Accessories

PROFIBUS®

E84AYCPM

Inverter Drives 8400 ________________________

Communication Manual EN

13422193

Lenze

Contents

1 About this documentation

This documentation exclusively describes the E84AYCPM communication module (PROFIBUS).

Note!

This documentation supplements the **mounting instructions** supplied with the communication module and the **"Inverter Drives 8400" hardware manual**.

The hardware manual contains safety instructions which must be observed!

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

Typical applications are explained with the help of examples.

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

This documentation does not describe the software of 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 control system (master).

All brand names used in this documentation are trademarks of their respective owners.

Tip!

Detailed information about PROFIBUS can be found on the website of the PROFIBUS & PROFINET user organisation:

www.profibus.com

Target group

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

Tip!

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

www.Lenze.com

Validity information

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

Extension module	Type designation	From hardware version	From software version
PROFIBUS communication module	E84AYCPM	VA	01.00

Screenshots/application examples

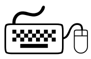
All screenshots in this documentation are application examples. Depending on the firmware version of the communication module and the software version of the Engineering tools installed (»Engineer«, »STEP7«), the screenshots in this documentation may differ from the actual screen display.

1.1 Document history

Version			Description
1.0	11/2007	TD17	First edition
2.0	11/2008	TD17	General revision
3.0	02/2010	TD17	• Update of chapter structure • General revision
4.0	11/2010	TD17	General revision
5.0	11/2011	TD17	• New layout • New: Going online with »Engineer« via TCI (📖 35) • General corrections

1.2 Conventions used

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

Type of information	Writing	Examples/notes
Numbers		
Decimal separator	Point	The decimal point is always used. Example: 1234.56
Hexadecimal	0x[0 ... 9, A ... F]	Example: 0x60F4
Binary • Nibble	In inverted commas Point	Example: '100' Example: '0110.0100'
Text		
Version information	Text colour blue	All pieces of information that only apply to or from a specific software version of the inverter are highlighted accordingly in this documentation. Example: This function extension is available from software version V3.0!
Program name	» «	The Lenze PC software »Engineer«...
Window	*italics*	The *message window*... / The *Options* dialog box ...
Variable name		Setting *bEnable* to TRUE...
Control element	**Bold**	The **OK** button ... / The **Copy** command ... / The Properties tab ... / The **Name** input field ...
Sequence of menu commands		If several successive commands are required for executing a function, the individual commands are separated from each other by an arrow: Select the command **File ➜ Open** to...
Hyperlink	underlined	Optically highlighted reference to another topic. Can be activated with a mouse-click in this online documentation.
Symbols		
Page reference	(🕮 6)	Optically highlighted reference to another page. Can be activated with a mouse-click in this online documentation.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

1.3 Terminology used

Term	Meaning
Inverter	Lenze inverter of the "Inverter Drives 8400" product series
Standard device	
Code	Parameters which serve to parameterise or monitor the inverter. This term is usually called "index".
Subcode	If a code contains several parameters, these are stored in subcodes. This manual uses a slash "/" as a separator between code and subcode (e.g. "C118/3"). This term is usually called "subindex".
GSD / GSE	Device data base file (device description for PROFIBUS stations)
HW	Hardware
Lenze setting	Settings with which the device is preconfigured ex works.
Basic setting	
	PROFIBUS® (Process Field Bus) is a widely-used fieldbus system for the automation of machines and production plants. PROFIBUS® is a registered trademark and patented technology licensed by the PROFIBUS & PROFINET International (PI) user organisation.
PDO	Process data object
PLC	Programmable Logic Controller (German designation: SPS - Speicherprogrammierbare Steuerung)
»STEP7«	Siemens software for programming and configuring PROFIBUS Siemens control systems
SW	Software
TCI	Tool Calling Interface

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 the safety instructions:

> **Danger!**
>
> (characterises the type and severity of danger)
>
> **Note**
>
> (describes the danger and informs how to prevent dangerous situations)

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

Application notes

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

2 Safety instructions

Note!

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

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

2.1 General safety and application instructions

Danger!

If you disregard the following basic safety measures, this can cause severe injury to persons and damage to material assets.

Lenze drive and automation components ...

- must only be used as directed.
  ▸ Application as directed (📖 13)
- must never be commissioned in the event of visible damage.
- must never be technically modified.
- must never be commissioned before they have been completely mounted.
- must never be operated without the covers required.
- can - depending on their degree of protection - have live, moving or rotating parts during and after operation. Surfaces can be hot.

For Lenze drive components ...

- use only the accessories approved.
- use only original spare parts from the manufacturer.

Observe all specifications given in the attached and associated documentation.

- This is the precondition for safe and trouble-free operation and for obtaining the product features specified.
  ▸ Features (📖 14)
- The procedural notes and circuit details described in this document are only proposals. It is up to the user to check whether they can be adapted to the particular applications. Lenze does not take any responsibility for the suitability of the procedures and circuit proposals described.

Only qualified personnel may work with and on Lenze drive and automation components. According to IEC 60364 and CENELEC HD 384, these are persons ...

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

2.2 Device- and application-specific safety instructions

- During operation, the communication module must be firmly connected to the standard device.
- Only use cables corresponding to the given specifications.
  ▸ Bus cable specification (📖 28)

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

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

2.3 Residual hazards

Protection of persons

If the Inverter Drives 8400 are used on a phase earthed mains with a rated mains voltage ≥ 400 V, protection against accidental contact is not ensured without implementing external measures.

▸ Protective insulation (📖 16)

Device protection

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

▸ Installation (📖 21)

3 Product description

3.1 Application as directed

The communication module ...

- is an accessory module for use in conjunction with the following Lenze standard devices:

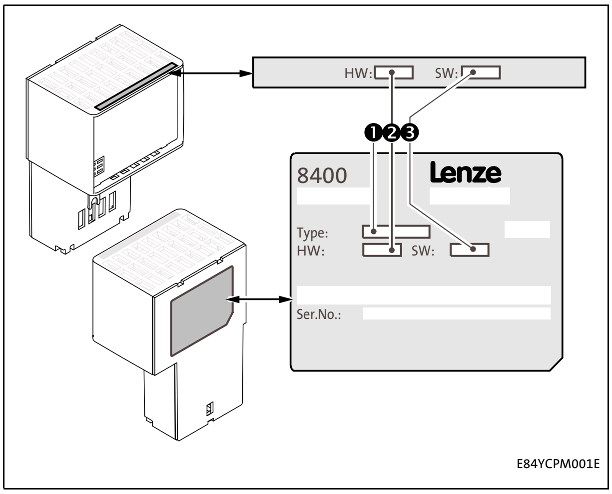
Product series	Type designation	From software version
Inverter Drives 8400 StateLine	E84AVSCxxxxx	01.00
Inverter Drives 8400 HighLine	E84AVHCxxxxx	01.00
Inverter Drives 8400 TopLine	E84AVTCxxxxx	01.00

- is a device intended for use in industrial power systems.
- is only to be operated under the operating conditions specified in this documentation.
- may only be used in PROFIBUS networks.

Any other use shall be deemed inappropriate!

3.2 Identification

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



1 Type designation (type)

E84	Product series
A	Version
Y	Module identification: extension module
C	Module type: communication module
PM	PROFIBUS
V/S	V: coated version S: standard version

2 Hardware version (HW)

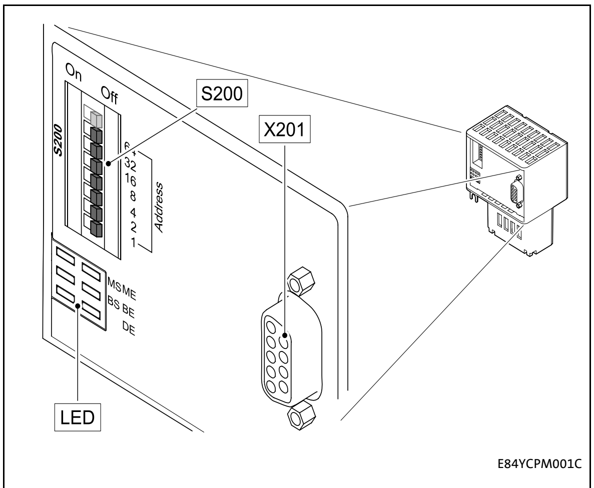
3 Software version (SW)

[3-1] Identification data

3.3 Features

- Interface module for the PROFIBUS communication system for connection to the expansion slots of the Inverter Drives 8400
- Support of parameter data channels DRIVECOM (DP-V0) and PROFIDrive (DP-V1)
- A maximum of 16 process data words per direction can be exchanged.
- The communication module is supplied with voltage via the standard device.
- Bus coupling via remote bus according to the RS485 standard
- Automatic detection of the baud rate (9.6 kbps to 12 Mbps)
- Setting of the station address is possible via DIP switch or code.
- Access to all Lenze parameters

3.4 Terminals and interfaces



[3-2] E84AYCPM communication module (PROFIBUS)

S200 DIP switches for setting the station address
▶ Setting the station address (📖 32)

X201 PROFIBUS connection
- 9-pin Sub-D socket

▶ Network topology (📖 25)
▶ PROFIBUS connection (📖 29)

MS ME BS BE DE 5 LED status displays for diagnostics
▶ Module status displays (📖 85)
▶ Fieldbus status displays (📖 86)

4 Technical data

"Inverter Drives 8400" hardware manual

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

4.1 General data and operating conditions

Area	Values
Order designation	• E84AYCPMV (coated version) • E84AYCPMS (standard version)
Communication profile	• PROFIBUS DP-V0 (DRIVECOM) • PROFIBUS DP-V1 (PROFIdrive)
Communication medium	RS485
Interface	9-pin Sub-D socket
Network topology	• Line (without repeater) • Tree/line (with repeater)
Bus device type	PROFIBUS slave
Number of slaves	• Max. 31 (without repeater) • Max. 125 (with repeater)
Max. cable length	1200 m (depending on the selected baud rate and the cable type used)
PNO identification number	0x0A89
Baud rate for cable type A (EN 50170)	9.6 kbps ... 12 Mbps (automatic detection)
Conformities, approvals	• CE • UL

4.2 Protective insulation

Danger!

Dangerous voltage

If the Inverter Drives 8400 are used on a phase earthed mains with a rated mains voltage ≥ 400 V, protection against accidental contact is not ensured without implementing external measures.

Possible consequences:

Death or severe injury

Protective measures:

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

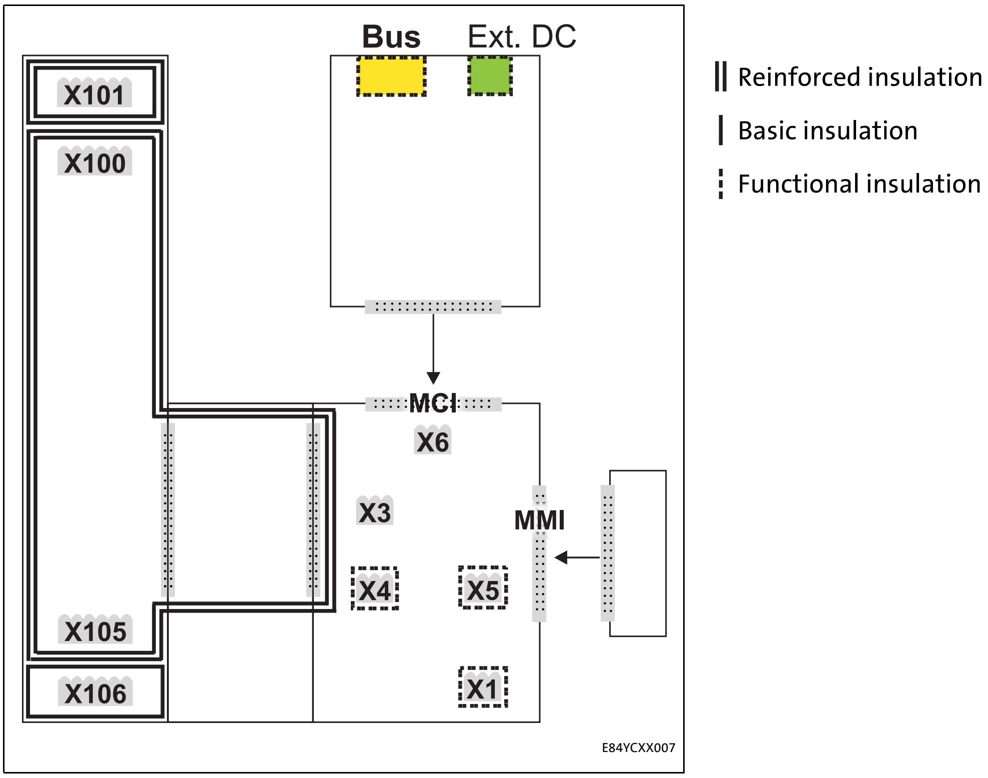
- a double isolating distance must be provided.
- the components to be connected must be provided with a second isolating distance.

Note!

The existing protective insulation in the Inverter Drives 8400 is implemented according to EN 61800-5-1.

The following illustration ...

- shows the arrangement of the terminal strips and the separate potential areas of the Inverter Drives 8400.
- serves to determine the decisive protective insulation between two terminals located in differently insulated separate potential areas.



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

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

Example

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

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

- The separate potential area of the device module's bus terminal is "functionally insulated".
- The separate potential area of the mains terminal has a "reinforced insulation".

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

4.3 Protocol data

Area	Values
Process data words (PCD)	1 ... 16 words (16 bits/word)
Cyclic parameter data channel (DP-V0)	4 words
Acyclic parameter data channel (DP-V1)	Max. 240 bytes
PROFIBUS user data length	1 ... 16 words process data channel + 4 words parameter data channel

4.4 Communication time

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

The communication times in a PROFIBUS network depend on ...

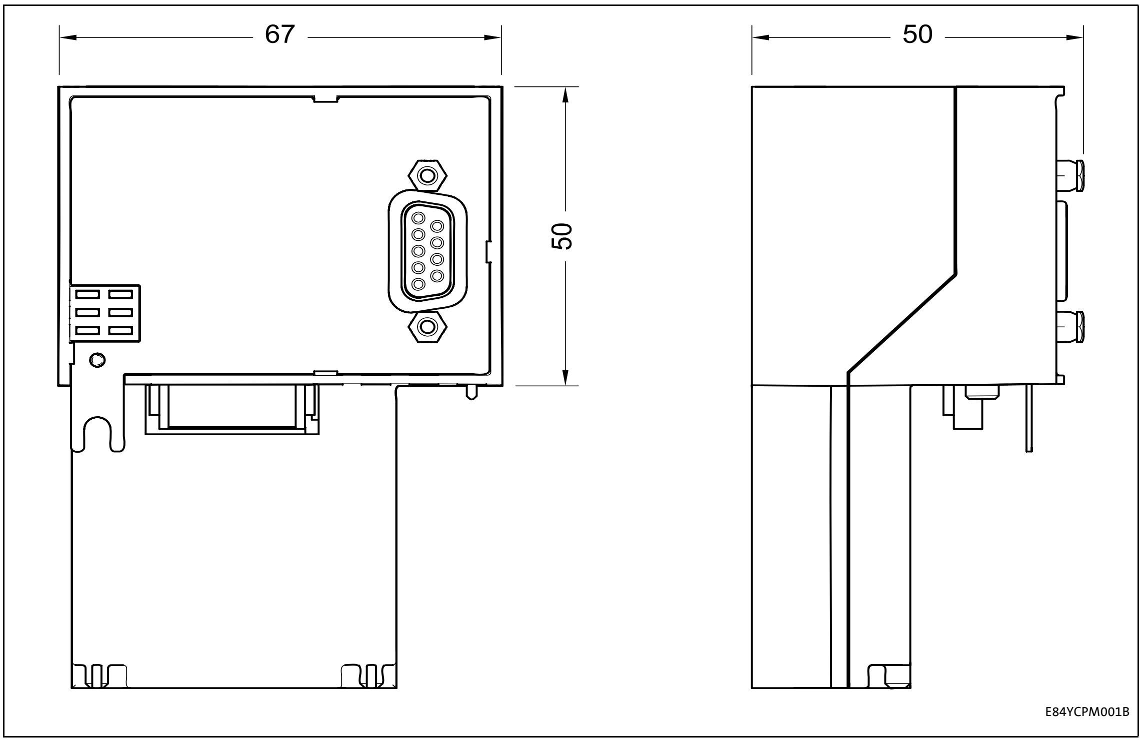
- the processing time in the inverter;
- the transmission delay time (baud rate / telegram length);
- the nesting depth of the network.

Processing time in the inverter

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

There are no interdependencies between parameter data and process data.

4.5 Dimensions



[4-2] Dimensions

Dimensions in mm

5 Installation

Stop!

Electrostatic discharge

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

Possible consequences:

- The communication module is defective.
- Communication via the fieldbus is not possible or faulty.

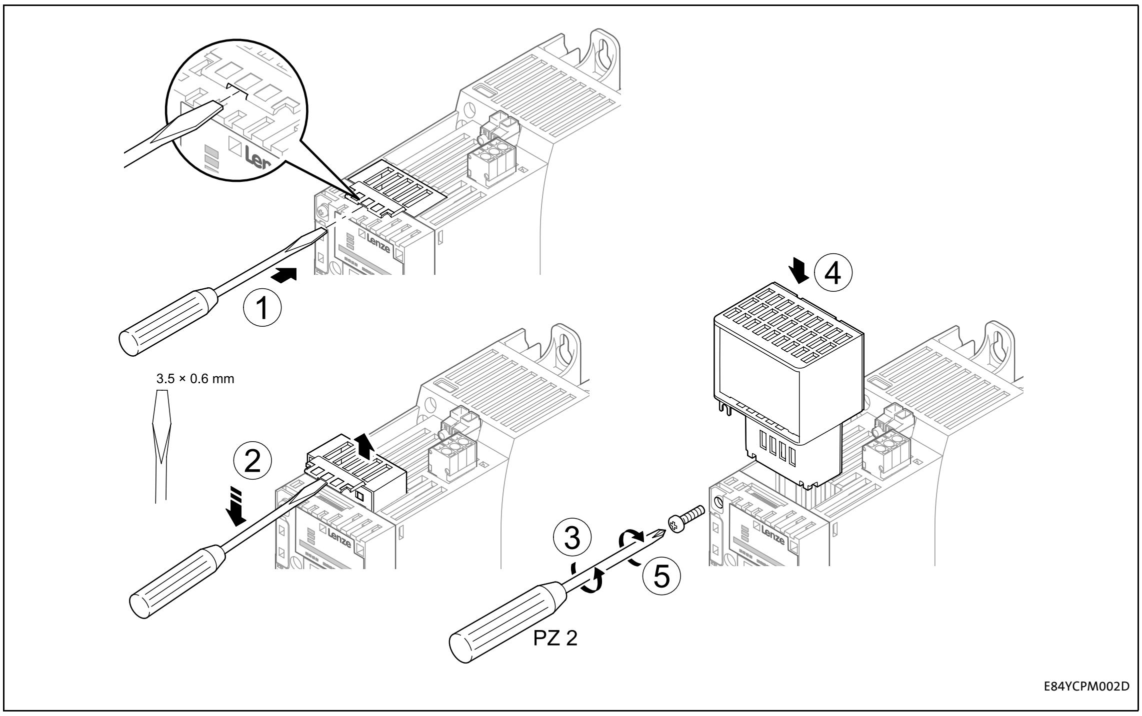
Protective measures

Discharge electrostatic charges before touching the module.

5.1 Mechanical installation

The communication module can be plugged into the MCI slot or unplugged while the inverter is switched on. When the module is plugged in, it is detected automatically, and a plausibility check regarding the function and version is carried out.

5.1.1 Mounting for 0.25 kW and 0.37 kW standard devices

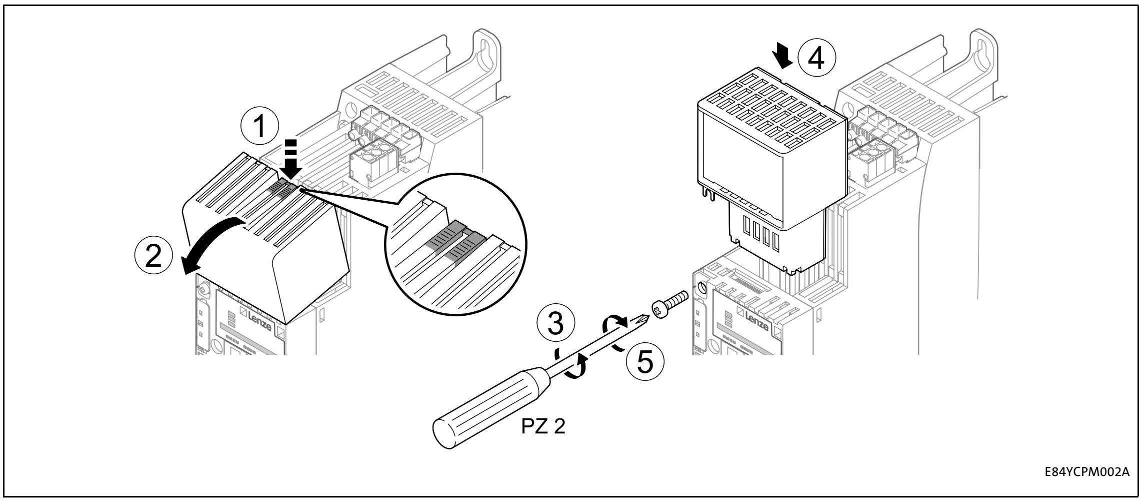


[5-1] Mounting for 0.25 kW and 0.37 kW standard devices

Mounting steps

1. Pry the cover out of the MCI slot using a screwdriver and remove it (1, 2).
2. Loosen the securing screw for the communication module on the standard device (3).
3. Insert the communication module into the MCI slot of the standard device (4).
4. Fasten the securing screw (5).

5.1.2 Mounting for standard devices of 0.55 kW and more

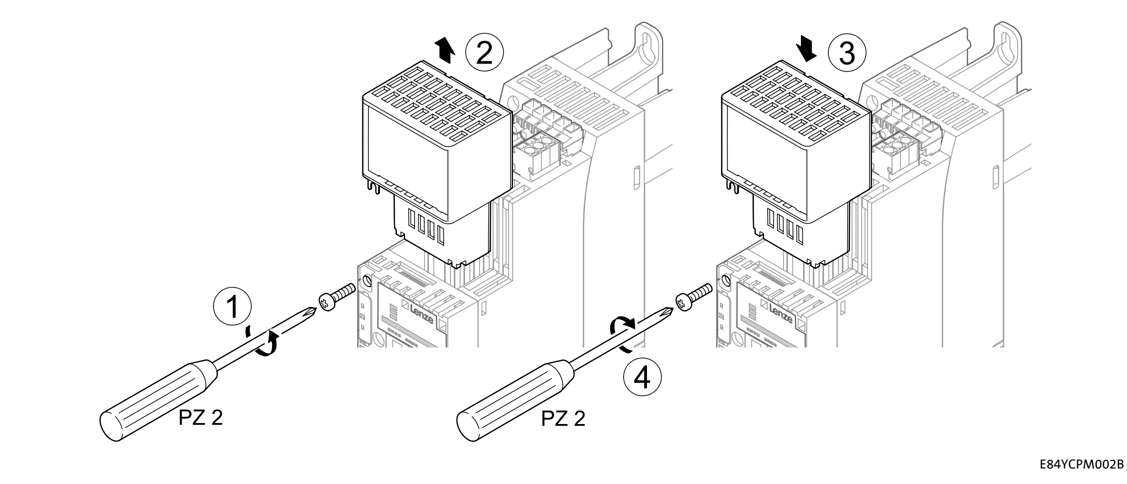


[5-2] Mounting for standard devices of 0.55 kW and more

Mounting steps

1. Slightly press on the area indicated in the illustration at the top of the cover for the standard device's MCI slot (1).
2. Tilt the cover forward and remove it from the standard device (2).
3. Loosen the securing screw for the communication module on the standard device (3).
4. Insert the communication module into the MCI slot of the standard device (4).
5. Fasten the securing screw (5).

5.1.3 Replacing the communication module



[5-3] Replacing the communication module

Mounting steps

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

5.2 Electrical installation

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

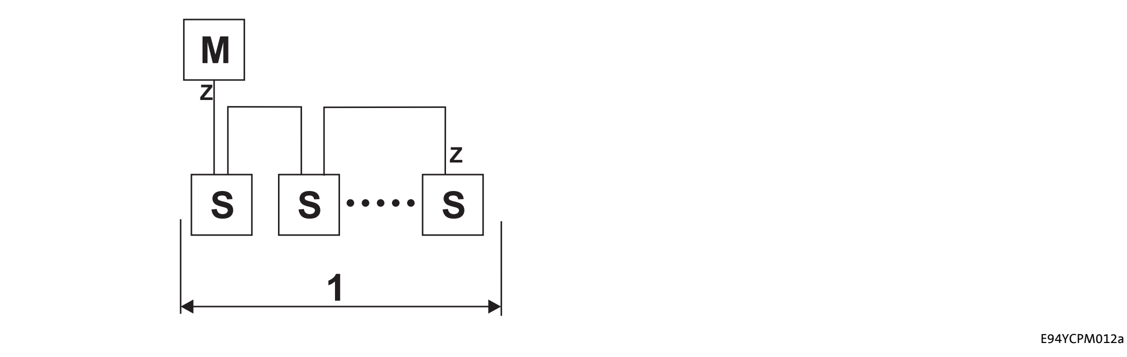
Observe the notes and wiring instructions given in the documentation.

5.2.1 Network topology

The following examples show two simple RS485 networks.

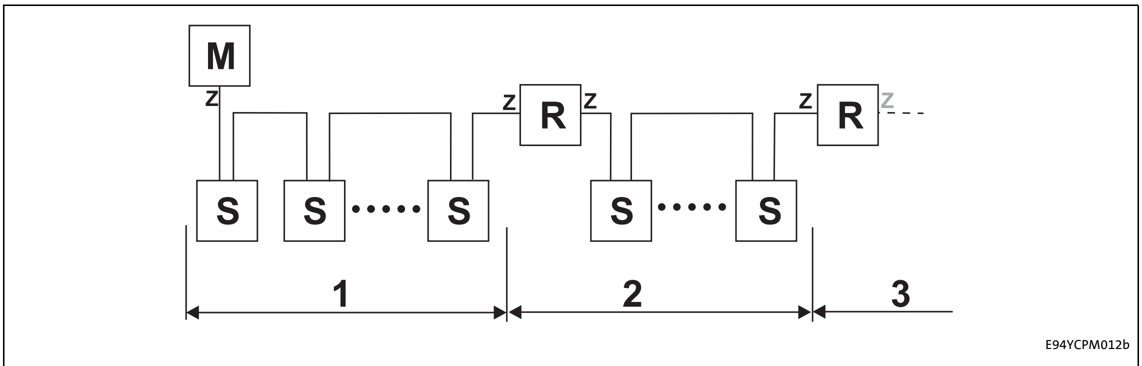
Every segment of the network must be terminated at both ends. The bus terminators of the PROFIBUS are marked with a "Z" in the below examples.

In the case of an RS485 network of only one segment, the PROFIBUS master (M) with the integrated bus terminator starts the segment while the bus terminating resistor in the connector of the last PROFIBUS station (S) must be activated.



[5-4] RS485 network with one segment

An RS485 network consisting of several segments contains repeaters (R) for coupling the segments. The repeaters are provided with integrated bus terminating resistors.



[5-5] RS485 network with a repeater

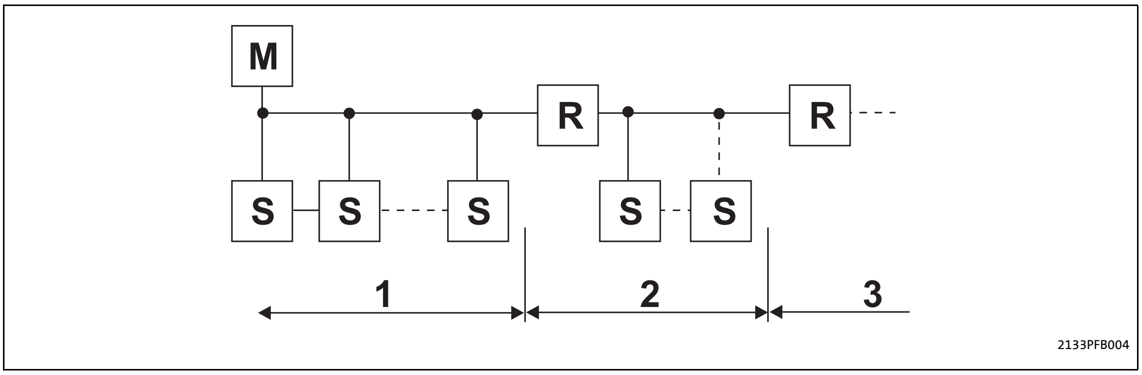
If no repeater is to be used at the end of the segment, the bus terminating resistor must be activated in the connector of the last device. The bus termination is supplied by the station itself.

Stop!

The bus terminator must always be supplied. Otherwise, the bus can get unstable.

▸ Activating the bus terminating resistor (📖 27)

Number of stations



[5-6] Number of stations

Segment	Master (M)	Slave (S)	Repeater (R)
1	1	31	-
	2	30	-
2	-	30	1
3	-	30	1

Tip!

Repeaters do not have a station address. When calculating the maximum number of stations, they reduce the number of stations by 1 on each side of the segment.

Repeaters can be used to build up line and tree topologies. The maximum total bus system expansion depends on ...

- the baud rate used;
- the number of repeaters used.

5.3 Activating the bus terminating resistor

The PROFIBUS must be terminated by a bus terminating resistor at the first and last physical bus station.

The bus terminating resistor in the bus connector of the bus cable is activated by means of a switch.

PROFIBUS cables with integrated bus terminating resistor are offered by several cable manufacturers.

Note!

If you want to disconnect individual bus stations, ensure that the bus terminators at the cable ends remain active.

Please observe that the bus termination is not active any longer if ...

- the bus connector has been disconnected;
- the voltage supply of the Inverter Drive 8400 has been switched off.

5.3.1 Bus cable specification

Note!

Only use cables which meet the listed specifications of the PROFIBUS user organisation.

Area	Values
Cable resistance	135 ... 165 Ω/km, (f = 3 ... 20 MHz)
Capacitance per unit length	≤ 30 nF/km
Loop resistance	< 110 Ω/km
Core diameter	> 0.64 mm
Core cross-section	> 0.34 mm^2
Cores	Twisted in pairs, insulated and shielded

Bus cable length

The length of the bus cable depends on the baud rate and cable type used. The data in the following table applies to PROFIBUS cables of "FC-Standard Cable" cable type .

Baud rate	Length
9.6 ... 93.75 kbps	1200 m
187.5 kbps	1000 m
500 kbps	400 m
1500 kbps	200 m
3000 ... 12000 kbps	100 m

Note!

The baud rate depending of the data volume, cycle time and number of stations should only be selected as high as required for the application.

Tip!

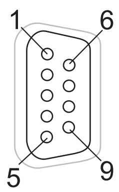
We recommend taking the use of optical fibres into consideration for high baud rates.

Advantages of optical fibres:
- External electromagnetic interferences have no effect on the transmission path.
- Bus lengths of several kilometres are also possible with higher baud rates.
- The bus length is ...
  - independent of the baud rate;
  - dependent on the optical fibre used.

5.3.2 PROFIBUS connection

The 9-pole Sub-D socket **X201** serves to connect the communication module to the bus system.

Assignment of the 9-pin Sub-D socket X201

View	Pin	Assignment	Description
	1	Not assigned	-
	2	Not assigned	-
	3	RxD/TxD-P	Data line B (received data/transmitted data, plus)
	4	RTS	Request To Send (received data/transmitted data, no differential signal)
	5	M5V2	Data ground (ground to 5 V)
	6	P5V2	5 V DC / 30 mA (bus termination)
	7	Not assigned	-
	8	RxD/TxD-N	Data line A (received data/transmitted data, minus)
	9	Not assigned	-

6 Commissioning

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

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

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

▶ Parameter reference (📖 95)

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.
  ▶ Activating the bus terminating resistor (📖 27)

6.2 Configuration of the controller (master)

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

Configuration for the controller (master) and the DP-V0 parameter data channel

For configuring the PROFIBUS you must read the device description file of the communication module into the master.

The device description file for the E84AYCPM communication module (PROFIBUS) can be found in the Download area at:

www.Lenze.com

The following language variants of the device description file can be used:

- LENZ0A89.GSD (source file, English)
- LENZ0A89.GSG (German)
- LENZ0A89.GSE (English)

Defining the user data length

The user data length is defined during the initialisation phase of the master.

The Inverter Drives 8400 support the configuration of a maximum of 16 process data words (max. 32 bytes). The optional activation of the cyclic parameter data channel additionally occupies 4 process data word (8 bytes).

The user data lengths for process input data and process output data are the same.

Description of the device data base file

Selection text	Parameter data with consistency	Process data		Assigned IO memory
		with consistency	without consistency	
DRIVECOM-PAR (cons) + PCD (**n**W cons)	Yes	**n** words	-	**4** + **n** words
PCD (**n**W cons)	-	**n** words	-	**n** words
PCD (**n**W)	-	-	**4** words	**4** words
n = 1 ... 16 process data words				

Example of selecting the device data base file

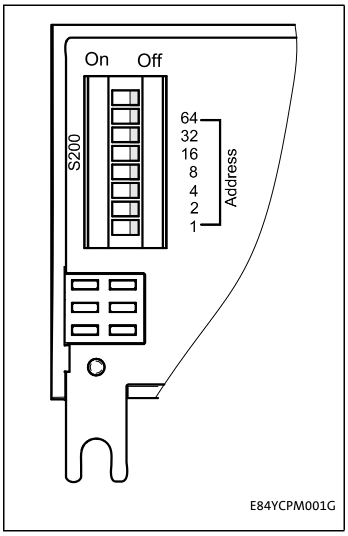
```
DRIVECOM-PAR (cons) + PCD (8W cons)
```

- "Drivecom-PAR (cons)" = DP-V0 parameter data channel (4 words)
- "PCD (8W cons)" = 8 process data words

Tip!

A detailed description of consistency is given in the chapter "Consistent parameter data" (📖 80).

6.3 Setting the station address



[6-1] DIP switch

The station address can be set via DIP switches **1 ... 64** or via the »Engineer« (code C13899).

The unlabelled DIP switch (topmost position) does not have any function.

Lenze setting: all switches in OFF position

The station addresses must differ from each other if several networked PROFIBUS stations are used.

The station address can be set via DIP switches **1 ... 64** or via the »Engineer« (code C13899).

	Setting the station address via ...	
	DIP switch	**C13899**
Condition	At least one switch **1 ... 64 = ON**	• Switches **1 ... 64 = OFF** • All switches **1 ... 64 = ON** (invalid value "127")

The housing labelling indicates the valencies of the individual DIP switches for setting the station 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
Station address	**= sum of the valencies = 16 + 4 + 2 + 1 = 23** ▶ DIP switch positions for setting the station address (📖 107)						

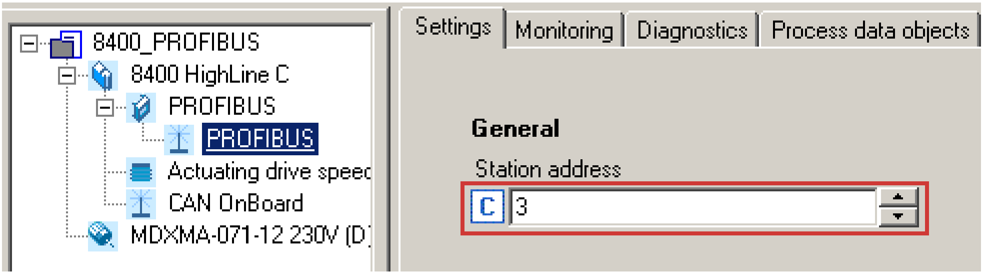
- Valid address range: 1 ... 126 (max. 126 slave stations)
- C13920: Display of the current address setting of the switches
- C13864: Display of the station address active on the PROFIBUS

Note!

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

Setting the station address via the »Engineer«

In the »Engineer«, the station address can be set via the **Settings** tab.



Impermissible addresses are displayed in red in the **Station address** (code C13899).

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

6.4 Initial switch-on

Documentation for the standard device

Observe the safety instructions and residual hazards stated.

Note!

Establishing communication

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

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

Protection against uncontrolled restart

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

In the Lenze setting of the Inverter Drives 8400, the restart protection is activated.

The restart behaviour of the inverter can be set via **C00142** ("Autostart Option"):

C00142 = 9 (Lenze setting)

- The inverter remains inhibited (even if the fault is no longer active).
- Bit 0 (inhibited at power-on) and bit 3 (inhibited at undervoltage) are set.
- The drive starts in a controlled mode by explicitly enabling the inverter: LOW-HIGH edge at digital input X4/RFR.

C00142 = 8 (enabled)

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

6.5 Going online with »Engineer« via TCI

Via Tool Calling Interfaces (TCI) you can connect to a TCI-capable integrated development environment and parameterise and diagnose your field devices without having to exit the integrated development environment.

You cannot set the TCI communication path directly in the »Engineer«. The selection is carried out by the »STEP7« Siemens software.

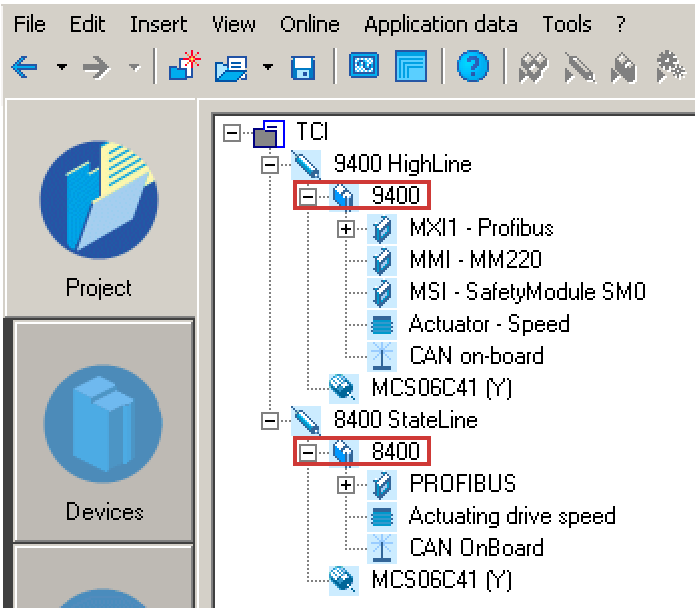
The TCI function requires a PN/DP-CPU. Information on the Siemens PLC types that are equipped with the TCI function is provided via the Siemens Support at:

http://support.automation.siemens.com

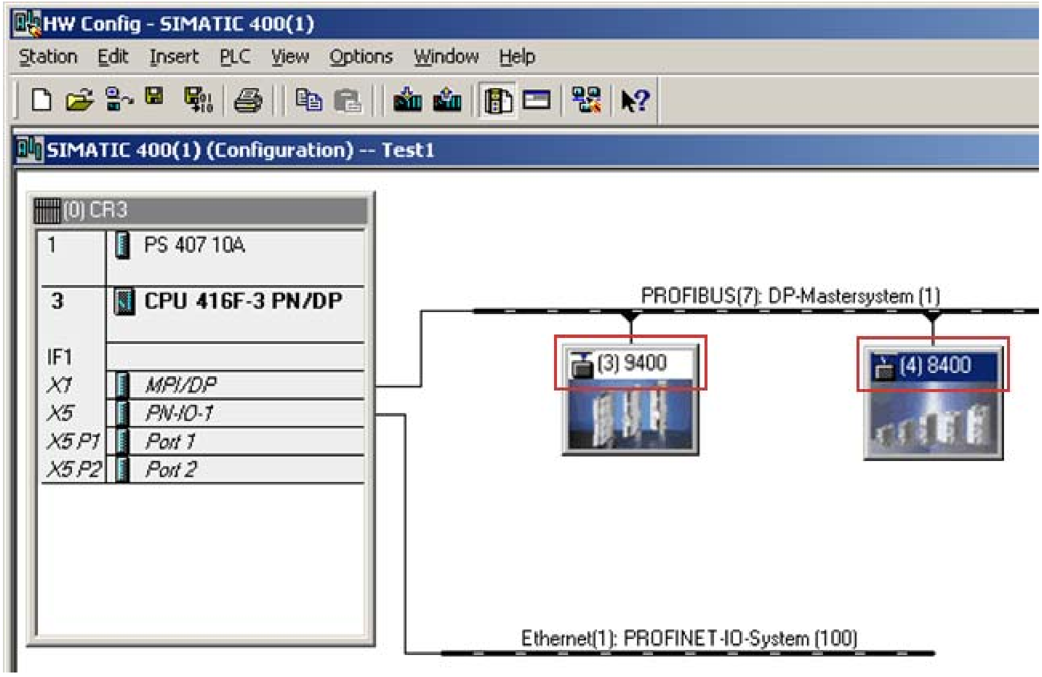
How to configure TCI communication:

1. Allocate names for the individual axes in the »Engineer« project.

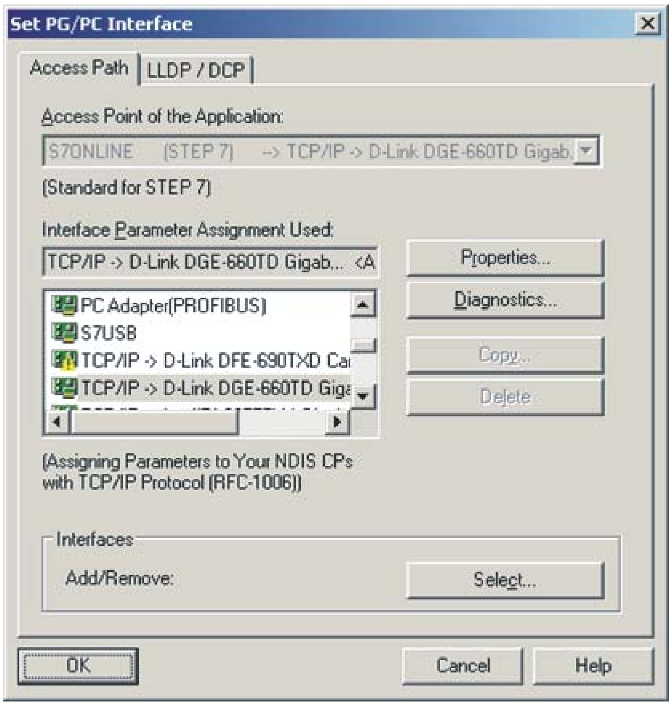
   In our case, "9400" was allocated for 9400 HighLine, and "8400" for 8400 StateLine:



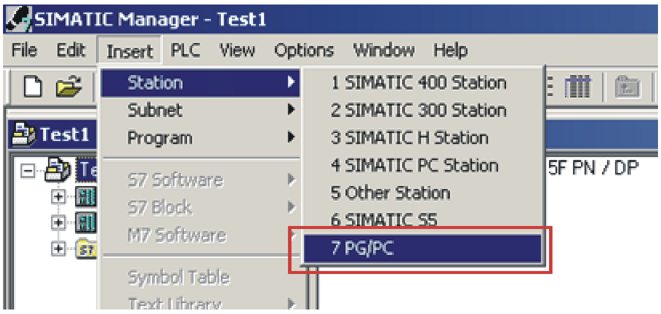
2. In »STEP7« in the »HW Konfig« ...
   - create the Lenze PROFIBUS stations with the corresponding station addresses and
   - create a PROFIBUS network.



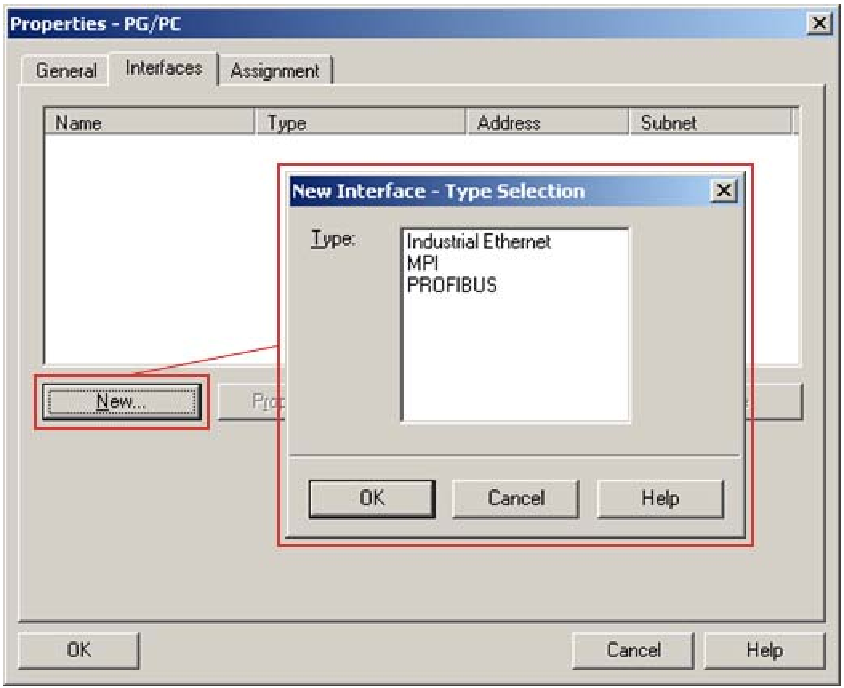
   Here a Servo Drive 9400 (address 3) and an Inverter Drive 8400 (address 4) are operated on the PROFIBUS.
   - The names of the PROFIBUS slaves in the »HW Konfig« must be identical to those of the corresponding Lenze axes in the »Engineer« (here "9400" and "8400").
   - The selection of the process data configuration has no impact on TCI communication.
3. Establish an Ethernet connection to the PROFIBUS CPU.



4. Load the »STEP7« project to the CPU.
5. Use the menu command **Insert → Station → 7 PG/PC** to integrate a PG/PC station into the »STEP7« project.

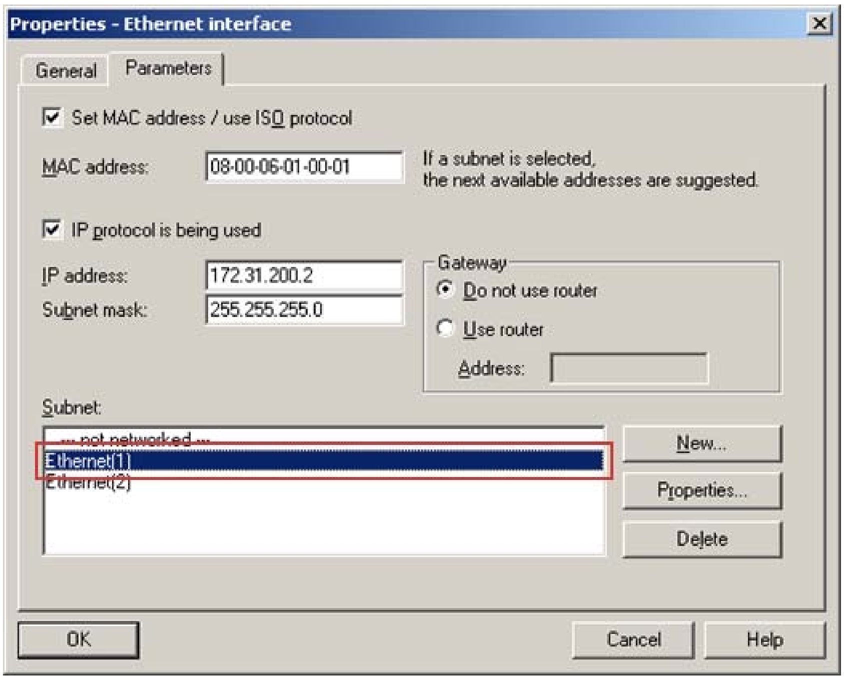


6. By double-clicking the PG/PC station inserted, open its "Properties" dialog.
7. Under the **Interfaces** tab, select a new Ethernet interface and confirm the selection with **OK**.



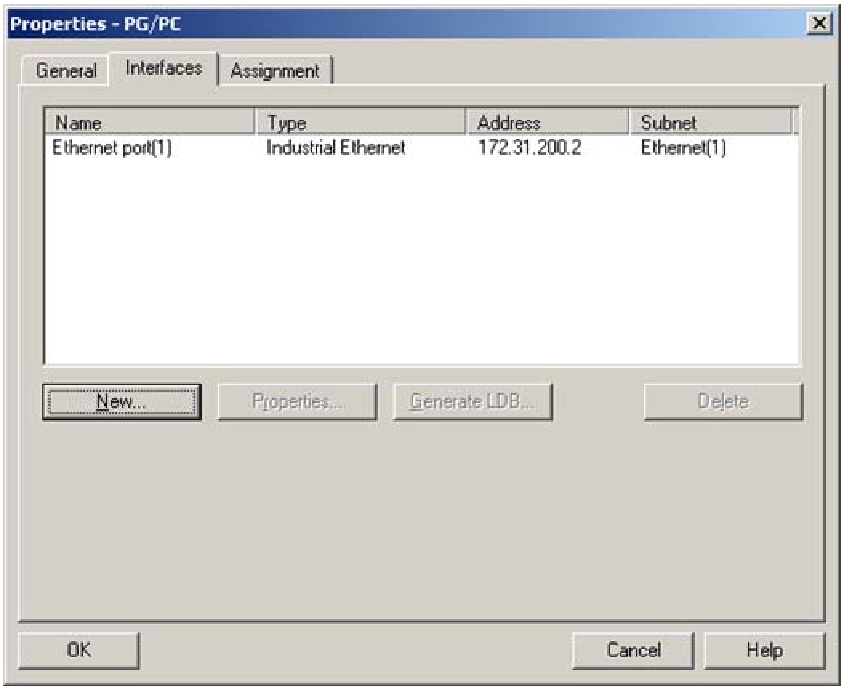
8. Select the Ethernet connection which you are using to go online with »STEP7« (the same Ethernet connection that has been configured in the »HW Konfig«).

   In our case this is the Ethernet(1) connection:

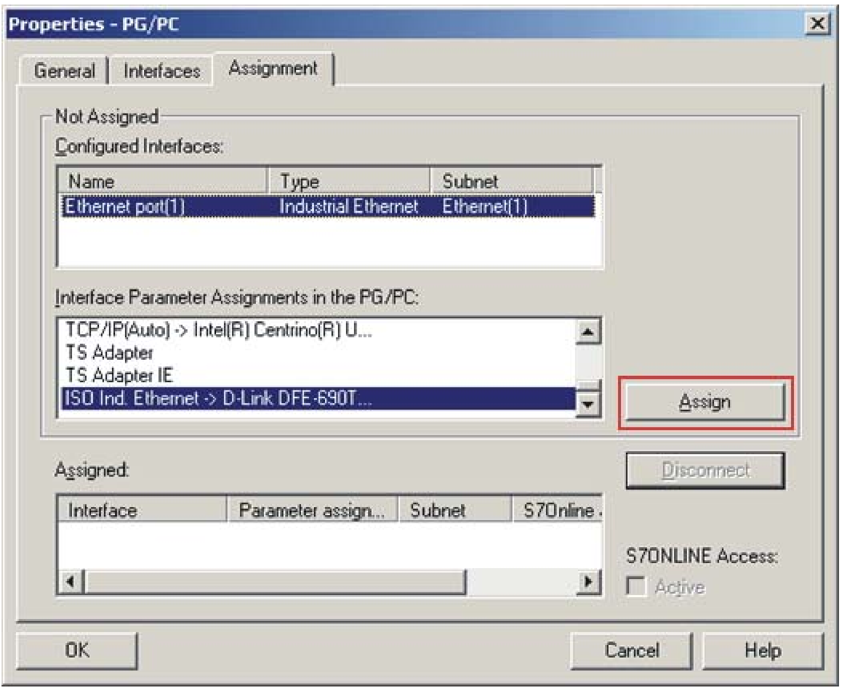


9. Confirm the selection with **OK**.

   The connection has been accepted.

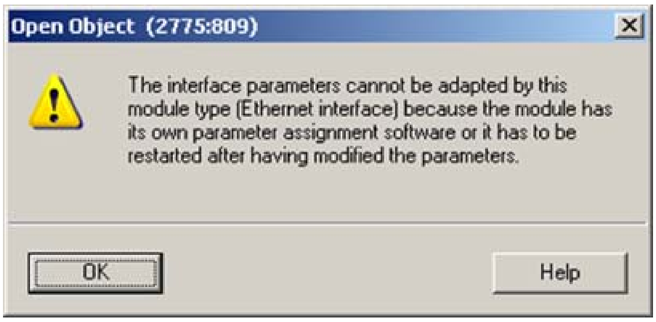


10. Select the actual PG/PC connection under the **Assignment** tab.

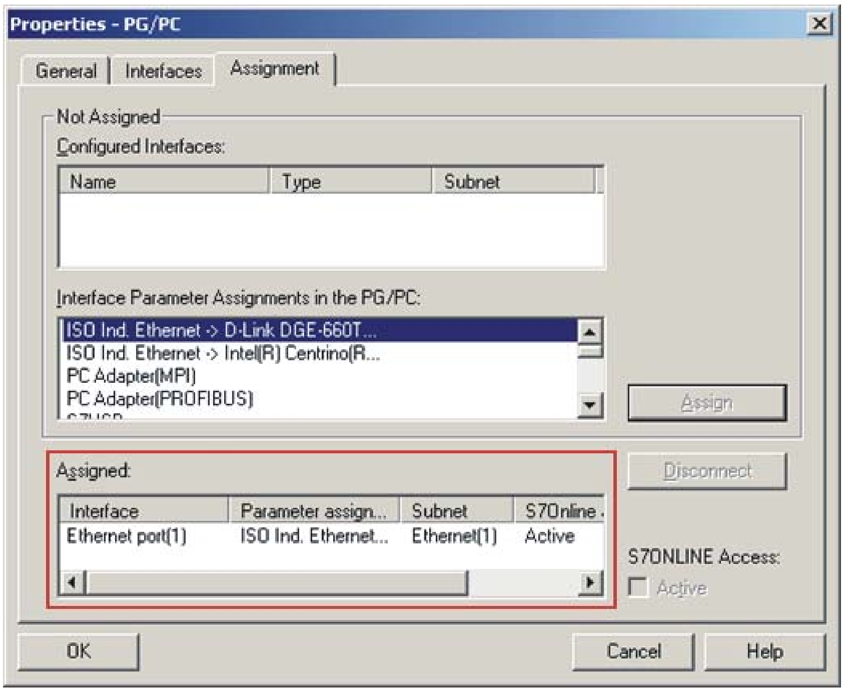


The connection highlighted is assigned by means of the **Assign** button.

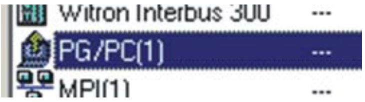
11. Confirm the following message with **OK**.



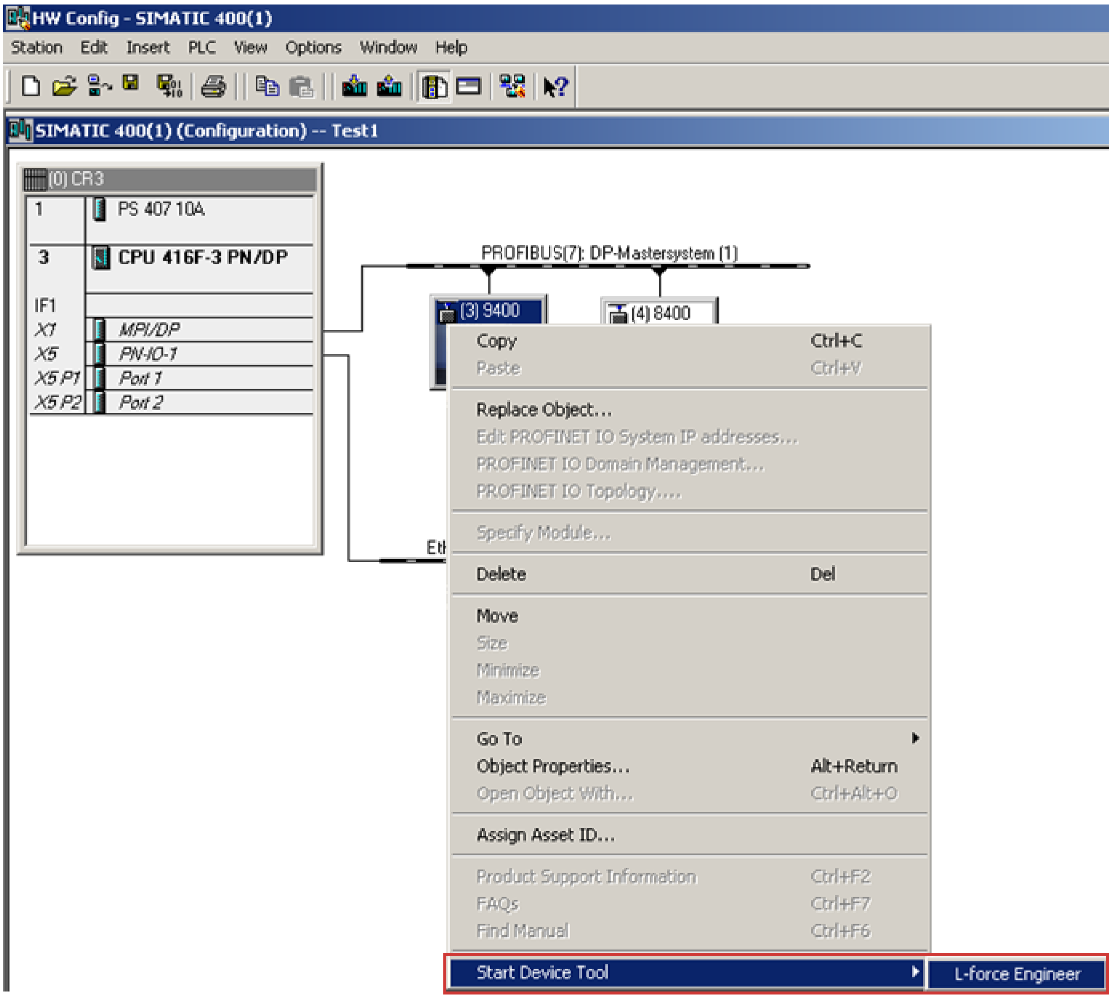
12. After the assignment, the connection appears in the "Assigned" display area. Close the dialog with **OK**.



13. In the »STEP7« project, the PG/PC station is marked with a yellow arrow. (The connection selected is active.)



14. Start the transfer of the TCI communication parameters in the »HW Konfig« using the right mouse button and the menu command **Start Device Tool→ L-force Engineer**.

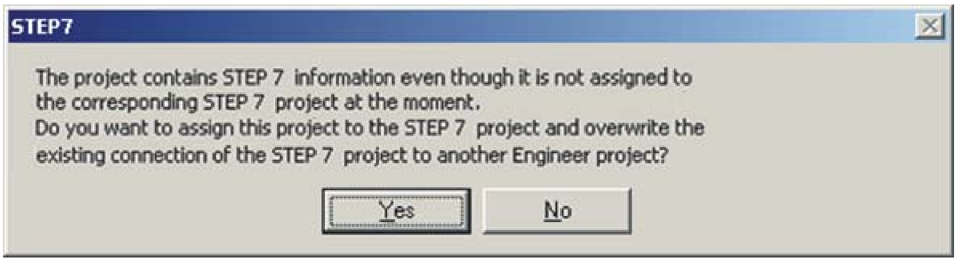


15. If the »Engineer« has already been started with the applicable project, the following message will appear:

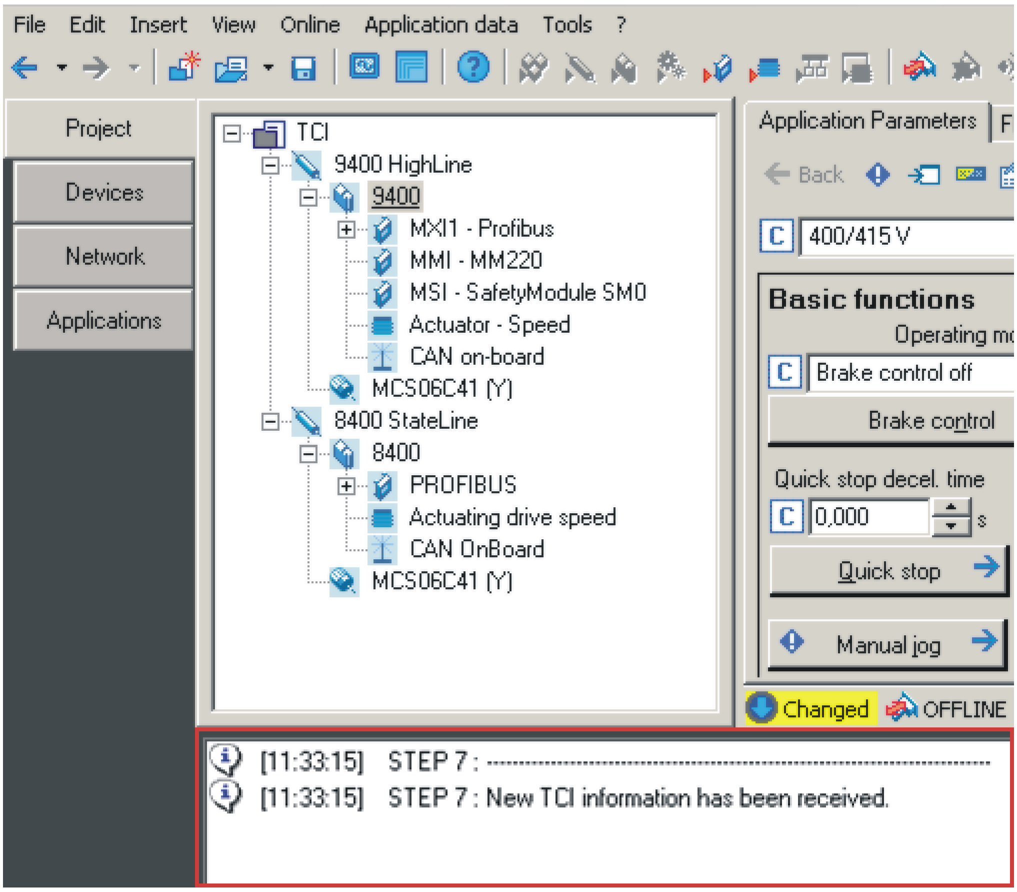
- The message says that the »Engineer« project is not set to a TCI communication path and provides information about whether this action is to be executed now.
- If you confirm the message with **Yes**, the applicable TCI communication parameter settings of the »STEP7« project are transferred to the »Engineer«.

If the »Engineer« has not already been started, it is started automatically now and you have to open the applicable project.

If the project selected has not been set to a TCI communication path yet, this can now be executed by clicking **Yes**:

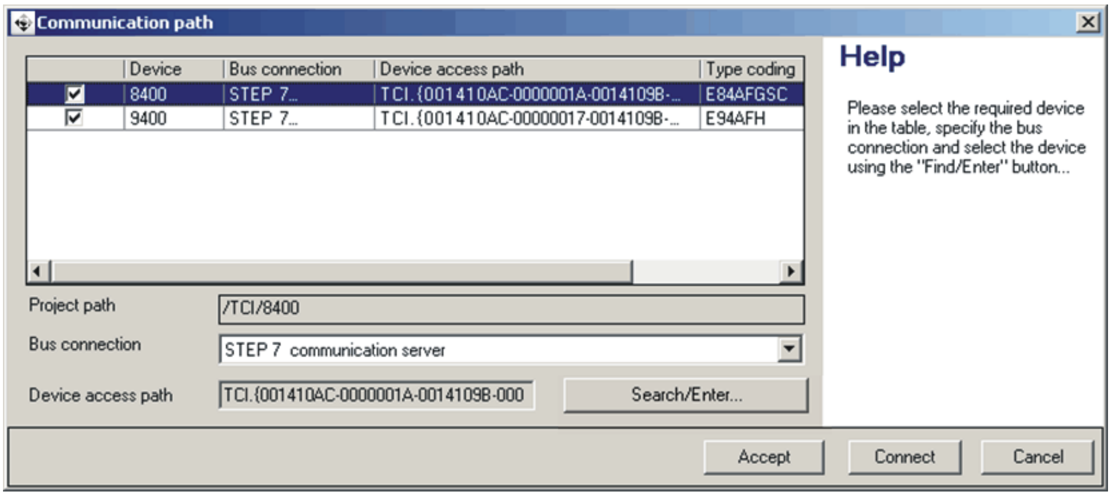


16. The transfer of the TCI communication parameters is documented in the »Engineer« message window.



Here the communication settings have been carried out successfully. The individual PROFIBUS addresses in the respective codes have been adapted to the »STEP7« project.

17. If you now call the "Go online" function of the »Engineer«, the TCI communication settings are displayed as follows:



- "STEP7 Communication Server" appears as bus connection.
- The device access path contains a very long string.
- Use the **Connect** button to establish an online connection.
- By means of the **Search/Enter** button, you can update the TCI communication parameters.

