Controls

I/O systems

Modular I/O system 1000 ___________________ Reference Manual EN

13514961

Lenze

Contents

1 About this documentation

This documentation shows the implementation of the modular I/O system 1000 into the Lenze "Controller-based Automation" system.

It provides an overview of the device architecture, the available bus coupler modules and the I/O compound modules. Moreover, information is provided on how to use different fieldbuses as well as information on how to configure and parameterise the I/O system with the engineering tools »PLC Designer« and »Engineer«.

Read the I/O system 1000 **system manual** before you start working.

The system manual contains safety instructions which must be observed!

Target group

This documentation is directed at all persons who would like to parameterise, configure, and diagnose the Lenze I/O system 1000 using the »PLC Designer« and »Engineer« engineering tools.

Information regarding the validity

The information in this documentation applies to:

Product series	Type designation	Version
I/O systems	I/O system 1000	From hardware version 1B, software version 10

Screenshots/application examples

All screenshots in this documentation are application examples. Depending on the firmware version of the device and software version of the installed engineering tools, the screenshots in this documentation may differ from the screen representation.

1.1 Document history

Version			Description
5.0	09/2016	TD17	General corrections
4.0	10/2015	TD17	Preparation of the documentation for updating the online help via the »EASY Package Manager«.
3.0	12/2013	TD06	Summarised descriptions for »PLC Designer« and »Engineer« in one documentation.
1.2	11/2012	TD05	General revision of the document structure, extended module overview by I/O compound modules EPM-S406 and EPM-S408.
1.1	07/2012	TD06	Extended by "Commissioning of a counter" example
1.0	11/2011	TD11	First edition

Tip!

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

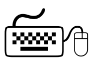
www.lenze.com

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
Decimal	[0 ... 9]	No prefix is used.
Hexadecimal	0x[0 ... F]	0x60F4 Exception: The prefix (0x) in a table is not needed if the table head contains the reference to a hexadecimal number.
Binary	0b[0/1]	0b11 Exception: The prefix (0b) in a table is not needed if the table head contains the reference to a binary number.
Text		
Version information	Blue text colour	All information that only applies to or from a certain software version of the inverter is marked correspondingly in this documentation. Example: This function extension is available from software version V3.0!
Index name	I-[Index]	For a clear distinction for the Lenze codes (Cxxxx), the indices of the process data objects contain a prefix "I-", e.g. B. I-1400 for the RxPDO1.
Program name	» «	The Lenze PC software »PLC Designer«...
Window	*italics*	The *message window*... / The *Options* dialog box ...
Variable names		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 commands must be used in sequence to carry out a function, 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	underlined	Optically highlighted reference to another topic. It is activated with a mouse-click in this online documentation.
Icons		
Page reference	(📖 11)	Optically highlighted reference to another page. It is activated with a mouse-click in this online documentation.
Step-by-step instructions		Step-by-step instructions are indicated by a pictograph.

