit 31	0 Emergency object valid.
	1 Emergency object invalid.



Note!

The identifier can only be changed in the "Emergency object invalid" status (bit 31 = 1).

I-1015 - Inhibit time EMCY

Index: I-1015	Name: Inhibit time EMCY				
Subindex	Default setting	Setting range (min. value unit max. value)		Access	Data type
0: inhibit time EMCY	0	0	0.1 ms	65535 rw	U32

Time which must elapse after an error message ([I-1014](#)) has been transmitted, before further error messages can be sent via the system bus (CANopen).

- The entered value multiplied by "0.1" gives the delay time in [ms]. The values are automatically rounded up to whole values in [ms].
- This object is related to the Lenze code [C13392](#) / [C14392](#).

I-1016 - Consumer heartbeat time

Index: I-1016	Name: Consumer heartbeat time					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	32	- (only read access)			ro	U32
1 ... 32: consumer heartbeat time	0	0		4294967295	rw	U32

Monitoring time for the nodes 1 ... 32 to be monitored via heartbeat. ▶ [Node guarding protocol](#) (74)

- ▶ The parameterised time is rounded down to an integer multiple of 5 ms and must have a higher value than the heartbeat producer time of the node to be monitored.

Subindex	Meaning	Lenze code
0	Number of nodes to be monitored	
1 ... 32	Node-ID and heartbeat time of the node 1 ... 32 to be monitored	C13385/1...32 C14385/1...32

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
Bit 31 ... bit 24	Bit 23 ... bit 16	Bit 15 ... bit 0	
0 (reserved)	Node ID	Heartbeat time in [ms]	

[14-8] Data telegram assignment

I-1017 - Producer heartbeat time

Index: I-1017	Name: Producer heartbeat time					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: Producer heartbeat time	0	0	ms	65535	rw	U32

Time interval for the transmission of the heartbeat telegram to one or several consumers. ▶ [Node guarding protocol](#) (74)

- ▶ The parameterised time is rounded down to an integer multiple of 5 ms.
- ▶ The heartbeat telegram is transmitted automatically as soon as a time > 0 ms is set. In this case, the node guarding monitoring function is deactivated.
- ▶ This object is related to the Lenze code [C13381](#) / [C14381](#).

I-1018 - Identity object

Index:
I-1018

Name:
Identity object

Subindex	Default setting	Display range (min. value unit max. value)			Access	Data type
0: highest subindex supported	see below	0		4294967295	ro	U32
1: Vendor ID						
2: Product code						
3: Revision number						
4: Serial number						

Subindex	Meaning						
1	Manufacturer identification number The identification number allocated to Lenze by the "CAN in Automation e. V." organisation is "0x0000003B".						
2	Product code <table> <tr> <td>0x94001</td> <td>9400 StateLine</td> </tr> <tr> <td>0x94002</td> <td>9400 HighLine / ServoPLC</td> </tr> <tr> <td>0x94004</td> <td>9400 regenerative power supply module</td> </tr> </table>	0x94001	9400 StateLine	0x94002	9400 HighLine / ServoPLC	0x94004	9400 regenerative power supply module
0x94001	9400 StateLine						
0x94002	9400 HighLine / ServoPLC						
0x94004	9400 regenerative power supply module						
3	Main version and subversion of the firmware						
4	Serial number						

I-1029 - Error behaviour

Index: I-1029	Name: Error behaviour					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	1	- (only read access)			ro	U8
1: Communication error	0	0		2	rw	U8

This object serves to set the communication status to which the controller is to change after a bus-off, a node/life guarding event or a heartbeat event.

Subindex	Meaning	Lenze code
1	Status change after bus off, node/life guarding event or heartbeat event:	C13625 / C14625
	0 State change from "Operational" to "Pre-operational"	
	1 No state change	
	2 State change to "Stopped"	

I-1200 - SDO1 server parameter

Index: I-1200	Name: SDO1 server parameter					
Subindex	Default setting	Display range (min. value unit max. value)			Access	Data type
0: highest subindex supported	2	2		2	ro	U8
1: COB-ID client -> server (Rx)	Node ID + 0x600	0		4294967295	ro	U32
2: COB-ID server -> client (Tx)	Node ID + 0x580	0		4294967295	ro	U32

Identifier for the SDO server channel 1 (basic SDO channel).

- The basic SDO channel can neither be changed nor deactivated according to DS301, V4.02.

Subindex	Meaning
1	Receive identifier specification For SDO server channel 1: node address (C13350 / C14350) + 0x600
2	Send identifier specification For SDO server channel 1: node address (C13350 / C14350) + 0x580

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0	0	Extended identifier*				11-bit identifier	
* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".							

* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".

[14-9] Data telegram assignment

I-1201 - SDO2 server parameter

Index: I-1201	Name: SDO2 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for the SDO server channel 2.

- ▶ The server SDO parameter is only valid if the bit 31 is set to "0" for both transmission directions (subindex 1 and 2).
- ▶ In the Lenze setting, the SDO server channels 2 ... 10 are deactivated (bit 31 = "1").
- ▶ The identifier may only be changed if the SDO is invalid (bit 31 = "1").

Subindex	Meaning
1	Receive identifier specification
2	Send identifier specification
3	Node address of the client

8th byte		7th byte		6th byte	5th byte
Data 4		Data 3		Data 2	Data 1
Bit 31	Bit 30	Bit 29 ... bit 11			Bit 10 ... bit 0
0/1	0	Extended identifier*			11-bit identifier

* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".

[14-10] Data telegram assignment

Bit	Setting
Bit 31	0 SDO valid.
	1 SDO invalid.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

Example The parameter data channel 2 of the controller with the node address 4 is to be activated.

- ▶ For this purpose, bit 31 in the subindexes 1 and 2 of the [I-1201](#) object must be set to the value "0" (= "SDO valid").
- ▶ Both "Write request" commands must be transmitted from the master to the nodes via the basic SDO channel.

Calculation of the identifiers

- ▶ Identifier (COB-ID) = basic identifier + node address (node ID)
- ▶ Basic identifier SDO2 from the master to the drive: 1600 (0x640)
→ identifier = 0x640 + 0x4 = 0x644
- ▶ Basic identifier SDO2 from the drive to the master: 1472 (0x5C0)
→ identifier = 0x5C0 + 0x4 = 0x5C4

Resulting data (data 1 ... data 4)

8th byte			7th byte			6th byte			5th byte		
Data 4			Data 3			Data 2			Data 1		
Bit 31	Bit 30	Bit 29 ... bit 11						Bit 10 ... bit 0			
0	0	Extended identifier = 0						11-bit identifier = 0x644			
0x00			0x00			0x06			0x44		

[14-11] Data telegram assignment for subindices 1

8th byte			7th byte			6th byte			5th byte		
Data 4			Data 3			Data 2			Data 1		
Bit 31	Bit 30	Bit 29 ... bit 11						Bit 10 ... bit 0			
0	0	Extended identifier = 0						11-bit identifier = 0x5C4			
0x00			0x00			0x05			0xC4		

[14-12] Data telegram assignment for subindices 2

Assignment of the user data

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x23	0x01	0x12	0x01	0x44	0x06	0x00	0x00

[14-13] Assignment of the user data for writing the subindex 1

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte
Command	Index		Subindex	Data 1	Data 2	Data 3	Data 4
0x23	0x01	0x12	0x02	0xC4	0x05	0x00	0x00

[14-14] Assignment of the user data for writing the subindex 2

I-1202 - SDO3 server parameter

Index: I-1202	Name: SDO3 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for SDO server channel 3. For description see object [I-1201](#).

I-1203 - SDO4 server parameter

Index: I-1203	Name: SDO4 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for SDO server channel 4. For description see object [I-1201](#).

I-1204 - SDO5 server parameter

Index: I-1204	Name: SDO5 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for SDO server channel 5. For description see object [I-1201](#).

I-1205 - SDO6 server parameter

Index: I-1205	Name: SDO6 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for SDO server channel 6. For description see object [I-1201](#).

I-1206 - SDO7 server parameter

Index: I-1206	Name: SDO7 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for SDO server channel 7. For description see object [I-1201](#).

I-1207 - SDO8 server parameter

Index: I-1207	Name: SDO8 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for SDO server channel 8. For description see object [I-1201](#).

I-1208 - SDO9 server parameter

Index: I-1208	Name: SDO9 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for SDO server channel 9. For description see object [I-1201](#).

I-1209 - SDO10 server parameter

Index: I-1209	Name: SDO10 server parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	3	- (only read access)			ro	U8
1: COB-ID client -> server (Rx)	0x80000000	0		4294967295	rw	U32
2: COB-ID server -> client (Tx)	0x80000000	0		4294967295	rw	U32
3: Node-ID of the SDO client	1 ... 127	- (only read access)			ro	U32

Setting of the identifiers for SDO server channel 10. For description see object [I-1201](#).

I-1400 - RPDO1 communication parameter

Index: I-1400	Name: RPDO1 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	5	- (only read access)			ro	U8
1: COB-ID used by RPDO	0x200 + node ID	0		4294967295	rw	U32
2: transmission type	254	0		255	rw	U8
3: inhibit time	-	- (for RPDOs unused)			rw	U16
4: compatibility entry	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	-	- (for RPDOs unused)			rw	U16

Communication parameter for the reception of process data via RPDO1

Subindex	Meaning	Lenze code
1	Identifier RPDO1 The basic setting is according to the "Predefined Connection Set": Identifier = 0x200 + node ID	C13321/1 / C14321/1
2	RPDO transmission type according to DS301, V4.02 ▶ Transmission type (□ 47)	C13323/1 / C14323/1

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0/1	0/1	Extended identifier*				11-bit identifier	

* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".

[14-15] Data telegram assignment

Bit	Setting
Bit 30	0 RTR to this PDO possible (cannot be set).
	1 RTR to this PDO not possible (Lenze).
Bit 31	0 PDO active.
	1 PDO inactive.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

I-1401 - RPDO2 communication parameter

Index: I-1401	Name: RPDO2 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	5	- (only read access)			ro	U8
1: COB-ID used by RPDO	0x300 + node ID	0		4294967295	rw	U32
2: transmission type	254	0		255	rw	U8
3: inhibit time	-	- (for RPDOs unused)			rw	U16
4: compatibility entry	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	-	- (for RPDOs unused)			rw	U16

Communication parameter for the reception of process data via RPDO2

Subindex	Meaning	Lenze code
1	Identifier RPDO2 The basic setting is according to the "Predefined Connection Set": Identifier = 0x300 + node ID	C13321/2 / C14321/2
2	RPDO transmission type according to DS301, V4.02 ► Transmission type (47)	C13323/2 / C14323/2

► For assignment of the data telegram see object [I-1400](#).

I-1402 - RPDO3 communication parameter

Index: I-1402	Name: RPDO3 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	5	- (only read access)			ro	U8
1: COB-ID used by RPDO	0x400 + node ID	0		4294967295	rw	U32
2: transmission type	254	0		255	rw	U8
3: inhibit time	-	- (for RPDOs unused)			rw	U16
4: compatibility entry	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	-	- (for RPDOs unused)			rw	U16

Communication parameter for the reception of process data via RPDO3

Subindex	Meaning	Lenze code
1	Identifier RPDO3 The basic setting is according to the "Predefined Connection Set": Identifier = 0x400 + node ID	C13321/3 / C14321/3
2	RPDO transmission type according to DS301, V4.02 ► Transmission type (47)	C13323/3 / C14323/3

► For assignment of the data telegram see object [I-1400](#).

I-1403 - RPDO4 communication parameter

Index: I-1403	Name: RPDO4 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	5	- (only read access)			ro	U8
1: COB-ID used by RPDO	0x500 + node ID	0		4294967295	rw	U32
2: transmission type	254	0		255	rw	U8
3: inhibit time	-	- (for RPDOs unused)			rw	U16
4: compatibility entry	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	-	- (for RPDOs unused)			rw	U16

Communication parameter for the reception of process data via RPDO4

Subindex	Meaning	Lenze code
1	Identifier RPDO4 The basic setting is according to the "Predefined Connection Set": Identifier = 0x500 + node ID	C13321/4 / C14321/4
2	RPDO transmission type according to DS301, V4.02 ▶ Transmission type (47)	C13323/4 / C14323/4

► For assignment of the data telegram see object [I-1400](#).

I-1600 - RPDO1 mapping parameter

Index: I-1600	Name: RPDO1 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

This object serves to receive parameter data as RPDO1.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO1

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
Bit 31 ... bit 16		Bit 15 ... bit 8	Bit 7 ... bit 0
Index		Subindex	Length

[14-16] Data telegram assignment

IEC 61131 process data words are mapped. Only whole bytes can be mapped (the granularity of the mapping entries is 1 byte).

I-1601 - RPDO2 mapping parameter

Index: I-1601	Name: RPDO2 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

This object serves to receive parameter data as RPDO2.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO2

► For assignment of the data telegram see object [I-1600](#).

I-1602 - RPDO3 mapping parameter

Index: I-1602	Name: RPDO3 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

This object serves to receive parameter data as RPDO3.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO3

► For assignment of the data telegram see object [I-1600](#).

I-1603 - RPDO4 mapping parameter

Index: I-1603	Name: RPDO4 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

This object serves to receive parameter data as RPDO4.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO4

► For assignment of the data telegram see object [I-1600](#).

I-1800 - TPDO1 communication parameter

Index: I-1800	Name: TPDO1 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	5	- (only read access)			ro	U8
1: COB-ID used by TPDO	0x180 + node ID	0		4294967295	rw	U32
2: transmission type	254	0		255	rw	U8
3: inhibit time	0 ms	0	0.1 ms	65535	rw	U16
4: reserved	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	0 ms	0	ms	65535	rw	U16

Communication parameter for the transmission of process data via TPDO1

Subindex	Meaning	Lenze code
1	Identifier TPDO1 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x180 + node ID	C13320/1 / C14320/1
2	TPDO transmission type according to DS301, V4.02 ► Transmission type (□ 47)	C13322/1 / C14322/1
3	Minimum time between the transmission of two identical TPDOs (see DS301, V4.02).	C13324/1 / C14324/1
5	Cycle time with which the PDOs are transmitted with the transmission type "254".	C13356/1 / C14356/1

8th byte		7th byte		6th byte		5th byte	
Data 4		Data 3		Data 2		Data 1	
Bit 31	Bit 30	Bit 29 ... bit 11				Bit 10 ... bit 0	
0/1	0/1	Extended identifier*				11-bit identifier	
* The extended identifier is not supported - bit 11 ... bit 29 must be set to "0".							

[14-17] Data telegram assignment

Bit	Setting
Bit 30	0 RTR to this PDO possible (Lenze).
	1 RTR to this PDO not possible (cannot be set).
Bit 31	0 PDO active.
	1 PDO inactive.

How to change the identifier:

1. Deactivate identifier (set bit 31 to "1").
2. Change identifier.
3. Activate identifier (set bit 31 to "0").

Subindex 3 - inhibit time

The delay time can only be changed if the PDO is inactive (subindex 1, bit 31 = 1). The entered value multiplied by 0.1 results in the delay time in [ms]. The calculated delay time is always rounded down to an inter value.

Example:

- ▶ Value entered: 26
- ▶ Calculated time = $26 * 0.1 \text{ [ms]} = 2.6 \text{ [ms]} \rightarrow \text{delay time} = 2 \text{ [ms]}$

I-1801 - TPDO2 communication parameter

Index: I-1801	Name: TPDO2 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	5	- (only read access)			ro	U8
1: COB-ID used by TPDO	0x280 + node ID	0		4294967295	rw	U32
2: transmission type	254	0		255	rw	U8
3: inhibit time	0 ms	0	0.1 ms	65535	rw	U16
4: reserved	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	0 ms	0	ms	65535	rw	U16

Communication parameter for the transmission of process data via TPDO2

Subindex	Meaning	Lenze code
1	Identifier TPDO2 The basic setting is according to the "Predefined Connection Set": Identifier = 0x280 + node ID	C13320/2 / C14320/2
2	TPDO transmission type according to DS301, V4.02 ▶ Transmission type (□ 47)	C13322/2 / C14322/2
3	Minimum time between the transmission of two identical TPDOs (see DS301, V4.02).	C13324/2 / C14324/2
5	Cycle time with which the PDOs are transmitted with the transmission type "254".	C13356/2 / C14356/2

- ▶ For assignment of the data telegram see object [I-1800](#).

I-1802 - TPDO3 communication parameter

Index: I-1802	Name: TPDO3 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	5	- (only read access)			ro	U8
1: COB-ID used by TPDO	0x380 + node ID	0		4294967295	rw	U32
2: transmission type	254	0		255	rw	U8
3: inhibit time	0 ms	0	0.1 ms	65535	rw	U16
4: reserved	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	0 ms	0	ms	65535	rw	U16

Communication parameter for the transmission of process data via TPDO3

Subindex	Meaning	Lenze code
1	Identifier TPDO3 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x380 + node ID	C13320/3 / C14320/3
2	TPDO transmission type according to DS301, V4.02 ► Transmission type (47)	C13322/3 / C14322/3
3	Minimum time between the transmission of two identical TPDOs (see DS301, V4.02).	C13324/3 / C14324/3
5	Cycle time with which the PDOs are transmitted with the transmission type "254".	C13356/3 / C14356/3

► For assignment of the data telegram see object [I-1800](#).

I-1803 - TPDO4 communication parameter

Index: I-1803	Name: TPDO4 communication parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: highest subindex supported	5	- (only read access)			ro	U8
1: COB-ID used by TPDO	0x480 + node ID	0		4294967295	rw	U32
2: transmission type	254	0		255	rw	U8
3: inhibit time	0 ms	0	0.1 ms	65535	rw	U16
4: reserved	-	- (reserved, write or read access results in error message 0x06090011)			rw	U8
5: event timer	0 ms	0	ms	65535	rw	U16

Communication parameter for the transmission of process data via TPDO4

Subindex	Meaning	Lenze code
1	Identifier TPDO4 • The basic setting is according to the "Predefined Connection Set": Identifier = 0x480 + node ID	C13320/4 / C14320/4
2	TPDO transmission type according to DS301, V4.02 ► Transmission type (47)	C13322/4 / C14322/4
3	Minimum time between the transmission of two identical TPDOs (see DS301, V4.02).	C13324/4 / C14324/4
5	Cycle time with which the PDOs are transmitted with the transmission type "254".	C13356/4 / C14356/4

► For assignment of the data telegram see object [I-1800](#).

I-1A00 - TPDO1 mapping parameter

Index: I-1A00	Name: TPDO1 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

This object serves to transmit parameter data as TPDO1.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entry 1 ... 8 for TPDO1

8th byte	7th byte	6th byte	5th byte
Data 4	Data 3	Data 2	Data 1
Bit 31 ... bit 16		Bit 15 ... bit 8	Bit 7 ... bit 0
Index		Subindex	Length

[14-18] Data telegram assignment

IEC 61131 process data words are mapped. Only whole bytes can be mapped (the granularity of the mapping entries is 1 byte).

I-1A01 - TPDO2 mapping parameter

Index: I-1A01	Name: TPDO2 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

This object serves to transmit parameter data as TPDO2.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO2

► For assignment of the data telegram see object [I-1A00](#).

I-1A02 - TPDO3 mapping parameter

Index: I-1A02	Name: TPDO3 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

This object serves to transmit parameter data as TPDO3.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for RPDO4

► For assignment of the data telegram see object [I-1A00](#).

I-1A03 - TPDO4 mapping parameter

Index: I-1A03	Name: TPDO4 mapping parameter					
Subindex	Default setting	Setting range (min. value unit max. value)			Access	Data type
0: number of mapped application objects in PDO	0	0		8	rw	U8
1 ... 8: application object 1 ... 8	0	0		4294967295	rw	U32

This object serves to transmit parameter data as TPDO4.

Subindex	Meaning
0	Number of mapped objects
1 ... 8	Mapping entries 1 ... 8 for TPDO4

► For assignment of the data telegram see object [I-1A00](#).

15 DIP switch positions for setting the CAN node address

The node address results from the sum of the binary values of switches 1 ... 64.

The following table shows the switch positions for the valid address range of 1 ... 127.

► [Setting the node address](#) (33)

Node address	DIP switch						
	64	32	16	8	4	2	1
1	OFF	OFF	OFF	OFF	OFF	OFF	ON
2	OFF	OFF	OFF	OFF	OFF	ON	OFF
3	OFF	OFF	OFF	OFF	OFF	ON	ON
4	OFF	OFF	OFF	OFF	ON	OFF	OFF
5	OFF	OFF	OFF	OFF	ON	OFF	ON
6	OFF	OFF	OFF	OFF	ON	ON	OFF
7	OFF	OFF	OFF	OFF	ON	ON	ON
8	OFF	OFF	OFF	ON	OFF	OFF	OFF
9	OFF	OFF	OFF	ON	OFF	OFF	ON
10	OFF	OFF	OFF	ON	OFF	ON	OFF
11	OFF	OFF	OFF	ON	OFF	ON	ON
12	OFF	OFF	OFF	ON	ON	OFF	OFF
13	OFF	OFF	OFF	ON	ON	OFF	ON
14	OFF	OFF	OFF	ON	ON	ON	OFF
15	OFF	OFF	OFF	ON	ON	ON	ON
16	OFF	OFF	ON	OFF	OFF	OFF	OFF
17	OFF	OFF	ON	OFF	OFF	OFF	ON
18	OFF	OFF	ON	OFF	OFF	ON	OFF
19	OFF	OFF	ON	OFF	OFF	ON	ON
20	OFF	OFF	ON	OFF	ON	OFF	OFF
21	OFF	OFF	ON	OFF	ON	OFF	ON
22	OFF	OFF	ON	OFF	ON	ON	OFF
23	OFF	OFF	ON	OFF	ON	ON	ON
24	OFF	OFF	ON	ON	OFF	OFF	OFF
25	OFF	OFF	ON	ON	OFF	OFF	ON
26	OFF	OFF	ON	ON	OFF	ON	OFF
27	OFF	OFF	ON	ON	OFF	ON	ON
28	OFF	OFF	ON	ON	ON	OFF	OFF
29	OFF	OFF	ON	ON	ON	OFF	ON
30	OFF	OFF	ON	ON	ON	ON	OFF
31	OFF	OFF	ON	ON	ON	ON	ON
32	OFF	ON	OFF	OFF	OFF	OFF	OFF
33	OFF	ON	OFF	OFF	OFF	OFF	ON
34	OFF	ON	OFF	OFF	OFF	ON	OFF
35	OFF	ON	OFF	OFF	OFF	ON	ON
36	OFF	ON	OFF	OFF	ON	OFF	OFF
37	OFF	ON	OFF	OFF	ON	OFF	ON
38	OFF	ON	OFF	OFF	ON	ON	OFF

Node address	DIP switch						
	64	32	16	8	4	2	1
39	OFF	ON	OFF	OFF	ON	ON	ON
40	OFF	ON	OFF	ON	OFF	OFF	OFF
41	OFF	ON	OFF	ON	OFF	OFF	ON
42	OFF	ON	OFF	ON	OFF	ON	OFF
43	OFF	ON	OFF	ON	OFF	ON	ON
44	OFF	ON	OFF	ON	ON	OFF	OFF
45	OFF	ON	OFF	ON	ON	OFF	ON
46	OFF	ON	OFF	ON	ON	ON	OFF
47	OFF	ON	OFF	ON	ON	ON	ON
48	OFF	ON	ON	OFF	OFF	OFF	OFF
49	OFF	ON	ON	OFF	OFF	OFF	ON
50	OFF	ON	ON	OFF	OFF	ON	OFF
51	OFF	ON	ON	OFF	OFF	ON	ON
52	OFF	ON	ON	OFF	ON	OFF	OFF
53	OFF	ON	ON	OFF	ON	OFF	ON
54	OFF	ON	ON	OFF	ON	ON	OFF
55	OFF	ON	ON	OFF	ON	ON	ON
56	OFF	ON	ON	ON	OFF	OFF	OFF
57	OFF	ON	ON	ON	OFF	OFF	ON
58	OFF	ON	ON	ON	OFF	ON	OFF
59	OFF	ON	ON	ON	OFF	ON	ON
60	OFF	ON	ON	ON	ON	OFF	OFF
61	OFF	ON	ON	ON	ON	OFF	ON
62	OFF	ON	ON	ON	ON	ON	OFF
63	OFF	ON	ON	ON	ON	ON	ON
64	ON	OFF	OFF	OFF	OFF	OFF	OFF
65	ON	OFF	OFF	OFF	OFF	OFF	ON
66	ON	OFF	OFF	OFF	OFF	ON	OFF
67	ON	OFF	OFF	OFF	OFF	ON	ON
68	ON	OFF	OFF	OFF	ON	OFF	OFF
69	ON	OFF	OFF	OFF	ON	OFF	ON
70	ON	OFF	OFF	OFF	ON	ON	OFF
71	ON	OFF	OFF	OFF	ON	ON	ON
72	ON	OFF	OFF	ON	OFF	OFF	OFF
73	ON	OFF	OFF	ON	OFF	OFF	ON
74	ON	OFF	OFF	ON	OFF	ON	OFF
75	ON	OFF	OFF	ON	OFF	ON	ON
76	ON	OFF	OFF	ON	ON	OFF	OFF
77	ON	OFF	OFF	ON	ON	OFF	ON
78	ON	OFF	OFF	ON	ON	ON	OFF
79	ON	OFF	OFF	ON	ON	ON	ON
80	ON	OFF	ON	OFF	OFF	OFF	OFF
81	ON	OFF	ON	OFF	OFF	OFF	ON