7 Data transfer

The PROFIBUS master and inverter communicate through the exchange of data telegrams via PROFIBUS. The user data area of the data telegram contains parameter data or process data. In the inverter, different communication channels are assigned to the parameter data and process data.

Communication channels

The process data channel serves to transfer process data.

- The process data serve to control the inverter.
- The controller (master) can directly access the process data. In the PLC, for instance, the data are directly saved to the I/O area.
- Process data are not saved in the inverter.
- Process data are transferred cyclically between the controller and the inverters (permanent exchange of current input / output data).
- Process data are, for instance, setpoints, actual values, control words and status words.
- The Inverter Drives 8400 can exchange a maximum of 16 process data word (16 bits/word) per direction.

Note!

Observe the direction of the information flow!
- Process input data (Rx data):
  - Process data from the inverter (slave) to the master
- Process output data (Tx data):
  - Process data from the master to the inverter (slave)

The parameter data channel serves to transfer parameter data.

- The parameter data channel provides access to all Lenze codes.
- In general, the parameter data transfer is not time-critical.
- Parameter data are, for instance, operating parameters, diagnostic information, and motor data.
- Parameter changes must be saved by means of code **C00002** of the Inverter Drive 8400.

8 Process data transfer

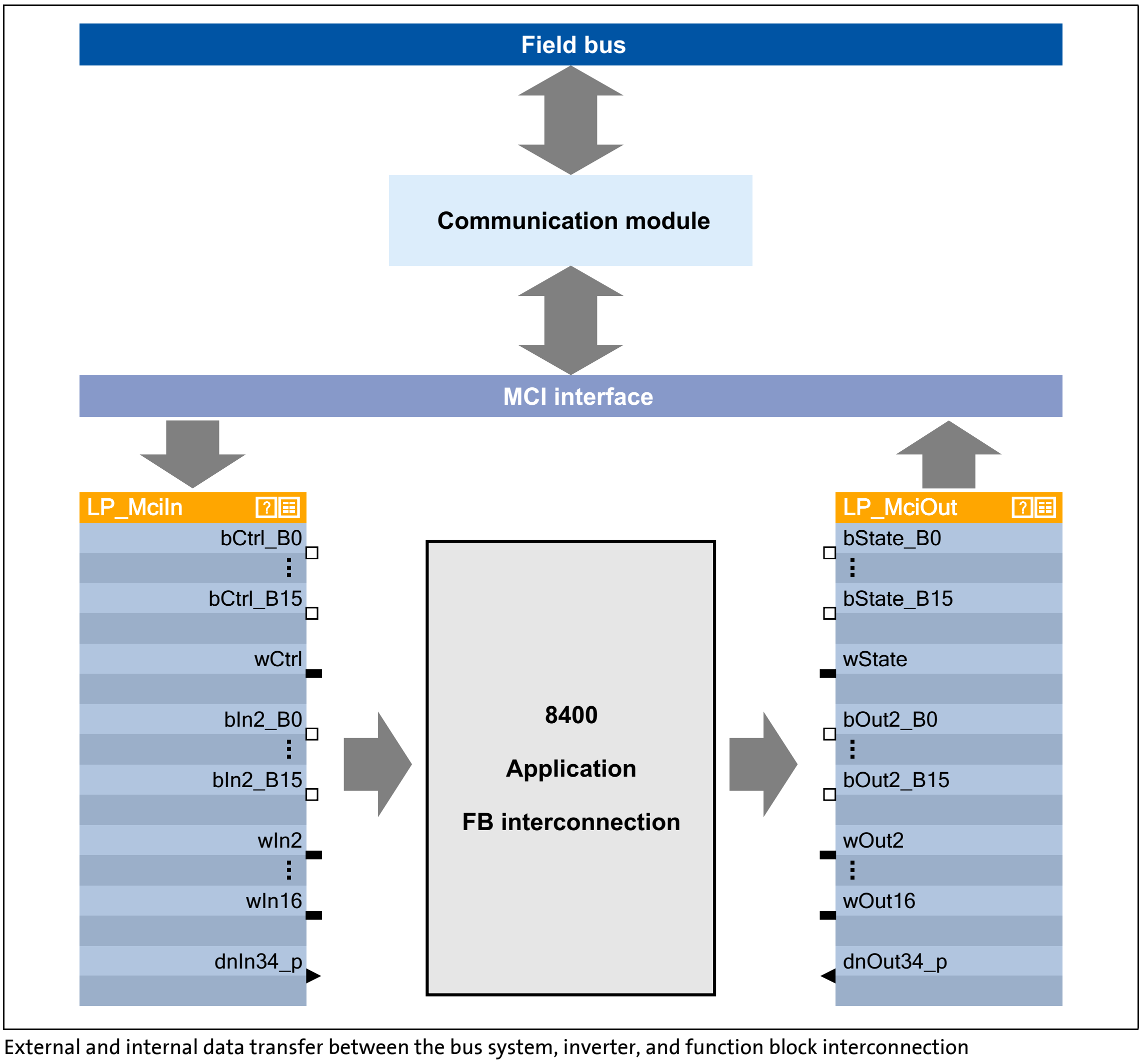
8.1 Access to process data / PDO mapping

Note!

PDO mapping and the objects required for this purpose are not supported in the software version 01.00 of the E84AYCPM communication module.

The process data (MCI-PDOs) are transferred via the MCI interface.

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



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

---

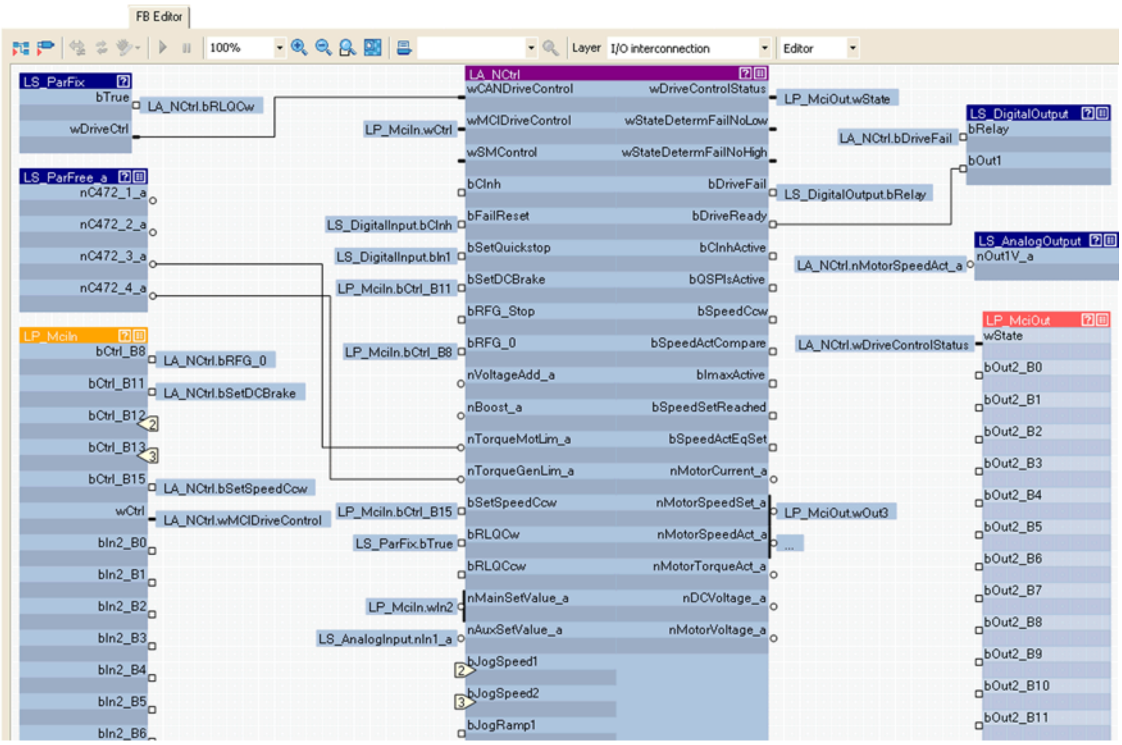
Software manual/ »Engineer« online help for Inverter Drives 8400

Here you'll find some detailed information about the port/function block interconnection in the »Engineer« and about port blocks.

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

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

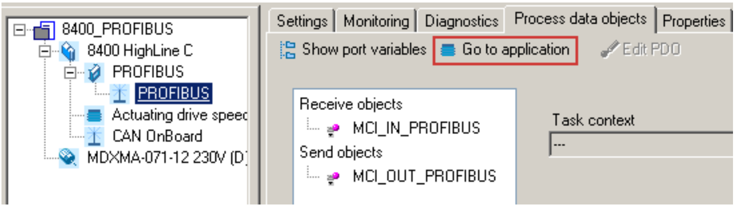
It is possible to display the port blocks "LP_MciIn" and "LP_MciOut" as well as the preconfigured interconnections in the »FB Editor« :



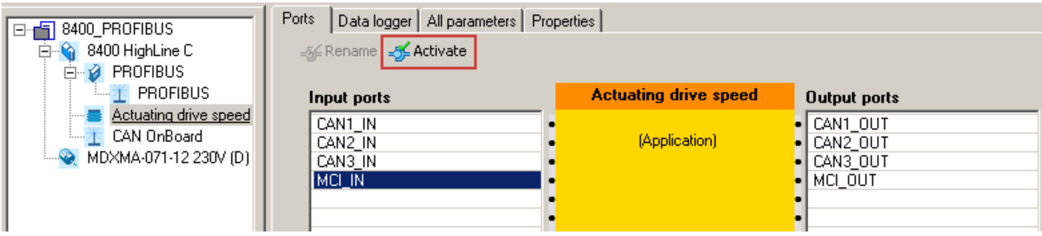
8.3 Free configuration of the port interconnection of process data objects (PDO)

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

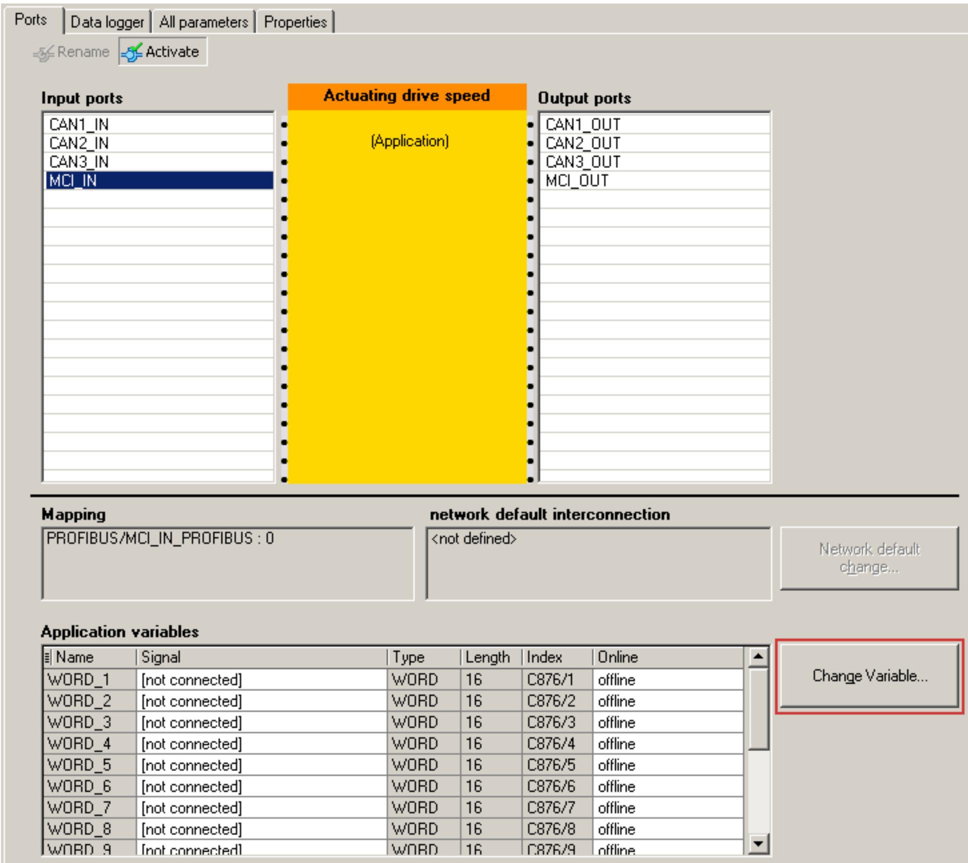
1. Go to the **Process data objects** tab and click **Go to application**.



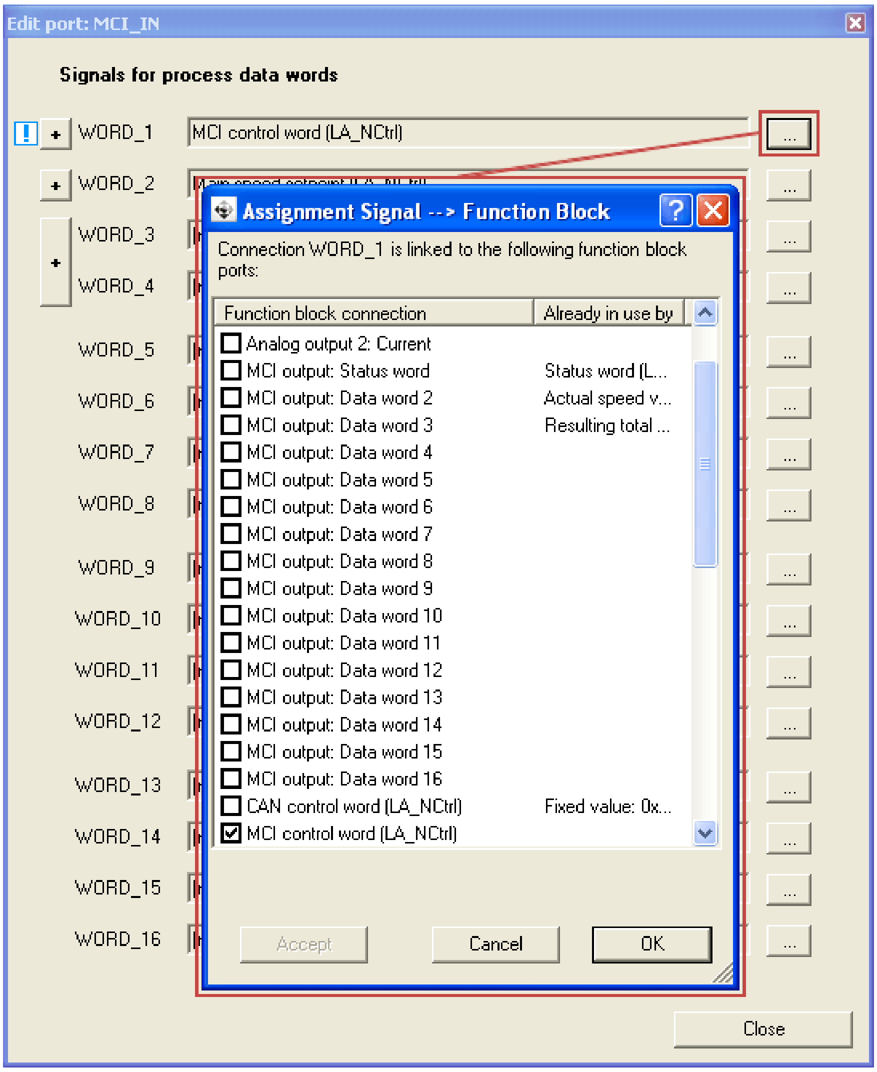
2. Go to the **Ports** tab and select the port block "MCI_IN" or "MCI_OUT" via mouse-click. Activate it by clicking **Activate**.



3. Click the **Edit port ...** button.

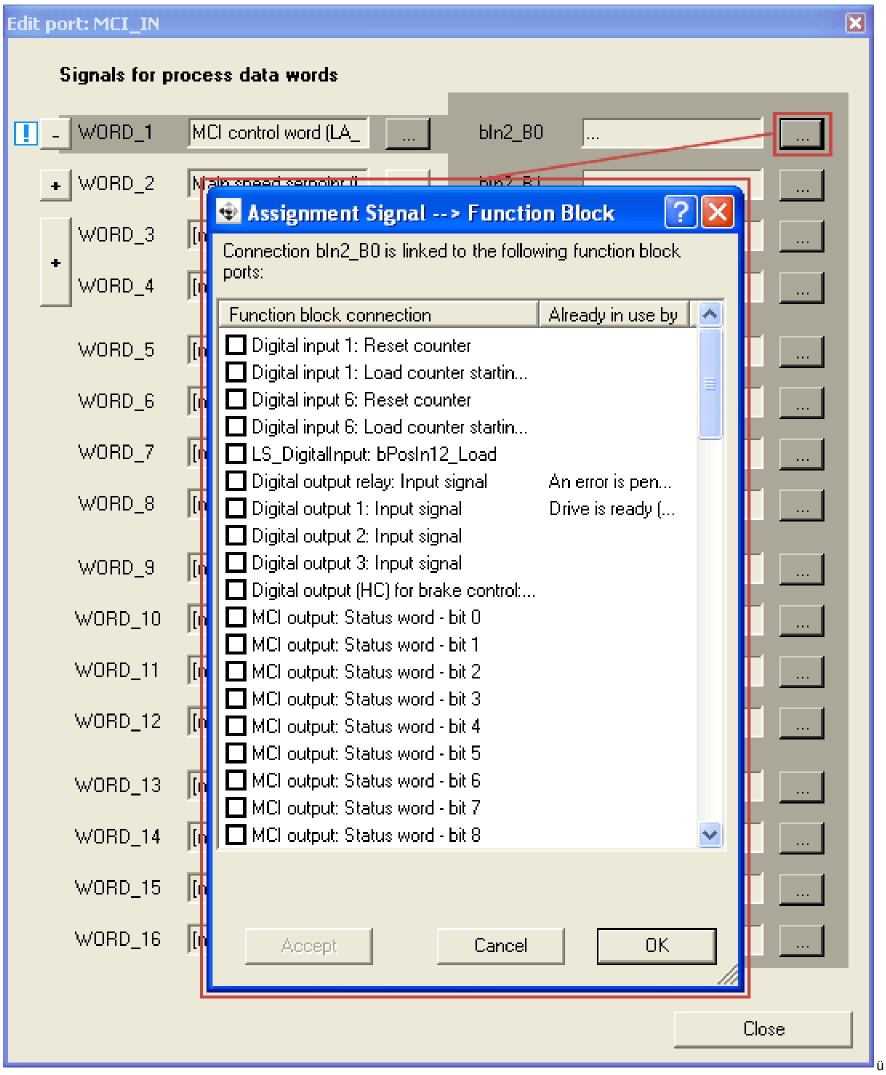


4. Via the [...] button, you can assign signals to the process data words in the *Assignment Signal --> Function Block* dialog box.
   → Select the signals and then confirm the selection with **OK**.



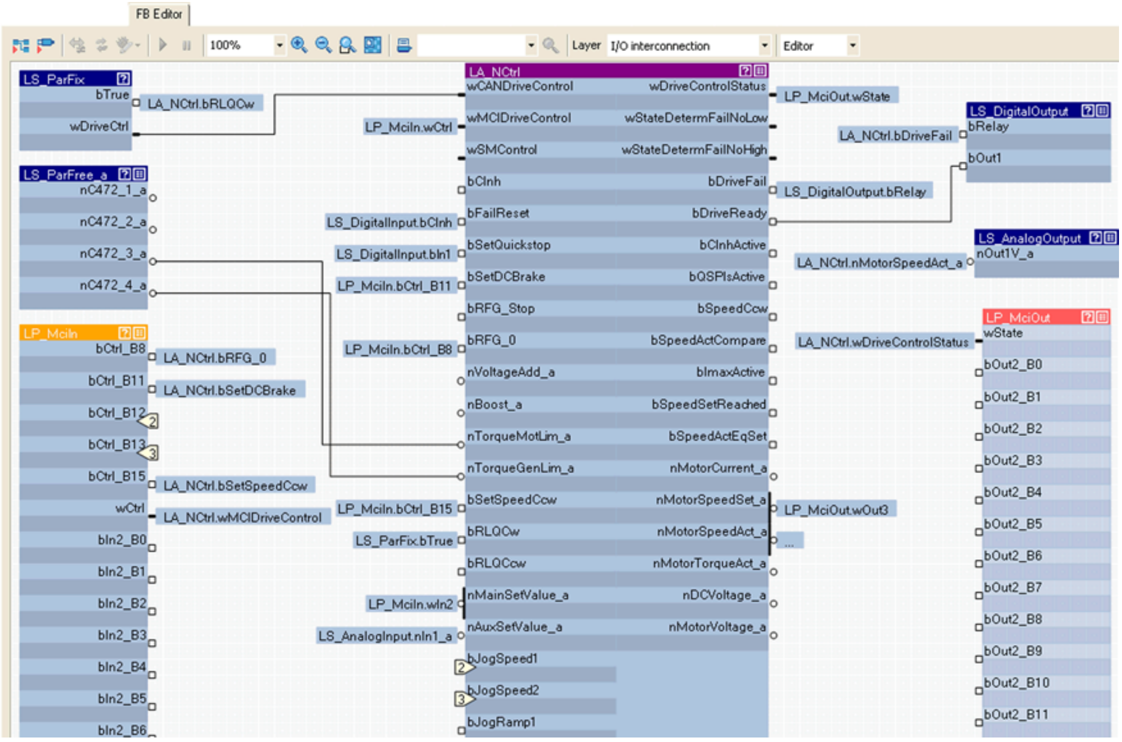
For the process data words WORD_1 and WORD_2, you can also assign signals to the individual control bits and status bits via the + and ... buttons.

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



Tip!

If the port blocks "LP_MciIn" and "LP_MciOut" are activated (see step 1), they are displayed in the »FB Editor«. Here you can also assign signals to the process data words.



9 Parameter data transfer

The E84AYCPM communication module supports the cyclic and acyclic transmission of parameter data:

- Cyclic DP-V0 parameter data are based on the DRIVECOM profile.
  If the DP-V0 parameter data channel is active, it additionally occupies 4 words of the input data and the output data.
- Acyclic DP-V1 parameter data are based on the PROFIdrive profile.

9.1 Addressing of the parameter data

The parameter data are addressed via codes which can be found in this documentation and in the corresponding documentation of your inverter.

▸ Parameter reference (📖 95)

Addressing of Lenze parameters

In the case of the DP-V0 parameter data channel, the parameters of a device are not addressed directly via Lenze code numbers, but via indices (bytes 3 + 4) and subindices (byte 2).

The conversion is made via an offset (24575 / 0x5FFF):

- PROFIBUS-DP $index_{dec}$ = 24575 - Lenze code number
- PROFIBUS-DP $index_{hex}$ = 0x5FFF - Lenze code $number_{hex}$

Example of C00105 (quick stop deceleration time):

- PROFIBUS-DP $index_{dec}$ = 24575 - 105 = 24470
- PROFIBUS-DP $index_{hex}$ = 0x5FFF - 0x69 = 0x5F96

The parameter values are entered into the user data (bytes 5 to 8) of the telegram.

9.2 DRIVECOM parameter data channel (DP-V0)

The DRIVECOM parameter data channel (DP-V0) ...

- enables parameter setting and diagnosing of the inverter.
- provides access to all Lenze parameters (codes).
- additionally occupies 4 words (16 bits/word) of the input and output data words in the master.
- is identical for both transmission directions.

9.2.1 Telegram structure (overview)

The telegram of the parameter data channel consists of a total of 8 bytes:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

The individual bytes are described in detail in the following subchapters.

9.2.2 Byte 1: Service

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

Request and response control for the parameter data channel

7	6	5	4	3	2	1	0

[9-1] Method of counting for bits 0 ... 7

Bit 0 ... 2: Request	
Read/write request from the master to the inverter	
000	No request
001	Read request ▸ Reading parameter data from the inverter (54)
010	Write request (write data to the inverter) ▸ Writing parameter data to the inverter (54)
100	Data transfer abort by the master ▸ Data transfer abort by the master (55)

Bit 3
Reserved

Bit 4/5: Data length	
Data length ≤ 4 bytes in the telegram bytes 5 ... 8 (data 1 ... 4 / error 1 ... 4)	
00	1 byte
01	2 bytes
10	3 bytes
11	4 bytes

Bit 6: Handshake
Indicates a new request. • The state of this (toggle) bit is changed by the master for every new request. • The inverter copies the bit into its response telegram.

Bit 7: Status	
Status information from the inverter to the master when sending the request confirmation. This status bit informs the master whether the request has been carried out without errors.	
0	Request completed without errors.
1	Request not completed because of an error. • The set status bit indicates that the telegram is an "error telegram". The data of bytes 5 ... 8 (data/error) must be interpreted as an error message. ▸ Error codes (58)

9.2.2.1 Reading parameter data from the inverter

General procedure:

1. Define the user data area of the inverter, i.e. define the location of the DP user data in the controller (observe manufacturer-specific information).
2. Enter the address of the required parameter in the "Index" and "Subindex" fields (DP output data).
3. Request in the service byte = read request.
   The handshake bit in the service byte must be changed (DP output data).
4. Check whether the handshake bit in the service byte is the same for the DP input data and the DP output data.
   - If the handshake bit is the same, the response has been received.
   - It is useful to implement a time monitoring tool.
5. Check whether the status bit in the service byte is set:
   - Status bit is not set: The "Data/Error" field contains the required Parameter value (data) (📖 57).
   - Status bit is set: The read request has not been executed correctly. The "Data/Error" field contains the Error codes (📖 58).

9.2.2.2 Writing parameter data to the inverter

General procedure:

1. Define the user data area of the inverter, i.e. define the location of the DP user data in the controller (observe manufacturer-specific information).
2. Enter the address of the required parameter in the "Index" and "Subindex" fields (DP output data).
3. Enter the parameter value in the "Data/Error" field.
4. Request in the service byte = write request.
   The handshake bit in the service byte must be changed (DP output data).
5. Check whether the handshake bit in the service byte is the same for the DP input data and the DP output data.
   - If the handshake bit is the same, the response has been received.
   - It is useful to implement a time monitoring tool.
6. Check whether the status bit in the service byte is set:
   - Status bit is not set: The write request has been executed correctly.
   - Status bit is set: The write request has not been executed correctly. The "Data/Error" field contains the Error codes (📖 58).

9.2.2.3 Abort of data transfer by the inverter

The error telegram is used to abort the transfer.

- The error telegram is marked by a set status bit in the service byte.
- The telegram can either be the response to an "Initiate Read/Write Service" or to a "Read/Write Segment Service".

Inverter response in the event of an error:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1
1t110000	SIDX	IDXH	IDXL	Error Class	Error code	Additional Code High	Additional Code Low

9.2.2.4 Data transfer abort by the master

The master can use this error telegram to abort a running segment transmission.

- The error telegram is marked by a set status bit in the service byte.
- The service byte also contains the request code "4" (100_{bin}).
- Bit 4 and bit 5 in the service byte (data length) are without meaning.
- Additional information (subindex, index, error information) is not transmitted.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1txx0100	0	0	0	0	0	0	0

Inverter response in the case of correct execution:

The inverter confirms the error telegram of the master by also sending an error telegram.

- The error telegram is marked by a set status bit in the service byte.
- In the case of correct execution, the telegram contains the error information "0x00000000" in bytes 5 ... 8.
- Additional information (subindex, index) is not transmitted.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	SIDX	IDXH	IDXL	Error Class	Error code	Additional Code High	Additional Code Low
1t110000	0	0	0	0	0	0	0

9.2.3 Byte 2: Subindex

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	**Subindex**	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

Additional addressing via the subindex is required for those codes of the Inverter Drives 8400 that contain a subcode (see code table).

9.2.4 Bytes 3 + 4: Index

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	**Index High byte**	**Index Low byte**	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

The parameter (Lenze code) is selected via these two bytes according to the formula:

- **Index = 24575 - Lenze code number**

(Also see "Addressing of Lenze parameters" (📖 51))

Example:

The parameter C00105 (quick stop (QSP) deceleration time) is to be addressed:

- Index = 24575 - 105 = 24470 = 0x5F96
- The entries in bytes 3 + 4 for this example would be:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	**0x5F**	**0x96**	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

9.2.5 Bytes 5 ... 8: Parameter value / error information

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	**Data 4 / Error 4**	**Data 3 / Error 3**	**Data 2 / Error 2**	**Data 1 / Error 1**

The state of status bit 7 in the service byte determines the meaning of this data field:

Status bit	Meaning of bytes 5 ... 8
0	Bytes 5 ... 8 contain the parameter value (data 1 ... 4).
1	Bytes 5 ... 8 contain an error message (error 1 ... 4) due to an invalid access. ▸ Error codes (📖 58)

Parameter value (data)

Note!

Strings or data blocks cannot be transmitted.

Depending on the data format, the length of the parameter value is between 1 and 4 bytes.

Data are saved in the Motorola format, i.e. first the high byte (high word), then the low byte (low word):

Byte 5	Byte 6	Byte 7	Byte 8
High byte	Low byte	High byte	Low byte
High word		Low word	
Double word			

Principle for the assignment of bytes 5 ... 8 with parameter values of different lengths:

Byte 5	Byte 6	Byte 7	Byte 8
Parameter value (length 1)	0x00	0x00	0x00
Parameter value (length 2)		0x00	0x00
Parameter value (length 4)			

9.2.6 Error codes

The following error messages may appear:

Byte 8	Byte 7	Byte 6	Byte 5	Meaning
Error 1	Error 2	Error 3	Error 4	
0x06	0x03	0x00	0x00	No right to access
0x06	0x05		0x11	Invalid subindex
0x06	0x05		0x12	Data length too large
0x06	0x05		0x13	Data length too small
0x06	0x07		0x00	Object does not exist
0x06	0x08		0x00	Data types do not comply with each other
0x08	0x00		0x00	Request cannot be executed
0x08	0x00		0x20	Request cannot be executed at the moment
0x08	0x00		0x22	Request cannot be executed due to the device status / The parameter can only be changed in the case of a controller inhibit
0x08	0x00		0x30	Value ranged exited
0x08	0x00		0x31	Parameter value too high
0x08	0x00		0x32	Parameter value too low
0x08	0x00		0x80	Hardware error

9.2.7 Telegram examples

9.2.7.1 Read request: Querying the heatsink temperature

The heatsink temperature of the inverter is to be read.

- Code to be read: C00061
- Heatsink temperature: 43 °C

Byte 1: Service (request)

Request = 0t110001$_{bin}$

- Bit 0 ... 2 = 001$_{bin}$ for read request
- Bit 3 = 0 (reserved)
- Bit 4/5 = 01$_{bin}$ for 2-byte data length (only relevant for the response telegram)
- Bit 6 = handshake bit (t ≡ status is changed in the response telegram)
- Bit 7 = status bit (only relevant for the response telegram)

Byte 2: Subindex

Subindex = 0 because code C00061 does not contain any subindices.

Bytes 3 + 4: Index

Index = 24575 - code number = 24575 - 61 = 24514 = 0x5FC2

- Byte 3 (high byte) = 0x5F
- Byte 4 (low byte) = 0xC2

Bytes 5 ... 8: Data

The response telegram contains the value of code C00061:

Data 3 + 4 = 43 [°C] x 1 (internal factor) = 43 = 0x002B

Result:

Request telegram from master to drive:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4	Data 3	Data 2	Data 1
0x01	0x00	0x5F	0xC2	0x00	0x00	0x00	0x00
0t000001$_{bin}$	00000000$_{bin}$	01011111$_{bin}$	11000010$_{bin}$	00000000$_{bin}$	00000000$_{bin}$	00000000$_{bin}$	00000000$_{bin}$
Waiting for change of handshake bit 6 in service byte 1 of the response.							

Response telegram from drive to master (for correct execution):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4	Data 3	Data 2	Data 1
0x11	0x00	0x5F	0xC2	0x00	0x2B	0x00	0x00
0t010001$_{bin}$	00000000$_{bin}$	01011111$_{bin}$	11000010$_{bin}$	00000000$_{bin}$	00101011$_{bin}$	00000000$_{bin}$	00000000$_{bin}$

9.2.7.2 Write request: Setting the deceleration time for quick stop (QSP)

In the inverter, the deceleration time for quick stop (QSP) is to be set to 50 ms.

- Code to be written: C00105

Byte 1: Service (request)

Request = $0t110010_{bin}$

- Bit 0 ... 2 = 010_{bin} for write request
- Bit 3 = 0 (reserved)
- Bit 4/5 = 11_{bin} for 4-byte data length
- Bit 6 = handshake bit (t ≡ status is changed in the response telegram)
- Bit 7 = status bit (only relevant for the response telegram)

Byte 2: Subindex

Subindex = 0 because code C00105 does not contain any subindices.

Bytes 3 + 4: Index

Index = 24575 - code number = 24575 - 105 = 24470 = 0x5F96

- Byte 3 (high byte) = 0x5F
- Byte 4 (low byte) = 0x96

Bytes 5 ... 8: Data

The parameter value of 0.05 s to be set is multiplied by the code-specific factor of "1000" and entered in the user data:

Data 1 ... 4 = 0.05 [s] x 1000 (internal factor) = 50 = 0x00000032

Result:

Request telegram from master to drive:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4	Data 3	Data 2	Data 1
0x72	0x00	0x5F	0x96	0x00	0x00	0x00	0x32
$0t110010_{bin}$	00000000_{bin}	01011111_{bin}	10010110_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00110010_{bin}
Waiting for change of handshake bit 6 in service byte 1 of the response							

Response telegram from drive to master (for correct execution):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4	Data 3	Data 2	Data 1
0x40	0x00	0x5F	0x96	0x00	0x00	0x00	0x32
$0t000000_{bin}$	00000000_{bin}	01011111_{bin}	10010110_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00110010_{bin}

9.3 PROFIdrive parameter data channel (DP-V1)

Data communication with PROFIBUS-DP-V0 is characterised by cyclic diagnostics and cyclic process data and parameter data transfer.

An optional service expansion is the acyclic parameter data transfer of PROFIBUS-DP-V1. This service does not impair the functionality of the standard services under PROFIBUS-DP-V0.

PROFIBUS-DP-V0 and PROFIBUS-DP-V1 can be operated simultaneously in the same network. This enables the step-by-step expansion or retrofitting of a system.

The services of PROFIBUS-DP-V1 can be used by the class 1 master (PLC) and the class 2 master (diagnostics master, etc.).

The integration of the acyclic service into the fixed bus cycle depends on the corresponding configuration of the class 1 master:

- With configuration, a time slot is reserved.
- Without configuration the acyclic service is *appended* when a class 2 master acyclically accesses a DP-V1 slave.

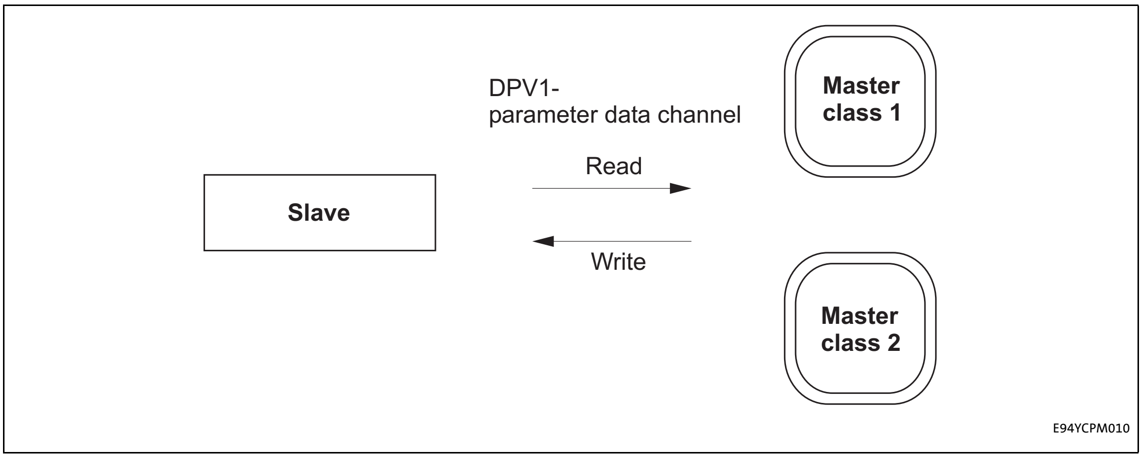
Features

- Parameter number and subindex addresses with a width of 16 bits each.
- Several parameter requests can be combined to one request (multi-parameter request).
- There is always only one parameter request in process (no pipelining).
- A parameter request/response must fit into a data block (max. 240 bytes). Requests/responses cannot be split into several data blocks.
- No spontaneous messages are transferred.
- There are only acyclic parameter requests.
- Profile-specific parameters can be read independently of the slave state.

9.3.1 Connection establishment between master and slave

A class 1 master can always request parameters from a slave if the slave is in the "Data_Exchange" state.

In addition to the class 1 master, a class 2 master can establish a communication connection to the slave:

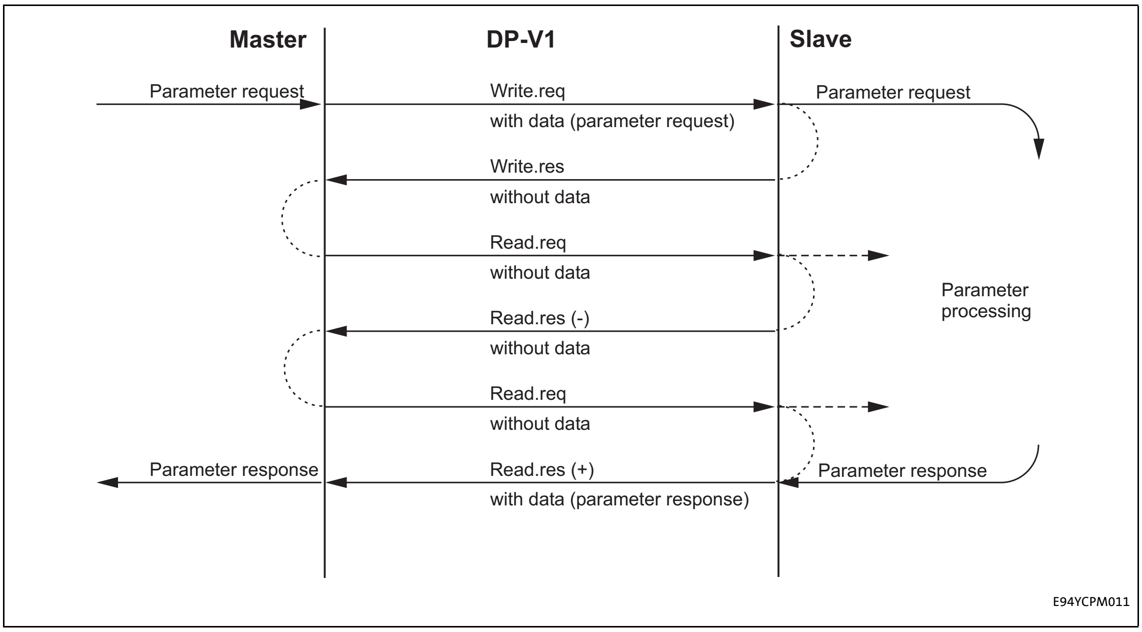


[9-2] Data communication via the DP-V1 parameter data channel

9.3.2 Acyclic data transfer

Note!

A parameter request refers to one or several parameter(s) (multi-parameter request).

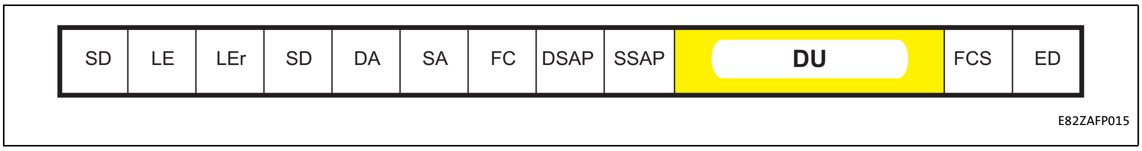


[9-3] Transmission directions

Explanation

- A "Write.req" is used to pass the data set (DB47) to the slave in the form of a parameter request.
- With "Write.res" the master receives the confirmation for the receipt of the message.
- The master requests the response of the slave with "Read.req".
- The slave responds with "Read.res (-)" if processing has not yet been completed.
- After parameter processing, the parameter request is completed by transmitting the parameter response to the master with "Read.res (+)".

9.3.3 Telegram structure



[9-4] PROFIBUS data telegram

The data unit (DU) contains the DP-V1 header and the parameter request or the parameter response.

The following subchapters describe the parameter request and the parameter response in detail.

Note!

The DP-V1 header consists of:
- Function identification
- Slot number
- Data set
- Length of the user data

Please refer to the corresponding PROFIBUS specification for further information on the DP-V1 header.

Assignment of the user data depending on the data type

Depending on the data type used, the user data are assigned as follows:

Data type	Length	User data assignment				
		Byte 1	Byte 2	Byte 3	Byte 4	Byte ...
String	x bytes					
U8	1 byte		0x00			
U16	2 bytes	High byte	Low byte			
U32	4 bytes	High word		Low word		
		High byte	Low byte	High byte	Low byte	

9.3.3.1 Reading parameter data from the inverter

Note!

- When a read request is processed, no parameter value is written to the slave.
- In the case of a multi-parameter read request, the parameter attribute, index, and subindex are repeated with the number "n" of the requested parameters.
- A read request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices

Field	Data type	Values
Request reference	U8	This value is specified by the master
Request identification	U8	0x01: Request parameters for reading
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter attribute

Byte 5	Byte 6
Attribute	Number of subindices

Field	Data type	Values
Attribute	U8	0x10: Value
Number of subindices	U8	0x00 (For array elements: Enter the number of array elements required.)

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Index	U16	0x0001 ... 0xFFFF (1 ... 65535)
Subindex	U16	0x0001 ... 0xFFFF (1 ... 65535)

9.3.3.2 Response to a correctly executed read request

Note!

Responses to a read request do not contain parameter attributes, indices and subindices.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x01: Parameter has been read
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter format

Byte 5	Byte 6
Format	Number of values

Field	Data type	Values
Format	U8	0x02: Integer8 0x03: Integer16 0x04: Integer32 0x05: Unsigned8 0x06: Unsigned16 0x07: Unsigned32 0x09: Visible string 0x0A: Octet string 0x40: Zero 0x41: Byte 0x42: Word 0x43: Double word
Number of values	U8	0x01 or number of requested subindices/parameters (with several subindices/parameters only the parameter value is repeated). In the case of string codes, the number of characters is entered here.

Parameter value

Byte 7	Byte 8	Byte 9	Byte 10
Value			

Field	Data type	Values
Value	String	Any (length > 4 bytes possible)
	U8	0x00 .... 0xFF
	U16	0x0000 .... 0xFFFF
	U32	0x0000 0000 .... 0xFFFF FFFF

9.3.3.3 Response to a read error

Note!

In the case of a multi-parameter request, correct and possible faulty messages are summarised in one telegram. They have the following data contents:

Correct message

- Format: data type of the value requested
- Number of values: as described in the chapter "Reading parameter data from the inverter" (65).
- Parameter value: value requested

Faulty message

- Format: 0x44
- Number of values: 0x01 or 0x02
- Error code without additional information (for number of values = 0x01) or
- Error code with additional information (for number of values = 0x02)

A faulty access to a parameter "n" is indicated at the nth position in the response telegram of a multi-parameter request.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x81: Parameter has not been read • The data in the bytes 7 + 8 must be interpreted as an error code.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter format

Byte 5	Byte 6
Format	Number of values

Field	Data type	Values
Format	U8	0x44: Error
Number of values	U8	0x01: Error code without additional information 0x02: Error code with additional information

Error code

Byte 7	Byte 8	Byte 9	Byte 10
Error code		Additional information (if available)	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Error code	U16	0x0000 .... 0xFFFF ▸ Error codes (📖 74)
Additional information (if available)	U16	

9.3.3.4 Writing parameter data to the inverter

Note!

When a multi-parameter write request is transferred, the ...
- Parameter attribute
- Index and subindex

and then the ...
- Parameter format
- Parameter value

... are repeated with the number "n" of the parameters addressed.

A write request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices

Field	Data type	Values
Request reference	U8	This value is defined by the master.
Request identification	U8	0x02: Write parameter
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters addressed)

Parameter attribute

Byte 5	Byte 6
Attribute	Number of subindices

Field	Data type	Values
Attribute	U8	0x10: Value
Number of subindices	U8	0x00 (For array elements: Enter the number of array elements required.)

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Index	U16	0x0001 ... 0xFFFF (1 ... 65535)
Subindex	U16	0x0001 ... 0xFFFF (1 ... 65535)

Parameter format

Byte 11	Byte 12
Format	Number of values

Field	Data type	Values
Format	U8	0x02: Integer8 0x03: Integer16 0x04: Integer32 0x05: Unsigned8 0x06: Unsigned16 0x07: Unsigned32 0x09: Visible string 0x0A: Octet string 0x40: Zero 0x41: Byte 0x42: Word 0x43: Double word
Number of values	U8	0x01 or number of written subindices/parameters (with several subindices/parameters only the parameter value is repeated). In the case of string codes, the number of characters is entered here.

Parameter value

Byte 13	Byte 14	Byte 15	Byte 16
Value			

Field	Data type	Values
Value	String	Any (length > 4 bytes possible)
	U8	0x00 .... 0xFF
	U16	0x0000 .... 0xFFFF
	U32	0x0000 0000 .... 0xFFFF FFFF

9.3.3.5 Response to a correctly executed write request

Note!

In the case of a multi-parameter request, correct and possible faulty messages are summarised in one telegram. They have the following data contents:

Correct message

- Format: 0x40
- Number of values: 0x00

Faulty message

- Format: 0x44
- Number of values: 0x01 or 0x02
- Error code without additional information (for number of values = 0x01) or with additional information (for number of values = 0x02)

A faulty access to a parameter "n" is indicated at the nth position in the response telegram of a multi-parameter request.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x02: Parameter has been written
Axis	U8	0x00 or 0x01
Number of indices	U8	0xn (n = number of parameter addressed)

9.3.3.6 Response to a write error

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x82: Parameter has not been written • The data in the bytes 7 + 8 must be interpreted as an error code.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters addressed)

Parameter format

Byte 5	Byte 6
Format	Number of values

Field	Data type	Values
Format	U8	0x44: Error
Number of values	U8	0x01: Error code without additional information 0x02: Error code with additional information

Error code

Byte 7	Byte 8	Byte 9	Byte 10
Error code		Additional information (if available)	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Error code	U16	0x0000 .... 0xFFFF ▸ Error codes (📖 74)
Additional information (if available)	U16	

9.3.4 Error codes

Error code	Description	Explanation	Additional information
0x0000	Impermissible parameter number	Access to unavailable parameter	-
0x0001	Parameter value cannot be changed	Change access to a parameter value that cannot be changed	Subindex
0x0002	Lower or upper value limit exceeded	Change access with value beyond the value limits	Subindex
0x0003	Faulty subindex	Access to unavailable subindex	Subindex
0x0004	No array	Access with subindex to non-indicated parameter	-
0x0005	Wrong data type	Change access with value that does not match the data type of the parameter	-
0x0006	No setting permitted (only resettable)	Change access with value unequal to 0 where this is not permitted	Subindex
0x0007	Description element cannot be changed	Change access to a description element that cannot be changed	Subindex
0x0008	Reserved	(PROFIdrive profile V2: PPO-Write requested in the IR is not available)	-
0x0009	Description data not available	Access to unavailable description (parameter value is available)	-
0x000A	Reserved	(PROFIdrive profile V2: Wrong access group)	-
0x000B	No parameter change rights	Change access without parameter change rights	-
0x000C	Reserved	(PROFIdrive profile V2: Wrong password)	-
0x000D	Reserved	(PROFIdrive profile V2: Text in the cyclic traffic cannot be read)	-
0x000E	Reserved	(PROFIdrive profile V2: Name in the cyclic traffic cannot be read)	-
0x000F	No text array available	Access to unavailable text array (parameter value is available)	-
0x0010	Reserved	(PROFIdrive profile V2: Missing PPO-Write)	-
0x0011	Request cannot be executed due to the operating state	Access is not possible due to temporary reasons not specified here	-
0x0012	Reserved	(PROFIdrive profile V2: Other error)	-
0x0013	Reserved	(PROFIdrive profile V2: date in the cyclic traffic cannot be read)	-
0x0014	Value impermissible	Change access with the value that is inside the value limits but not permissible for other permanent reasons (parameters with defined individual values)	Subindex
0x0015	Response too long	The length of the current response exceeds the maximum transmittable length	-
0x0016	Parameter address impermissible	Impermissible or non-supported value for attribute, number of subindices, parameter number, or subindex, or a combination	-
0x0017	Format impermissible	Write request: Impermissible or non-supported format of parameter data	-
0x0018	Number of values not consistent	Write request: Number of values of the parameter data do not match the number of subindices in the parameter address	-
0x0019 ... 0x0064	Reserved	-	-

Error code	Description	Explanation	Additional information
0x0065	Manufacturer-specific	-	-
...			
0x00FF			

9.3.5 Telegram examples

9.3.5.1 Read request: Querying the heatsink temperature

The heatsink temperature of the inverter is to be read.

- Code to be read: C00061
- Heatsink temperature: 43 °C

Parameter request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices
0xXX	0x01	0x00	0x01
	Request parameters for reading		

Byte 5	Byte 6
Attribute	Number of subindices
0x10	0x00
Value	No subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte
0x5F	0xC2	0x00	0x00
Index = 24575 - code no. = 24575 - 61 = 24514 = 0x5F C2		No subindex	

Parameter response to a correctly executed read request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices
0xXX	0x01	0x00	0x01
	Parameter has been read		

Byte 5	Byte 6
Format	Number of values
0x03	0x01
Integer16	1 value

Byte 7	Byte 8
Value	
High byte	Low byte
0x00	0x2B
Value read = 0x 00 2B = 43 x 1 (internal factor) = 43 [°C]	

Parameter response to a read error

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices
0xXX	0x81	0x00	0x01
	Parameter has not been read		

Byte 5	Byte 6
Format	Number of values
0x44	0x01
Error	Error code without additional information

Byte 7	Byte 8
Error code	
High byte	Low byte
For the meaning, see the "Error codes" (74) chapter	

9.3.5.2 Write request: Setting the deceleration time for quick stop (QSP)

In the inverter, the deceleration time for quick stop (QSP) is to be set to 50 ms.

Code to be written: C00105

Parameter request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices
0xXX	0x02	0x00	0x01
	Write parameter	Axis 0	1 index

Byte 5	Byte 6
Attribute	Number of subindices
0x10	0x00
Value	No subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte
0x5F	0x96	0x00	0x00
Index = 24575 - code no. = 24575 - 105 = 24470 = 0x5F 96		No subindex	

Byte 11	Byte 12
Format	Number of values
0x43	0x01
Double word	1 value

Byte 13	Byte 14	Byte 15	Byte 16
Value			
High word: high byte	High word: low byte	Low- word: high byte	Low word: low byte
0x00	0x00	0x00	0x32
Value to be written = 0.05 [s] x 1000 (internal factor) = 50 = 0x00 00 00 32			

Parameter response to a correctly executed write request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices
0xXX	0x02	0x00	0x01
	Parameter has been written		1 index

Parameter response to a read error

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices
0xXX	0x82	0x00	0x01
	Parameter has not been written		1 index

Byte 5	Byte 6
Format	Number of values
0x44	0x01
Error	Error code without additional information

Byte 7	Byte 8
Error code	
High byte	Low byte
For the meaning, see the Error codes (📖 74)	

9.4 Consistent parameter data

In the PROFIBUS communication system, data are permanently exchanged between the control system (CPU + PROFIBUS master) and the inverter via the plugged-on slave interface module. Both the PROFIBUS master and the CPU (central processing unit) of the control system access a joint memory: the dual port memory (DPM).

The DPM permits a data exchange in both directions (write/read):

It could happen that a slower PROFIBUS master writing would be overtaken by a faster CPU reading within a cycle time without any further data organisation.

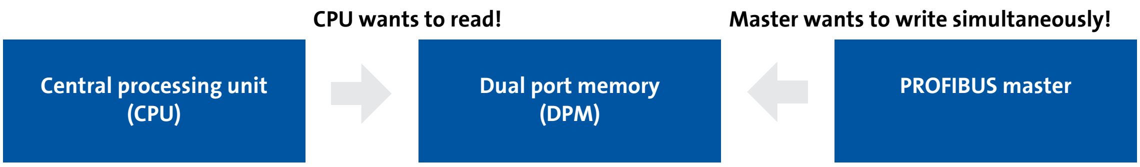
In order to avoid such an impermissible state, the parameter data to be transmitted must be marked as "consistent".

Data communication with consistent data

With consistency, either "reading" or "writing" is possible when the master and the CPU simultaneously access the memory:

- The PROFIBUS master transfers data only as a complete data set.
- The CPU can only access completely updated data sets.
- The PROFIBUS master cannot read or write data as long as the CPU accesses consistent data.

The result becomes clear from the example below:



1. As the master can only write if the CPU does not read, the master waits until the data are read completely by the CPU.
2. The master only writes a complete data set into DPM.

Configuring consistent data

Note!

Consistency is achieved by an appropriate PROFIBUS master configuration.

For this purpose, refer to the documentation for your configuring software.

10 Monitoring

10.1 Permanent interruption of PROFIBUS communication

If PROFIBUS communication is interrupted permanently, e.g. by cable breakage or failure of the PROFIBUS master, no process data are transmitted to the slave being in the "Data Exchange" state.

After the watchdog monitoring time determined by the master has expired, the response parameterised in C13880/1 is executed in the inverter (slave).

The process data are treated according to the setting in C13885. (The data sent last by the master can be used or can be set to zero.)

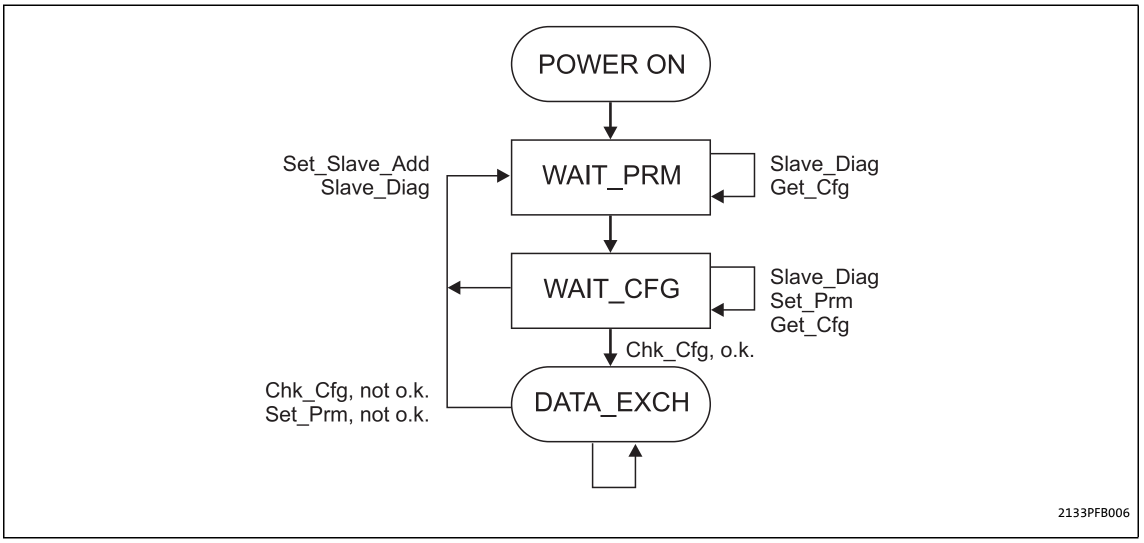
Preconditions for a inverter (slave) response

- A monitoring time of 1 ... 65534 ms for the "Data_Exchange" status (C13881) is set.
  A value of "65535 ms" (Lenze setting) deactivates the monitoring.
- A response for the slave is set in C13880/1 (Lenze setting "No response").
- The slave is in the "Data_Exchange" state.
- The watchdog monitoring time is configured correctly in the master.

If one of these preconditions is not met, the response to the absence of cyclic process data telegrams from the master is not executed.

▶ Settings and displays in the »Engineer« (📖 83)

10.2 Short-time interruption of PROFIBUS communication



[10-1] DP states (Decentralized Peripherals) for short-time interruption of communication

The master detects the communication fault and, only after a few microseconds, transfers the slave to the "WAIT_PRM" status of the DP state machine (see fig. [10-1]).

Only after the state chain of the DP state machine ending in the "Data_Exchange" state (DATA_EXCH) has been passed through, the watchdog monitoring time calculated for the slave (in milliseconds) continues to run.

Note!

The watchdog monitoring time does not continue running if the slave does not reach the "Data_Exchange" state due to repeated communication errors (e.g. caused by loose contact).

Additional monitoring for the data exchange

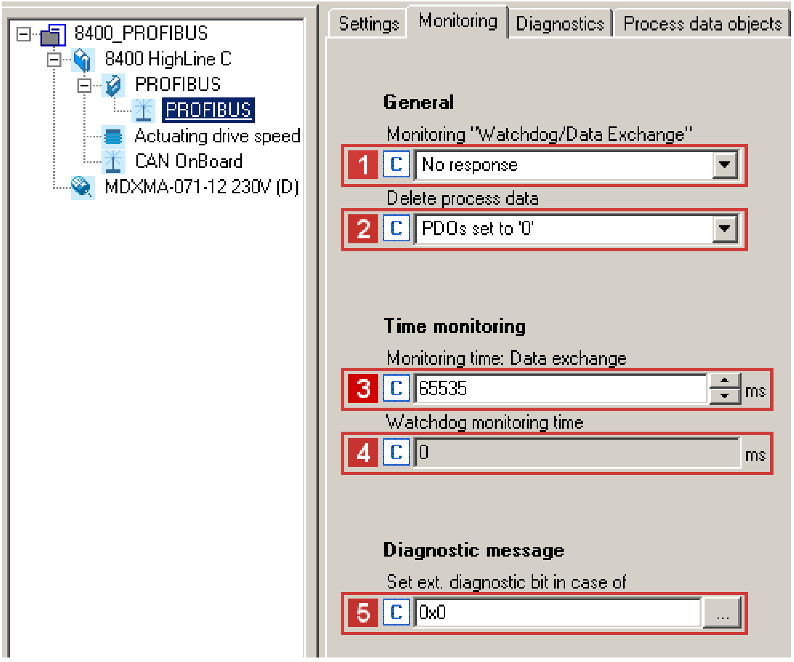
For this reason an additional monitoring function for the data exchange is available with C13881, which is activated when "Data_Exchange" is exited and the parameterised time (0 ... 65535 ms) has expired. The active monitoring triggers the response parameterised in C13880/1.

Note!

Observe the following condition for the time setting:

Monitoring time for the data exchange (C13881) ≤ watchdog monitoring time of the PROFIBUS (C13882/1).

10.3 Settings and displays in the »Engineer«



On the **Monitoring** tab of the »Engineer«, you can set or display the following parameters:

Parameter		Description
1	**Reaction on communication fault** (C13880/1)	The response set here takes place if the PROFIBUS station ... • does not receive a message from the master within the watchdog monitoring time (displayed in C13882/1) if there is an active connection; • recognises that it is not in the "Data_Exchange" status anymore. Please see also the information on 3.
2	**Clear process data** (C13885)	Selection of the process data which the inverter will process in the event of a PROFIBUS failure in order to maintain internal communication. The process data sent last by the master can be used or the process data can be set to zero.
3	**Monitoring time: Data exchange** (C13881)	After the monitoring time set here has elapsed, the response set in 1 takes place for the data exchange. • The value "65535" deactivates the monitoring function. • The monitoring time set here must be smaller than the watchdog monitoring time 4. • A change in monitoring is effective immediately. ▸ Permanent interruption of PROFIBUS communication (📖 81)
4	**Monitoring time: Watchdog** (C13882/1)	Display of the watchdog monitoring time determined by the PROFIBUS master • Monitoring starts with the receipt of the first telegram. • When a value of "0" is displayed, the monitoring function is deactivated. • A change in the watchdog monitoring time in the master is immediately effective. ▸ Permanent interruption of PROFIBUS communication (📖 81)
5	**Set ext. diagnostic bit upon** (C13886)	Bit-coded selection of the error responses in the standard device causing the external diagnostic bit ("diag bit") to be set (see PROFIBUS specification; bit 3 of byte 1 of the DP diagnostic messages). • The diagnostic bit is sent to the PROFIBUS master where it is evaluated separately. • The diagnostic bit is always set when a system error occurs. • The Lenze setting "0" means that the diagnostic bit is not set for the following error responses. • An advanced diagnostic message is always sent.

11 Diagnostics

For diagnosing faults of the PROFIBUS module, the LEDs on the front panel are provided. Furthermore you can query the current bus status via code C13861.

11.1 LED status displays

Note!

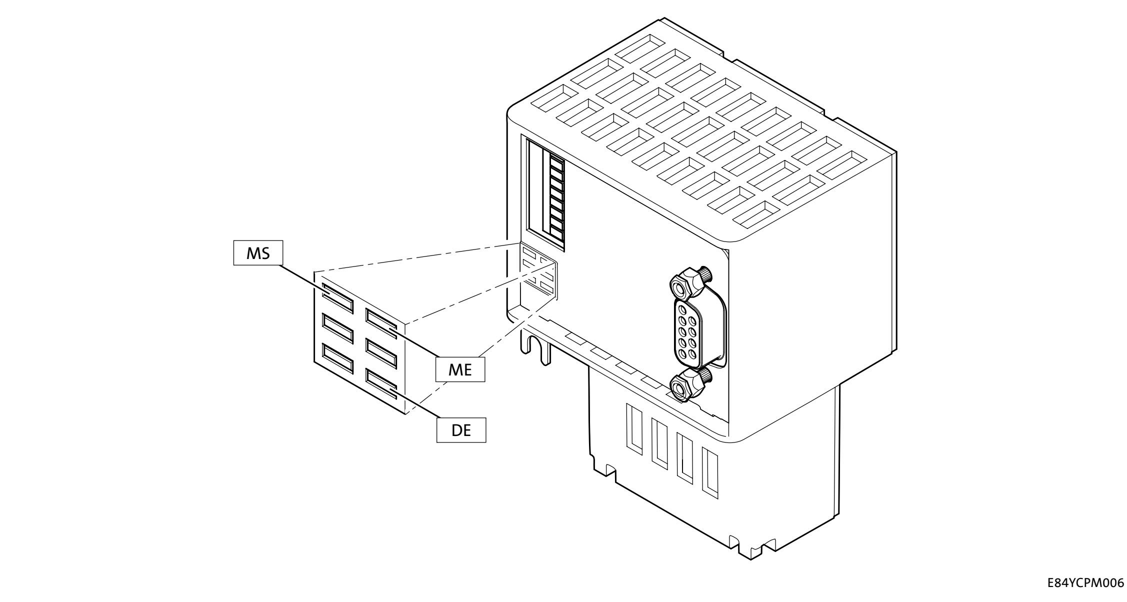
During normal operation, the LED **BS** (📖 86) blinks and the LED **MS** (📖 86) is lit permanently.

The following status displays are distinguished:

- Module status displays (📖 85)
- Fieldbus status displays (📖 86)

11.1.1 Module status displays

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

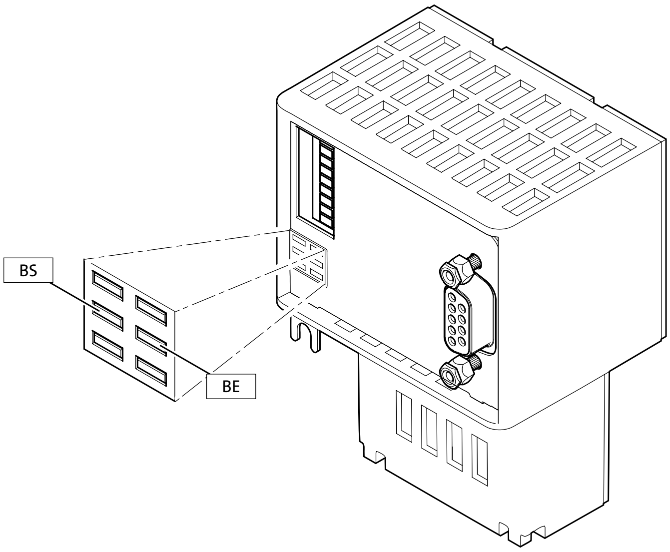


[11-1] LED status displays MS, ME, and DE

LED	Colour	State	Description
MS	Green	On	The communication module is supplied with voltage and has established a connection to the standard device.
		Blinking	 The communication module is supplied with voltage, but has not yet established a connection to the standard device. (Standard device is switched off, initialising or not present.)
ME	Red	On	An error concerning the communication module has occurred.
DE	Red	On	The communication module is not accepted by the standard device or the standard device is not active. (See notes in the documentation for the standard device.)