1.3 Terminology used

Term	Meaning
Controllers	The Controller is the central component of the Lenze automation system which control the motion sequences by means of the application software. The Controller communicates with the field devices (inverters) via the fieldbus.
Engineering PC	The Engineering PC and the Engineering tools installed serve to configure and parameterise the system "Controller-based Automation". The Engineering PC communicates with the controller via Ethernet.
Engineering tools	Software solutions for easy engineering in all phases which serve to commission, configure, parameterise and diagnose the Lenze automation system.
	»EASY Navigator« – ensures easy operator guidance • All convenient Lenze Engineering tools at a glance • Tools can be quickly selected • The clear structure simplifies the engineering process from the start
	»EASY Starter« – easy-to-use tool for service technicians • Especially designed for commissioning and maintaining Lenze devices • Graphic surface with very few icons • Easy to run online diagnostics, set parameters and perform commissioning • No risk of accidentally changing an application • Loading of ready-to-use applications to the device
	»Engineer« – multi-device engineering • For all products in the Lenze portfolio • Practical user interface • Graphic interfaces make it easy to navigate • Can be applied in every phase of a project (project planning, commissioning, production) • Parameter setting and configuration
	»PLC Designer« –For programming processes • Creating your own programs • Programming Logic & Motion according to IEC 61131-3 (AWL, KOP, FUP, ST, AS and CFC-Editor), based on CoDeSys V3 • Certified function blocks according to PLCopen part 1 + 2 • Graphic DIN 66025 Editor (G code) with DXF import • Integrated visualisation for easy process visualisation • All important information at a glance during the commissioning process
Fieldbus node	Devices integrated in the bus system as, for instance, Controller and inverter
Field device	
Inverters	Generic term for Lenze frequency inverters, servo inverters
PLC	Programmable Logic Controller (PLC)
Bus systems	
CAN	CAN (Controller Area Network) is an asynchronous, serial fieldbus system.
CANopen	CANopen® is a communication protocol based on CAN. The Lenze system bus (CAN on board) operates with a subset of this communication protocol. CANopen® is a registered community trademark of the CAN user organisation CiA® (CAN in Automation e. V.).
DeviceNet	DeviceNet™ is a fieldbus system based on CAN (Controller Area Network). DeviceNet™ is a trademark and patented technology, licensed by the user organisation ODVA (Open DeviceNet Vendor Association), USA.
EtherCAT	EtherCAT® (**Ether**net for **C**ontroller and **A**utomation **T**echnology) is an Ethernet-based fieldbus system which fulfils the application profile for industrial real-time systems. EtherCAT® is a registered trademark and patented technology, licenced by Beckhoff Automation GmbH, Germany.

Term	Meaning
Modbus	The Modbus protocol is an open communication protocol based on a client/server architecture. It has been developed for the communication with programmable logic controllers. Further development is carried out by the international user organisation Modbus Organization, USA.
PROFIBUS	PROFIBUS® (Process Field Bus) is a widely used fieldbus system for the automation of machines and production lines. PROFIBUS® is a registered trademark and patented technology licensed by the PROFIBUS & PROFINET International (PI) user organisation.
PROFINET	PROFINET® (Process Field Network) is a real-time capable fieldbus system based on Ethernet. PROFINET® is a registered trademark and patented technology licensed by the PROFIBUS & PROFINET International user organisation (PI).

1.4 Definition of the notes used

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

Safety instructions

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 Product description

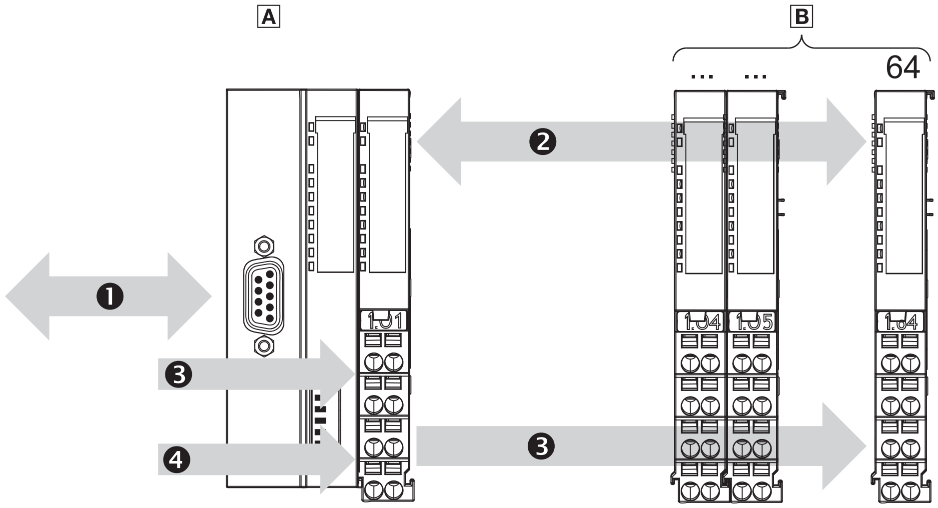
Features of the I/O system 1000

The I/O system 1000 is suitable for the implementation of complex automation applications consisting of the following components: bus coupler module, I/O compound modules, and backplane bus.