E94AYCCA communication manual (CANopen®)

DIP switch positions for setting the CAN node address

Node address	DIP switch						
	64	32	16	8	4	2	1
82	ON	OFF	ON	OFF	OFF	ON	OFF
83	ON	OFF	ON	OFF	OFF	ON	ON
84	ON	OFF	ON	OFF	ON	OFF	OFF
85	ON	OFF	ON	OFF	ON	OFF	ON
86	ON	OFF	ON	OFF	ON	ON	OFF
87	ON	OFF	ON	OFF	ON	ON	ON
88	ON	OFF	ON	ON	OFF	OFF	OFF
89	ON	OFF	ON	ON	OFF	OFF	ON
90	ON	OFF	ON	ON	OFF	ON	OFF
91	ON	OFF	ON	ON	OFF	ON	ON
92	ON	OFF	ON	ON	ON	OFF	OFF
93	ON	OFF	ON	ON	ON	OFF	ON
94	ON	OFF	ON	ON	ON	ON	OFF
95	ON	OFF	ON	ON	ON	ON	ON
96	ON	ON	OFF	OFF	OFF	OFF	OFF
97	ON	ON	OFF	OFF	OFF	OFF	ON
98	ON	ON	OFF	OFF	OFF	ON	OFF
99	ON	ON	OFF	OFF	OFF	ON	ON
100	ON	ON	OFF	OFF	ON	OFF	OFF
101	ON	ON	OFF	OFF	ON	OFF	ON
102	ON	ON	OFF	OFF	ON	ON	OFF
103	ON	ON	OFF	OFF	ON	ON	ON
104	ON	ON	OFF	ON	OFF	OFF	OFF
105	ON	ON	OFF	ON	OFF	OFF	ON
106	ON	ON	OFF	ON	OFF	ON	OFF
107	ON	ON	OFF	ON	OFF	ON	ON
108	ON	ON	OFF	ON	ON	OFF	OFF
109	ON	ON	OFF	ON	ON	OFF	ON
110	ON	ON	OFF	ON	ON	ON	OFF
111	ON	ON	OFF	ON	ON	ON	ON
112	ON	ON	ON	OFF	OFF	OFF	OFF
113	ON	ON	ON	OFF	OFF	OFF	ON
114	ON	ON	ON	OFF	OFF	ON	OFF
115	ON	ON	ON	OFF	OFF	ON	ON
116	ON	ON	ON	OFF	ON	OFF	OFF
117	ON	ON	ON	OFF	ON	OFF	ON
118	ON	ON	ON	OFF	ON	ON	OFF
119	ON	ON	ON	OFF	ON	ON	ON
120	ON	ON	ON	ON	OFF	OFF	OFF
121	ON	ON	ON	ON	OFF	OFF	ON
122	ON	ON	ON	ON	OFF	ON	OFF
123	ON	ON	ON	ON	OFF	ON	ON
124	ON	ON	ON	ON	ON	OFF	OFF

Node address	DIP switch						
	64	32	16	8	4	2	1
125	ON	ON	ON	ON	ON	OFF	ON
126	ON	ON	ON	ON	ON	ON	OFF
127	ON	ON	ON	ON	ON	ON	ON

16 Index

A

Activating changed settings [36](#)
 Addressing through index and subindex [65](#)
 Application as directed [14](#)
 Application notes (representation) [11](#)
 Approvals [17](#)

B

Baud rates [17](#)
 Before initial switch-on [32](#)
 Bus cable length [28](#)
 Bus cable specification [27](#)

C

C00369 - CAN sync transmission cycle time [101](#)
 C00615 - Resp. to imp. device config. [102](#)
 C00636 - Resp. to new module in MX11 [102](#)
 C00637 - Resp. to new module in MX12 [102](#)
 C01120 - Sync source [103](#)
 C01121 - Sync cycle time [103](#)
 C01122 - Sync phase position [104](#)
 C01123 - Sync tolerance [104](#)
 C01124 - Sync-PLL increment [105](#)
 C01130 - CAN SYNC application cycle [106](#)
 C13311 - CAN TPDO1 mask byte x [107](#)
 C13312 - CAN TPDO2 mask byte x [107](#)
 C13313 - CAN TPDO3 mask byte x [108](#)
 C13314 - CAN TPDO4 mask byte x [108](#)
 C13320 - CAN TPDOx identifier [109](#)
 C13321 - CAN RPDOx identifier [110](#)
 C13322 - CAN TPDOx Tx mode [110](#)
 C13323 - CAN RPDOx Rx mode [111](#)
 C13324 - CAN TPDOx delay time [111](#)
 C13343 - CAN TPDO counter [111](#)
 C13344 - CAN RPDO counter [112](#)
 C13345 - CAN error [112](#)
 C13346 - CAN heartbeat activity [112](#)
 C13347 - CAN heartbeat status [113](#)
 C13348 - CAN status DIP switch [113](#)
 C13349 - CAN setting of DIP switch [113](#)
 C13350 - CAN node address [114](#)
 C13351 - CAN baud rate [114](#)
 C13352 - CAN slave/master [114](#)
 C13356 - CAN TPDOx cycle time [115](#)
 C13357 - CAN RPDOx monitoring time [115](#)
 C13359 - CAN status [115](#)
 C13360 - CAN telegram and error counter [116](#)
 C13361 - CAN bus load [116](#)
 C13367 - CAN SYNC Rx identifier [117](#)
 C13368 - CAN SYNC Tx identifier [117](#)
 C13372 - CAN SDO server Rx identifier [117](#)

C13373 - CAN SDO server Tx identifier [118](#)
 C13374 - CAN SDO client node address [118](#)
 C13375 - CAN SDO client Rx identifier [119](#)
 C13376 - CAN SDO client Tx identifier [119](#)
 C13377 - CAN SDO server node address [120](#)
 C13378 - CAN delay boot-up - Operational [120](#)
 C13381 - CAN Heartbeat Producer Time [120](#)
 C13382 - CAN Guard Time [120](#)
 C13383 - CAN Life Time Factor [121](#)
 C13385 - CAN Heartbeat Consumer Time [121](#)
 C13386 - CAN Node Guarding [122](#)
 C13387 - CAN Node Guarding Activity [122](#)
 C13388 - CAN node guarding status [122](#)
 C13390 - CAN Error Register (DS301V402) [123](#)
 C13391 - CAN Emergency Object [123](#)
 C13392 - CAN emergency delay time [124](#)
 C13393 - CAN result - bus scan [124](#)
 C13394 - CAN predefined error field (DS301V402) [124](#)
 C13591 - Resp. to CAN-RPDOx error [125](#)
 C13595 - Resp. to CAN bus OFF [125](#)
 C13612 - Resp. to CAN node guarding error [126](#)
 C13613 - Resp. to CAN heartbeat error [126](#)
 C13614 - Resp. to CAN life guarding error [127](#)
 C13625 - CAN behaviour in case of fault [127](#)
 C14311 - CAN TPDO1 mask byte x [128](#)
 C14312 - CAN TPDO2 mask byte x [128](#)
 C14313 - CAN TPDO3 mask byte x [129](#)
 C14314 - CAN TPDO4 mask byte x [129](#)
 C14320 - CAN TPDOx identifier [130](#)
 C14321 - CAN RPDOx identifier [131](#)
 C14322 - CAN TPDOx Tx mode [131](#)
 C14323 - CAN RPDOx Rx mode [132](#)
 C14324 - CAN TPDOx delay time [132](#)
 C14343 - CAN TPDO counter [132](#)
 C14344 - CAN RPDO counter [133](#)
 C14345 - CAN error [133](#)
 C14346 - CAN heartbeat activity [133](#)
 C14347 - CAN heartbeat status [134](#)
 C14348 - CAN status DIP switch [134](#)
 C14349 - CAN setting of DIP switch [134](#)
 C14350 - CAN node address [135](#)
 C14351 - CAN baud rate [135](#)
 C14352 - CAN slave/master [135](#)
 C14356 - CAN TPDOx cycle time [136](#)
 C14357 - CAN RPDOx monitoring time [136](#)
 C14359 - CAN status [136](#)
 C14360 - CAN telegram and error counter [137](#)
 C14361 - CAN bus load [137](#)
 C14367 - CAN SYNC Rx identifier [138](#)
 C14368 - CAN SYNC Tx identifier [138](#)
 C14372 - CAN SDO server Rx identifier [138](#)

- C14373 - CAN SDO server Tx identifier [139](#)
- C14374 - CAN SDO client node address [139](#)
- C14375 - CAN SDO client Rx identifier [140](#)
- C14376 - CAN SDO client Tx identifier [140](#)
- C14377 - CAN SDO server node address [141](#)
- C14378 - CAN delay boot-up - Operational [141](#)
- C14381 - CAN Heartbeat Producer Time [141](#)
- C14382 - CAN Guard Time [141](#)
- C14383 - CAN Life Time Factor [142](#)
- C14385 - CAN Heartbeat Consumer Time [142](#)
- C14386 - CAN Node Guarding [143](#)
- C14387 - CAN Node Guarding Activity [143](#)
- C14388 - CAN node guarding status [143](#)
- C14390 - CAN Error Register (DS301V402) [144](#)
- C14391 - CAN Emergency Object [144](#)
- C14392 - CAN emergency delay time [145](#)
- C14393 - CAN result - bus scan [145](#)
- C14394 - CAN predefined error field (DS301V402) [145](#)
- C14591 - Resp. to CAN-RPDOx error [146](#)
- C14595 - Resp. to CAN bus OFF [146](#)
- C14612 - Resp. to CAN node guarding error [147](#)
- C14613 - Resp. to CAN heartbeat error [147](#)
- C14614 - Resp. to CAN life guarding error [148](#)
- C14625 - CAN behaviour in case of fault [148](#)
- CAN baud rate (C13351) [114](#)
- CAN baud rate (C14351) [135](#)
- CAN behaviour in case of fault (C13625) [127](#)
- CAN behaviour in case of fault (C14625) [148](#)
- CAN bus load (C13361) [116](#)
- CAN bus load (C14361) [137](#)
- CAN data telegram [37](#)
- CAN delay boot-up - Operational (C13378) [120](#)
- CAN delay boot-up - Operational (C14378) [141](#)
- CAN emergency delay time (C13392) [124](#)
- CAN emergency delay time (C14392) [145](#)
- CAN emergency object (C13391) [123](#)
- CAN emergency object (C14391) [144](#)
- CAN error (C13345) [112](#)
- CAN error (C14345) [133](#)
- CAN error register (DS301V402) (C13390) [123](#)
- CAN error register (DS301V402) (C14390) [144](#)
- CAN guard time (C13382) [120](#)
- CAN guard time (C14382) [141](#)
- CAN heartbeat activity (C13346) [112](#)
- CAN heartbeat activity (C14346) [133](#)
- CAN heartbeat consumer time (C13385) [121](#)
- CAN heartbeat consumer time (C14385) [142](#)
- CAN heartbeat producer time (C13381) [120](#)
- CAN heartbeat producer time (C14381) [141](#)
- CAN heartbeat status (C13347) [113](#)
- CAN heartbeat status (C14347) [134](#)
- CAN life time factor (C13383) [121](#)
- CAN life time factor (C14383) [142](#)
- CAN module (MXI1)
 - basic configuration invalid (error number) [92](#)
 - bus-off (error message) [91](#)
 - emergency configuration faulty (error message) [93](#)
 - heartbeat error index 1 (error message) [92](#)
 - invalid node address 0 (error message) [91](#)
 - life guarding error (error message) [92](#)
 - NMT master configuration faulty (error message) [93](#)
 - NMT slave configuration faulty (error message) [92](#)
 - node guarding error 1 (error message) [93](#)
- CAN module (MXI1) PDO manager
 - configuration faulty (error message) [95](#)
- CAN module (MXI1) RPDO1
 - telegram not received or faulty (error message) [93](#)
- CAN module (MXI1) RPDO2
 - telegram not received or faulty (error message) [94](#)
- CAN module (MXI1) RPDO3
 - telegram not received or faulty (error message) [94](#)
- CAN module (MXI1) RPDO4
 - telegram not received or faulty (error message) [94](#)
- CAN module (MXI1) RPDO5
 - telegram not received or faulty (error message) [94](#)
- CAN module (MXI1) RPDO6
 - telegram not received or faulty (error message) [94](#)
- CAN module (MXI1) RPDO7
 - telegram not received or faulty (error message) [95](#)
- CAN module (MXI1) RPDO8
 - telegram not received or faulty (error message) [95](#)
- CAN module (MXI1) SDO client
 - configuration faulty (error message) [96](#)
- CAN module (MXI1) SDO server
 - configuration faulty (error message) [95](#)
- CAN module (MXI2)
 - basic configuration invalid (error number) [96](#)
 - bus-off (error message) [96](#)
 - emergency configuration faulty (error message) [97](#)
 - heartbeat error index 1 (error message) [97](#)
 - invalid node address 0 (error message) [96](#)
 - life guarding error (error message) [97](#)
 - NMT master configuration faulty (error message) [98](#)
 - NMT slave configuration faulty (error message) [97](#)
 - node guarding error 1 (error message) [98](#)
- CAN module (MXI2) PDO manager
 - configuration faulty (error message) [100](#)
- CAN module (MXI2) RPDO1
 - telegram not received or faulty (error message) [98](#)
- CAN module (MXI2) RPDO2
 - telegram not received or faulty (error message) [98](#)
- CAN module (MXI2) RPDO3
 - telegram not received or faulty (error message) [99](#)
- CAN module (MXI2) RPDO4
 - telegram not received or faulty (error message) [99](#)
- CAN module (MXI2) RPDO5
 - telegram not received or faulty (error message) [99](#)

- CAN module (MXI2) RPDO6
 - telegram not received or faulty (error message) [99](#)
- CAN module (MXI2) RPDO7
 - telegram not received or faulty (error message) [99](#)
- CAN module (MXI2) RPDO8
 - telegram not received or faulty (error message) [100](#)
- CAN module (MXI2) SDO client
 - configuration faulty (error message) [100](#)
- CAN module (MXI2) SDO server
 - configuration faulty (error message) [100](#)
- CAN node address (C13350) [114](#)
- CAN node address (C14350) [135](#)
- CAN node guarding (C13386) [122](#)
- CAN node guarding (C14386) [143](#)
- CAN node guarding activity (C13387) [122](#)
- CAN node guarding activity (C14387) [143](#)
- CAN node guarding status (C13388) [122](#)
- CAN node guarding status (C14388) [143](#)
- CAN predefined error field (DS301V402) (C13394) [124](#)
- CAN predefined error field (DS301V402) (C14394) [145](#)
- CAN result - bus scan (C13393) [124](#)
- CAN result - bus scan (C14393) [145](#)
- CAN RPDO counter (C13344) [112](#)
- CAN RPDO counter (C14344) [133](#)
- CAN RPDOx identifier (C13321) [110](#)
- CAN RPDOx identifier (C14321) [131](#)
- CAN RPDOx monitoring time (C13357) [115](#)
- CAN RPDOx monitoring time (C14357) [136](#)
- CAN RPDOx Rx mode (C13323) [111](#)
- CAN RPDOx Rx mode (C14323) [132](#)
- CAN SDO client node address (C13374) [118](#)
- CAN SDO client node address (C14374) [139](#)
- CAN SDO client Rx identifier (C13375) [119](#)
- CAN SDO client Rx identifier (C14375) [140](#)
- CAN SDO client Tx identifier (C13376) [119](#)
- CAN SDO client Tx identifier (C14376) [140](#)
- CAN SDO server node address (C13377) [120](#)
- CAN SDO server node address (C14377) [141](#)
- CAN SDO server Rx identifier (C13372) [117](#)
- CAN SDO server Rx identifier (C14372) [138](#)
- CAN SDO server Tx identifier (C13373) [118](#)
- CAN SDO server Tx identifier (C14373) [139](#)
- CAN setting of DIP switch (C13349) [113](#)
- CAN setting of DIP switch (C14349) [134](#)
- CAN slave/master (C13352) [114](#)
- CAN slave/master (C14352) [135](#)
- CAN status (C13359) [115](#)
- CAN status (C14359) [136](#)
- CAN status DIP switch (C13348) [113](#)
- CAN status DIP switch (C14348) [134](#)
- CAN SYNC application cycle (C01130) [106](#)
- CAN SYNC Rx identifier (C13367) [117](#)
- CAN SYNC Rx identifier (C14367) [138](#)
- CAN sync transmission cycle time (C00369) [101](#)
- CAN SYNC Tx identifier (C13368) [117](#)
- CAN SYNC Tx identifier (C14368) [138](#)
- CAN telegram and error counter (C13360) [116](#)
- CAN telegram and error counter (C14360) [137](#)
- CAN TPDO counter (C13343) [111](#)
- CAN TPDO counter (C14343) [132](#)
- CAN TPDO1 mask byte x (C13311) [107](#)
- CAN TPDO1 mask byte x (C14311) [128](#)
- CAN TPDO2 mask byte x (C13312) [107](#)
- CAN TPDO2 mask byte x (C14312) [128](#)
- CAN TPDO3 mask byte x (C13313) [108](#)
- CAN TPDO3 mask byte x (C14313) [129](#)
- CAN TPDO4 mask byte x (C13314) [108](#)
- CAN TPDO4 mask byte x (C14314) [129](#)
- CAN TPDOx cycle time (C13356) [115](#)
- CAN TPDOx cycle time (C14356) [136](#)
- CAN TPDOx delay time (C13324) [111](#)
- CAN TPDOx delay time (C14324) [132](#)
- CAN TPDOx identifier (C13320) [109](#)
- CAN TPDOx identifier (C14320) [130](#)
- CAN TPDOx Tx mode (C13322) [110](#)
- CAN TPDOx Tx mode (C14322) [131](#)
- CANopen error messages
 - Causes and remedies [91](#)
- CANopen error messages (short overview) [89](#)
- CANopen objects [153](#)
- COB-ID [38](#), [46](#), [62](#)
- COB-ID EMCY (I-1014) [161](#)
- COB-ID SYNC message (I-1005) [157](#)
- Codes [101](#)
- Command [64](#)
- Commissioning [32](#)
- Communication cycle period (I-1006) [158](#)
- Communication medium [17](#)
- Communication phases [41](#)
- Communication profile [17](#)
- Communication time [19](#)
- Communication-relevant parameters of the standard device [101](#)
- Conformities [17](#)
- Consumer heartbeat time (I-1016) [162](#)
- Conventions [9](#)
- Conventions used [9](#)
- Copyright [2](#)
- D**
- Data 1 ... data 4 [66](#)
- Data telegram [37](#)
- Data transfer [37](#)
- Device and application-specific safety instructions [13](#)
- Device protection [13](#)
- Device type (I-1000) [155](#)
- Diagnostics with the »Engineer« [88](#)
- Dimensions [23](#)

DIP switch positions for setting the CAN node address [178](#)

DIP switch settings [33](#), [178](#)

Dismounting [25](#)

Document history [8](#)

E

Emergency telegram [84](#)

Error behaviour (I-1029) [163](#)

Error codes [68](#)

Error messages [89](#)

Causes and remedies [91](#)

Error messages (short overview) [89](#)

Error messages (system bus) [67](#)

Error number

0x008c0017 [91](#)

0x008c0018 [91](#)

0x009d0000 [91](#)

0x009d0001 [91](#)

0x009d0002 [92](#)

0x009e0000 [92](#)

0x009e0020 [92](#)

0x009e0021 [92](#)

0x009f0000 [93](#)

0x00a00000 [93](#)

0x00a00020 [93](#)

0x00a10000 [93](#)

0x00a10001 [94](#)

0x00a10002 [94](#)

0x00a10003 [94](#)

0x00a10004 [94](#)

0x00a10005 [94](#)

0x00a10006 [95](#)

0x00a10007 [95](#)

0x00a10008 [95](#)

0x00a20000 [95](#)

0x00a30000 [96](#)

0x00ac0000 [96](#)

0x00ac0001 [96](#)

0x00ac0002 [96](#)

0x00ad0000 [97](#)

0x00ad0020 [97](#)

0x00ad0021 [97](#)

0x00ae0000 [97](#)

0x00af0000 [98](#)

0x00af0020 [98](#)

0x00b00000 [98](#)

0x00b00001 [98](#)

0x00b00002 [99](#)

0x00b00003 [99](#)

0x00b00004 [99](#)

0x00b00005 [99](#)

0x00b00006 [99](#)

0x00b00007 [100](#)

0x00b00008 [100](#)

0x00b10000 [100](#)

0x00b20000 [100](#)

Error register (I-1001) [155](#)

Establishing communication [36](#)

F

Features [15](#)

G

General data [17](#)

General safety and application notes [12](#)

Guard time [76](#)

Guard time (I-100C) [158](#)

H

Heartbeat consumer time [82](#)

Heartbeat event [82](#)

Heartbeat producer time [81](#)

Heartbeat protocol [80](#)

I

I-1000 - Device type [155](#)

I-1001 - Error register [155](#)

I-1003 - Pre-defined error field [156](#)

I-1005 - COB-ID SYNC message [157](#)

I-1006 - Communication cycle period [158](#)

I-100C - Guard time [158](#)

I-100D - Life time factor [158](#)

I-1010 - Store parameters [159](#)

I-1011 - Restore default parameters [160](#)

I-1014 - COB-ID EMCY [161](#)

I-1015 - Inhibit time EMCY [161](#)

I-1016 - Consumer heartbeat time [162](#)

I-1017 - Producer heartbeat time [162](#)

I-1018 - Identity object [163](#)

I-1029 - Error behaviour [163](#)

I-1200 - SDO1 server parameter [164](#)

I-1201 - SDO2 server parameter [165](#)

I-1202 - SDO3 server parameter [167](#)

I-1203 - SDO4 server parameter [167](#)

I-1204 - SDO5 server parameter [167](#)

I-1205 - SDO6 server parameter [167](#)

I-1206 - SDO7 server parameter [168](#)

I-1207 - SDO8 server parameter [168](#)

I-1208 - SDO9 server parameter [168](#)

I-1209 - SDO10 server parameter [168](#)

I-1400 - RPDO1 communication parameter [169](#)

I-1401 - RPDO2 communication parameter [170](#)

I-1402 - RPDO3 communication parameter [170](#)

I-1403 - RPDO4 communication parameter [171](#)

I-1600 - RPDO1 mapping parameter [171](#)

I-1601 - RPDO2 mapping parameter [172](#)

I-1602 - RPDO3 mapping parameter [172](#)

I-1603 - RPDO4 mapping parameter [172](#)

I-1800 - TPDO1 communication parameter [173](#)

I-1801 - TPDO2 communication parameter [174](#)

I-1802 - TPDO3 communication parameter [175](#)
I-1803 - TPDO4 communication parameter [175](#)
I-1A00 - TPDO1 mapping parameter [176](#)
I-1A01 - TPDO2 mapping parameter [176](#)
I-1A02 - TPDO3 mapping parameter [177](#)
I-1A03 - TPDO4 mapping parameter [177](#)
Identification [14](#)
Identifier (CAN) [38](#)
Identifier assignment [39](#)
Identifiers of the parameter data objects (COB-ID) [62](#)
Identifiers of the process data objects (COB-ID) [46](#)
Identity object (I-1018) [163](#)
Index [65](#)
Inhibit time EMCY (I-1015) [161](#)
Initial switch-on [36](#)
Installation [24](#)
Instant of PDO acceptance [54](#)
Instant of PDO transmission [54](#)
Interfaces [16](#)

L

LED status displays [86](#)
Life guarding event [77](#)
Life time factor (I-100D) [158](#)
LS_SyncInput (system block) [88](#)
LS_SyncInput system block [88](#)

M

Masking of the TPDOs for event control [49](#)
Master functionality (CAN) [44](#)
Mechanical installation [25](#)
Monitoring [74](#)
Mounting [25](#)
MXI1
CAN module is missing or is incompatible (error message) [91](#)
MXI2
CAN module is missing or is incompatible (error message) [91](#)