11.1.2 Fieldbus status displays

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

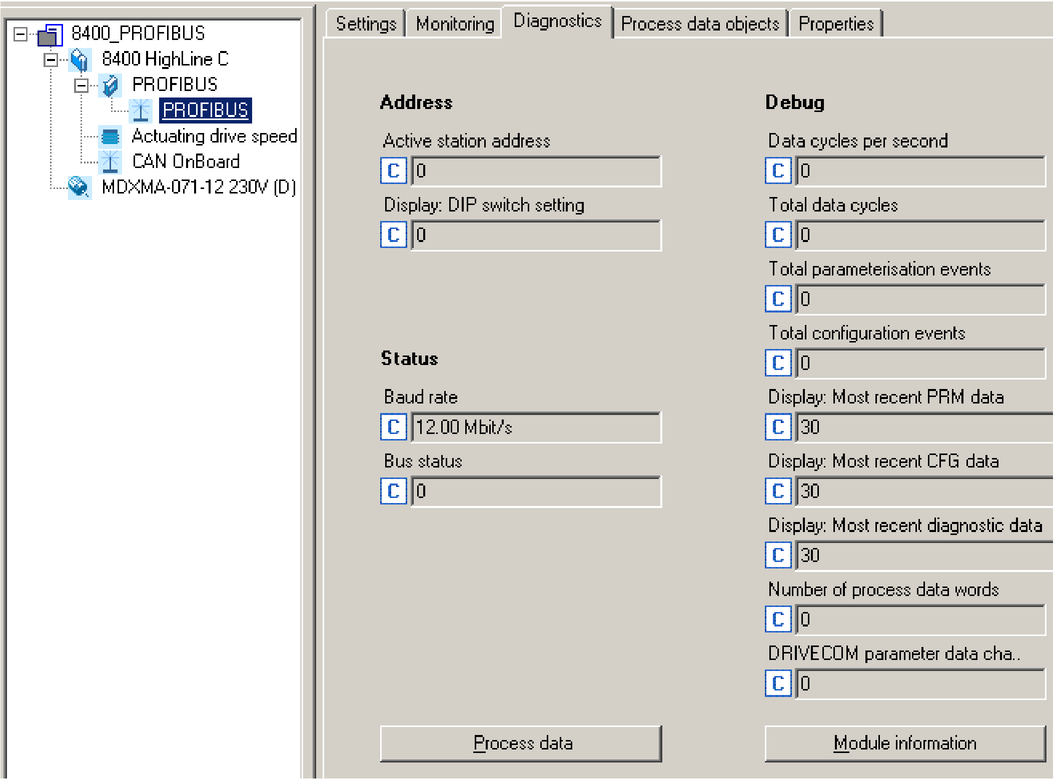


[11-2] LED status displays BS and BE

LED	Colour	State	Description
BS	Green	Off	The communication module is not active on the fieldbus or is being initialised.
		Blinking	
			The communication module is in the DATA_EXCH state ("Data_Exchange"). Data are exchanged via PROFIBUS.
BE	Red	Blinking	
			Incorrect setting for the station address. The communication module is initialised and internally operates with the respective default values.
		On	
			Bus error/fault is active (e.g. bus cable unplugged).

11.2 Diagnosing with the »Engineer«

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



Querying the current bus status

Code C13861 displays the current PROFIBUS status in a bit-coded form:

Bit assignment				Description	
Bit 3	Bit 2	Bit 1	Bit 0	Reserved	
		Bit 5	Bit 4	**Status of the DP state machine (DP-STATE)**	
		0	0	WAIT_PRM	The slave waits for a parameter data telegram after acceleration. Other types of telegrams will not be processed. Data exchange is not yet possible.
		0	1	WAIT_CFG	The slave waits for the configuration telegram that specifies the number of input and output bytes. The master informs the slave about the number of I/O bytes that will be transferred.
		1	0	DATA_EXCH	If the parameter settings as well as the configuration have been accepted by the firmware and by the application, the slave state changes to DATA_EXCH ("Data Exchange", exchange of user data with the master).
		1	1	Not possible	
		Bit 7	Bit 6	**Status of the watchdog state machine (WD-STATE)**	
		0	0	BAUD_SEARCH	The PROFIBUS slave is able to automatically detect the baud rate.
		0	1	BAUD_CONTROL	After recognising the correct baud rate, the slave status changes to BAUD_CONTROL and the baud rate is monitored.
		1	0	DP_CONTROL	The DP_CONTROL status serves for response monitoring of the master.
		1	1	Not possible	
Bit 11	Bit 10	Bit 9	Bit 8	**PROFIBUS baud rate detected**	
0	0	0	0	12 Mbps	
0	0	0	1	6 Mbps	
0	0	1	0	3 Mbps	
0	0	1	1	1.5 Mbps	
0	1	0	0	500 kbps	
0	1	0	1	187.5 kbps	
0	1	1	0	93.75 kbps	
0	1	1	1	45.45 kbps	
1	0	0	0	19.2 kbps	
1	0	0	1	9.6 kbps	
Bit 15	Bit 14	Bit 13	Bit 12	Reserved	

11.3 Advanced diagnostic message

Errors in the inverter and its plugged-in modules are transmitted to the PROFIBUS master in the form of advanced diagnostic messages.

Structure of the diagnostic message

Byte	Description
1	Bit 0: Station does not exist (set by the master). Bit 1: Slave is not ready for data exchange. Bit 2: Configuration data do not correspond. Bit 3: Slave has extended diagnostic data. Bit 4: Requested function is not supported by the slave. Bit 5: Slave response is invalid (set by the master) Bit 6: Incorrect parameter setting Bit 7: Slave has been parameterised by another master (set by the master).
2	Bit 0: Slave must be parameterised again. Bit 1: Static diagnostics Bit 2: Permanently set to "1". Bit 3: Watchdog active Bit 4: Freeze command received. Bit 5: Sync command received. Bit 6: Reserved Bit 7: Slave is deactivated (set by the master).
3	Bit 7: Diagnostics overflow - amount of diagnostic information present in the slave is too large to fit into one telegram.
4	Bits 0 ... 7: Master address after parameterisation ("0xFF" without parameterisation)
5	Bits 0 ... 7: ID number (high byte)
6	Bits 0 ... 7: ID number (low byte)
7	Header • The header contains the block length of the advanced diagnostics including the header byte. • In this case, the value of the entry is "0x0A" (bytes 7 ... 16 = 10 bytes).
8	Status_Type The value of this entry is fixed. For the following bit assignment it is "0x81": • Bit 7 = 1: "status" • Bit 0 = 1: "status message" • Value of all other bits = 0
9	Slot_Number The value of the slot number is "0x00".
10	Specifier • An indicated error is entered in the specifier with the identification "0x1" (status coming). • An eliminated error is entered in the specifier with the identification "0x02" (status going). • If no errors are indicated, the entry in the specifier has the value "0x00" (no further differentiation).
11	Reserved
12	
13 ... 16	Error code of the Inverter Drive 8400 • Code **C00165** can be used to read out the contents of the fault memory. • Detailed information regarding the error codes of the Inverter Drive 8400 can be found in the documentation of the inverter.

Example: "Short circuit (OC1)" error in the Inverter Drive 8400

Byte	Value [hex]	Description
1	x	Standard data (PRM_Fault)
...		
6		
7	0A	Block length of the advanced diagnostics = 10 bytes
8	81	Status message
9	00	Slot 0
10	01	Status coming
11	00	
12	00	
13	0B	Error message 0x11C4000B "Short circuit (OC1)" • Error type: "Warning locked" • Subject area: 0x11C4 (current) • Error ID: 0x000B The error number "0x11C4000B" indicates the following: In the "Current" subject area, an overcurrent has been detected. The error response to this is a "Warning locked", which must be unlocked separately after the error has been eliminated.
14	00	
15	C4	
16	11	

12 Error messages

This chapter supplements the error list contained in the software manual and in the »Engineer« online help for Inverter Drives 8400 by the error messages of the communication module.

Software manual/online help for Inverter Drives 8400

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

12.1 Short overview of the PROFIBUS error messages

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

Tip!

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

Error number			Error text	Error type	Adjustable in
hex	dec (subject area no.)	dec (error no.)			
0x01bc3100	444	12544	Connection to 8400 standard device lost	Error	-
0x01bc5531	444	21809	Memory: No access	Error	-
0x01bc5532	444	21810	Memory: Read error	Error	-
0x01bc5533	444	21811	Memory: Write error	Error	-
0x01bc6010	444	24592	Restart after watchdog reset	Error	-
0x01bc6011	444	24593	Internal error	Error	-
0x01bc6100	444	24832	Internal error	Error	-
0x01bc6101	444	24833	Internal error	Error	-
0x01bc6110	444	24848	Internal error	Error	-
0x01bc641f	444	25631	Invalid parameter set	Error	-
0x01bc6420	444	25632	Error: Lenze settings loaded	Error	-
0x01bc8130	444	33072	Profibus watchdog: Monitoring time elapsed	No response	C13880/1
0x01bc8131	444	33073	Profibus: Data_Exchange state exited	No response	C13880/1
0x01bc8132	444	33074	Profibus Watchdog: DP-V1 MSC2 monitoring time exceeded	No response	C13880/2

12.2 Possible causes and remedies

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

Connection to 8400 standard device lost [0x01bc3100]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
• Network cable (plug) is defective. • Network cable is not connected to the PROFIBUS terminal X201. • Voltage supply is interrupted.	Check cables and terminals. Connect network cable to the PROFIBUS terminal X201.

Memory: No access [0x01bc5531]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Access to memory was not possible.	Repeat the download of the application (including module)

Memory: Read error [0x01bc5532]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Parameter could not be read.	Repeat the download of the application (including module)

Memory: Write error [0x01bc5533]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Parameter could not be written.	Repeat the download of the application (including module)

Restart after watchdog reset [0x01bc6010]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Module defective.	If this occurs repeatedly, contact the Lenze service.

Internal error [0x01bc6011]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Module defective.	If this occurs repeatedly, contact the Lenze service.

Internal error [0x01bc6100]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Internal error.	If this occurs repeatedly, contact the Lenze service.

Internal error [0x01bc6101]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
The communication module carries out an automatic software reset and reinitialises itself.	If this occurs repeatedly, contact the Lenze service.

Internal error [0x01bc6110]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Module defective.	If this occurs repeatedly, contact the Lenze service.

Invalid parameter set [0x01bc641f]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
No active parameter set could be loaded.	Repeat the download of the application (including module)

Error: Lenze settings loaded [0x01bc6420]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Access to parameter set was not successful.	Repeat the download of the application (including module).

Profibus watchdog: Monitoring time elapsed [0x01bc8130]

Response (Lenze setting printed in bold)	**Setting:** C13880/1 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
Permanent interruption of communication to the PROFIBUS master. Also see the chapter "Permanent interruption of PROFIBUS communication" (📖 81).	Check cables and terminals.

Profibus: Data_Exchange state exited [0x01bc8131]

Response (Lenze setting printed in bold)	**Setting:** C13880/1 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
Data exchange via PROFIBUS has been stopped. Also see the chapter "Permanent interruption of PROFIBUS communication" (📖 81).	Check cables and terminals. The slave must receive new parameterisation and configuration files from the master in order to be able to exchange data again.

Profibus watchdog: DP-V1 MSC2 monitoring time exceeded [0x01bc8132]

Response (Lenze setting printed in bold)	**Setting:** C13880/2 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
Permanent interruption of communication to C2-PROFIBUS master. Also see the chapter "Permanent interruption of PROFIBUS communication" (📖 81).	Check cables and terminals.

13 Parameter reference

This chapter supplements the parameter list and the table of attributes contained in the software manual and in the »Engineer« online help for Inverter Drives 8400 by the parameters of the E84AYCPM communication module (PROFIBUS).

Software manual/»Engineer« online help for Inverter Drives 8400
Here you can find general information on parameters.

13.1 Parameters of the communication module

This chapter lists the parameters of the E84AYCPM communication module (PROFIBUS) in numerically ascending order.

C13850

Parameter | Name: **C13850 | All words to master**			Data type: UNSIGNED_16 Index: 10725_d = $29E5_h$
Display of the process data words transferred from the communication module to the PROFIBUS master. In subcodes 1 ... 16, all process data words to the master are displayed. Only the process data words configured are valid.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13850/1			
...			
C13850/16			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13851

Parameter | Name: **C13851 | All words from master**			Data type: UNSIGNED_16 Index: 10724_d = $29E4_h$
Display of the process data words transferred from the PROFIBUS master to the communication module. In subcodes 1 ... 16, all process data words to the master are displayed. Only the process data words configured are valid.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13851/1			
...			
C13851/16			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13852

Parameter | Name: **C13852 | All words to standard device**			Data type: UNSIGNED_16 Index: 10723_d = $29E3_h$
Display of process data words 1 ... 16 which are transferred from the communication module to the standard device. In subcodes 1 ... 16, all process data words from the communication module are displayed.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13852/1			
...			
C13852/16			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13853

Parameter | Name: **C13853 | All words from standard device**			Data type: UNSIGNED_16 Index: 10722_d = $29E2_h$
Display of process data words 1 ... 16 which are transferred from the standard device to the communication module. In subcodes 1 ... 16, all process data words from the standard device are displayed.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13853/1			
...			
C13853/16			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13860

Parameter | Name: **C13860 | Settings**			Data type: UNSIGNED_8 Index: 10715_d = $29DB_h$
Display of the current configuration data.			
Display area (min. value | unit | max. value)			
0		255	
Subcodes			**Info**
C13860/1			Reserved
C13860/2			Number of process data words (1 ... 16 words)
C13860/3			DRIVECOM parameter data channel • 0: Not active • 1: Active
C13860/4			Reserved
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13861

Parameter | Name: **C13861 | Bus status**			Data type: UNSIGNED_16 Index: 10714_d = $29DA_h$
Bit-coded display of the current bus state. ▸ Querying the current bus status (📖 88)			
Display area (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13862

Parameter | Name: **C13862 | Bus counter**			Data type: UNSIGNED_16 Index: 10713_d = $29D9_h$
When the maximum count value of 65535 is reached, the counter starts again with 0.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13862/1			Data cycles per second
C13862/2			Total data cycles
C13862/3			Total parameterisation events
C13862/4			Total configuration events
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13863

Parameter | Name: **C13863 | Baud rate**		Data type: UNSIGNED_8 Index: 10712_d = $29D8_h$
Display of the baud rate		
Selection list (read only)		
0	12.00 Mbps	
1	6.00 Mbps	
2	3.00 Mbps	
3	1.50 Mbps	
4	500.00 kbps	
5	187.50 kbps	
6	93.75 kbps	
7	45.45 kbps	
8	19.20 kbps	
9	9.60 kbps	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13864

Parameter | Name: **C13864 | Active station address**			Data type: UNSIGNED_8 Index: 10711_d = $29D7_h$
Display of the active station address If all DIP switches 1 ... 64 are in the "OFF" position (Lenze setting), the station address set in C13899 becomes active and is displayed here after switching on. ▸ Setting the station address (📖 32)			
Display area (min. value | unit | max. value)			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13865

Parameter | Name: **C13865 | Display: Most recent PRM data**	Data type: OCTET_STRING Index: 10710_d = $29D6_h$
Display of the last parameter data sent by the PROFIBUS master with the "Set-Prm" telegram (ASCII string with 24 characters)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13866

Parameter | Name: **C13866 | Display: Most recent CFG data**	Data type: OCTET_STRING Index: 10709_d = $29D5_h$
Display of the last configuration data sent by the PROFIBUS master with the "Chk-Cfg" telegram (ASCII string with 22 characters)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13867

Parameter | Name: **C13867 | Display: Most recent diagnostic data**	Data type: OCTET_STRING Index: 10708_d = $29D4_h$
Display of the last diagnostic data sent to the PROFIBUS master (ASCII string with 16 characters) ▸ Advanced diagnostic message (📖 89)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13880

Parameter | Name: **C13880 | Reaction on communication fault**		Data type: UNSIGNED_8 Index: 10695_d = $29C7_h$
Monitoring response to a communication fault on the PROFIBUS A change in the monitoring response is effective immediately. ▸ Permanent interruption of PROFIBUS communication (📖 81)		
Selection list		
0	No response	
1	Error	
3	Quick stop by trouble	
4	Warning locked	
6	Information	
Subcodes	**Lenze setting**	**Info**
C13880/1	0: No response	The response set here for the "Watchdog/Data Exchange" monitoring function is executed if the bus station ... • does not receive a message from the master within the watchdog monitoring time (displayed in C13882/1) if there is an active connection. • detects that it is no longer in the "Data_Exchange" status. Please see also the notes given under C13881.
C13880/2	0: No response	The response set here for the "DPV1 MSAC2" monitoring is executed if the bus station does not receive any "DPV1 MSAC2" message from the master within the monitoring time (displayed in C13882/2 if there is an active connection) and the MSAC2 connection is stopped by the slave. **Note:** We recommend only setting "information" as response so that no drive-relevant response is executed.
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13881

Parameter | Name: **C13881 | Monitoring time: Data exchange**			Data type: UNSIGNED_16 Index: 10694_d = $29C6_h$
If the "Data Exchange" state is exited, the response parameterised under C13880/1 is carried out when the monitoring time for data exchange set here has expired. • A value of "65535" in this code deactivates the monitoring function. • A change in monitoring is effective immediately. • The value set here for the monitoring time must be smaller than the watchdog monitoring time (C13882/1). ▸ Permanent interruption of PROFIBUS communication (📖 81)			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	65535	**65535 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13882

Parameter | Name: **C13882 | Monitoring time: Watchdog**			Data type: UNSIGNED_32 Index: 10693_d = $29C5_h$
Display of the watchdog monitoring time determined by the PROFIBUS master • A change in the watchdog monitoring time is immediately effective. • Monitoring starts with the receipt of the first telegram. • When a value of "0" is displayed, the monitoring function is deactivated. ▸ Permanent interruption of PROFIBUS communication (📖 81)			
Display area (min. value | unit | max. value)			
0	ms	4294967295	
Subcodes			**Info**
C13882/1			Watchdog monitoring time
C13882/2			DP-V1 MSC2
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13885

Parameter | Name: **C13885 | Clear process data**		Data type: UNSIGNED_8 Index: 10690_d = $29C2_h$
Selection of the process data which the inverter will process in the event of a PROFIBUS failure in order to maintain internal communication.		
Selection list (Lenze setting printed in bold)		
0	Use of most recent master PDOs	
1	**PDOs are set to the value '0'**	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13886

Parameter | Name: **C13886 | Set ext. diagnostic bit by**		Data type: BITFIELD_8 Index: 10689_d = $29C1_h$
Bit-coded selection of the error responses in the standard device causing the external diagnostic bit ("diag bit") to be set (see PROFIBUS specification; bit 3 of byte 1 of the DP diagnostic messages). • The diagnostic bit is sent to the PROFIBUS master where it is evaluated separately. • The diagnostic bit is always set when a system error occurs. • The Lenze setting "0" means that the diagnostic bit is not set for the following error responses. • An advanced diagnostic message is always sent.		
Value is bit-coded:		
Bit 0	Error	
Bit 1	Trouble	
Bit 2	Quick stop by trouble	
Bit 3	Warning locked	
Bit 4	Warning	
Bit 5	Reserved	
Bit 6	Reserved	
Bit 7	Reserved	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13887

Parameter | Name: **C13887 | Suppress signalling diag. mess. upon**	Data type: BITFIELD_8 Index: 10688_d = $29C0_h$
Selection of the error responses not causing a diagnostic request to the PROFIBUS master. The Lenze setting "0" means that for each of the following error responses a diagnostic request is signalled.	

Value is bit-coded:	
Bit 0	Error
Bit 1	Trouble
Bit 2	Quick stop by trouble
Bit 3	Warning locked
Bit 4	Warning
Bit 5	Reserved
Bit 6	Reserved
Bit 7	Reserved

☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT

C13899

Parameter | Name: **C13899 | Station address**	Data type: UNSIGNED_8 Index: 10676_d = $29B4_h$

Optional setting of the station address (instead of setting via DIP switches 1 ... 64)
- The station address set here only becomes effective if the DIP switches 1 ... 64 have been set to OFF prior to mains switching.
- The active station address is displayed under C13864.

Note: A change of the station address will not be effective until the " Save parameter set" device command has been executed and another mains switching for the communication module/inverter has been performed.

▸ Setting the station address (🕮 32)

Setting range (min. value | unit | max. value)			**Lenze setting**
3		126	3

☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☑ COM ☐ MOT

C13900

Parameter | Name: **C13900 | Firmware product type**	Data type: VISIBLE_STRING Index: 10675_d = $29B3_h$
Display of the product type (string with a length of 8 bytes) The following identification code is displayed: "E84AFYPM".	

☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT

C13901

Parameter | Name: **C13901 | Firmware compilation date**	Data type: VISIBLE_STRING Index: 10674_d = $29B2_h$
Display of the compilation date of the firmware (string with a length of 20 bytes) The date ("MMM DD YYYY") and time ("hh:mm:ss") are displayed, e.g. "Mar 21 2005 12:31:21".	

☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT

C13902

Parameter | Name: **C13902 | Firmware version**	Data type: VISIBLE_STRING Index: 10673_d = $29B1_h$
Display of the firmware version (string with a length of 5 bytes) Example: "01.00"	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13920

Parameter | Name: **C13920 | Display: DIP switch setting**			Data type: UNSIGNED_8 Index: 10655_d = $299F_h$
Display of the current DIP switch setting • The displayed value corresponds to the sum of the individual DIP switch values 1 ... 64. • The active station address is displayed under C13864. ▶ Setting the station address (📖 32)			
Display area **(min. value | unit | max. value)**			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

13.2 Table of attributes

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

How to read the table of attributes:

<table>
<tr><th colspan="2">Column</th><th>Meaning</th><th colspan="2">Entry</th></tr>
<tr><td colspan="2">Code</td><td>Parameter designation</td><td colspan="2">Cxxxxx</td></tr>
<tr><td colspan="2">Name</td><td>Parameter short text (display text)</td><td colspan="2">Text</td></tr>
<tr><td rowspan="2">Index</td><td>dec</td><td rowspan="2">Index by which the parameter is addressed.
The subindex for array variables corresponds to the Lenze subcode number.</td><td>24575 - Lenze code number</td><td rowspan="2">Is only required for access via a bus system.</td></tr>
<tr><td>hex</td><td>$5FFF_h$ - Lenze code number</td></tr>
<tr><td rowspan="16">Data</td><td rowspan="2">DS</td><td rowspan="2">Data structure</td><td>E</td><td>Single variable
#(only one parameter element)</td></tr>
<tr><td>A</td><td>Array variable
(several parameter elements)</td></tr>
<tr><td>DA</td><td>Number of array elements (subcodes)</td><td colspan="2">Number</td></tr>
<tr><td rowspan="11">DT</td><td rowspan="11">Data type</td><td>BITFIELD_8</td><td>1 byte, bit-coded</td></tr>
<tr><td>BITFIELD_16</td><td>2 bytes, bit-coded</td></tr>
<tr><td>BITFIELD_32</td><td>4 bytes, bit-coded</td></tr>
<tr><td>INTEGER_8</td><td>1 byte with sign</td></tr>
<tr><td>INTEGER_16</td><td>2 bytes with sign</td></tr>
<tr><td>INTEGER_32</td><td>4 bytes with sign</td></tr>
<tr><td>UNSIGNED_8</td><td>1 byte without sign</td></tr>
<tr><td>UNSIGNED_16</td><td>2 bytes without sign</td></tr>
<tr><td>UNSIGNED_32</td><td>4 bytes, without sign</td></tr>
<tr><td>VISIBLE_STRING</td><td rowspan="2">ASCII string</td></tr>
<tr><td>OCTET_STRING</td></tr>
<tr><td>Factor</td><td>Factor for data transmission via a bus system, depending on the number of decimal positions</td><td>Factor</td><td>1 ≡ no decimal positions
10 ≡ 1 decimal position
100 ≡ 2 decimal positions
1000 ≡ 3 decimal positions</td></tr>
<tr><td rowspan="3">Access</td><td>R</td><td>Read access</td><td colspan="2">☑ Reading permitted</td></tr>
<tr><td>W</td><td>Write access</td><td colspan="2">☑ Writing permitted</td></tr>
<tr><td>CINH</td><td>Controller inhibit required</td><td colspan="2">☑ Writing is only possible if controller inhibit is set</td></tr>
</table>

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C13850	All words to master	10725	29E5	A	16	UNSIGNED_16	1	☑		
C13851	All words from master	10724	29E4	A	16	UNSIGNED_16	1	☑		
C13852	All words to standard device	10723	29E3	A	16	UNSIGNED_16	1	☑		
C13853	All words from standard device	10722	29E2	A	16	UNSIGNED_16	1	☑		
C13860	Settings	10715	29DB	A	4	UNSIGNED_8	1	☑		
C13861	Bus status	10714	29DA	E	1	UNSIGNED_16	1	☑		
C13862	Bus counter	10713	29D9	A	4	UNSIGNED_16	1	☑		
C13863	Baud rate	10712	29D8	E	1	UNSIGNED_8	1	☑		
C13864	Active station address	10711	29D7	E	1	UNSIGNED_8	1	☑		
C13865	Display: Most recent PRM data	10710	29D6	E	1	OCTET_STRING		☑		
C13866	Display: Most recent CFG data	10709	29D5	E	1	OCTET_STRING		☑		
C13867	Display: Most recent diagnostic data	10708	29D4	E	1	OCTET_STRING		☑		
C13880	Reaction on communication fault	10695	29C7	A	2	UNSIGNED_8	1	☑	☑	
C13881	Monitoring time: Data exchange	10694	29C6	E	1	UNSIGNED_16	1	☑	☑	
C13882	Monitoring time: Watchdog	10693	29C5	A	2	UNSIGNED_32	1	☑		
C13885	Clear process data	10690	29C2	E	1	UNSIGNED_8	1	☑	☑	
C13886	Set ext. diagnostic bit upon	10689	29C1	E	1	BITFIELD_8		☑	☑	
C13887	Suppress signalling diag. mess. upon	10688	29C0	E	1	BITFIELD_8		☑	☑	
C13899	Station address	10676	29B4	E	1	UNSIGNED_8	1	☑	☑	
C13900	Firmware product type	10675	29B3	E	1	VISIBLE_STRING		☑		
C13901	Firmware compilation date	10674	29B2	E	1	VISIBLE_STRING		☑		
C13902	Firmware version	10673	29B1	E	1	VISIBLE_STRING		☑		
C13920	Display: DIP switch setting	10655	299F	E	1	UNSIGNED_8	1	☑		

13.3 Implemented PROFIdrive objects (DP-V1)

I-918

Index | Name: **0x918 | Display of station address**			Data type: U16
Display of the station address set			
Display area (min. value | unit | max. value)			
1		126	
☑ Read access ☐ Write access			

I-963

Index | Name: **0x963 | Baud rate**		Data type: U16
Display of the PROFIBUS baud rate		
Selection list (read only)		
0	9.6 kbps	
1	19.2 kbps	
2	93.75 kbps	
3	187.5 kbps	
4	500 kbps	
6	1.5 Mbps	
7	3 Mbps	
8	6 Mbps	
9	12 Mbps	
10	31.25 kbps	
11	45.45 kbps	
☑ Read access ☐ Write access		

I-964

Index | Name: **0x964 | Device identification**		Data type: U16
Display of identification data		
Subindex	Display	**Info**
0x964/0	262	Manufacturer: Lenze
0x964/1	8400	Device type
0x964/2	xxyy	Software version, e.g. 0100 (V 01.00)
0x964/3	yyyy	Firmware date (year), e.g. 2007
0x964/4	ddmm	Firmware date (day/month), e.g. 0506 (5th June)
☑ Read access ☐ Write access		

I-974

Index | Name: **0x974 | Maximum time per DPV1 parameter access**		Data type: U16
Display of access statistics		
Subindex	Display	**Info**
0x974/0	240 bytes	Maximum block length
0x974/1	40	Maximum number of parameter accesses
0x974/2	0	Maximum time per access
☑ Read access ☐ Write access		

14 DIP switch positions for setting the station address

The station address results from the sum of the binary valencies of switches 1 ... 64.

The following table shows the switch positions for the valid address range 1 ... 126.

▶ Setting the station address (📖 32)

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**

Station address	DIP switch						
	64	32	16	8	4	2	1
38	OFF	**ON**	OFF	OFF	**ON**	**ON**	OFF
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

Station address	DIP switch						
	64	**32**	**16**	**8**	**4**	**2**	**1**
81	**ON**	OFF	**ON**	OFF	OFF	OFF	**ON**
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**

Station address	DIP switch						
	64	**32**	**16**	**8**	**4**	**2**	**1**
124	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	OFF
125	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	**ON**
126	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	OFF

FEEDBACK

Your opinion is important to us

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

If you have suggestions for improvement, please e-mail us to:

feedback-docu@Lenze.de

Thank you for your support.

Your Lenze documentation team

E84AYCPM communication module (PROFIBUS®) · Communication Manual · EDS84AYCPM · 13422193 · DMS 5.0 EN · 11/2012 · TD17

Lenze Drives GmbH
Postfach 10 13 52
D-31763 Hameln
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

Accessories

PROFIBUS®

E84DGFCpxxx

Inverter Drives 8400 motec

Communication Manual EN

13564909

Lenze

Contents

1 About this documentation

Contents

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

Note!

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

The hardware manual contains safety instructions which must be observed!

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

Examples illustrate typical applications.

This documentation also contains the following:

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

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

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

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

Tip!

Detailed information about PROFIBUS can be found on the website of the PROFIBUS user organisation:

www.profibus.com

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 (»Engineer«), screenshots in this documentation may differ from the representation on the screen.

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 **PROFIBUS Communication Unit**	E84DGFCPxNx	PROFIBUS
	E84DGFCPxJx	PROFIBUS + Safety

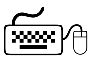
▶ Features and variants (📖 14)

1.1 Document history

Version			Description
4.0	02/2019	TD23	• General revision
3.0	11/2011	TD17	• General revision • Digital and analog input information (📖 38) supplemented. • Description of code C13887 (from version 02.00) supplemented.
2.0	01/2011	TD17	• DIP switch settings (📖 30) corrected. • »Engineer« screenshots updated.
1.0	09/2010	TD17	First edition

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	underlined	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.

1.3 Terminology used

Term	Meaning
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

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

Application notes

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

2 Safety instructions

Note!

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

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

2.1 General safety and application notes

Danger!

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

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

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.
  ▸ Bus cable specification (📖 24)

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

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

2.3 Residual hazards

Device protection

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

3 Product description

3.1 Application as directed

The communication unit PROFIBUS ...

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

Product series	Type designation
Inverter Drives 8400 motec **Drive Unit**	E84DGDVxxxxxxxx
Inverter Drives 8400 motec **Wiring Unit**	E84DGVNxx

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

Any other use shall be deemed inappropriate!

3.2 Features and variants

The communication unit PROFIBUS is available in the following versions:

Product series	Type designation	Product features				
		Enclosure	Connection PROFIBUS	I/O: Connection via terminal	I/O: Connection via M12	Safety
Inverter Drives 8400 motec **Communication unit PROFIBUS**	E84DGFCPFNP	IP 65	M12	3× DI 1× DO	2× DI	
	E84DGFCPENP	IP 65	M12	2× DI	3× DI 1× DO	
	E84DGFCPFJP	IP 65	M12	3× DI 1× DO 1× AI	2× DI	●
	E84DGFCPEJP	IP 65	M12	3× DI	2× DI 1× DO 1× AI	●

- The PROFIBUS communication unit is ...
  - mounted on top of the Wiring Unit (E84DGVNxx);
  - supplied internally via the Drive Unit (E84DGDVxxxxxxxx) or externally via a separate voltage source.
- The I/O connections can be brought into the device via M12 connectors or cable glands.
- Devices without an integrated safety system (safety option) have no analog input and no relay output.
- The integrated safety system can be used on machines for the protection of persons.
- Support of the parameter data channel DRIVECOM (DP-V0), PROFIDrive (DP-V1) in preparation
- Exchange of up to 8 process data words per direction
- Bus coupling via remote bus according to the RS485 standard
- Automatic detection of the baud rate (9.6 kbps to 12 Mbps)
- Setting of the station address is possible via DIP switch or code.
- Communication with the Lenze »Engineer« (access to all Lenze parameters) is executed via the diagnostic interface of the Drive Unit.

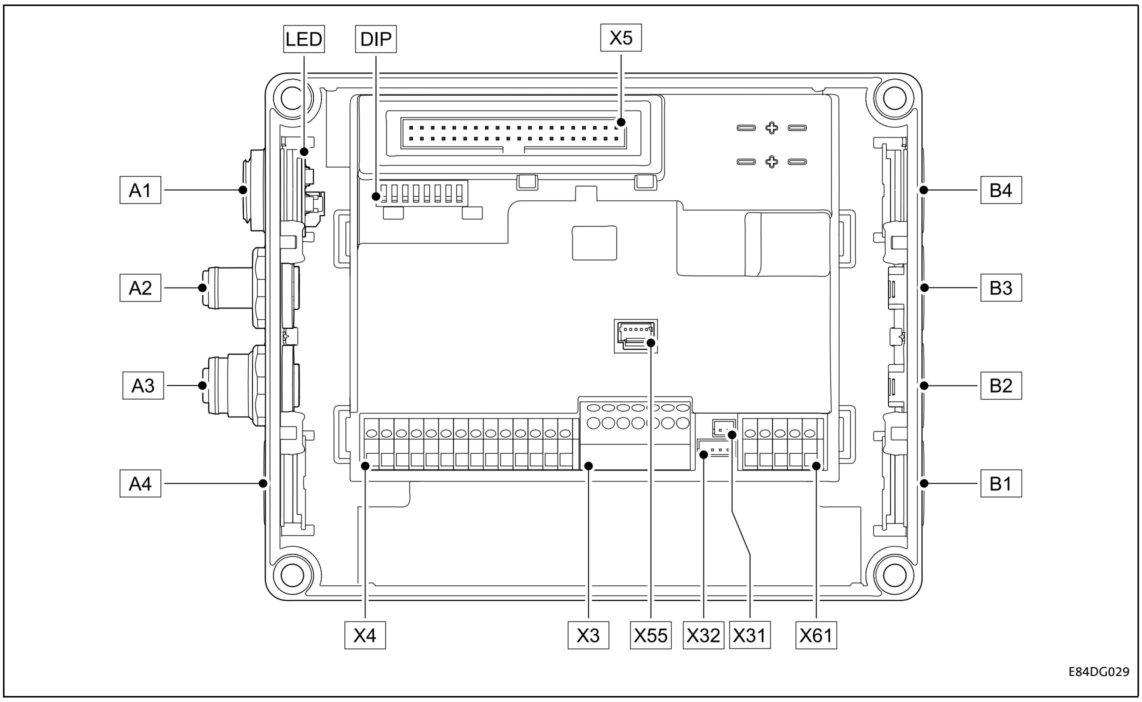
"Inverter Drives 8400 motec" hardware manual

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).

3.3 Connections and interfaces



[3-1] PROFIBUS Communication Unit

Pos.	Description
DIP	DIP switch ▸ Possible settings via DIP switch (📖 29)
A1 / LED	Position of LEDs for PROFIBUS status display ▸ LED status displays (📖 70)
A2	PROFIBUS input (M12 male, 5-pin) ▸ PROFIBUS connection (📖 25)
A3	PROFIBUS output (M12 female, 5-pin) ▸ PROFIBUS connection (📖 25)
A4 **B1 ... B4**	Positions for further freely designable inputs and outputs: • Digital inputs • Digital output • Analog input (only for E84DGFCPxJx) • Relay output (only for E84DGFCPxJx) • Connection of "Safety Option" safety system (only for E84DGFCPxJx)
X3 / X4 / X61	Terminal strips for wiring the connections at A4 and B1 ... B4
X5	Plug connector for connection to the Drive Unit
X31	Plug connector for wiring the PROFIBUS input at A2
X32	Plug connector for wiring the PROFIBUS output at A3
X55	Plug connector for the wiring of the LEDs to A1

- By default, the PROFIBUS connections and the LEDs for the PROFIBUS status displays are already mounted and wired:
  - PROFIBUS input to plug connector X31
  - PROFIBUS output to plug connector X32
  - LEDs on plug connector X55
- At positions A1 ... A4 and B1 ... B4, it is also possible to design the PROFIBUS connections and other connections (e.g. digital inputs) freely.
- For the connections, 5-pin M12 connectors or - alternatively - cable glands (cable cross-section max. 1.0 mm^2, AWG 18) can be used.
- The M12 connectors, cable glands and prefabricated system cables can be obtained from various manufacturers.
- Wire the M12 connectors or cable glands used to the corresponding contacts of the terminal strips/plug connectors X3, X4 and X61.

"Inverter Drives 8400 motec" hardware manual

Observe the notes and wiring instructions contained in this documentation.

4 Technical data

"Inverter Drives 8400 motec" hardware manual

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

4.1 General data and operating conditions

Range	Values
Order designation	• E84DGFCPxNx (PROFIBUS) • E84DGFCPxJx (PROFIBUS + Safety)
Communication profile	• PROFIBUS DP-V0 (DRIVECOM) • PROFIBUS DP-V1 (PROFIdrive), from SW version 2.0
Standards / specifications	• IEC 61158 / EN 50170 • IEC 61784
Communication medium	RS485
Interface for communication	• PROFIBUS input: M12 pins, 5-pole, B-coded • PROFIBUS output: M12 socket, 5-pole, B-coded
Max. cable length	1200 m (depending on the selected baud rate, the used cable type and hardware (repeaters))
Bus termination	Bus terminating resistors are required at the first and last PROFIBUS node (implemented in the connector of the bus cable)
Network topology	• Line (without repeater) • Tree/line (with repeater)
Type of node	PROFIBUS slave
Slave node number	• Max. 31 (without repeater) • Max. 125 (with repeater)
PNO identification number	0x0A89
Baud rate for cable type A (EN 50170)	9.6 kbps ... 12 Mbps (automatic detection)
External voltage supply	• U = 24 V DC (20 V - 0 % ... 29 V + 0 %) • I_{max} = 120 mA
Conformities, approvals	• CE • UR / cUR

4.2 Protocol data

Range	Values
Process data words (PCD)	1 ... 8 words (16 bits/word)
Cyclic parameter data channel (DP-V0)	4 words
Acyclic parameter data channel (DP-V1)	Max. 240 bytes
PROFIBUS user data length	1 ... 8 words process data channel + 4 words parameter data channel

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 PROFIBUS 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.

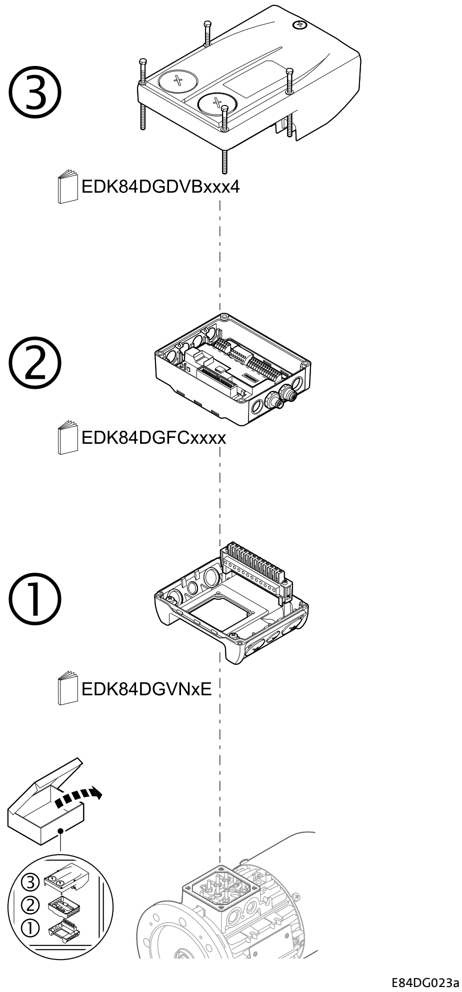
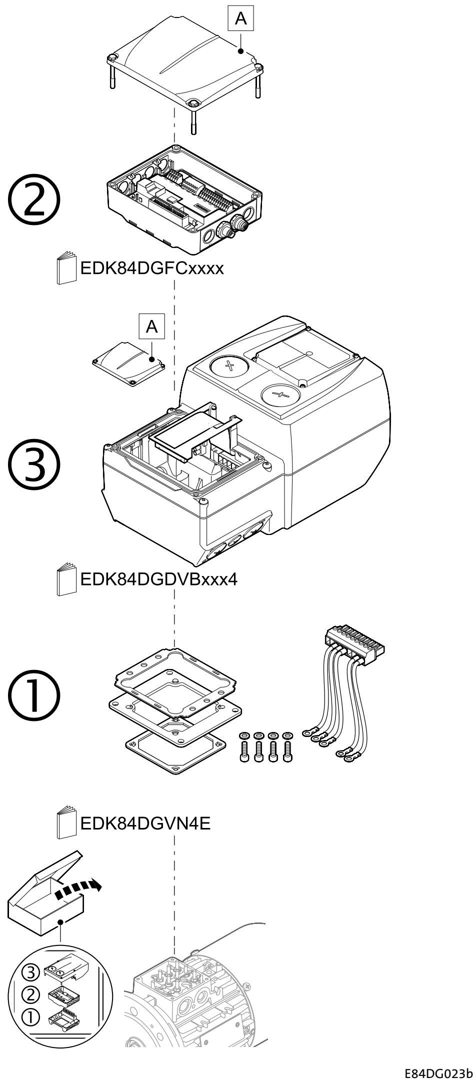
Protective measures

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.

0.37 ... 3.0 kW	4.0 ... 7.5 kW
	

[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.

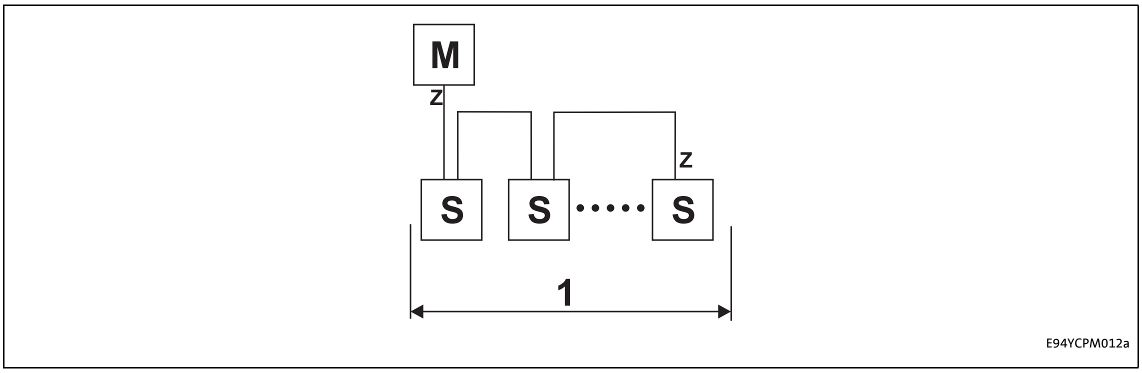
Observe the notes and wiring instructions contained in this documentation.

5.2.1 Network topology

Two simple RS485 networks are described in the following examples.

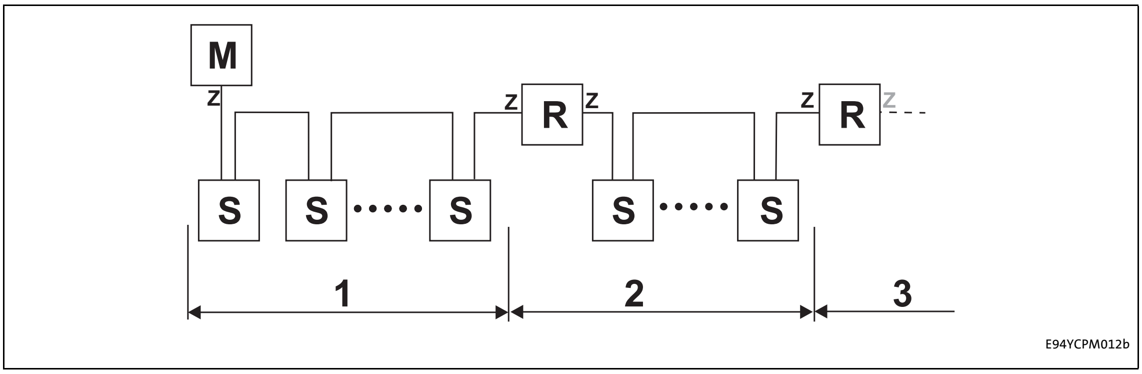
Every segment of the network must be terminated at both ends. The bus terminators of the PROFIBUS are marked with a "Z" in the below examples.

With an RS485 network of only one segment, the PROFIBUS master (M) with the integrated bus terminating resistor starts the segment and the bus of the last PROFIBUS node (S) must be terminated with a bus terminating resistor.



[5-2] RS485 network with one segment

An RS485 network consisting of several segments contains repeaters (R) for connecting the segments. The repeaters are provided with integrated bus terminating resistors.



[5-3] RS485 network with a repeater

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.

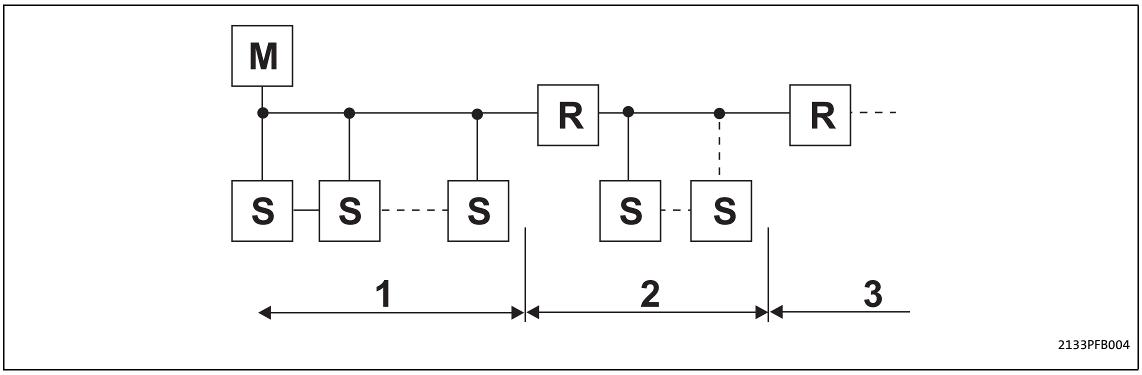
External supply of the communication unit allows for the separation of the bus termination supply from the inverter supply.

Note!

The bus terminator must always be supplied. Otherwise, the bus can get unstable.

▶ Bus termination (📖 23)
▶ External voltage supply (📖 26)

Number of nodes



[5-4] Number of nodes

Segment	Master (M)	Slave (S)	Repeater (R)
1	1	31	-
	2	30	-
2	-	30	1
3	-	30	1

Tip!

Repeaters do not have a station address. When calculating the maximum number of stations, they reduce the number of stations by 1 on each side of the segment.

By means of repeaters, you can establish line or tree topologies. The maximum total dimension of the bus system depends on ...

- the used baud rate;
- the number of repeaters.

5.2.2 Bus termination

The PROFIBUS must be terminated by means of a bus terminating resistor at the first and last physical station.

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 PROFIBUS connections (input and output) must be installed in an enclosed manner. Please use either a connection cable, an enclosed bus terminator connector (M12 male, 4-pin, B-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.
- Please observe that the bus termination is not active any longer if ...
  - the bus terminator connector has been disconnected;
  - the mains supply of the drive unit and the external 24V supply of the communication unit have been switched off at the same time.

5.2.3 Bus cable specification

Note!

Only use cables that correspond to the given specifications of the PROFIBUS user organisation.

Range	Values
Cable resistance	135 ... 165 Ω/km, (f = 3 ... 20 MHz)
Capacitance per unit length	≤ 30 nF/km
Loop resistance	< 110 Ω/km
Core diameter	> 0.64 mm
Core cross-section	> 0.34 mm^2
Cores	Twisted in pairs, insulated and shielded

Bus cable length

The length of the bus cable depends on the baud rate and cable type used. The data in the following table applies to PROFIBUS cables of "FC-Standard Cable" cable type .

Baud rate	Length
9.6 ... 93.75 kbps	1200 m
187.5 kbps	1000 m
500 kbps	400 m
1500 kbps	200 m
3000 ... 12000 kbps	100 m

Note!

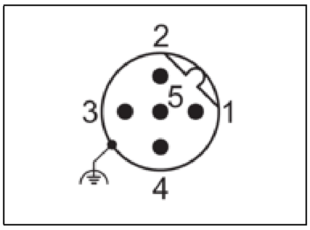
The baud rate depending on the data volume, cycle time and number of stations should only be selected as high as required for the application.

Tip!

We recommend taking the use of optical fibres into consideration for high baud rates.

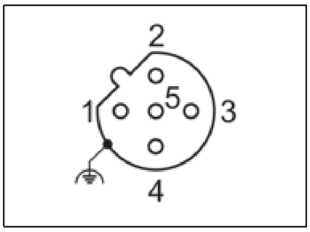
Advantages of optical fibres:
- External electromagnetic interferences have no effect on the transmission path.
- Bus lengths of several kilometres are also possible with higher baud rates.
- The bus length is ...
  - independent of the baud rate;
  - dependent on the optical fibre used.

5.2.4 PROFIBUS connection



- PROFIBUS input: M12 pins, 5-pole, B-coded
- Wiring at terminal strip X31
- Shield connection via housing

PROFIBUS input		
Pin	**Signal**	**Description**
1	-	Not assigned
2	RxD/TxD-N (A)	Data line A (received/transmitted data, minus)
3	-	Not assigned
4	RxD/TxD-P (B)	Data line B (received/transmitted data, plus)
5	-	Not assigned



- PROFIBUS output: M12 socket, 5-pole, B-coded
- Wiring at terminal strip X32
- Shield connection via housing

PROFIBUS output		
Pin	**Signal**	**Description**
1	P5V2	5 V DC / 30 mA (bus termination)
2	RxD/TxD-N (A)	Data line A (received/transmitted data, minus)
3	M5V2	Data ground (ground to 5 V)
4	RxD/TxD-P (B)	Data line B (received/transmitted data, plus)
5	-	Not assigned

5.2.5 External voltage supply

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

"Inverter Drives 8400 motec" hardware manual
Here you can find detailed information on how to wire the Communication Unit.

6 Commissioning

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

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

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

Parameter reference (79)

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 requires configuration of the host (master) first.

Configuration for the host (master) and the DP-V0 parameter data channel

For the configuration of the PROFIBUS, the PROFIBUS device description file of the Inverter Drive 8400 motec must be read into the master.

The device description file is available on Lenze's website in the "Services & Downloads" area at:

www.Lenze.com

The following language variants of the device description file can be used:

- LENZE84D.GSD (source file, English)
- LENZE84D.GSG (German)
- LENZE84D.GSE (English)

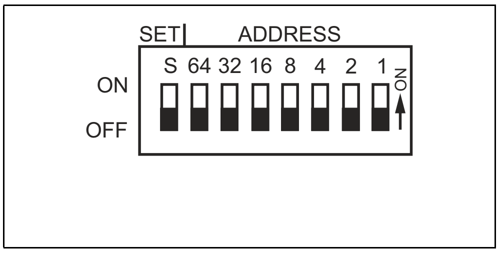
Defining the user data length

The user data length is defined during the initialisation phase of the master.

The communication unit PROFIBUS supports the configuration of max. 8 process data words (max. 16 bytes).

The user data lengths for process input data and process output data are identical.

6.3 Possible settings via DIP switch



[6-1] DIP switch

The DIP switches serve to ...

- Setting the station address (📖 30) (switches: 1 ... 64)
- Receiving the station address via the master (📖 29) (switch: S)

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 Receiving the station address via the master

Set the DIP switch **S = OFF**, in order to receive the station address automatically via the master.

- The station address active at the PROFIBUS is displayed in C13864.
- The settings of the DIP switches **1 ... 64** have no effect.
- When the mains connection is established, first the station address (DIP switch or C13899) set on the device is applied. Then the station address is obtained via the master.

6.3.2 Setting the station address

The station addresses must differ from each other if several networked PROFIBUS stations are used.

The station address can be set via the DIP switches **1 ... 64** or via the »Engineer« with code C13899.

The setting with C13899 requires DIP switches **1 ... 64** to be either **OFF** or **ON**.

Note!

The valid address range is 0 ... 126 (max. 125 slaves).

DIP switch settings

DIP switch								Station address
S	**64**	**32**	**16**	**8**	**4**	**2**	**1**	
OFF	...	...	...	...	...	...	...	Autom. via master
ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	Value from C13899
ON	OFF	OFF	OFF	OFF	OFF	OFF	**ON**	1
ON	...	...	...	...	...	...	...	...
ON	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	126
ON	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	Value from C13899

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

Example :

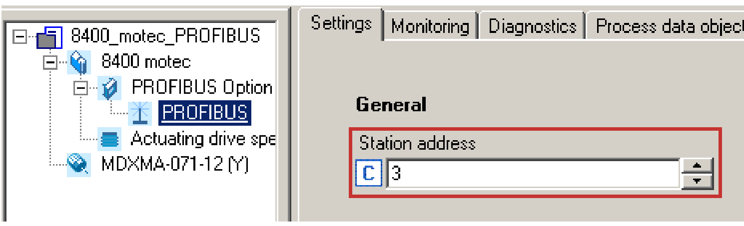
DIP switch	**64**	**32**	**16**	**8**	**4**	**2**	**1**
Switch position	OFF	OFF	**ON**	OFF	**ON**	**ON**	**ON**
Value	0	0	16	0	4	2	1
Station address	**= sum of the valencies = 16 + 4 + 2 + 1 = 23**						

- The current address set with the switches is displayed in C13920.
- The station address active at the PROFIBUS is displayed in C13864.

▶ DIP switch positions for setting the station address (🕮 91)

Setting the station address via the »Engineer«

In the »Engineer«, the station address can be set via the **Settings** tab.



- Impermissible addresses are displayed in red in the **Station address** (code C13899).
- Save changed settings with the device command **C00002/11** (save all parameter sets).

6.4 Initial switch-on

Establishing communication

- To establish communication, the inverter drive must be supplied with mains voltage.
- PROFIBUS communication requires voltage supply of the communication unit.
  If this is 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 PROFIBUS communication can be established.
- The external voltage supply serves to maintain PROFIBUS communication if the main supply fails.
  ▶ External voltage supply (📖 26)
- When the mains connection is established, all parameters (codes) and the DIP switch settings are read.
- The settings of the DIP switches determine whether the station address is selected automatically by the PROFIBUS master or via code C13899.
  ▶ Possible settings via DIP switch (📖 29)

7 Data transfer

PROFIBUS master and inverter communicate with each other by exchanging data telegrams via PROFIBUS. The user data area of the data telegram contains parameter data or process data. In the inverter, different communication channels are assigned to the parameter data and process data.

Communication channels

- The process data channel transmits process data.
  - The process data serve to control the inverter.
  - The host (master) can directly access the process data. In the PLC, for instance, the data are directly saved to the I/O area.
  - Process data are not saved in the inverter.
  - Process data are transferred cyclically between the host (master) and the inverters (slaves) (continuous exchange of current input and output data).
  - Process data are e.g. setpoints, actual values, control words, and status words.
  - The Inverter Drive 8400 motec can exchange a maximum of 8 process data words (16 bits/word) per direction.
  - In addition to the process data, digital and analog input information can also be queried. These signals are set permanently to 2 additional data words which must be parameterised correspondingly in the HW manager.

  ▶ Digital and analog input information (📖 38)

> **Note!**
>
> Please observe the direction of the flow of information!
> - Process input data (Rx data):
>   - Process data from the inverter (slave) to the master
> - Process output data (Tx data):
>   - Process data from the master to the inverter (slave)

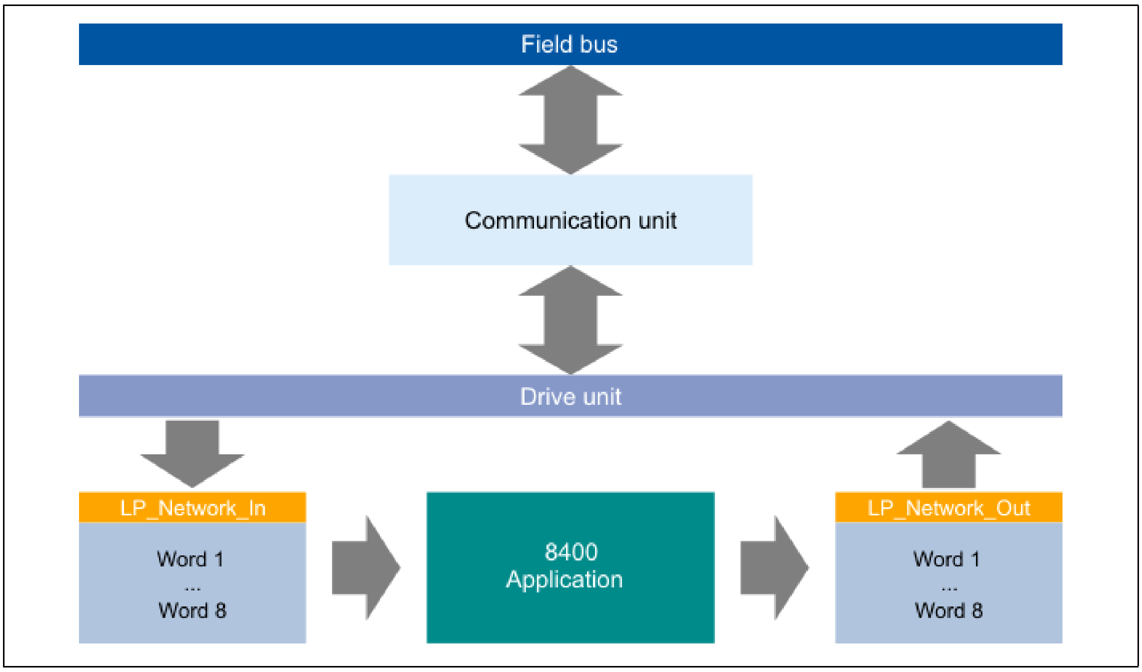
- The parameter data channel serves to transfer parameter data.
  - The parameter data channel provides access to all Lenze codes.
  - The transmission of parameter data is usually not time-critical.
  - Parameter data are, for instance, operating parameters, motor data and diagnostic information.
  - Parameter changes must be stored via code **C00002** of the Inverter Drive 8400 motec.

8 Process data transfer

8.1 Access to process data / PDO mapping

Process data are transferred via the MCI/CAN interface.

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



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

Software manual / »Engineer« online help for the Inverter Drive 8400 motec
Here you can find detailed information on port blocks and the port/function block interconnection in the »Engineer«.

8.2 Port interconnection of process data objects (PDO)

Note!

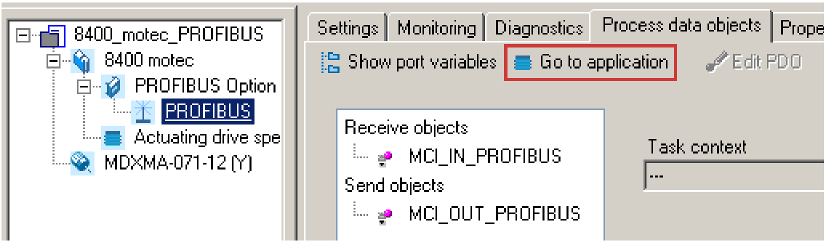
The »Engineer« screenshots shown on the following pages are only examples for the setting sequence and the resulting screens.

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.

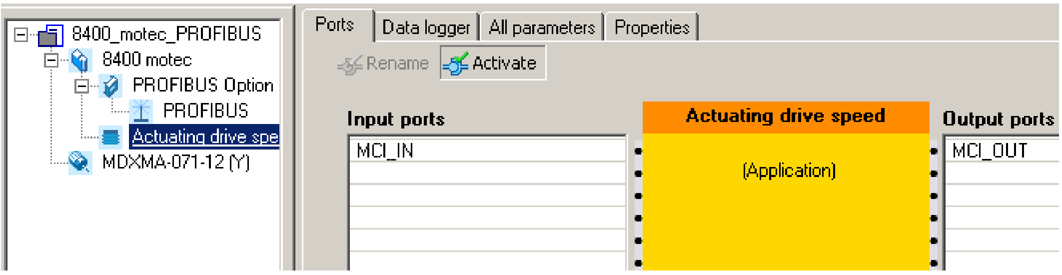
The preconfigured port interconnection of the process data objects is activated by setting code **C00007 = 40: Network (MCI/CAN)**.

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

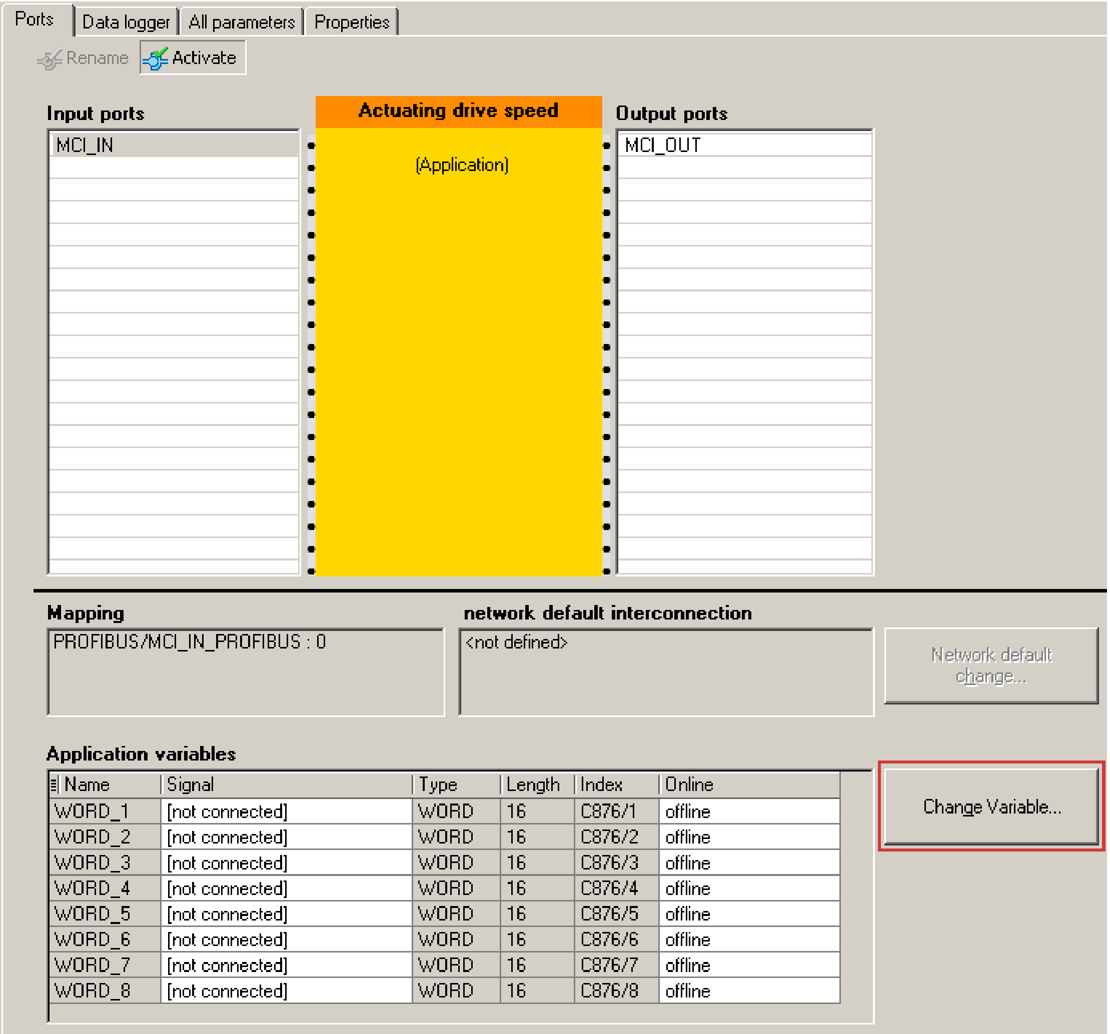
1. Go to the **Process data objects** tab and click **Go to application**.