- Modular system
- Dimensions
  - I/O compound module 109 × 76.5 × 12.5 mm
  - Bus coupler module 109 × 76.5 × 48.5 mm
- Mounting on standard DIN rail (35 mm)
- Shield connection to standard metal rail (10 × 3 mm)
- The two-piece structure (separation of electronics and process integration) enables a quick exchange of modules in the event of service
- Wiring level via spring terminal (max. 1.5 mm^2)
- Operation at DC 24 V (DC 20.4 ... DC 28.8 V)
- Supply voltage of electronics and process level are separated
- Number of I/O compound modules
  - A bus coupler can contain up to 64 modules
- Individual labelling by insertable labels (item designation)
- Electrical wiring diagrams are directly printed on the module
- Electrical isolation to the fieldbus and the process level
- Creation of electrical isolations by power supply modules

2.1 Device architecture

Device architecture of the I/O system 1000



A	Bus coupler module • CANopen: EPM-S110 • PROFIBUS: EPM-S120 • EtherCAT: EPM-S130 • PROFINET: EPM-S140 • DeviceNet: EPM-S150 • Modbus TCP: EPM-S160
B	I/O compound modules, max. 64
❶	Connection to the bus system (CANopen/EtherCAT) • CANopen, PROFIBUS: via a 9-pole Sub-D plug (see illustration) • EtherCAT: RJ45, double
❷	Internal backplane bus for communication between the bus coupler module and the I/O compound modules • Side-by-side mounting of the module feet of the I/O compound modules provides for the electrical connection to the bus coupler module.
❸	Supply voltage DC 24 V, max. 10 A for consumers at output modules • Side-by-side mounting of the module feet of the I/O compound modules provides for the electrical connection to the bus coupler module.
❹	Supply voltage DC 24 V for the electronics • Side-by-side mounting of the module feet of the I/O compound modules provides for the electrical connection to the bus coupler module.

2.2 Available bus coupler modules

Type designation	Short designation	Description
EPM-S110	CANopen	CANopen bus coupler module
EPM-S120	PROFIBUS	PROFIBUS bus coupler module
EPM-S130	EtherCAT	EtherCAT bus coupler module
EPM-S140	PROFINET	Bus coupler module PROFINET
EPM-S150	DeviceNet	Bus coupler module DeviceNet
EPM-S160	Modbus TCP	Bus coupler module Modbus TCP

Note!

Bus coupler module EPM-S130 (EtherCAT)

Only **I/O compound modules EPM-Sxxx from hardware version 1B** are supported.