N

Nameplate [14](#)
Network management [41](#)
Network management data [40](#)
Network management telegram (NMT) [43](#)
Network topology [17](#)
NMT (network management) [43](#)
Node address [38](#)
Node addresses [17](#)
Node guarding event [77](#)
Node guarding protocol [74](#)
Node ID [38](#)
Node life time [76](#)
Notes used [11](#)

Number of nodes [17](#)

O

Operating conditions [17](#)

P

Parameter data [17](#), [40](#)
Parameter data telegram (examples) [69](#)
Parameter data transfer [61](#)
Parameter reference [101](#)
Parameters of the communication module for slot MXI1 [107](#)
Parameters of the communication module for slot MXI2 [128](#)
PDO delay [54](#)
PDO mapping [56](#)
Pre-defined error field (I-1003) [156](#)
Process data [17](#), [40](#)
Process data transfer [45](#)
Producer heartbeat time (I-1017) [162](#)
Product description [14](#)
Protection against uncontrolled restart [36](#)
Protection of persons [13](#)
Protective insulation [20](#)
Protocols [18](#)

R

Reading block parameters (example) [71](#)
Reading parameters (example) [69](#)
Residual hazards [13](#)
Resp. to CAN bus OFF (C13595) [125](#)
Resp. to CAN bus OFF (C14595) [146](#)
Resp. to CAN heartbeat error (C13613) [126](#)
Resp. to CAN heartbeat error (C14613) [147](#)
Resp. to CAN life guarding error (C13614) [127](#)
Resp. to CAN life guarding error (C14614) [148](#)
Resp. to CAN node guarding error (C13612) [126](#)
Resp. to CAN node guarding error (C14612) [147](#)
Resp. to CAN-RPDOx error (C13591) [125](#)
Resp. to CAN-RPDOx error (C14591) [146](#)
Resp. to imp. device config. (C00615) [102](#)
Resp. to new module in MXI1 (C00636) [102](#)
Resp. to new module in MXI2 (C00637) [102](#)
Restore default parameters (I-1011) [160](#)
RPDO1 communication parameter (I-1400) [169](#)
RPDO1 mapping parameter (I-1600) [171](#)
RPDO2 communication parameter (I-1401) [170](#)
RPDO2 mapping parameter (I-1601) [172](#)
RPDO3 communication parameter (I-1402) [170](#)
RPDO3 mapping parameter (I-1602) [172](#)
RPDO4 communication parameter (I-1403) [171](#)
RPDO4 mapping parameter (I-1603) [172](#)

S

Safety instructions [12](#)
 Safety instructions (representation) [11](#)
 Screenshots [6](#)
 SDO1 server parameter (I-1200) [164](#)
 SDO10 server parameter (I-1209) [168](#)
 SDO2 server parameter (I-1201) [165](#)
 SDO3 server parameter (I-1202) [167](#)
 SDO4 server parameter (I-1203) [167](#)
 SDO5 server parameter (I-1204) [167](#)
 SDO6 server parameter (I-1205) [167](#)
 SDO7 server parameter (I-1206) [168](#)
 SDO8 server parameter (I-1207) [168](#)
 SDO9 server parameter (I-1208) [168](#)
 Segment cable length [29](#)
 Setting the baud rate [34](#)
 Setting the node address [33](#)
 Settings in the »Engineer« [35](#), [85](#)
 State transitions [42](#)
 Store parameters (I-1010) [159](#)
 Subindex [65](#)
 Sync application cycle [53](#)
 Sync cycle time [51](#)
 Sync cycle time (C01121) [103](#)
 Sync phase position [52](#), [54](#)
 Sync phase position (C01122) [104](#)
 Sync source [51](#)
 Sync source (C01120) [103](#)
 Sync telegram [50](#)
 Sync tolerance [52](#)
 Sync tolerance (C01123) [104](#)
 Synchronisation of PDOs [50](#)
 Sync-PLL increment [52](#)
 Sync-PLL increment (C01124) [105](#)
 System bus (CANopen) connection [26](#)
 System error messages [89](#)

T

Table of attributes [149](#)
 Target group [7](#)
 Technical data [17](#)
 Terminals [16](#)
 Terminology used [10](#)
 Terms [10](#)
 Total cable length [28](#)
 TPDO1 communication parameter (I-1800) [173](#)
 TPDO1 mapping parameter (I-1A00) [176](#)
 TPDO2 communication parameter (I-1801) [174](#)
 TPDO2 mapping parameter (I-1A01) [176](#)
 TPDO3 communication parameter (I-1802) [175](#)
 TPDO3 mapping parameter (I-1A02) [177](#)
 TPDO4 communication parameter (I-1803) [175](#)
 TPDO4 mapping parameter (I-1A03) [177](#)

Transfer mode for TPDOs [17](#)

Transmission type [47](#)

U

Use of repeaters [30](#)

User data [40](#), [63](#)

Using the communication module [14](#)

V

Validity of the documentation [7](#)

Voltage supply [17](#), [31](#)

W

Writing parameters (example) [70](#)



© 06/2012



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

EDS94AYCCA ■ 13411721 ■ EN 5.0 ■ TD17

10 9 8 7 6 5 4 3 2 1



PC-based Automation

The CANopen logo is displayed on a white rectangular background. The word "CAN" is in a bold, green, sans-serif font, and the word "open" is in a lighter green, lowercase, sans-serif font.

CANopen control technology
Commissioning & Configuration

Contents

1	About this documentation	6
1.1	Document history	8
1.2	Conventions used	9
1.3	Terminology used	10
1.4	Notes used	11
2	Safety instructions	12
3	The "PC-based Automation" system	13
4	System bus (CAN) / CANopen	15
4.1	CANopen (Logic) / CANopen (Motion)	15
4.1.1	Combination with other bus systems	16
4.1.2	Field devices	17
4.2	CANopen Hardware for your Industrial PC	18
5	Technical data	20
5.1	General data	20
5.2	Technical data of the MC-CAN2 communication card	21
5.3	Bus cable specification	21
5.4	Bus cable length	22
5.4.1	Total cable length	22
5.4.2	Segment cable length	23
5.4.3	Check use of repeater	24
6	Planning the CANopen network	25
6.1	Example of an overview screen	28
6.2	Device specifications of the field devices	29
6.2.1	Special features of the Servo Drives 9400	30
6.2.2	Special features of the Inverter Drives 8400	31
6.2.3	Special features for the I/O-System IP20 (EPM-Txxx)	32
6.2.4	Special features of the I/O-System 1000 (EPM-Sxxx)	33
6.2.5	Special features of the 8200 vector frequency inverter	34
6.2.6	Special features of the ECS servo system	35
7	Preparing the field devices	37
7.1	Installing field devices	37
7.2	Setting node addresses and the baud rate	37
7.3	Connecting the Engineering PC to the Industrial PC	38

8	<u>Commissioning the CANopen Logic bus</u>	40
8.1	<u>Overview of the commissioning steps</u>	40
8.2	<u>Creating a project folder</u>	41
8.3	<u>Commissioning of field devices</u>	42
8.3.1	<u>Going online</u>	43
8.3.2	<u>Commissioning the Servo Drives 9400</u>	44
8.3.3	<u>Commissioning of 8400 Inverter Drives</u>	47
8.3.4	<u>Commissioning of I/O system IP20 (EPM-Txxx)</u>	50
8.3.5	<u>Commissioning of I/O system 1000 (EPM-Sxxx)</u>	51
8.3.6	<u>Commissioning of 8200 vector frequency inverter</u>	52
8.3.7	<u>Commissioning of ECS devices</u>	53
8.4	<u>Creating a PLC program</u>	54
8.5	<u>Configuring the CAN master</u>	57
8.6	<u>Integrating field devices (slaves) into the PLC program</u>	59
8.7	<u>Setting of CAN parameters and CAN mapping</u>	60
8.7.1	<u>Special features of the 9400 Servo Drives</u>	63
8.7.2	<u>Special features of the 8400 Inverter Drives</u>	63
8.7.3	<u>Special features of the I/O modules IP20 "1×counter/16×digital input" and "SSI interface"</u>	63
8.7.4	<u>Special features of the 8200 vector frequency inverter</u>	63
8.7.5	<u>Special features of the ECS servo system</u>	64
8.8	<u>Creating a program code to control the device</u>	65
8.8.1	<u>Special features of the Servo Drives 9400</u>	65
8.8.2	<u>Special features of the Inverter Drives 8400</u>	65
8.8.3	<u>Special features of the I/O system IP20 (EPM-Txxx)</u>	66
8.8.4	<u>Special features of the I/O system 1000 (EPM-Sxxx)</u>	66
8.8.5	<u>Special features of the 8200 vector frequency inverter</u>	66
8.8.6	<u>Special features of the ECS servo system</u>	67
8.9	<u>Preparing the restart</u>	68
8.9.1	<u>Special features of the Servo Drives 9400</u>	69
8.9.2	<u>Special features of the Inverter Drives 8400</u>	69
8.9.3	<u>Special features of the I/O-System IP20 (EPM-Txxx)</u>	70
8.9.4	<u>Special features of the I/O-System 1000 (EPM-Sxxx)</u>	71
8.9.5	<u>Special features of the ECS servo system</u>	72

9	Commissioning the CANopen Motion bus	73
9.1	Overview of the commissioning steps	73
9.2	Commissioning of field devices	74
9.3	Creating a PLC program	75
9.4	Creating a Motion task	78
9.5	Creating a control configuration	79
9.6	Creating a program code to control the Motion drives	82
9.7	Preparing the restart	82
9.8	Optimisation of signal propagation delays (for HighLine CiA402 only)	83
9.8.1	Example 1: 3 drives in 1 ms at 1 Mbit/s	83
9.8.2	Example 2: 4 drives in 2 ms at 1 Mbps	84
10	CANopen with PROFIBUS	85
11	The function library LenzeCANdrive.lib	86
12	Defining the minimum cycle time of the PLC project	87
12.1	Calculating the total access time to the peripheral devices ($T_{\text{Correction}}$)	87
12.2	Detecting the task utilisation of the application ($T_{\text{Task utilisation}}$)	88
12.2.1	Display of the system utilisation in the »PLC Designer« with the task Editor	88
12.2.2	Detecting the task utilisation	89
12.3	Calculating the minimum cycle time	90
12.4	Optimising the system	91
13	Diagnostics	92
13.1	Reading codes	92
13.2	Viewing the logbook of the IPC	92
13.3	Error messages if communication card MC-CAN2 is not available	93
13.4	Searching the CANopen bus for nodes using the Engineering PC	93
13.5	The global variable wState	94
14	Parameter reference	96
14.1	Parameters of the MC-CAN2 communication card in slot 1	97
14.2	Parameters of the MC-CAN2 communication card in slot 2	98
15	Appendix	99
15.1	»PCAN-View« for diagnostic purposes	99
15.2	Use »PCAN-View« to set all nodes to the "Operational" state	101
15.3	Notes on visualisation using »VisiWinNET®«	102
16	Index	103

1 About this documentation

This documentation ...

- provides detailed information on how to commission, configure and diagnose the CANopen bus system in the field of Lenze control technology.
- belongs to the "PC based Automation" manual collection which includes 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 • PROFIBUS control technology • EtherCAT control technology
(Software) manual "PC-based Automation"	• Industrial PC - <i>Parameterisation & Configuration</i>
Operating Instructions "Embedded Line Panel PC"	• EL x8xx - Built-in panel PC with TFT display
Operating Instructions "Command Station"	• CS x8xx - stand-alone operator terminal
Operating Instructions "Control Cabinet PC"	• CPC x8xx - 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>Parameterisation & Configuration</i> • »Engineer« • »PLC Designer« / »PLC Designer - SoftMotion« / »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 the Inverter Drives 8400
 MAs for the 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!**

Documentation and software updates on 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	06/2008	TD17	First edition
-	2.0	09/2008	TD17	Amended by chapter " CANopen with PROFIBUS " (□ 85).
13296254	3.0	06/2009	TD17	General revision
13317281	4.0	10/2009	TD17	General revision
13369325	4.1	01/2011	TD17	Update for control technology release 2.5
13383676	4.2	07/2011	TD17	Chapter Error messages if communication card MC-CAN2 is not available (□ 93) supplemented.

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	<pre> IF var1 < var2 THEN a = a + 1 END IF </pre>
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	( 9)	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.
»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

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 damages 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 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 the 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!

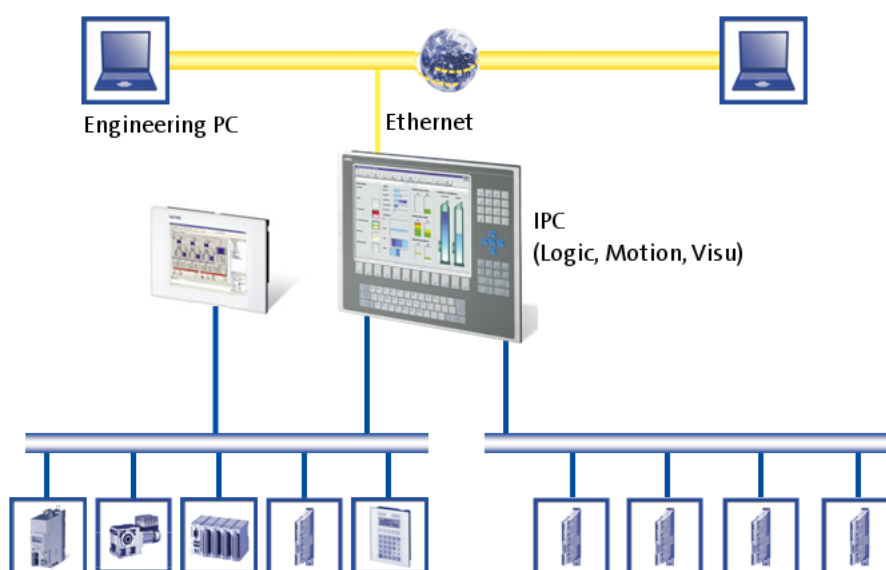
3 The "PC-based Automation" system

Industrial PCs (IPCs) become more and more important in the field of automation technology. Due to their scaling options and various combinations of visualisation and control on one device, Industrial PCs provide clear advantages for many applications.

Lenze Industrial PCs are available with the following software equipment:

- ▶ 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 enables 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!

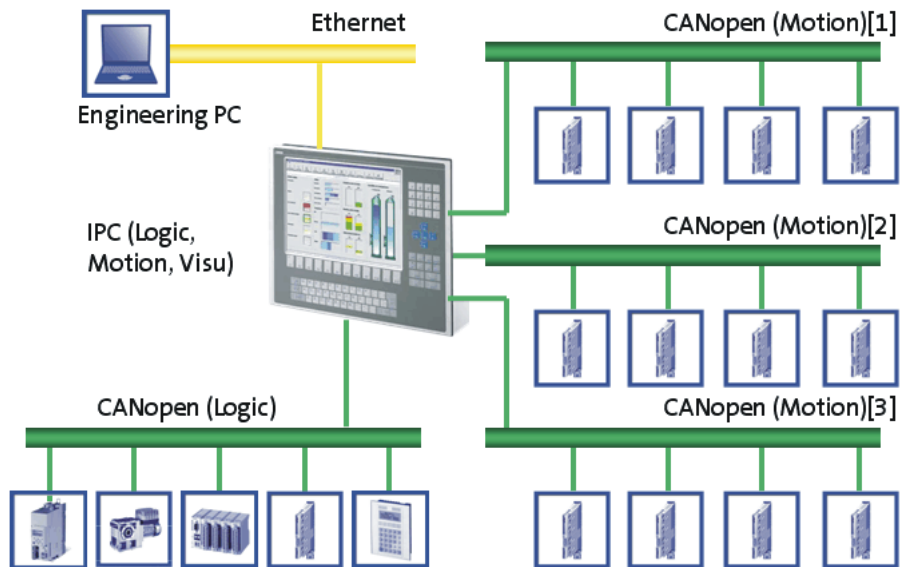
Moreover, the Z EL 1xx PLC HMI series is part of the "PC-based Automation" system. These devices differ considerably from the Industrial PCs in performance and various other details. However, the devices of the EL 1xx PLC HMI series are able to perform smaller control tasks.

- ▶ Engineering tools for the Engineering PC
 - The Engineering PC communicates with the IPC via Ethernet.
 - Different engineering tools serve to configure and parameterise the system.
- ▶ Fieldbuses
- ▶ Field devices

4 System bus (CAN) / CANopen

Lenze device series 8200 vector, 9300 and ECS have an on-board system bus (CAN) connection. The protocol used there is a subset of CANopen. Thus the devices are not CANopen-conform but can be driven by a CANopen-compatible control under "L-force Controls" - also in connection with other CANopen-compatible nodes.

4.1 CANopen (Logic) / CANopen (Motion)



Due to the demands on the real-time behaviour of the bus system and the limited transfer capacity, the CANopen bus must be divided into a Logic and a Motion bus.

The Logic bus and the Motion buses can be connected to many different field devices.

To establish a CANopen bus, use the [Communication card MC-CAN2](#) (□ 18).



Note!

Depending on the required Motion node number and bus cycle time, up to 4 Motion buses can be established.

Only 2 buses are possible for the CS x8xx Command Station IPC series.

Conventions for "PC-based Automation"

- Interface CAN1: CANopen (Logic) or CANopen (Motion)
- Interface CAN2 ... 4: CANopen (Motion)

CANopen (Logic)

The Logic bus is used to operate controllers which

- ▶ carry out simple movements,
- ▶ do not have a Motion functionality,
- ▶ are controlled via PLC functionalities only.

CANopen (Motion)

The Motion bus is used to control controllers which carry out e.g. synchronised movements.

The "L-force Motion" runtime software contains the PLCopen libraries and supports the SoftMotion control to control the "Servo Drives 9400 HighLine CiA402" series and the "ECSxM" axis module.

4.1.1 Combination with other bus systems

The CANopen bus system can be combined with PROFIBUS. This makes sense if not all field devices are available for the same bus system or a Motion bus (CANopen) is required in parallel to the PROFIBUS (as Logic bus). The bus systems are synchronised in the control.



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".
- Release 2.5 does not facilitate a combination of PROFIBUS and EtherCAT.
- In the control configuration, the PROFIBUS master must be in the first position – upstream to the CANopen motion nodes.

4.1.2 Field devices

The Lenze control system supports the following Logic/Motion components:

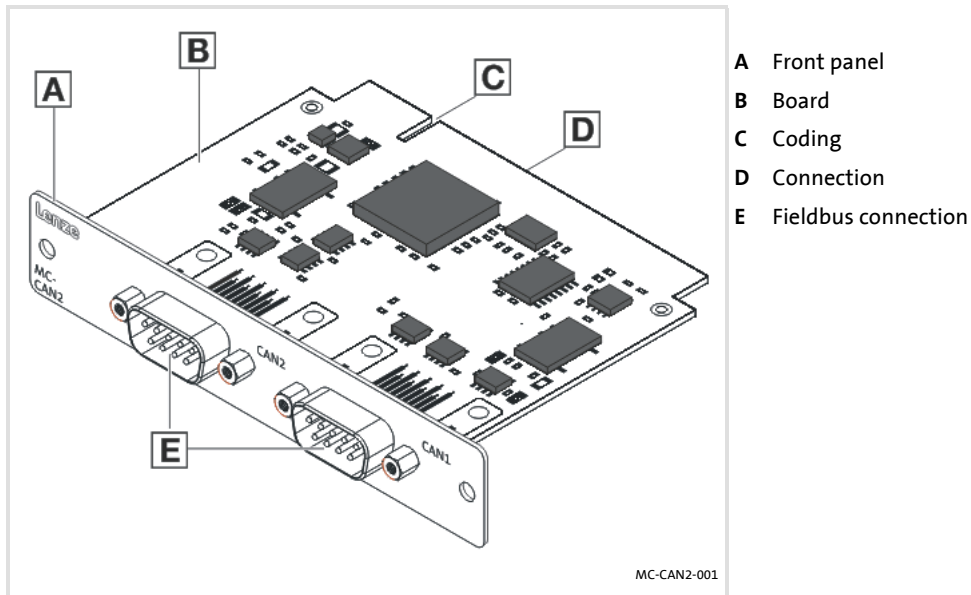
Standard device		Logic	Motion
Industrial PCs	EL x1xx PLC	●	-
	EL x8xx	●	●
	CS x8xx	●	●
	CPC x8xx	●	●
Servo Drives 9400	HighLine 1)	●	-
	HighLine CiA402	●	●
	PLC	●	-
Inverter Drives 8400	BaseLine	●	-
	StateLine	●	-
	HighLine	●	-
	TopLine	●	-
I/O system IP20	EPM-Txxx	●	-
I/O system 1000	EPM-Sxxx	●	-
Frequency inverter	8200 vector	●	-
ECS servo system (from firmware version 2.0)	ECSxE	●	-
	ECSXS (Speed & Torque)	●	-
	ECSxP (Posi & Shaft)	●	-
	ECSxM (Motion)	-	●
	ECSxA (Application)	●	-

1) with technology application (TA)

4.2 CANopen Hardware for your Industrial PC

Communication card MC-CAN2

The MC-CAN2 communication card is a plug-in card to connect an Industrial PC to a CAN fieldbus. It has two independent CAN bus connections.

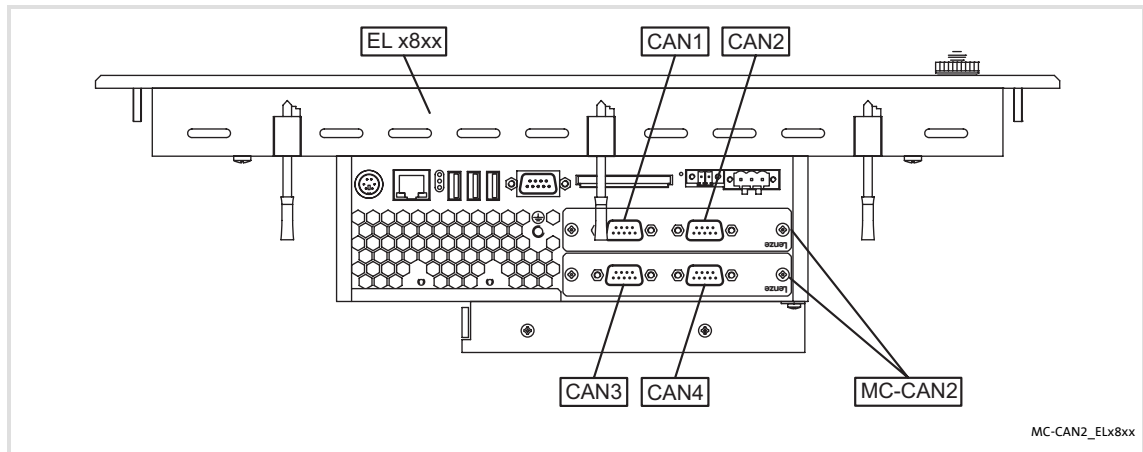


► [Technical data of the MC-CAN2 communication card](#) (21)

Possible applications

The MC-CAN2 communication card can be plugged into slot 1 and slot 2 of the Industrial PC. Your Industrial PC can have several CANopen communication cards.

Example: The EL x8xx Industrial PC with MC-CAN2 in slots 1 and 2



Legend

EL x8xx	Industrial PC of the EL x8xx series
CAN1 ... 4	CAN bus connections • CAN1: CANopen (Logic) or CANopen (Motion) • CAN2 ... 4: CANopen (Motion)
MC-CAN2	Communication card MC-CAN2

5 Technical data

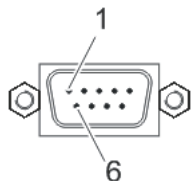
5.1 General data

Field	Values
Communication profile	CANopen (DS301, V4.02)
Standards	CAN, ISO 11898 / EN 50325-4
Network topology	Line, terminated at both ends with 120 Ω (e.g. terminated with Sub-D plug of type EWZ0046)
Max. number of nodes	127
Adjustable node addresses	1 ... 127 (adjustable for Lenze communication modules via DIP switches)
Baud rates [kbps]	• 10 • 20 • 50 • 125 • 250 • 500 • 1000
Parameter data	Max. 10 client and server SDO channels with 1 ... 8 bytes
Cycle time - Motion/CNC task	1 ... 16 ms
Max. number of drives/ms on the Motion bus	Max. 3 drives/ms
Signal propagation delay - drive → control → drive	4 cycles
Cross communication	Only possible with CANopen (Logic) In case of CANopen (Motion) the communication is executed centrally via the Industrial PC.
Number of DI + DO (bits/ms)	384 (max. 6 PDOs/ms on the Logic bus)
Cycle synchronisation with locked PLL (Jitter)	+/-10 μ s

5.2 Technical data of the MC-CAN2 communication card

Field	Values
Type within the network	Master or slave
Max. number of nodes	63
Max. baud rate [kbit/s]	1000
Bus length	See Bus cable length (□ 22)
Connection	SUB-D, 9-pole plug

CANopen bus connection (SUB-D, 9-pole plug)

View	Pin	Assignment	Description
	1	free	-
	2	LO	CAN-LOW
	3	CG	CAN-Ground
	4	free	-
	5	free	-
	6	CG	CAN-Ground
	7	HI	CAN-HIGH
	8	free	-
	9	free	-

5.3 Bus cable specification

We recommend to use CAN cables according to ISO 11898-2:

CAN cables according to ISO 11898-2	
Cable type	Paired cable with shield
Impedance	120 Ω (95 ... 140 Ω)
Cable resistance / cross-section	
Cable length $\leq$ 300 m:	$\leq$ 70 m Ω /m / 0.25 ... 0.34 mm ² (AWG22)
Cable length 301 ... 1000 m:	$\leq$ 40 m Ω /m / 0.5 mm ² (AWG20)
Signal propagation delay	$\leq$ 5 ns/m

5.4 Bus cable length



Note!