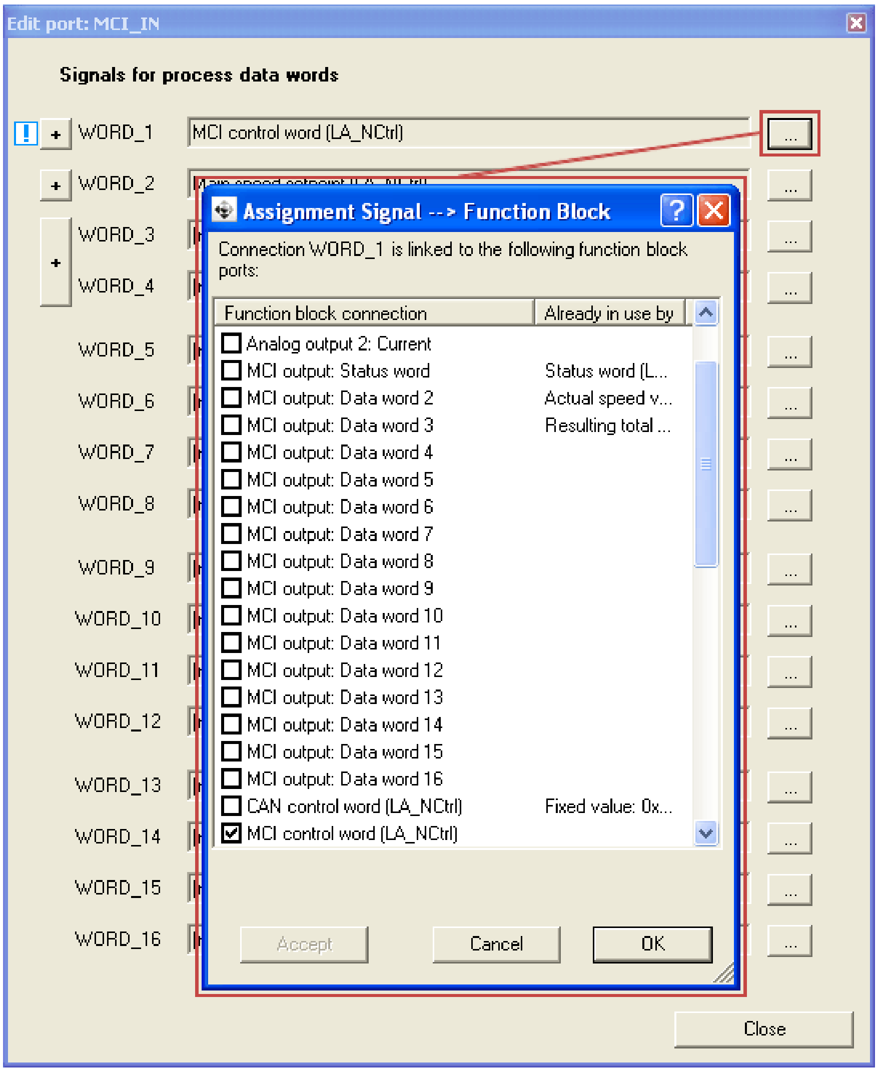
2. The **Ports** tab displays the port blocks MCI_IN and MCI_OUT.



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

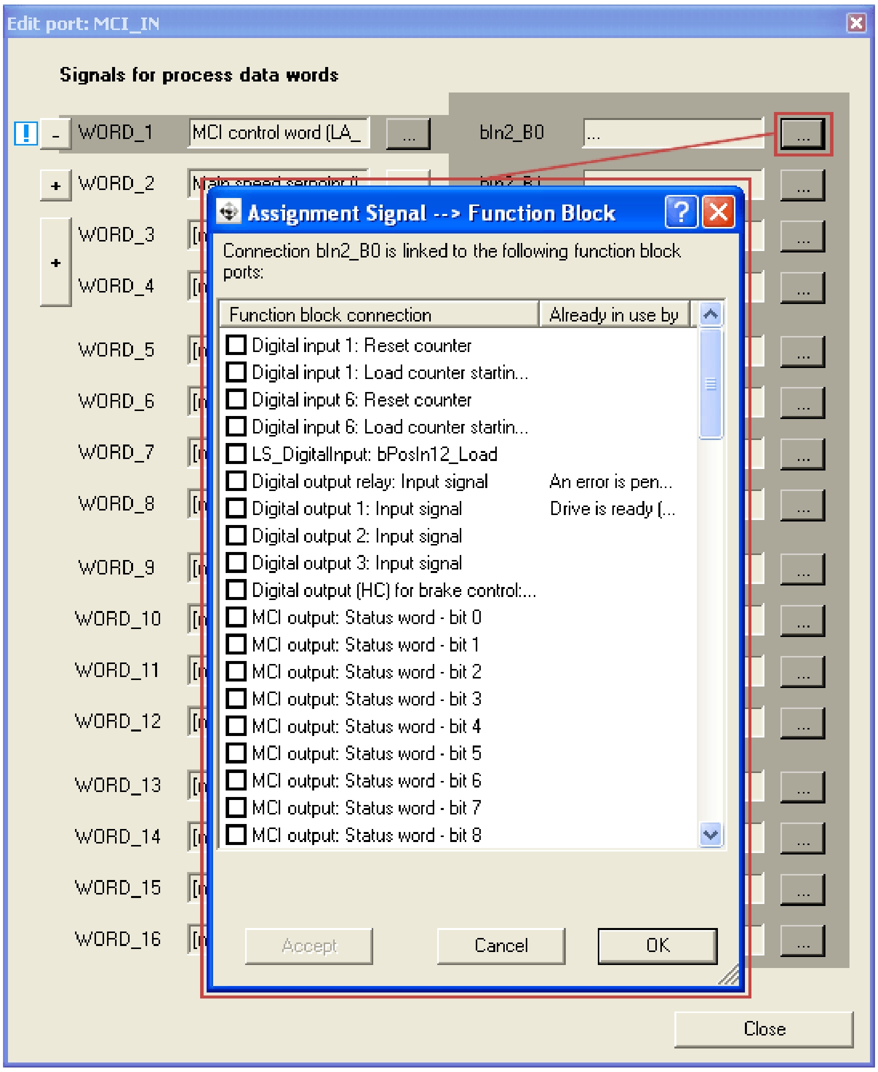


4. Via the ... button, you can assign signals to the process data words in the *Assignment Signal --> Function Block* dialog window.
   → Select the signals and then confirm the selection with **OK**.



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)**.

8.3 Digital and analog input information

In addition to the process data, also digital and analog input information can be queried. The signals are stored permanently on 2 additional data words which have to be parameterised correspondingly in the HW manager. These two data words can even be accessed if the mains voltage of the inverter is switched off and only the communication unit is supplied externally with DC 24 V. For all other data words, the mains voltage of the inverter must be switched on.

Word	Bit	Function	Values / states
1	0 ... 9	Analog input value (0 ... 10 V)	10 V = 1000_{dec} (1111101000_{bin})
	10	Digital input 3	0: Closed / not active 1: Open / active
	11	Digital input 4	0: Closed / not active 1: Open / active
	12	Digital input 5	0: Closed / not active 1: Open / active
	13	Reserved	
	14	I/O status	0: I/O data are invalid. 1: I/O data are valid. • For determining the I/O status, the checksums are generated from the I/O data in the master and in the slave (inverter), respectively, and are then compared to each other.
	15	Drive status	0: Inverter is 'offline'. 1: Inverter is 'online'.
2	0	RFR (controller enable)	0: Inverter is enabled. 1: Inverter is not enabled (inhibited).
	1	Digital input 1	0: Closed / not active 1: Open / active
	2	Digital input 2	0: Closed / not active 1: Open / active
	3	Digital input 3	0: Closed / not active 1: Open / active
	4	Digital input 4	0: Closed / not active 1: Open / active
	5	Digital input 5	0: Closed / not active 1: Open / active
	6 ... 13	Reserved	
	14	I/O status	0: I/O data are invalid. 1: I/O data are valid.
	15	Drive status	0: Inverter is 'offline'. 1: Inverter is 'online'.

9 Parameter data transfer

The PROFIBUS communication unit supports the cyclic and acyclic transmission of parameter data:

- Cyclic DP-V0 parameter data are based on the DRIVECOM profile.
  If the DP-V0 parameter data channel is active, it additionally occupies 4 words of the input and output data.
- The acyclic DP-V1 parameter data are based on the PROFIdrive profile (PROFIDrive (DP-V1) in preparation)

9.1 Addressing of the parameter data

The parameter data are addressed via codes which you'll find in a code table in this documentation and in the corresponding documentation of your inverter.

▸ Parameter reference (📖 79)

Addressing of Lenze parameters

In the case of the DP-V0 parameter data channel, the parameters of a device are not addressed directly via Lenze code numbers, but via indices (bytes 3 + 4) and subindices (byte 2).

- The conversion is made via an offset (24575 / 0x5FFF):
  - PROFIBUS-Index$_{dec}$ = 24575 - Lenze code numbers
  - PROFIBUS-DP index$_{hex}$ = 0x5FFF - Lenze code number$_{hex}$
- Example of C00105 (quick stop deceleration time):
  - PROFIBUS-Index$_{dec}$ = 24575 - 105 = 24470
  - PROFIBUS-DP index$_{hex}$ = 0x5FFF - 0x69 = 0x5F96
- The parameter values are entered into the user data (bytes 5 to 8) of the telegram.

9.2 DRIVECOM parameter data channel (DP-V0)

The DRIVECOM parameter data channel (DP-V0) ...

- enables the parameterisation and diagnostics of the inverter.
- provides access to all Lenze parameters (codes).
- additionally occupies 4 words (16 bits/word) of the input and output data words in the master.
- is identical for both transmission directions.

9.2.1 Telegram structure (overview)

The telegram of the parameter data channel consists of a total of 8 bytes:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

The individual bytes are described in detail in the following subchapters.

9.2.2 Byte 1: Service

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

Request and response control for the parameter data channel

7	6	5	4	3	2	1	0

[9-1] Method of counting for bits 0 ... 7

Bit 0 ... 2: Request	
Read/write request from the master to the inverter	
000	No request
001	Read request ▸ Reading parameter data from the inverter (📖 42)
010	Write request (write data to the inverter) ▸ Reading parameter data from the inverter (📖 42)
100	Data transfer abort by the master ▸ Data transfer abort by the master (📖 43)

Bit 3
Reserved

Bit 4/5: Data length	
Data length ≤ 4 bytes in the telegram bytes 5 ... 8 (data 1 ... 4 / error 1 ... 4)	
00	1 byte
01	2 bytes
10	3 bytes
11	4 bytes

Bit 6: Handshake
Indicates a new request. • The state of this (toggle) bit is changed by the master for every new request. • The inverter copies the bit into its response message.

Bit 7: Status	
Status information from the inverter to the master with the order confirmation. • This status bit informs the master whether the request has been carried out without errors.	
0	Request completed without errors.
1	Request not completed because of an error. • The status bit set indicates that the telegram is an "error telegram". The data of bytes 5 ... 8 (data/error) must be interpreted as an error message. ▸ Error codes (📖 46)

9.2.2.1 Reading parameter data from the inverter

General procedure:

1. Determine the user data area of the inverter, i. e. where is the storage location of the DP user data in the master computer (observe manufacturer-specific details).
2. Enter the address of the required parameter in the "Index" and "Subindex" fields (DP output data).
3. Request in the service byte = read request.
   - The handshake bit in the service byte must be changed (DP output data).
4. Check whether the handshake bit in the service byte is the same for the DP input data and the DP output data.
   - If the handshake bit is the same, the response has been received.
   - It is useful to implement a time monitoring tool.
5. Check whether the status bit in the service byte is set:
   - Status bit is not set: The "Data/Error" field contains the required Parameter value (data) (📖 45).
   - Status bit is set: The read request has not been executed correctly. The "Data/Error" field contains the Error codes (📖 46).

9.2.2.2 Writing parameter data to the inverter

General procedure:

1. Determine the user data area of the inverter, i. e. where is the storage location of the DP user data in the master computer (observe manufacturer-specific details).
2. Enter the address of the required parameter in the "Index" and "Subindex" fields (DP output data).
3. Enter the parameter value in the "Data/Error" field.
4. Request in the service byte = write request.
   - The handshake bit in the service byte must be changed (DP output data).
5. Check whether the handshake bit in the service byte is the same for the DP input data and the DP output data.
   - If the handshake bit is the same, the response has been received.
   - It is useful to implement a time monitoring tool.
6. Check whether the status bit in the service byte is set:
   - Status bit is not set: The write request has been executed correctly.
   - Status bit is set: The write request has not been executed correctly. The "Data/Error" field contains the Error codes (📖 46).

9.2.2.3 Data transfer abort by the inverter

To abort the transfer, the error telegram is used.

- The error telegram is marked by a set status bit in the service byte.
- The telegram can either be the response to an "Initiate Read/Write Service" or to a "Read/Write Segment Service".

Response of the inverter in the event of an error:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1
1t110000	SIDX	IDXH	IDXL	Error Class	Error code	Additional Code High	Additional Code Low

9.2.2.4 Data transfer abort by the master

The master can use this error telegram to abort a running segment transmission.

- The error telegram is marked by a set status bit in the service byte.
- The service byte also contains the request code "4" (100_{bin}).
- Bit 4 and bit 5 in the service byte (data length) are without meaning.
- Additional information (subindex, index, error information) is not transmitted.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1txx0100	0	0	0	0	0	0	0

Response of the inverter if no errors occur:

The inverter also acknowledges the error telegram of the master with an error telegram.

- The error telegram is marked by a set status bit in the service byte.
- In the case of correct execution, the telegram contains the error information "0x00000000" in bytes 5 ... 8.
- Additional information (subindex, index) is not transmitted.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	SIDX	IDXH	IDXL	Error Class	Error code	Additional Code High	Additional Code Low
1t110000	0	0	0	0	0	0	0

9.2.3 Byte 2: Subindex

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	**Subindex**	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

Additional addressing via the subindex is required for those codes of the Inverter Drives 8400 motec that contain subcodes (see code table).

9.2.4 Bytes 3 + 4: Index

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	**Index High byte**	**Index Low byte**	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

The parameter (Lenze code) is selected via these two bytes according to the formula:

- **Index = 24575 - Lenze code number**

(Also see "Addressing of Lenze parameters" (📖 39))

Example :

The parameter C00105 (quick stop (QSP) deceleration time) is to be addressed:

- Index = 24575 - 105 = 24470 = 0x5F96
- The entries in bytes 3 + 4 for this example would be:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	**0x5F**	**0x96**	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

9.2.5 Bytes 5 ... 8: Parameter value / error information

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	**Data 4 / Error 4**	**Data 3 / Error 3**	**Data 2 / Error 2**	**Data 1 / Error 1**

The state of status bit 7 in the service byte determines the meaning of this data field:

Status bit	Meaning of bytes 5 ... 8
0	Bytes 5 ... 8 contain the parameter value (data 1 ... 4). ▸ Parameter value (data) (📖 45)
1	Bytes 5 ... 8 contain an error message (error 1 ... 4) due to an invalid access. ▸ Error codes (📖 46)

Parameter value (data)

Note!

Strings or data blocks cannot be transmitted.

Depending on the data format, the length of the parameter value is between 1 and 4 bytes.

- Data are saved in the Motorola format, i.e. first the high byte (high word), then the low byte (low word):

<table>
<tr><th>Byte 5</th><th>Byte 6</th><th>Byte 7</th><th>Byte 8</th></tr>
<tr><td>High byte</td><td>Low byte</td><td>High byte</td><td>Low byte</td></tr>
<tr><td colspan="2">High word</td><td colspan="2">Low word</td></tr>
<tr><td colspan="4">Double word</td></tr>
</table>

- Principle for the assignment of bytes 5 ... 8 with parameter values of different lengths:

<table>
<tr><th>Byte 5</th><th>Byte 6</th><th>Byte 7</th><th>Byte 8</th></tr>
<tr><td>Parameter value (length 1)</td><td>00</td><td>00</td><td>00</td></tr>
<tr><td colspan="2">Parameter value (length 2)</td><td>00</td><td>00</td></tr>
<tr><td colspan="4">Parameter value (length 4)</td></tr>
</table>

9.2.6 Error codes

The following error messages may appear:

Byte 8	Byte 7	Byte 6	Byte 5	Meaning
Error 1	Error 2	Error 3	Error 4	
0x06	0x03	0x00	0x00	No right to access
0x06	0x05		0x11	Invalid subindex
0x06	0x05		0x12	Data length too large
0x06	0x05		0x13	Data length too small
0x06	0x07		0x00	Object does not exist
0x06	0x08		0x00	Data types do not comply with each other
0x08	0x00		0x00	Request cannot be executed
0x08	0x00		0x20	Request cannot be executed at the moment
0x08	0x00		0x22	Request cannot be executed due to the device status / The parameter can only be changed in the case of a controller inhibit
0x08	0x00		0x30	Out of value range
0x08	0x00		0x31	Parameter value too high
0x08	0x00		0x32	Parameter value too low
0x08	0x00		0x80	Hardware error

9.2.7 Telegram examples

9.2.7.1 Read request: Querying the heatsink temperature

The heatsink temperature of the inverter is to be read.

- Code to be read: C00061
- Heatsink temperature: 43 °C

Byte 1: Service (request)

- Request = $0t110001_{bin}$
  - Bit 0 ... 2 = 001_{bin} for read request
  - Bit 3 = 0 (reserved)
  - Bit 4/5 = 01_{bin} for 2-byte data length (only relevant for the response telegram)
  - Bit 6 = handshake bit (t ≡ status is changed in the response telegram)
  - Bit 7 = status bit (only relevant for the response telegram)

Byte 2: Subindex

- Subindex = 0 because code C00061 does not contain any subindices.

Bytes 3 + 4: Index

- Index = 24575 - code number = 24575 - 61 = 24514 = 0x5FC2
  - Byte 3 (high byte) = 0x5F
  - Byte 4 (low byte) = 0xC2

Bytes 5 ... 8: Data

- The response telegram contains the value of code C00061:
  - Data 3 + 4 = 43 [°C] x 1 (internal factor) = 43 = 0x002B

Result:

- Request telegram from master to drive:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index high byte	Index low byte	Data 4	Data 3	Data 2	Data 1
0x01	0x00	0x5F	0xC2	0x00	0x00	0x00	0x00
$0t000001_{bin}$	00000000_{bin}	01011111_{bin}	11000010_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}
Waiting for change of handshake bit 6 in service byte 1 of the response.							

- Response telegram from drive to master (for correct execution):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index high byte	Index low byte	Data 4	Data 3	Data 2	Data 1
0x11	0x00	0x5F	0xC2	0x00	0x2B	0x00	0x00
$0t010001_{bin}$	00000000_{bin}	01011111_{bin}	11000010_{bin}	00000000_{bin}	00101011_{bin}	00000000_{bin}	00000000_{bin}

9.2.7.2 Write request: Setting the deceleration time for quick stop (QSP)

In the inverter, the deceleration time for quick stop (QSP) is to be set to 50 ms.

- Code to be written: C00105

Byte 1: Service (request)

- Request = 0t110010$_{bin}$
  - Bit 0 ... 2 = 010$_{bin}$ for write request
  - Bit 3 = 0 (reserved)
  - Bit 4/5 = 11$_{bin}$ for 4-byte data length
  - Bit 6 = handshake bit (t ≡ status is changed in the response telegram)
  - Bit 7 = status bit (only relevant for the response telegram)

Byte 2: Subindex

- Subindex = 0 because code C00105 does not contain any subindices.

Bytes 3 + 4: Index

- Index = 24575 - code number = 24575 - 105 = 24470 = 0x5F96
  - Byte 3 (high byte) = 0x5F
  - Byte 4 (low byte) = 0x96

Bytes 5 ... 8: Data

- The parameter value of 0.05 s to be set is multiplied by the code-specific factor of "1000" and entered in the user data:
  - Data 1 ... 4 = 0.05 [s] x 1000 (internal factor) = 50 = 0x00000032

Result:

- Request telegram from master to drive:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index high byte	Index low byte	Data 4	Data 3	Data 2	Data 1
0x72	0x00	0x5F	0x96	0x00	0x00	0x00	0x32
0t110010$_{bin}$	00000000$_{bin}$	01011111$_{bin}$	10010110$_{bin}$	00000000$_{bin}$	00000000$_{bin}$	00000000$_{bin}$	00110010$_{bin}$
Waiting for change of handshake bit 6 in service byte 1 of the response							

- Response telegram from drive to master (for correct execution):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index high byte	Index low byte	Data 4	Data 3	Data 2	Data 1
0x40	0x00	0x5F	0x96	0x00	0x00	0x00	0x32
0t000000$_{bin}$	00000000$_{bin}$	01011111$_{bin}$	10010110$_{bin}$	00000000$_{bin}$	00000000$_{bin}$	00000000$_{bin}$	00110010$_{bin}$

9.3 PROFIdrive parameter data channel (DP-V1)

Data communication with PROFIBUS DP-V0 is characterised by cyclic diagnostics and cyclic process data and parameter data transfer.

An optional service extension is the acyclic parameter data transfer of PROFIBUS DP-V1 (in preparation). This service does not impair the functionality of the standard services under PROFIBUS DP-V0.

PROFIBUS DP-V0 and PROFIBUS DP-V1 can be operated simultaneously in the same network. This enables the step-by-step expansion or modification of a system.

The services of PROFIBUS DP-V1 can be used by the class 1 master (PLC) and the class 2 master (diagnostics master, etc.).

The integration of the acyclic service into the fixed bus cycle depends on the corresponding configuration of the class 1 master:

- With configuration, a time slot is reserved.
- Without configuration the acyclic service is *appended* when a class 2 master acyclically accesses a DP-V1 slave.

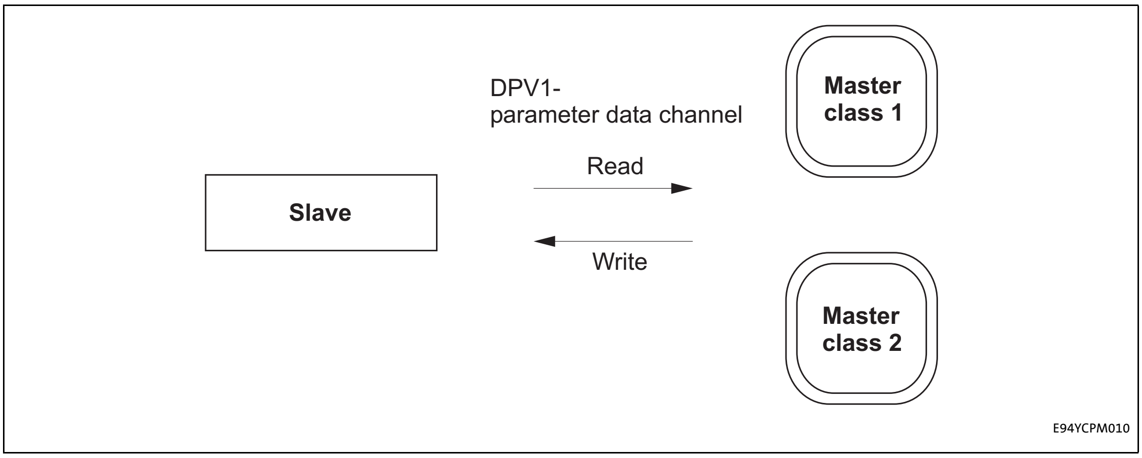
Product features

- Parameter number and subindex addresses with a width of 16 bits each.
- Several parameter requests can be combined to one request (multi-parameter request).
- There is always only one parameter request in process (no pipelining).
- A parameter request/response must fit into a data block (max. 240 bytes). Requests/responses cannot be split into several data blocks.
- No spontaneous messages are transferred.
- There are only acyclic parameter requests.
- Profile-specific parameters can be read independently of the slave state.

9.3.1 Connection establishment between master and slave

A class 1 master can always be used to request parameters from a slave if the slave is in the "Data_Exchange" state.

In addition to the class 1 master, a class 2 master can establish a communication connection to the slave:

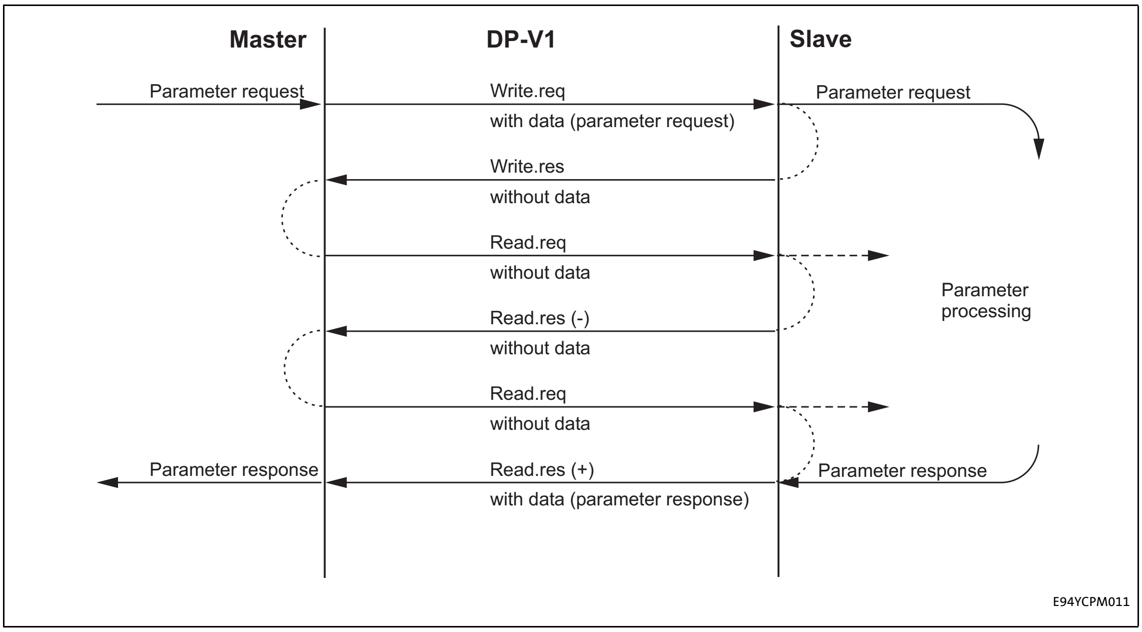


[9-2] Data communication via the DP-V1 parameter data channel

9.3.2 Acyclic data transfer

Note!

A parameter request refers to one or several parameter(s) (multi-parameter request).

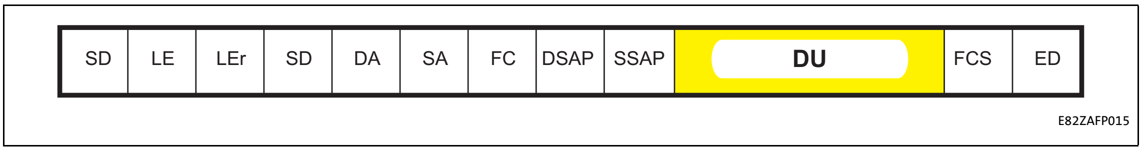


[9-3] Transmission directions

Explanation

- A "Write.req" is used to pass the data set (DB47) to the slave in the form of a parameter request.
- With "Write.res" the master receives the confirmation for the receipt of the message.
- The master requests the response of the slave with "Read.req".
- The slave responds with "Read.res (-)" if processing has not yet been completed.
- After parameter processing, the parameter request is completed by transmitting the parameter response to the master with "Read.res (+)".

9.3.3 Telegram structure



[9-4] PROFIBUS data telegram

The data unit (DU) contains the DP-V1 header and the parameter request or the parameter response.

The following subchapters describe the parameter request and the parameter response in detail.

Note!

The DP-V1 header consists of:
- Function identification
- Slot number
- Data set
- Length of the user data

Please refer to the corresponding PROFIBUS specification for further information on the DP-V1 header.

Assignment of the user data depending on the data type

Depending on the data type used, the user data are assigned as follows:

Data type	Length	User data assignment				
		Byte 1	Byte 2	Byte 3	Byte 4	Byte ...
String	x bytes					
U8	1 byte		00			
U16	2 bytes	High byte	Low byte			
U32	4 bytes	High word		Low word		
		High byte	Low byte	High byte	Low byte	

9.3.3.1 Reading parameter data from the inverter

Note!

- When a read request is processed, no parameter value is written to the slave.
- In the case of a multi-parameter read request, the parameter attribute, index, and subindex are repeated with the number "n" of the parameters requested.
- A read request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices

Field	Data type	Values
Request reference	U8	This value is specified by the master
Request identification	U8	0x01: Request parameters for reading
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter attribute

Byte 5	Byte 6
Attribute	Number of subindices

Field	Data type	Values
Attribute	U8	0x10: Value
Number of subindices	U8	0x00 (For array elements: Enter the number of array elements required.)

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Index	U16	0x0001 ... 0xFFFF (1 ... 65535)
Subindex	U16	0x0001 ... 0xFFFF (1 ... 65535)

9.3.3.2 Response to a correctly executed read request

Note!

Responses to a read request do not contain parameter attributes, indices and subindices.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x01: Parameter has been read
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter format

Byte 5	Byte 6
Format	Number of values

Field	Data type	Values
Format	U8	0x02: Integer8 0x03: Integer16 0x04: Integer32 0x05: Unsigned8 0x06: Unsigned16 0x07: Unsigned32 0x09: Visible string 0x0A: Octet string 0x40: Zero 0x41: Byte 0x42: Word 0x43: Double word
Number of values	U8	0x01 or number of requested subindices/parameters (with several subindices/parameters only the parameter value is repeated). In the case of string codes, the number of characters is entered here.

Parameter value

Byte 7	Byte 8	Byte 9	Byte 10
Value			

Field	Data type	Values
Value	String	Any (length > 4 bytes possible)
	U8	0x00 .... 0xFF
	U16	0x0000 .... 0xFFFF
	U32	0x0000 0000 .... 0xFFFF FFFF

9.3.3.3 Response to a read error

Note!

In the case of a multi-parameter request, correct and possible faulty messages are summarised in one telegram. They have the following data contents:

- Correct message:
  - Format: data type of the value requested
  - Number of values: as described in the chapter "Reading parameter data from the inverter" (📖 42).
  - Parameter value: value requested
- Faulty message
  - Format: 0x44
  - Number of values: 0x01 or 0x02
  - Error code without additional information (for number of values = 0x01) or
  - Error code with additional information (for number of values = 0x02)

A faulty access to a parameter "n" is indicated at the nth position in the response telegram of a multi-parameter request.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x81: Parameter has not been read • The data in the bytes 7 + 8 must be interpreted as an error code.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter format

Byte 5	Byte 6
Format	Number of values

Field	Data type	Values
Format	U8	0x44: Error
Number of values	U8	0x01: Error code without additional information 0x02: Error code with additional information

Error code

Byte 7	Byte 8	Byte 9	Byte 10
Error code		Additional information (if available)	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Error code	U16	0x0000 .... 0xFFFF ▸ Error codes (📖 61)
Additional information (if available)	U16	

9.3.3.4 Writing parameter data to the inverter

Note!

When a multi-parameter write request is transferred, the ...
- Parameter attribute
- Index and subindex

and then the ...
- Parameter format
- Parameter value

... are repeated with the number "n" of the parameters addressed.

A write request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices

Field	Data type	Values
Request reference	U8	This value is defined by the master.
Request identification	U8	0x02: Write parameter
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters addressed)

Parameter attribute

Byte 5	Byte 6
Attribute	Number of subindices

Field	Data type	Values
Attribute	U8	0x10: Value
Number of subindices	U8	0x00 (For array elements: Enter the number of array elements required.)

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Index	U16	0x0001 ... 0xFFFF (1 ... 65535)
Subindex	U16	0x0001 ... 0xFFFF (1 ... 65535)

Parameter format

Byte 11	Byte 12
Format	Number of values

Field	Data type	Values
Format	U8	0x02: Integer8 0x03: Integer16 0x04: Integer32 0x05: Unsigned8 0x06: Unsigned16 0x07: Unsigned32 0x09: Visible string 0x0A: Octet string 0x40: Zero 0x41: Byte 0x42: Word 0x43: Double word
Number of values	U8	0x01 or number of written subindices/parameters (with several subindices/parameters only the parameter value is repeated). In the case of string codes, the number of characters is entered here.

Parameter value

Byte 13	Byte 14	Byte 15	Byte 16
Value			

Field	Data type	Values
Value	String	Any (length > 4 bytes possible)
	U8	0x00 .... 0xFF
	U16	0x0000 .... 0xFFFF
	U32	0x0000 0000 .... 0xFFFF FFFF

9.3.3.5 Response to a correctly executed write request

Note!

In the case of a multi-parameter request, correct and possible faulty messages are summarised in one frame. They have the following data contents:

- Correct message
  - Format: 0x40
  - Number of values: 0x00
- Faulty message
  - Format: 0x44
  - Number of values: 0x01 or 0x02
  - Error code without additional information (for number of values = 0x01) or with additional information (for number of values = 0x02)

A faulty access to a parameter "n" is indicated at the nth position in the response frame of a multi-parameter request.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x02: Parameter has been written
Axis	U8	0x00 or 0x01
Number of indices	U8	0xn (n = number of parameters addressed)

9.3.3.6 Response to a write error

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x82: Parameter has not been written • The data in the bytes 7 + 8 must be interpreted as an error code.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters addressed)

Parameter format

Byte 5	Byte 6
Format	Number of values

Field	Data type	Values
Format	U8	0x44: Error
Number of values	U8	0x01: Error code without additional information 0x02: Error code with additional information

Error code

Byte 7	Byte 8	Byte 9	Byte 10
Error code		Additional information (if available)	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Error code	U16	0x0000 .... 0xFFFF ▸ Error codes (📖 61)
Additional information (if available)	U16	

9.3.4 Error codes

Error code	Description	Explanation	Additional information
0x0000	Impermissible parameter number	Access to unavailable parameter	-
0x0001	Parameter value cannot be changed	Change access to a parameter value that cannot be changed	Subindex
0x0002	Lower or upper value limit exceeded	Change access with value beyond the value limits	Subindex
0x0003	Faulty subindex	Access to unavailable subindex	Subindex
0x0004	No array	Access with subindex to non-indicated parameter	-
0x0005	Wrong data type	Change access with value that does not match the data type of the parameter	-
0x0006	No setting permitted (only resettable)	Change access with value unequal to 0 where this is not permitted	Subindex
0x0007	Description element cannot be changed	Change access to a description element that cannot be changed	Subindex
0x0008	Reserved	(PROFIdrive profile V2: PPO-Write requested in the IR is not available)	-
0x0009	Description data not available	Access to unavailable description (parameter value is available)	-
0x000A	Reserved	(PROFIdrive profile V2: Wrong access group)	-
0x000B	No parameter change rights	Change access without parameter change rights	-
0x000C	Reserved	(PROFIdrive profile V2: Wrong password)	-
0x000D	Reserved	(PROFIdrive profile V2: Text in the cyclic traffic cannot be read)	-
0x000E	Reserved	(PROFIdrive profile V2: Name in the cyclic traffic cannot be read)	-
0x000F	No text array available	Access to unavailable text array (parameter value is available)	-
0x0010	Reserved	(PROFIdrive profile V2: Missing PPO-Write)	-
0x0011	Request cannot be executed due to the operating state	Access is not possible due to temporary reasons not specified here	-
0x0012	Reserved	(PROFIdrive profile V2: Other error)	-
0x0013	Reserved	(PROFIdrive profile V2: date in the cyclic traffic cannot be read)	-
0x0014	Value impermissible	Change access with the value that is inside the value limits but not permissible for other permanent reasons (parameters with defined individual values)	Subindex
0x0015	Response too long	The length of the current response exceeds the maximum transmittable length	-
0x0016	Parameter address impermissible	Impermissible or non-supported value for attribute, number of subindices, parameter number, or subindex, or a combination	-
0x0017	Format impermissible	Write request: Impermissible or non-supported format of parameter data	-
0x0018	Number of values not consistent	Write request: Number of values of the parameter data do not match the number of subindices in the parameter address	-
0x0019	Reserved	-	-
...			
0x0064			

Error code	Description	Explanation	Additional information
0x0065	Manufacturer-specific	-	-
...			
0x00FF			

9.3.5 Telegram examples

9.3.5.1 Read request: Querying the heatsink temperature

The heatsink temperature of the inverter is to be read.

- Code to be read: C00061
- Heatsink temperature: 43 °C

Parameter request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices
0xXX	0x01	0x00	0x01
	Request parameters for reading		

Byte 5	Byte 6
Attribute	Number of subindices
0x10	0x00
Value	No subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte
0x5F	0xC2	0x00	0x00
Index = 24575 - code no. = 24575 - 61 = 24514 = 0x5F C2		No subindex	

Parameter response to a correctly executed read request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Response identification	Axis	Number of indices
0xXX	0x01	0x00	0x01
(mirrored)	Parameter has been read	(mirrored)	

Byte 5	Byte 6
Format	Number of values
0x03	0x01
Integer16	1 value

Byte 7	Byte 8
Value	
High byte	Low byte
0x00	0x2B
Value read = 0x 00 2B = 43 x 1 (internal factor) = 43 [°C]	

Parameter response to a read error

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Response identification	Axis	Number of indices
0xXX	0x81	0x00	0x01
(mirrored)	Parameter has not been read	(mirrored)	

Byte 5	Byte 6
Format	Number of values
0x44	0x01
Error	Error code without additional information

Byte 7	Byte 8
Error code	
High byte	Low byte
For the meaning, see the "Error codes" (📖 61) chapter	

9.3.5.2 Write request: Setting the deceleration time for quick stop (QSP)

In the inverter, the deceleration time for quick stop (QSP) is to be set to 50 ms.

- Code to be written: C00105

Parameter request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices
0xXX	0x02	0x00	0x01
	Write parameters	Axis 0	1 index

Byte 5	Byte 6
Attribute	Number of subindices
0x10	0x00
Value	No subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte
0x5F	0x96	0x00	0x00
Index = 24575 - code no. = 24575 - 105 = 24470 = 0x5F 96		No subindex	

Byte 11	Byte 12
Format	Number of values
0x43	0x01
Double word	1 value

Byte 13	Byte 14	Byte 15	Byte 16
Value			
High word: high byte	High word: low byte	Low- word: high byte	Low word: low byte
0x00	0x00	0x00	0x32
Value to be written = 0.05 [s] x 1000 (internal factor) = 50 = 0x00 00 00 32			

Parameter response to a correctly executed write request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Response identification	Axis	Number of indices
0xXX	0x02	0x00	0x01
(mirrored)	Parameter has been written	(mirrored)	1 index

Parameter response to a read error

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Response identification	Axis	Number of indices
0xXX	0x82	0x00	0x01
(mirrored)	Parameter has not been written	(mirrored)	1 index

Byte 5	Byte 6
Format	Number of values
0x44	0x01
Error	Error code without additional information

Byte 7	Byte 8
Error code	
High byte	Low byte
For the meaning, see the "Error codes" (61) chapter	

10 Monitoring

10.1 Permanent interruption of PROFIBUS communication

If the PROFIBUS communication is interrupted permanently, e.g. by cable breakage or failure of the PROFIBUS master, no process data are transmitted to the slave being in the "Data_Exchange" state.

After the watchdog monitoring time determined by the PROFIBUS master has expired, the response parameterised in C13880/1 is executed.

The process data are treated according to the setting in C13885. (The data sent last by the master can be used or can be set to zero.)

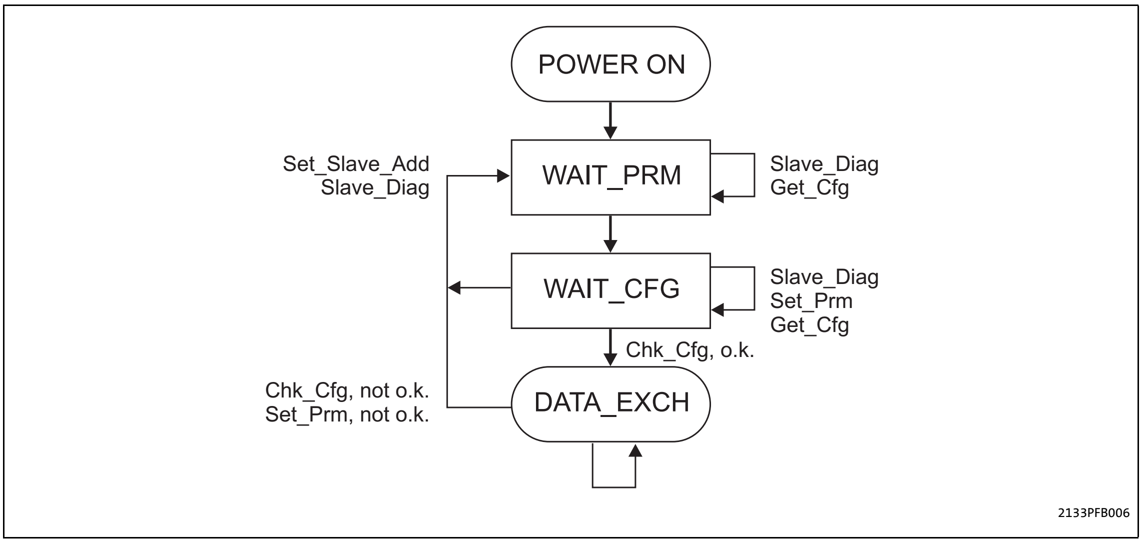
Preconditions for a response of the inverter (slave)

- A monitoring time of 1 ... 65534 ms for the "Data_Exchange" status (C13881) is set.
  A value of "65535 ms" (Lenze setting) deactivates the monitoring.
- A response for the slave is set in C13880/1 (Lenze setting "No response").
- The slave is in the "Data_Exchange" state.
- The watchdog monitoring time is configured correctly in the master.

If one of these preconditions is not met, the response to the absence of cyclic process data telegrams from the master is not executed.

▶ Settings and displays in the »Engineer« (📖 69)

10.2 Short-time interruption of PROFIBUS communication



[10-1] Sequence for short-time interruption of communication

The master detects the communication fault and only after a few microseconds it transfers the slave to the WAIT_PRM state of the DP state machine (see fig. [10-1]).

Only after the state chain of the DP state machine ending in the "Data_Exchange" state (DATA_EXCH) has been passed through, the watchdog monitoring time calculated for the slave (in milliseconds) continues to run.

Note!

The watchdog monitoring time does not continue running if the slave does not reach the "Data_Exchange" state due to repeated communication errors (e.g. caused by loose contact).

Additional monitoring for the data exchange

An additional monitoring function for data exchange is available under code C13881. This monitoring function already becomes active when the "Data_Exchange" state is exited and the parameterised time (0 ... 65535 ms) has expired. The monitoring function then triggers the response parameterised under C13880/1.

Note!

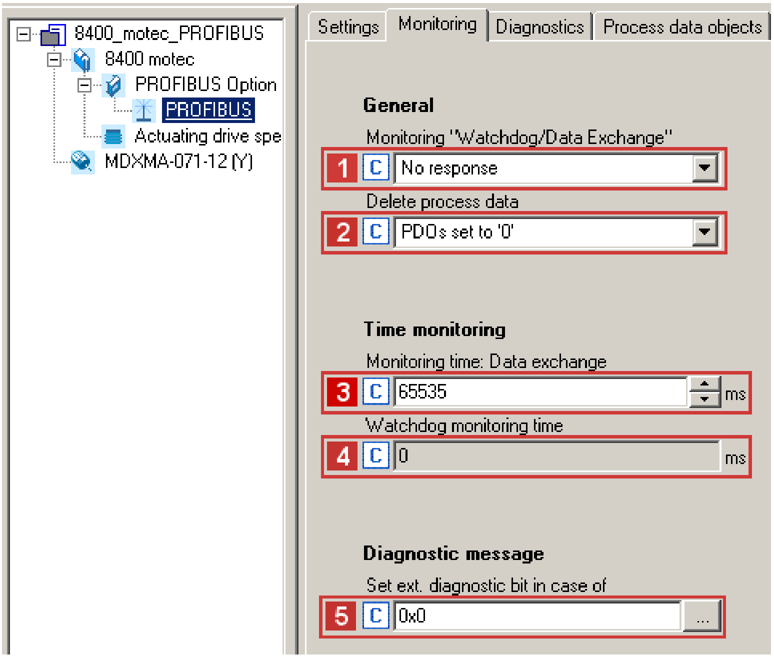
Observe the following condition for the time setting:

Monitoring time for the data exchange (C13881) ≥ watchdog monitoring time of the PROFIBUS (C13882/1).

10.3 Settings and displays in the »Engineer«

For monitoring the PROFIBUS communication, you can set a Lenze-internal monitoring time in the »Engineer« via the **Monitoring** tab 3 (C13881) and a response of the inverter 1 (C13880).

The watchdog monitoring time 4 defined in the PROFIBUS master is displayed in code C13882.



If the inverter does not receive any valid process data in the "Data_Exchange" state, the process data are treated according to the setting in 2 C13885. (In this way, the data sent last by the master can be used or set to zero.)

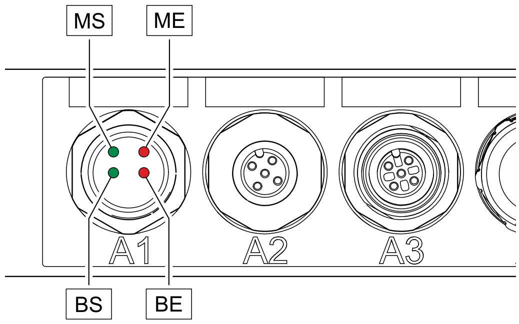
Furthermore you can set the error responses causing the external diagnostic bit ("Diag-Bit") to be set in the inverter 5 (C13886).

11 Diagnostics

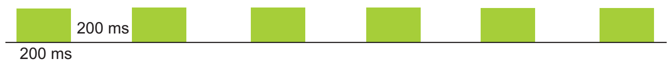
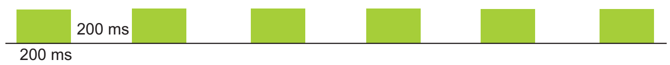
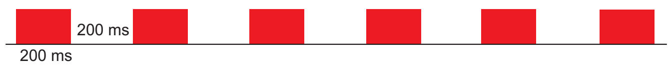
PROFIBUS communication faults can be diagnosed via the LEDs of the communication unit.

Furthermore, the »Engineer« provides some diagnostic information on PROFIBUS.

11.1 LED status displays

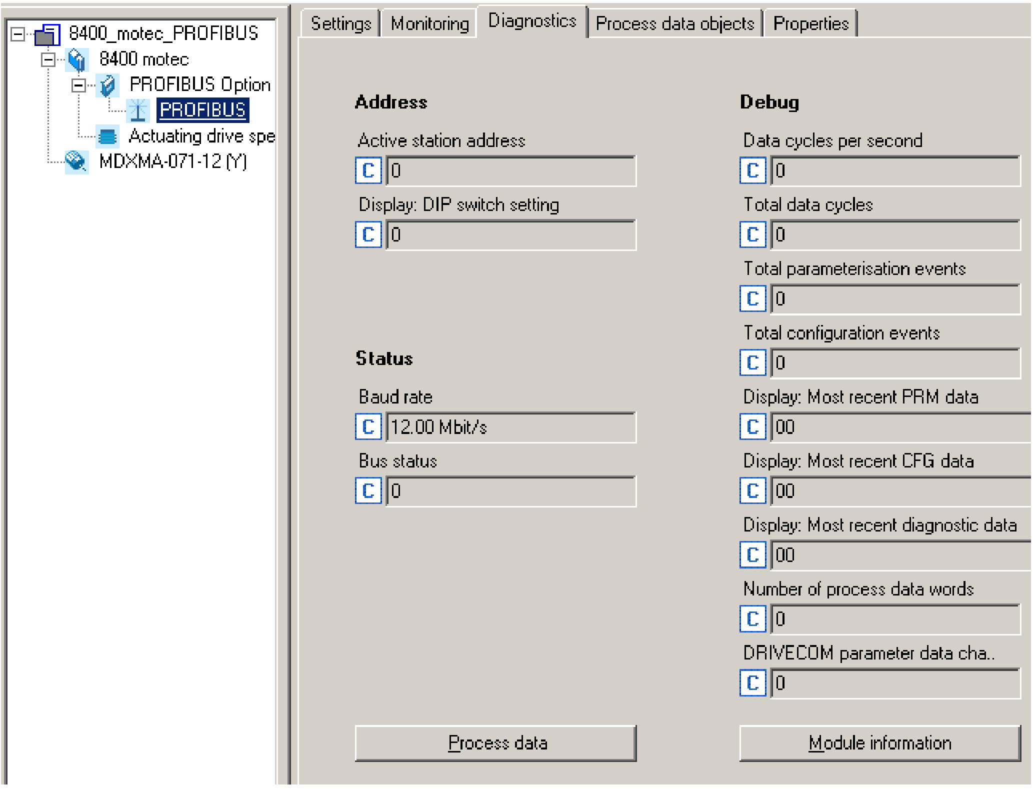


E84DG056

LED	Colour	Status	Description
MS	green	On	 The communication module is supplied with voltage and is connected to the standard device.
		blinking	 The communication module is supplied with voltage, but is not connected to the standard device. (Standard device is switched off, in the initialisation phase, or not available.)
ME	red	On	 An error has occurred in the communication module.
BS	green	Off	The communication module is not active on the fieldbus or is in the "Init" state.
		blinking	 The communication module is in the "Data_Exchange" status.
BE	red	blinking	 Incorrect setting for the station address. The communication module has been initialised and continues to work internally with the respective standard values.
		On	 Bus error/fault is active (e.g. bus cable unplugged).

11.2 Diagnostics with the »Engineer«

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



11.3 Querying the current bus status

Code C13861 displays the current PROFIBUS status in a bit-coded form:

Bit assignment				Description	
Bit 3	**Bit 2**	**Bit 1**	**Bit 0**	Reserved	
		Bit 5	**Bit 4**	**State of the DP state machine (DP-STATE)**	
		0	0	WAIT_PRM	The slave waits for a parameter data telegram after acceleration. Other types of telegrams will not be processed. Data exchange is not yet possible.
		0	1	WAIT_CFG	The slave waits for the configuration telegram that specifies the number of input and output bytes. The master informs the slave about the number of I/O bytes that will be transferred.
		1	0	DATA_EX	If the parameter settings and the configuration have been accepted by the firmware and by the application, the slave state changes to "Data_Exchange" (exchange of user data with the master).
		1	1	Not possible	
		Bit 7	**Bit 6**	**State of the Watchdog-State-Machine (WD-STATE)**	
		0	0	BAUD_SEARCH	The slave is able to recognise the baud rate automatically.
		0	1	BAUD_CONTROL	After recognising the correct baud rate, the slave status changes to BAUD_CONTROL, and the baud rate is monitored.
		1	0	DP_CONTROL	The DP_CONTROL state serves for response monitoring of the master.
		1	1	Not possible	
Bit 11	**Bit 10**	**Bit 9**	**Bit 8**	**Recognised PROFIBUS baud rate**	
0	0	0	0	12 Mbps	
0	0	0	1	6 Mbps	
0	0	1	0	3 Mbps	
0	0	1	1	1.5 Mbps	
0	1	0	0	500 kbps	
0	1	0	1	187.5 kbps	
0	1	1	0	93.75 kbps	
0	1	1	1	45.45 kbps	
1	0	0	0	19.2 kbps	
1	0	0	1	9.6 kbps	
Bit 15	**Bit 14**	**Bit 13**	**Bit 12**	Reserved	

11.4 Diagnostic data

- Present diagnostic data are signalled to the master by the PROFIBUS slave via an alarm message.
- Errors and warnings of the inverter are sent to the master as extended diagnostic messages.

General structure of diagnostic messages

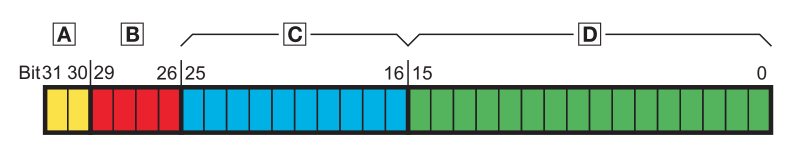
Byte	Description
1	Bit 0: Station does not exist (set by the master). Bit 1: Slave is not ready for data exchange. Bit 2: Configuration data do not correspond. Bit 3: Slave has extended diagnostic data. Bit 4: Requested function is not supported by the slave. Bit 5: Slave response is invalid (set by the master) Bit 6: Incorrect parameter setting Bit 7: Slave has been parameterised by another master (set by the master).
2	Bit 0: Slave must be parameterised again. Bit 1: Static diagnostics Bit 2: Permanently set to "1". Bit 3: Watchdog active Bit 4: Freeze command received. Bit 5: Sync command received. Bit 6: Reserved Bit 7: Slave is deactivated (set by the master).
3	Bit 7: Diagnostics overflow - amount of diagnostic information present in the slave is too large to fit into one telegram.
4	Bits 0 ... 7: Master address after parameterisation ("0xFF" without parameterisation)
5	Bits 0 ... 7: ID number (high byte)
6	Bits 0 ... 7: ID number (low byte)
7	header • The header contains the block length of the advanced diagnostics including the header byte. • In this case, the value of the entry is "0x0A" (bytes 7 ... 16 = 10 bytes).

8	Status_Type
	• The value of this entry is fixed. For the following bit assignment it is "0x81":
	  • Bit 7 = 1: "status"
	  • Bit 0 = 1: "status message"
	  • Value of all other bits = 0

9	Slot_Number • The value of the slot number is "0x00".
10	Specifier • An indicated error is entered in the specifier with the identification "0x1" (status coming). • An eliminated error is entered in the specifier with the identification "0x02" (status going). • If no errors are indicated, the entry in the specifier has the value "0x00" (no further differentiation).
11	Reserved
12	
13 ... 16	Error code of the inverter

Error code of the Inverter Drive 8400 motec

If an error occurs in the inverter, a 32-bit value is stored in the error format in the logbook, composed of the following information:



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

[11-1] Structure of the error number

- Bytes 13 ... 16 of the diagnostic message contain the error code of the inverter.
- In the logbook and in code **C00165**, the error number is shown in the following syntax in order to facilitate the readability:
  [error type].[error subject area no.].[error ID]

Software manual/»Engineer« online help for Inverter Drives 8400 motec
Here you'll find detailed information on the structure and contents of the error codes.

Example: Error "Short circuit (OC1)"

Byte	Value [hex]	Description
1	x	Standard data (PRM_Fault)
...		
6		
7	0A	Block length of the advanced diagnostics = 10 bytes
8	81	Status message
9	00	Slot 0
10	01	Status coming
11	00	
12	00	
13	0B	Error message 0x11C4000B ("Short circuit (OC1)") • Error type: "Warning locked" • Subject area: 0x11C4 (current) • Error ID: 0x000B Thus, the error number "0x11C4000B" means: An overcurrent has been detected in the "current" subject area causing a "Warning locked" error response which must be unlocked after the error has been removed.
14	00	
15	C4	
16	11	

12 Error messages

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

Software manual/»Engineer« online help for the Inverter Drive 8400 motec

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

12.1 Short overview of the PROFIBUS error messages

The following table lists all PROFIBUS error messages in the numerical order of the error numbers. Furthermore, the preset error response and - if available - the parameter for setting the error response are specified.

Tip!

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

Error number			Error text	Error type	Adjustable in
hex	dec (Subject area no.)	dec (Error no.)			
0x01bc3100	444	12544	Connection to 8400 standard device lost	1: No Response	C01501/2
0x01bc5531	444	21809	Memory: No access	1: No Response	C01501/2
0x01bc5532	444	21810	Memory. Read error	1: No Response	C01501/2
0x01bc5533	444	21811	Memory. Write error (nt14: COM fault 14)	1: No Response	C01501/2
0x01bc6010	444	24592	Restart after watchdog reset	1: No Response	C01501/2
0x01bc6011	444	24593	Internal error	1: No Response	C01501/2
0x01bc6100	444	24832	Internal error	1: No Response	C01501/2
0x01bc6101	444	24833	Internal error	1: No Response	C01501/2
0x01bc6110	444	24848	Internal error (nt15: COM fault 15)	1: No Response	
0x01bc641f	444	25631	Invalid parameter set	1: No Response	C01501/2
0x01bc6420	444	25632	Error: Lenze settings loaded	1: No Response	
0x01bc8130	444	33072	Profibus Watchdog: Monitoring time elapsed (nt03: COM fault 3)	0: No Response	C13880/1
0x01bc8131	444	33073	Profibus: State Data_Exchange left (nt04: COM fault 4)	0: No Response	C13880/1
0x01bc8132	444	33074	Profibus Watchdog: DP-V1 MSC2 monitoring time exceeded (nt05: COM fault 5)	0: No Response	C13880/1

12.2 Possible causes and remedies

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

Connection to 8400 standard device lost [0x01bc3100]

Response (Lenze setting printed in bold)	**Setting:** C01501/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ warning ☐ Information	

Cause	Remedy
• The Communication Unit is supplied with external voltage, but the Inverter Drive 8400 motec is not supplied with voltage. • The Communication Unit is not connected correctly to the Drive Unit.	• Switch off and on again the voltage supply of the Inverter Drive 8400 motec. • Check wiring and terminals. • Check internal plug connection between Communication Unit and Drive Unit. For this purpose, the Inverter Drive 8400 motec must be unscrewed. Please observe the information in the mounting instructions of the Communication Unit and the Drive Unit! • If this error continues to occur, please contact the Lenze Service. (if required, the Communication Unit must be replaced.)

Memory: No access [0x01bc5531]

Response (Lenze setting printed in bold)	**Setting:** C01501/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ warning ☐ Information	

Cause	Remedy
Access to memory was not possible.	Send the device and a description of the fault to Lenze.

Memory. Read error [0x01bc5532]

Response (Lenze setting printed in bold)	**Setting:** C01501/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ warning ☐ Information	

Cause	Remedy
Parameter could not be read.	• Download application again (including module). • Send the device and a description of the fault to Lenze.

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

Response (Lenze setting printed in bold)	**Setting:** C01501/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ warning ☐ Information	

Cause	Remedy
Parameter could not be written.	• Download application again (including module). • Send the device and a description of the fault to Lenze.

Restart by watchdog reset [0x01bc6010]

Response (Lenze setting printed in bold)	**Setting:** C01501/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ warning ☐ Information	

Cause	Remedy
Device is defective.	Send the device and a description of the fault to Lenze.

Internal error [0x01bc6011]

Response (Lenze setting printed in bold)	**Setting:** C01501/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ warning ☐ Information	

Cause	Remedy
Device is defective.	Send the device and a description of the fault to Lenze.

Internal error [0x01bc6100]

Response (Lenze setting printed in bold)	**Setting:** C01501/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ warning ☐ Information	

Cause	Remedy
Device is defective.	Send the device and a description of the fault to Lenze.

Internal error [0x01bc6101]

Response (Lenze setting printed in bold)	**Setting:** C01501/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ warning ☐ Information	

Cause	Remedy
Device is defective.	Send the device and a description of the fault to Lenze.

Internal error [0x01bc6110] (nt15: COM fault 15)

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ warning ☐ Information	

Cause	Remedy
Device is defective.	Send the device and a description of the fault to Lenze.

Invalid parameter record [0x01bc641f]

Response (Lenze setting printed in bold)	**Setting:** C01501/2 (☑ Adjustable response)
☑ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☑ Warning locked ☐ warning ☐ Information	

Cause	Remedy
No active parameter set could be loaded.	• Download application again (including module). • Send the device and a description of the fault to Lenze.

Error: Lenze settings loaded [0x01bc6420]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ warning ☐ Information	

Cause	Remedy
Access to parameter set was not possible.	• Download application again (including module). • Send the device and a description of the fault to Lenze.

Profibus watchdog: monitoring time exceeded [0x01bc8130] (nt03: COM fault 3)

Response (Lenze setting printed in bold)	**Setting:** C13880/1 (☑ adjustable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
Permanent interruption of communication to PROFIBUS master. • Also see the chapter "Permanent interruption of PROFIBUS communication" (📖 67).	Check cables and terminals. **Note:** We recommend to set "Warning locked" for the response (no drive-relevant response).

Profibus: Data_Exchange status exited [0x01bc8131] (nt04: COM fault 4)

Response (Lenze setting printed in bold)	**Setting:** C13880/1 (☑ adjustable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
Data exchange via PROFIBUS has been stopped. • Also see the chapter "Permanent interruption of PROFIBUS communication" (📖 67).	Check cables and terminals. The slave must receive new parameterisation and configuration files from the master in order to be able to exchange data again.

Profibus Watchdog: DP-V1 MSC2 monitoring time exceeded [0x01bc8132] (nt05: COM fault 5)

Response (Lenze setting printed in bold)	**Setting:** C13880/1 (☑ adjustable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
Permanent interruption of communication to PROFIBUS master. • Also see the chapter "Permanent interruption of PROFIBUS communication" (📖 67).	Check cables and terminals. **Note:** We recommend to set "Warning locked" for the response (no drive-relevant response).

13 Parameter reference

This chapter complements the parameter list and table of attributes in the software manual and the »Engineer« online help for the Inverter Drive 8400 motec by the parameters for the PROFIBUS communication.

Software manual/»Engineer« online help "Inverter Drive 8400 motec"
Here you will find general information on parameters.