2.3 Available I/O compound modules

Type designation	Short designation	Description	Process data
Digital-I/O			
EPM-S200	DI2, DC 24V	2 digital inputs	2 bits
EPM-S201	DI4, DC 24V	4 digital inputs	4 bits
EPM-S202	DI8, DC 24V	8 digital inputs	8 bits
EPM-S203	DI4, DC 24V	4 digital inputs • Three-wire conductor connection	4 bits
EPM-S204	DI2, NPN, DC 24V	2 digital inputs NPN	2 bits
EPM-S205	DI4, NPN, DC 24V	4 digital inputs NPN	4 bits
EPM-S206	DI8, NPN, DC 24V	8 digital inputs NPN	8 bits
EPM-S207	DI2 Time Stamp, DC 24V	2 digital inputs ENP	Write 8 bytes
EPM-S300	DO2, DC 24V, 0.5A	2 digital outputs, 0.5 A	2 bits
EPM-S301	DO4, DC 24V, 0.5A	4 digital outputs, 0.5 A	4 bits
EPM-S302	DO8, DC 24V, 0.5A	8 digital outputs, 0.5 A	8 bits
EPM-S303	DO2, NPN, DC 24V, 0.5A	2 digital outputs, npn	2 bits
EPM-S304	DO4, NPN, DC 24V, 0.5A	4 digital outputs, npn	4 bits
EPM-S305	DO8, NPN, DC 24V, 0.5A	8 digital outputs, npn	8 bits
EPM-S306	DO2, DC 24V, 2A	2 digital outputs, 2 A	2 bits
EPM-S308	DO2, relay 230V, 3A	2 relay outputs	2 bits
EPM-S309	DO4, DC 24V, 2A	4 digital outputs, 2 A	4 bits
EPM-S310	DO2 Time Stamp, DC 24V	2 digital outputs, 0.5 A, ENP	8 bytes, reading 2 bytes, writing
Analog I/O			
EPM-S400	AI2, 12BIT, DC 0...10V	2 analog inputs, 0 ... 10 V DC	4 bytes
EPM-S401	AI4, 12BIT, DC 0...10V	4 analog inputs, 0 ... 10 V DC	8 bytes
EPM-S402	AI2, 12BIT, DC 0/4...20mA	2 analog inputs, 0/4 ... 20 mA	4 bytes
EPM-S403	AI4, 12BIT, DC 0/4...20mA	4 analog inputs, 0/4 ... 20 mA	8 bytes
EPM-S406	AI2, 16BIT, DC +/-10V	2 analog inputs for voltage measurement ±10 V	4 bytes
EPM-S408	AI2, 16BIT, DC 0/4...20mA	2 analog inputs for current measurement	4 bytes
EPM-S500	AO2, 12BIT, DC 0...10V	2 analog outputs, 0 ... 10 V DC	4 bytes
EPM-S501	AO4, 12BIT, DC 0...10V	4 analog outputs, 0 ... 10 V DC	8 bytes
EPM-S502	AO2, 12BIT, DC 0/4...20mA	2 analog outputs, 0/4 ... 20 mA	4 bytes
EPM-S503	AO4, 12BIT, DC 0/4...20mA	4 analog outputs, 0/4 ... 20 mA	8 bytes
Temperature measurement			
EPM-S404	AI2, 16BIT, resistor	4 analog inputs for resistance measurement	4 bytes
EPM-S405	AI2, 16BIT, thermo	2 analog inputs for thermocouple measurement	4 bytes
Counter			
EPM-S600	Counter 1, DC 24V	1 counter 32 bits, 24 V DC (reading, setting, comparing)	12 bytes, reading 10 bytes, writing
EPM-S601	Counter 2, DC 24V	2 counters 32 bits, 24 V DC (reading, setting)	12 bytes, reading 12 bytes, writing

Type designation	Short designation	Description	Process data
EPM-S602	Counter 1, DC 5V	1 counter 32 bits, 5 V DC (reading, setting)	8 bytes, reading 10 bytes, writing
EPM-S603	Counter 2, DC 24V	2 counters 32 bits, 24 V DC (reading)	12 bytes, reading 4 bytes, writing
Encoder evaluation			
EPM-S604	SSI	1 SSI absolute value encoder	6 bytes, reading
Pulse width modulation (PWM)			
EPM-S620	PWM 2, DC 24V	2 digital outputs, 0.5 A PWM	12 bytes, reading 4 bytes, writing
Communication			
EPM-S640	RS232	RS232 module	-
EPM-S650	RS422/RS485	RS422/RS485 module	-

Note!

I/O compound modules EPM-S207/EPM-S310

For using the modules EPM-S207 and EPM-S310, the **L_SM3_DriveUtil.lib** function library provides the function blocks **L_SMC_AbortTrigger_FastIO** and **L_SMC_TouchProbe_FastIO**.

If, for instance, motor positions are to be evaluated via these modules, additional modules with ticker function are required, as e.g. EPM-S600 (HTL), EPM-S602 (TTL) or EPM-S604 (SSI) for detecting the encoder signals.

I/O compound modules EPM-S640/EPM-S650

For using the modules EPM-S640 and EPM-S650, the **L_COM_EPMS640** function library is available.

Bus coupler module EPM-S130 (EtherCAT)

Only **I/O compound modules EPM-Sxxx from hardware version 1B** are supported.

2.4 Digital-I/O EPM-S207, EPM-S310 – time stamp function (example)

Note!

The time stamp modules **EPM-S207** and **EPM-S310** can only be operated on front modules with an integrated µs ticker.