- It is absolutely necessary to comply with the permissible cable lengths.
- Observe the reduction of the total cable length due to the signal delay of the repeater. ▶ [Check use of repeater](#) (p. 24)
- If the total cable lengths of the nodes are different at the same baud rate, the smaller value must be used to determine the max. cable length.

5.4.1 Total cable length

The total cable length is also specified by the baud rate.

Baud rate [kbps]	Max. bus length [m]					
	9400 Servo Drives	Inverter Drives 8400	I/O-System IP20 (EPM-Txxx) CAN gateway	I/O-System 1000 (EPM-Sxxx) CANopen bus coupler	8200 vector frequency inverter	Servo System ECS
10	8075	-	5000	5000	7434	-
20	4012	-	2500	2500	3934	-
50	1575	1620	1000	1000	1534	1500
125	600	600	500	500	614	630
250	275	260	250	250	274	290
500	112	90	80	80	104	120
1000	12	5	25	25	9	25

5.4.2 Segment cable length

The segment cable length is determined by the used cable cross-section and the number of nodes. Repeaters divide the total cable length into segments. Without a repeater, the segment cable length corresponds to the total cable length.

Max. number of nodes per segment	Cable cross-section (interpolation is permissible)			
	0.25 mm ² (AWG 24)	0.50 mm ² (AWG 21)	0.75 mm ² (AWG 19)	1.00 mm ² (AWG 18)
2	240 m	430 m	650 m	940 m
5	230 m	420 m	640 m	920 m
10	230 m	410 m	620 m	900 m
20	210 m	390 m	580 m	850 m
32	200 m	360 m	550 m	800 m
63	170 m	310 m	470 m	690 m
100	150 m	270 m	410 m	600 m

Example: Selection help

Given:	
Total cable length to be implemented	200 m
Number of nodes	63

Results	
Max. possible baud rate	250 kbps (derived from table Total cable length (📖 22))
Required cable cross-section (interpolated)	0.30 mm ² (AWG23) (derived from table Segment cable length (📖 23))
Cable cross-section - standard CAN cable	0.34 mm ² (AWG22) ▶ Bus cable specification (📖 21)

5.4.3 Check use of repeater

Compare the values from the tables [Total cable length \(□ 22\)](#) and [Segment cable length \(□ 23\)](#).

- ▶ If the total segment cable length is shorter than the total cable length to be implemented, either repeaters must be used or the cable cross-section must be increased.
- ▶ If, due to the use of repeaters, the max. possible total cable length is reduced to a value smaller than the total cable length to be implemented, either the cable cross-section must be increased and the number of repeaters must be reduced or the baud rate must be reduced.
- ▶ The use of another repeater is recommended as ...
 - Service interface
Advantage: Trouble-free coupling during bus operation is possible.
 - Calibration interface
Advantage: The calibration/programming unit remains electrically isolated.

Example

Given	
Total cable length to be implemented	450 m
Number of nodes	32
Cable cross-section	0.50 mm ² (AWG 21)
Baud rate	125 kbit/s
Used repeater	Lenze repeater EMF2176IB
Reduction of the max. total cable length per repeater (EMF2176IB)	30 m

Results	
Max. possible total cable length	600 m (cp. table Total cable length (□ 22))
Max. segment cable length	360 m (cp. table Segment cable length (□ 23))
Comparison	The max. segment cable length is shorter than the total cable length to be implemented.
Conclusion	After the determined max. segment cable length of 360 m at the latest, a repeater must be used.

Results with 1 repeater	
Max. possible total cable length	570 m (Reduction of the Total cable length (□ 22) by 30 m)
Total segment cable length	720 m
Comparison	Both the possible total cable length and the segment cable length are longer than the total cable length to be implemented.
Conclusion	One repeater suffices to implement the total cable length of 450 m.

6 Planning the CANopen network

Before establishing a CANopen network, create a plan of your Logic bus and/or your Motion buses.

For this purpose, create an overview screen of the planned CANopen network with all field devices to be implemented. Start with the Industrial PC and arrange the other field devices below it. (see [Example of an overview screen](#) (□ 28)).

Provide the following data for each device:

Type	Type designation of the field device
Used CAN interface of the device	"Logic before Motion": - Always connect an existing Logic bus to the 1st CAN interface (CAN1). - Motion buses, however, can be connected to every CAN interface. ▶ CANopen (Logic) / CANopen (Motion) (□ 15)
Unambiguous CAN node address	- If system bus (CAN) devices are used, max. 63 nodes/node addresses are possible. - With CANopen-compliant devices, up to 127 nodes/node addresses are possible. Note: Do not use the node address 1, in order to avoid unintentional mistakes and conflicts with a device containing the factory adjustment.
Baud rate	- The baud rate applies to all nodes of the CANopen network. 50, 125, 250 and 500 kbit/s are supported by all device types of the system. - Observe the dependency between bus cable length and baud rate. ▶ Bus cable length (□ 22)
Master role of the device (NMT master/sync master)	- An NMT master sets itself and then the NMT slaves to the "Operational" state. In this state, process data can be communicated. Generally, there can be an optional number of NMT masters on one CANopen bus. - A <u>sync master</u> cyclically sends a sync telegram providing for an exactly simultaneous processing of process data and/or a simultaneous task start in all sync receivers. - Via CAN synchronisation you can influence the exact time of the following events in the field device: – Acceptance and transmission of sync-controlled PDOs – Starting time of the task of the application (only possible for Servo Drives 9400) - You only need to use CAN synchronisation on the Logic bus if an exact simultaneity in the range of milliseconds is of importance. A mere operating periphery (operator button, control lamps, etc.) does not require CAN synchronisation.
CAN objects and COB-IDs	- Plan your COB-IDs according to the CANopen DS301 communication profile. This convention is optimised for the communication with a central master device. ▶ COB-IDs acc. to DS301 (□ 26) - Up to 4 PDOs per device can be identified with this scheme. If you require more, e.g. for a modular I/O system with more than 8 modules, you can add them later. - You can easily assign the node during the bus diagnostics by means of the COB-IDs. - COB-ID = basic identifier + node address



Observe device-specific information on CAN configuration provided in the documentation of the field devices to be integrated.

COB-IDs acc. to DS301

Object		Direction		Basic identifier	
		from the drive	to the drive	Dec	Hex
NMT				0	0
Sync				128	80
Time Stamp				256	100
Emergency		●		128	80
PDO1 (process data channel 1)	TPDO1	●		384	180
	RPDO1		●	512	200
PDO2 (process data channel 2)	TPDO2	●		640	280
	RPDO2		●	768	300
PDO3 (process data channel 3)	TPDO3	●		896	380
	RPDO3		●	1024	400
PDO4 (process data channel 4)	TPDO4	●		1152	480
	RPDO4		●	1280	500
SDO (parameter data channel 1)		●		1408	580
			●	1536	600
NMT Error Control		●		1792	700



Note!

In system bus (CAN) devices, two SDO channels are permanently active, in CANopen devices, only one by default.

When using CANopen devices, activate a second SDO channel for accesses of the »Engineer« or »Global Drive Control«. Otherwise the communication with the device will be interfered if you go online with the »Engineer« or the »Global Drive Control«, while the IPC has also access.

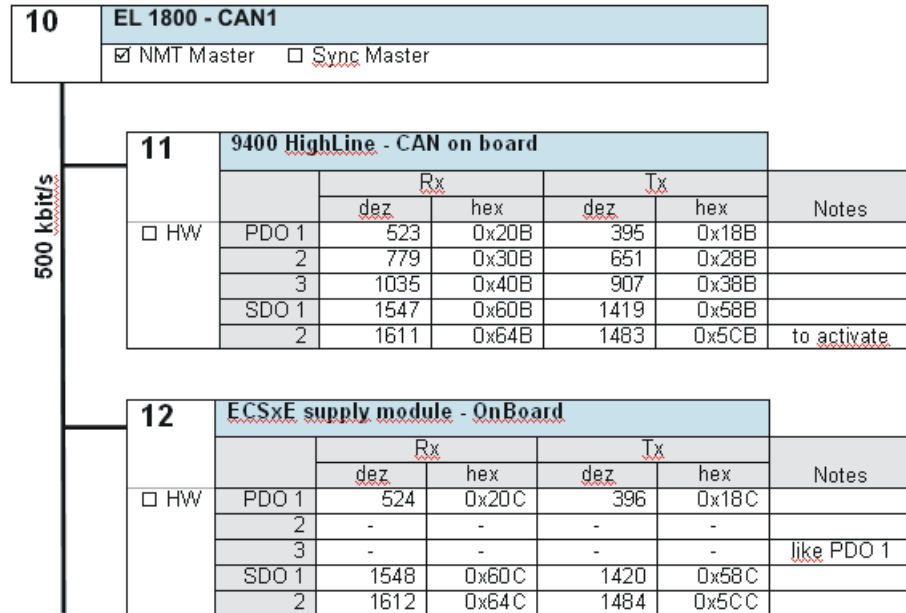
The COB-IDs for your CANopen network can be calculated according to the following formula:

$$\text{COB-ID} = \text{basic identifier} + \text{node address}$$

- ▶ [Basic identifier - 9400 Servo Drives](#) (📖 30)
- ▶ [Basic identifier - 8400 Inverter Drives](#) (📖 31)
- ▶ [Basic identifier - I/O system IP20 \(EPM-Txxx\)](#) (📖 32)
- ▶ [Basic identifier - I/O system 1000 \(EPM-Sxxx\)](#) (📖 33)
- ▶ [Basic identifier - 8200 vector with fieldbus function module CANopen E82ZAFUC0xx](#) (📖 34)
- ▶ [Basic identifier - ECS servo system](#) (📖 36)

6.1 Example of an overview screen

The illustration shows you an example of an overview screen for planning a CANopen network:



6.2 Device specifications of the field devices

When planning your CANopen network, consider the device specifications of the implemented field devices.

Overview of the device specifications when being operated subordinate to a control

	Servo Drives 9400	Inverter Drives 8400	I/O-System IP20 (EPM-Txxx)
CAN interface	on board and/or CANopen module	on board	on board
Available PDOs	4 Transmit (Tx) + 4 Receive (Rx)	3 Transmit (Tx) + 3 Receive (Rx)	10 Transmit (Tx) + 10 Receive (Rx)
Can unused PDOs be deactivated?	yes	yes	yes
Can PDO COB-IDs be freely selected?	yes	yes	yes
Can PDO transfer characteristics be adjusted?	yes	yes	yes
Available SDO channels	1 ex works (fixed), 9 further can be activated	2 ex works (fixed)	2 ex works (Only possible with V1.3 in CANopen mode.)
Can SDO COB-IDs be freely selected?	only for channel 2 ... 10	no	no

	I/O-System 1000 (EPM-Sxxx)	8200 vector frequency inverter	ECS servo system
CAN interface	on board	Fieldbus function module CANopen E82ZAFUC0xx	2 x CAN on board: - Terminal X4: Motion bus (CAN) - Terminal X14: System bus (CAN)
Available PDOs	10 Transmit (Tx) + 10 Receive (Rx)	3 Transmit (Tx) + 3 Receive (Rx)	1 Transmit (Tx) + 1 Receive (Rx)
Can unused PDOs be deactivated?	yes	yes	no
Can PDO COB-IDs be freely selected?	yes	yes	yes
Can PDO transfer characteristics be adjusted?	yes	yes	yes
Available SDO channels	1 ex works (fixed), 1 more can be activated	1 ex works (fixed), 1 more can be activated	2 ex works (fixed)
Can SDO COB-IDs be freely selected?	no	no	no

6.2.1 Special features of the Servo Drives 9400

- ▶ The parameter data channel 1 is always active.
- ▶ The *optional* parameter data channels 2 ... 10 can be activated via the subcodes of the codes Cxx372 and Cxx373.

SDO identifier	Code
CANopen SDO server Rx identifier	C00372: CAN on board
	C13372: Module in slot 1
	C14372: Module in slot 2
CANopen SDO server Tx identifier	C00373: CAN on board
	C13373: Module in slot 1
	C14373: Module in slot 2

- ▶ If bit 31 is set ($0x8nnnnnnn_{\text{hex}}$), the corresponding SDO server is deactivated.
- ▶ In order to change the COB-ID of a currently active parameter data channel, you have to first deactivate it and then activate it with a changed COB-ID. Both processes must be rendered effective by a "Reset Node" command via C00002.

Basic identifier - 9400 Servo Drives

The default setting of the basic identifier is as follows:

Object		Direction		Basic identifier	
		from the drive	to the drive	dec	hex
NMT				0	0
Sync 1)				128	80
Emergency		●		128	80
PDO1 (process data channel 1)	TPDO1	●		384	180
	RPDO1		●	512	200
PDO2 (process data channel 2)	TPDO2	●		640	280
	RPDO2		●	768	300
PDO3 (process data channel 3)	TPDO3	●		896	380
	RPDO3		●	1024	400
PDO4 (process data channel 4)	TPDO4	●		1152	480
	RPDO4		●	1280	500
SDO1 (parameter data channel 1)	TSDO1	●		1408	580
	RSDO1		●	1536	600
SDO2 ... 10 (parameter data channel 2 ... 10)	TSDOx	●		1472	5C0
	RSDOx		●	1600	640
Node guarding, heartbeat		●		1792	700

1) When creating the sync transmit/receive identifier manually, observe the use of the emergency telegram because of the same COB-ID.

6.2.2 Special features of the Inverter Drives 8400

Basic identifier - 8400 Inverter Drives

The default setting of the basic identifier is as follows:

Object		Direction		Basic identifier	
		from the drive	to the drive	dec	hex
NMT				0	0
Sync 1)				128	80
Emergency		●		128	80
PDO1 (process data channel 1)	TPDO1	●		384	180
	RPDO1		●	512	200
PDO2 (process data channel 2)	TPDO2	●		640	280
	RPDO2		●	641	281
PDO3 (process data channel 3)	TPDO3	●		768	300
	RPDO3		●	769	301
SDO1 (parameter data channel 1)	TSDO1	●		1408	580
	RSDO1		●	1536	600
SDO2 (parameter data channel 2)	TSDO2	●		1472	5C0
	RSDO2		●	1600	640
Heartbeat		●		1792	700
Boot-up 2)		●		1792	700

1) When creating the sync transmit/receive identifier manually, observe the use of the emergency telegram because of the same COB-ID.

2) When the boot-up identifier is set manually, observe the use of heartbeat because of the same COB-ID.

6.2.3 Special features for the I/O-System IP20 (EPM-Txxx)

- ▶ EPM T110 V1.2 has fixed parameter data channels, in CANopen mode as well.
- ▶ EPM T110 V1.3 has only one parameter data channel in CANopen mode.

Basic identifier - I/O system IP20 (EPM-Txxx)

The default setting of the basic identifier is as follows:

Object		Direction		Basic identifier	
		from the drive	to the drive	dec	hex
NMT				0	0
Sync 1)				128	80
Emergency		●		128	80
PDO1 (process data channel 1)	TPDO1	●		384	180
	RPDO1		●	512	200
PDO2 (process data channel 2)	TPDO2	●		640	280
	RPDO2		●	768	300
PDO3 (process data channel 3)	TPDO3	●		896	380
	RPDO3		●	1024	400
PDO4 (process data channel 4)	TPDO4	●		1152	480
	RPDO4		●	1280	500
PDO5 (process data channel 1)	TPDO5	●		1664	680
	RPDO5		●	1920	780
PDO6 (process data channel 2)	TPDO6	●		448	1C0
	RPDO6		●	576	240
PDO7 (process data channel 3)	TPDO7	●		704	2C0
	RPDO7		●	832	340
PDO8 (process data channel 4)	TPDO8	●		960	3C0
	RPDO8		●	1088	440
PDO9 (process data channel 1)	TPDO9	●		1216	4C0
	RPDO9		●	1344	540
PDO10 (process data channel 2)	TPDO10	●		1728	6C0
	RPDO10		●	1984	7C0
SDO1 (parameter data channel 1)	TSDO1	●		1408	580
	RSDO1		●	1536	600
Node guarding		●		1792	700

1) When creating the sync transmit/receive identifier manually, observe the use of the emergency telegram because of the same COB-ID.

6.2.4 Special features of the I/O-System 1000 (EPM-Sxxx)**Basic identifier - I/O system 1000 (EPM-Sxxx)**

The default setting of the basic identifier is as follows:

Object		Direction		Basic identifier	
		from the drive	to the drive	dec	hex
NMT				0	0
Sync 1)				128	80
Emergency		●		128	80
PDO1 (process data channel 1)	TPDO1	●		384	180
	RPDO1		●	512	200
PDO2 (process data channel 2)	TPDO2	●		640	280
	RPDO2		●	768	300
PDO3 (process data channel 3)	TPDO3	●		896	380
	RPDO3		●	1024	400
PDO4 (process data channel 4)	TPDO4	●		1152	480
	RPDO4		●	1280	500
PDO5 (process data channel 1)	TPDO5	●		1664	680
	RPDO5		●	1920	780
PDO6 (process data channel 2)	TPDO6	●		448	1C0
	RPDO6		●	576	240
PDO7 (process data channel 3)	TPDO7	●		704	2C0
	RPDO7		●	832	340
PDO8 (process data channel 4)	TPDO8	●		960	3C0
	RPDO8		●	1088	440
PDO9 (process data channel 1)	TPDO9	●		1216	4C0
	RPDO9		●	1344	540
PDO10 (process data channel 2)	TPDO10	●		1728	6C0
	RPDO10		●	1984	7C0
SDO1 (parameter data channel 1)	TSDO1	●		1408	580
	RSDO1		●	1536	600
Node guarding		●		1792	700

1) When creating the sync transmit/receive identifier manually, observe the use of the emergency telegram because of the same COB-ID.

6.2.5 Special features of the 8200 vector frequency inverter

Basic identifier - 8200 vector with fieldbus function module CANopen E82ZAFUC0xx

The default setting of the basic identifier is as follows:

Object		Direction		Basic identifier	
		from the drive	to the drive	dec	hex
NMT				0	0
Sync 1)				128	80
Emergency		●		128	80
PDO1 (process data channel 1)	TPDO1	●		384	180
	RPDO1		●	512	200
PDO2 (process data channel 2)	TPDO2	●		640	280
	RPDO2		●	768	300
PDO3 (process data channel 3)	TPDO3	●		896	380
	RPDO3		●	1024	400
SDO1 (parameter data channel 1)	TSDO1	●		1408	580
	RSDO1		●	1536	600
SDO2 (parameter data channel 2)	TSDO2	●		1472	5C0
	RSDO2		●	1600	640
Node guarding		●		1792	700

1) When creating the sync transmit/receive identifier manually, observe the use of the emergency telegram because of the same COB-ID.

6.2.6 Special features of the ECS servo system

Modules of the ECS servo system

Module	Type	Application software	CANopen		CAN interface
			Logic	Motion	
Power supply module	ECSxE	-	●	-	X4
Axis module	ECSXS	"Speed and Torque"	●	-	X4
	ECSxP	"Posi and Shaft"	●	-	X4
	ECSxM	"Motion"	-	●	X4
	ECSxA	"Application" (PLC)	●	-	X4 or X14
Capacitor module	ECSxK	-	No CAN communication		

Special features of the ECSxE power supply module

In case of ECSxE power supply modules one refers to CAN1 and CAN3. In fact, here there is one single process data channel (PDO). It is referred to as CAN1 if it operates in a sync-controlled manner and as CAN3 if it operates in a time or event-controlled manner. CAN1 and CAN3 cannot be used at the same time. The change-over of the transmission mode and therefore between CAN1 and CAN3 is effected in C0360. The ECSxE power supply module has no CAN2.

For historical reasons, the ECSxE power supply module has several subcodes for setting the PDO properties.

Code	Subcodes	Description
C0353	1, 3	Mode for ID creation (COB-ID) CAN-IN/OUT
C0354	1, 3	ID offset CAN-IN/OUT
C0355	1, 3	CAN-IN/OUT identifier (COB-IDs), read only
C0356	1, 3, 4	CAN3-IN/OUT time settings: cycle time, activation delay
C0357	1, 3	CAN-IN monitoring

The existence of these subcodes does not mean that they are independent PDOs. The respective subcode that is effective depends on the setting in C0360. The subcodes 2 with regard to CAN2 do not have any effect.

Basic identifier - ECS servo system

The default setting of the basic identifier is as follows:

Object		Direction		Basic identifier	
		from the drive	to the drive	dec	hex
NMT				0	0
Sync				128	80
PDO1 (process data channel 1)	TPDO1	●		384	180
	RPDO1		●	512	200
PDO2 (process data channel 2)	TPDO2	●		640	280
	RPDO2		●	641	281
PDO3 (process data channel 3)	TPDO3	●		768	300
	RPDO3		●	769	301
SDO1 (parameter data channel 1)	TSDO1	●		1408	580
	RSDO1		●	1536	600
SDO2 (parameter data channel 2)	TSDO2	●		1472	5C0
	RSDO2		●	1600	640
Heartbeat		●		1792	700
Boot-up 1)		●		1792	700

1) When the boot-up identifier is set manually, observe the use of heartbeat because of the same COB-ID.

Devices with two active parameter data channels (SDO) respond with a fieldbus scan in the address range 1 ... 127 with two node addresses each (with offset 64).

7 Preparing the field devices

7.1 Installing field devices

Install the field devices according to the data given in the device-specific mounting instructions.

Make sure that ...

- ▶ the CANopen installation complies with your overview screen.
- ▶ all devices are supported by the control technology system on the Logic bus and Motion bus.
- ▶ in case of devices with several CAN interfaces, the correct interfaces are connected to the fieldbus.
- ▶ a terminating resistor is connected to the first and last node.
- ▶ the fieldbus is not unintentionally interrupted in switchable CAN connectors.

7.2 Setting node addresses and the baud rate

Set the intended node address and baud rate at the field devices.

- ▶ The easiest way to do this is using the DIP switches (if provided at the device).
- ▶ Mark the devices the settings of which you have changed in your overview screen.
- ▶ Attach address labels to the devices.

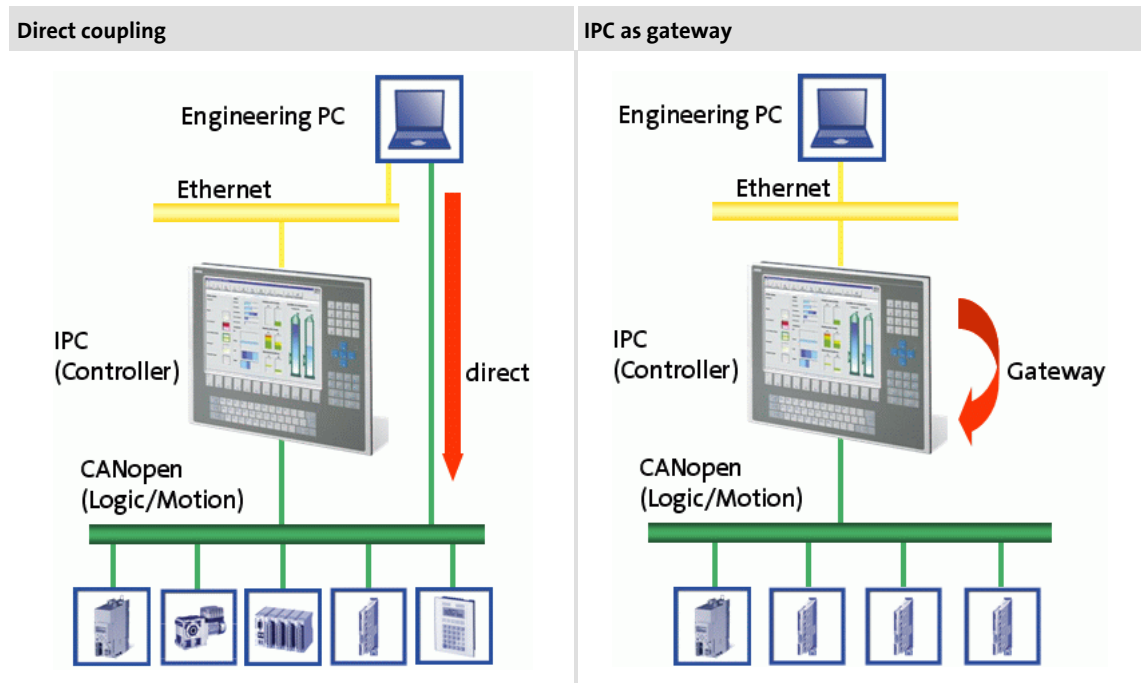


Note!