13.1 Communication-relevant parameters of the operating system

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

C01501

<table>
<tr><td colspan="2">Parameter | Name:
C01501 | Response in case of communication fault with MCI</td><td>Data type: UNSIGNED_8
Index: 23074_d = $5A22_h$</td></tr>
<tr><td colspan="3">Configuration of monitoring functions for the Communication Unit</td></tr>
<tr><td colspan="2">**Selection list**</td><td></td></tr>
<tr><td>0</td><td>No response</td><td></td></tr>
<tr><td>1</td><td>Error</td><td></td></tr>
<tr><td>4</td><td>Warning Locked</td><td></td></tr>
<tr><td>**Subcodes**</td><td>**Lenze setting**</td><td>**Info**</td></tr>
<tr><td>C01501/1</td><td>1: Fault</td><td>Resp. to MCI fault 1
• Response to a communication fault.</td></tr>
<tr><td>C01501/2</td><td>1: Fault</td><td>Resp. to MCI fault 2
• Response to an incompatible communication unit.</td></tr>
<tr><td colspan="3">☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1</td></tr>
</table>

C01503

<table>
<tr><td colspan="3">Parameter | Name:
C01503 | MCI timeout</td><td>Data type: UNSIGNED_16
Index: 23072_d = $5A20_h$</td></tr>
<tr><td colspan="3">**Setting range** **(min. value | unit | max. value)**</td><td></td></tr>
<tr><td>0</td><td>ms</td><td>1000</td><td></td></tr>
<tr><td>**Subcodes**</td><td colspan="2">**Lenze setting**</td><td>**Info**</td></tr>
<tr><td>C01503/1</td><td colspan="2">200 ms</td><td>MCI timeout</td></tr>
<tr><td colspan="4">☑ Read access ☑ Write access ☐ CINH ☐ PLC STOP ☐ No transfer ☐ COM ☐ MOT Scaling factor: 1</td></tr>
</table>

13.2 Parameters relevant for PROFIBUS communication

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

C13850

Parameter | Name: **C13850 | All words to master**			Data type: UNSIGNED_16 Index: 10725_d = $29E5_h$
Display of the process data words transferred from the communication unit to the PROFIBUS master. In the subcodes 1 ... 8, all process data words to the master are displayed. However, only the configured process data words are valid.			
Display range **(min. value | unit | max. value)**			
0		65535	
Subcodes			**Info**
C13850/1			Word 1 to master
...			...
C13850/8			Word 8 to master
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13851

Parameter | Name: **C13851 | All words from master**			Data type: UNSIGNED_16 Index: 10724_d = $29E4_h$
Display of the process data words transferred from the PROFIBUS master to the communication unit. In the subcodes 1 ... 8, all process data words from the master are displayed. However, only the configured process data words are valid.			
Display range **(min. value | unit | max. value)**			
0		65535	
Subcodes			**Info**
C13851/1			Word 1 from master
...			...
C13851/8			Word 8 from master
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13852

Parameter | Name: **C13852 | All words to standard device**			Data type: UNSIGNED_16 Index: 10723_d = $29E3_h$
Display of process data words 1 ... 8 which are transmitted from the communication unit to the drive unit. Subcodes 1 ... 8 display all the process data words from the communication unit.			
Display range **(min. value | unit | max. value)**			
0		65535	
Subcodes			**Info**
C13852/1			Word 1 to drive unit
...			...
C13852/8			Word 8 to drive unit
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13853

Parameter | Name: **C13853 | All words from standard device**			Data type: UNSIGNED_16 Index: 10722_d = $29E2_h$
Display of process data words 1 ... 8 which are transmitted from the drive unit to the communication unit. Subcodes 1 ... 8 display all the process data words from the drive unit.			
Display range (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13853/1			Word 1 from drive unit
...			...
C13853/8			Word 8 from drive unit
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13860

Parameter | Name: **C13860 | Settings**			Data type: UNSIGNED_8 Index: 10715_d = $29DB_h$
Display of the current configuration data.			
Display range (min. value | unit | max. value)			
0		255	
Subcodes			**Info**
C13860/1			Reserved
C13860/2			Number of process data words • 1 ... 8 words
C13860/3			DRIVECOM parameter data channel • 0: Not active • 1: Active
C13860/4			Reserved
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13861

Parameter | Name: **C13861 | Bus status**			Data type: UNSIGNED_16 Index: 10714_d = $29DA_h$
Bit-coded display of the current bus state. ▸ Querying the current bus status (📖 72)			
Display range (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13862

Parameter | Name: **C13862 | Bus counter**			Data type: UNSIGNED_16 Index: 10713_d = $29D9_h$
When the maximum count value of 65535 is reached, the counter starts again with 0.			
Display range (min. value | unit | max. value)			
0		65535	
Subcodes		**Info**	
C13862/1		Data cycles per second	
C13862/2		Total data cycles	
C13862/3		Total parameterisation events	
C13862/4		Total configuration events	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13863

Parameter | Name: **C13863 | Baud rate**		Data type: UNSIGNED_8 Index: 10712_d = $29D8_h$
Display of the baud rate		
Selection list (read only)		
0	12.00 Mbps	
1	6.00 Mbps	
2	3.00 Mbps	
3	1.50 Mbps	
4	500.00 kbps	
5	187.50 kbps	
6	93.75 kbps	
7	45.45 kbps	
8	19.20 kbps	
9	9.60 kbps	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13864

Parameter | Name: **C13864 | Active station address**			Data type: UNSIGNED_8 Index: 10711_d = $29D7_h$
Display of the active station address ▸ Setting the station address (📖 30)			
Display range (min. value | unit | max. value)			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13865

Parameter | Name: **C13865 | Display: Most recent PRM data**	Data type: OCTET_STRING Index: 10710_d = $29D6_h$
Display of the last parameter data sent by the PROFIBUS master with the "Set-Prm" telegram (ASCII string with 24 characters)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13866

Parameter | Name: **C13866 | Display: Most recent CFG data**	Data type: OCTET_STRING Index: 10709_d = $29D5_h$
Display of the last configuration data sent by the PROFIBUS master with the "Chk-Cfg" telegram (ASCII string with 22 characters)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13867

Parameter | Name: **C13867 | Display: Most recent diagnostic data**	Data type: OCTET_STRING Index: 10708_d = $29D4_h$
Display of the last diagnostic data sent to the PROFIBUS master (ASCII string with 16 characters)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13880

Parameter | Name: **C13880 | Reaction on communication failure**		Data type: UNSIGNED_8 Index: 10695_d = $29C7_h$
Monitoring response to a communication fault on the PROFIBUS • A change in the monitoring response becomes immediately effective. ▸ Permanent interruption of PROFIBUS communication (📖 67)		
Selection list		
0	No response	
1	Error	
4	Warning Locked	
Subcodes	**Lenze setting**	**Info**
C13880/1	0: No Response	The response set here for the "watchdog/data exchange" monitoring is executed if the bus station ... • does not receive any message from the master within the watchdog monitoring time (with an active connection, it is displayed in C13882/1 / C13882/2). • detects that it is no longer in the "Data_Exchange" status. Please see also the notes given under C13881.
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13881

Parameter | Name: **C13881 | Monitoring time: Data exchange**			Data type: UNSIGNED_16 Index: 10694_d = $29C6_h$
If the "Data_Exchange" state has been exited, the response parameterised with C13880/1 is carried out when the monitoring time set here for the data exchange has expired. • A value of "65535" in this code deactivates the monitoring function. • A change of monitoring will be effective immediately. • Recommendation: The monitoring time set here should be longer than the watchdog monitoring time (displayed in C13882/1 / C13882/2). ▸ Permanent interruption of PROFIBUS communication (📖 67)			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	65535	**65535 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13882

Parameter | Name: **C13882 | Monitoring time: Watchdog**			Data type: UNSIGNED_32 Index: 10693_d = $29C5_h$
Display of the watchdog monitoring time determined by the PROFIBUS master • Monitoring starts with the receipt of the first telegram. • When a value of "0" is displayed, the monitoring function is deactivated. • A change in the watchdog monitoring time in the master will become effective immediately. ▸ Permanent interruption of PROFIBUS communication (📖 67)			
Display range **(min. value | unit | max. value)**			
0	ms	4294967295	
Subcodes			**Info**
C13882/1			Watchdog monitoring time
C13882/2			DP-V1 MSAC2
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13885

Parameter | Name: **C13885 | Delete process data**		Data type: UNSIGNED_8 Index: 10690_d = $29C2_h$
Setting of the process data which are to be further processed by the inverter for maintaining internal communication if the PROFIBUS has failed.		
Selection list **(Lenze setting printed in bold)**		
0	Use of most recent master PDOs	
1	**PDOs are set to the value 0'**	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13886

Parameter | Name: **C13886 | Set ext. diagnostic bit in case of**		Data type: BITFIELD_8 Index: 10689_d = $29C1_h$
Bit-coded selection of the error responses in the drive unit causing the external diagnostic bit ("diag bit") to be set (see PROFIBUS specification; bit 3 of byte 1 of the DP diagnostic messages). • The diagnostic bit is sent to the PROFIBUS master by the communication unit and is evaluated separately there. • The diagnostic bit is always set when a system error or an error message by the safety module occurs. • The Lenze setting "0" means that the diagnostic bit is not set for the following error responses.		
Value is bit-coded:		
Bit 0	Error	
Bit 1	Reserved	
Bit 2	Reserved	
Bit 3	Warning Locked	
Bit 4	Reserved	
...	...	
Bit 7	Reserved	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13887

Parameter | Name: **C13887 | Suppress signalling diag. mess.**			Data type: BITFIELD_8 Index: 10688_d = $29C0_h$
From version 02.00 Bit coded selection of the error responses in the drive unit, at which diagnostic signalling is suppressed.			
Value is bit-coded:		**Info**	
Bit 0	Error		
Bit 1	Fault		
Bit 2	Reserved		
Bit 3	Warning Locked		
Bit 4	Reserved		
...	...		
Bit 7	Reserved		
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13899

Parameter | Name: **C13899 | Station address**			Data type: UNSIGNED_8 Index: 10676_d = $29B4_h$
Optional setting of the station address (instead of setting via DIP switches 1 ... 64) • The station address set here only becomes effective if the DIP switch **S has been set to ON** and the DIP switches **1 ... 64 have been set to OFF** prior to power-on. • The active station address is displayed under C13864. • A change in the station address will only be effective after the parameter set has been stored and if the mains of the communication unit or the inverter has been switched again subsequently. ▸ Setting the station address (📖 30)			
Setting range (min. value | unit | max. value)			**Lenze setting**
3		126	**3**
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☑ COM ☐ MOT			

C13900

Parameter | Name: **C13900 | Firmware product type**	Data type: VISIBLE_STRING Index: 10675_d = $29B3_h$
Display of the product type (string with a length of 8 bytes) • The following identification code is displayed: "E84DGFCP".	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13901

Parameter | Name: **C13901 | Firmware compilation date**	Data type: VISIBLE_STRING Index: 10674_d = $29B2_h$
Display of the compilation date of the firmware (string with a length of 20 bytes) • The date ("MMM DD YYYY") and time ("hh:mm:ss") are output, e.g. "Mar 21 2005 12:31:21".	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13902

Parameter | Name: **C13902 | Firmware version**	Data type: VISIBLE_STRING Index: 10673_d = $29B1_h$
Display of the firmware version (string with a length of 5 bytes) • An identification code is displayed, e.g. "00.80".	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13920

Parameter | Name: **C13920 | Display: DIP switch setting**			Data type: UNSIGNED_8 Index: 10655_d = $299F_h$
Display of the current DIP switch setting • The displayed value corresponds to the sum of the individual DIP switch values 1 ... 64. • The active station address is displayed under C13864. ▸ Setting the station address (📖 30)			
Display range (min. value | unit | max. value)			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☑ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13950

Parameter | Name: **C13950 | Module internal communication status**			Data type: UNSIGNED_8 Index: 10625_d = 2981_h
Display of the internal status of the communication unit			
Selection list (read only)		**Info**	
0	Module not initialised		
1	Module ready for initialization		
2	Reading module parameters		
3	Module parameters have been read		
4	Initialisation of external protocol		
5	Online		
6	Module timeout		
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

13.3 Table of attributes

The table of attributes contains information required for a communication with the Inverter Drives 8400 motec via parameters.

How to read the table of attributes:

Column		Meaning	Entry	
Code		Parameter name	Cxxxxx	
Name		Parameter short text (display text)	*Text*	
Index	dec	Index under which the parameter is addressed. The subindex for array variables corresponds to the Lenze subcode number.	24575 - Lenze code number	Is only required for access via a bus system.
	hex		$5FFF_h$ - Lenze code number	
Data	DS	Data structure	E	Single variable (only one parameter element)
			A	Array variable (several parameter elements)
	DA	Number of array elements (subcodes)	*Number*	
	DT	Data type	BITFIELD_8	1 byte, bit-coded
			BITFIELD_16	2 bytes, bit-coded
			BITFIELD_32	4 bytes, bit-coded
			INTEGER_8	1 byte, with sign
			INTEGER_16	2 bytes with sign
			INTEGER_32	4 bytes, with sign
			UNSIGNED_8	1 byte without sign
			UNSIGNED_16	2 bytes without sign
			UNSIGNED_32	4 bytes, without sign
			VISIBLE_STRING	ASCII string
			OCTET_STRING	
	Factor	Factor for data transmission via a bus system, depending on the number of decimal positions	*Factor*	1 ≡ No decimal positions 10 ≡ 1 decimal position 100 ≡ 2 decimal positions 1000 ≡ 3 decimal positions
Access	R	Read access	☑ Reading permitted	
	W	Write access	☑ Writing permitted	
	CINH	Controller inhibit required	☑ Writing is only possible if controller inhibit is set	

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	Data type	Factor	R	W	CINH
C13850	All words from drive to master	10725	29E5	A	8	UNSIGNED_16	1	☑		
C13851	All words from master to drive	10724	29E4	A	8	UNSIGNED_16	1	☑		
C13852	All words to the basic device	10723	29E3	A	8	UNSIGNED_16	1	☑		
C13853	All words to the basic device	10722	29E2	A	8	UNSIGNED_16	1	☑		
C13860	Settings	10715	29DB	A	4	UNSIGNED_8	1	☑		
C13861	Bus status	10714	29DA	E	1	UNSIGNED_16	1	☑		
C13862	Bus counter	10713	29D9	A	4	UNSIGNED_16	1	☑		
C13863	Baud rate	10712	29D8	E	1	UNSIGNED_8	1	☑		
C13864	Active station address	10711	29D7	E	1	UNSIGNED_8	1	☑		
C13865	Display: Most recent PRM data	10710	29D6	E	1	OCTET_STRING		☑		
C13866	Display: Most recent CFG data	10709	29D5	E	1	OCTET_STRING		☑		
C13867	Display: Most recent diagnostic data	10708	29D4	E	1	OCTET_STRING		☑		
C13880	Monitoring Reaction	10695	29C7	A	1	UNSIGNED_8	1	☑	☑	
C13881	Monitoring time: Data exchange	10694	29C6	E	1	UNSIGNED_16	1	☑	☑	
C13882	Monitoring time: Watchdog	10693	29C5	A	2	UNSIGNED_32	1	☑		
C13885	Delete process data	10690	29C2	E	1	UNSIGNED_8	1	☑	☑	
C13886	Set ext. diagnostic bit in case of	10689	29C1	E	1	BITFIELD_8		☑	☑	
C13887	Suppress signalling diag. mess. upon	10688	29C0	E	1	BITFIELD_8		☑	☑	
C13899	Station address	10676	29B4	E	1	UNSIGNED_8	1	☑	☑	
C13900	Firmware Product Type	10675	29B3	E	1	VISIBLE_STRING		☑		
C13901	Firmware Compilation Date	10674	29B2	E	1	VISIBLE_STRING		☑		
C13902	Firmware Version	10673	29B1	E	1	VISIBLE_STRING		☑		
C13920	Display: DIP switch setting	10655	299F	E	1	UNSIGNED_8	1	☑		
C13950	Module internal communication status	10625	2981	E	1	UNSIGNED_8	1	☑		

13.4 Implemented PROFIdrive objects (DP-V1)

I-918

Index | Name: **0x918 | Display of station address**			Data type: U16
Display of the set station address (see also C13864)			
Display range (min. value | unit | max. value)			
1		126	
☑ Read access ☐ Write access			

I-963

Index | Name: **0x963 | Baud rate**		Data type: U16
Display of the PROFIBUS baud rate (see also C13863)		
Selection list (read only)		
0	9.6 kbps	
1	19.2 kbps	
2	93.75 kbps	
3	187.5 kbps	
4	500 kbps	
6	1.5 Mbps	
7	3 Mbps	
8	6 Mbps	
9	12 Mbps	
☑ Read access ☐ Write access		

I-964

Index | Name: **0x964 | Device identification**		Data type: U16
Display of identification data		
Subindex	Display	**Info**
0x964/0	262	Manufacturer: Lenze
0x964/1	8400	Device type
0x964/2	xxyy	Software version, e.g. 0100 (V 01.00)
0x964/3	yyyy	Firmware date (year), e.g. 2007
0x964/4	ddmm	Firmware date (day/month), e.g. 0506 (5th June)
☑ Read access ☐ Write access		

I-974

Index | Name: **0x974 | Maximum time per DPV1 parameter access**		Data type: U16
Display of access statistics		
Subindex	Display	**Info**
0x974/0	240 bytes	Maximum block length
0x974/1	40	Maximum number of parameter accesses
0x974/2	0	Maximum time per access
☑ Read access ☐ Write access		

14 DIP switch positions for setting the station address

The station address results from the sum of the binary valencies of switches 1 ... 64.

The following table shows the switch positions for the valid address range 0 ... 126.

▶ Setting the station address (📖 30)

Station address	DIP switch						
	64	32	16	8	4	2	1
0	OFF	OFF	OFF	OFF	OFF	OFF	OFF
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**

Station address	DIP switch						
	64	32	16	8	4	2	1
38	OFF	**ON**	OFF	OFF	**ON**	**ON**	OFF
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

Station address	DIP switch						
	64	32	16	8	4	2	1
81	**ON**	OFF	**ON**	OFF	OFF	OFF	**ON**
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**

Station address	DIP switch						
	64	**32**	**16**	**8**	**4**	**2**	**1**
124	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	OFF
125	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	**ON**
126	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	OFF

15 Index

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

Accessories

PROFIBUS®

E94AYCPM

Servo Drives 9400 ___________________________

Communication Manual EN

13422195

Lenze

Contents

1 About this documentation

This documentation exclusively describes the communication module E94AYCPM (PROFIBUS).

Note!

This documentation supplements the **Mounting Instructions** supplied with the communication module and the **"Servo Drives 9400" Hardware Manual**.

The Mounting Instructions contain safety instructions which must be observed!

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

Typical applications are explained with the help of examples.

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

This documentation does not describe the software of 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 host system (master).

All brand names used in this documentation are trademarks of their respective owners.

Tip!

Detailed information about PROFIBUS can be found on the website of the PROFIBUS & PROFINET user organisation:

www.profibus.com

Target group

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

Tip!

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

www.Lenze.com

Validity information

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

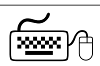
Extension module	Type designation	From hardware version	From software version
PROFIBUS communication module	E94AYCPM	1A	01.11

Screenshots/application examples

All screenshots in this documentation are application examples. Depending on the firmware version of the communication module and the software version of the Engineering tools installed (»Engineer«, »STEP7«), the screenshots in this documentation my differ from the actual screen display.

1.1 Conventions used

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

Type of information	Writing	Examples/notes
Numbers		
Decimal separator	Point	The decimal point is always used. Example: 1234.56
Hexadecimal	0x[0 ... 9, A ... F]	Example: 0x60F4
Binary • Nibble	In inverted commas Point	Example: ’100’ Example: ’0110.0100’
Text		
Version information	Text colour blue	All pieces of information that only apply to or from a specific software version of the controller are highlighted accordingly in this documentation. Example: This function extension is available from software version V3.0!
Program name	» «	The Lenze PC software »Engineer«...
Window	*italics*	The *message window*... / The *Options* dialog box ...
Variable name		Setting *bEnable* to TRUE...
Control element	**Bold**	The **OK** button ... / The **Copy** command ... / The Properties tab ... / The **Name** input field ...
Sequence of menu commands		If several successive commands are required for executing a function, the individual commands are separated from each other by an arrow: Select the command **File** ➜ **Open** to...
Hyperlink	Underlined	Optically highlighted reference to another topic. Can be activated with a mouse-click in this online documentation.
Symbols		
Page reference	(📖 8)	Optically highlighted reference to another page. Can be activated with a mouse-click in this online documentation.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

1.2 Terminology used

Term	Meaning
Controller	Lenze controller of the "Servo Drives 9400" product range
Standard device	
HW	Hardware
Lenze setting	Settings with which the device is preconfigured ex works.
Basic setting	
GSD / GSE	Device data base file (device description for PROFIBUS nodes)
	PROFIBUS® (Process Field Bus) is a widely-used fieldbus system for the automation of machines and production plants. PROFIBUS® is a registered trademark and patented technology licensed by the PROFIBUS & PROFINET International (PI) user organisation.
PDO	Process data object
PLC	Programmable Logic Controller
»STEP7«	Siemens software for programming and configuring PROFIBUS Siemens control systems
SW	Software
TCI	Tool Calling Interface

1.3 Definition of the notes used

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

Safety instructions

Structure of the safety instructions:

Danger!

(characterises the type and severity of danger)

Note

(describes the danger and informs 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 other documents

2 Safety instructions

Note!

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

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

2.1 General safety and application instructions

Danger!

If you disregard the following basic safety measures, this can cause severe injury to persons and damage to material assets.

Lenze drive and automation components ...

- must only be used as directed.
  Application as directed (📖 12)
- must never be commissioned in the event of visible damage.
- must never be technically modified.
- must never be commissioned before they have been completely mounted.
- must never be operated without the covers required.
- can - depending on their degree of protection - have live, moving or rotating parts during and after operation. Surfaces can be hot.

For Lenze drive components ...

- use only the accessories approved.
- use only original spare parts from the manufacturer.

Observe all specifications given in the attached and associated documentation.

- This is the precondition for safe and trouble-free operation and for obtaining the product features specified.
  Features (📖 13)
- The procedural notes and circuit details described in this document are only proposals. It is up to the user to check whether they can be adapted to the particular applications. Lenze does not take any responsibility for the suitability of the procedures and circuit proposals described.

Only qualified personnel may work with and on Lenze drive and automation components. According to IEC 60364 and CENELEC HD 384, these are persons ...

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

2.2 Device- and application-specific safety instructions

- During operation, the communication module must be firmly connected to the standard device.
- With external voltage supply, always use a separate power supply unit, safely separated to EN 61800-5-1 in every control cabinet (SELV / PELV).
- Only use cables corresponding to the given specifications.
  ▸ Bus cable specification (🕮 26)

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

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

2.3 Residual hazards

Protection of persons

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

▸ Protective insulation (🕮 16)

Device protection

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

▸ Installation (🕮 21)

3 Product description

3.1 Application as directed

The communication module ...

- is an accessory module for use in conjunction with the following Lenze standard devices:

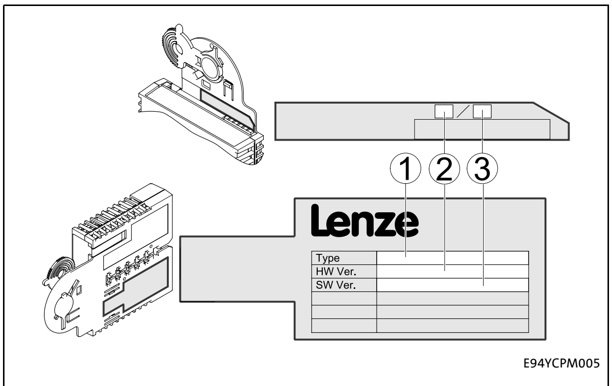
Product series	Type designation	From hardware version	From software version
Servo Drives 9400	E94AxxExxxx	1A	01.35
Servo Drives 9400 PLC	E94AxPExxxx	VA	02.00
Regenerative power supply module	E94ARNxxxx	VA	01.00

- is a device intended for use in industrial power systems.
- is only to be operated under the operating conditions specified in this documentation.
- may only be used in PROFIBUS networks.

Any other use shall be deemed inappropriate!

3.2 Identification

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



[3-1] Nameplate

1 Type designation (type)

E94	Product series
A	Version
Y	Module identification: extension module
C	Module type: communication module
PM	PROFIBUS

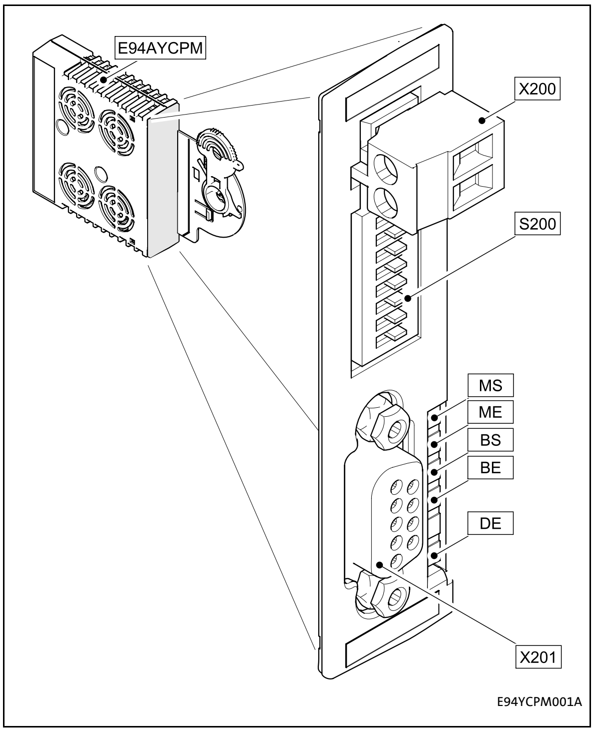
2 Hardware version (HW)

3 Software version (SW)

3.3 Features

- Interface module for the PROFIBUS communication system to be connected to the expansion slots of the Servo Drives 9400
- Support of parameter data channels DRIVECOM (DP-V0) and PROFIDrive (DP-V1)
- Transfer of safe information via the PROFIsafe protocol if an SM301 safety module (E94AYAE) is used simultaneously
- A maximum of 32 process data words per direction can be exchanged.
- The communication module can either be supplied internally by the Servo Drive 9400 or externally by a separate voltage source.
- Bus coupling via remote bus according to the RS485 standard
- Automatic detection of the baud rate (9.6 kbps to 12 Mbps)
- Setting of the station address is possible via DIP switch or code.
- Software compatibility with the EMF2133IB communication module
- Access to all Lenze parameters

3.4 Terminals and interfaces



[3-2] E94AYCPM communication module (PROFIBUS)

X200 2-pin plug connector with screw connection for external voltage supply
▶ External voltage supply (📖 28)

S200 DIP switches for setting the ...
- Station address
- compatibility with the EMF2133IB communication module

▶ Possible DIP switch settings (📖 31)

X201 PROFIBUS connection:
9-pin Sub-D socket
▶ PROFIBUS connection (📖 27)

MS ME BS BE DE 5 LED status displays for diagnostic purposes
▶ LED status displays (📖 82)

4 Technical data

"Servo Drives 9400" Hardware Manual

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

4.1 General data and operating conditions

Area	Values
Order designation	E94AYCPM
Communication profile	• PROFIBUS DP-V0 (DRIVECOM) • PROFIBUS DP-V1 (PROFIdrive)
Communication medium	RS485
Interface	9-pin Sub-D socket
Network topology	• Line (without repeater) • Tree/line (with repeater)
Bus device type	PROFIBUS slave
Number of slaves	• Max. 31 (without repeater) • Max. 125 (with repeater)
Max. cable length	1200 m (depending on the selected baud rate and the cable type used)
PNO identification number	0x07A8
Baud rate for cable type A (EN 50170)	9.6 kbps ... 12 Mbps (automatic detection)
Voltage supply	External supply via the 2-pin plug connector • Terminal "+": U = 24 V DC (20.4 V - 0 % ... 28.8 V + 0 %), I = 130 mA • Terminal "-": Reference potential for external voltage supply
Conformities, approvals	• CE • UL

4.2 Protective insulation

Danger!

Dangerous voltage

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

Possible consequences:

Death or severe injury

Protective measures:

If protection against accidental contact is required for the control terminals of the controller and the terminals of the plugged-in device modules, ...

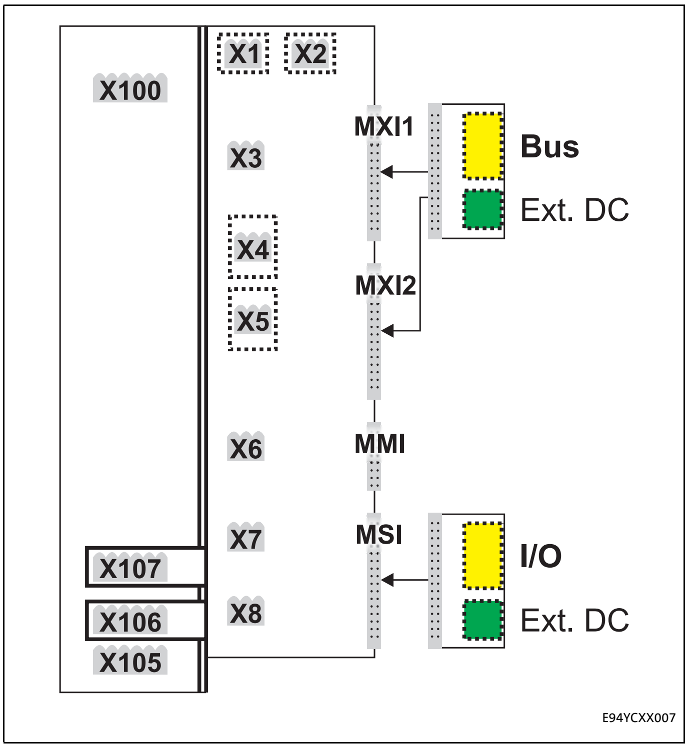
- a double isolating distance must be provided.
- the components to be connected must be provided with a second isolating distance.

Note!

The existing protective insulation in the Servo Drives 9400 is implemented according to EN 61800-5-1.

The following illustration ...

- shows the arrangement of the terminal strips and the separate potential areas of the Servo Drive 9400.
- serves to determine the decisive protective insulation between two terminals located in differently insulated separate potential areas.



|| Reinforced insulation
| Basic insulation
┊ Functional insulation

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

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

Example

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

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

- The separate potential area of the device module's bus terminal is "functionally insulated".
- The separate potential area of the mains terminal has a "reinforced insulation".

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

4.3 Protocol data

Area	Values
Process data words (PCD)	1 ... 32 words (16 bits/word)
Cyclic parameter data channel (DP-V0)	4 words
Acyclic parameter data channel (DP-V1)	Max. 240 bytes
Safety data	4 words
PROFIBUS user data length	1 ... 32 words (process data) + 4 words (parameter data or safety data)

4.4 Communication time

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

The communication times in a PROFIBUS network depend on ...

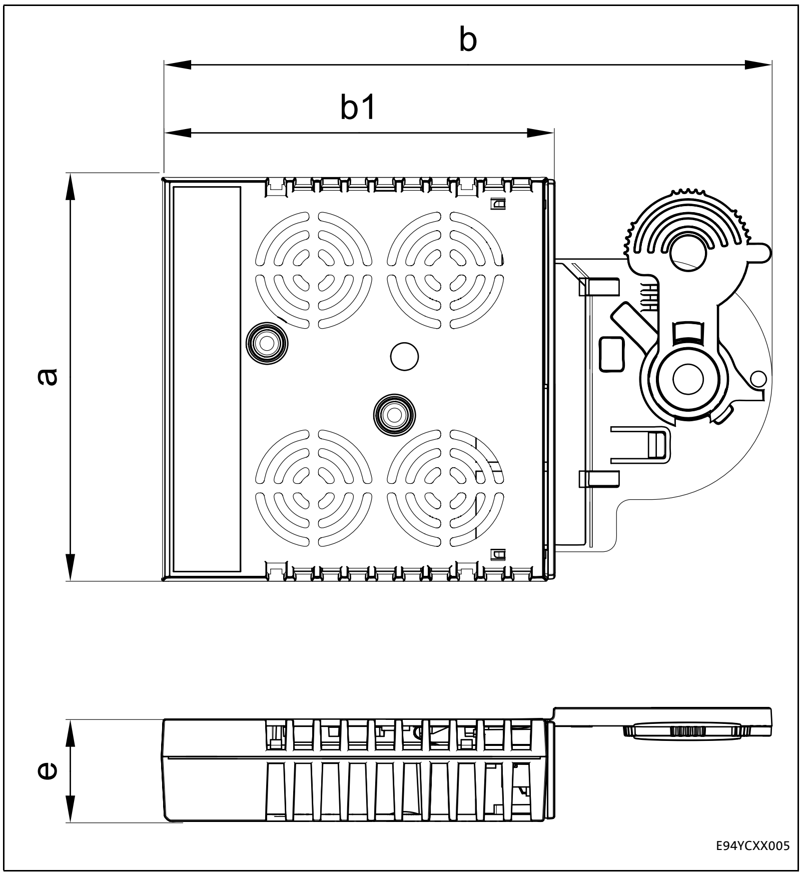
- the processing time in the controller;
- the transmission delay time (baud rate / telegram length);
- the nesting depth of the network.

Processing time in the controller

Data	Processing time
Process data	Approx. 4 ms update cycle + 0 ... 1 ms processing time in the module + 1 ... x ms application task runtime of the technology application used (tolerance)
Parameter data	Approx. 30 ms + 20 ms tolerance (typical) For some codes, the processing time may be longer (see software manual/ online help for the Servo Drive 9400).

There are no interdependencies between parameter data and process data.

4.5 Dimensions



a 89 mm
b 134 mm
b1 87 mm
e 23 mm

[4-2] Dimensions

5 Installation

Stop!

Electrostatic discharge

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

Possible consequences:
- The communication module is defective.
- Communication via the fieldbus is not possible or faulty.

Protective measures

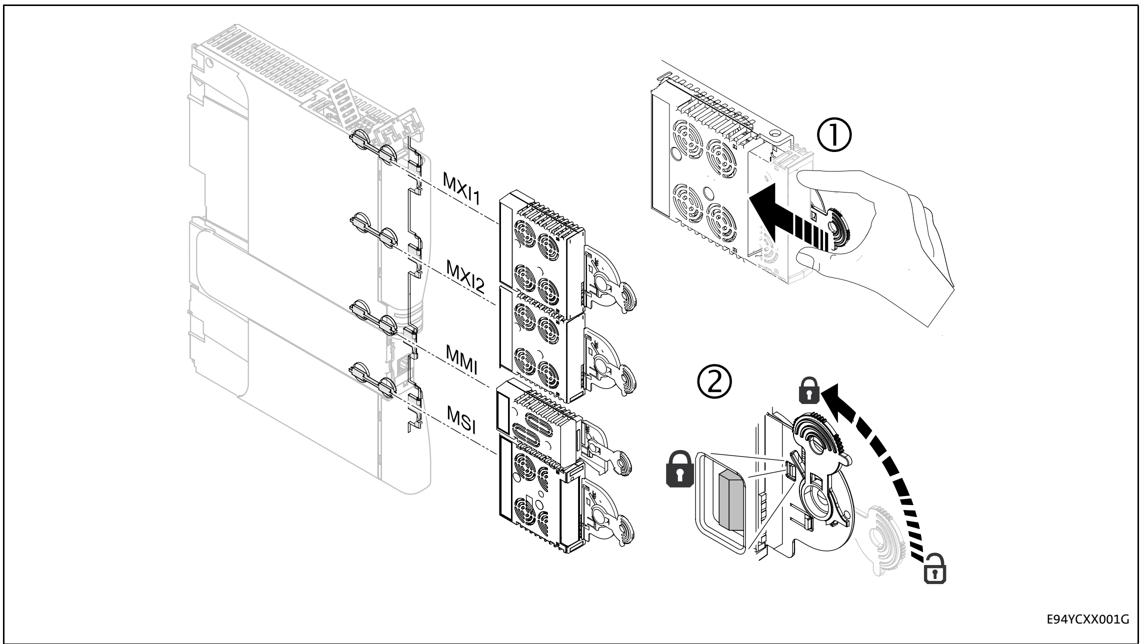
Discharge electrostatic charges before touching the module.

5.1 Mechanical installation

Note!

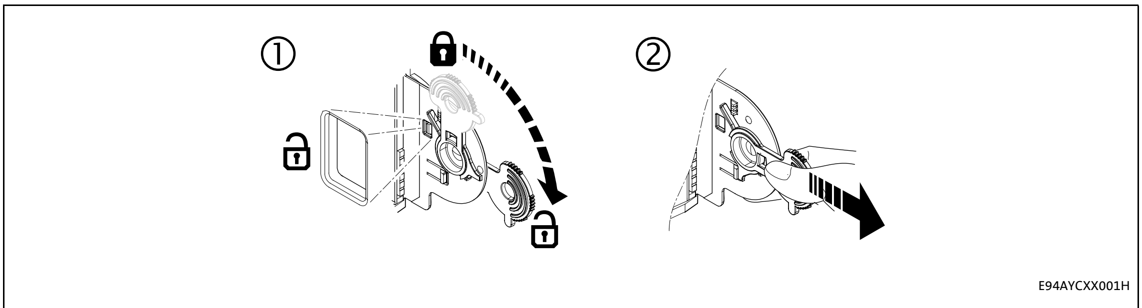
A safety bus system (PROFIsafe) can only be operated via the upper module slot (MXI1) of the Servo Drive 9400.

5.1.1 Mounting



[5-1] Mounting

5.1.2 Dismounting



[5-2] Dismounting

5.2 Electrical installation

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

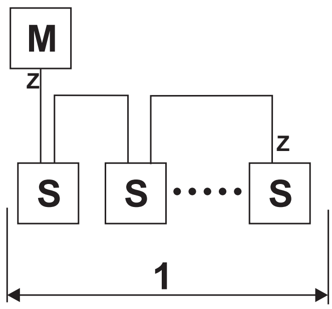
Observe the notes and wiring instructions given in the documentation.

5.2.1 Network topology

The following examples show two simple RS485 networks.

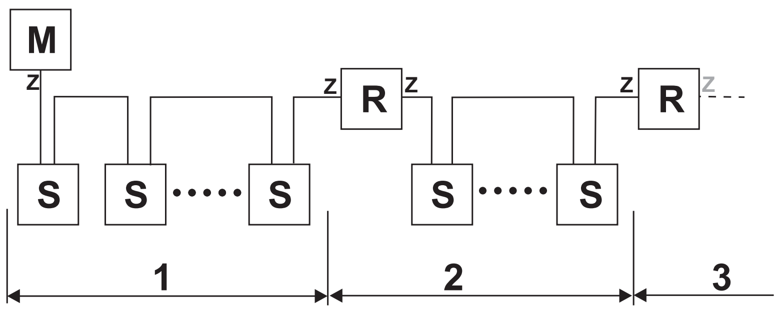
Every segment of the network must be terminated at both ends. The bus terminators of the PROFIBUS are marked with a "Z" in the below examples.

In the case of an RS485 network of only one segment, the PROFIBUS master (M) with the integrated bus terminator starts the segment while the bus terminating resistor in the connector of the last PROFIBUS station (S) must be activated.



[5-3] RS485 network with one segment

An RS485 network consisting of several segments contains repeaters (R) for connecting the segments. The repeaters are provided with integrated bus terminating resistors.



[5-4] RS485 network with a repeater

If no repeater is to be used at the end of the segment, the bus terminating resistor must be activated in the connector of the last device. The bus termination is supplied by the station itself.

If the communication module is supplied externally, the bus terminator can be supplied independently of the standard device supply.

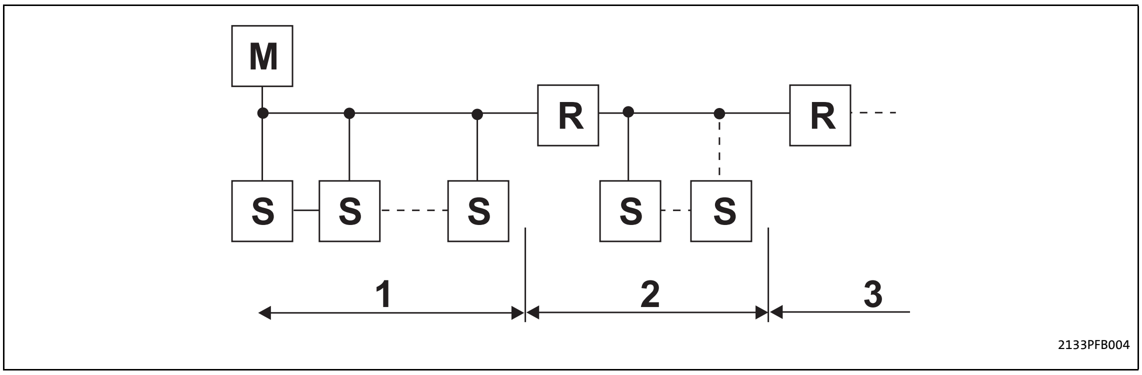
▶ External voltage supply (📖 28)

Note!

The bus terminator must always be supplied. Otherwise, the bus can get unstable.

▶ Activating the bus terminating resistor (📖 25)

Number of stations



[5-5] Number of stations

Segment	Master (M)	Slave (S)	Repeater (R)
1	1	31	-
	2	30	-
2	-	30	1
3	-	30	1

Tip!

Repeaters do not have a station address. When calculating the maximum number of stations, they reduce the number of stations by 1 on each side of the segment.

Repeaters can be used to build up line and tree topologies. The maximum total bus system expansion depends on ...

- the baud rate used;
- the number of repeaters used.

5.2.2 Activating the bus terminating resistor

The PROFIBUS must be terminated by a bus terminating resistor at the first and last physical bus station.

The bus terminating resistor in the bus connector of the bus cable is activated by means of a switch.

PROFIBUS cables with integrated bus terminating resistor are offered by several cable manufacturers.

Note!

If you want to disconnect individual bus stations, ensure that the bus terminators at the cable ends remain active.

Please observe that the bus termination is not active any longer if ...

- the bus connector has been disconnected;
- the voltage supply of the Servo Drive 9400 has been switched off;
- the External voltage supply (📖 28) of the communication module has been switched off.

5.2.3 Bus cable specification

Note!

Only use cables which meet the listed specifications of the PROFIBUS user organisation.

Area	Values
Cable resistance	135 ... 165 Ω/km, (f = 3 ... 20 MHz)
Capacitance per unit length	≤ 30 nF/km
Loop resistance	< 110 Ω/km
Core diameter	> 0.64 mm
Core cross-section	> 0.34 mm^2
Cores	Twisted in pairs, insulated and shielded

Bus cable length

The length of the bus cable depends on the baud rate and cable type used. The data in the following table applies to PROFIBUS cables of "FC-Standard Cable" cable type .

Baud rate	Length
9.6 ... 93.75 kbps	1200 m
187.5 kbps	1000 m
500 kbps	400 m
1500 kbps	200 m
3000 ... 12000 kbps	100 m

Note!

The baud rate depending of the data volume, cycle time and number of stations should only be selected as high as required for the application.

Tip!

We recommend taking the use of optical fibres into consideration for high baud rates.

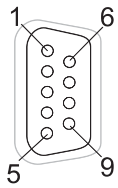
Advantages of optical fibres:

- External electromagnetic interferences have no effect on the transmission path.
- Bus lengths of several kilometres are also possible with higher baud rates.
- The bus length is ...
  - independent of the baud rate;
  - dependent on the optical fibre used.

5.2.4 PROFIBUS connection

The 9-pin Sub-D socket **X201** serves to connect the communication module to the bus system.

Assignment of the 9-pin Sub-D socket X201

View	Pin	Assignment	Description
	1	Not assigned	-
	2	Not assigned	-
	3	RxD/TxD-P	Data line B (received data/transmitted data, plus)
	4	RTS	Request To Send (received data/transmitted data, no differential signal)
	5	M5V2	Data ground (ground to 5 V)
	6	P5V2	5 V DC / 30 mA (bus termination)
	7	Not assigned	-
	8	RxD/TxD-N	Data line A (received data/transmitted data, minus)
	9	Not assigned	-

5.2.5 External voltage supply

Note!

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

External voltage supply of the communication module is necessary if the bus communication is to continue when the supply of the standard device fails.

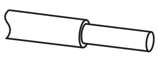
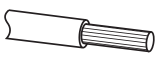
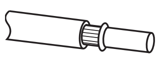
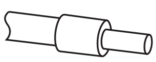
It is not possible to access the parameters of a standard device disconnected from the mains.

If required, feed the communication module with a separate supply voltage via the 2-pin plug connector.

Assignment of the 2-pin plug connector (X200)

Designation	Explanation
+	V = 24 V DC (20.4 V - 0 % ... 28.8 V + 0 %) I = 130 mA
-	Reference potential for external voltage supply

Terminal data

<table>
<tr><th>Area</th><th colspan="2">Values</th></tr>
<tr><td>Electrical connection</td><td colspan="2">Plug connector with screw connection</td></tr>
<tr><td rowspan="7">Possible connections</td><td colspan="2">Rigid:</td></tr>
<tr><td></td><td>0.2 ... 1.5 mm^2 (AWG 24 ... 16)</td></tr>
<tr><td colspan="2">Flexible:</td></tr>
<tr><td></td><td>Without wire end ferrule
0.2 ... 1.5 mm^2 (AWG 24 ... 16)</td></tr>
<tr><td></td><td>With wire end ferrule, without plastic sleeve
0.2 ... 1.5 mm^2 (AWG 24 ... 16)</td></tr>
<tr><td></td><td>With wire end ferrule, with plastic sleeve
0.2 ... 1.5 mm^2 (AWG 24 ... 16)</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td>Tightening torque</td><td colspan="2">0.5 ... 0.6 Nm (4.4 ... 5.3 lb-in)</td></tr>
<tr><td>Stripping length</td><td colspan="2">6 mm</td></tr>
</table>

6 Commissioning

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

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

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

Note!

When parameterising the communication module, please observe that the code number depends on the slot of the Servo Drive 9400 in which the communication module has been inserted.

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

- C**13**nnn for slot MXI1
  - Parameters of the communication module for slot MXI1 (93)
- C**14**nnn for slot MXI2
  - Parameters of the communication module for slot MXI2 (102)

You also have to set the Communication-relevant parameters of the standard device (91).

6.1 Before initial switch-on

Stop!

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

- the entire wiring for completeness, short circuit and earth fault:
- whether the bus system is terminated through a bus terminating resistor at the first and last physical bus station.
  - Activating the bus terminating resistor (25)

6.2 Configuration of the host (master)

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

Configuration for the host system (master) and the DP-V0 parameter data channel

For configuring the PROFIBUS you must read the device description file of the communication module into the master.

The device description file for the E94AYCPM communication module (PROFIBUS) can be found in the download area at:

www.Lenze.com

The following language variants of the device description file can be used:

- LENZ07A8.GSD (source file, English)
- LENZ07A8.GSG (German)
- LENZ07A8.GSE (English)

Defining the user data length

The user data length is defined during the initialisation phase of the master.

The Servo Drives 9400 support the configuration of a maximum of 32 process data words (max. 64 bytes). The optional activation of the cyclic parameter data channel additionally occupies 4 process data words (8 bytes). The transfer of PROFIsafe messages, too, occupies 4 additional process data words.

The user data lengths for process input data and process output data are the same.

Description of the device data base file

Selection text	Parameter data with consistency	Process data		Assigned IO memory
		with consistency	without consistency	
DRIVECOM-PAR (cons) + PCD (**n**W cons)	Yes	**n** words	-	**4** + **n** words
PCD (**n**W cons)	-	**n** words	-	**n** words
PCD (**n**W)	-	-	**4** words	**4** words
Safety (4W)	-	**4** words	-	**4** words
n = 1 ... 32 process data words				

Example of selecting the device data base file

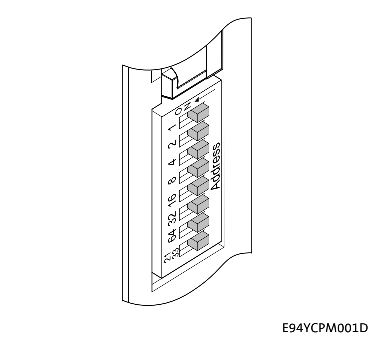
```
DRIVECOM-PAR (cons) + PCD (8W cons)
```

- "Drivecom-PAR (cons)" = DP-V0 parameter data channel (4 words)
- "PCD (8W cons)" = 8 process data words

Tip!

A detailed description of consistency is given in the chapter "Consistent parameter data" (📖 76).

6.3 Possible DIP switch settings



[6-1] DIP switch

The front-panel DIP switches can be used to set:

- The station address (switches 1 ... 64)
- Compatibility with the communication module EMF2133IB (switch 2133)

Lenze setting: all switches in OFF position

6.3.1 Setting the station address

The station addresses must differ from each other if several networked PROFIBUS stations are used.

The station address can be set via the DIP switches **1 ... 64** or via the »Engineer« (code C13899 / C14899).

	Setting the station address via ...	
	DIP switch	**C13899 / 14899**
Condition	At least one switch **1 ... 64 = ON**	• Switches **1 ... 64 = OFF** • Switch **1 ... 64 = ON** (invalid value "127")

The housing labelling indicates the valencies of the individual DIP switches for setting the station 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
Station address	**= sum of the valencies = 16 + 4 + 2 + 1 = 23** ▶ DIP switch positions for setting the station address (📖 116)						

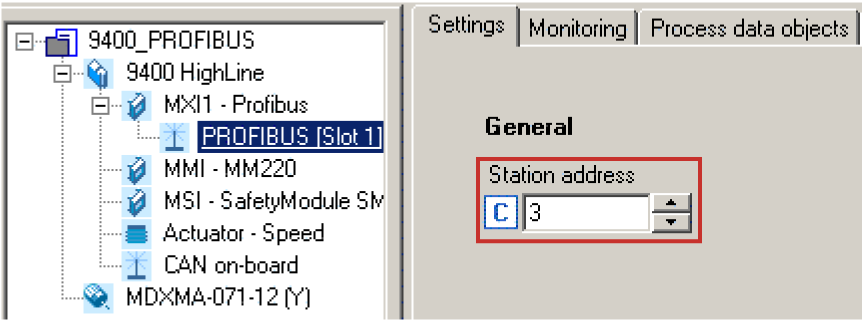
- Valid address range: 1 ... 126 (max. 126 slave stations)
- C13920 / C14920: Display of the current address setting of the switches
- C13864 / C14864: Display of the station address active on the PROFIBUS

Note!

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

Setting the station address via the »Engineer«

In the »Engineer«, the station address can be set via the **Settings** tab.



Invalid addresses are displayed in red in the **Station address** input field (code C13899 / C14899).

Save the changed settings via the device command **C00002 = 11** (save start parameters).

6.3.2 Establishing EMF2133IB compatibility

Establishing EMF2133IB compatibility is necessary if you need to communicate with systems using an EMF2133IB PROFIBUS communication module.

Software compatibility can be set via DIP switch **2133**.

Switch	Switch position	Function
2133	ON	Establishing EMF2133IB compatibility
	OFF	No EMF2133IB compatibility

Note!

- In the compatibility mode "EMF2133IB", only the DRIVECOM parameter data channel (DP-V0) and the Lenze device control can be used.
- Switch off the voltage supply of the communication module and then on again in order to activate changed settings.

Communication manual for the EMF2133IB module

Here you can find information on the configuration of the EMF2133IB communication module and the identification of the module on the bus.

6.4 First switch-on

Documentation for the standard device

Observe the safety instructions and residual hazards stated.

Note!

Establishing communication

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

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

Protection against uncontrolled restart

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

In the Lenze setting of the Servo Drives 9400, the restart protection is activated.

Via **C00142** ("Auto-restart after mains connection"), you can configure 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 in a controlled mode by explicitly enabling the controller: LOW-HIGH edge at digital input X5/RFR.

C00142 = "1: Enabled"

- An uncontrolled restart of the drive is possible.

6.5 Going online with »Engineer« via TCI

Via Tool Calling Interfaces (TCI) you can connect to a TCI-capable integrated development environment and parameterise and diagnose your field devices without having to exit the integrated development environment.

You cannot set the TCI communication path directly in the »Engineer«. The selection is carried out by the »STEP7« Siemens software.

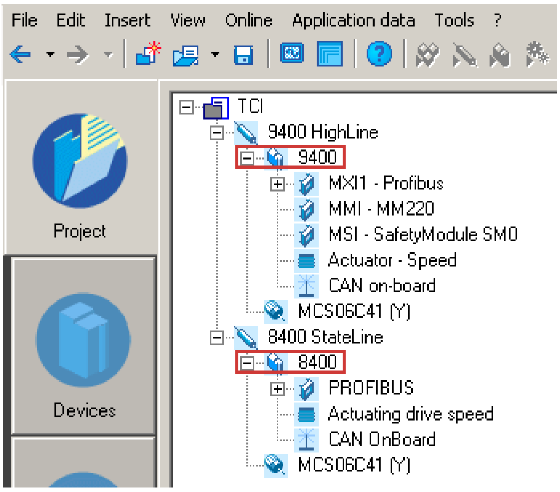
The TCI function requires a PN/DP-CPU. Information on the Siemens PLC types that are equipped with the TCI function is provided via the Siemens Support at:

http://support.automation.siemens.com

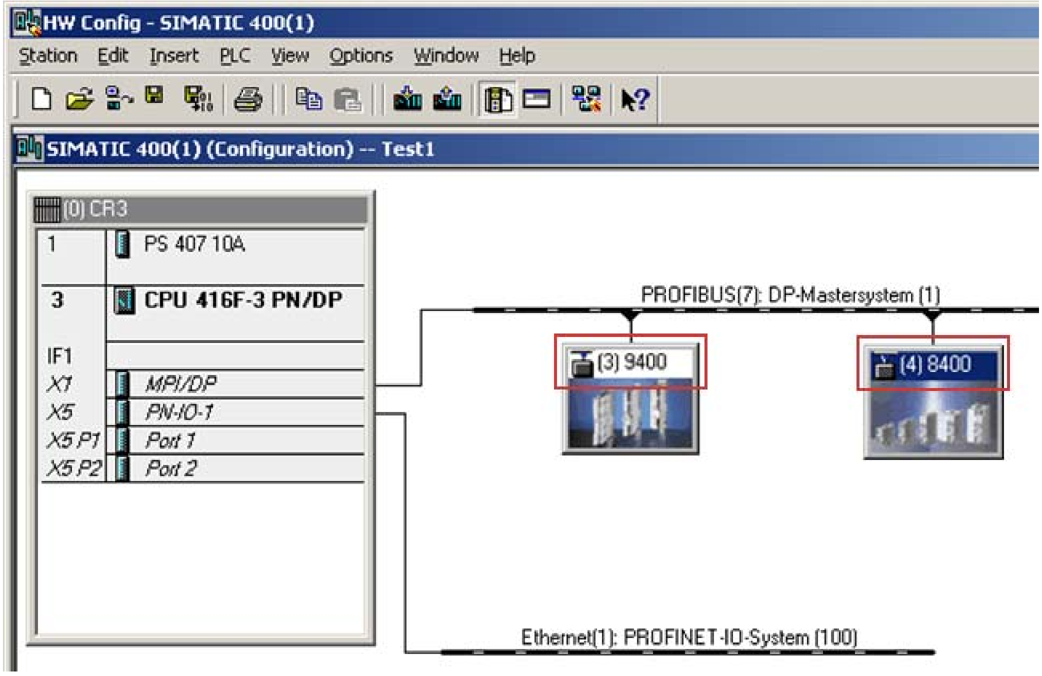
How to configure TCI communication:

1. Alllocate names for the individual axes in the »Engineer« project.

   In our case, "9400" was allocated for 9400 HighLine, and "8400" for 8400 StateLine:



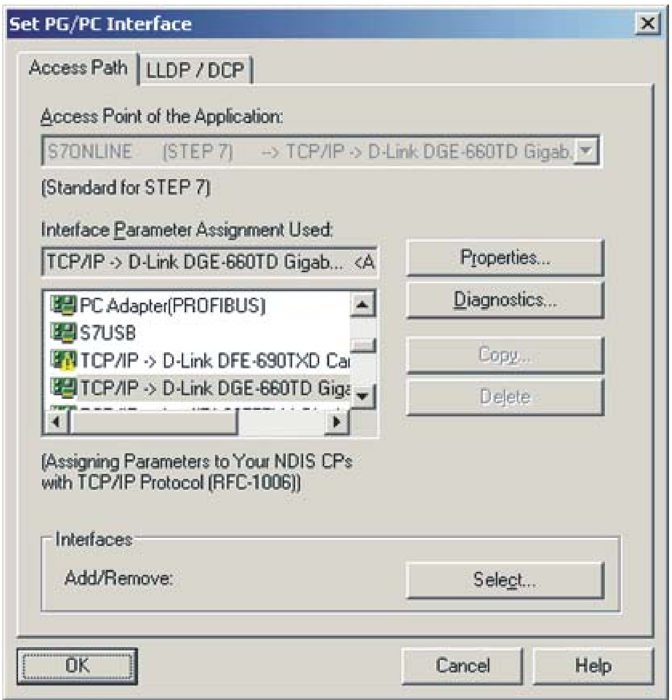
2. In »STEP7« in the »HW Config« ...
   - create the Lenze PROFIBUS stations with the corresponding station addresses and
   - create a PROFIBUS network.



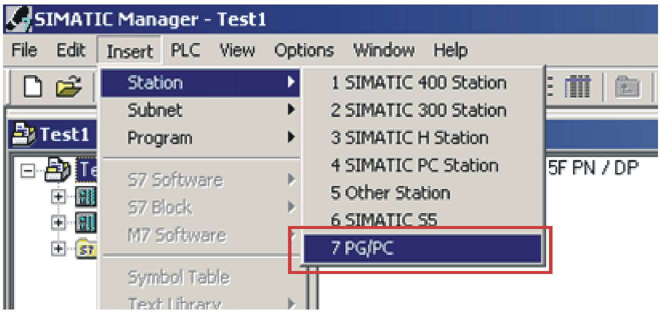
Here a Servo Drive 9400 (address 3) and an Inverter Drive 8400 (address 4) are operated on the PROFIBUS.

- The names of the PROFIBUS slaves in the »HW Config« must be identical to those of the corresponding Lenze axes in the »Engineer« (here "9400" and "8400").
- The selection of the process data configuration has no impact on TCI communication.

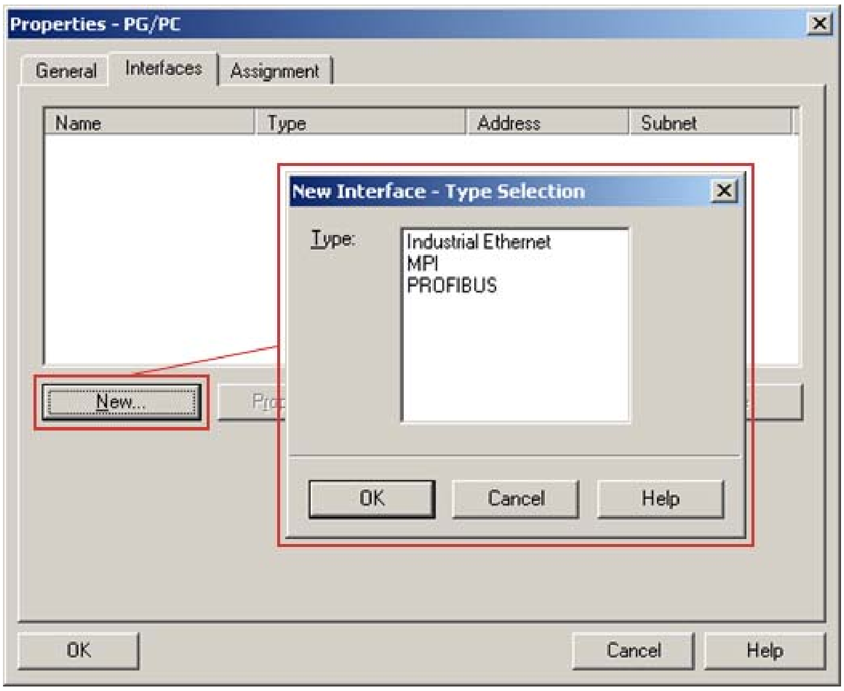
3. Establish an Ethernet connection to the PROFIBUS CPU.



4. Load the »STEP7« project to the CPU.
5. Use the menu command **Insert → Station → 7 PG/PC** to integrate a PG/PC station into the »STEP7« project.

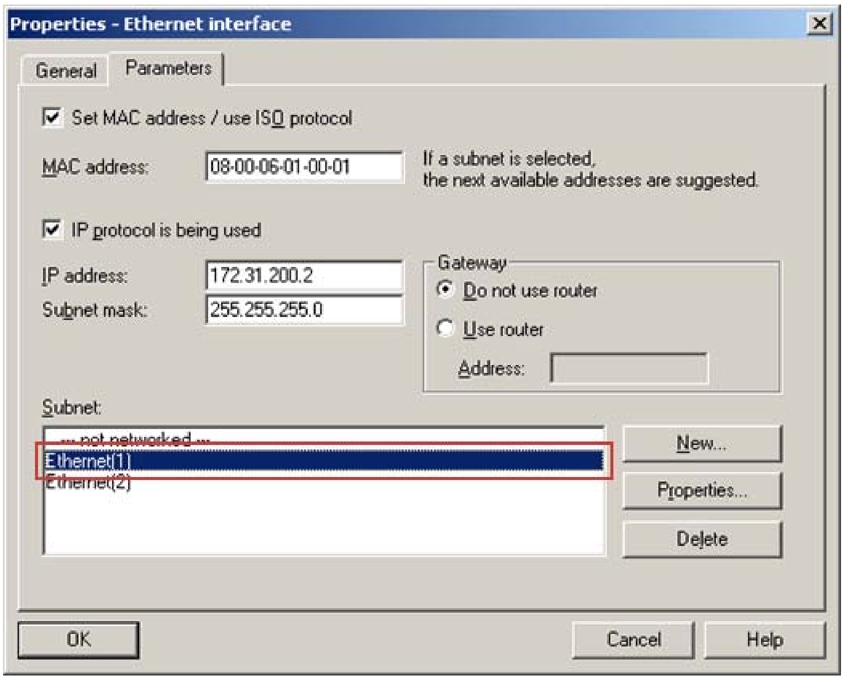


6. By double-clicking the PG/PC station inserted, open its "Properties" dialog.
7. Under the **Interfaces** tab, select a new Ethernet interface and confirm with **OK**.



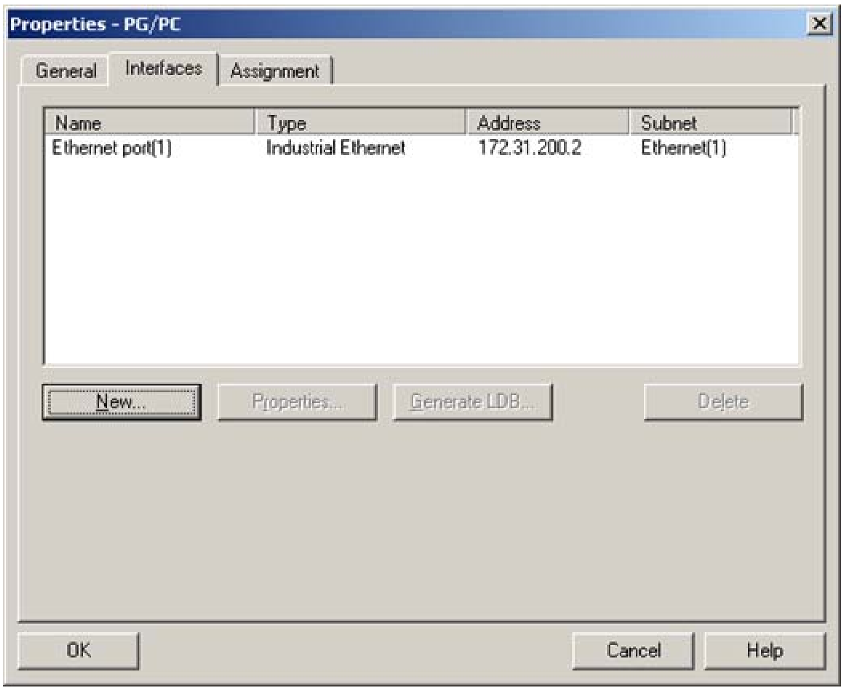
8. Select the connection which you are using to go online with »STEP7« (the same Ethernet connection that has been configured in the »HW Config«).

   In our case this is the Ethernet(1) connection:

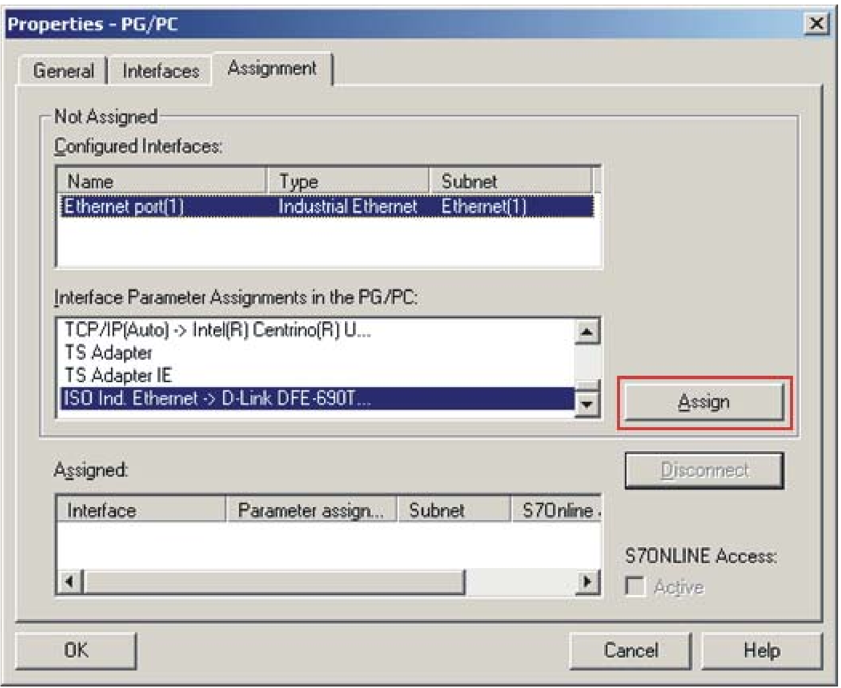


9. Confirm the selection with **OK**.

   The connection has been accepted.

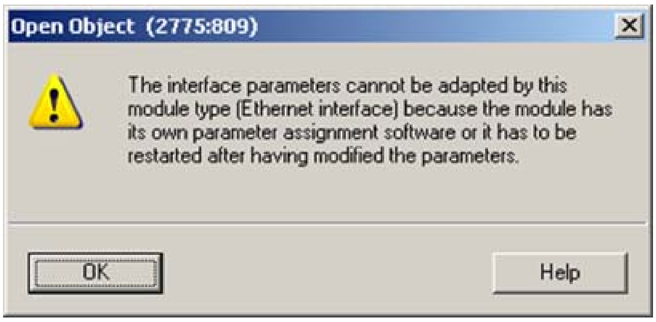


10. Select the actual PG/PC connection under the **Assignment** tab.

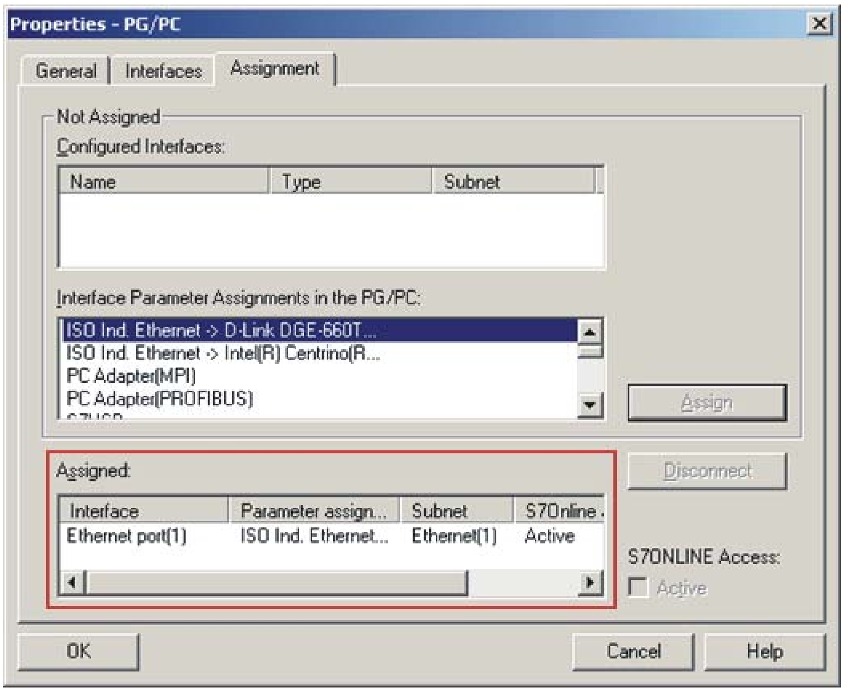


The connection highlighted is assigned by means of the **Assign** button.