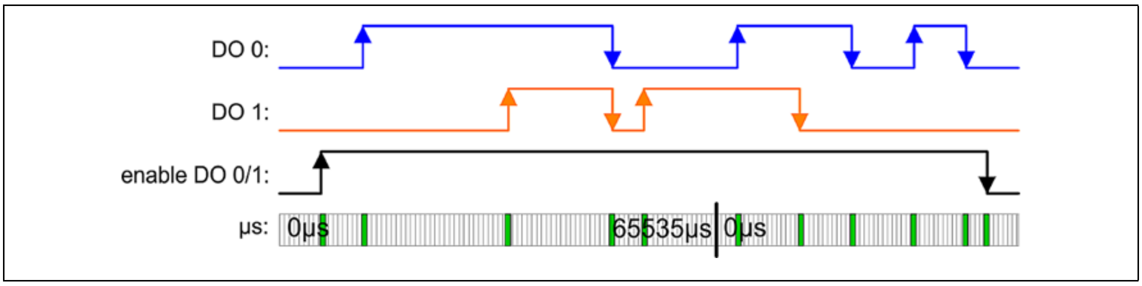
The example shows in which order the time stamp entries are stored and processed.

Here, a module is projected that assigns 20 bytes for five time stamp entries in the output area.

At the given times of the ticker, the outputs should have the following states:

Consecutive number (RN)	Ticker value in µs	Status DO 0 (Bit 7)	Status DO 1 (Bit 6)	Enable DO 0 (Bit 5)	Enable DO 1 (Bit 4)
0x01	6000	0	0	1	1
0x02	12506	1	0	1	1
0x03	34518	1	1	1	1
0x04	49526	0	0	1	1
0x05	54529	0	1	1	1
0x06	3500	1	1	1	1
0x07	12443	1	0	1	1
0x08	20185	0	0	1	1
0x09	30140	1	0	1	1
0x0A	37330	0	0	1	1
0x0B	40000	0	0	0	0

The values of the table result in the following time diagram:



[2-1] Time diagram: Example time stamp function

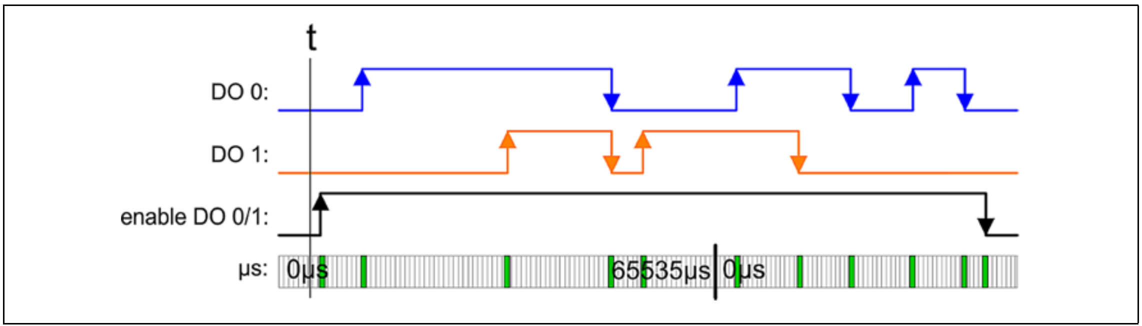
Writing time stamp entries 1 ... 5

After writing the first 5 time stamp entries (RN = 0x01 ... 0x05) into the process output data, these are directly transmitted to the FIFO memory of the module.

The input data includes the corresponding status bytes.

FIFO	Output data	Consecutive number (RN)	Ticker value in µs	Input data
1	0b00110000	0x01	6000	RN_LAST: 0x45 RN_NEXT: 0xC1 STS_FIFO: 0x00 NUM_ETS: 0x05
2	0b10110000	0x02	12506	
3	0b11110000	0x03	34518	
4	0b00110000	0x04	49526	
5	0b01110000	0x05	54529	
6	0b00000000	0x00	0	
...	0b00000000	0x00	0	
31	0b00000000	0x00	0	

The diagram indicates the status of the outputs at the "t" time.