- Each node address must be unambiguous and may only be assigned once in the CANopen network.
- The baud rate must be set identically for all nodes.
- Observe the dependency between bus cable length and baud rate. ▶ [Bus cable length](#) (□ 22)

7.3 Connecting the Engineering PC to the Industrial PC

To commission the field devices, an online connection is required between the Engineering PC and the field device. To establish an online connection between an Engineering PC and a field device (like a controller), two ways are possible:



If the control is not started yet, directly connect the Engineering PC to the CANopen bus to commission the field devices. To activate the Engineering PC, use e.g. the USB system bus adapter (EMF2177IB). Then the download times are optimal and it is not necessary to commission the control first.

As soon as the control has been commissioned, no direct coupling should be used anymore since it may disturb the real-time capability of the bus. This especially applies to the bus line CANopen Motion bus. Here, the transmission of the sync telegram on time can be prevented so that an increased jitter on the fieldbus may be the result.

Moreover, each field device requires a second parameter data channel in case of an independent bus access by two masters. For some device types, the parameter data channel must be installed separately, e.g. in case of the 9400 Servo Drives.

As an option, some controllers can operate two independent CAN interfaces (e.g. ECS servo system). In this case, one interface can be used for the connection with the control system, the other for direct coupling of the Engineering PC. Thus, two buses are created which are physically independent. In this case, the real-time capability of the nodes at the Motion bus cannot be influenced even with direct coupling. However, the wiring effort increases.

The communication speed with the field devices, when being commissioned, mainly depends on whether the control is running or stopped. In the latter case, the total bandwidth of the bus is available for the gateway so that the speed advantage in case of direct coupling would only be marginal. Thus, the use of the IPC as gateway within the scope of the control technology should be clearly preferred.



Depending on the standard device and connection type used, detailed information about establishing a connection and "going online" can be found in the documentation listed below:

- (Software) manual/online help "PC-based Automation"
Industrial PC - *Parameter setting & Configuration*
- Software manual/online help "PC based automation"
IPC as gateway - *Parameter setting & Configuration*
- Software manual/online help »Global Drive Control«
IPC as gateway - *Parameterisation & Configuration*
- Software manual/online help L-force »Engineer«

8 Commissioning the CANopen Logic bus

This chapter provides information on commissioning the Lenze control system using the CANopen Logic bus.

Depending on the field devices used, the following Lenze engineering tools are required:

- ▶ »PLC Designer«
- ▶ »Engineer«
- ▶ »Global Drive Control« (GDC)



Tip!

For using other fieldbus systems, you may require further engineering software. More information can be found in the corresponding communication manuals.

8.1 Overview of the commissioning steps

Step	Activity	Lenze software to be used
1.	Creating a project folder (📖 41)	
2.	Commissioning of field devices (📖 42) ▶ Going online (📖 43) ▶ Commissioning the Servo Drives 9400 (📖 44) ▶ Commissioning of 8400 Inverter Drives (📖 47) ▶ Commissioning of I/O system IP20 (EPM-Txxx) (📖 50) ▶ Commissioning of I/O system 1000 (EPM-Sxxx) (📖 51) ▶ Commissioning of 8200 vector frequency inverter (📖 52) ▶ Commissioning of ECS devices (📖 53)	»Engineer« or »Global Drive Control« (depending on the used device)
3.	Creating a PLC program (📖 54)	»PLC Designer«
4.	Configuring the CAN master (📖 57)	»PLC Designer«
5.	Integrating field devices (slaves) into the PLC program (📖 59)	»PLC Designer«
6.	Setting of CAN parameters and CAN mapping (📖 60)	»PLC Designer«
7.	Creating a program code to control the device (📖 65)	»PLC Designer«
8.	Preparing the restart (📖 68)	»PLC Designer«

The individual commissioning steps are described in the following. Observe the given instructions step-by-step to commission your system.



More detailed information about how to work with the Lenze engineering tools can be found in the corresponding manuals and online helps.

8.2 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:

- ▶ Project data created in the »Engineer« or »GDC«
- ▶ The project file created in the »PLC Designer«
- ▶ Project data of other engineering tools



Tip!

Create a separate project folder for every PROFIBUS configuration for storing the project files.

8.3 Commissioning of field devices

Parameterise the Lenze field devices connected to the CANopen bus either with the »Engineer« or with the »GDC«, depending on the device.

The CANopen is exclusively configured with the »PLC Designer« ([book 54](#)).



Observe the information on commissioning provided in the documentation of the field devices.



Tip!

We recommend to commission each field device individually and then integrate them into the PLC program.



How to commission the field devices:

1. To commission the field devices, you have to go online.
 - ▶ [Going online](#) ([book 43](#))
2. Make the basic settings and CAN settings of the devices integrated in the CANopen network.
 - ▶ [Commissioning the Servo Drives 9400](#) ([book 44](#))
 - ▶ [Commissioning of 8400 Inverter Drives](#) ([book 47](#))
 - ▶ [Commissioning of I/O system IP20 \(EPM-Txxx\)](#) ([book 50](#))
 - ▶ [Commissioning of I/O system 1000 \(EPM-Sxxx\)](#) ([book 51](#))
 - ▶ [Commissioning of 8200 vector frequency inverter](#) ([book 52](#))
 - ▶ [Commissioning of ECS devices](#) ([book 53](#))

8.3.1 Going online

Standard device	Going online with	connection via
Industrial PC	»Global Drive Control« or »WebConfig«	Ethernet
Servo Drives 9400	»Engineer«	• IPC as gateway • Diagnostic adapter • Ethernet module E94AYCEN • CANopen module E94AYCCA • CAN device interface
Inverter Drives 8400	»Engineer«	• IPC as gateway • CAN device interface
I/O system IP20 (EPM-Txxx)	»Engineer« or »Global Drive Control«	• IPC as gateway • CAN device interface
I/O system 1000 (EPM-Sxxx)	»Engineer«	• IPC as gateway • CAN device interface
8200 vector frequency inverter	»Engineer« or »Global Drive Control«	• IPC as gateway • CANopen module E82ZAFUCxxx • CAN device interface 1)
ECS servo system (ECSxE/S/P/M/A)	»Global Drive Control«	• IPC as gateway • CANopen module EMF2178IB • CAN device interface 1)

1) CANopen only with system bus adapter EMF2177IB (if required, observe standard device specifications!)

**Note!**

When selecting the connection type, please observe the notes in chapter ["Connecting the Engineering PC to the Industrial PC"](#) (38).

We recommend to use the connection type "IPC as gateway".



Depending on the standard device and connection type used, detailed information about establishing a connection and "going online" can be found in the documentation listed below:

- (Software) manual/online help "PC-based Automation"
Industrial PC - *Parameter setting & Configuration*
- Software manual/online help "PC based automation"
IPC as gateway - *Parameter setting & Configuration*
- Software manual/online help »Global Drive Control«
IPC as gateway - *Parameter setting & Configuration*
- Software manual/online help L-force »Engineer«

8.3.2 Commissioning the Servo Drives 9400



How to commission the Servo Drives 9400:

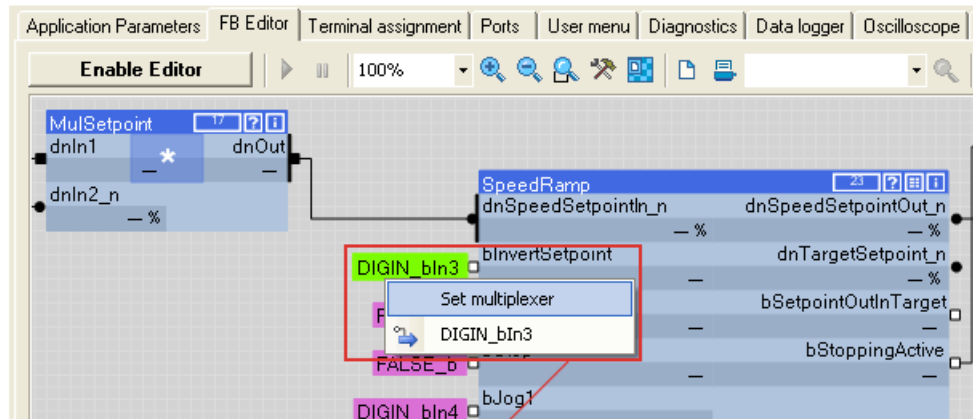
1. Start the »Engineer«.
2. Open and create an »Engineer« project.
 - Enter an axis consisting of Servo Drive 9400, device modules, application, and motor. Type and version of the planned device must comply with the real device.
3. Set the intended node address and baud rate.
 - Hardware setting via DIP switch *or*
 - in the parameter list via codes:
 - For "CAN on board" interface: C00350 and C00351
 - For CANopen module in slot 1: C13350 and C13351
 - For CANopen module in slot 2: C14350 and C14351

The codes can only be parameterised if the node address "0" and the baud rate "0" are set via the DIP switches (all DIP switches in OFF position).

A change of node address and baud rate gets only effective after a CAN reset node.

4. Set the sync phase position.
 - When the "CAN on board" interface is used, set C01122 = 120 µs.
 - When a CANopen module is used, set C01122 = 300 µs.
5. Go online with the Servo Drive 9400 and transfer the application to the device.
6. Make the motor rotate and, if required, parameterise the controller settings.

7. Connect the signal which are to be communicated with the IPC with ports via the **FB Editor** of the »Engineer«. If possible, use the predefined multiplexers and ports for this purpose.



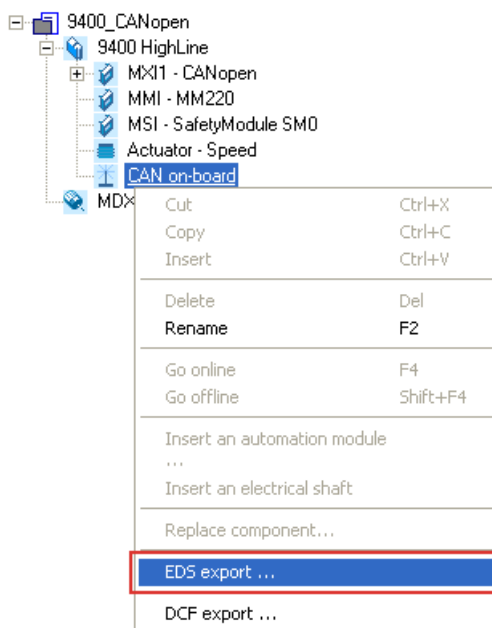
	Co...	S...	Name	Value
	3009	1	X Reverse with	Digital input 3
	3009	2	X Hold setpoint with	FALSE
	3009	3	X Stop drive with	FALSE
	3009	4	X Activ. fixed setpoint 1 with	Digital input 4
	3009	5	X Activ. fixed setpoint 2 with	FALSE
	3009	6	X Activ. fixed setpoint 4 with	FALSE
	3009	7	X Activ. fixed setpoint 8 with	FALSE
	3009	8	X Activ. ramp parameter 1 with	FALSE
	3009	9	X Activ. ramp parameter 2 with	FALSE
	3009	10	X Activ. ramp parameter 4 with	FALSE
	3009	11	X Activ. ramp parameter 8 with	FALSE

- If the possible settings of the predefined multiplexers are not sufficient for your purposes, activate the **FB Editor** and draw lines.
- Ports are the variables of the application which can be communicated outwards. In one port, several signals can be combined to a structure. If the predefined ports are not sufficient for your purposes, you can change the structure of the ports or define your own ports under the **Ports** tab. Before this, you must activate the application in the **FB Editor**.
- When the function block interconnection is completed and all required ports are defined, update the project.
- Retransfer the application to the Servo Drive 9400. Execute a "CAN Reset Node" on the correct interface via C00002. The device now expects its initialisation through the control. The green CAN-LED at the Servo Drive 9400 or CAN module is blinking.

8. Export the EDS file.

- To clearly identify the EDS file in the »PLC Designer« during import, clearly rename the device before export in the »Engineer«. (you can e.g. manually precede each device name in the »Engineer« with the project name.)
- The »PLC Designer« only displays the device name contained in the EDS file.

Highlight the CAN interface of the Servo Drive 9400 to be connected to the control in the project tree. Then select the "**EDS export ...**" command from the context menu.



9. Save the file to any folder, logically to your total project directory. You should not change the suggested file name. The »Engineer« reports that the file has been created.

You must reimport the EDS file after doing the following in the »Engineer« ...

- replacing the component,
- changing the module assembly,
- replacing the application,
- adding, deleting, renaming a port under the **Ports** tab or changing its application variables,
- adding, deleting function blocks that can be parameterised in the **FB Editor** or changing their code number range,
- adding, deleting, or changing user codes in the **FB Editor**.

If you reimport the EDS file, you should overwrite the existing EDS file. It is not reasonable to create several file versions with the same device name since you cannot distinguish them in the »PLC Designer«.

8.3.3 Commissioning of 8400 Inverter Drives



Note!

The basic identifiers for calculating the PDO COB-IDs do not comply with the DS301 profile. In this case, adapt the COB-IDs accordingly.

► [Basic identifier - 8400 Inverter Drives](#) (31)



How to commission the Inverter Drives 8400:

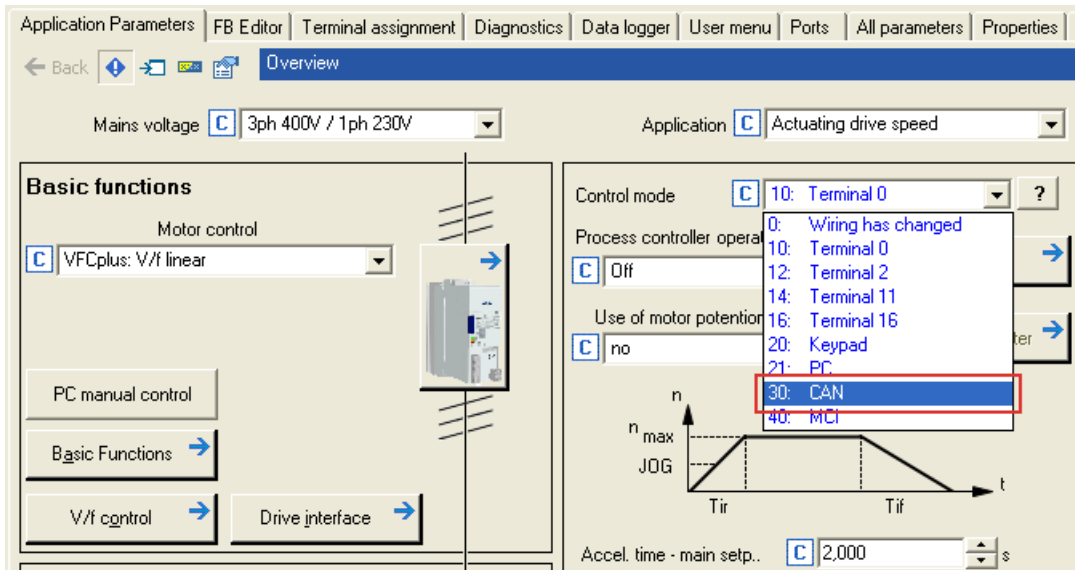
1. Start the »Engineer«.
2. Open or create an »Engineer« project.
 - Enter an axis consisting of Inverter Drive 8400, device modules, application, and motor. Type and version of the planned device must comply with the real device.
3. Set the intended node address and baud rate.
 - Hardware setting via DIP switch *or*
 - in the parameter list via the codes C00350 (node address) and C00351 (baud rate).

The codes can only be parameterised if the node address "0" and the baud rate "0" are set via the DIP switches (all DIP switches in OFF position).

A change of node address and baud rate gets only effective after a CAN reset node.

4. Go online with the Inverter Drive 8400 and transfer the application to the device.
5. Make the motor rotate and, if required, parameterise the controller settings.

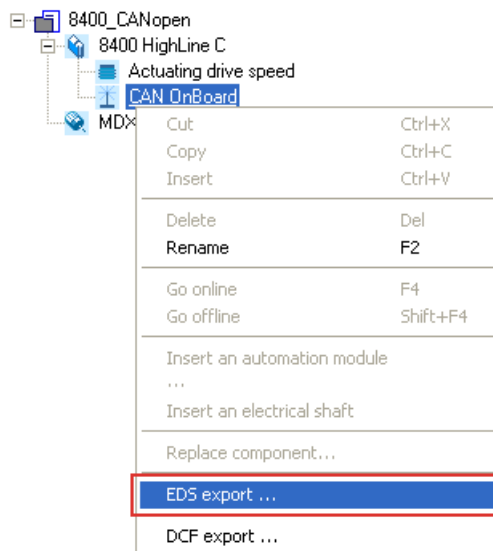
6. Set "CAN" under the **Application Parameters** tab in the **Control mode** field:



7. Export the EDS file.

- To clearly identify the EDS file in the »PLC Designer« during import, clearly rename the device before export in the »Engineer«. (you can e.g. manually precede each device name in the »Engineer« with the project name.)
- The »PLC Designer« only displays the device name contained in the EDS file.

Highlight the CAN interface of the Inverter Drive 8400 to be connected to the control in the project tree. Then select the "EDS export ..." command from the context menu.



8. Save the file to any folder, logically to your total project directory. You should not change the suggested file name. The »Engineer« reports that the file has been created.

You must reimport the EDS file after doing the following in the »Engineer« ...

- replacing the application,
- adding, deleting, renaming a port under the **Ports** tab or changing its application variables,
- adding, deleting function blocks that can be parameterised in the **FB Editor** or changing their code number range.

If you reimport the EDS file, you should overwrite the existing EDS file. It is not reasonable to create several file versions with the same device name since you cannot distinguish them in the »PLC Designer«.

8.3.4 Commissioning of I/O system IP20 (EPM-Txxx)



Note!

CAN settings must be made in the »PLC Designer« since the settings are transferred from the control to the I/O system.

The I/O system can be parameterised in three ways:



Parameter setting with the »Engineer«

1. Start the »Engineer«.
2. Open or create an »Engineer« project.
3. Insert the I/O system as component.
4. Go online with the I/O system via the CANopen bus.
5. Parameterise the I/O system.
6. Save the parameter set with mains failure protection via C16380.



Parameter setting with the »Global Drive Control«

1. Start the »GDC«.
2. Start the search for controllers at the CANopen bus and select the found I/O system.
Or first select the I/O system offline and then go online.
3. Parameterise the I/O system.
4. Save the parameter set with mains failure protection via CANopen Index 1010.

Parameter setting from the control

- In this case you can implement the I/O system immediately into the PLC program and enter the required values into the **Control Configuration** under the **Service Data Objects** tab.
- This is appropriate if you only want to change a few parameters compared to the Lenze default setting and know the values to be set.

8.3.5 Commissioning of I/O system 1000 (EPM-Sxxx)



Note!

CAN settings must be made in the »PLC Designer« since the settings are transferred from the control to the I/O system.

The I/O system can be parameterised in two ways:



Parameter setting with the »Engineer«

1. Start the »Engineer«.
2. Open or create an »Engineer« project.
3. Insert the I/O system as component.
4. Go online with the I/O system via the CANopen bus.
5. Parameterise the I/O system.
6. Save the parameter set with mains failure protection via CANopen Index 1010.

Parameter setting from the control

In this case you can implement the I/O system immediately into the PLC program and enter the required values into the **Control Configuration** under the **Service Data Objects** tab.

8.3.6 Commissioning of 8200 vector frequency inverter



Note!

- For CAN communication, plug on the fieldbus function module CANopen (E82ZAFUCxxx).
- CAN settings must be made in the »PLC Designer« since the settings are transferred from the control to the 8200 vector frequency inverter.



Parameter setting with the »Global Drive Control«

1. Start the »GDC«.
2. Start the search for controllers at the CANopen bus and select the found 8200 vector frequency inverter.

Or first select the suitable 8200 vector frequency inverter with fieldbus function module CANopen (E82ZAFUCxxx) offline and then go online.
3. Parameterise the inverter.
4. Save the parameter set with mains failure protection.
5. Go offline.

8.3.7 Commissioning of ECS devices



Note!

- In the ECS servo system, each device has a parameter memory which is only created once when the parameter setting is created via the »Global Drive Control«. This is the difference to CANopen devices where the parameterisation is written into the field device at each system start.
- The EMF2221IB card module serves to read parameters saved on an SD card or SmartMedia card into every ECS device.
- Only 1 receive-PDO and 1 transmit-PDO are available.
- The basic identifiers for calculating the PDO COB-IDs do not comply with the DS301 profile. In this case, adapt the COB-IDs accordingly.
 ▶ [Basic identifier - ECS servo system](#) (□ 31)
- The device has two permanent SDO channels.



Parameter setting with the »Global Drive Control«

1. Start the »GDC«.
2. Start the search for controllers at the CANopen bus and select the found ECS device.
 Or first select the suitable ECS device offline and then go online.
3. Parameterise the ECS device.
4. Save the parameter set with mains failure protection via C0003 = 1.

8.4 Creating a PLC program

The »PLC Designer« serves to illustrate the field device topology in the control configuration.



Tip!

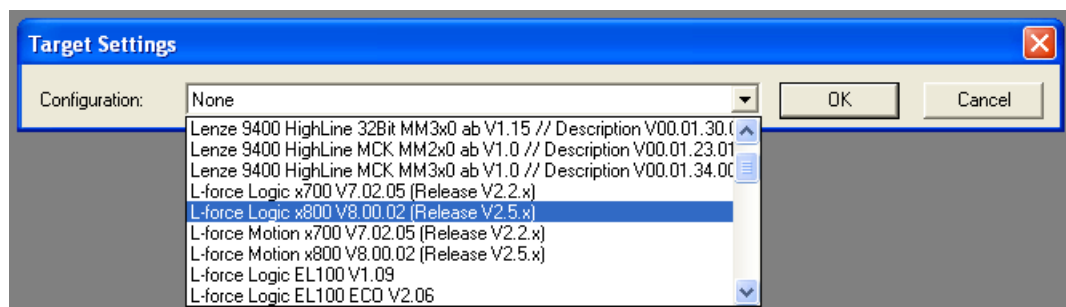
The »PLC Designer« serves to configure CANopen nodes and nodes on other fieldbus systems.

► [CANopen with PROFIBUS](#) (85)



How to create a PLC program 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:



Target system	Application with Industrial PC			
	EL x8xx	CS x8xx	CPC x8xx	EL 1xx PLC
L-force Logic x800 V8.xx.xx	●	●	●	-
L-force Logic EL1xx V1.xx	-	-	-	●

The target systems of the release 2.2/2.3 (L-force Logic/Motion x700) can also be used for device series **EL x8xx**, **CS x8xx**, and **CPC x8xx**.

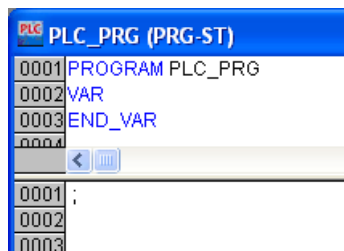
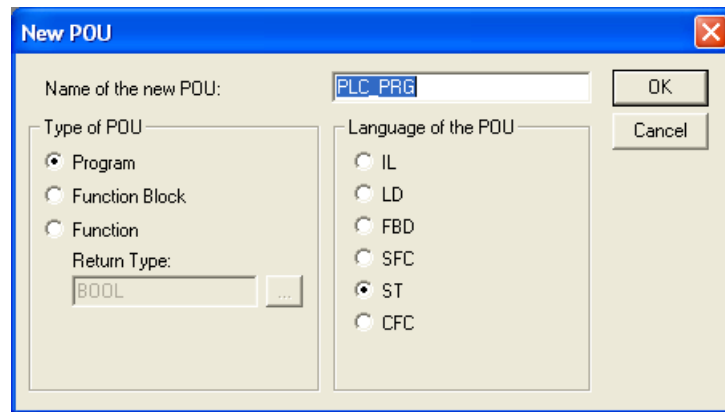
3. Confirm the configuration of the target system setting by clicking the **OK** button.

4. Create a block:



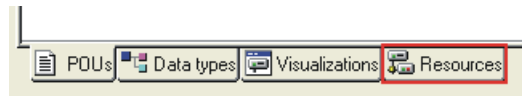
Note!

The block must contain at least one instruction to function properly.