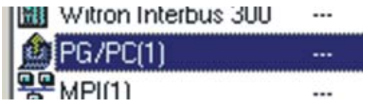
11. Confirm the following message with **OK**.



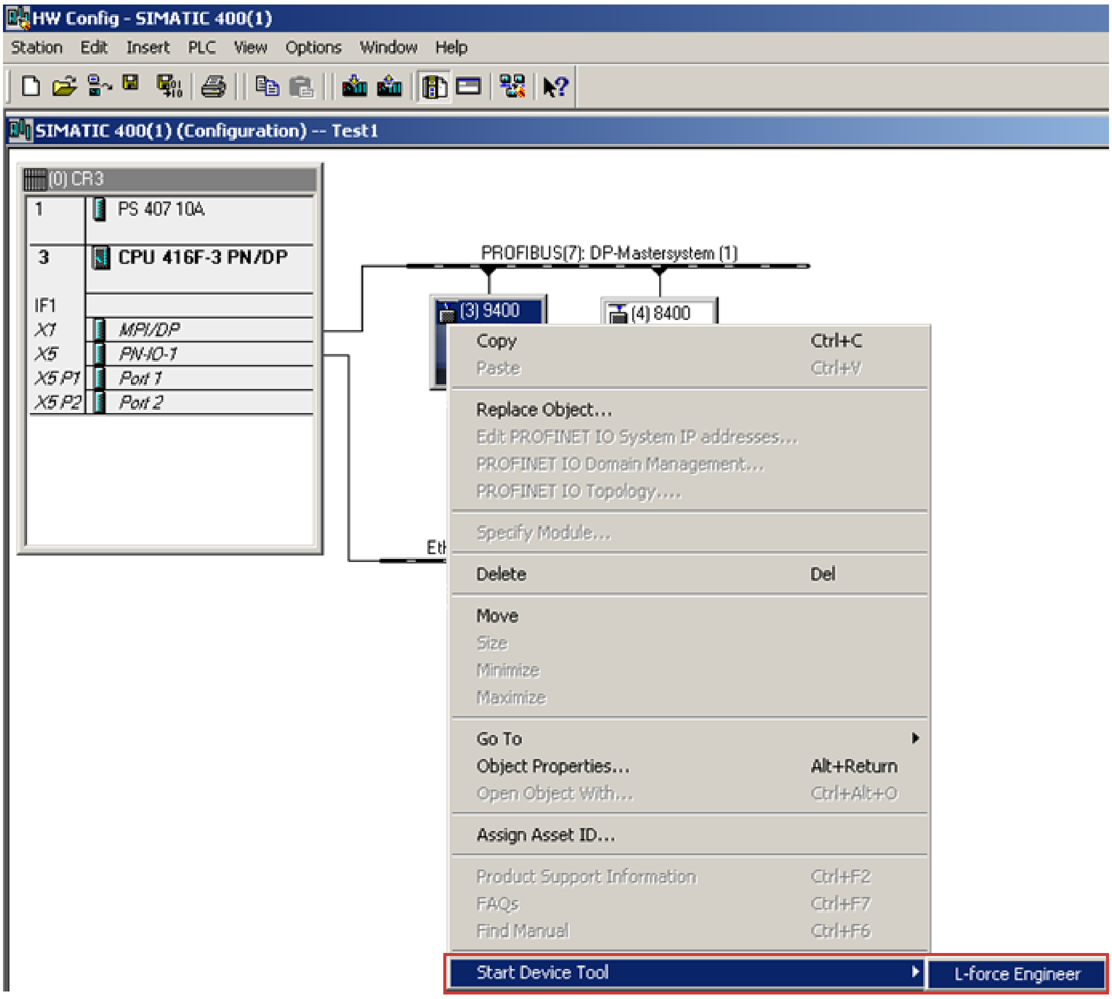
12. After the assignment, the connection appears in the "Assigned" display area. Complete the dialog with **OK**.



13. In the »STEP7« project, the PG/PC station is marked with a yellow arrow. (The connection selected is active.)



14. Start the transfer of the TCI communication parameters in the »HW Config« using the right mouse button and the menu command **Start device tool → L-force Engineer**.

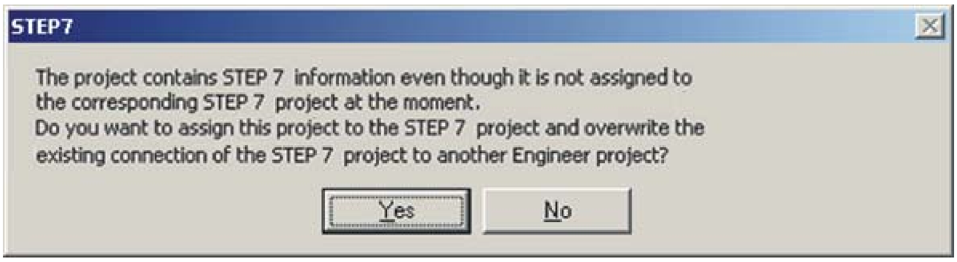


15. If the »Engineer« has already been started with the applicable project, the following message will appear:

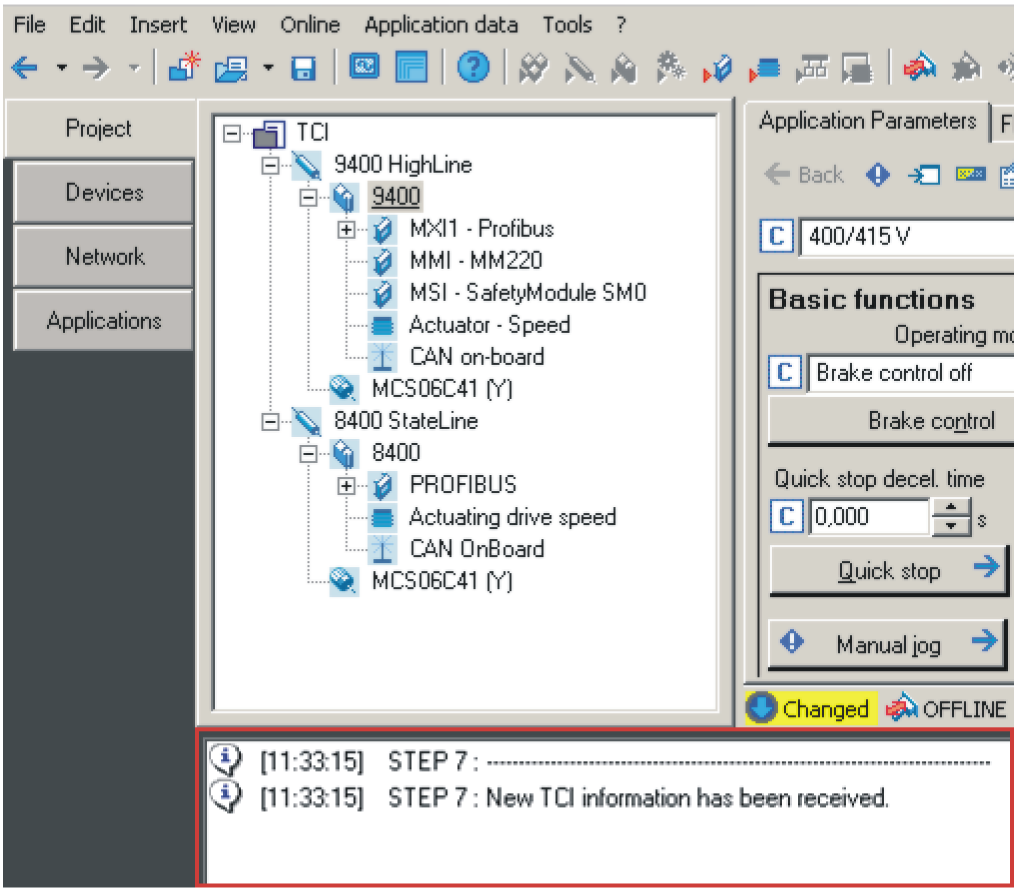
- The message says that the »Engineer« project is not set to a TCI communication path and provides information on whether this action is to be executed now.
- If you confirm the message with **Yes**, the applicable TCI communication parameter settings of the »STEP7« project are transferred to the »Engineer«.

If the »Engineer« has not already been started, it is started automatically now and you have to open the applicable project.

If the project selected has not been set to a TCI communication path yet, this can now be executed with **Yes**:

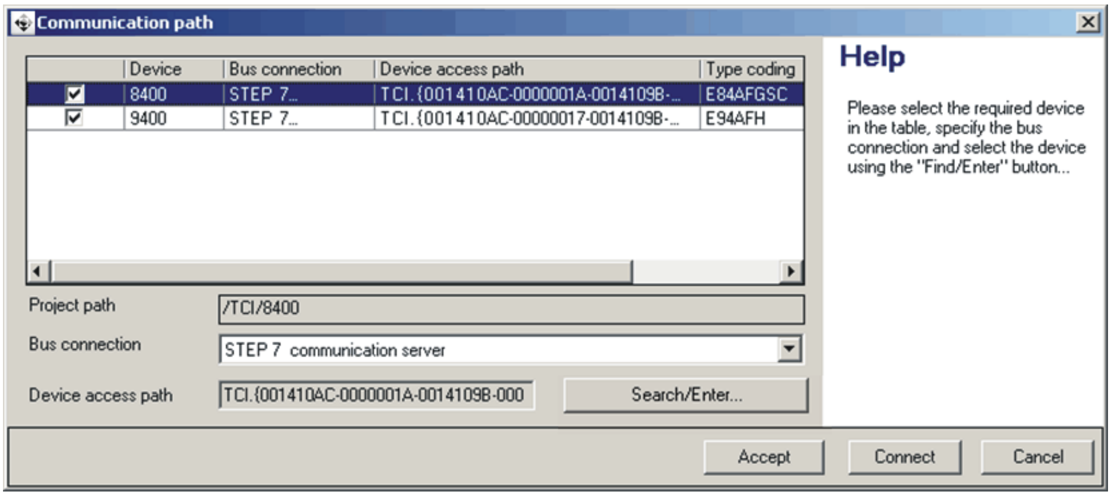


16. The transfer of the TCI communication parameters is documented in the »Engineer« message window.



Here the communication settings have been carried out successfully. The individual PROFIBUS addresses in the respective codes have been adapted to the »STEP7« project.

17. If you now call the "Go online" function of the »Engineer«, the TCI communication settings are displayed as follows:



- "STEP7 Communication Server" appears as bus connection.
- The device access path contains a very long string.
- Use the **Connect** button to establish an online connection.
- By means of the **Search/Enter** button, you can update the TCI communication parameters.

7 Data transfer

The PROFIBUS master and controller communicate through the exchange of data telegrams via PROFIBUS. The user data area of the data telegram contains parameter data or process data. In the controller, different communication channels are assigned to the parameter data and process data.

Communication channels

The process data channel serves to transfer process data.

- The process data serve to control the drive controller.
- The host (master) can directly access the process data. In the PLC, for instance, the data are directly saved to the I/O area.
- Process data are not saved in the controller.
- Process data are transferred cyclically between the host system and the controllers (permanent exchange of current input / output data).
- Process data are, for instance, setpoints, actual values, control words and status words.
- The Servo Drives 9400 can exchange a maximum of 32 process data words (16 bits/word) per direction.

Note!

Observe the direction of the information flow!

Process input data (Rx data):
- Process data from the controller (slave) to the master

Process output data (Tx data):
- Process data from the master to the controller (slave)

The parameter data channel serves to transfer parameter data.

- The parameter data channel provides access to all Lenze codes.
- In general, the parameter data transfer is not time-critical.
- Parameter data are, for instance, operating parameters, diagnostic information, and motor data.
- Parameter data changes must be saved via code **C00002** of the Servo Drives 9400.

8 Process data transfer

PDO mapping

The Servo Drives 9400 HighLine enable the individual mapping of process data. For this purpose, the »Engineer« is provided with a port configurator.

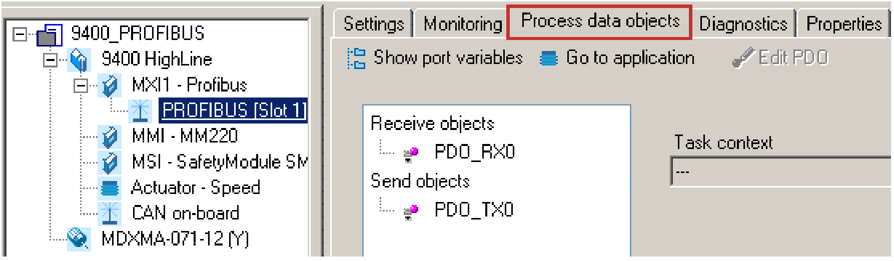
Below you can find a description of the steps required to implement a process data communication with a higher-level control, in which a control word/status word and a 32-bit setpoint/actual value are exchanged.

Note!

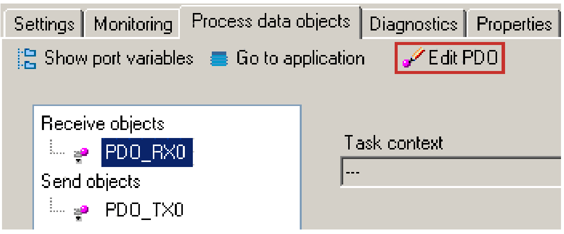
The »Engineer« screenshots shown on the following pages are only examples for the setting sequence and the resulting screens.

PDO mapping with the »Engineer«

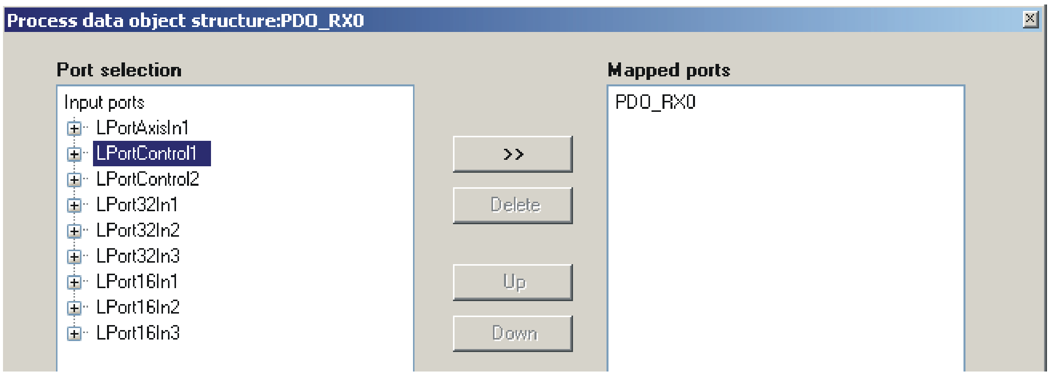
1. You can carry out the mapping of the process data in the »Engineer« on the **Process data objects** tab of the respective fieldbus communication module:



2. Select the receive object **PDO_RX0**:



3. Click the **Edit PDO** button. The **Process data object structure:PDO_RX0** selection window opens:

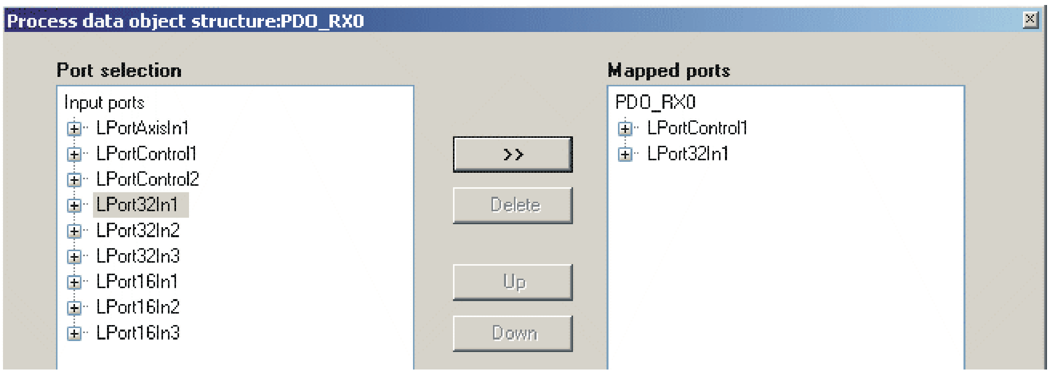


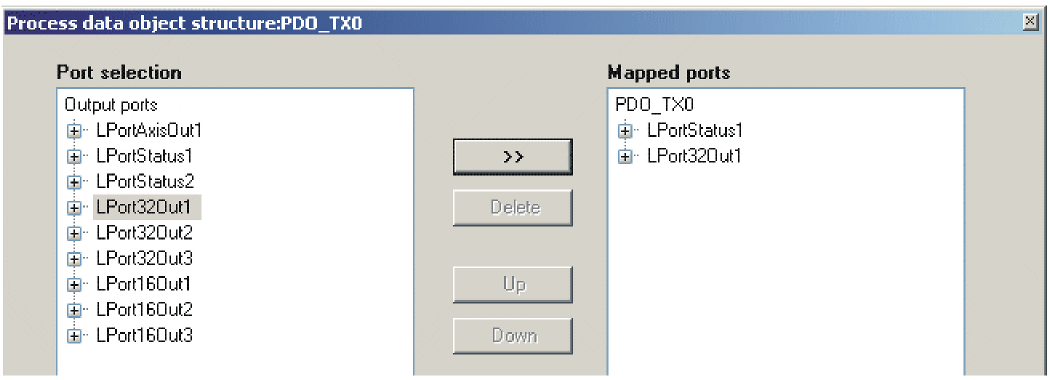
Here you can map the individual ports from the **Port selection** list into the receive PDO "PDO_RX0" by clicking the **>>** button. With the **Up** and **Down** buttons, you can change the order of the ports within the PDO.

Note

The port mapping is not a configuration, which can be carried out online for the Servo Drive 9400 HighLine. For this purpose, the »Engineer« project must be updated and then the application must be downloaded.

In the following example, the ports "LPortControl1" and "Lport32In1" have been mapped into the receive PDO "PDO_RX0" and the ports "LPortStatus1" and "LPort32Out1" have been mapped into the transmit PDO "PDO_TX0":





4. Now link the mapped ports to the application signals in the selected technology application.
   - If the »FB Editor« is not activated, you can use the multiplexer codes (from code C03000 onwards) for this purpose.
   - If the »FB Editor« is activated, the multiplexer codes are no longer available. In this case, you have to do the linking directly in the »FB Editor«.

9 Parameter data transfer

The E94AYCPM communication module supports the cyclic and acyclic transmission of parameter data:

- Cyclic DP-V0 parameter data are based on the DRIVECOM profile.
  If the parameter data channel is active according to DP-V0, it is assigned an additional 4 words of input / output data.
- Acyclic DP-V1 parameter data are based on the PROFIdrive profile.

9.1 Addressing of the parameter data

The parameter data are addressed via codes which can be found in this documentation and in the corresponding documentation of your controller.

▸ Parameter reference (📖 91)

Addressing of Lenze parameters

In the case of the DP-V0 parameter data channel, the parameters of a device are not addressed directly via Lenze code numbers, but via indices (bytes 3 + 4) and subindices (byte 2).

The conversion is made via an offset (24575 / 0x5FFF):

- PROFIBUS-DP $index_{dec}$ = 24575 - Lenze code number
- PROFIBUS-DP $index_{hex}$ = 0x5FFF - Lenze code $number_{hex}$

Example of C00105 (quick stop deceleration time):

- PROFIBUS-DP $index_{dec}$ = 24575 - 105 = 24470
- PROFIBUS-DP $index_{hex}$ = 0x5FFF - 0x69 = 0x5F96

The parameter values are entered into the user data (bytes 5 to 8) of the telegram.

9.2 DRIVECOM parameter data channel (DP-V0)

The DRIVECOM parameter data channel (DP-V0) ...

- enables parameter setting and diagnosing of the controller;
- provides access to all Lenze parameters (codes);
- additionally occupies 4 words (16 bits/word) of the input and output data words in the master;
- is identical for both transmission directions.

9.2.1 Telegram structure (overview)

The telegram of the parameter data channel consists of a total of 8 bytes:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

The individual bytes are described in detail in the following subchapters.

9.2.2 Byte 1: Service

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

Request and response control for the parameter data channel

7	6	5	4	3	2	1	0

[9-1] Method of reading for bits 0 ... 7

Bit 0 ... 2: Request	
Read/write request from the master to the controller	
000	No request
001	Read request ▸ Reading parameter data from the controller (📖 50)
010	Write request (write data to the controller) ▸ Writing parameter data to the controller (📖 50)
100	Data transfer abort by the master ▸ Data transfer abort by the master (📖 51)

Bit 3
Reserved

Bit 4/5: Data length	
Data length ≤ 4 bytes in the telegram bytes 5 ... 8 (data 1 ... 4 / error 1 ... 4)	
00	1 byte
01	2 bytes
10	3 bytes
11	4 bytes

Bit 6: Handshake
Indicates a new request. • The state of this (toggle) bit is changed by the master for every new request. • The controller copies the bit into its response telegram.

Bit 7: Status	
Status information from the controller to the master when sending the request confirmation. This status bit informs the master whether the request has been carried out without errors.	
0	Request completed without errors. Bytes 5 ... 8 contain a parameter value (Data 1 ... 4).
1	Request not completed because of an error. • The set status bit indicates that the telegram is an "error telegram". • Bytes 5 ... 8 contain an error code (Error 1 ... 4). ▸ Error codes (📖 54)

9.2.2.1 Reading parameter data from the controller

General procedure:

1. Define the user data area of the controller, i.e. define the location of the DP user data in the host (observe manufacturer-specific information).
2. Enter the address of the required parameter in the "Index" and "Subindex" fields (DP output data).
3. Request in the service byte = read request.
   The handshake bit in the service byte must be changed (DP output data).
4. Check whether the handshake bit in the service byte is the same for the DP input data and the DP output data.
   - If the handshake bit is the same, the response has been received.
   - It is useful to implement a time monitoring tool.
5. Check whether the status bit in the service byte is set:
   - Status bit isnot set: The "Data/Error" field contains the required Parameter value (data) (🕮 53).
   - Status bit is set: The read request has not been executed correctly. The "Data/Error" field contains the Error codes (🕮 54).

9.2.2.2 Writing parameter data to the controller

General procedure:

1. Define the user data area of the controller, i.e. define the location of the DP user data in the host (observe manufacturer-specific information).
2. Enter the address of the required parameter in the "Index" and "Subindex" fields (DP output data).
3. Enter the parameter value in the "Data/Error" field.
4. Request in the service byte = write request.
   The handshake bit in the service byte must be changed (DP output data).
5. Check whether the handshake bit in the service byte is the same for the DP input data and the DP output data.
   - If the handshake bit is the same, the response has been received.
   - It is useful to implement a time monitoring tool.
6. Check whether the status bit in the service byte is set:
   - Status bit is not set: The write request has been executed correctly.
   - Status bit is set: The write request has not been executed correctly. The "Data/Error" field contains the Error codes (🕮 54).

9.2.2.3 Abort of data transfer by the controller

The error telegram is used to abort the transfer.

- The error telegram is marked by a set status bit in the service byte.
- The telegram can either be the response to an "Initiate Read/Write Service" or to a "Read/Write Segment Service".

Controller response in the event of an error:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1
1t110000	SIDX	IDXH	IDXL	Error Class	Error code	Additional Code High	Additional Code Low

9.2.2.4 Data transfer abort by the master

The master can use this error telegram to abort a running segment transmission.

- The error telegram is marked by a set status bit in the service byte.
- The service byte also contains the request code "4" (100_{bin}).
- Bit 4 and bit 5 in the service byte (data length) are without meaning.
- Additional information (subindex, index, error information) is not transmitted.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
1txx0100	0	0	0	0	0	0	0

Controller response in the case of correct execution:

The controller confirms the error telegram of the master by also sending an error telegram.

- The error telegram is marked by a set status bit in the service byte.
- In the case of correct execution, the telegram contains the error information "0x00000000" in bytes 5 ... 8.
- Additional information (subindex, index) is not transmitted.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	SIDX	IDXH	IDXL	Error Class	Error code	Additional Code High	Additional Code Low
1t110000	0	0	0	0	0	0	0

9.2.3 Byte 2: Subindex

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	**Subindex**	Index High byte	Index Low byte	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

Additional addressing via the subindex is required for those codes of the Servo Drives 9400 that contain a subcode (see code table).

9.2.4 Bytes 3 + 4: Index

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	**Index High byte**	**Index Low byte**	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

The parameter (Lenze code) is selected via these two bytes according to the formula:

Index = 24575 - Lenze code number

(See also "Addressing of Lenze parameters" (📖 47))

Example:

The parameter C00105 (quick stop (QSP) deceleration time) is to be addressed:

- Index = 24575 - 105 = 24470 = 0x5F96
- The entries in bytes 3 + 4 for this example would be:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	**0x5F**	**0x96**	Data 4 / Error 4	Data 3 / Error 3	Data 2 / Error 2	Data 1 / Error 1

9.2.5 Bytes 5 ... 8: Parameter value / error information

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	**Data 4 / Error 4**	**Data 3 / Error 3**	**Data 2 / Error 2**	**Data 1 / Error 1**

The state of status bit 7 in the service byte determines the meaning of this data field:

Status bit	Meaning of bytes 5 ... 8
0	Bytes 5 ... 8 contain a parameter value (Data 1 ... 4).
1	Bytes 5 ... 8 contain an error code (Error 1 ... 4). ▸ Error codes (📖 54)

Parameter value (data)

Note!

Strings or data blocks cannot be transmitted.

Depending on the data format, the length of the parameter value is between 1 and 4 bytes.

Data are saved in the Motorola format, i.e. first the high byte (high word), then the low byte (low word):

Byte 5	Byte 6	Byte 7	Byte 8
High byte	Low byte	High byte	Low byte
High word		Low word	
Double word			

Principle for the assignment of bytes 5 ... 8 with parameter values of different lengths:

Byte 5	Byte 6	Byte 7	Byte 8
Parameter value (length 1)	0x00	0x00	0x00
Parameter value (length 2)		0x00	0x00
Parameter value (length 4)			

9.2.6 Error codes

The following error messages may appear:

Byte 8	Byte 7	Byte 6	Byte 5	Meaning
Error 1	Error 2	Error 3	Error 4	
0x06	0x03	0x00	0x00	No right to access
0x06	0x05	0x00	0x11	Invalid subindex
0x06	0x05	0x00	0x12	Data length too large
0x06	0x05	0x00	0x13	Data length too small
0x06	0x07	0x00	0x00	Object does not exist
0x06	0x08	0x00	0x00	Data types do not comply with each other
0x08	0x00	0x00	0x00	Request cannot be executed
0x08	0x00	0x00	0x20	Request cannot be executed at the moment
0x08	0x00	0x00	0x22	Request cannot be executed due to the device status / The parameter can only be changed in the case of a controller inhibit
0x08	0x00	0x00	0x30	Value ranged exited
0x08	0x00	0x00	0x31	Parameter value too high
0x08	0x00	0x00	0x32	Parameter value too low
0x08	0x00	0x00	0x80	Hardware error

9.2.7 Telegram examples

9.2.7.1 Read request: Querying the heatsink temperature

The heatsink temperature of the controller is to be read.

- Code to be read: C00061
- Heatsink temperature: 43 °C

Byte 1: Service (request)

Request = $0t110001_{bin}$

- Bit 0 ... 2 = 001_{bin} for read request
- Bit 3 = 0 (reserved)
- Bit 4/5 = 01_{bin} for 2-byte data length (only relevant for the response telegram)
- Bit 6 = handshake bit (t ≡ status is changed in the response telegram)
- Bit 7 = status bit (only relevant for the response telegram)

Byte 2: Subindex

Subindex = 0 because code C00061 does not contain any subindices.

Bytes 3 + 4: Index

Index = 24575 - code number = 24575 - 61 = 24514 = 0x5FC2

- Byte 3 (high byte) = 0x5F
- Byte 4 (low byte) = 0xC2

Bytes 5 ... 8: Data

The response telegram contains the value of code C00061:

Data 3 + 4 = 43 [°C] x 1 (internal factor) = 43 = 0x002B

Result

Request telegram from master to drive:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4	Data 3	Data 2	Data 1
0x01	0x00	0x5F	0xC2	0x00	0x00	0x00	0x00
$0t000001_{bin}$	00000000_{bin}	01011111_{bin}	11000010_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}
Waiting for change of handshake bit 6 in service byte 1 of the response.							

Response telegram from drive to master (for correct execution):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4	Data 3	Data 2	Data 1
0x11	0x00	0x5F	0xC2	0x00	0x2B	0x00	0x00
$0t010001_{bin}$	00000000_{bin}	01011111_{bin}	11000010_{bin}	00000000_{bin}	00101011_{bin}	00000000_{bin}	00000000_{bin}

9.2.7.2 Write request: Setting the deceleration time for quick stop (QSP)

In the controller, the deceleration time for quick stop (QSP) is to be set to 50 ms.

- Code to be written: C00105

Byte 1: Service (request)

Request = $0t110010_{bin}$

- Bit 0 ... 2 = 010_{bin} for write request
- Bit 3 = 0 (reserved)
- Bit 4/5 = 11_{bin} for 4-byte data length
- Bit 6 = handshake bit (t ≡ status is changed in the response telegram)
- Bit 7 = status bit (only relevant for the response telegram)

Byte 2: Subindex

Subindex = 0 because code C00105 does not contain any subindices.

Bytes 3 + 4: Index

Index = 24575 - code number = 24575 - 105 = 24470 = 0x5F96

- Byte 3 (high byte) = 0x5F
- Byte 4 (low byte) = 0x96

Bytes 5 ... 8: Data

The parameter value of 0.05 s to be set is multiplied by the code-specific factor of "1000" and entered in the user data:

Data 1 ... 4 = 0.05 [s] x 1000 (internal factor) = 50 = 0x00000032

Result:

Request telegram from master to drive:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4	Data 3	Data 2	Data 1
0x72	0x00	0x5F	0x96	0x00	0x00	0x00	0x32
$0t110010_{bin}$	00000000_{bin}	01011111_{bin}	10010110_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00110010_{bin}
Waiting for change of handshake bit 6 in service byte 1 of the response							

Response telegram from drive to master (for correct execution):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Service	Subindex	Index High byte	Index Low byte	Data 4	Data 3	Data 2	Data 1
0x40	0x00	0x5F	0x96	0x00	0x00	0x00	0x32
$0t000000_{bin}$	00000000_{bin}	01011111_{bin}	10010110_{bin}	00000000_{bin}	00000000_{bin}	00000000_{bin}	00110010_{bin}

9.3 PROFIdrive parameter data channel (DP-V1)

Data communication with PROFIBUS-DP-V0 is characterised by cyclic diagnostics and cyclic process data and parameter data transfer.

An optional service expansion is the acyclic parameter data transfer of PROFIBUS-DP-V1. This service does not impair the functionality of the standard services under PROFIBUS-DP-V0.

PROFIBUS-DP-V0 and PROFIBUS-DP-V1 can be operated simultaneously in the same network. This enables the step-by-step expansion or retrofitting of a system.

The services of PROFIBUS-DP-V1 can be used by the class 1 master (PLC) and the class 2 master (diagnostics master, etc.).

The integration of the acyclic service into the fixed bus cycle depends on the corresponding configuration of the class 1 master:

- With configuration, a time slot is reserved.
- Without configuration the acyclic service is appended when a class 2 master acyclically accesses a DP-V1 slave.

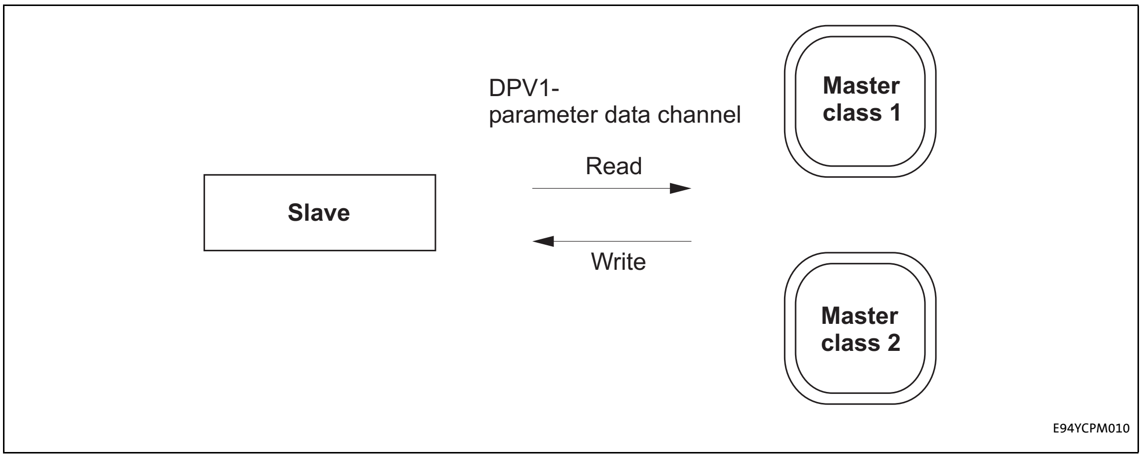
Features

- Parameter number and subindex addresses with a width of 16 bits each.
- Several parameter requests can be combined to one request (multi-parameter request).
- There is always only one parameter request in process (no pipelining).
- A parameter request/response must fit into a data block (max. 240 bytes). Requests/responses cannot be split into several data blocks.
- No spontaneous messages are transferred.
- There are only acyclic parameter requests.
- Profile-specific parameters can be read independently of the slave state.

9.3.1 Connection establishment between master and slave

A class 1 master can always request parameters from a slave if the slave is in the "Data_Exchange" state.

In addition to the class 1 master, a class 2 master can establish a communication connection to the slave:

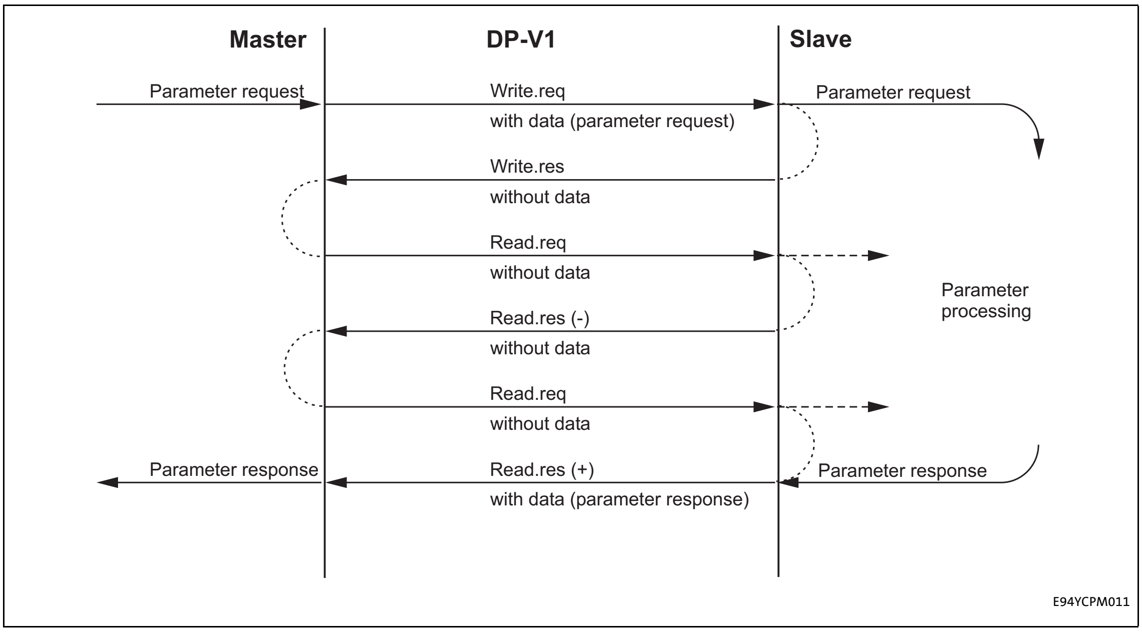


[9-2] Data communication via the DP-V1 parameter data channel

9.3.2 Acyclic data transfer

Note!

A parameter request refers to one or several parameter(s) (multi-parameter request).

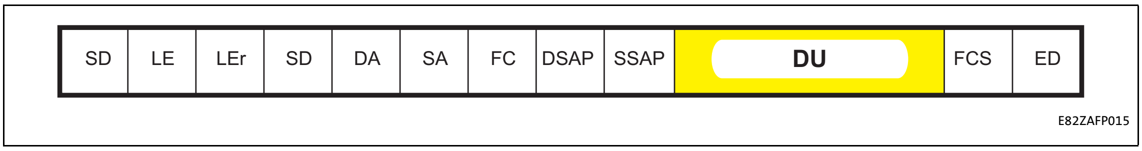


[9-3] Transmission directions

Explanation

- A "Write.req" is used to pass the data set (DB47) to the slave in the form of a parameter request.
- With "Write.res" the master receives the confirmation for the receipt of the message.
- The master requests the response of the slave with "Read.req".
- The slave responds with "Read.res (-)" if processing has not yet been completed.
- After parameter processing, the parameter request is completed by transmitting the parameter response to the master with "Read.res (+)".

9.3.3 Telegram structure



[9-4] PROFIBUS data telegram

The data unit (DU) contains the DP-V1 header and the parameter request or the parameter response.

The following subchapters describe the parameter request and the parameter response in detail.

Note!

The DP-V1 header consists of:

- Function identification
- Slot number
- Data set
- Length of the user data

Please refer to the corresponding PROFIBUS specification for further information on the DP-V1 header.

Assignment of the user data depending on the data type

Depending on the data type used, the user data are assigned as follows:

Data type	Length	User data assignment				
		Byte 1	Byte 2	Byte 3	Byte 4	Byte ...
String	x bytes					
U8	1 byte		0x00			
U16	2 bytes	High byte	Low byte			
U32	4 bytes	High word		Low word		
		High byte	Low byte	High byte	Low byte	

9.3.3.1 Reading parameter data from the controller

Note!

- When a read request is processed, no parameter value is written to the slave.
- In the case of a multi-parameter read request, the parameter attribute, index, and subindex are repeated with the number "n" of the requested parameters.
- A read request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices

Field	Data type	Values
Request reference	U8	This value is specified by the master
Request identification	U8	0x01: Request parameters for reading
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter attribute

Byte 5	Byte 6
Attribute	Number of subindices

Field	Data type	Values
Attribute	U8	0x10: Value
Number of subindices	U8	0x00 (For array elements: Enter the number of array elements required.)

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Index	U16	0x0001 ... 0xFFFF (1 ... 65535)
Subindex	U16	0x0001 ... 0xFFFF (1 ... 65535)

9.3.3.2 Response to a correctly executed read request

Note!

Responses to a read request do not contain parameter attributes, indices and subindices.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x01: Parameter has been read
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter format

Byte 5	Byte 6
Format	Number of values

Field	Data type	Values
Format	U8	0x02: Integer8 0x03: Integer16 0x04: Integer32 0x05: Unsigned8 0x06: Unsigned16 0x07: Unsigned32 0x09: Visible string 0x0A: Octet string 0x40: Zero 0x41: Byte 0x42: Word 0x43: Double word
Number of values	U8	0x01 or number of requested subindices/parameters (with several subindices/parameters only the parameter value is repeated). In the case of string codes, the number of characters is entered here.

Parameter value

Byte 7	Byte 8	Byte 9	Byte 10
Value			

Field	Data type	Values
Value	String	Any (length > 4 bytes possible)
	U8	0x00 .... 0xFF
	U16	0x0000 .... 0xFFFF
	U32	0x0000 0000 .... 0xFFFF FFFF

9.3.3.3 Response to a read error

Note!

In case of a multi-parameter request, correct and possibly faulty messages are summarised in one telegram.

Correct message

- Format: data type of the value requested
- Number of values: as described in chapter "Reading parameter data from the controller" (📖 61).
- Parameter value: value requested

Faulty message

- Format: 0x44
- Number of values: 0x01 or 0x02
- Error code without additional information (for number of values = 0x01) or
- Error code with additional information (for number of values = 0x02)

A faulty access to a parameter "n" is indicated at the nth position in the response telegram of a multi-parameter request.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x81: Parameter has not been read The data in the bytes 7 + 8 must be interpreted as an error code.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters requested)

Parameter format

Byte 5	Byte 6
Format	Number of values

Field	Data type	Values
Format	U8	0x44: Error
Number of values	U8	0x01: Error code without additional information 0x02: Error code with additional information

Error code

Byte 7	Byte 8	Byte 9	Byte 10
Error code		Additional information (if available)	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Error code	U16	0x0000 .... 0xFFFF ▶ Error codes (📖 70)
Additional information (if available)	U16	

9.3.3.4 Writing parameter data to the controller

Note!

When a multi-parameter write request is transferred, the ...
- Parameter attribute
- Index and subindex

and then the ...
- Parameter format
- Parameter value

... are repeated with the number "n" of the parameters addressed.

A write request must not exceed the maximum data length of 240 bytes.

Request header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices

Field	Data type	Values
Request reference	U8	This value is defined by the master.
Request identification	U8	0x02: Write parameter
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters addressed)

Parameter attribute

Byte 5	Byte 6
Attribute	Number of subindices

Field	Data type	Values
Attribute	U8	0x10: Value
Number of subindices	U8	0x00 (For array elements: Enter the number of array elements required.)

Index and subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Index	U16	0x0001 ... 0xFFFF (1 ... 65535)
Subindex	U16	0x0001 ... 0xFFFF (1 ... 65535)

Parameter format

Byte 11	Byte 12
Format	Number of values

Field	Data type	Values
Format	U8	0x02: Integer8 0x03: Integer16 0x04: Integer32 0x05: Unsigned8 0x06: Unsigned16 0x07: Unsigned32 0x09: Visible string 0x0A: Octet string 0x40: Zero 0x41: Byte 0x42: Word 0x43: Double word
Number of values	U8	0x01 or number of written subindices/parameters (with several subindices/parameters only the parameter value is repeated). In the case of string codes, the number of characters is entered here.

Parameter value

Byte 13	Byte 14	Byte 15	Byte 16
Value			

Field	Data type	Values
Value	String	Any (length > 4 bytes possible)
	U8	0x00 .... 0xFF
	U16	0x0000 .... 0xFFFF
	U32	0x0000 0000 .... 0xFFFF FFFF

9.3.3.5 Response to a correctly executed write request

Note!

In the case of a multi-parameter request, correct and possible faulty messages are summarised in one telegram. They have the following data contents:

Correct message

- Format: 0x40
- Number of values: 0x00

Faulty message

- Format: 0x44
- Number of values: 0x01 or 0x02
- Error code without additional information (for number of values = 0x01) or with additional information (for number of values = 0x02)

A faulty access to a parameter "n" is indicated at the nth position in the response telegram of a multi-parameter request.

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x02: Parameter has been written
Axis	U8	0x00 or 0x01
Number of indices	U8	0xn (n = number of parameter addressed)

9.3.3.6 Response to a write error

Response header

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices

Field	Data type	Values
Request reference	U8	Mirrored value of the parameter request
Response identification	U8	0x82: Parameter has not been written The data in the bytes 7 + 8 must be interpreted as an error code.
Axis	U8	0x00 or 0x01
Number of indices	U8	0x"n" (n = number of parameters addressed)

Parameter format

Byte 5	Byte 6
Format	Number of values

Field	Data type	Values
Format	U8	0x44: Error
Number of values	U8	0x01: Error code without additional information 0x02: Error code with additional information

Error code

Byte 7	Byte 8	Byte 9	Byte 10
Error code		Additional information (if available)	
High byte	Low byte	High byte	Low byte

Field	Data type	Values
Error code	U16	0x0000 .... 0xFFFF ▸ Error codes (📖 70)
Additional information (if available)	U16	

9.3.4 Error codes

Error code	Description	Explanation	Additional information
0x0000	Impermissible parameter number	Access to unavailable parameter	-
0x0001	Parameter value cannot be changed	Change access to a parameter value that cannot be changed	Subindex
0x0002	Lower or upper value limit exceeded	Change access with value beyond the value limits	Subindex
0x0003	Faulty subindex	Access to unavailable subindex	Subindex
0x0004	No array	Access with subindex to non-indicated parameter	-
0x0005	Wrong data type	Change access with value that does not match the data type of the parameter	-
0x0006	No setting permitted (only resettable)	Change access with value unequal to 0 where this is not permitted	Subindex
0x0007	Description element cannot be changed	Change access to a description element that cannot be changed	Subindex
0x0008	Reserved	(PROFIdrive profile V2: PPO-Write requested in the IR is not available)	-
0x0009	Description data not available	Access to unavailable description (parameter value is available)	-
0x000A	Reserved	(PROFIdrive profile V2: Wrong access group)	-
0x000B	No parameter change rights	Change access without parameter change rights	-
0x000C	Reserved	(PROFIdrive profile V2: Wrong password)	-
0x000D	Reserved	(PROFIdrive profile V2: Text in the cyclic traffic cannot be read)	-
0x000E	Reserved	(PROFIdrive profile V2: Name in the cyclic traffic cannot be read)	-
0x000F	No text array available	Access to unavailable text array (parameter value is available)	-
0x0010	Reserved	(PROFIdrive profile V2: Missing PPO-Write)	-
0x0011	Request cannot be executed due to the operating state	Access is not possible due to temporary reasons not specified here	-
0x0012	Reserved	(PROFIdrive profile V2: Other error)	-
0x0013	Reserved	(PROFIdrive profile V2: date in the cyclic traffic cannot be read)	-
0x0014	Value impermissible	Change access with the value that is inside the value limits but not permissible for other permanent reasons (parameters with defined individual values)	Subindex
0x0015	Response too long	The length of the current response exceeds the maximum transmittable length	-
0x0016	Parameter address impermissible	Impermissible or non-supported value for attribute, number of subindices, parameter number, or subindex, or a combination	-
0x0017	Format impermissible	Write request: Impermissible or non-supported format of parameter data	-
0x0018	Number of values not consistent	Write request: Number of values of the parameter data do not match the number of subindices in the parameter address	-
0x0019 ... 0x0064	Reserved	-	-

Error code	Description	Explanation	Additional information
0x0065	Manufacturer-specific	-	-
...			
0x00FF			

9.3.5 Telegram examples

9.3.5.1 Read request: Querying the heatsink temperature

The heatsink temperature of the controller is to be read.

- Code to be read: C00061
- Heatsink temperature: 43 °C

Parameter request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices
0xXX	0x01	0x00	0x01
	Request parameters for reading		

Byte 5	Byte 6
Attribute	Number of subindices
0x10	0x00
Value	No subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte
0x5F	0xC2	0x00	0x00
Index = 24575 - code no. = 24575 - 61 = 24514 = 0x5F C2		No subindex	

Parameter response to a correctly executed read request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices
0xXX	0x01	0x00	0x01
	Parameter has been read		

Byte 5	Byte 6
Format	Number of values
0x03	0x01
Integer16	1 value

Byte 7	Byte 8
Value	
High byte	Low byte
0x00	0x2B
Value read = 0x 00 2B = 43 x 1 (internal factor) = 43 [°C]	

Parameter response to a read error

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices
0xXX	0x81	0x00	0x01
	Parameter has not been read		

Byte 5	Byte 6
Format	Number of values
0x44	0x01
Error	Error code without additional information

Byte 7	Byte 8
Error code	
High byte	Low byte
For the meaning, see the "Error codes" (📖 70) chapter	

9.3.5.2 Write request: Setting the deceleration time for quick stop (QSP)

In the controller, the deceleration time for quick stop (QSP) is to be set to 50 ms.

Code to be written: C00105

Parameter request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference	Request identification	Axis	Number of indices
0xXX	0x02	0x00	0x01
	Write parameter	Axis 0	1 index

Byte 5	Byte 6
Attribute	Number of subindices
0x10	0x00
Value	No subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte
0x5F	0x96	0x00	0x00
Index = 24575 - code no. = 24575 - 105 = 24470 = 0x5F 96		No subindex	

Byte 11	Byte 12
Format	Number of values
0x43	0x01
Double word	1 value

Byte 13	Byte 14	Byte 15	Byte 16
Value			
High word: high byte	High word: low byte	Low- word: high byte	Low word: low byte
0x00	0x00	0x00	0x32
Value to be written = 0.05 [s] x 1000 (internal factor) = 50 = 0x00 00 00 32			

Parameter response to a correctly executed write request

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices
0xXX	0x02	0x00	0x01
	Parameter has been written		1 index

Parameter response to a read error

Byte 1	Byte 2	Byte 3	Byte 4
Request reference (mirrored)	Response identification	Axis (mirrored)	Number of indices
0xXX	0x82	0x00	0x01
	Parameter has not been written		1 index

Byte 5	Byte 6
Format	Number of values
0x44	0x01
Error	Error code without additional information

Byte 7	Byte 8
Error code	
High byte	Low byte
For the meaning, see Error codes (📖 70)	

9.4 Consistent parameter data

In the PROFIBUS communication system, data are permanently exchanged between the host (CPU + PROFIBUS master) and the standard device via the plugged-on communication module. Both the PROFIBUS master and the CPU (central processing unit) of the host access a joint memory: the dual port memory (DPM).

The DPM permits a data exchange in both directions (write/read):

It could happen that a slower PROFIBUS master writing would be overtaken by a faster CPU reading within a cycle time without any further data organisation.

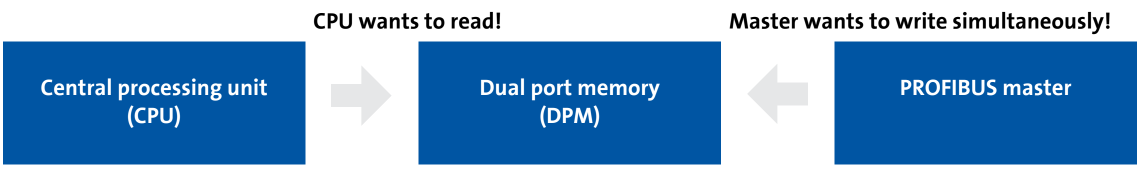
In order to avoid such an impermissible state, the parameter data to be transmitted must be marked as "consistent".

Data communication with consistent data

With consistency, either "reading" or "writing" is possible when the master and the CPU simultaneously access the memory:

- The PROFIBUS master transfers data only as a complete data set.
- The CPU can only access completely updated data sets.
- The PROFIBUS master cannot read or write data as long as the CPU accesses consistent data.

The result becomes clear from the example below:



1. As the master can only write if the CPU does not read, the master waits until the data are read completely by the CPU.
2. The master only writes a complete data set into DPM.

Configuring consistent data

Note!

Consistency is achieved by an appropriate PROFIBUS master configuration.

For this purpose, refer to the documentation for your configuring software.

10 PROFIsafe

PROFIBUS enables the transfer of safe information via the PROFIsafe protocol in accordance with the "PROFIsafe - Profile for Safety Technology" specification, version 2.0.

Note!

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

- The PROFIsafe data are sent in the first slot of a PROFIBUS data telegram.
- One bit of the PROFIsafe data is always used to control specific safety functions.
- The structure of the PROFIsafe data is described in the PROFIsafe profile.
- The length of the PROFIsafe data (also called PROFIsafe message) always is 8 bytes.

Documentation for the safety module SM301 (E94AYAE)

Here you can find detailed information on the PROFIsafe connection.

11 Monitoring

11.1 Permanent interruption of PROFIBUS communication

If PROFIBUS communication is interrupted permanently, e.g. by cable breakage or failure of the PROFIBUS master, no process data are transmitted to the slave being in the "Data Exchange" state.

After the watchdog monitoring time determined by the PROFIBUS master has expired, the response parameterised under C13880/1 / C14880/1 is executed in the controller (slave).

The process data handling depends on the setting in C13885 / C14885. (The last data sent by the master can either be used or set to zero.)

Preconditions for a controller (slave) response

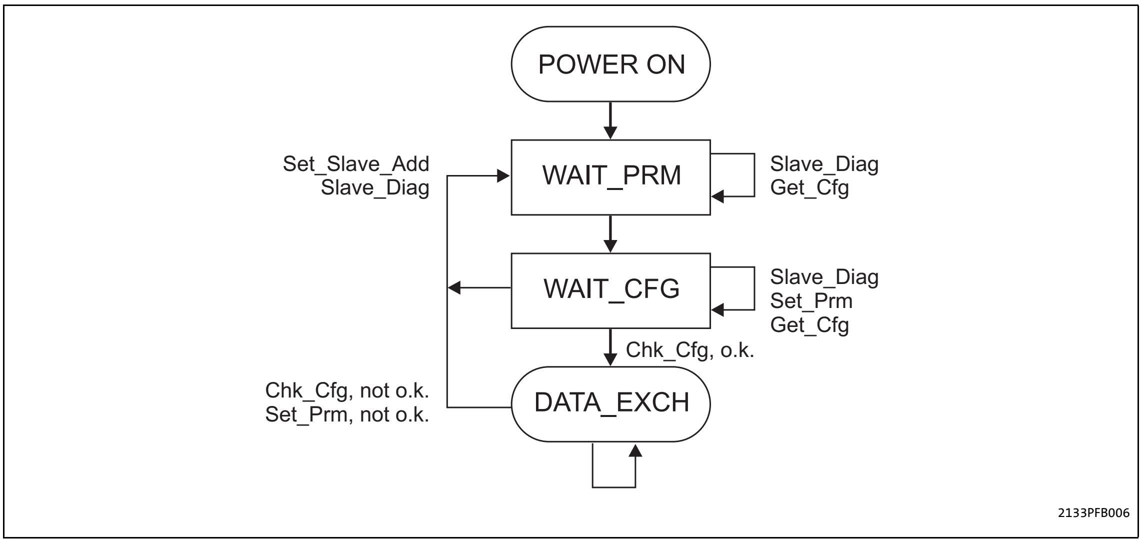
- A monitoring time of 1 ... 65534 ms is set for the "Data_Exchange" state (C13881 / C14881).
  A value of "65535 ms" (Lenze setting) deactivates the monitoring.
- A response is set for the slave in C13880/1 / C14880/1 (Lenze setting: "No response").
- The slave is in the "Data_Exchange" state.
- The watchdog monitoring time is configured correctly in the master.

If one of these preconditions is not met, the response to the absence of cyclic process data telegrams from the master is not executed.

▶ Settings and displays in the »Engineer« (📖 80)

11.2 Short-time interruption of PROFIBUS communication

From software version 01.21, the current DP states are displayed in C13868 / C14868.



[11-1] DP states (Decentralized Peripherals) for short-time interruption of communication

The master detects the communication fault and transfers the slave to the "WAIT_PRM" state of the DP state machine after only a few microseconds (see Fig. [11-1]).

Only after the state chain of the DP state machine ending in the "Data_Exchange" state (DATA_EXCH) has been passed through, the watchdog monitoring time calculated for the slave (in milliseconds) continues to run.

Note!

The watchdog monitoring time does not continue to run when the slave does not reach the "Data_Exchange" state due to repeated communication errors (e.g. caused by loose contact).

Additional monitoring for the data exchange

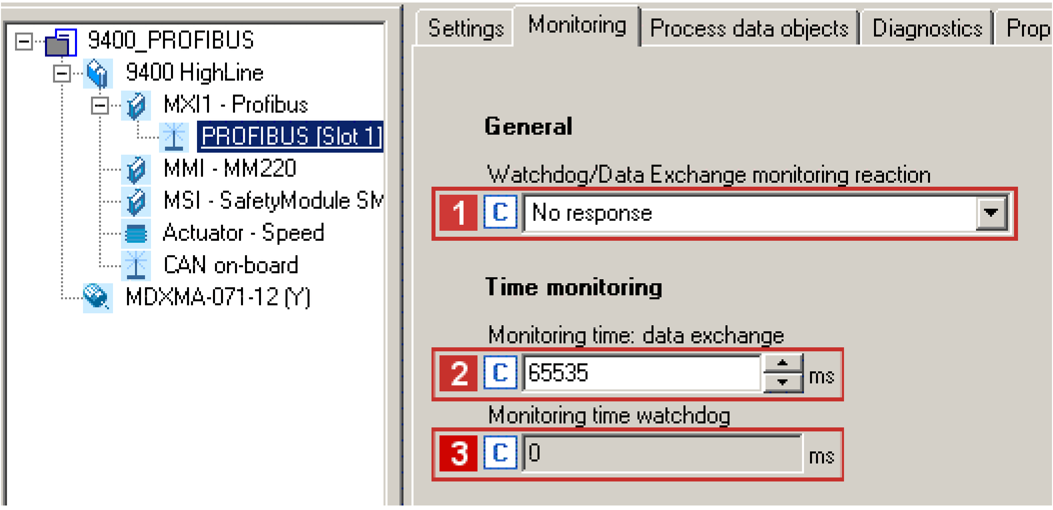
An additional monitoring function for data exchange is available under C13881 / C14881. This monitoring function already becomes active when the "Data_Exchange" state is exited and the parameterised time (0 ... 65535 ms) has expired. The active monitoring function triggers the response parameterised under C13880/1 / C14880/1.

Note!

Observe the following condition for the time setting:

Monitoring time for data exchange (C13881 / C14881) ≤ watchdog monitoring time of PROFIBUS (C13882/1 / C14882/1).

11.3 Settings and displays in the »Engineer«



On the **Monitoring** tab of the »Engineer«, you can set or display the following parameters:

- Set a response 1 (C13880/1 / C14880/1) to the interruption of communication.
- Set a monitoring time 2 (C13881 / C14881) for the "Data_Exchange" state.
- The watchdog monitoring time specified in the master 3 (C13882/1 / C14882/1).

11.4 Interruption of internal communication

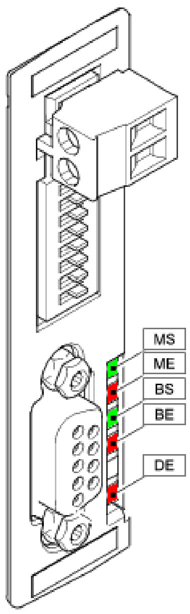
The response to a communication error between the communication module and the standard device can be set via the standard device codes C01501 (module in slot MXI1) and C01502 (module in slot MXI2) (📖 92).

12 Diagnostics

For diagnosing faults of the PROFIBUS module, the LEDs on the front panel are available. Furthermore, you can query the current bus status via codes C13861 / C14861.

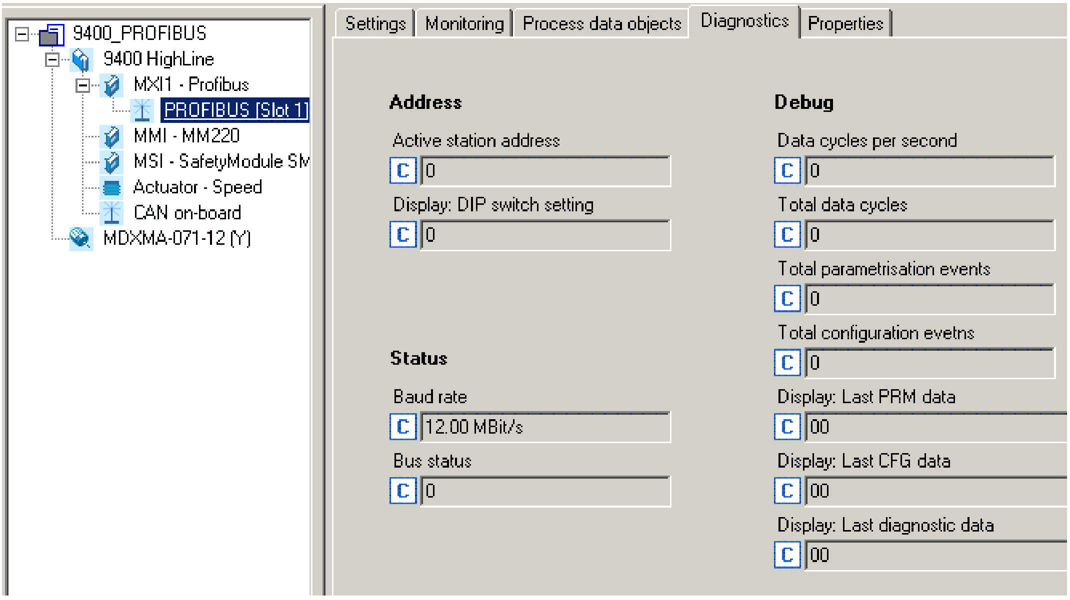
12.1 LED status displays

> **Note!**
>
> During normal operation, the LED **BS** blinks and the LED **MS** is permanently lit.

LED status display	Pos.	Colour	State	Description
	MS	Green	On	The communication module is supplied with voltage and has established a connection to the standard device.
			Blinking	The communication module is supplied with voltage, but has not yet established a connection to the standard device. (Standard device is switched off, initialising or not present.)
	ME	Red	On	An error concerning the communication module has occurred.
	BS	Green	Blinking	The communication module is in the DATA_EXCH state ("Data_Exchange"). Data are exchanged via the PROFIBUS.
			Off	The communication module is not active on the fieldbus or is being initialised.
	BE	Red	On	Bus error/fault is active (e.g. bus cable unplugged).
	DE	Red	On	The communication module is not accepted by the standard device or the standard device is not active. (See notes in the documentation for the standard device.)

12.2 Diagnostics with the »Engineer«

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



Querying the current bus status

C13861 / C14861 indicate the current PROFIBUS state in bit-coded form:

Bit assignment				Description	
Bit 3	Bit 2	Bit 1	Bit 0	Reserved	
		Bit 5	Bit 4	**Status of the DP state machine (DP-STATE)**	
		0	0	WAIT_PRM	The slave waits for a parameter data telegram after acceleration. Other types of telegrams will not be processed. Data exchange is not yet possible.
		0	1	WAIT_CFG	The slave waits for the configuration telegram that specifies the number of input and output bytes. The master informs the slave about the number of I/O bytes that will be transferred.
		1	0	DATA_EXCH	If the parameter settings as well as the configuration have been accepted by the firmware and by the application, the slave state changes to DATA_EXCH ("Data Exchange", exchange of user data with the master).
		1	1	Not possible	
		Bit 7	Bit 6	**Status of the watchdog state machine (WD-STATE)**	
		0	0	BAUD_SEARCH	The PROFIBUS slave is able to detect the baud rate automatically.
		0	1	BAUD_CONTROL	After recognising the correct baud rate, the slave status changes to BAUD_CONTROL and the baud rate is monitored.
		1	0	DP_CONTROL	The DP_CONTROL state serves for response monitoring of the PROFIBUS master.
		1	1	Not possible	
Bit 11	Bit 10	Bit 9	Bit 8	**Detected PROFIBUS baud rate**	
0	0	0	0	12 Mbps	
0	0	0	1	6 Mbps	
0	0	1	0	3 Mbps	
0	0	1	1	1.5 Mbps	
0	1	0	0	500 kbps	
0	1	0	1	187.5 kbps	
0	1	1	0	93.75 kbps	
0	1	1	1	45.45 kbps	
1	0	0	0	19.2 kbps	
1	0	0	1	9.6 kbps	
Bit 15	Bit 14	Bit 13	Bit 12	Reserved	

From software version 01.21, the current DP states are displayed in C13868 / C14868 (see fig. [11-1], chapter Short-time interruption of PROFIBUS communication (79)).

12.3 Advanced diagnostic message

Errors in the standard device and its plugged-in modules are transmitted to the PROFIBUS master in the form of advanced diagnostic messages.

Structure of the diagnostic message

Byte	Description
1	Bit 0: Station does not exist (set by the master). Bit 1: Slave is not ready for data exchange. Bit 2: Configuration data do not correspond. Bit 3: Slave has extended diagnostic data. Bit 4: Requested function is not supported by the slave. Bit 5: Slave response is invalid (set by the master) Bit 6: Incorrect parameter setting Bit 7: Slave has been parameterised by another master (set by the master).
2	Bit 0: Slave must be parameterised again. Bit 1: Static diagnostics Bit 2: Permanently set to "1". Bit 3: Watchdog active Bit 4: Freeze command received. Bit 5: Sync command received. Bit 6: Reserved Bit 7: Slave is deactivated (set by the master).
3	Bit 7: Diagnostics overflow - amount of diagnostic information present in the slave is too large to fit into one telegram.
4	Bits 0 ... 7: Master address after parameter setting ("0xFF" without parameterisation)
5	Bits 0 ... 7: ID number (high byte)
6	Bits 0 ... 7: ID number (low byte)
7	Header • The header contains the block length of the advanced diagnostics including the header byte. • In this case, the value of the entry is "0x0A" (bytes 7 ... 16 = 10 bytes).
8	Status_Type The value of this entry is fixed. For the following bit assignment it is "0x81": • Bit 7 = 1: "status" • Bit 0 = 1: "status message" • Value of all other bits = 0
9	Slot_Number The value of the slot number is "0x00".
10	Specifier • A detected error is entered in the specifier with the identification "0x0" (status coming). • An eliminated error is entered in the specifier with the identification "0x02" (status going). • If no errors are indicated, the entry in the specifier has the value "0x00" (no further differentiation).
11 12	PROFIsafe, error number of the safety module SM301 (E94AYAE) • If an error occurs in the safety module, byte 11 (low byte) and byte 12 (high byte) contain the corresponding error number. • See also documentation for the safety module.
13 ... 16	Error code of the Servo Drive 9400 • Code **C00168** can be used to read out the contents of the fault memory. • Detailed information on the error codes of the Servo Drive 9400 can be found in the documentation of the controller.