[2-2] Time diagram:Write time stamp entries 1 ... 5

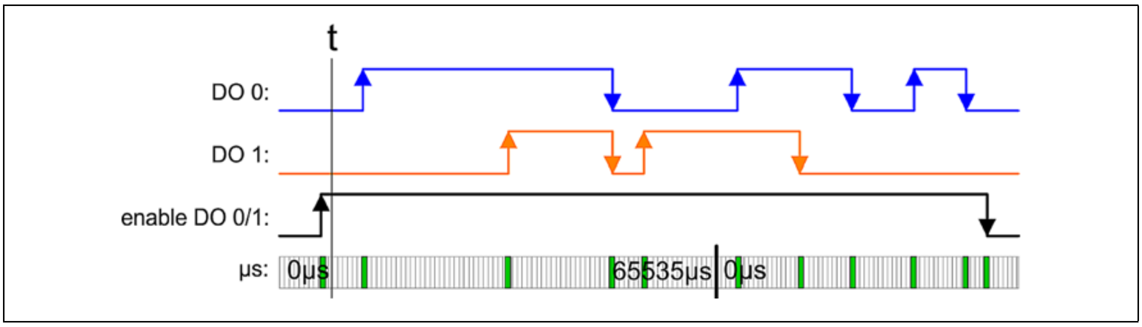
Executing the time stamp function for the 1st entry

In order that the outputs can be triggered accordingly, you have to enable them before. In this example, you enable both outputs with the 1st "running number" (RN).

The 1st time stamp entry (RN = 0x01) is executed and deleted from the FIFO memory.

FIFO	Output data	Consecutive number (RN)	Ticker value in µs	Input data
1	0b10110000	0x02	12506	RN_LAST: 0x45 RN_NEXT: 0xC2 STS_FIFO: 0x00/0x02 NUM_ETS: 0x04
2	0b11110000	0x03	34518	
3	0b00110000	0x04	49526	
4	0b01110000	0x05	54529	
5	0b00000000	0x00	0	
...	0b00000000	0x00	0	
31	0b00000000	0x00	0	

The diagram indicates the status of the outputs at the "t" time.



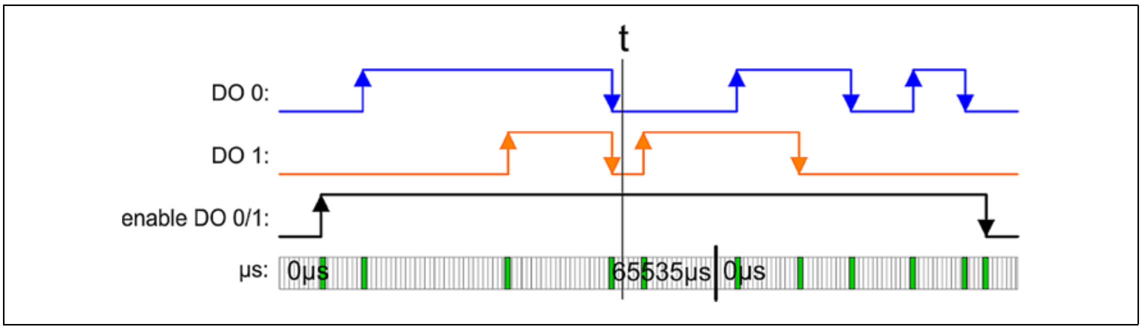
[2-3] Time diagram: Execute time stamp function for the 1st entry

Executing the time stamp function for the entries 2 ... 4

The states of the time stamp entries 2 ... 4 (RN = 0x02 ... 0x04) are output successively and deleted from the FIFO memory.

FIFO	Output data	Consecutive number (RN)	Ticker value in µs	Input data
1	0b01110000	0x05	54529	RN_LAST: 0x45 RN_NEXT: 0xC5 STS_FIFO: 0x00/0x02 NUM_ETS: 0x01
2	0b00000000	0x00	0	
...	0b00000000	0x00	0	
31	0b00000000	0x00	0	

The diagram indicates the status of the outputs at the "t" time.



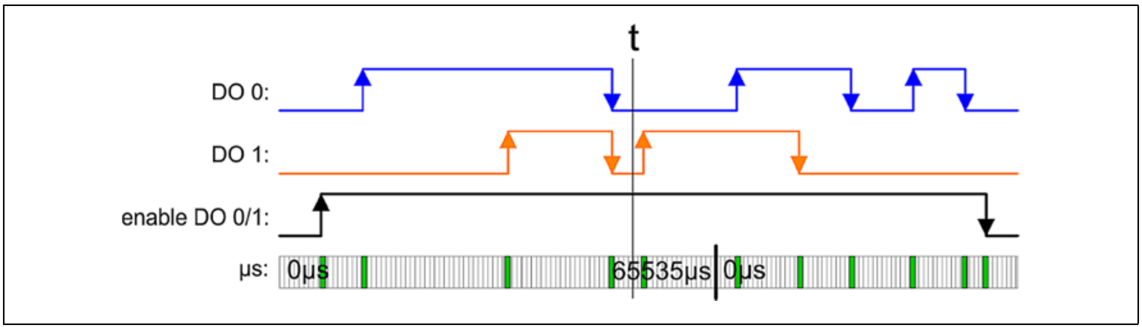
[2-4] Time diagram: Execute the time stamp function for the entries 2 ... 4