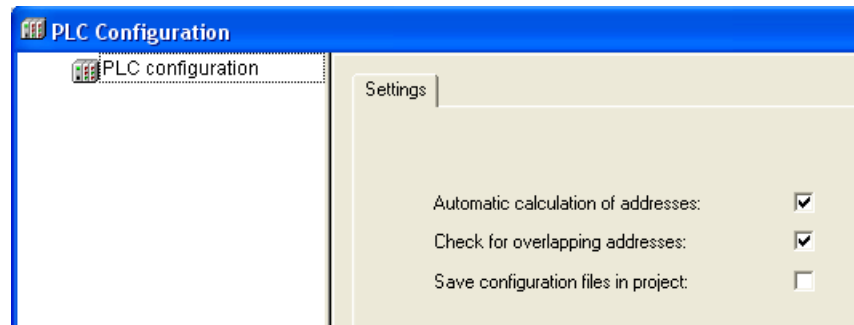


5. Create the control configuration:

- 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/online help of the »PLC Designer«.

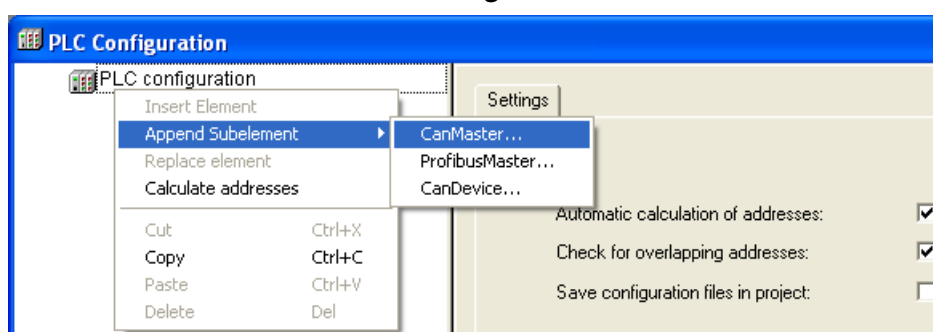
8.5 Configuring the CAN master

**Note!**

An Industrial PC in the CANopen network must be configured in the »PLC Designer« since the complete configuration of the IPC is written to the Servo Drives 9400 when the IPC is started. During this process, previous »Engineer« settings are overwritten.

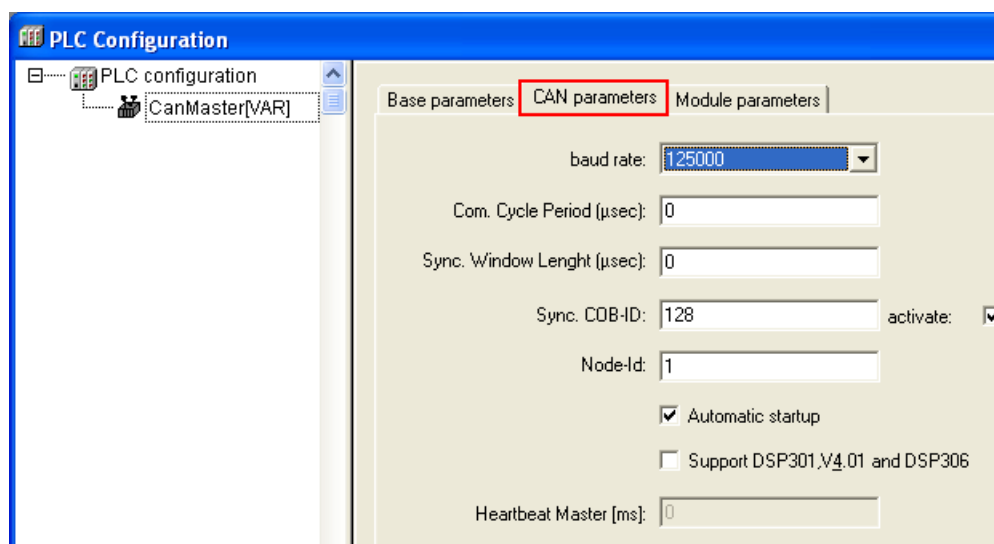


1. Add the bus interface to the PLC configuration:



The "CanMaster" subelement represents the CAN interface of the IPC which is connected to the Logic bus (typically CAN1).

2. Change to the **CAN parameters** tab:



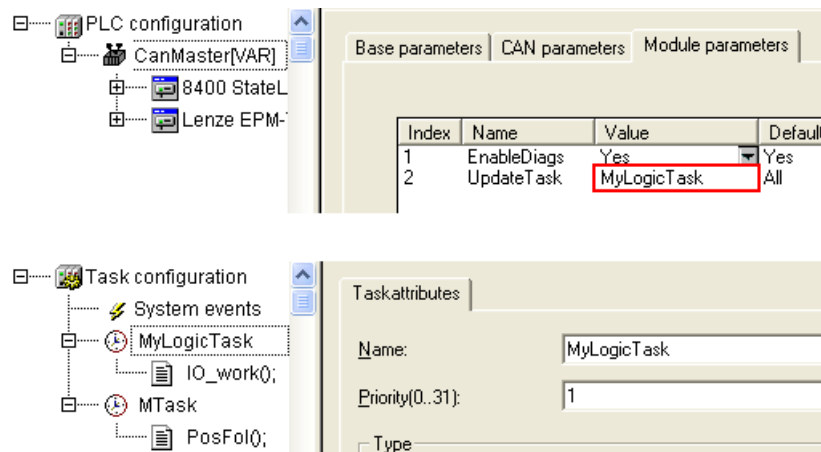


Note!

The baud rate set in »PLC Designer« overwrites the baud rate set for the field devices via »WebConfig«/»Engineer«/»Global Drive Control«.

In a CANopen network, set the same baud rate for all nodes.

- Set the planned baud rate and node address (node ID).
 - If you want to use the CAN synchronisation at the Logic bus, set a checkmark in the "Activate" input field. This is only required if at least one PDO with sync-controlled processing is used on the fieldbus and/or if you want to achieve that the applications run in synchronism in several controllers (slaves).
 - The sync cycle time can be set in the "Com. Cycle Period" input field. Enter an integer multiple of the cycle time of the assigned task (see 3., module parameters) as sync cycle time.
 - Activate the DSP support. This is required so that the IPC CANopen-conform Logic initialises field devices.
3. Specify the "UpdateTask" and its corresponding cycle time in the **Module parameters** tab by double-clicking the "Value" field:



- As "UpdateTask", enter the name of the task which accesses the CAN bus. If more than one task has access to the CAN bus, enter the task with the shortest cycle time.
 - All CAN telegrams are received and transmitted within the time base of this task. This also applies to the sync telegram which however is not transmitted more frequently than the set sync cycle time.
 - Communication from different tasks is not possible.
4. Your PLC program is now ready for the Logic field devices to be integrated. Save your PLC project.

8.6 Integrating field devices (slaves) into the PLC program



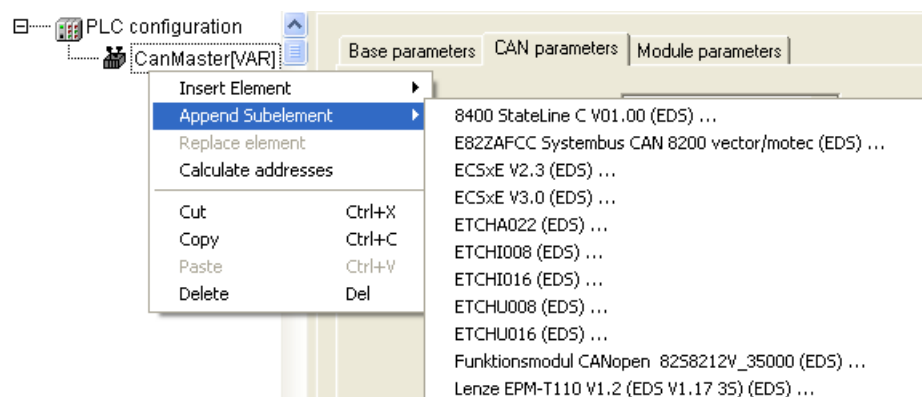
Note!

- EDS files regarding the **I/O system IP20 (EPM-Txxx)**, the **I/O system 1000 (EPM-Sxxx)**, **Frequency Inverters 8200 with CANopen fieldbus function modules (E82ZAFUCxxx)**, and other **ECS devices** can be found on the Internet in the Services & Downloads area of the Lenze homepage. Please be sure to always use the latest EDS file and also observe the software versions.
- EDS files for **Servo Drives 9400** and **Inverter Drives 8400** must be created via the »Engineer«.
- See:
 - [Commissioning the Servo Drives 9400](#) (44)
 - [Commissioning of 8400 Inverter Drives](#) (47)
- You can also integrate EDS files for devices of other manufacturers into the PLC program.



How to integrate field devices into the PLC program:

1. Import the device-specific EDS file in the »PLC Designer« via the menu item **Extras→Add configuration file**.
2. Add the corresponding field device as a "subelement" below the CAN master via the **Control configuration**.



The field device appears in the selection list with the same name as during the export of the EDS file in the »Engineer«, extended by the name of the interface and device type.

3. Give the entered element a name which ...
 - only contains the characters A ... Z, a ... z, 0 ... 9 and _ ,
 - does not start with a digit.

Example:



8.7 Setting of CAN parameters and CAN mapping



How to set CAN parameters and CAN mapping:

1. Change to the **CAN parameters** tab of the corresponding slave:

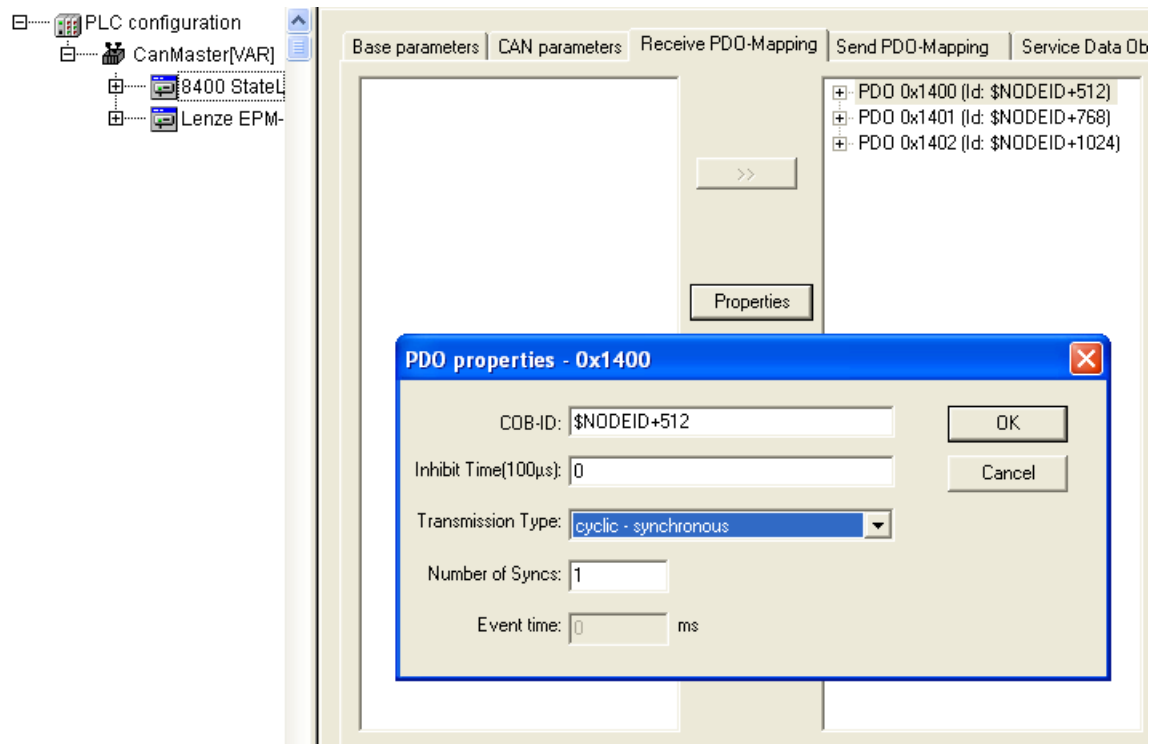
- Set the CAN node address and baud rate in accordance with the settings of the field devices.
- Set node guarding, emergency telegram, and communication cycle if you need them for your application.
- CAN Logic bus slaves provide the monitoring function of the communication cycle (area: "Communication Cycle"). The function can be switched on/off in the »PLC Designer« interface. If no sync telegram is received within the set monitoring time, the slave triggers an error message.



Note!

Switching on/off the monitoring function of the communication cycle in the »PLC Designer« is transmitted incorrectly to the slave or not at all. The last active state is kept in the slave (once active, always active).

2. Change to the **Receive PDO-Mapping / Send PDO-Mapping** tab:



- Here, configure the PDOs the field device is to receive and transmit. Define the transmission properties.
- The ports of the application are on the left, the available PDOs of the used interface are on the right. Define suitable transmission properties for the PDOs to be filled. Remove the PDOs which cannot be transmitted.
- Mapped variables in the PDOs also appear in the control configuration tree.
- We recommend to set the "cyclic - synchronous" transmission type and to specify the number of syncs at which the PDOs are to be sent.
- The settings of the event time via the "PDO properties - ..." dialog are not evaluated.



Note!

If you operate a fieldbus without CAN synchronisation, please observe the following:

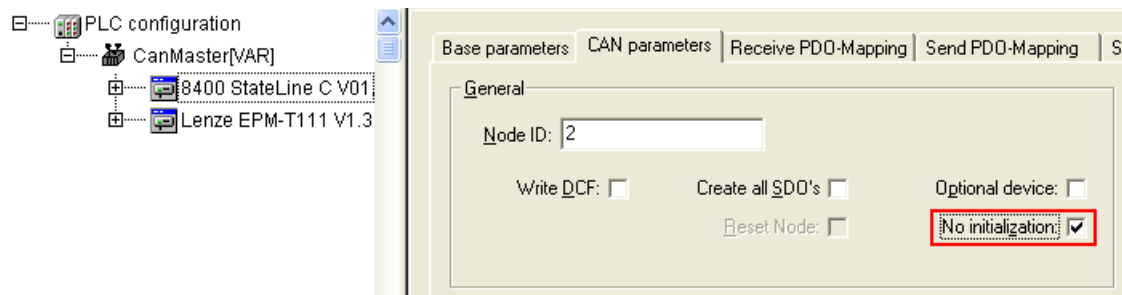
- The control system always transmits asynchronous PDOs from an unsolicited task in an event-controlled way. To achieve a time-controlled transmission of asynchronous PDOs by the control system, you must assign the CAN master to a cyclic task.
- The control does not support any monitoring times for asynchronous receive-PDOs. This is only possible with field devices.

Cross communication between the slaves

When the CAN bus is used for control, a cross communication between the slaves is possible.

For this purpose, you must configure the CAN communication and the PDO mapping in the »Engineer« or in the »Global Drive Control« and write it into the controller.

Since the mapping for the cross communication between the slaves is not available in the control configuration, you must set the "Not initialise" option in the »PLC Designer« in the **CAN parameters** tab:



Thus, the CAN and mapping settings in the slave drives will not be overwritten by the control when the PLC program starts.



Note!

In addition to the cross communication between the slaves, the communication to the master in the »Engineer« or in the »Global Drive Control« must be configured as well. Also this part of the PDO mapping will not be written anymore when the "Not initialise" option is active.

8.7.1 Special features of the 9400 Servo Drives

The mapping required for cross-communication between the control and the slaves must be created in the »PLC Designer«. When the PLC is started, the complete configuration / PDO mapping is written to the Servo Drives 9400. During this process, previous mapping entries from the »Engineer« are overwritten.

Please also observe the notes regarding [Cross communication between the slaves](#) (p. 62).

8.7.2 Special features of the 8400 Inverter Drives

The mapping required for cross-communication between the control and the slaves must be created in the »PLC Designer«. When the PLC is started, the complete configuration / PDO mapping is written to the Inverter Drives 9400. During this process, previous mapping entries from the »Engineer« are overwritten.

Please also observe the notes regarding [Cross communication between the slaves](#) (p. 62).

8.7.3 Special features of the I/O modules IP20 "1×counter/16×digital input" and "SSI interface"

- ▶ The I/O modules support the system bus (CAN) and CANopen operating modes. In connection with the control technology, set CANopen.
- ▶ The module 1×counter/16×digital input always assigns the next to last and the SSI interface module always the last of the PDOs used.
- ▶ The modules cannot be assigned to PDO1 and PDO2. Thus, only eight of these modules can be maximally used in a system.
- ▶ The modules assign a whole PDO (8 bytes) each.

8.7.4 Special features of the 8200 vector frequency inverter

- ▶ The standard mapping corresponds to the Lenze setting of the 8200 vector frequency inverter.
- ▶ In the standard setting, sync-controlled PDOs are used.

8.7.5 Special features of the ECS servo system



1. Set the process data transfer in the ECS device.

Process data channel CAN3 (event-controlled/cyclic without sync telegram):

- C0360 = 0
- Use individual addressing from C0354: C0353/3 = 1
- COB-ID of the receive-PDO1: C0354/5 = 128 + node address
- COB-ID of the transmit-PDO1: C0354/6 = node address
- Transmit-cycle time: C0356/3

Process data channel CAN1 (synchronised/cyclic with sync telegram):

- C0360 = 1
- Use addressing from C0350: C0353/1 = 0

2. Trigger a "CAN reset node": C0358 = 1
3. Save the parameter set: C0003 = 1

8.8 Creating a program code to control the device



Note!

All program blocks which are operated with an SDO communication must be called up in a Logic task. Otherwise, jobs will get lost.



1. Create the program code to control the field device. The device must be used in the program code so that the SDO initialisation can take place.

If you insert more devices in the control configuration, the addresses of the existing % variables may change. Thus, do not use any % variables directly in the program code but allocate separate names:



2. Compile the program and transfer it to the control system.

8.8.1 Special features of the Servo Drives 9400

- When the program starts, the control initialises the Servo Drive 9400. The controller changes to the "Operational" state.

8.8.2 Special features of the Inverter Drives 8400

- When the program starts, the control initialises the Inverter Drive 8400. The frequency inverter changes to the "Operational" state.

8.8.3 Special features of the I/O system IP20 (EPM-Txxx)

- ▶ The I/O modules support the system bus (CAN) and CANopen operating modes. In connection with the control technology, set CANopen.
- ▶ When the program starts, the control initialises the I/O system. It changes to the "Operational" state.
- ▶ While the control initialises the I/O system, »Engineer« or »Global Drive Control« must not be online on the same SDO channel.
- ▶ Operate the I/O system in the "CANopen" mode.
If you operate it in the "Lenze system bus" mode, specify in the program code that a "reset node" is transmitted after initialisation. Do not use the SDO "reset node" under the **Service Data Objects** tab as the control would try endlessly to initialise the I/O system.

8.8.4 Special features of the I/O system 1000 (EPM-Sxxx)

- ▶ When the program starts, the control initialises the I/O system. It changes to the "Operational" state.
- ▶ When the control initialises the I/O system, the »Engineer« must not be online on the same SDO channel.
- ▶ Set the "Reset Node" command under the "Service Data Objects" tab.

8.8.5 Special features of the 8200 vector frequency inverter

- ▶ When the program is started, the control initialises the 8200 vector frequency inverter. The fieldbus function module CANopen (E82ZAFUCxxx) connected to the 8200 vector frequency inverter changes to the "Operational" state. The lower green LED is lit constantly.
- ▶ While the control initialises the 8200 vector frequency inverter, the »Global Drive Control« must not be online on the same SDO channel.

8.8.6 Special features of the ECS servo system

- ▶ When the program is started, the control initialises the ECS device:
 - The control transmits, as common for CANopen, the CAN settings. However, the device as a Lenze system bus device does not accept these settings.
 - The control transmits the SDOs of the **Service Data Objects** tab if you changed anything there.
 - The control transmits an "NMT Start Remote Node" telegram. The device changes to the "Operational" state.
- ▶ While the control initialises the ECS device, the »Global Drive Control« must not be online on the same SDO channel.
- ▶ The ECSxE and ECSxM devices support toggle bit monitoring, which can cause the "TRIP CE4" error.
 - You can either program that the control cyclically changes the state of bit 0 in each transmitted PDO. The first transmitted control word must be 0 after the initialisation phase is completed, or
 - deactivate the toggle bit monitoring function in the device:
for ECSxE with C0595 = 3, for ECSxM with C3160 = 3.

8.9 Preparing the restart

In the control technology system you can use the control to transmit the entire parameter setting via SDO initialisation to the field devices when the machine is switched on.

According to DS301, the control always initialises the CAN parameters of the field devices. Moreover, it can initialise further parameters. The values for this must be stored in the **control configuration** under the **Service Data Objects** tab.

Usually, the control only transmits those SDO projects for which you have stored another value than the standard value. The control does not compare these values with the existing values in the field device. Thus, not all parameters changed there may be set correctly.

If you want that a factory adjustment is carried out in the field device before SDO initialisation, go to the **Parameter** tab and set a checkmark at "**Reset Node**".



Note!

When a node is reset, the parameter setting in the field device which you have made with the »Engineer« or the »Global Drive Control« gets lost. For this reason you have to transmit all parameter values manually to the **Service Data Objects** tab. This only makes sense when commissioning is completed and all parameters are optimised. If you change something afterwards via the »Engineer« or the »Global Drive Control«, you have to maintain it in the PLC program.

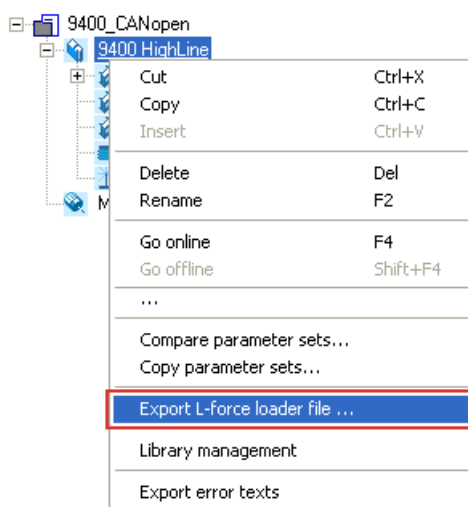
The **Service Data Objects** tab contains the codes which are written in the EDS file. The EDS file contains all write codes.

8.9.1 Special features of the Servo Drives 9400

Servo Drives 9400 are not purely parameterisable devices. They require an application download, where several files are transmitted to the memory module.

To operate a Servo Drive 9400,

- ▶ plug on the memory module.
- ▶ transmit the application using the »Engineer«. For this, you must keep the original »Engineer« project.
- ▶ transmit the application using the L-force »Loader«. For this, you must export and keep the required files from the »Engineer« project:



8.9.2 Special features of the Inverter Drives 8400

Inverter Drives 8400 are purely parameterisable devices.

To operate an Inverter Drive 8400, you can transmit the application using the »Engineer«. For this, you must keep the original »Engineer« project.

8.9.3 Special features of the I/O-System IP20 (EPM-Txxx)



Note!

For the SDO initialisation of the I/O system, please observe that ...

- a factory adjustment sets the I/O system to the system bus mode.
 - In connection with control technology, set CANopen.
- when being in the system bus mode, the I/O system only accepts changed CAN settings after a "reset node" has been executed.
- the control does not transmit a "reset node" after the SDO initialisation, unless you have programmed it accordingly.

Thus the following strategies are possible for the I/O system restart:

► Automatic:

You want the control to automatically initialise the I/O system after the device has been replaced.

- Operate the I/O system in CANopen mode.
- Enter all required parameter values in the **Control Configuration** under the **Service Data Objects** tab.
- Go to the **Parameters** tab and do not set a checkmark at "**Reset Node**" in order that the control does not execute a factory adjustment.
- Go to the **Parameters** tab and set a checkmark at **Create all SDOs** in order that the control initialises all parameters.
- After changing the I/O system:
Set the node address and baud rate at the coding switch for the CANopen mode and then start the control.

► Using the »Engineer« or the »Global Drive Control«:

1. You have implemented the I/O system successfully into the PLC program.
2. You have parameterised some codes of the I/O system using the »Engineer« or the »Global Drive Control«.
3. You do not want to transmit the parameter setting to the **control configuration** now, since you can assume that after a possible device replacement a project planning tool will be available.
 - Operate the I/O system in CANopen mode.
 - Go to the **Parameters** tab and do not set a checkmark at "**Reset Node**" in order that the control does not execute a factory adjustment.
 - Keep the »Engineer« project or the »Global Drive Control« file at the machine.
 - After changing the I/O system:
Set the node address and baud rate at the coding switch for the CANopen mode and transmit the archived parameter setting to the I/O system. Then start the control.

► **Automatic with factory adjustment:**

With an automatic restart, a factory adjustment is executed.