12.3.1 Example 1: Error in parameter data "Incorrect PROFIsafe target address"

This error was caused while communication of the PROFIBUS master was established because the PROFIsafe target address configured in the PROFIBUS master does not correspond to the set address anymore.

Byte	Value [hex]	Description
1	x	Standard data (PRM_Fault)
...		
6		
7	0A	Block length of the advanced diagnostics = 10 bytes
8	81	Status message
9	00	Slot 0
10	01	Status coming
11	40	PROFIsafe error "0x0040" (64) GSG: "F address differ to F_Dest_Add"
12	00	
13	00	No other error message
...	...	
16	00	

12.3.2 Example 2: Error "Undervoltage in the DC bus" in the Servo Drive 9400

Byte	Value [hex]	Description
1	x	Standard data (PRM_Fault)
...		
6		
7	0A	Block length of the advanced diagnostics = 10 bytes
8	81	Status message
9	00	Slot 0
10	01	Status coming
11	00	No PROFIsafe/safety error
12	00	
13	0F	Error message 0x447B000F "Undervoltage in DC bus" • Response: "Fault" • Instance ID: 1 • Module ID: 0x7B • Error ID: 0x0F
14	00	
15	7B	
16	44	

13 Error messages

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

Software manual/online help for Servo Drive 9400

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

13.1 Short overview of the PROFIBUS error messages

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

Tip!

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

Error number		Designation	Response (Lenze setting)	Adjustable in
hex	dec			
0x00c83100	13119744	Profibus: Lost connection to 9400 base device	0: No response	-
0x00c85531	13129009	Profibus: No memory access	6: Information	-
0x00c85532	13129010	Profibus: Read memory error	6: Information	-
0x00c85533	13129011	Profibus: Write memory error	6: Information	-
0x00c86010	13131792	Profibus: Restart after watchdogreset	1: Fault	-
0x00c86011	13131793	Profibus: Internal error	1: Fault	-
0x00c86100	13132032	Profibus: Internal error	0: No response	-
0x00c86101	13132033	Profibus: Internal error	1: Fault	-
0x00c86110	13132048	Profibus: Internal mapping error	4: Warning locked	-
0x00c8641f	13132831	Profibus: Invalid parameter record	1: Fault	-
0x00c86420	13132832	Profibus: Error: Lenze setting loaded	1: Fault	-
0x00c88130	13140272	Profibus watchdog: Monitoring time elapsed	0: No response	C13880/1 C14880/1
0x00c88131	13140273	Profibus: Data_Exchange state exited	0: No response	C13880/1 C14880/1
0x00c88132	13140274	Profibus watchdog: DP-V1 MSAC2 elapsed	0: No response	C13880/2 C14880/2

13.2 Possible causes and remedies

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

Profibus: Lost connection to 9400 standard device [0x00c83100]

Response (Lenze setting printed in bold)	**Setting:** not possible
☒ **None** ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
• Network cable (plug) is defective. • Network cable is not connected to the PROFIBUS terminal X201. • Voltage supply is interrupted.	Check cables and terminals. Connect network cable to the PROFIBUS terminal X201.

Profibus: No memory access [0x00c85531]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	

Cause	Remedy
Access to memory was not possible.	Repeat the download of the application (including module)

Profibus: Read memory error [0x00c85532]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	

Cause	Remedy
Parameter could not be read.	Repeat the download of the application (including module)

Profibus: Write memory error [0x00c85533]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☒ **Information**	

Cause	Remedy
Parameter could not be written.	Repeat the download of the application (including module)

Profibus: Restart after watchdog reset [0x00c86010]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Module defective.	If this occurs repeatedly, contact the Lenze service.

Profibus: Internal error [0x00c86011]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Module defective.	If this occurs repeatedly, contact the Lenze service.

Profibus: Internal error [0x00c86100]

Response (Lenze setting printed in bold)	**Setting:** not possible
☒ **None** ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Internal error.	If this occurs repeatedly, contact the Lenze service.

Profibus: Internal error [0x00c86101]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
The communication module carries out an automatic software reset and reinitialises itself.	If this occurs repeatedly, contact the Lenze service.

Profibus: Internal mapping error [0x00c86110]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☐ Fault ☐ Trouble ☐ Quick stop by trouble ☒ **Warning locked** ☐ Warning ☐ Information	

Cause	Remedy
The selected PDO mapping is invalid: • An object not supporting the required properties has been specified. • More than 64 bytes of data have been mapped.	Repeat the PDO mapping.

Profibus: Invalid parameter set [0x00c8641f]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
No active parameter set could be loaded.	Repeat the download of the application (including module)

Profibus: Error: Lenze setting loaded [0x00c86420]

Response (Lenze setting printed in bold)	**Setting:** not possible
☐ None ☐ System fault ☒ **Fault** ☐ Trouble ☐ Quick stop by trouble ☐ Warning locked ☐ Warning ☐ Information	

Cause	Remedy
Access to parameter set was not successful.	Repeat the download of the application (including module).

Profibus watchdog: Monitoring time elapsed [0x00c88130]

Response (Lenze setting printed in bold)	**Setting:** C13880/1 / C14880/1 (☑ Adjustable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
Permanent interruption of communication to the PROFIBUS master. See also chapter "Permanent interruption of PROFIBUS communication" (📖 78).	Check cables and terminals.

Profibus: Data_Exchange state exited [0x00c88131]

Response (Lenze setting printed in bold)	**Setting:** C13880/1 / C14880/1 (☑ Configurable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
Data exchange via PROFIBUS has been terminated. See also chapter "Permanent interruption of PROFIBUS communication" (📖 78).	Check cables and terminals. The slave must receive new parameterisation and configuration files from the master in order to be able to exchange data again.

Profibus watchdog: DP-V1 MSAC2 elapsed [0x00c88132]

Response (Lenze setting printed in bold)	**Setting:** C13880/2 / C14880/2 (☑ Configurable response)
☒ **None** ☐ System fault ☑ Fault ☐ Trouble ☑ Quick stop by trouble ☑ Warning locked ☐ Warning ☑ Information	

Cause	Remedy
Permanent interruption of communication to C2-PROFIBUS master. See also chapter "Permanent interruption of PROFIBUS communication" (📖 78).	Check cables and terminals.

14 Parameter reference

This chapter supplements the parameter list and the table of attributes contained in the software manual and in the »Engineer« online help for the Servo Drive 9400 with the parameters of the E94AYCPM (PROFIBUS) communication module.

Software manual/»Engineer« online help for Servo Drive 9400
Here you can find general information on parameters.

14.1 Communication-relevant parameters of the standard device

This chapter lists the communication-relevant parameters of the Servo Drive 9400 in numerically ascending order.

C00615

Parameter | Name: **C00615 | Resp. to imp. device conf.**		Data type: UNSIGNED_32 Index: 23960_d = $5D98_h$
Response to incorrect device configuration		
Selection list		
1	Error	
3	Quick stop by trouble	
4	Warning locked	
6	Information	
0	No response	
Subcodes	**Lenze setting**	**Info**
C00615/1	0: No response	Reserved
C00615/2	1: Fault	Resp. to imp. module in MXI1
C00615/3	1: Fault	Resp. to imp. module in MXI2
C00615/4	0: No response	Reserved
C00615/5	0: No response	Reserved
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer		

C00636

Parameter | Name: **C00636 | Resp. to new module in MXI1**		Data type: UNSIGNED_32 Index: 23939_d = $5D83_h$
Response when a new module has been inserted in module slot 1 of the standard device.		
Selection list (Lenze setting printed in bold)		
1	Error	
6	Information	
5	Warning	
4	**Warning locked**	
3	Quick stop by trouble	
0	No response	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer		

C00637

Parameter | Name: **C00637 | Resp. to new module in MXI2**		Data type: UNSIGNED_32 Index: 23939_d = $5D83_h$
Response when a new module has been inserted in module slot 2 of the standard device.		
Selection list (Lenze setting printed in bold)		
1	Error	
6	Information	
5	Warning	
4	**Warning locked**	
3	Quick stop by trouble	
0	No response	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer		

C01501

Parameter | Name: **C01501 | Resp. to comm. error with MXI1**		Data type: UNSIGNED_32 Index: 23074_d = $5A22_h$
Response to a communication error between an "intelligent" module in module slot 1 and the standard device		
Selection list (Lenze setting printed in bold)		
0	**No response**	
1	Error	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer		

C01502

Parameter | Name: **C01502 | Resp. to comm. error with MXI2**		Data type: UNSIGNED_32 Index: 23074_d = $5A22_h$
Response to a communication error between an "intelligent" module in module slot 2 and the standard device		
Selection list (Lenze setting printed in bold)		
0	**No response**	
1	Error	
3	Quick stop by trouble	
4	Warning locked	
5	Warning	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer		

14.2 Parameters of the communication module for slot MXI1

This chapter lists the parameters of the E94AYCPM communication module (PROFIBUS) for slot MXI2 of the Servo Drive 9400 in numerically ascending order.

C13231

Parameter | Name: **C13231 | Service code**	Data type: UNSIGNED_64 Index: 11344_d = $2C50_h$
This code is for device-internal use only and must not be overwritten by the user!	

C13489

Parameter | Name: **C13489 | Service code**	Data type: UNSIGNED_8 Index: 11086_d = $2B4E_h$
This code is for device-internal use only and must not be overwritten by the user!	

C13531

Parameter | Name: **C13531 | Service code**	Data type: UNSIGNED_64 Index: 11044_d = $2B24_h$
This code is for device-internal use only and must not be overwritten by the user!	

C13789

Parameter | Name: **C13789 | Service code**	Data type: UNSIGNED_8 Index: 10786_d = $2A22_h$
This code is for device-internal use only and must not be overwritten by the user!	

C13850

Parameter | Name: **C13850 | All words to master**			Data type: UNSIGNED_16 Index: 10725_d = $29E5_h$
Display of the process data words transferred from the communication module to the PROFIBUS master. In the subcodes 1 ... 32, all process data words to the master are displayed. Only the configured process data words are valid.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13850/1			
...			
C13850/32			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13851

Parameter | Name: **C13851 | All words from master**			Data type: UNSIGNED_16 Index: 10724_d = $29E4_h$
Display of the process data words transferred from the PROFIBUS master to the communication module. In the subcodes 1 ... 32, all process data words to the master are displayed. Only the configured process data words are valid.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13851/1			
...			
C13851/32			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13852

Parameter | Name: **C13852 | All words to standard device**			Data type: UNSIGNED_16 Index: 10723_d = $29E3_h$
Display of the process data words 1 ... 32 transferred from the communication module to the standard device. In the subcodes 1 ... 32, all process data words from the communication module are displayed.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13852/1			
...			
C13852/32			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13853

Parameter | Name: **C13853 | All words from standard device**			Data type: UNSIGNED_16 Index: 10722_d = $29E2_h$
Display of the process data words 1 ... 32 transferred from the standard device to the communication module. In the subcodes 1 ... 32, all process data words from the standard device are displayed.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13853/1			
...			
C13853/32			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13854

Parameter | Name: **C13854 | COM word to standard device**	Data type: BITFIELD_32 Index: 10721_d = $29E1_h$
Available from software version 01.21! **This code is for device-internal use only and must not be overwritten by the user!**	

C13855

Parameter | Name: **C13855 | COM word from standard device**	Data type: BITFIELD_32 Index: 10720_d = $29E0_h$
Available from software version 01.21! This code is for device-internal use only and must not be overwritten by the user!	

C13860

Parameter | Name: **C13860 | Settings**			Data type: UNSIGNED_8 Index: 10715_d = $29DB_h$
Display of the current configuration data.			
Display area (min. value | unit | max. value)			
0		255	
Subcodes			**Info**
C13860/1			PROFIsafe • 0: Not active • 1: Active
C13860/2			Number of process data words (1 ... 32 words)
C13860/3			DRIVECOM parameter data channel • 0: Not active • 1: Active
C13860/4			Number of PROFIsafe words (0 or 4 words)
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13861

Parameter | Name: **C13861 | Bus status**			Data type: UNSIGNED_16 Index: 10714_d = $29DA_h$
Bit-coded display of the current bus state. ▸ Querying the current bus status (📖 84)			
Display area (min. value | unit | max. value)			
0		65535	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13862

Parameter | Name: **C13862 | Bus counter**			Data type: UNSIGNED_16 Index: 10713_d = $29D9_h$
When the maximum count value of 65535 is reached, the counter starts again with 0.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C13862/1			Data cycles per second
C13862/2			Total data cycles
C13862/3			Total parameterisation events
C13862/4			Total configuration events
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13863

Parameter | Name: **C13863 | Baud rate**		Data type: UNSIGNED_8 Index: 10712_d = $29D8_h$
Display of the baud rate		
Selection list (read only)		
0	12.00 Mbps	
1	6.00 Mbps	
2	3.00 Mbps	
3	1.50 Mbps	
4	500.00 kbps	
5	187.50 kbps	
6	93.75 kbps	
7	45.45 kbps	
8	19.20 kbps	
9	9.60 kbps	
15	No baud rate detected	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13864

Parameter | Name: **C13864 | Active station address**			Data type: UNSIGNED_8 Index: 10711_d = $29D7_h$
Display of the active station address If all DIP switches 1 ... 64 are in the "OFF" position (Lenze setting), the station address set in C13899 becomes active and is displayed here after switching on. ▸ Setting the station address (🕮 31)			
Display area (min. value | unit | max. value)			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13865

Parameter | Name: **C13865 | Display: Most recent PRM data**	Data type: OCTET_STRING Index: 10710_d = $29D6_h$
Display of the last parameter data sent by the PROFIBUS master with the "Set-Prm" telegram (ASCII string with 24 characters)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13866

Parameter | Name: **C13866 | Display: Most recent CFG data**	Data type: OCTET_STRING Index: 10709_d = $29D5_h$
Display of the last configuration data sent by the PROFIBUS master with the "Chk-Cfg" telegram (ASCII string with 22 characters)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13867

Parameter | Name: **C13867 | Display: Most recent diagnostic data**	Data type: OCTET_STRING Index: 10708_d = $29D4_h$
Display of the last diagnostic data sent to the PROFIBUS master (ASCII string with 16 characters) ▸ Advanced diagnostic message (📖 85)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13868

Parameter | Name: **C13868 | Display: DP state**		Data type: UNSIGNED_8 Index: 10707_d = $29D3_h$
Available from software version 01.21! Display of the DP states (Decentralized Peripherals) ▸ Short-time interruption of PROFIBUS communication (📖 79), see fig. [11-1]		
Selection list (read only)		
0	Waiting for parameterisation	
1	Waiting for configuration	
2	Data exchange	
3	n/a	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13880

Parameter | Name: **C13880 | Reaction on communication fault**		Data type: UNSIGNED_8 Index: 10695_d = $29C7_h$
Monitoring response to a communication fault on the PROFIBUS A change in the monitoring response is effective immediately. ▸ Permanent interruption of PROFIBUS communication (📖 78)		
Selection list		
0	No response	
1	Error	
3	Quick stop by trouble	
4	Warning locked	
6	Information	
Subcodes	**Lenze setting**	**Info**
C13880/1	0: No response	The response set here for the "Watchdog/Data Exchange" monitoring function is executed if the bus station ... • does not receive a message from the master within the watchdog monitoring time (displayed in C13882/1) if there is an active connection. • detects that it is no longer in the "Data_Exchange" status. Please see also the notes given under C13881.
C13880/2	0: No response	The response set here for the "DPV1 MSAC2" monitoring is executed if the bus station does not receive a "DPV1 MSAC2" message from the master within the monitoring time (displayed in C13882/2) if there is an active connection) and the MSAC2 connection is stopped by the slave. **Note:** We recommend only setting "information" as response so that no drive-relevant response is executed.
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13881

Parameter | Name: **C13881 | Monitoring time: Data exchange**			Data type: UNSIGNED_16 Index: 10694_d = $29C6_h$
If the "Data Exchange" state is exited, the response parameterised under C13880/1 is carried out when the monitoring time for data exchange set here has expired. • A value of "65535" in this code deactivates the monitoring function. • A change in monitoring is effective immediately. • The value set here for the monitoring time must be smaller than the watchdog monitoring time (C13882/1). ▸ Permanent interruption of PROFIBUS communication (🕮 78)			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	65535	**65535 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13882

Parameter | Name: **C13882 | Monitoring time: Watchdog**			Data type: UNSIGNED_32 Index: 10693_d = $29C5_h$
Display of the watchdog monitoring time determined by the PROFIBUS master • A change in the watchdog monitoring time is immediately effective. • Monitoring starts with the receipt of the first telegram. • When a value of "0" is displayed, the monitoring function is deactivated. ▸ Permanent interruption of PROFIBUS communication (🕮 78)			
Display area (min. value | unit | max. value)			
0	ms	4294967295	
Subcodes			**Info**
C13882/1			Watchdog monitoring time
C13882/2			DP-V1 MSC2
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13885

Parameter | Name: **C13885 | Clear process data**		Data type: UNSIGNED_8 Index: 10690_d = $29C2_h$
Selection of the process data which the controller will process in the event of a PROFIBUS failure in order to maintain internal communication.		
Selection list (Lenze setting printed in bold)		
0	Use of most recent master PDOs	
1	**PDOs are set to the value '0'**	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13886

Parameter | Name: **C13886 | Set ext. diagnostic bit upon**		Data type: BITFIELD_8 Index: 10689_d = $29C1_h$
Bit-coded selection of the error responses in the standard device causing the external diagnostic bit ("diag bit") to be set (see PROFIBUS specification; bit 3 of byte 1 of the DP diagnostic messages). • The diagnostic bit is sent to the PROFIBUS master where it is evaluated separately. • The diagnostic bit is always set when a system error occurs. • The Lenze setting "0" means that the diagnostic bit is not set for the following error responses. • An advanced diagnostic message is always sent.		
Value is bit-coded:		
Bit 0	Error	
Bit 1	Trouble	
Bit 2	Quick stop by trouble	
Bit 3	Warning locked	
Bit 4	Warning	
Bit 5	Reserved	
Bit 6	Reserved	
Bit 7	Reserved	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13887

Parameter | Name: **C13887 | Suppress signalling diag. mess. upon**		Data type: BITFIELD_8 Index: 10688_d = $29C0_h$
Available from software version 01.20! Selection of the error responses not causing a diagnostic request to the PROFIBUS master. A diagnostic request is always signalled if a system error or an error message from the SM301 (E94AYAE) safety module occurs. The Lenze setting "0" means that for each of the following error responses a diagnostic request is signalled.		
Value is bit-coded:		**Info**
Bit 0	Error	
Bit 1	Trouble	
Bit 2	Quick stop by trouble	
Bit 3	Warning locked	
Bit 4	Warning	
Bit 5	Reserved	
Bit 6	Reserved	
Bit 7	Error message 0x00c88130	Available from software version 01.21! Selective suppression of the error message "Available connection to 9400 standard device lost"
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C13897

Parameter | Name: **C13897 | PROFIsafe: Target address**			Data type: UNSIGNED_16 Index: 10678_d = $29B6_h$
Setting of the PROFIsafe target address • Only required if the SM301 (E94AYAE) safety module is used. • A unique PROFIsafe target address must be assigned to each controller.			
Setting range (min. value | unit | max. value)			**Lenze setting**
1		65534	**1**
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☑ COM ☐ MOT			

C13899

Parameter | Name: **C13899 | Station address**			Data type: UNSIGNED_8 Index: 10676_d = $29B4_h$
Optional setting of the station address (instead of setting via DIP switches 1 ... 64) • The station address set here only becomes effective if the DIP switches 1 ... 64 have been set to OFF prior to mains switching. • The active station address is displayed under C13864. **Note:** A change of the station address will not be effective until the " Save parameter set" device command has been executed and another mains switching for the communication module/controller has been performed. ▸ Setting the station address (📖 31)			
Setting range (min. value | unit | max. value)			**Lenze setting**
3		126	**3**
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☑ COM ☐ MOT			

C13900

Parameter | Name: **C13900 | Firmware product type**	Data type: VISIBLE_STRING Index: 10675_d = $29B3_h$
Display of the product type (string with a length of 8 bytes) The following identification code is displayed: "E94AFCPM".	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13901

Parameter | Name: **C13901 | Firmware compilation date**	Data type: VISIBLE_STRING Index: 10674_d = $29B2_h$
Display of the compilation date of the firmware (string with a length of 20 bytes) The date ("MMM DD YYYY") and time ("hh:mm:ss") are displayed, e.g. "Mar 21 2005 12:31:21".	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13902

Parameter | Name: **C13902 | Firmware version**	Data type: VISIBLE_STRING Index: 10673_d = $29B1_h$
Display of the firmware version (string with a length of 5 bytes) Example: "01.21"	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C13920

Parameter | Name: **C13920 | Display: DIP switch setting**			Data type: UNSIGNED_8 Index: 10655_d = $299F_h$
Display of the current DIP switch setting • The displayed value corresponds to the sum of the individual DIP switch values 1 ... 64. • The active station address is displayed under C13864. ▸ Possible DIP switch settings (📖 31)			
Display area **(min. value | unit | max. value)**			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C13990

Parameter | Name: **C13990 | ICM statistic counter**	Data type: UNSIGNED_16 Index: 10585_d = 2959_h
Available from software version 01.21! **This code is for device-internal use only and must not be overwritten by the user!**	

14.3 Parameters of the communication module for slot MXI2

This chapter lists the parameters of the E94AYCPM communication module (PROFIBUS) for slot MXI2 of the Servo Drive 9400 in numerically ascending order.

C14231

Parameter | Name: **C14231 | Service code**	Data type: UNSIGNED_64 Index: $10344_d = 2868_h$
This code is for device-internal use only and must not be overwritten by the user!	

C14489

Parameter | Name: **C14489 | Service code**	Data type: UNSIGNED_8 Index: $10086_d = 2766_h$
This code is for device-internal use only and must not be overwritten by the user!	

C14531

Parameter | Name: **C14531 | Service code**	Data type: UNSIGNED_64 Index: $10044_d = 273C_h$
This code is for device-internal use only and must not be overwritten by the user!	

C14789

Parameter | Name: **C14789 | Service code**	Data type: UNSIGNED_8 Index: $9786_d = 263A_h$
This code is for device-internal use only and must not be overwritten by the user!	

C14850

Parameter | Name: **C14850 | All words to master**			Data type: UNSIGNED_16 Index: $9725_d = 25FD_h$
Display of the process data words transferred from the communication module to the PROFIBUS master. In the subcodes 1 ... 32, all process data words to the master are displayed. Only the configured process data words are valid.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C14850/1			
...			
C14850/32			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14851

Parameter | Name: **C14851 | All words from master**			Data type: UNSIGNED_16 Index: 9724_d = $25FC_h$
Display of the process data words transferred from the PROFIBUS master to the communication module. In the subcodes 1 ... 32, all process data words to the master are displayed. Only the configured process data words are valid.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C14851/1			
...			
C14851/32			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14852

Parameter | Name: **C14852 | All words to standard device**			Data type: UNSIGNED_16 Index: 9723_d = $25FB_h$
Display of the process data words 1 ... 32 transferred from the communication module to the standard device. In the subcodes 1 ... 32, all process data words from the communication module are displayed.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C14852/1			
...			
C14852/32			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14853

Parameter | Name: **C14853 | All words from standard device**			Data type: UNSIGNED_16 Index: 9722_d = $25FA_h$
Display of the process data words 1 ... 32 transferred from the standard device to the communication module. In the subcodes 1 ... 32, all process data words from the standard device are displayed.			
Display area (min. value | unit | max. value)			
0		65535	
Subcodes			**Info**
C14853/1			
...			
C14853/32			
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14854

Parameter | Name: **C14854 | COM word to standard device**	Data type: BITFIELD_32 Index: 9721_d = $25F9_h$
Available from software version 01.21! **This code is for device-internal use only and must not be overwritten by the user!**	

C14855

Parameter | Name: **C14855 | COM word from standard device**	Data type: BITFIELD_32 Index: 9720_d = $25F8_h$
Available from software version 01.21! **This code is for device-internal use only and must not be overwritten by the user!**	

C14860

Parameter | Name: **C14860 | Settings**				Data type: UNSIGNED_8 Index: 9715_d = $25F3_h$
Display of the current configuration data.				
Display area (min. value | unit | max. value)				
0		255		
Subcodes			**Info**	
C14860/1			PROFIsafe • 0: Not active • 1: Active	
C14860/2			Number of process data words (1 ... 32 words)	
C14860/3			DRIVECOM parameter data channel • 0: Not active • 1: Active	
C14860/4			Number of PROFIsafe words (0 or 4 words)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT				

C14861

Parameter | Name: **C14861 | Bus status**				Data type: UNSIGNED_16 Index: 9714_d = $25F2_h$
Bit-coded display of the current bus state. ▸ Querying the current bus status (📖 84)				
Display area (min. value | unit | max. value)				
0		65535		
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT				

C14862

Parameter | Name: **C14862 | Bus counter**				Data type: UNSIGNED_16 Index: 9713_d = $25F1_h$
When the maximum count value of 65535 is reached, the counter starts again with 0.				
Display area (min. value | unit | max. value)				
0		65535		
Subcodes			**Info**	
C14862/1			Data cycles per second	
C14862/2			Total data cycles	
C14862/3			Total parameterisation events	
C14862/4			Total configuration events	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT				

C14863

Parameter | Name: **C14863 | Baud rate**		Data type: UNSIGNED_8 Index: 9712_d = $25F0_h$
Display of the baud rate		
Selection list (read only)		
0	12.00 Mbps	
1	6.00 Mbps	
2	3.00 Mbps	
3	1.50 Mbps	
4	500.00 kbps	
5	187.50 kbps	
6	93.75 kbps	
7	45.45 kbps	
8	19.20 kbps	
9	9.60 kbps	
15	No baud rate detected	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14864

Parameter | Name: **C14864 | Active station address**			Data type: UNSIGNED_8 Index: 9711_d = $25EF_h$
Display of the active station address If all DIP switches 1 ... 64 are in the "OFF" position (Lenze setting), the station address set in C14899 becomes active and is displayed here after switching on. ▸ Setting the station address (📖 31)			
Display area (min. value | unit | max. value)			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14865

Parameter | Name: **C14865 | Display: Most recent PRM data**	Data type: OCTET_STRING Index: 9710_d = $25EE_h$
Display of the last parameter data sent by the PROFIBUS master with the "Set-Prm" telegram (ASCII string with 24 characters)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C14866

Parameter | Name: **C14866 | Display: Most recent CFG data**	Data type: OCTET_STRING Index: 9709_d = $25ED_h$
Display of the last configuration data sent by the PROFIBUS master with the "Chk-Cfg" telegram (ASCII string with 22 characters)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C14867

Parameter | Name: **C14867 | Display: Most recent diagnostic data**	Data type: OCTET_STRING Index: 9708_d = $25EC_h$
Display of the last diagnostic data sent to the PROFIBUS master (ASCII string with 16 characters) ▸ Advanced diagnostic message (📖 85)	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C14868

Parameter | Name: **C14868 | Display: DP state**		Data type: UNSIGNED_8 Index: 9707_d = $25EB_h$
Available from software version 01.21! Display of the DP states (Decentralized Peripherals) ▸ Short-time interruption of PROFIBUS communication (📖 79), see fig. [11-1]		
Selection list (read only)		
0	Waiting for parameterisation	
1	Waiting for configuration	
2	Data exchange	
3	n/a	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14880

Parameter | Name: **C14880 | Reaction on communication fault**		Data type: UNSIGNED_8 Index: 9695_d = $25DF_h$
Monitoring response to a communication fault on the PROFIBUS A change in the monitoring response is effective immediately. ▸ Permanent interruption of PROFIBUS communication (📖 78)		
Selection list		
0	No response	
1	Error	
3	Quick stop by trouble	
4	Warning locked	
6	Information	
Subcodes	**Lenze setting**	**Info**
C14880/1	0: No response	The response set here for the "Watchdog/Data Exchange" monitoring function is executed if the bus station ... • does not receive a message from the master within the watchdog monitoring time (displayed in C14882/1) if there is an active connection. • detects that it is no longer in the "DATA_EXCHANGE" status. Please see also the notes given under C14881.
C14880/2	0: No response	The response set here for the "DPV1 MSAC2" monitoring is executed if the bus station does not receive a "DPV1 MSAC2" message from the master within the monitoring time (displayed in C14882/2) if there is an active connection) and the MSAC2 connection is stopped by the slave. **Note:** We recommend only setting "information" as response so that no drive-relevant response is executed.
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14881

Parameter | Name: **C14881 | Monitoring time: Data exchange**			Data type: UNSIGNED_16 Index: 9694_d = $25DE_h$
If the "Data Exchange" state is exited, the response parameterised under C14880/1 is carried out when the monitoring time for data exchange set here has expired. • A value of "65535" in this code deactivates the monitoring function. • A change in monitoring is effective immediately. • The value set here for the monitoring time must be smaller than the watchdog monitoring time (C14882/1). ▸ Permanent interruption of PROFIBUS communication (📖 78)			
Setting range (min. value | unit | max. value)			**Lenze setting**
0	ms	65535	**65535 ms**
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14882

Parameter | Name: **C14882 | Monitoring time: Watchdog**			Data type: UNSIGNED_32 Index: 9693_d = $25DD_h$
Display of the watchdog monitoring time determined by the PROFIBUS master • A change in the watchdog monitoring time is immediately effective. • Monitoring starts with the receipt of the first telegram. • When a value of "0" is displayed, the monitoring function is deactivated. ▸ Permanent interruption of PROFIBUS communication (📖 78)			
Display area (min. value | unit | max. value)			
0	ms	4294967295	
Subcodes			**Info**
C14882/1			Watchdog monitoring time
C14882/2			DP-V1 MSC2
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14885

Parameter | Name: **C14885 | Clear process data**		Data type: UNSIGNED_8 Index: 9690_d = $25DA_h$
Selection of the process data which the controller will process in the event of a PROFIBUS failure in order to maintain internal communication.		
Selection list (Lenze setting printed in bold)		
0	Use of most recent master PDOs	
1	**PDOs are set to the value '0'**	
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT		

C14886

Parameter | Name: **C14886 | Set ext. diagnostic bit upon**	Data type: BITFIELD_8 Index: 9689_d = $25D9_h$
Bit-coded selection of the error responses in the standard device causing the external diagnostic bit ("diag bit") to be set (see PROFIBUS specification; bit 3 of byte 1 of the DP diagnostic messages). • The diagnostic bit is sent to the PROFIBUS master where it is evaluated separately. • The diagnostic bit is always set when a system error occurs. • The Lenze setting "0" means that the diagnostic bit is not set for the following error responses. • An advanced diagnostic message is always sent.	

Value is bit-coded:		
Bit 0	Error	
Bit 1	Trouble	
Bit 2	Quick stop by trouble	
Bit 3	Warning locked	
Bit 4	Warning	
Bit 5	Reserved	
Bit 6	Reserved	
Bit 7	Reserved	

☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT

C14887

Parameter | Name: **C14887 | Suppress signalling diag. mess. upon**	Data type: BITFIELD_8 Index: 9688_d = $25D8_h$
Available from software version 01.20! Selection of the error responses not causing a diagnostic request to the PROFIBUS master. A diagnostic request is always signalled if a system error or an error message from the SM301 (E94AYAE) safety module occurs. The Lenze setting "0" means that for each of the following error responses a diagnostic request is signalled.	

Value is bit-coded:		**Info**
Bit 0	Error	
Bit 1	Trouble	
Bit 2	Quick stop by trouble	
Bit 3	Warning locked	
Bit 4	Warning	
Bit 5	Reserved	
Bit 6	Reserved	
Bit 7	Error message 0x00c88130	Available from software version 01.21! Selective suppression of the error message "Available connection to 9400 standard device lost"

☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT

C14897

Parameter | Name: **C14897 | PROFIsafe: Target address**			Data type: UNSIGNED_16 Index: 9678_d = $25CE_h$
Setting of the PROFIsafe target address • Only required if the SM301 (E94AYAE) safety module is used. • A unique PROFIsafe target address must be assigned to each controller.			
Einstellbereich (min. Wert | Einheit | max. Wert)			**Lenze setting**
1		65534	**1**
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☑ COM ☐ MOT			

C14899

Parameter | Name: **C14899 | Station address**			Data type: UNSIGNED_8 Index: 9676_d = $25CC_h$
Optional setting of the station address (instead of setting via DIP switches 1 ... 64) • The station address set here only becomes effective if the DIP switches 1 ... 64 have been set to OFF prior to mains switching. • The active station address is displayed under C14864. **Note:** A change of the station address will not be effective until the " Save parameter set" device command has been executed and another mains switching for the communication module/controller has been performed. ▸ Setting the station address (📖 31)			
Einstellbereich (min. Wert | Einheit | max. Wert)			**Lenze setting**
3		126	**3**
☑ Read access ☑ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☑ COM ☐ MOT			

C14900

Parameter | Name: **C14900 | Firmware product type**	Data type: VISIBLE_STRING Index: 9675_d = $25CB_h$
Display of the product type (string with a length of 8 bytes) The following identification code is displayed: "E94AFCPM".	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C14901

Parameter | Name: **C14901 | Firmware compilation date**	Data type: VISIBLE_STRING Index: 9674_d = $25CA_h$
Display of the compilation date of the firmware (string with a length of 20 bytes) The date ("MMM DD YYYY") and time ("hh:mm:ss") are displayed, e.g. "Mar 21 2005 12:31:21".	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C14902

Parameter | Name: **C14902 | Firmware version**	Data type: VISIBLE_STRING Index: 9673_d = $25C9_h$
Display of the firmware version (string with a length of 5 bytes) Example: "01.21"	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT	

C14920

Parameter | Name: **C14920 | Display: DIP switch setting**			Data type: UNSIGNED_8 Index: 9655_d = $25B7_h$
Display of the current DIP switch setting • The displayed value corresponds to the sum of the individual DIP switch values 1 ... 64. • The active station address is displayed under C14864. • ▸ Possible DIP switch settings (📖 31)			
Display area (min. value | unit | max. value)			
0		255	
☑ Read access ☐ Write access ☐ CINH ☐ PLC-STOP ☐ No transfer ☐ PDO_MAP_RX ☐ PDO_MAP_TX ☐ COM ☐ MOT			

C14990

Parameter | Name: **C14990 | ICM statistic counter**	Data type: UNSIGNED_16 Index: 9585_d = 2571_h
Available from software version 01.21! **This code is for device-internal use only and must not be overwritten by the user!**	

14.4 Table of attributes

The table of attributes contains information required for communication 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 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	

Table of attributes

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C13850	All words to master	10725	29E5	A	32	UNSIGNED_16	1	☑		
C13851	All words from master	10724	29E4	A	32	UNSIGNED_16	1	☑		
C13852	All words to standard device	10723	29E3	A	32	UNSIGNED_16	1	☑		
C13853	All words from standard device	10722	29E2	A	32	UNSIGNED_16	1	☑		
C13860	Settings	10715	29DB	A	4	UNSIGNED_8	1	☑		
C13861	Bus status	10714	29DA	E	1	UNSIGNED_16	1	☑		
C13862	Bus counter	10713	29D9	A	4	UNSIGNED_16	1	☑		
C13863	Baud rate	10712	29D8	E	1	UNSIGNED_8	1	☑		
C13864	Active station address	10711	29D7	E	1	UNSIGNED_8	1	☑		
C13865	Display: Most recent PRM data	10710	29D6	E	1	OCTET_STRING		☑		
C13866	Display: Most recent CFG data	10709	29D5	E	1	OCTET_STRING		☑		
C13867	Display: Most recent diagnostic data	10708	29D4	E	1	OCTET_STRING		☑		
C13868	Display: DP state	10707	29D3	E	1	UNSIGNED_8	1	☑		
C13880	Reaction on communication fault	10695	29C7	A	2	UNSIGNED_8	1	☑	☑	
C13881	Monitoring time: Data exchange	10694	29C6	E	1	UNSIGNED_16	1	☑	☑	
C13882	Monitoring time: Watchdog	10693	29C5	A	2	UNSIGNED_32	1	☑		
C13885	Clear process data	10690	29C2	E	1	UNSIGNED_8	1	☑	☑	
C13886	Set ext. diagnostic bit upon	10689	29C1	E	1	BITFIELD_8		☑	☑	
C13887	Suppress signalling diag. mess. upon	10688	29C0	E	1	BITFIELD_8		☑	☑	
C13897	PROFIsafe: Target address	10678	29B6	E	1	UNSIGNED_16	1	☑	☑	
C13899	Station address	10676	29B4	E	1	UNSIGNED_8	1	☑	☑	
C13900	Firmware product type	10675	29B3	E	1	VISIBLE_STRING		☑		
C13901	Firmware compilation date	10674	29B2	E	1	VISIBLE_STRING		☑		
C13902	Firmware version	10673	29B1	E	1	VISIBLE_STRING		☑		
C13920	Display: DIP switch setting	10655	299F	E	1	UNSIGNED_8	1	☑		
C14850	All words to master	9725	25FD	A	32	UNSIGNED_16	1	☑		
C14851	All words from master	9724	25FC	A	32	UNSIGNED_16	1	☑		
C14852	All words to standard device	9723	25FB	A	32	UNSIGNED_16	1	☑		
C14853	All words from standard device	9722	25FA	A	32	UNSIGNED_16	1	☑		
C14860	Settings	9715	25F3	A	4	UNSIGNED_8	1	☑		
C14861	Bus status	9714	25F2	E	1	UNSIGNED_16	1	☑		
C14862	Bus counter	9713	25F1	A	4	UNSIGNED_16	1	☑		
C14863	Baud rate	9712	25F0	E	1	UNSIGNED_8	1	☑		
C14864	Active station address	9711	25EF	E	1	UNSIGNED_8	1	☑		
C14865	Display: Most recent PRM data	9710	25EE	E	1	OCTET_STRING		☑		
C14866	Display: Most recent CFG data	9709	25ED	E	1	OCTET_STRING		☑		
C14867	Display: Most recent diagnostic data	9708	25EC	E	1	OCTET_STRING		☑		
C14868	Display: DP state	9707	25EB	E	1	UNSIGNED_8	1	☑		
C14880	Reaction on communication fault	9695	25DF	A	2	UNSIGNED_8	1	☑	☑	
C14881	Monitoring time: Data exchange	9694	25DE	E	1	UNSIGNED_16	1	☑	☑	
C14882	Monitoring time: Watchdog	9693	25DD	A	2	UNSIGNED_32	1	☑		
C14885	Clear process data	9690	25DA	E	1	UNSIGNED_8	1	☑	☑	
C14886	Set ext. diagnostic bit upon	9689	25D9	E	1	BITFIELD_8		☑	☑	
C14887	Suppress signalling diag. mess. upon	9688	25D8	E	1	BITFIELD_8		☑	☑	
C14897	PROFIsafe: Target address	9678	25CE	E	1	UNSIGNED_16	1	☑	☑	
C14899	Station address	9676	25CC	E	1	UNSIGNED_8	1	☑	☑	
C14900	Firmware product type	9675	25CB	E	1	VISIBLE_STRING		☑		
C14901	Firmware compilation date	9674	25CA	E	1	VISIBLE_STRING		☑		

Code	Name	Index		Data				Access		
		dec	hex	DS	DA	DT	Factor	R	W	CINH
C14902	Firmware version	9673	25C9	E	1	VISIBLE_STRING		☑		
C14920	Display: DIP switch setting	9655	25B7	E	1	UNSIGNED_8	1	☑		

14.5 Implemented PROFIdrive objects (DP-V1)

I-918

Index | Name: **0x918 | Display of station address**			Data type: U16
Display of the station address set			
Display area (min. value | unit | max. value)			
1		126	
☑ Read access ☐ Write access			

I-963

Index | Name: **0x963 | Baud rate**		Data type: U16
Display of the PROFIBUS baud rate		
Selection list (read only)		
0	9.6 kbps	
1	19.2 kbps	
2	93.75 kbps	
3	187.5 kbps	
4	500 kbps	
6	1.5 Mbps	
7	3 Mbps	
8	6 Mbps	
9	12 Mbps	
10	31.25 kbps	
11	45.45 kbps	
☑ Read access ☐ Write access		

I-964

Index | Name: **0x964 | Device identification**		Data type: U16
Display of identification data		
Subindex	**Display**	**Info**
0x964/0	262	Manufacturer: Lenze
0x964/1	9400	Device type
0x964/2	xxyy	Software version, e.g. 0111 (V 1.20)
0x964/3	yyyy	Firmware date (year), e.g. 2006
0x964/4	ddmm	Firmware date (day/month), e.g. 0506 (5th June)
☑ Read access ☐ Write access		

I-974

Index | Name: **0x974 | Maximum time per DPV1 parameter access**		Data type: U16
Display of access statistics		
Subindex	**Display**	**Info**
0x974/0	240 bytes	Maximum block length
0x974/1	40	Maximum number of parameter accesses
0x974/2	0	Maximum time per access
☑ Read access ☐ Write access		

15 DIP switch positions for setting the station address

The station address results from the sum of the binary valencies of switches 1 ... 64.

The following table shows the switch positions for the valid address range 1 ... 126.

Setting the station 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**

Station address	DIP switch						
	64	32	16	8	4	2	1
38	OFF	**ON**	OFF	OFF	**ON**	**ON**	OFF
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

Station address	DIP switch						
	64	**32**	**16**	**8**	**4**	**2**	**1**
81	**ON**	OFF	**ON**	OFF	OFF	OFF	**ON**
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**

Station address	DIP switch						
	64	**32**	**16**	**8**	**4**	**2**	**1**
124	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	OFF
125	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	**ON**
126	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	OFF

FEEDBACK

Your opinion is important to us

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

If you have suggestions for improvement, please e-mail us to:

feedback-docu@Lenze.de

Thank you for your support.

Your Lenze documentation team

E94AYCPM communication module (PROFIBUS®) · Communication Manual · EDS94AYCPM · 13422195 · DMS 12.0 EN · 11/2012 · TD17

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

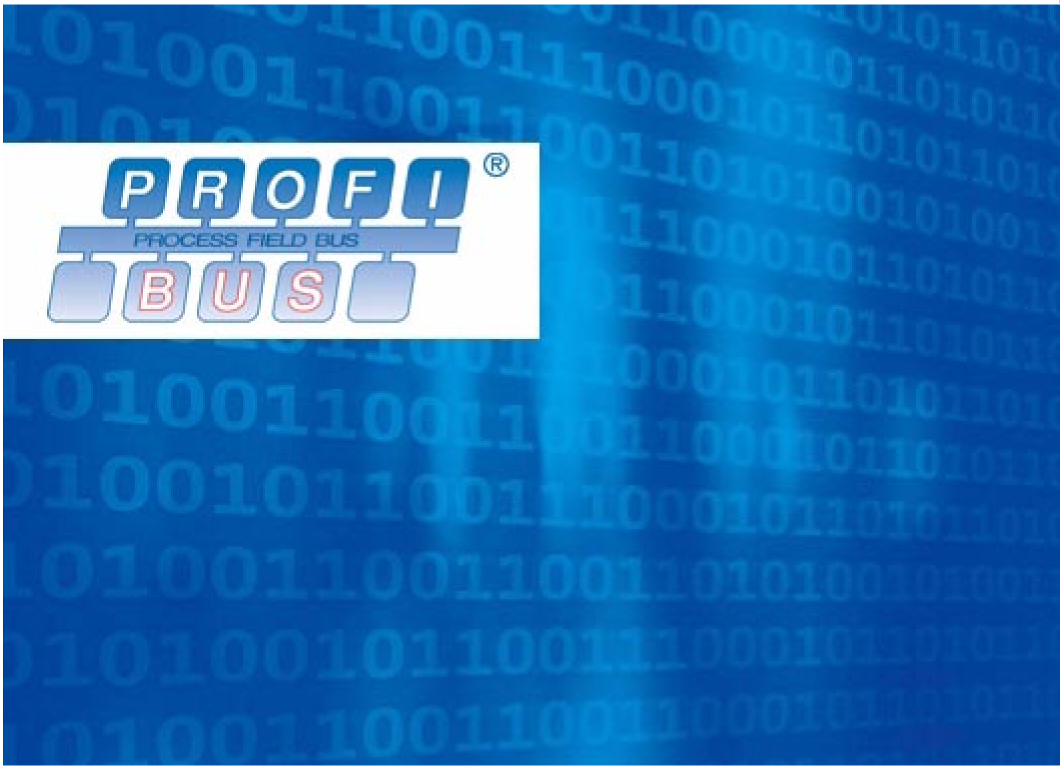
Lenze Service GmbH
Breslauer Straße 3
D-32699 Extertal
Germany
00 80 00 / 24 4 68 77 (24 h helpline)
+49 (0)51 54 / 82-11 12
Service@Lenze.de

KHBPBPCBAUTO
13383678

L-force *Controls*

Communication manual

PC-based Automation



PROFIBUS control technology

Commissioning & configuration

Lenze

Contents

1 About this documentation

This documentation ...

- contains detailed information on how to commission, configure, and diagnose the PROFIBUS bus system within the Lenze control technology.
- belongs to the "PC-based Automation" manual collection which consists of the following documentation:

Documentation	Subject
System manuals "PC-based automation"	• Control technology - *System structure & configuration* • Control technology - *System structure & components*
Communication manuals "PC-based automation"	• CANopen control technology • PROFIBUS control technology • EtherCAT control technology
(Software) Manual "PC-based automation"	• Industrial PC - *Parameter setting & configuration*
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 - *Parameter setting & configuration* • »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 Inverter Drives 8400
🗎 MAs for Servo Drives 9400
🗎 MA EPM-Txxx (I/O system IP20)
🗎 MA EPM-Sxxx (I/O system 1000)
🗎 MA 8200 vector
🗎 Wiring according to EMC, 8200 vector
🗎 MAs for the ECS servo system
🗎 MA MC-CAN2 communication card
🗎 MA MC-ETC communication card
🗎 MA MC-ETH communication card
🗎 MA MC-PBM communication card
🗎 MA MC-PBS communication card
🗎 MA MC-MPI communication card
🗎 MAs for communication modules

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

Legend:

🗎 Printed documentation

☐ Online help/PDF

Abbreviations used:

SHB	System Manual
BA	Operating Instructions
MA	Mounting Instructions
SW	Software manual
KHB	Communication manual

Tip!

Documentation and software updates for Lenze products can be found in the Download area at:

http://www.Lenze.com

Target group

This documentation is intended for all persons who plan, install, commission, and maintain the networking of devices in the field of control technology.

1.1 Document history

Material no.	Version			Description
13294525	1.0	05/2009	TD17	First edition
13319345	2.0	10/2009	TD17	General revision
13369327	2.1	01/2011	TD17	Update for control technology release 2.5
13383678	2.2	07/2011	TD17	Chapter Error messages if communication card MC-PBM is not available (📖 62) 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.

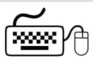
Your Lenze documentation team

1.2 Conventions used

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

Type of information		Highlighting	Examples/notes
Spelling of numbers			
	Decimal separator	Point	The decimal point is always used. For example: 1234.56
Text			
	Version information	Blue text colour	Information that is only 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	*Italics*	The *Message window*... / The *Options* dialog box...
	Variable identifier		By setting *bEnable* to TRUE...
	Control element	**Bold**	The **OK** button... / the **Copy** command... / the **Characteristics** tab... / the **Name** input field...
	Sequence of menu commands		If the execution of a function requires several commands in a row, the individual commands are separated by an arrow: Select **File→Open** to ...
	Shortcut	**<Bold>**	Use **<F1>** to open the online help.
			If a key combination is required for a command, a "+" is placed between the key identifiers: With **<Shift>+<ESC>**...

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

	Keyword	**`Courier bold`**	
	Hyperlink	<u>Underlined</u>	Optically highlighted reference to another topic. It is activated with a mouse-click in this documentation.
Symbols			
	Page reference	(📖 8)	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
GSE	Device data base file (device description for PROFIBUS devices)

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 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 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!

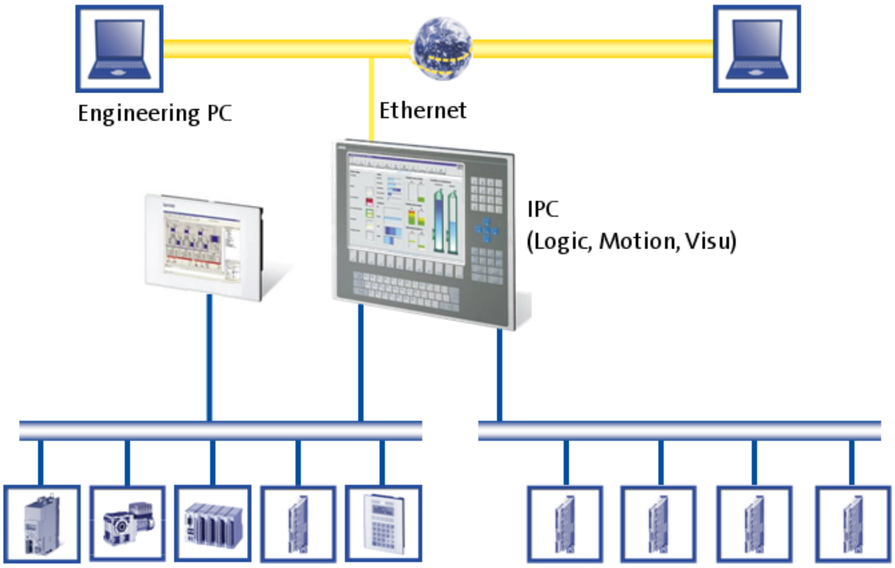
3 The "PC-based automation" system

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 controls the logic and motion functionalities by means of the runtime software.
  - The IPC communicates with the field devices via the fieldbus.
  - The IPCs are available in different designs.

Note!

Moreover, the HMI series Z EL 1xx PLC belongs to the "PC-based Automation" system. These devices differ considerably from the industrial PCs in performance and various other details. However, the devices of the HMI series EL 1xx PLC are able to fulfil smaller control functions.

- Engineering tools for the engineering PC
  - The engineering PC communicates with the IPC via Ethernet.
  - Different engineering tools (📖 22) serve to configure and parameterise the system.
- Fieldbuses
- Field devices

4 The Lenze control system with PROFIBUS

Note!

In the Lenze control system, only the PROFIBUS master functionality (logic bus) is supported.

This chapter provides basic information about ...

- the PROFIBUS bus system in the Lenze control system;
- the structure of the Lenze control system with the PROFIBUS master;
- the components required for PROFIBUS communication.

4.1 Brief description of PROFIBUS

Today, PROFIBUS is the most commonly used fieldbus system. Because it comes with the widest range of various field devices, PROFIBUS is occasionally prioritised over more modern bus systems. Due to the low bandwidth and synchronisation mechanisms, PROFIBUS is only provided as logic bus within "PC-based Automation".

We recommend to use PROFIBUS for the following applications:

- Control of system parts that have already been automated with PROFIBUS and another control system.
- Use of field devices that are not available for other bus systems (e.g. CANopen or EtherCAT).
- Combination of PROFIBUS as logic bus with CANopen as motion bus (📖 18)

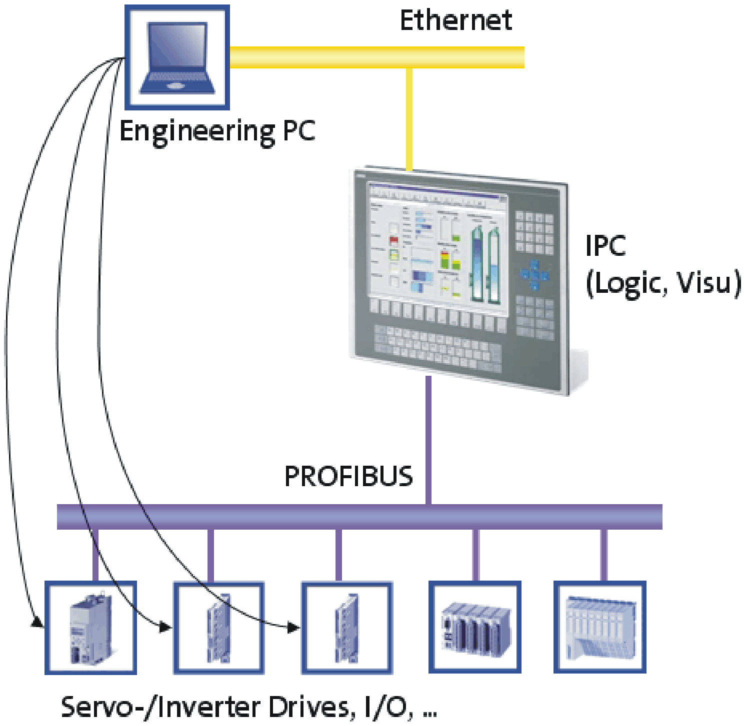
Tip!

Detailed information on PROFIBUS can be found on the internet page of the PROFIBUS user organisation:

www.profibus.com

4.1.1 Structure of the PROFIBUS system

Basic structure



Physical structure

- The industrial PC (IPC) is the PROFIBUS master. It can communicate with one or several stations (slaves).
- PROFIBUS has an internal line topology (without repeater) or a tree topology (with repeater).
  Basic wiring of PROFIBUS (17)
- The PROFIBUS network must be terminated at the first and last station. The bus terminating resistor is integrated in the bus connector and is activated by a switch.

Parameter setting

The PROFIBUS stations can be parameterised in different ways:

- Direct access of the engineering software (from the engineering PC) to the slave field device. Depending on the device type, via the following interfaces:
  - CAN
  - Ethernet
  - LECOM
  - Diagnostic interface
- Parameter transfer from the control (the parameter transfer must be programmed manually)
  - PROFIBUS is exclusively configured with the »PLC Designer«:
    Structure of the control configuration and creation of the PLC program
  - All PROFIBUS slaves can be switched on via device data base files (GSE files).

Note!

The "IPC as gateway" function is not available in connection with PROFIBUS.

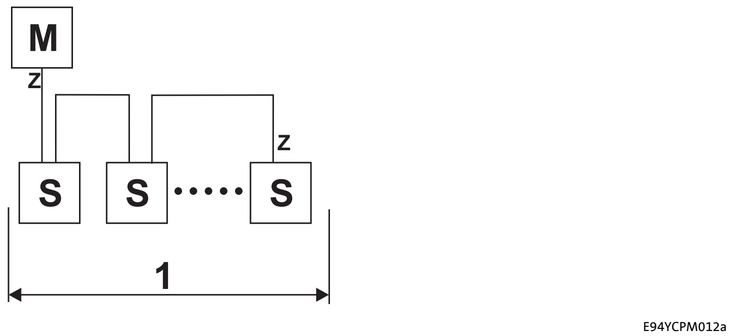
Thus, "Going online" with the »Engineer« or the »GDC« via the IPC as gateway is not possible.

4.1.2 Basic wiring of PROFIBUS

The following examples show two simple PROFIBUS networks.

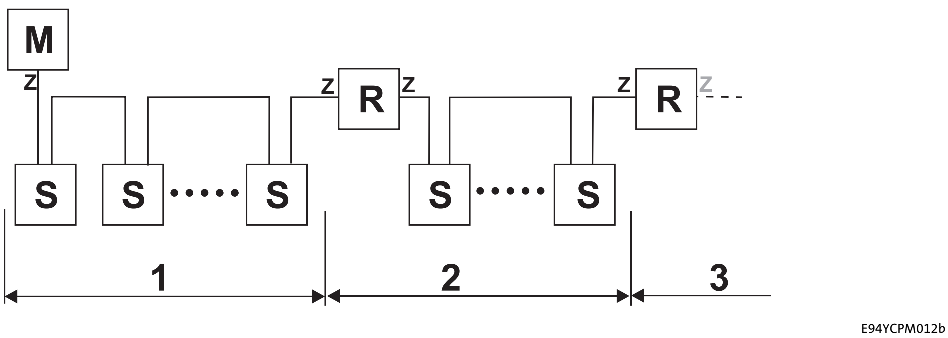
Each segment of the network must be terminated at both ends. The bus terminators of PROFIBUS are marked with a "Z" in each of the following examples.

In a PROFIBUS network of only one segment, the PROFIBUS master (M) with an integrated bus terminator starts the segment, and the connector of the last device (S) with the bus terminator ends it.



[4-1] PROFIBUS network with one segment

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



[4-2] PROFIBUS network with repeater

If no repeater is used at the end of the segment, the bus terminator in the plug of the last device must be activated.

Note!

Repeaters do not have a station address. When calculating the maximum number of stations, they reduce the number of stations by 1 on each side of the segment.

Repeaters can be used to build up line and tree topologies. The maximum total bus system expansion depends on the baud rate used and the number of repeaters.

4.1.3 Combination with other bus systems

The PROFIBUS bus system can be combined with CANopen. 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 PROFIBUS (as logic bus). The bus systems are synchronised in the control system.

Note!

- Mixed operation is only possible with industrial PCs which have two additional slots for communication cards. Mixed operation is not possible with the "Command Station".
- In release 2.5, PROFIBUS cannot be combined with EtherCAT.
- In the control configuration the PROFIBUS master must be arranged in the first position – in front of the CANopen motion stations.

4.1.4 Field devices

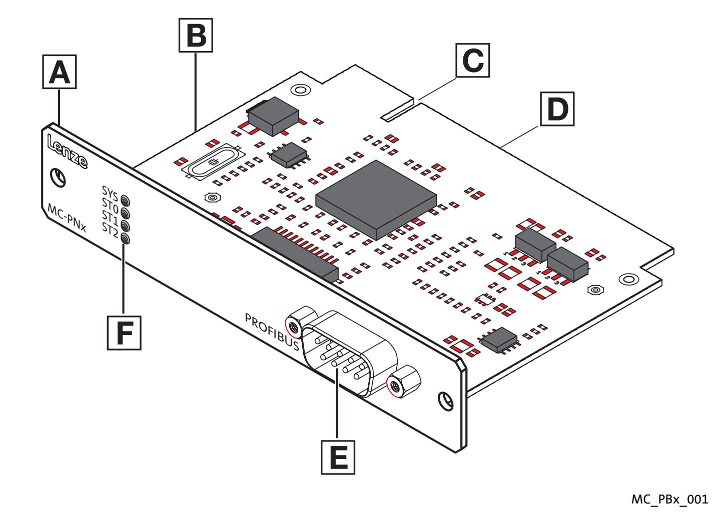
The Lenze control system supports the following logic components for PROFIBUS:

Standard devices		PROFIBUS communication cards/modules
Industrial PCs	EL x1xx PLC	MC-PBM (PROFIBUS master)
	EL x8xx	
	CS x8xx	
	CPC x8xx	
Servo Drives 9400	HighLine	E94AYCPM
	PLC	
Inverter Drives 8400	BaseLine	E84AYCPM
	StateLine	
	HighLine	
	TopLine	
I/O system IP20	EPM-T120	
	EPM-T121	
I/O system 1000	EPM-S120 (*in preparation*)	
Frequency inverter	8200 vector	E82ZAFPCxxx
ECS servo system	ECSxE	EMF2133IB
	ECSXS (Speed & Torque)	
	ECSxP (Posi & Shaft)	
	ECSxA (Application)	

4.2 PROFIBUS hardware for the industrial PC

MC-PBM communication card

The MC-PBM communication card is a plug-in card for connecting an industrial PC as PROFIBUS master to a PROFIBUS network.



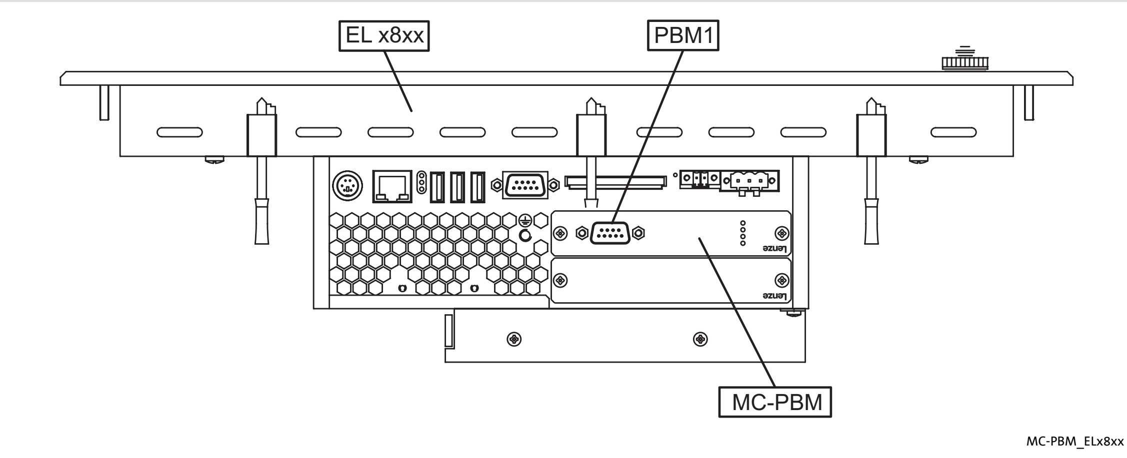
A Front panel
B Printed circuit board
C Coding
D Terminal for industrial PC
E Connection of PROFIBUS (SUB-D, 9-pole plug) (📖 20)
F LED status displays (📖 63)

▸ Technical data of the MC-PBM communication card (📖 20)

Possible applications

The MC-PBM communication card can be plugged into slot 1 and slot 2 of the industrial PC. Several PROFIBUS communication cards can be used for each industrial PC.

Example: Industrial PC EL x8xx with MC-PBM in slot 1



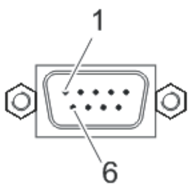
Legend	
EL x8xx	Industrial PC of the EL x8xx series
PBM1	PROFIBUS connection
MC-PBM	PROFIBUS master communication card

5 Technical data

5.1 Technical data of the MC-PBM communication card

Field	Values
Protocol	PROFIBUS-DP (V0, V1), ISO 7498
Communication medium	RS485
Network topology	Line terminated on both sides (without repeater) / tree (with repeater) • Termination with Sub-D plug
Type within the network	Master
Max. number of stations per segment	63
Max. number of stations per network	128
Baud rate	See chapter "Baud rate / bus cable length" (📖 21)
Bus length	
Connection	SUB-D, 9-pole plug

Connection of PROFIBUS (SUB-D, 9-pole plug)

View	Pin	Assignment	Description
	1	free	-
	2	free	-
	3	RxD/TxD-P	Data line B (received data/transmitted data plus)
	4	RTS	Request To Send (received data / transmitted data, no differential signal)
	5	M5V2	Data ground (ground to 5 V)
	6	P5V2	5 V DC / 30 mA (bus termination)
	7	free	-
	8	RxD/TxD-N	Data line A (received data / transmitted data minus)
	9	free	-

5.2 Bus cable specification

Please follow the specifications of the PROFIBUS user organisation for bus cables.

Field	Values
Cable resistance	135 ... 165 Ω/km, (f = 3 ... 20 MHz)
Capacitance per unit length	≤ 30 nF/km
Loop resistance	< 110 Ω/km
Core diameter	> 0.64 mm
Core cross-section	> 0.34 mm^2
Cores	Twisted in pairs, insulated and shielded

Baud rate / bus cable length

Adapt the baud rate to the length of the bus cable.

Baud rate [kbps]	Max. bus length [m]
9.6 ... 93.75	1200
187.5	1000
500	200
1500	200
3000 ... 12000	100

Note!

The baud rate depending of the data volume, cycle time, and number of stations should only be selected as high as required for the application.

6 Commissioning of PROFIBUS

This chapter provides information about how to commission the Lenze control system with PROFIBUS.

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.

6.1 Overview of the commissioning steps

The main commissioning steps are listed in the following table:

Step	Action	Software to be used
1.	Planning the bus topology (📖 23)	
2.	Installing field devices (📖 23)	
3.	Creating a project folder (📖 24)	
4.	Storing the device data base files (GSE) (📖 24)	
5.	Commissioning of field devices (📖 25)	»Engineer« »GDC«
6.	Creating a PLC program (📖 26)	»PLC Designer«
7.	Configuring the PROFIBUS master (📖 29)	»PLC Designer«
8.	Configuring the PROFIBUS slave (📖 33)	»PLC Designer«
9.	Compiling project data (📖 36)	»PLC Designer«
10.	Logging on to the control system with the »PLC Designer« (📖 36)	»PLC Designer«
11.	Loading and starting the PLC program (📖 36)	»PLC Designer«

6.2 Detailed commissioning steps

The individual commissioning steps are described in the following sections.

Follow the 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.

6.2.1 Planning the bus topology

Before you start to set up a PROFIBUS network, first create a plan of your network.

Note!

Observe the connection between the bus cable length and baud rate.