Writing time stamp entries 6 ... 10

After writing the time stamp entries 6 ... 10 (RN = 0x06 ... 0x0A) into the process output data, these are directly transmitted to the FIFO memory of the module.

FIFO	Output data	Consecutive number (RN)	Ticker value in µs	Input data
1	0b01110000	0x05	54529	RN_LAST: 0x4A RN_NEXT: 0xC5 STS_FIFO: 0x00/0x02 NUM_ETS: 0x06
2	0b11110000	0x06	3500	
3	0b10110000	0x07	12443	
4	0b00110000	0x08	20185	
5	0b10110000	0x09	30140	
6	0b00110000	0x0A	37330	
7	0b00000000	0x00	0	
...	0b00000000	0x00	0	
31	0b00000000	0x00	0	

The diagram indicates the status of the outputs at the "t" time.



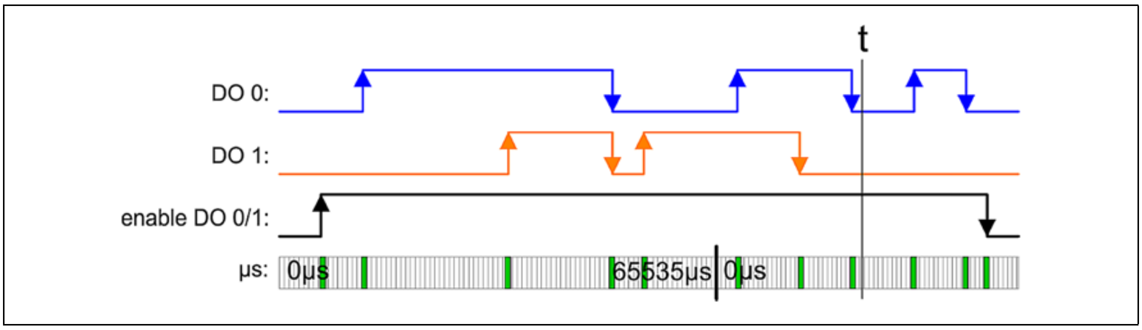
[2-5] Time diagram: Write time stamp entries 6 ... 10

Executing the time stamp function for the entries 6 ... 8

The states of the time stamp entries 6 ... 8 (RN = 0x06 ... 0x08) are output successively and deleted from the FIFO memory.

FIFO	Output data	Consecutive number (RN)	Ticker value in µs	Input data
1	0b10110000	0x09	30140	RN_LAST: 0x4A RN_NEXT: 0xC5 STS_FIFO: 0x00/0x02 NUM_ETS: 0x02
2	0b00110000	0x0A	37330	
3	0b00000000	0x00	0	
...	0b00000000	0x00	0	
31	0b00000000	0x00	0	

The diagram indicates the status of the outputs at the "t" time.



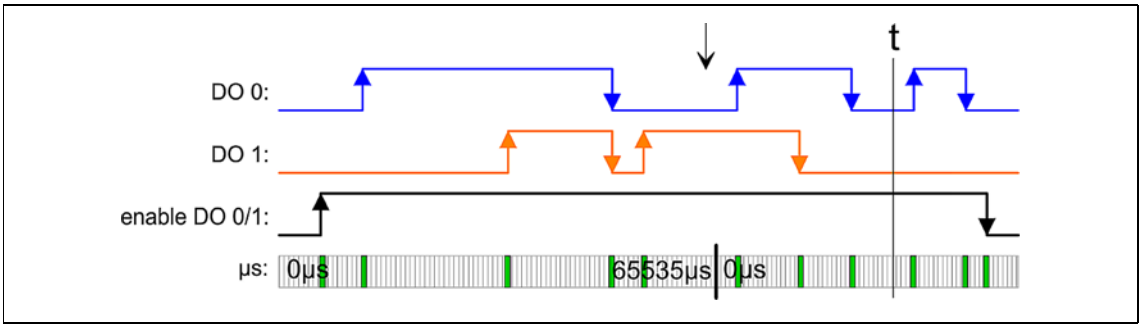
[2-6] Time diagram: Execute the time stamp function for the entries 6 ... 8