- Enter all required parameter values in the **Control Configuration** under the **Service Data Objects** tab.
- Set the checkmark at "**Reset Node**" in order that the control executes the factory adjustment. This causes the I/O system to change to the "Lenze System Bus" mode and remains there.
- Program a "Reset Node" in your program code. Do not use the SDO "Reset Node" under the **Service Data Objects** tab. Otherwise the initialisation will be repeated endlessly.
- After changing the I/O system:
Set the node address and baud rate at the coding switch for the CANopen mode and then start the control.

8.9.4 Special features of the I/O-System 1000 (EPM-Sxxx)

There are different viable strategies for the I/O system restart:

► **Automatic:**

You want the control to automatically initialise the I/O system after the device has been replaced.

- Enter all required parameter values in the **Control Configuration** under the **Service Data Objects** tab.
- Go to the **Parameters** tab and do not set a checkmark at "**Reset Node**" in order that the control does not execute a factory adjustment.
- Go to the **Parameters** tab and set a checkmark at **Create all SDOs** in order that the control initialises all parameters.
- After changing the I/O system:
Set the node address and baud rate at the coding switch and then start the control.

► **Using the »Engineer«:**

1. You have implemented the I/O system successfully into the PLC program.
2. You have parameterised some CANopen indexes of the I/O system using the »Engineer«.
3. You do not want to transmit the parameter setting to the **control configuration** now, since you can assume that after a possible device replacement a project planning tool will be available.
 - Operate the I/O system in CANopen mode.
 - Go to the **Parameters** tab and do not set a checkmark at "**Reset Node**" in order that the control does not execute a factory adjustment.
 - Save the »Engineer« project near the machine.
 - After changing the I/O system:
Set the node address and baud rate at the coding switch and transmit the archived parameter setting to the I/O system. Then start the control.

► **Automatic with factory adjustment:**

With an automatic restart, a factory adjustment is executed.

- Enter all required parameter values in the **Control Configuration** under the **Service Data Objects** tab.
- Set a checkmark at **Reset Node** for the control to perform the factory adjustment.
- Set the "Reset Node" command under the "**Service Data Objects**" tab.
- After changing the I/O system:
Set the node address and baud rate at the coding switch and then start the control.

8.9.5 Special features of the ECS servo system

- In the ECS servo system, each device has a parameter memory which is only created once when the parameter setting is created via the »Global Drive Control«. This is the difference to CANopen devices where the parameterisation is written into the field device at each system start.
- The EMF2221B card module serves to read parameters saved on an SD card or SmartMedia card into every ECS device.
- When using ECS devices, you cannot set all codes/parameters via the **Service Data Objects** tab as the EDS files do not contain all codes/parameters.

9 Commissioning the CANopen Motion bus

This chapter provides information on commissioning the Lenze control system using the CANopen Motion bus.

Depending on the field devices used, the following Lenze engineering tools are required:

- ▶ »PLC Designer«
- ▶ »Engineer«
- ▶ »Global Drive Control« (GDC)



Tip!

For using other fieldbus systems, you may require further engineering software. More information can be found in the corresponding communication manuals.

9.1 Overview of the commissioning steps

Step	Activity	Lenze software to be used
1.	Commissioning of field devices (📖 74)	»Engineer« or »Global Drive Control« (depending on the used device)
2.	Creating a PLC program (📖 75)	»PLC Designer«
3.	Creating a Motion task (📖 78)	»PLC Designer«
4.	Creating a control configuration (📖 79)	»PLC Designer«
5.	Creating a program code to control the Motion drives (📖 82)	»PLC Designer«
6.	Preparing the restart (📖 82)	»PLC Designer«
7.	Optimisation of signal propagation delays (for HighLine CiA402 only) (📖 83)	»PLC Designer«

9.2 Commissioning of field devices

Not all parameters required for operating a Motion drive are set automatically via the control.

Set the following parameters manually via the »Engineer« or the »Global Drive Control«:

- ▶ Servo Drives 9400 HighLine CiA402:
 - Homing mode (C02640, to be set machine-dependent)
 - Touch-probe interface (to be set machine-dependent)
 - The behaviour is automatically set via the control after the home position is detected.
 - Sync phase position:
 - When the "CAN on board" interface is used, set C01122 = 120 µs.
 - When a CANopen module is used, set C01122 = 300 µs.
- ▶ ECSxM axis module:
 - Touch-probe source (C0428, to be set machine-dependent)
 - Touch-probe dead time compensation (C0429, to be set machine-dependent)
 - Touch-probe edge (C0431, to be set machine-dependent)
 - Homing mode (C3010, to be set machine-dependent)
 - Synchronisation phase (C01122 = 0.46 ms = (Lenze setting))
 - Control of a possibly existing 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 the bit 2 in this parameter (disable stop => does not apply the brake in standstill).

9.3 Creating a PLC program

The »PLC Designer« serves to illustrate the field device topology in the control configuration.



Tip!

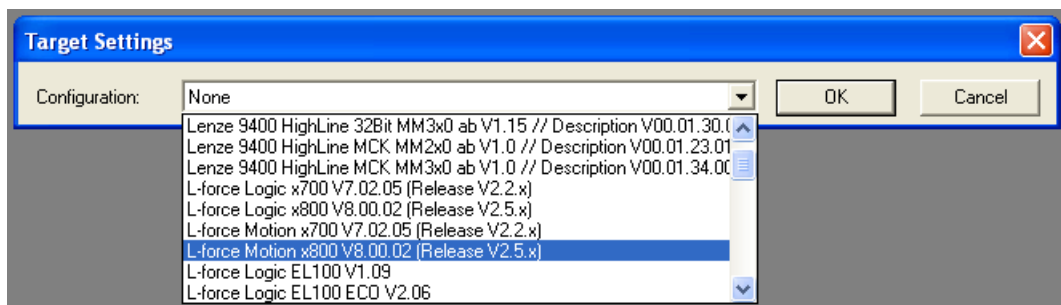
The »PLC Designer« serves to configure CANopen nodes and nodes on other fieldbus systems.

► [CANopen with PROFIBUS](#) (85)



How to create a PLC program in the »PLC Designer«:

1. Create a new »PLC Designer« project:
 - Menu command: **File→New**
2. In the *Target settings* dialog box, select the **L-force Motion x800 V8.xx.xx** target system:



The target systems of the release 2.2/2.3 (L-force Logic/Motion x700) can also be used for device series **EL x8xx**, **CS x8xx**, and **CPC x8xx**.

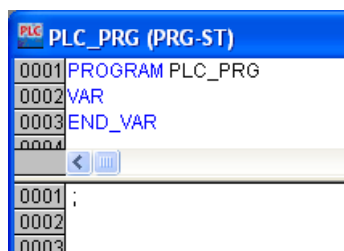
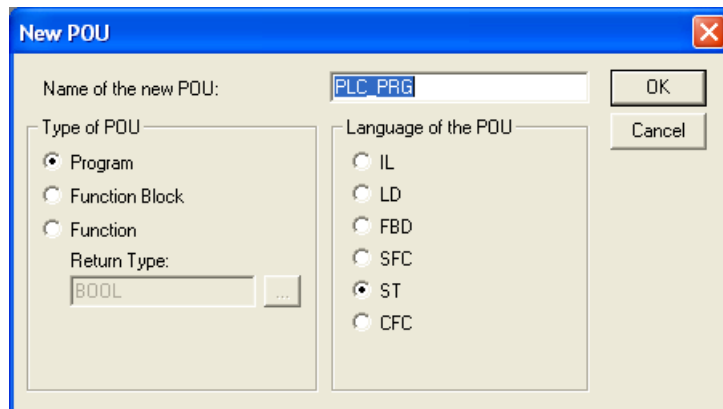
3. Confirm the configuration of the target system setting by clicking the **OK** button.

4. Create a block:



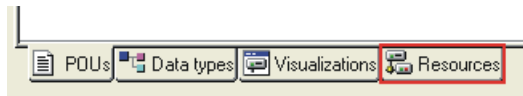
Note!

The block must contain at least one instruction to function properly.

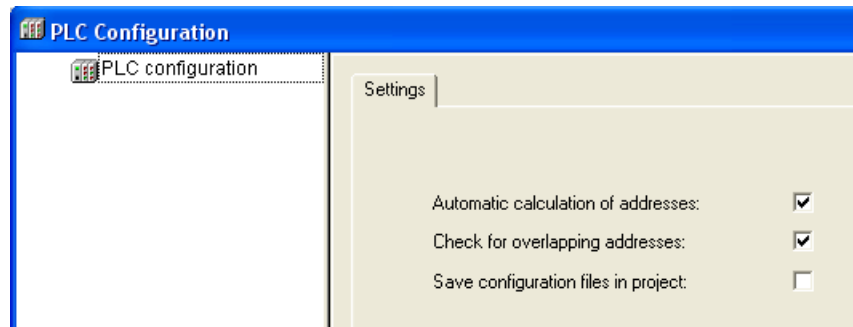


5. Create the control configuration:

- 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 topic can be found in the documentation/online help of the »PLC Designer«.

9.4 Creating a Motion task



How to create a Motion task:

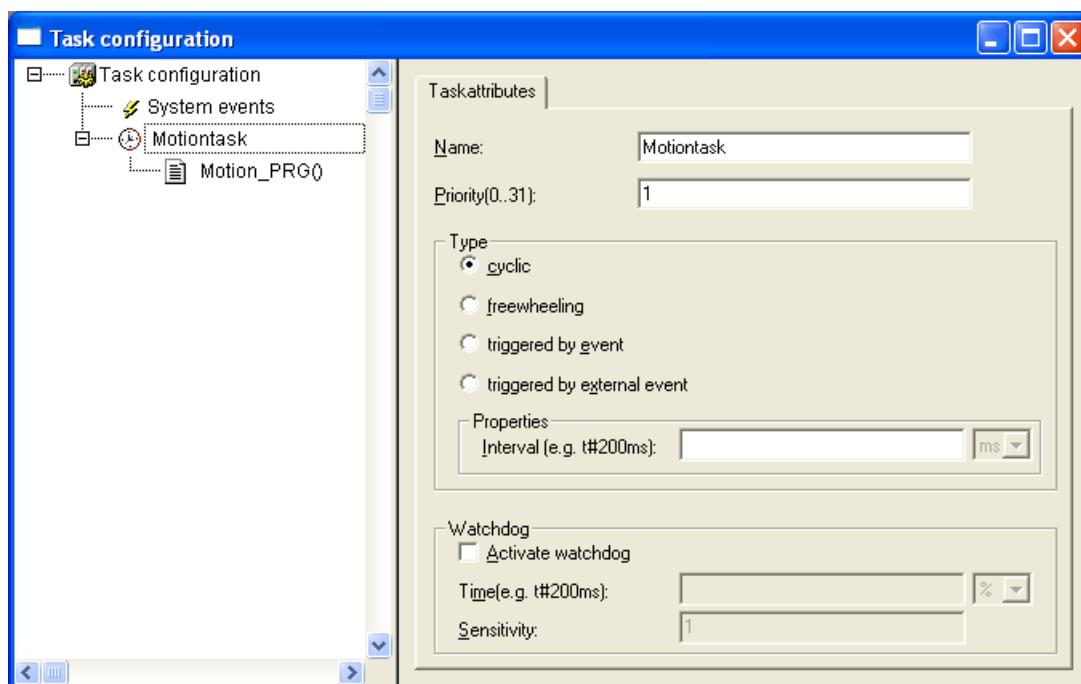
1. Go to the »PLC Designer« and change to **Task Configuration**.
2. Insert a new task and assign an appropriate task name (e.g. "Motion task")
 - in the "**Name**" input field and
 - in the task configurator tree (left window). A mouse-click on "NewTask" will activate the name for editing.
3. Enter an appropriate cycle time in milliseconds in the "Type" area under "Features" in the "**Interval**" input field.

The cycle time to be entered depends on the number of Motion axes and the runtime of the PLC application. In case of a small PLC application, the minimum cycle time (T_{cycl}) is determined by the number of Motion axes due to the transfer rate of the CANopen bus.

$$T_{cycl} [\text{ms}] = \text{number of Motion axes} / 3$$

4. Add a program call to the task (e.g. "Motion_PRG").

The following task configuration is the result:



5. Compile the new project completely.

Menu bar: **Project**→**Compile all**

9.5 Creating a control configuration

**How to create a control configuration:**

1. Go to »PLC Designer« and change to **control configuration**.
2. Add the "BusInterface_CAN" subelement to the "PLC Configuration" basic node.
The "BusInterface_CAN" subelement generally represents the fieldbus connection via the CANopen (Motion) interfaces of the IPC.
3. Add the H "Axis_Group_CAN" subelement to the "BusInterface_CAN" element.
The "Axis_Group_CAN" subelement represents the single CANopen (Motion) interface of the IPC (the special Sub-D plug).
4. Assign a task to "Axis_Group_CAN" (e.g. the "Motion task" created before).

**Note!**

To prevent the sync telegram from jittering, all Motion phases must be assigned to the same task.

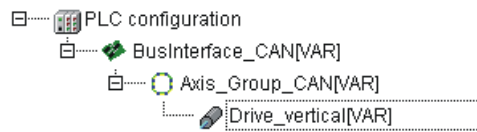
5. Enter the suitable controller number in the "Controller No." field in the "Specific Settings" area.

This serves to define the number of the CANopen interface via which the Motion drives are to be controlled. The following assignment applies:

CAN interface	Controller no.	Notes
CAN1	0	The CAN1 connection is usually used for the Logic bus. Thus the first Motion bus mostly is CAN2.
CAN2	1	
CAN3	2	
CAN4	3	

6. Set the corresponding baud rate of the Motion axes in the "Baud Rate" input field in the "Specific settings" area.
7. Add the "Drive" subelement to the "Axis_Group_CAN".
8. Assign an appropriate name for the "Drive" Motion drive (e.g. "Drive_vertical") in the task configurator tree (left window). With a mouse-click on "Drive", the name is activated for editing.
9. Adapt the "Drive ID".
The "Drive ID" corresponds to the node number of the Motion drive.
10. Enter the value 66536 into the "**Increments**" input field in the "Conversion Factor" area.
11. Repeat the steps 7 to 10 until the required number of Motion drives is reached.

12. The following control configuration is the result with a minimum configuration with one drive:



13. A single **rotary axis** Motion drive (type: rotary, 360° / revolution, ratio 1:1) shall then be configured as follows:

Drive	CANDrive	Module parameters												
Drive id:	1													
Type:	☒ rotary (modulo) ☐ linear													
Scale <table border="0"> <tr> <td> 65536 Increments</td> <td><-></td> <td> 1 Motor rotations</td> </tr> <tr> <td> 1 Input rotations of gear</td> <td><-></td> <td> 1 Output rotations of gear</td> </tr> <tr> <td> 1 Output rotations of gear</td> <td><-></td> <td> 360 SoftMotion units</td> </tr> </table>			65536 Increments	<->	1 Motor rotations	1 Input rotations of gear	<->	1 Output rotations of gear	1 Output rotations of gear	<->	360 SoftMotion units			
65536 Increments	<->	1 Motor rotations												
1 Input rotations of gear	<->	1 Output rotations of gear												
1 Output rotations of gear	<->	360 SoftMotion units												
Settings for rotary drive Modulo value [SM units]: 360.00000														
Cyclic communications data <table border="0"> <tr> <td>PLC -> Drive:</td> <td>☒ POS</td> <td>☐ VEL</td> <td>☐ ACC</td> <td>☐ TOR</td> <td>☐ CUR</td> </tr> <tr> <td>Drive -> PLC:</td> <td>☒ POS</td> <td>☐ VEL</td> <td>☐ ACC</td> <td>☐ TOR</td> <td>☐ CUR</td> </tr> </table>			PLC -> Drive:	☒ POS	☐ VEL	☐ ACC	☐ TOR	☐ CUR	Drive -> PLC:	☒ POS	☐ VEL	☐ ACC	☐ TOR	☐ CUR
PLC -> Drive:	☒ POS	☐ VEL	☐ ACC	☐ TOR	☐ CUR									
Drive -> PLC:	☒ POS	☐ VEL	☐ ACC	☐ TOR	☐ CUR									
<table border="0"> <tr> <td> Maximum values (CNC: SMC_ControlAxisByPos) Velocity: 100.00000 SM-units/s Acceleration: 100.00000 SM-units/s² Deceleration: 100.00000 SM-units/s² </td> <td> Velocity ramp type ☒ trapezoid ☐ sigmoidal ☐ parabolic Jerk: 100000.00 SM-units/s³ </td> </tr> </table>			Maximum values (CNC: SMC_ControlAxisByPos) Velocity: 100.00000 SM-units/s Acceleration: 100.00000 SM-units/s² Deceleration: 100.00000 SM-units/s²	Velocity ramp type ☒ trapezoid ☐ sigmoidal ☐ parabolic Jerk: 100000.00 SM-units/s³										
Maximum values (CNC: SMC_ControlAxisByPos) Velocity: 100.00000 SM-units/s Acceleration: 100.00000 SM-units/s² Deceleration: 100.00000 SM-units/s²	Velocity ramp type ☒ trapezoid ☐ sigmoidal ☐ parabolic Jerk: 100000.00 SM-units/s³													

When using a **linear axis** Motion drive (type: linear), you can define the positions of the software limit switches in the "Settings for linear drive" area.

Drive

Drive id:

Type: ☐ rotary (modulo) ☒ linear

Scale

Increments <-> Motor rotations

Input rotations of gear <-> Output rotations of gear

Output rotations of gear <-> SoftMotion units

Settings for linear drive

☐ Use software endswitch

neg. endswitch: pos. endswitch:

9.6 Creating a program code to control the Motion drives

This depends on the automation task, the use of PLCopen blocks or the CNC programming.



Note!

All SoftMotion function blocks, SoftMotion functions, and the read/write block parameters that access the SoftMotion drives (e.g. MC_WriteParameter, MC_WriteBoolParameter, LenzeECSReadParameter, LenzeECSWriteParameter) may only be called from the Motion task.

If they are called from another task, their execution may be incorrect.

See also "[Creating a program code to control the device](#)" (chapter [\[8.8\]](#)  65).

9.7 Preparing the restart

Save parameter set of the drive via the »Engineer« or the »Global Drive Control«.

Please also see "[Preparing the restart](#)" (chapter [\[8.9\]](#)  68).

9.8 Optimisation of signal propagation delays (for HighLine CiA402 only)

An optimised setting is possible at a low packing density of the telegrams on the fieldbus.
An optimised setting serves to achieve shorter signal propagation delays (Control → Drive → Control) when the packing density is < 0.7 :

$$(\text{Sum of all telegram runtimes}) / (\text{set sync cycle time}) < 0.7$$

Telegram runtimes

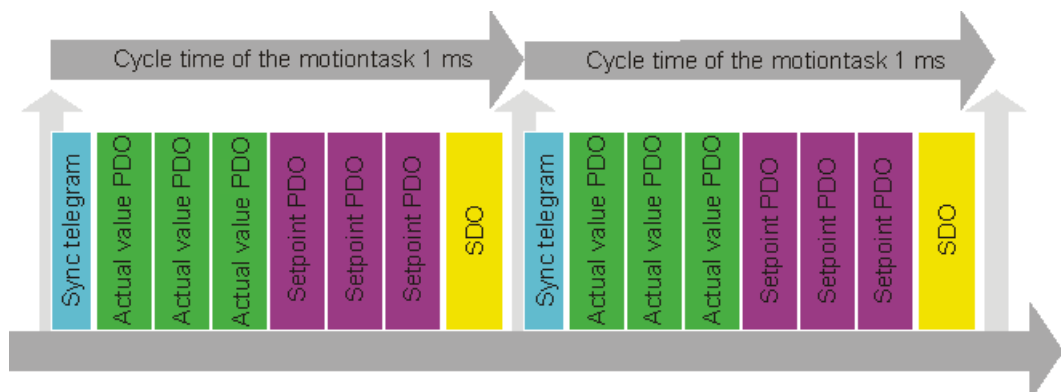
Telegram/data object	Telegram runtime at 500 kbits/s	Telegram runtime at 1 Mbits/s
Sync telegram	100 μs	50 μs
Actual value PDO	240 μs	120 μs
Setpoint PDO	240 μs	120 μs
SDO	240 μs	120 μs

9.8.1 Example 1: 3 drives in 1 ms at 1 Mbit/s

- Set sync cycle time (C1121): 1000 μs
- Set sync phase position (C1122): 120 μs

Telegram runtimes	
1 sync telegram	50 μs
1 actual value PDO per drive	360 μs (120 μs x 3)
1 setpoint PDO per drive	360 μs (120 μs x 3)
1 SDO	120 μs
Sum of all telegram runtimes	890 μs

- This results in the following telegram order within one cycle:



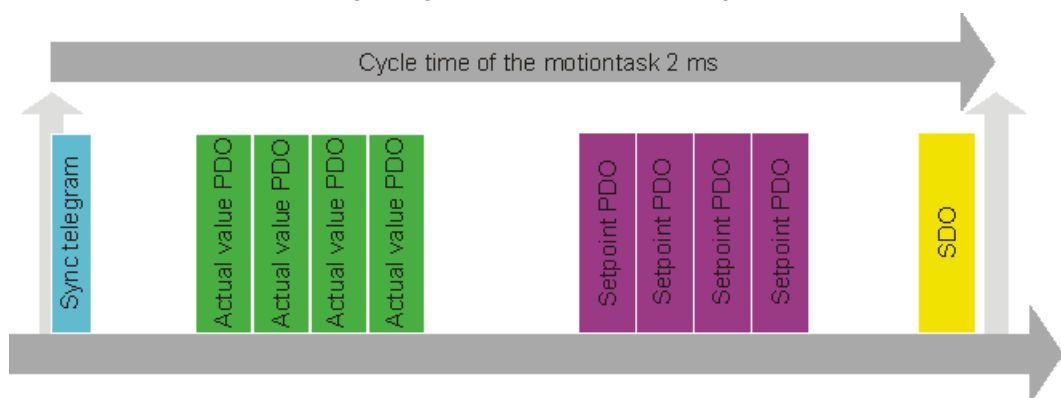
- Packing density: $890 \mu\text{s} / 1000 \mu\text{s} = 0.89$
→ Use standard setting.

9.8.2 Example 2: 4 drives in 2 ms at 1 Mbps

- Set sync cycle time (C1121): 2000 μ s
- Set sync phase position (C1122): 120 μ s

Telegram runtimes	
1 sync telegram	50 μ s
1 actual value PDO per drive	480 μ s (120 μ s x 4)
1 setpoint PDO per drive	480 μ s (120 μ s x 4)
1 SDO	120 μ s
Sum of all telegram runtimes	1130 μ s

- This results in the following telegram order within one cycle:



- Packing density: $1130 \mu\text{s} / 2000 \mu\text{s} = 0.565$
→ standard setting or optimised setting possible.

Calculation of the sync phase position for the optimised setting

$$\text{Sync phase position (C1122)} = \text{sync cycle time (C1121)} / 2$$

Thus the following results for example 2:

- Sync phase position (C1122): $2000 \mu\text{s} / 2 = 1000 \mu\text{s}$

10 CANopen with PROFIBUS

The CANopen bus system can be combined with PROFIBUS. This makes sense if not all field devices are available for the same bus system or a Motion bus (CANopen) is required in parallel to the PROFIBUS (as Logic bus). The bus systems are synchronised in the control.



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".
- Release 2.5 does not facilitate a combination of PROFIBUS and EtherCAT.
- In the control configuration, the PROFIBUS master must be in the first position – upstream to the CANopen motion nodes.

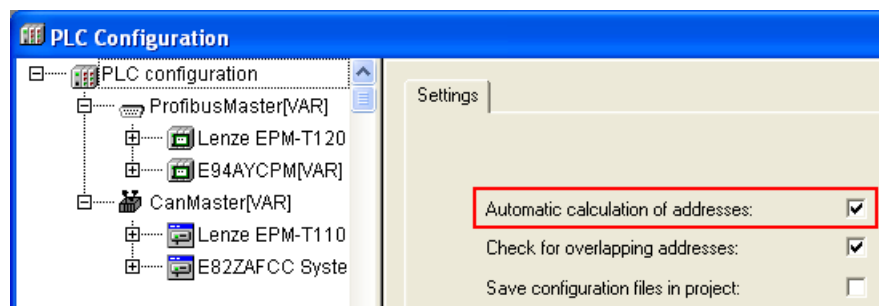


"PROFIBUS control technology" communication manual

Here you can find detailed information on how to commission PROFIBUS components.

Addressing the CANopen and PROFIBUS nodes

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 topic can be found in the documentation/online help of the »PLC Designer«.

11 The function library LenzeCANdrive.lib

The **LenzeCANdrive.lib** function library supports the SoftMotion control to control the "Servo Drives 9400 HighLine CiA402" series and the "ECSxM" axis module.