▸ Baud rate / bus cable length (📖 21)

How to plan the bus topology for your configuration:

1. Create an overview screen of the planned PROFIBUS network with all field devices to be implemented.
2. Start with the industrial PC (master).
3. Add the other field devices (slaves) below.

6.2.2 Installing field devices

For the installation of a field device, follow the mounting instructions for the respective device.

Activation of the bus terminating resistor

The PROFIBUS network must be terminated at the first and last station. The bus terminating resistor is integrated in the bus connector and is activated by a switch.

Note!

If you want to disconnect individual bus stations, ensure that the bus terminators at the cable ends remain active.

The bus termination is not active anymore if e.g. in the event of service the plug has been removed or the module supply has been switched off.

6.2.3 Creating a project folder

Create a project folder on the engineering PC.

Use this project folder to store the below data generated in the different project configuration steps:

- 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.

6.2.4 Storing the device data base files (GSE)

The data of the bus pheripherals required for the master control are stored in a GSE file. This file is required for programming the control.

Current GSE files (e.g. for the I/O system IP20 or for Lenze PROFIBUS communication modules) can be downloaded in the Download area at http://www.Lenze.com.

Store the required GSE files in the »PLC Designer« target directory:

- C:\...\Lenze\PlcDesigner\Targets\L-force_Logic_x800_V8\PlcConfPCMatic
- C:\...\Lenze\PlcDesigner\Targets\L-force_Motion_x800_V8\PlcConfPCMaticMotion

Note!

PROFIBUS can be used in motion and logic systems. In the Lenze control system, PROFIBUS is the logic bus. Only use PROFIBUS in a motion system if an additional motion bus (e.g. CAN) is used.

▸ CANopen with PROFIBUS (📖 37)

Tip!

The GSE files can also be stored via the »PLC Designer« with the menu command **Extras→Add configuration file**.

6.2.5 Commissioning of field devices

Parameterise the Lenze field devices connected to PROFIBUS either with the »Engineer« or with »GDC«, depending on the device.

PROFIBUS is exclusively configured with the »PLC Designer«.

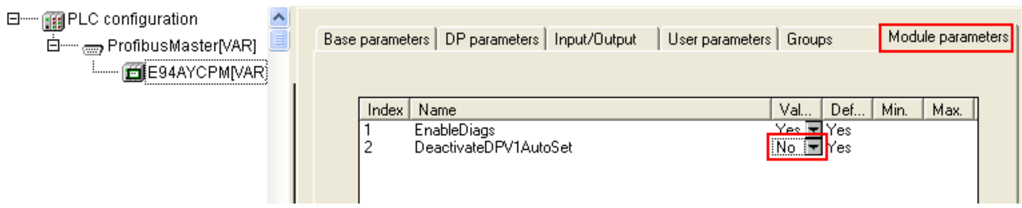
Observe the information with regard to commissioning in the documentation for the field devices.

Tip!

We recommend to commission each field device individually and then integrate them into the PLC program.

Servo Drives 9400 HighLine

During commissioning of a Servo Drive 9400 HighLine, the module parameter **DeactivateDPV1AutoSet** must be set to the value **"No"**:



6.2.6 Configuration in the »PLC Designer«

The »PLC Designer« serves to map the field device topology in the control configuration.

Tip!

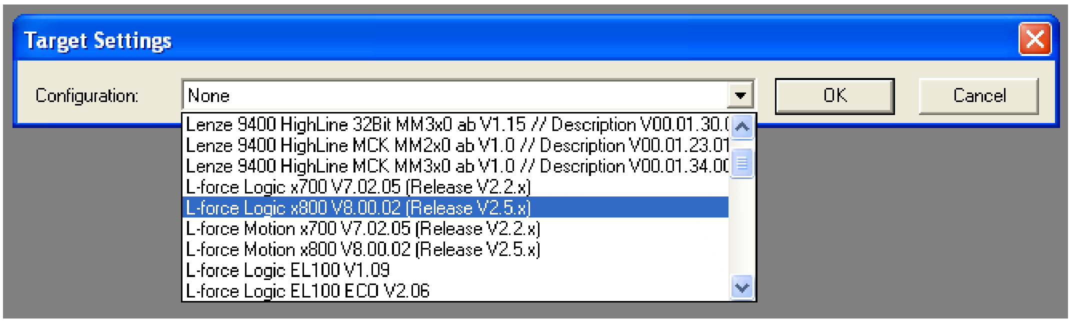
The »PLC Designer« serves to configure PROFIBUS stations and nodes on other fieldbus systems.

▸ CANopen with PROFIBUS (📖 37)

6.2.6.1 Creating a PLC program

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	Use with IPC series			
	EL x8xx	CS x8xx	CPC x8xx	EL 1xx PLC
L-force Logic x800 V8.xx.xx	●	●	●	-
L-force Logic EL1xx V1.xx	-	-	-	●

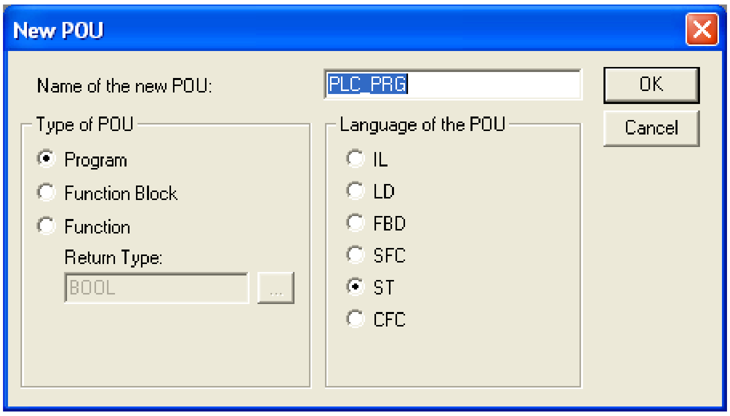
The target systems of release 2.2/2.3 (L-force Logic x700) can also be used for device series **EL x8xx**, **CS x8xx**, and **CPC x8xx**.

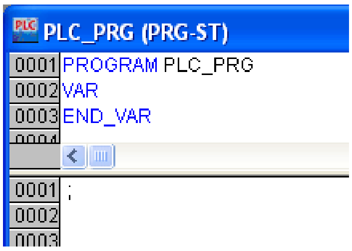
3. Confirm the configuration of the target system setting by clicking the **OK** button.

4. Create a block:

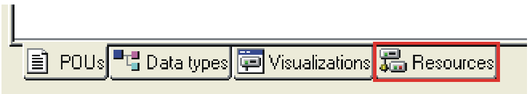
Note!

The block must contain at least one instruction to function properly.

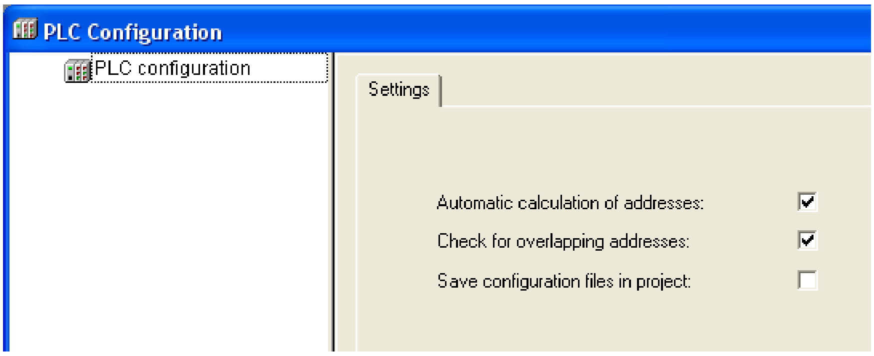




5. Create the control configuration:
    - Open the *Resources* dialog box:



    - 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!

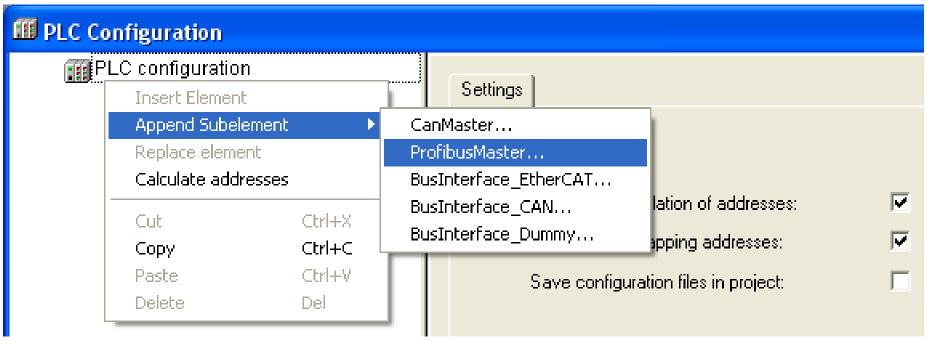
We recommend to keep the standard setting. In the case of a manual address allocation, you must ensure that each object address is **non-ambiguous** in the entire control configuration.

Detailed information on this can be found in the documentation/online help of the »PLC Designer«.

6.2.6.2 Configuring the PROFIBUS master

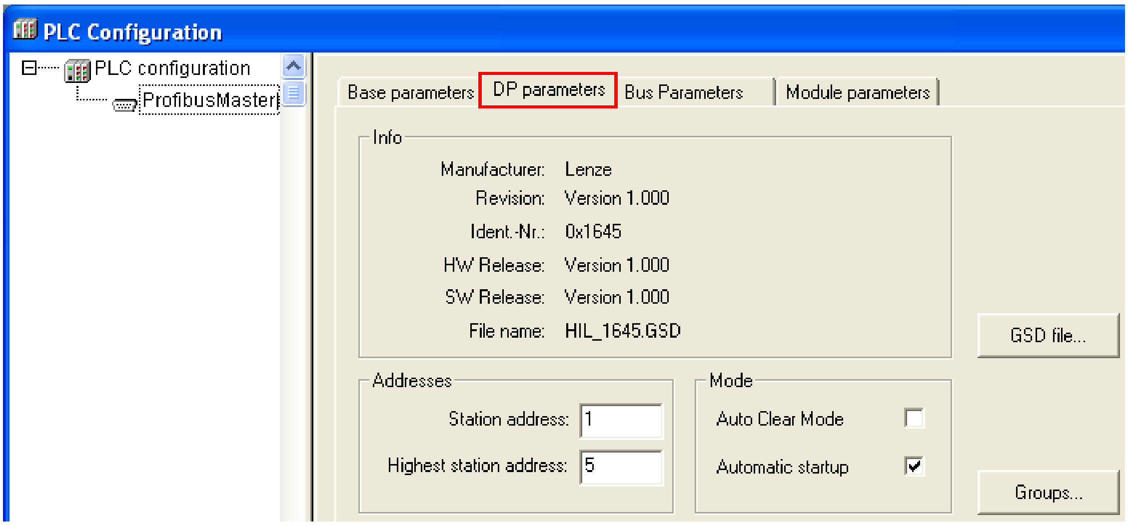
How to configure the PROFIBUS master:

1. Add the bus interface to the PLC configuration:

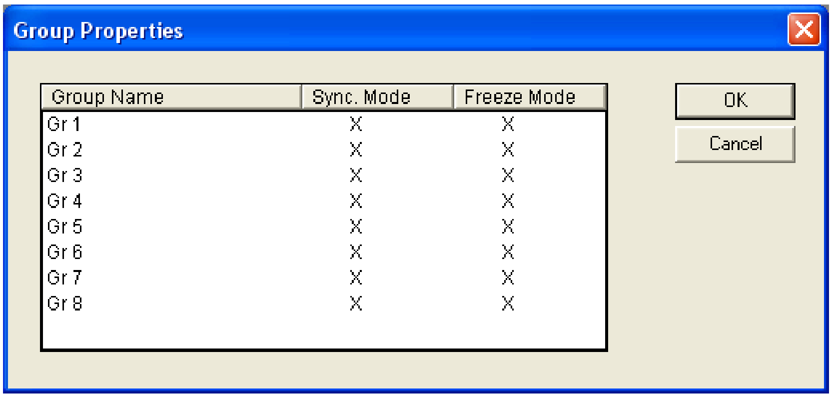


The "ProfibusMaster" subelement represents the PROFIBUS interface of the IPC to which the logic bus is connected.

2. Set DP parameters for the PROFIBUS master:



- The standard setting of the PROFIBUS master station address is '1'. Only change the setting if the address is to deviate from '1'.
- The "highest station address" is incremented automatically with each added slave. Thus the value should not be changed manually.
- The **GSE file...** button serves to open and inspect the device-related GSE file.
- The **Groups...** button leads to the *Group properties* dialog. The group properties refer to the slaves assigned to the master.



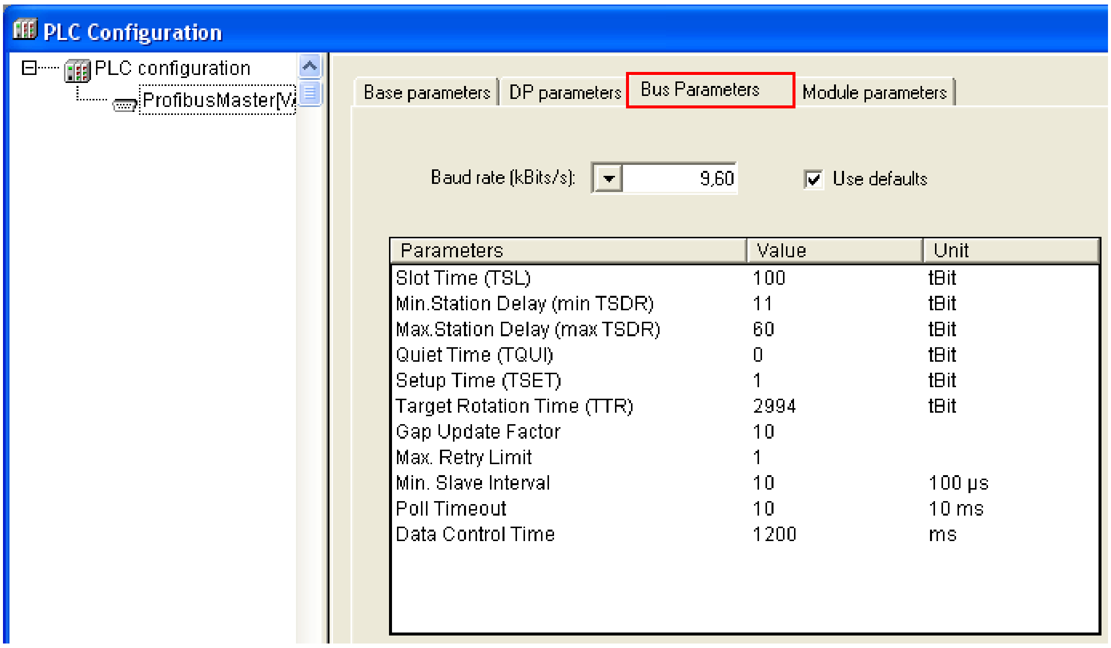
Up to eight groups can be arranged. Set for each group whether they are to be operated in freeze mode and/or sync mode. By assigning the slaves (see "Properties of the DP slave", "Group assignment") to different groups, the data exchange from the master via a global control command can be synchronised.

A freeze command causes a master, a slave, or a group, to "freeze" the input in the current status and transfer these data in the next data exchange process.

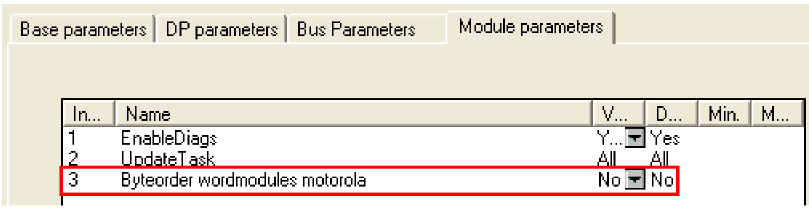
A sync command causes the slaves to connect the data received by the master through to the outputs synchronously regarding time with the next sync command.

In order to switch on/off the freeze and sync option for one group, left-click the corresponding position in the table to place/remove an 'X' at the requested option or right-click it to activate/deactivate the option via a context menu. Moreover you can edit the group names here.

3. Go to the **Bus parameters** tab to set the baud rate for PROFIBUS:



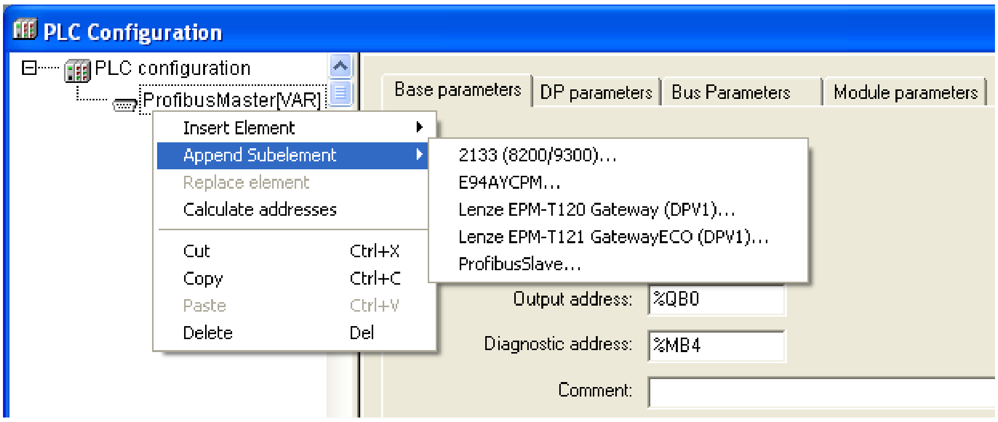
4. Go to the **Module parameters** tab to set the "Byteorder wordmodules motorola" parameter:



The "Byteorder wordmodules motorola" parameter determines how the data are copied from PROFIBUS to the process image.

Setting	Description
No	Standard setting: The data are received by the bus in Motorola format (Big Endian) and copied into the Intel format (Little Endian). The user sees a correct integer value (INT) (example: ’0x1234’ → ’0x1234’). **We recommend to keep this setting. Like this, the slave data are correctly transferred to the process image.**
Yes	The data are copied without being transposed from the bus to the process image. The user sees a transposed integer value (INT) (example: ’0x1234’ → ’0x3412’). • Compatibility with other projects from the version 1.5

5. Attach PROFIBUS slave:



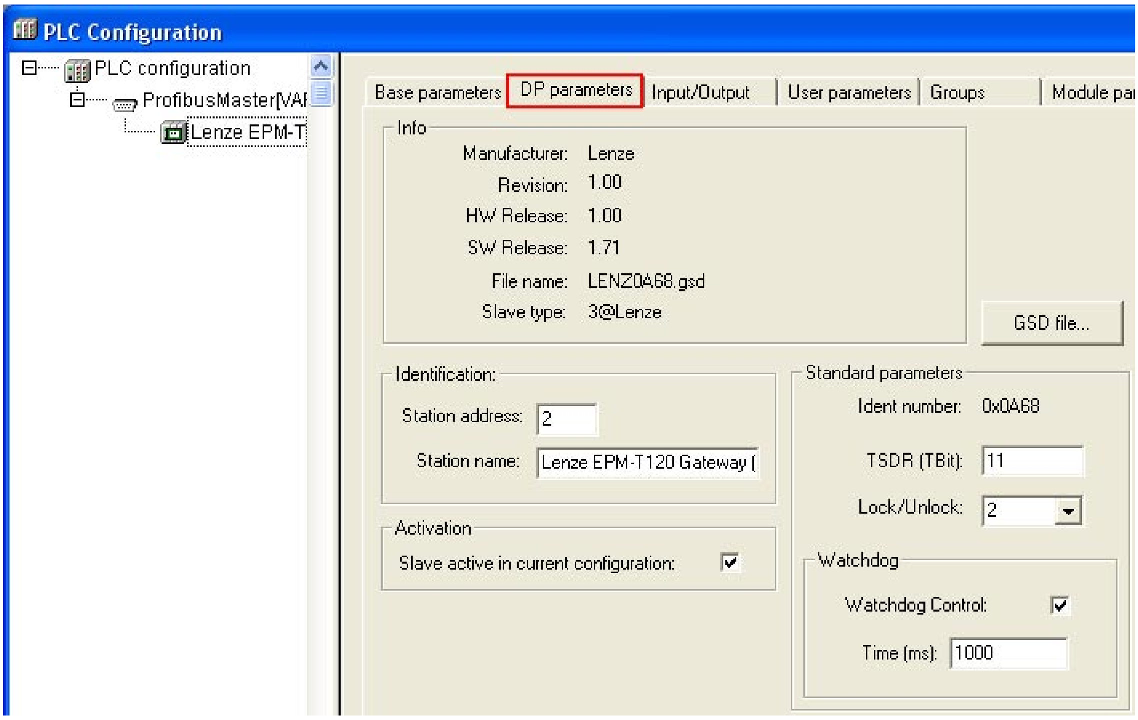
The GSE file of the PROFIBUS slave must be stored in the »PLC Designer« target directory.

▸ Storing the device data base files (GSE) (📖 24)

6.2.6.3 Configuring the PROFIBUS slave

How to configure the PROFIBUS slave:

1. Set DP parameters for the PROFIBUS slave:

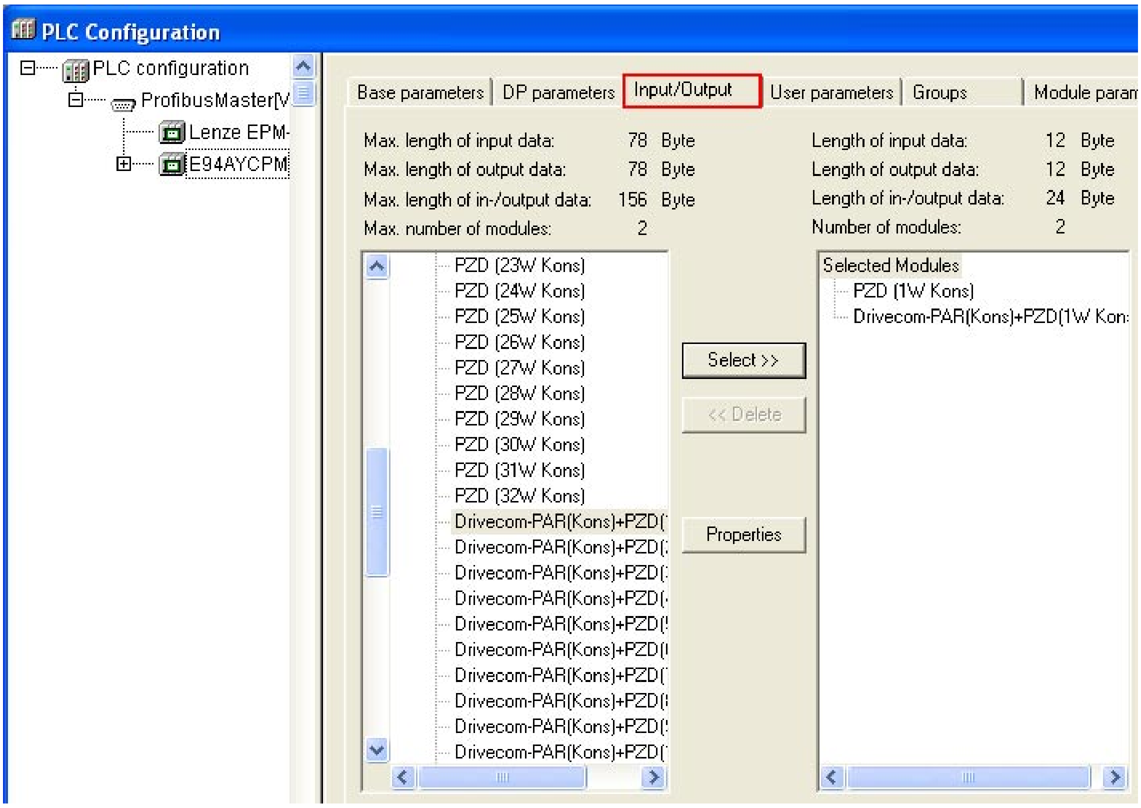


- Enter the station address of the PROFIBUS slave here.
- A baud rate does not need to be set since the slave recognises the baud rate automatically.
- The **GSE file...** button serves to open and inspect the device-related GSE file.

2. Via the **Input/Output** tab, configure the input and output objects for the slaves.

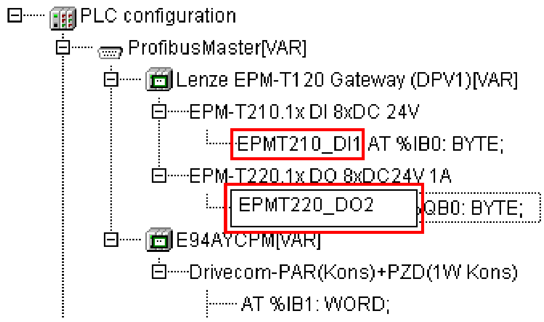
Note!

The input and output objects must be configured in the same sequence as they are arranged physically at the bus.



- In the left window, the dialog lists all input and output modules, process data objects (PCD) and DRIVECOM parameter objects (e.g. for Servo Drives 9400) which are available in the GSE file of the slave.
- The right window contains the configuration of the inputs and outputs currently selected for this device.
- This type of selection is not possible with non-modular slaves. These cause a closed representation of their inputs and outputs in the right window.
- The **Properties** button leads to the *Module properties* dialog of the input or output object currently selected in the left or right list.

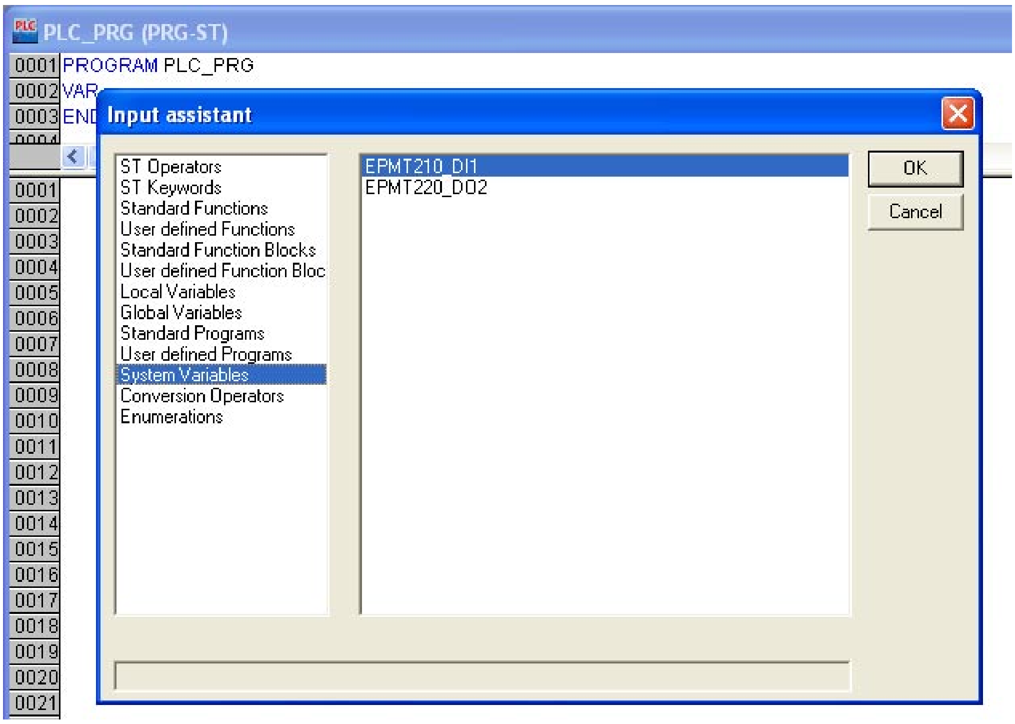
3. Assign for each address of the input and output objects (e.g. *%IB0*, *%QB0*, ...) a symbolic name which is **non-ambiguous** in the entire control configuration in accordance with the IEC 61131 syntax (no blanks and leading digits in the variable name):
   - Symbolic names can be entered by a mouse-click in front of 'AT %... ;'.



Note!

When symbolic names are entered, corresponding system variables are created for the PLC program.

Always use the system variables within the PLC program in order to access the input and output objects or assign values to them.



The **<F2>** function key serves to open the input assistance in the »PLC Designer«.

6.2.6.4 Compiling project data

To compile the project data, select the **Project→Build** menu command or press the **<F11>** function key.

- If errors occurred during the compilation process, you can locate and eliminate them by means of the »PLC Designer« error messages. Then compile the project data again.
- If no errors occurred during the compilation process, save the »PLC Designer« project to the project folder.

6.2.6.5 Logging on to the control system with the »PLC Designer«

To log the »PLC Designer« on to the control system, select the menu command **Online→Login**.

- For this, the PLC program must be error-free.
- Confirm the appearing query dialog whether the new program is to be loaded with **Yes**.

6.2.6.6 Loading and starting the PLC program

How to load and start the PLC program on the IPC:

1. Select the **Online→Download** menu command.
2. Select the required file in the appearing dialog window.
3. Confirm the selection by clicking the **Open** button.
   - The file is loaded onto the IPC and saved there under the same name.
   - The PROFIBUS is initialised.
4. Select the **Online→Run** menu command.
   - The PLC program is executed.
   - The cyclic data transfer starts.

Note!

- The bus starts even if not all stations at the bus are available.
- When the PLC program is stopped, (menu command **Online→Stop**) the cyclic data transfer continues to run until a reset source is executed (Menu command **Online→Reset (original)**).

Tip!

The menu command **Online→Read file from PLC** can be used to reload a file from the IPC into the »PLC Designer« project.

7 CANopen with PROFIBUS

The PROFIBUS bus system can be combined with CANopen. 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 PROFIBUS (as logic bus). The bus systems are synchronised in the control system.

Note!

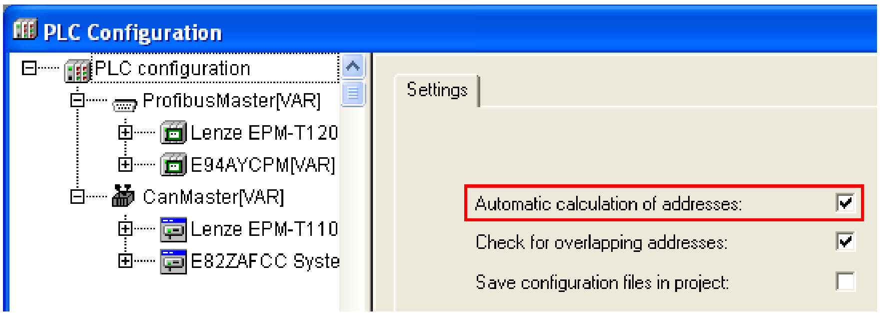
- Mixed operation is only possible with industrial PCs which have two additional slots for communcation cards. Mixed operation is not possible with the "Command Station".
- In release 2.5, PROFIBUS cannot be combined with EtherCAT.
- In the control configuration the PROFIBUS master must be arranged in the first position – in front of the CANopen motion stations.

"CANopen control technology" communication manual

Here you can find detailed information on how to commission CANopen components.

Addressing the CANopen and PROFIBUS stations

The addresses for input and output objects of the PROFIBUS and CANopen stations are automatically allocated in the »PLC Designer« (standard setting):



Note!

We recommend to keep the standard setting. In the 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 of the »PLC Designer«.

8 Function libraries

For configuring PROFIBUS and for diagnostic purposes, the following function libraries are available in the »PLC Designer«:

Function libraries	Application
▶ BusDiag.lib function library (📖 39)	This library serves to query diagnostics information from the PROFIBUS master and the slaves.
▶ NetXPBInfo.lib function library (📖 44)	This library serves to query various information of the PROFIBUS master (e.g. error counter, bus cycle counter).
▶ HilscherNetX.lib function library (📖 46)	This library enables direct access to the package interface of the PROFIBUS communication card.
▶ SysLibDPV1Hilscher.lib function library (📖 50)	This library supports the acyclic PROFIBUS DPV1 - class 1 write and read services for data transfer between the master and the slaves. (For this see PROFIBUS-DP standard.)

More information on the function libraries can be found in the **documentation/ online help of the »PLC Designer«**.

8.1 BusDiag.lib function library

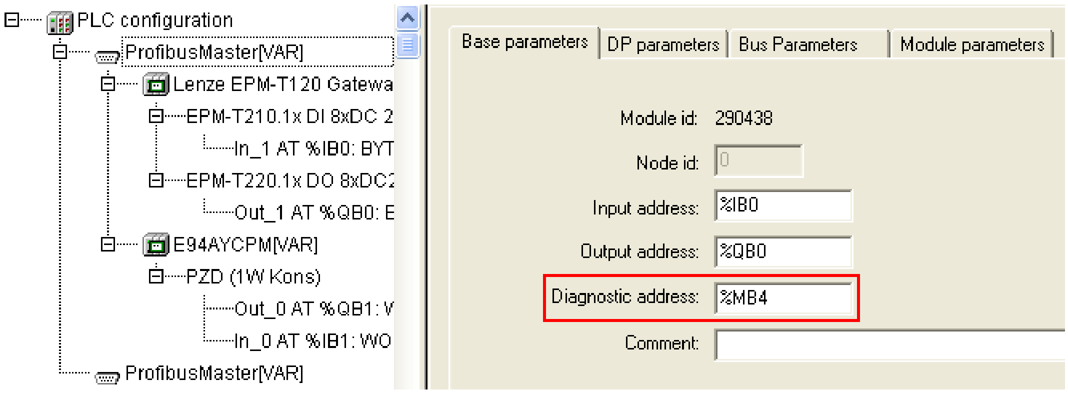
The **BusDiag.lib** function library contains the following function blocks for diagnostics:

- DiagGetBusState function block (📖 39)
- DiagGetState function block (📖 41)

8.1.1 DiagGetBusState function block

This block serves to display the current bus status.

DiagGetBusState must be set via *AT %MByy* to the diagnostics address of the PROFIBUS station to be diagnosed. The status is updated in the background. Thus, the input variables (VAR_INPUT) do not need to be pre-assigned.



Inputs (VAR_INPUT)

The status is updated in the background. Thus, the input variables do not need to be pre-assigned.

Identifier/data type	Meaning/possible settings
ENABLE BOOL	The function block is activated in an edge-controlled manner: • Positive edge (TRUE) = diagnostics information is detected and READY is set to TRUE.
DRIVERNAME POINTER TO STRING	Name of the driver (address of the name) to which the diagnostics order is to be transmitted. • If a '0' is entered here, the diagnostics order is passed to all available drivers.
DEVICENUMBER INT	Identification of the bus which is managed by this module (driver). • The instance results from the PLC configuration. The first node attached corresponds to instance 0, the second one to the instance 1 and so on. • For a second MC-PBM communication card, you must create a second *GetBusState* instance with another *%MByy* address. In order to obtain the extended diagnostics with the DiagGetState function block (📖 41), you must call the block with *Instance := 1*.

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings	
READY BOOL	Always TRUE: Processing of the diagnostics order is completed.	
STATE INT	When READY = TRUE, STATE indicates the current status of the block by one of the following values. These values are assigned to global constants. The constants are stored in the **BusDiag.lib** function library.	
	Value = 1	The bus is ok, no error. • Constant BUSSTATE_BUSOK
	Value = 2	A bus error has occurred. • Constant BUSSTATE_BUSFAULT
	Value = 3	There is no PROFIBUS communication or communication has been aborted. • Constant BUSSTATE_BUSNOTCOMMUNICATING
	Value = 4	The PROFIBUS is set to STOPPED. • Constant BUSSTATE_BUSSTOPPED
EXTENDEDINFO ARRAY [0...129] OF BYTE	1:1 relation between the ARRAY index and the station number of the slave. Only the first three bits of the byte are used:	
	Bit 0	The PROFIBUS station is configured.
	Bit 1	The PROFIBUS station is active at the bus.
	Bit 2	The PROFIBUS station sends an error message. • Detailed information can be obtained via the DiagGetState function block (📖 41).

Note!

After the bus is activated, the slaves set the Error-Flag (bit 2) so that the master first reads out the diagnostics information. If the diagnostics information for the corresponding slave is read out via the DiagGetState function block (📖 41), the Error-Flag is reset.

8.1.2 DiagGetState function block

If a station available on the bus reports an error, its specific diagnostics information can be read with the **DiagGetState** block.

DiagGetState must be called explicitly with the device number and the bus member ID (station address).

Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
ENABLE BOOL	The function block is activated in an edge-controlled manner: • Positive edge (TRUE) = diagnostics information is detected and READY is set to TRUE.
DRIVERNAME POINTER TO STRING	Always set to '0': The diagnostics order is passed to all available drivers.
DEVICENUMBER INT	Identification of the bus which is managed by this module (driver). • The instance results from the PLC configuration. The first PROFIBUS station attached corresponds to the instance 0, the second to the instance 1 etc. • For a second MC-PBM communication card, for instance, there is a second *GetBusState* instance with another *%MByy* address. In order to obtain extended diagnostics with the DiagGetState function block (41), you must call the block with *Instance := 1*.
BUSMEMBERID DWORD	Station address of the PROFIBUS station for which the diagnostics information is to be requested.

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings	
READY BOOL	FALSE: The diagnostics order is processed. TRUE: Processing of the diagnostics order is completed.	
STATE INT	When READY = TRUE, STATE indicates the current status of the block by one of the following values. These values are assigned to global constants. The constants are stored in the **BusDiag.lib** function library.	
	Value = -1	Invalid input parameter • Constant NDSTATE_INVALID_INPUTPARAM
	Value = 0	Diagnostics information is not released. • Constant NDSTATE_NOTENABLED
	Value = 1	Diagnostics information is requested. • Constant NDSTATE_GETDIAG_INFO
	Value = 2	Diagnostics information is available. • Constant NDSTATE_DIAGINFO_AVAILABLE
	Value = 3	No diagnostics information available (error when reading data). • Constant NDSTATE_DIAGINFO_NOTAVAILABLE

<table>
<tr><th>Identifier/data type</th><th colspan="2">Meaning/possible settings</th></tr>
<tr><td rowspan="8">EXTENDEDINFO
ARRAY [0...129]
OF BYTE</td><td colspan="2">Contains the slave-specific diagnostics information.</td></tr>
<tr><td>Byte 0</td><td>Station status 1</td></tr>
<tr><td>Byte 1</td><td>Station status 2</td></tr>
<tr><td>Byte 2</td><td>Station status 3</td></tr>
<tr><td>Byte 3</td><td>Master station number</td></tr>
<tr><td>Byte 4</td><td>Manufacturer's identification mark (high byte)</td></tr>
<tr><td>Byte 5</td><td>Manufacturer's identification mark (low byte)</td></tr>
<tr><td>Byte 6 ... n</td><td>Slave-specific diagnostics information (see documentation of the slave)</td></tr>
</table>

Example: Diagnostics information of Servo Drives 9400 in ’EXTENDEDINFO’

<table>
<tr><th>Byte in EXTENDEDINFO</th><th>Meaning</th></tr>
<tr><td>6</td><td>Bit 0: Station does not exist (set by the master).
Bit 1: Slave is not ready for data exchange.
Bit 2: Configuration data do not match each other.
Bit 3: Slave has extended diagnostic data.
Bit 4: Requested service is not supported by the slave.
Bit 5: Slave response is invalid (set by the master)
Bit 6: Incorrect parameter setting
Bit 7: Slave is parameterised by another master (set by the master).</td></tr>
<tr><td>7</td><td>Bit 0: Slave has to be parameterised again.
Bit 1: Static diagnostics
Bit 2: Fixedly set to "1".
Bit 3: Watchdog active
Bit 4: Freeze command received.
Bit 5: Sync command received.
Bit 6: Reserved
Bit 7: Slave is deactivated (set by the master).</td></tr>
<tr><td>8</td><td>Bit 7: Diagnostics overflow - amount of diagnostics data present in the slave is too large to fit into one telegram.</td></tr>
<tr><td>9</td><td>Bits 0 ... 7: Master address after parameter setting ("0xFF" without parameterisation)</td></tr>
<tr><td>10</td><td>Bits 0 ... 7: ID number (high byte)</td></tr>
<tr><td>11</td><td>Bits 0 ... 7: ID number (low byte)</td></tr>
<tr><td>12</td><td>Header
• The header contains the block length of extended diagnostics and the header byte.
• In this case, the value of the entry is "0x0A" (bytes 6 ... 15 = 10 bytes).</td></tr>
<tr><td>13</td><td>Status_Type
• The value of this entry is fixed. For the following bit assignment it is "0x81":
– Bit 7 = 1: "status"
– Bit 0 = 1: "status message"
– Values of all other bits = 0</td></tr>
<tr><td>14</td><td>Slot_Number
• Value of the slot number: "0x00"</td></tr>
<tr><td>15</td><td>Specifier
• A detected error is entered in the specifier with the identification "0x0" (status coming).
• An eliminated error is entered in the specifier with the identification "0x02" (status going).
• If no errors are indicated, the entry in the specifier has the value "0x00" (no further differentiation).</td></tr>
<tr><td>16</td><td rowspan="2">PROFIsafe, error number of the safety module
• If an error occurs in the safety module, byte 10 (low byte) and byte 11 (high byte) contain the corresponding error number.
• More information can be found in the documentation of the corresponding safety module.</td></tr>
<tr><td>17</td></tr>
</table>

Byte in EXTENDEDINFO	Meaning
18 ... 21	Error code of the PROFIBUS station • More information can be found in the documentation of the corresponding PROFIBUS station.

8.2 NetXPBInfo.lib function library

The **NetXPBInfo.lib** function library serves to query various information on the PROFIBUS master (e.g. error counter, bus cycle counter).

8.2.1 Structure of NETXGETPBINFOSTYP

```
TYPE NETXGETPBINFOSTYP :
STRUCT
   iDev : INT;
   iDummy : INT;
   ulLastCycleTime : DWORD;
   ulCycleStarts : DWORD;
   ulCycleEnds : DWORD;
   usBus_error_cnt : WORD;
   usMsg_Timeout : WORD;
   usRX_Overflow : WORD;
   usBus_Off_cnt : WORD;
   ucErrNumber : BYTE;
   ucErrStationAdr: BYTE;
END_STRUCT
END_TYPE
```

Description of the components

Identifier/data type	Meaning/possible settings
iDev INT	Device number of the PROFIBUS master
iDummy INT	Not used
ulLastCycleTime DWORD	The time measured last to start the bus cycle.
ulCycleStarts DWORD	Number of the bus cycles started
ulCycleEnds DWORD	Number of the bus cycles completed
usBus_error_cnt WORD	Number of the bus error events
usMsg_Timeout WORD	Number of the bus message errors
usRX_Overflow WORD	Number of the transmit overflow events
usBus_Off_cnt WORD	Number of the Bus_OFF events
ucErrNumber POINTER TO BYTE	General error counter
ucErrStationAdr POINTER TO BYTE	Last faulty slave address

8.2.2 NetXGetPBInfos function

This function serves to request various information on the NetX PROFIBUS master. The information is entered into the memory transferred. The return value contains the error code. ’0’ indicates an error, a value unequal ’0’ indicates a successful query.

Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
pAddress POINTER TO NETXGETPBINFOSTYP	Pointer to an instance of the **NETXGETPBINFOSTYP** structure. The data are entered into this structure.

8.3 HilscherNetX.lib function library

The **HilscherNetX.lib** function library enables direct access to the package interface of the PROFIBUS communication card.

In order to use the functions of the library, you must be provided with the corresponding documentation of the respective NetX Hilscher firmware. The documentation can be procured from the Lenze service.

8.3.1 CIFX_PACKET structure

```
TYPE CIFX_PACKET
STRUCT
   tHeader : CIFX_PACKET_HEADER;
   abData : ARRAY [0..1559] OF BYTE;
END_STRUCT
END_TYPE
```

Description of the components

Identifier/data type	Meaning/possible settings
tHeader POINTER TO CIFX_PACKET_HEADER	Command and management data of the package.
abData ARRAY [0...1559] OF BYTE	Data to be sent or received

8.3.2 CIFX_PACKET_HEADER structure

```
TYPE CIFX_PACKET_HEADER :
STRUCT
   ulDest : UDINT;
   ulSrc : UDINT;
   ulDestId : UDINT;
   ulSrcId : UDINT;
   ulLen : UDINT;
   ulId : UDINT;
   ulState : UDINT;
   ulCmd : UDINT;
   ulExt : UDINT;
   ulRout : UDINT;
END_STRUCT
END_TYPE
```

Description of the components

Identifier/data type	Meaning/possible settings
ulDest UDINT	Target of the package in the process flow
ulSrc UDINT	Source of the package in the process flow
ulDestId UDINT	Target reference of the package
ulSrcId UDINT	Source reference of the package
ulLen UDINT	Length of the package without header
ulId UDINT	Identification handle of the transmitter
ulState UDINT	Status of the order
ulCmd UDINT	Package command • See NetX documentation for detailed information.
ulExt UDINT	Extension • Is not used (value = ’0’)
ulRout UDINT	Routing • Is not used (value = ’0’)

8.3.3 CIFXGetChannelHandle function

This function provides the handle of the NetX channel as return value. This handle can be used to request the **CIFXGetPacket** functions and **CIFXPutPacket**. In the event of an error, ’0’ is returned.

Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
iDevice INT	Device number of the NetX station

8.3.4 CIFXPutPacket function

This functions transmits a package to the NetX channel. A package of the CIFX_PACKET type must be transmitted. When it is sent, the *ulSrcId* value is written by the driver. This is important to retrieve the response to the request. The return value is the error code. ’0’ means error-free, unequal ’0’ is the error case. The Hilscher firmware documentation describes the error codes in detail.

Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
hChannel UDINT	Handle of the channel which can be queried via **CIFXGetChannelHandle**.
pPacket POINTER TO CIFX_PACKET	Pointer to the package which is to be transmitted.
ulTimeout UDINT	Transmit time-out which is waited for.

8.3.5 CIFXGetPacket function

This function retrieves the response to a package transmitted before. For this, the management data of the transmitted package must be accepted unchanged. It is important that the *ulSrcId* value changed by the driver is transferred. The return value is the error code. ’0’ means error-free, unequal ’0’ is the error case. The Hilscher firmware documentation describes the error codes in detail.

Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
hChannel UDINT	Handle of the channel which can be queried via **CIFXGetChannelHandle**.
udiSize UDINT	Size of the memory area which is transferred together with *pPacket*.
pPacket POINTER TO CIFX_PACKET	Pointer to the package which is to be transmitted.
ulTimeout UDINT	Receipt time-out which is waited for.
ulRemAddress UDINT	Set to ’0’.

8.4 SysLibDPV1Hilscher.lib function library

The **SysLibDPV1Hilscher.lib** function library supports the acyclic PROFIBUS DPV1 - class 1 write and read services for data transfer between the master and the slaves. The data is addressed within the slave stations via slot and index (for this see the PROFIBUS-DP standard).

If the target system is supported, the following function blocks are available:

- DPV1_Read / DPV1_ReadEx function block (📖 51)
- DPV1_Write / DPV1_WriteEx function block (📖 52)

8.4.1 V1State structure

This structure is used by the blocks of the **SysLibDPV1Hilscher.lib** library with the *State* output variable. It describes the status of the order.

```
TYPE V1State :
   (NotEnabled := 0, InvalidParam, Started, Done, DoneWithError );
END_TYPE
```

Description of the components

Identifier/data type	Meaning/possible settings
NotEnabled := 0	The function block is not active.
InvalidParam	Invalid input parameter
Started	The function block has started processing.
Done	The function block has completed processing.
DoneWithError	The function block has aborted processing with an error.

8.4.2 DPV1_Read / DPV1_ReadEx function block

This function block serves to read data.

Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
ENABLE BOOL	The function block is activated in an edge-controlled manner: • Positive edge (TRUE) = diagnostics information is detected and READY is set to TRUE.
Device INT	Index of the Hilscher card to which the order is transferred.
StationAddr INT	Station address of the slave in PROFIBUS.
Slot INT	Data slot for identifying the data in the slave.
Index INT	Data index for identifying the data in the slave.
Len INT	Length of the data to be read/written in bytes. • Here, the maximum length of the *buffer* data buffer is transferred.
buffer DWORD	The local address of the data (created with ADR()).

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings	
READY BOOL	FALSE: The read request is processed. TRUE: The read request is completed.	
STATE POINTER TO V1State	Information on the status of the order • See V1State structure (📖 50)	
Size INT	Length of the actually read/written data when executed successfully.	
Error ARRAY [0 ... 7] OF BYTE	Information on errors occurred • Only contained in **DPV1_ReadEx** function block.	
	Byte 1	Hilscher error code • See documentation of Hilscher PROFIBUS cards "Protocol Interface Manual Profibus DP" for detailed information.
	Byte 2	Error class code • See PROFIBUS standard for detailed information.
	Byte 3 + 4	Index 2 + 3: Slave-specific error information • See documentation of the slave for detailed information.

8.4.3 DPV1_Write / DPV1_WriteEx function block

This function block serves to write data.

Inputs (VAR_INPUT)

Identifier/data type	Meaning/possible settings
ENABLE BOOL	The function block is activated in an edge-controlled manner: • Positive edge (TRUE) = diagnostics information is detected and READY is set to TRUE.
Device INT	Index of the Hilscher card to which the order is transferred.
StationAddr INT	Station address of the slave in PROFIBUS.
Slot INT	Data slot for identifying the data in the slave.
Index INT	Data index for identifying the data in the slave.
Len INT	Length of the data to be read/written in bytes. • Here, the maximum length of the *buffer* data buffer is transferred.
buffer DWORD	The local address of the data (created with ADR()).

Outputs (VAR_OUTPUT)

Identifier/data type	Meaning/possible settings	
READY BOOL	FALSE: The read request is processed. TRUE: The read request is completed.	
STATE POINTER TO V1State	Information on the status of the order • See V1State structure (📖 50)	
Size INT	Length of the actually read/written data when executed successfully.	
Error ARRAY [0 ... 7] OF BYTE	Information on errors occurred • Only contained in **DPV1_WriteEx** function block.	
	Byte 1	Hilscher error code • See documentation of Hilscher PROFIBUS cards "Protocol Interface Manual Profibus DP" for detailed information.
	Byte 2	Error class code • See PROFIBUS standard for detailed information.
	Byte 3 + 4	Index 2 + 3: Slave-specific error information • See documentation of the slave for detailed information.

8.4.4 Telegram examples of the PROFIdrive parameter data channel (DP-V1)

In the following, a parameter read order and a parameter write order for a Servo Drive 9400 are described.

8.4.4.1 Example of read request: Query heatsink temperature

The heatsink temperature of the Servo Drive 9400 is to be read.

- Code to be read: C00061
- Heatsink temperature: 43 °C

Parameter request

Byte 1	Byte 2	Byte 3	Byte 4
Job reference	Job identification	Axis	Number of indexes
0xXX	0x01	0x00	0x01
	Request parameter for reading		

Byte 5	Byte 6
Attribute	Number of subindexes
0x10	0x00
Value	No subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte
0x5F	0xC2	0x00	0x00
Index = 24575 - code no. = 24575 - 61 = 24514 = 0x5F C2		No subindex	

Parameter response to a correctly executed read request

Byte 1	Byte 2	Byte 3	Byte 4
Job reference	Response identification	Axis	Number of indexes
0xXX	0x01	0x00	0x01
(Mirrored)	Parameter read	(Mirrored)	

Byte 5	Byte 6
Format	Number of values
0x43	0x01
Double word	1 value

Byte 7	Byte 8	Byte 9	Byte 10
Value			
High word: high byte	High word: low byte	Low word: high byte	Low word: low byte
0x00	0x00	0x00	0x2B
Value read = 0x00 00 00 2B = 43 x 1 (internal factor) = 43 [°C]			

Parameter response to a read error

Byte 1	Byte 2	Byte 3	Byte 4
Job reference	Response identification	Axis	Number of indexes
0xXX	0x81	0x00	0x01
(Mirrored)	Parameter not read	(Mirrored)	

Byte 5	Byte 6
Format	Number of values
0x44	0x01
Error	Error code without additional information

Byte 7	Byte 8
Error code	
High byte	Low byte
See documentation of the PROFIBUS station.	

8.4.4.2 Example of write request: Set deceleration time for quick stop

In the Servo Drive 9400, the ramp time for quick stop is to be set to 50 ms.

- ▶ Code to be written to: C00105

Parameter request

Byte 1	Byte 2	Byte 3	Byte 4
Job reference	Job identification	Axis	Number of indexes
0xXX	0x02	0x00	0x01
	Write parameter	Axis 0	1 index

Byte 5	Byte 6
Attribute	Number of subindexes
0x10	0x00
Value	No subindex

Byte 7	Byte 8	Byte 9	Byte 10
Index		Subindex	
High byte	Low byte	High byte	Low byte
0x5F	0x96	0x00	0x00
Index = 24575 - code no. = 24575 - 105 = 24470 = 0x5F 96		No subindex	

Byte 11	Byte 12
Format	Number of values
0x43	0x01
Double word	1 value

Byte 13	Byte 14	Byte 15	Byte 16
Value			
High word: high byte	High word: low byte	Low word: high byte	Low word: low byte
0x00	0x00	0x00	0x32
Value to be written = 0.05 [s] x 1000 (internal factor) = 50 = 0x00 00 00 32			

Parameter response to a correctly executed write request

Byte 1	Byte 2	Byte 3	Byte 4
Job reference	Response identification	Axis	Number of indexes
0xXX	0x02	0x00	0x01
(Mirrored)	Parameter written	(Mirrored)	1 index

Parameter response to a read error

Byte 1	Byte 2	Byte 3	Byte 4
Job reference	Response identification	Axis	Number of indexes
0xXX	0x82	0x00	0x01
(Mirrored)	Parameter not written	(Mirrored)	1 index

Byte 5	Byte 6
Format	Number of values
0x44	0x01
Error	Error code without additional information

Byte 7	Byte 8
Error code	
High byte	Low byte
See documentation of the PROFIBUS station.	

9 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 comprises the following steps:

1. Calculating the total access time $T_{Correction}$ to the peripheral devices.
   - Calculating the total access time to the peripheral devices ($T_{Correction}$) (57)
2. Determining the task utilisation $T_{Task\ utilisation}$ of the application during operation.
   - Detecting the task utilisation of the application ($T_{Task\ utilisation}$) (58)
3. Calculating the minimum cycle time.
   - Calculating the minimum cycle time (60)
4. Optimising the system.
   - Optimising the system (61)

9.1 Calculating the total access time to the peripheral devices ($T_{Correction}$)

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
1 PROFIBUS master (logic)	140 µs
1. axis (16 input words and 16 output words)	40 µs
For each further axis (16 input words and 16 output words)	10 µs

Example

Access times for an industrial PC (ATOM 1.6 GHz) with 3 axes	
Access time of PROFIBUS master	140 µs
+ access time of 1. axis	40 µs
+ access time for 2 further axes	20 µs
= total access time	200 µs

9.2 Detecting the task utilisation of the application ($T_{Task\ utilisation}$)

The time $T_{Task\ utilisation}$ 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«.

9.2.1 Display of the system utilisation in the »PLC Designer« with the task editor

Note!

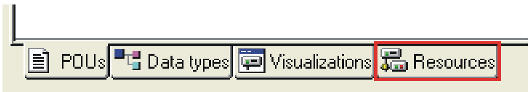
In order to be able to display the utilisation for all tasks, the IEC 61131 *SysTaskInfo* library has to be included in the project.

The task editor contains a dialog window consisting of two parts.

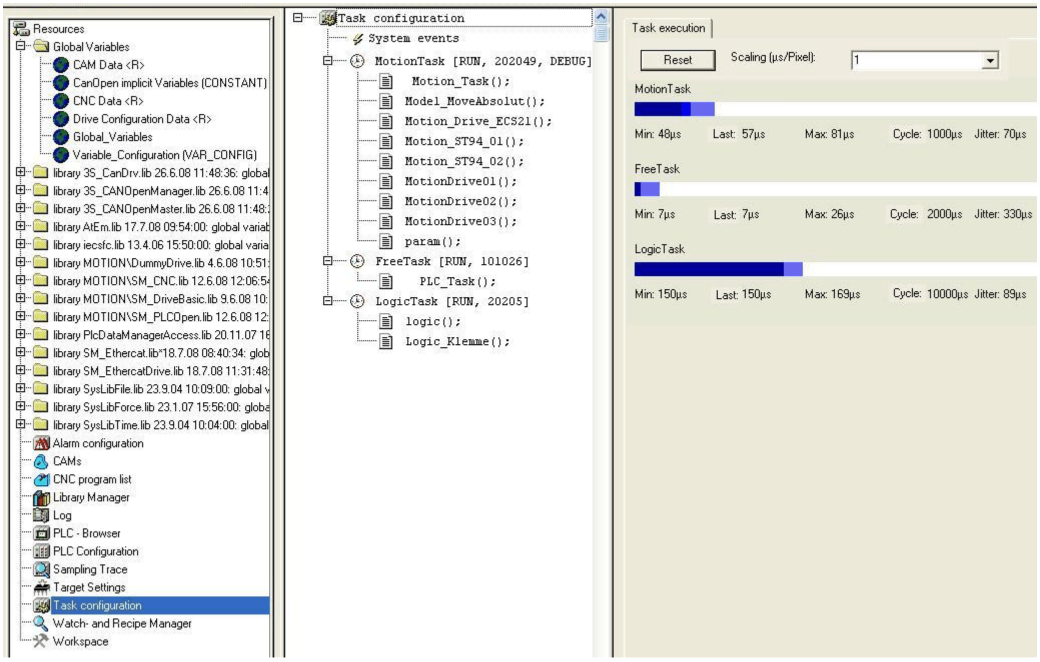
- The left part represents the tasks in a configuration tree.
- If the *Task configuration* entry is highlighted, the utilisation for all tasks is shown in bar diagrams in the right dialog window.

How to display the system utilisation:

1. Select the *Resources* tab:



2. Open the *Task configuration* in the online mode of the »PLC Designer«:



9.2.2 Detecting the task utilisation

Initial situation

A project with, for instance, a motion task and two 2 tasks of a lower priority is created completely.

How to detect the task utilisation $T_{Task\ utilisation}$:

1. For a first measurement of $T_{Task\ utilisation}$ 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 ($T_{Task\ utilisation}$).

9.3 Calculating the minimum cycle time

Note!

For calculating the minimum cycle time, a safety factor of 1.5 is included.

The minimum cycle time T_{min} for a system results from the sum of the times detected before, multiplied by the safety factor:

$$T_{min} > \text{safety factor} \times (T_{Task\ utilisation} + T_{Correction})$$

Example

Configuration: Industrial PC (ATOM 1.6 GHz) with 3 axes		
Access time determined		**Result**
Calculated correction value	$T_{Correction}$	**200 µs** (140 µs + 40 µs +(2 x 10 µs))
Value read from task configuration:	$T_{Task\ utilisation}$	**500 µs**
Actual required computing time		**700 µs**
Minimum cycle time including a safety factor of 1.5	T_{min}	**1050 µs**
Actual cycle time		**2000 µs**

9.4 Optimising the system

How to optimise the system:

1. Log in and load project.
2. Check the task processing times.
3. Optimising the cycle times:
   - If required technologically, the cycle times of the remaining tasks with lower priorities can be decreased.
   - Condition: No task with a low priority must assign more than 60 percent of the corresponding cycle time in its task utilisation.

10 Diagnostics

PROFIBUS can be diagnosed in the »PLC Designer« by means of specific diagnostics function blocks.

Moreover, the field devices, PROFIBUS communication modules, and the MC-PBM communication card are provided with LED status displays for diagnostics.

10.1 Diagnostics in the »PLC Designer«

For the diagnostics of PROFIBUS in the »PLC Designer«, the BusDiag.lib function library (📖 39) provides two function blocks:

- DiagGetBusState function block (📖 39):
  - Shows the current bus status.
- DiagGetState function block (📖 41)
  - If a station available on the bus reports an error, its specific diagnostics information can be read.

10.2 Error messages if communication card MC-PBM is not available

If no communication card MC-PBM has been inserted into the industrial PC, error messages occur during the download of the »PLC Designer« project.

Remedy: Insert the communication card MC-PBM into the industrial PC.

10.3 LED status displays

The LED status displays of the field devices, PROFIBUS communication modules, and the MC-PBM communication card provide information on the device and communication states.

Detailed information on the LED status displays of the field devices and PROFIBUS communication modules can be found in the corresponding documentation.

LED status displays of the MC-PBM communication card

LED	Colour	Status	Description
SYS	Green	On	Communication is active: The device has established at least one connection to a configured station.
		Blinking 5 times per second (5 Hz)	No error in the configuration: Communication is stopped or the device is ready for communication. However, there is no connection to a slave.
		Blinking irregularly	• Starting action: Missing or faulty configuration • Runtime: Host watchdog time error
	Yellow	Blinking once per second (1 Hz)	The device is in bootstrap loader mode and is waiting for the firmware download
		Blinking 5 times per second (5 Hz)	The firmware download is executed.
		Blinking irregularly	A hardware or severe system error has been detected.
	-	Off	No voltage supply or hardware is defective.
ST0	-	Off	No function
ST1	-	Off	No function
ST2	Red	On	The device has a communication problem with at least one PROFIBUS slave or has detected a short circuit.
	Yellow	On	The devices holds the PROFIBUS token and can transmit telegrams.
		Blinking irregularly	The device is in the PROFIBUS network and shares the token with other PROFIBUS master devices.
	-	Off	There is a connection to PROFIBUS.

11 Parameter reference

This chapter supplements the parameter list provided in the online documentation for the industrial PC by the parameters of the **MC-PBM communication card**:

- Parameters of the MC-PBM communication card in slot 1 (📖 65)
- Parameters of the MC-PBM communication card in slot 2 (📖 66)

Note!

- Several MC_PBM communication cards can be used per industrial PC. The name of the card in the »WebConfig« is *MC-PBM*.
- Depending on the slot used, the code numbers differ by an offset of ’500’. To obtain the code numbers for a communication card inserted in slot 2, an offset of ’500’ must be added to the code numbers of a card inserted in slot 1.

Tip!

For general information about parameters please see the online documentation for the industrial PC.

11.1 Parameters of the MC-PBM communication card in slot 1

The parameters are listed in numerically ascending order.

C1031

Parameter | Name: **C1031 | Device: type key**	Data type: VISIBLE_STRING Index: 23544_d = $5BF8_h$
Identification of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1032

Parameter | Name: **C1032 | Device: type version**	Data type: VISIBLE_STRING Index: 23543_d = $5BF7_h$
Version number of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1033

Parameter | Name: **C1033 | Device: Name**	Data type: VISIBLE_STRING Index: 23542_d = $5BF6_h$
Device name of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1034

Parameter | Name: **C1034 | Device: Software version**	Data type: VISIBLE_STRING Index: 23541_d = $5BF5_h$
Software version of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1035

Parameter | Name: **C1035 | Device: Hardware version**	Data type: VISIBLE_STRING Index: 23540_d = $5BF4_h$
Hardware version of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1036

Parameter | Name: **C1036 | Device: Serial number**	Data type: VISIBLE_STRING Index: 23539_d = $5BF3_h$
Serial number of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1037

Parameter | Name: **C1037 | Device: Manufacturer**	Data type: VISIBLE_STRING Index: 23538_d = $5BF2_h$
Manufacturer of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1038

Parameter | Name: **C1038 | Device: Manufacturing date**	Data type: VISIBLE_STRING Index: 23537_d = $5BF1_h$
Manufacturing date of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

11.2 Parameters of the MC-PBM communication card in slot 2

The parameters are listed in numerically ascending order.

C1531

Parameter | Name: **C1531 | Device: type key**	Data type: VISIBLE_STRING Index: 23044_d = $5A04_h$
Identification of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1532

Parameter | Name: **C1532 | Device: type version**	Data type: VISIBLE_STRING Index: 23043_d = $5A03_h$
Version number of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1533

Parameter | Name: **C1533 | Device: Name**	Data type: VISIBLE_STRING Index: 23042_d = $5A02_h$
Device name of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1534

Parameter | Name: **C1534 | Device: Software version**	Data type: VISIBLE_STRING Index: 23041_d = $5A01_h$
Software version of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1535

Parameter | Name: **C1535 | Device: Hardware version**	Data type: VISIBLE_STRING Index: 23040_d = $5A00_h$
Hardware version of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1536

Parameter | Name: **C1536 | Device: Serial number**	Data type: VISIBLE_STRING Index: 23039_d = $59FF_h$
Serial number of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1537

Parameter | Name: **C1537 | Device: Manufacturer**	Data type: VISIBLE_STRING Index: 23038_d = $59FE_h$
Manufacturer of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

C1538

Parameter | Name: **C1538 | Device: Manufacturing date**	Data type: VISIBLE_STRING Index: 23037_d = $59FD_h$
Manufacturing date of the card	
☑ Read access ☐ Write access ☐ CINH ☐ PLC STOP ☐ No transfer	

12 Index

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

KHBPBPCBAUTO ▪ 13383678 ▪ EN 2.2 ▪ TD17

10 9 8 7 6 5 4 3 2 1