Features

- RegulatorOn, DriveStart
- Error message and acknowledgement
- Reading and writing of SoftMotion and drive parameters:
 - Access via index 0xaabb_{hex} and subindex 0xcc_{hex} with the length 0xdd_{hex} in bytes (only required for writing) or
 - use of MC_Read/Write(Bool)Parameter with parameter number ddaabbc_{hex} or
 - use of the function blocks LenzeECSReadParameter and LenzeECSWriteParameter to directly access Lenze codes or
 - use of the function blocks SMC_ReadCANParameter and SMC_WriteCANParameter to address standard CAN objects via the index/subindex.
- Reading of drive parameters (as string) with LenzeECSReadString
- Any ratio factors (dwRatioTechUnitsDenom/iRatioTechUnitsNum)
- Linear or rotary axes
- Control modes: position, velocity (9400), torque (9400)
 - Use SMC_SetControllerMode to change the mode.
- Drive-internal homing
 - ECSxM: Configure C3010, C0935, C0936
Note: During the homing mode, the current position is not indicated by the ECS drive!
 - Configure 9400: 6098_{hex}, 6099_{hex}, 609A_{hex}
- Latching: 1 channel (trigger number = 1), only ECSxM
- ECSxM: Depending on the settings in code C3175 the hardware limit switches are monitored by the control (C3175 = 3) or the drive.
- Configuration from file is possible
- Configuration from dialogs in the PLC configuration is possible
- Supported sync generators (to be set in the PLC configuration, AxisGroup: PLC, 1st drive, sync device)



Detailed information on the **LenzeCANdrive.lib** function library is provided in the **online help of the »PLC Designer«** and the **software manual »PLC Designer - SoftMotion«**.

12 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 calculation of the minimum cycle time is divided into the following steps:

1. Calculating the total access time $T_{\text{Correction}}$ to the peripheral devices.
 ▶ [Calculating the total access time to the peripheral devices \(\$T_{\text{Correction}}\$ \)](#) (87)
2. Detecting the task utilisation $T_{\text{Task utilisation}}$ of the application during operation.
 ▶ [Detecting the task utilisation of the application \(\$T_{\text{Task utilisation}}\$ \)](#) (88)
3. Calculating the minimum cycle time.
 ▶ [Calculating the minimum cycle time](#) (90)
4. Optimising the system.
 ▶ [Optimising the system](#) (91)

12.1 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 ATOM 1.6 GHz
CAN master (Logic)	80 μs
per axis (Motion)	60 μs

Example

Access times Industrial PC (ATOM 1.6 GHz) with 9 Motion axes	
Access time CAN master	80 μs
+ access time 9 Motion axes	540 μs
= total access time	620 μs

12.2 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«.

12.2.1 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 IEC 61131 *SysTaskInfo* library must be included in the project.

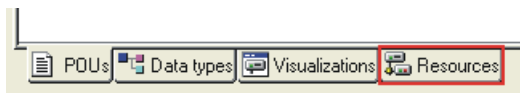
The task editor contains a dialog window consisting of two parts.

- ▶ 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.

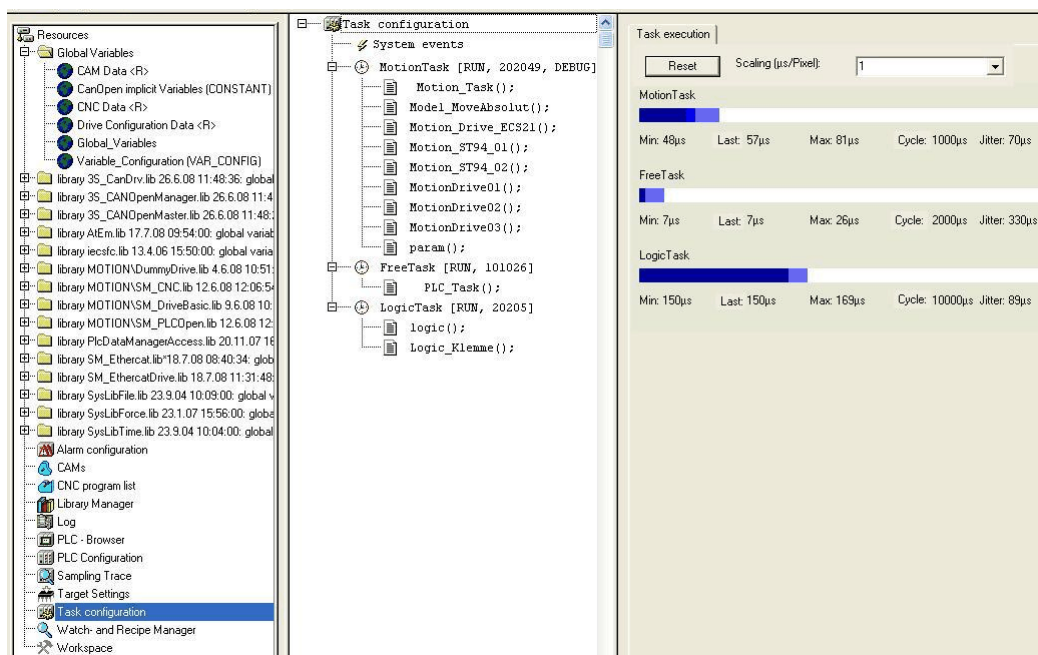


How to display the system utilisation:

1. Select the *Resources* tab:



2. Open the *Task configuration* in the online mode of the »PLC Designer«:



12.2.2 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 ($T_{\text{Task utilisation}}$).

12.3 Calculating the minimum cycle time



Note!

A safety factor of 1.5 is included in the calculation of the minimum cycle time.

The minimum cycle time $T_{\min}$ for a system results from the sum of the times detected before, multiplied by the safety factor:

$$T_{\min} > \text{safety factor} \times (T_{\text{Task utilisation}} + T_{\text{Correction}})$$

Example

Configuration: Industrial PC (ATOM 1.6 GHz) with 9 Motion axes		
Detected access time		Result
Calculated correction value	$T_{\text{Correction}}$	620 μs (80 μs + (9 x 60 μs))
Value read from task configuration:	$T_{\text{Task utilisation}}$	500 μs
Actual required computing time		1120 μs
Minimum cycle time including a safety factor of 1.5	$T_{\min}$	1680 μs
Actual cycle time		2000 μs

12.4 Optimising the system



How to optimise the system:

1. Log in and load project.
2. Check the task processing times.
3. Optimising 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.

13 Diagnostics

13.1 Reading codes

The »Engineer« and the »Global Drive Control« contain the corresponding diagnostic codes.

13.2 Viewing the logbook of the IPC

In the web browser you have access to the logbook of the PC. Use the display filter to search for entries concerning CANopen.



Note!

"ClearLog" deletes the complete logbook on the IPC without any prompt.

The screenshot shows the Lenze LogViewer web interface. On the left is a sidebar with navigation links: Logging, LogViewer, Device Page 1, Device Page 2, Command, Ethernet 1, Panel, PLC, User 1-5, User 6-10, Slot Mpci 1, Slot Mpci 2, McDualCan 1, McCanIf 1, McCanIf 2, and Services. The main content area has a blue header with the 'Lenze' logo. Below the header are two buttons: 'Apply & Refresh' and 'ClearLog'. The 'LogBook' section displays a list of log entries with timestamps and messages. Below the log entries is a 'Time Period Selection' section with a dropdown menu set to 'All'. There are also input fields for 'From' and 'To' dates/times, and a 'Last' dropdown set to '1 Days'. At the bottom, there are checkboxes for 'All', 'WindowsCE', 'DeviceScanner', and 'DataManager'.

Time Period Selection	All
- From	yyyy-mm-dd hh:mm:ss 2007 11 01 15 46 44
- To	yyyy-mm-dd hh:mm:ss 2007 11 01 15 46 44
- Last	1 Days
	All ☐
	WindowsCE ☐
	DeviceScanner ☐
	DataManager ☐

13.3 Error messages if communication card MC-CAN2 is not available

If no communication card MC-CAN2 has been inserted into the industrial PC, error messages occur during the download of the »PLC Designer« project.

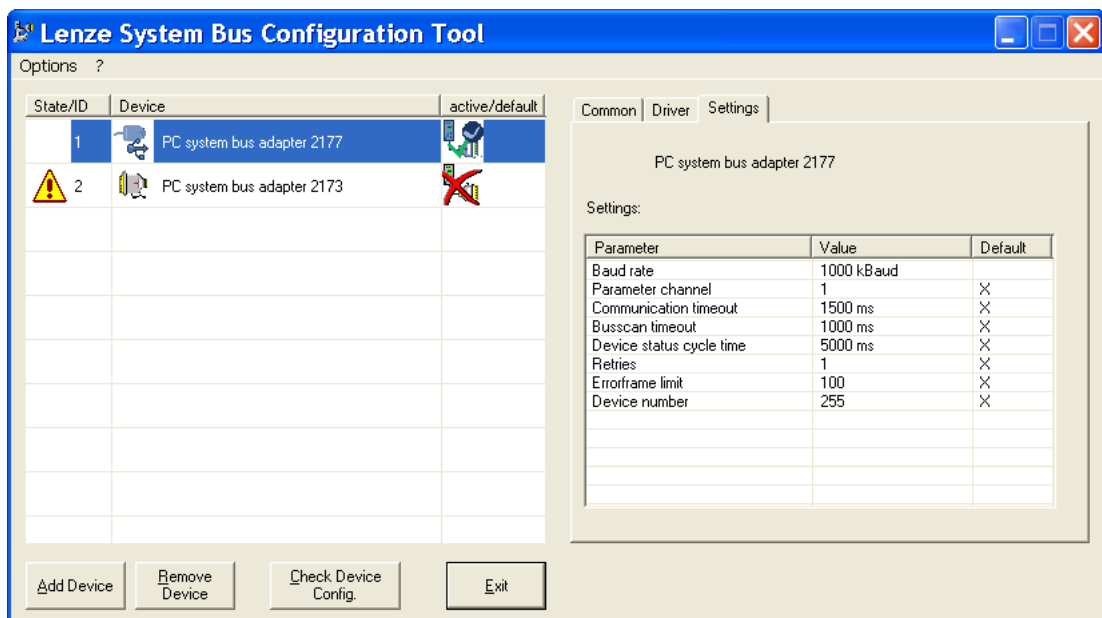
Remedy: Insert the communication card MC-CAN2 into the industrial PC.

13.4 Searching the CANopen bus for nodes using the Engineering PC



1. First directly connect your notebook via the USB system bus adapter EMF2177IB to the CANopen bus.
2. Start the program »System bus configurator«:
 - in the »Engineer« in the menu bar under "Online"
 - or under "Start - Programs - Lenze - Communication"
3. Activate the USB system bus adapter (EMF2177IB).
4. Check the following under the "Settings" tab:
 - baud rate = as on the devices
 - parameter data channel = 0:
The entire address range is scanned. Devices responding on several SDO channels are displayed with several node addresses.

Example:



5. Click the **Communication diagnostics** button on the **Common** tab to start the search.

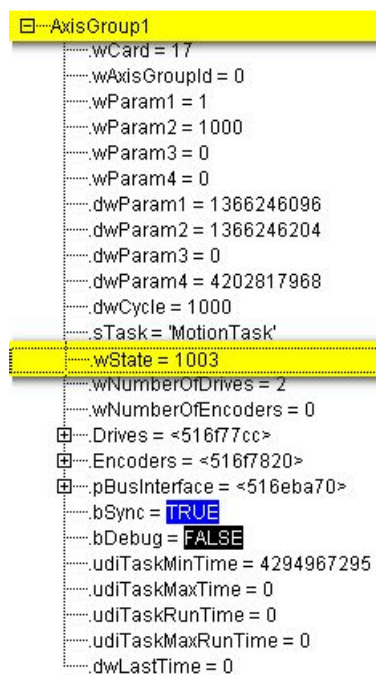
13.5 The global variable *wState*

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.

14 Parameter reference

This chapter adds the parameters of the **MC-CAN2 communication card** to the parameter list provided in the online documentation for the Industrial PC:

▶ [Parameters of the MC-CAN2 communication card in slot 1](#) (📖 97)

▶ [Parameters of the MC-CAN2 communication card in slot 2](#) (📖 98)



Note!

- Several MC-CAN2 communication cards can be used per Industrial PC. The designation of the card in the »WebConfig« is *MC-CAN2*.
- Depending on the used slot, the code numbers differ by an offset of '500'. Hence, for a communication card in slot 2, an offset of '500' has been added to the code numbers of a card in slot 1.



Tip!

General information about parameters is provided in the online documentation for the Industrial PC.

14.1 Parameters of the MC-CAN2 communication card in slot 1

The parameters are listed in numerically ascending order.

C1031

Parameter Name:	Data type: VISIBLE_STRING Index: 23544 _d = 5BF8 _h
C1031 Device: Identification	
Identification of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer	

C1032

Parameter Name:	Data type: VISIBLE_STRING Index: 23543 _d = 5BF7 _h
C1032 Device: Version	
Version number of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer	

C1033

Parameter Name:	Data type: VISIBLE_STRING Index: 23542 _d = 5BF6 _h
C1033 Device: Name	
Device name of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer	

C1034

Parameter Name:	Data type: VISIBLE_STRING Index: 23541 _d = 5BF5 _h
C1034 Device: Software version	
Software version of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer	

C1035

Parameter Name:	Data type: VISIBLE_STRING Index: 23540 _d = 5BF4 _h
C1035 Device: Hardware version	
Hardware version of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer	

C1036

Parameter Name:	Data type: VISIBLE_STRING Index: 23539 _d = 5BF3 _h
C1036 Device: Serial number	
Serial number of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer	

C1037

Parameter Name:	Data type: VISIBLE_STRING Index: 23538 _d = 5BF2 _h
C1037 Device: Manufacturer	
Manufacturer of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer	

C1038

Parameter Name:	Data type: VISIBLE_STRING Index: 23537 _d = 5BF1 _h
C1038 Device: Manufacturing date	
Manufacturing date of the card	
☒ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer	

14.2 Parameters of the MC-CAN2 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: Identification	
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: 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 version	
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 version	
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	

15 Appendix

This chapter describes how to use the »PCAN-View« program for diagnostics of your CANopen network.

»PCAN-View« is the basic version of the »PCAN-Explorer« program for Windows® of PEAK System Technik GmbH. The program permits a simultaneous transmission and receipt of CAN messages which can be sent manually and periodically. Errors on the bus system and memory overflows of the controlled CAN hardware are displayed.

Finally important information on visualisation using »VisiWinNET®« is provided.

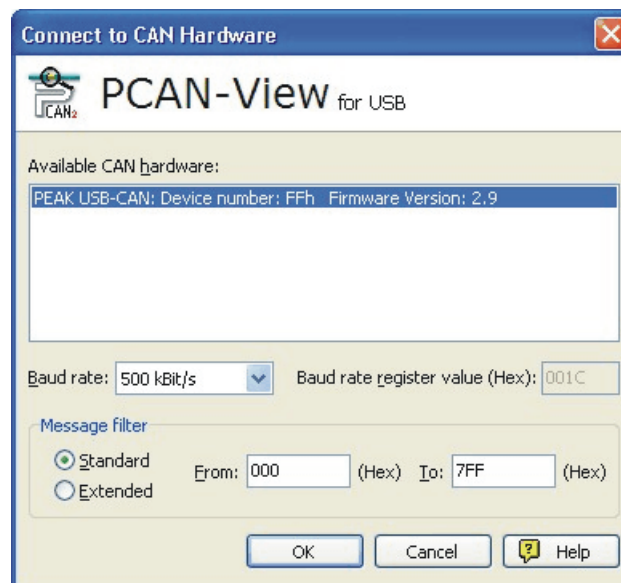
15.1 »PCAN-View« for diagnostic purposes



Monitor telegram traffic on the CANopen bus

1. First directly connect your Engineering PC via the USB system bus adapter EMF2177IB to the CANopen bus.
2. Start the program »PCAN-View«.
3. Connect the »PCAN-View« with "**Connect to CAN Hardware**" according to the USB system bus adapter and the baud rate.

Example:



Now the CAN telegrams are constantly displayed in the "Receive" and "Transmit" windows:

Message	Length	Data	Period	Count
100h	1	FF	100	1053
101h	8	9C 01 2B 01 BB 01 46 01	50	2107
102h	4	43 00 00 00	49	2107
205h	8	00 00 00 00 00 00 00 00	50	2106
206h	5	64 64 64 64 64	50	2106

Message	Length	Data	Period	Count	Trigger
300h	4	25 4B 8D 76	10	3602	Time
301h	4	76 DF FF AB	Wait	56	Manual
00FF77BBh	1	00	Wait	0	

Connected to PEAK USB-CAN (500 kBit/s) Overruns: 0 QXmtFull: 0

On the basis of the IDs displayed and the IDs in your overview you can assign the telegrams to the devices.

If no telegrams are displayed, there may be several causes:

- Is your Engineering PC connected to the correct CANopen bus?
- Is the correct system bus adapter activated under "System control, CAN hardware"?
- What does it say in the status bar of the »PCAN-View«?
If it says "Bus Heavy" mostly a node with an incorrect baud rate interferes with the bus communication.
- Are the devices in the "Operational" state?

Use »PCAN-View« to set all nodes to the "Operational" state.

15.2 Use »PCAN-View« to set all nodes to the "Operational" state.

- Under "New transmit message" create the following CAN message.

New transmit message

ID (Hex): FFAA77 Length: 3 Data (0..7): FE 11 2

Period: 0 ms ☒ Extended Frame ☐ Remote Request

OK Cancel Help

- Select the CAN message in the "Transmit" window and press the <space bar> once to send it.

PCAN-View for USB

Client Transmit Help

Message	Length	Data	Period	Count
100h	1	FF	100	1053
101h	8	9C 01 2B 01 BB 01 46 01	50	2107
102h	4	43 00 00 00	49	2107
205h	8	00 00 00 00 00 00 00 00	50	2106
206h	5	64 64 64 64 64	50	2106

Message	Length	Data	Period	Count	Trigger
300h	4	25 4B 8D 76	10	3602	Time
301h	4	76 DF FF AB	Wait	56	Manual
00FF77BBh	1	00	Wait	0	

Connected to PEAK USB-CAN (500 kBit/s) Overruns: 0 QXmtFull: 0

15.3 Notes on visualisation using »VisiWinNET®«

Problem description:

For visualisation purposes a VWGET timer (1 second) is used to read a variable via the CAN bus (SDO) using the OPC server. If no nodes are connected and no information is returned, a time-out is generated. Waiting for the time-out causes the page-turning function of the visualisation program to slow down.

Solution:

- The visualisation program can be operated again if the following lines of the time-out are changed in the **PosStart.txt** file:

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\CAN2Bus1]
"SDOTimeout"=dword:0xc8
"Timeout"=dword:0xc8
```

- Otherwise VWSET and VWGET must not be used for visualisation purposes.

16 Index

A

Addressing the CANopen and PROFIBUS nodes [85](#)

Appendix [99](#)

Application notes (representation) [11](#)

B

Basic identifier - 8200 vector with fieldbus function module CANopen E82ZAFUC0xx [34](#)

Basic identifier - 8400 Inverter Drives [31](#)

Basic identifier - 9400 Servo Drives [30](#)

Basic identifier - ECS servo system [36](#)

Basic identifier - I/O system 1000 (EPM-Sxxx) [33](#)

Basic identifier - I/O system IP20 (EPM-Txxx) [32](#)

Baud rates [20](#)

Bus cable length [22](#)

C

C1031 | Device - identification [97](#)

C1032 | Device - version [97](#)

C1033 | Device - name [97](#)

C1034 | Device - software version [97](#)

C1035 | Device - hardware version [97](#)

C1036 | Device - serial number [97](#)

C1037 | Device - manufacturer [97](#)

C1038 | Device - manufacturing date [97](#)

C1531 | Device - identification [98](#)

C1532 | Device - version [98](#)

C1533 | Device - name [98](#)

C1534 | Device - software version [98](#)

C1535 | Device - hardware version [98](#)

C1536 | Device - serial number [98](#)

C1537 | Device - manufacturer [98](#)

C1538 | Device - manufacturing date [98](#)

Cable specification for bus cable [21](#)

Calculating the minimum cycle time [90](#)

Calculating the total access time to the peripheral devices [87](#)

Calculation of the sync phase position [84](#)

CANopen [15](#)

CANopen (Logic) [15](#)

CANopen (Motion) [15](#)

CANopen Hardware for your Industrial PC [18](#)

CANopen with PROFIBUS [85](#)

COB-IDs acc. to DS301 [26](#)

Codes [96](#)

Combination with other bus systems [16](#)

Commissioning of 8200 vector [52](#)

Commissioning of 8200 vector frequency inverter [52](#)

Commissioning of 8400 Inverter Drives [47](#)

Commissioning of ECS devices [53](#)

Commissioning of field devices [42](#), [74](#)

Commissioning of I/O system 1000 (EPM-Sxxx) [51](#)

Commissioning of I/O system IP20 (EPM-Txxx) [50](#)

Commissioning the CANopen Logic bus [40](#)

Commissioning the CANopen Motion bus [73](#)

Commissioning the Servo Drives 9400 [44](#)

Communication card MC-CAN2 [18](#)

Communication profile [20](#)

Configuring the CAN master (Logic node) [57](#)

Connecting the Engineering PC with the Industrial PC [38](#)

Connection of CAN bus (SUB-D, 9-pole plug) [21](#)

Conventions used [9](#)

Copyright [2](#)

Creating a control configuration [79](#)

Creating a Motion task [78](#)

Creating a PLC program (Logic node) [54](#)

Creating a PLC program (Motion node) [75](#)

Creating a program code to control the device [65](#)

Creating a program code to control the Motion drives [82](#)

Creating a project folder [41](#)

Cross communication [20](#)

Cross communication between the slaves [62](#)

Cycle synchronisation [20](#)

Cycle time [20](#)

D

Defining the minimum cycle time of the PLC project [87](#)

Detecting the task utilisation [89](#)

Detecting the task utilisation of the application [88](#)

Device

Hardware version (C1035) [97](#)

Hardware version (C1535) [98](#)

Identification (C1031) [97](#)

Identification (C1531) [98](#)

Manufacturer (C1037) [97](#)

Manufacturer (C1537) [98](#)

Manufacturing date (C1038) [97](#)

Manufacturing date (C1538) [98](#)

Name (C1033) [97](#)

Name (C1533) [98](#)

Serial number (C1036) [97](#)

Serial number (C1536) [98](#)

Software version (C1034) [97](#)

Software version (C1534) [98](#)

Version (C1032) [97](#)

Version (C1532) [98](#)

Device specifications of the field devices [29](#)

Diagnostics [92](#)

Diagnostics with »PCAN-View« [99](#)

Direct coupling (PC - CANopen bus) [38](#)

Display of the system utilisation in the »PLC Designer« with the task editor [88](#)

Document history [8](#)

E

Engineering tools [40](#), [73](#)

F

Field devices [17](#)

G

General data [20](#)

Going online [43](#)

I

Installing field devices [37](#)

Integrating field devices (slaves) into the PLC program [59](#)

IPC as gateway [38](#)

L

LenzeCANDrive.lib [86](#)

LenzeCANDrive.lib function library [86](#)

Logbook of the IPC [92](#)

M

Mixed operation (CANopen with PROFIBUS) [85](#)

Max. number of drives/ [20](#)

N

Network topology [20](#)

Node addresses [20](#)

Notes used [11](#)

Number of DI + DO (bits/ms) [20](#)

Number of nodes [20](#)

O

Optimise signal propagation delays [83](#)

Optimising the signal propagation delays (optional) [83](#)

Optimising the system [91](#)

P

Parameter data [20](#)

Parameter reference [96](#)

Parameters of the MC-CAN2 communication card in slot 1 [97](#)

Parameters of the MC-CAN2 communication card in slot 2 [98](#)

Planning the CAN network [25](#)

Preparing the restart [68](#), [82](#)

PROFIBUS with CANopen [85](#)

S

Safety [12](#)

Safety instructions (representation) [11](#)

Segment cable length [23](#)

Setting of CAN mapping [60](#)

Setting of CAN parameters [60](#)

Setting the baud rate [37](#)

Setting the node addresses [37](#)

Signal propagation delay [20](#)

Standards [20](#)

Sync phasing [84](#)

System bus (CAN) [15](#)

T

Target group [8](#)

Technical data [20](#)

Technical data of the MC-CAN2 communication card [21](#)

Telegram runtimes [83](#)

Terms [10](#)

Total cable length [22](#)

U

Use of repeaters [24](#)

Using PCAN view [99](#)

V

Variable wState [94](#)



© 07/2011

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

KHBCANPCBAUTO ■ 13383676 ■ EN 4.2 ■ TD17

10 9 8 7 6 5 4 3 2 1