Writing the last time stamp entry

Since less than five time stamp entries are written, bit 6 of the "running number" (RN) must always be set for the last time stamp entry. RN = 0x0B becomes 0x4B.

FIFO	Output data	Consecutive number (RN)	Ticker value in µs	Input data
1	0b10110000	0x09	30140	RN_LAST: 0x4B RN_NEXT: 0xC9 STS_FIFO: 0x80/0x82 NUM_ETS: 0x03
2	0b00110000	0x0A	37330	
3	0b00000000	0x4B	40000	
4	0b00000000	0x00	0	
...	0b00000000	0x00	0	
31	0b00000000	0x00	0	

The diagram indicates the status of the outputs at the "t" time.



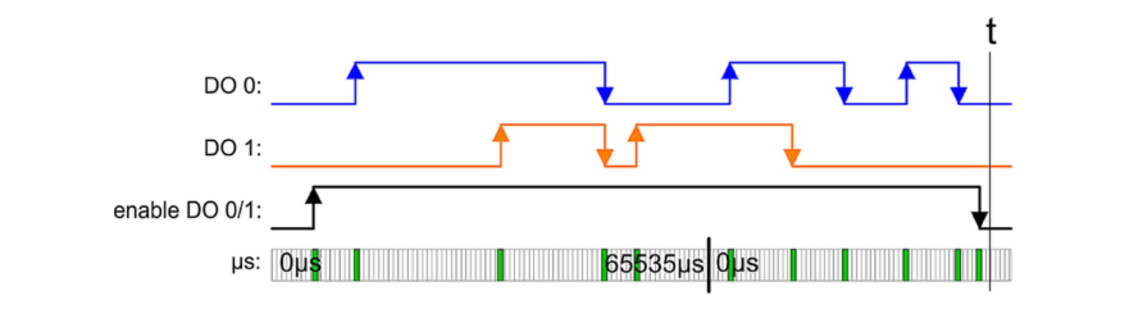
[2-7] Time diagram: Write the last time stamp entry

Executing the time stamp function for the last entries

The states of the last time stamp entries (RN = 0x09 ... 0x4B) are output successively and deleted from the FIFO memory.

FIFO	Output data	Consecutive number (RN)	Ticker value in µs	Input data
1	0b00000000	0x00	0	RN_LAST: 0x4B RN_NEXT: 0xCC STS_FIFO: 0x80/0x82 NUM_ETS: 0x00
2	0b00000000	0x00	0	
...	0b00000000	0x00	0	
31	0b00000000	0x00	0	

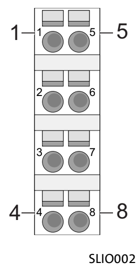
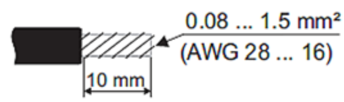
The diagram indicates the status of the outputs at the "t" time.



[2-8] Time diagram: Execute the time stamp function for the last entries

2.5 Analog I/O EPM-S404 – 4(2) analog inputs for resistance measurement

Connections

<table>
<tr><th colspan="4">Module/spring pressure terminals</th></tr>
<tr><th>View</th><th>Name</th><th>Description</th><th>Terminal data</th></tr>
<tr><td rowspan="8">
</td><td>1</td><td>Analog input AI1 (+)</td><td rowspan="8">
</td></tr>
<tr><td>2</td><td>Analog input AI1 (GND)</td></tr>
<tr><td>3</td><td>Analog input AI3 (+)</td></tr>
<tr><td>4</td><td>Analog input AI3 (GND)</td></tr>
<tr><td>5</td><td>Analog input AI2 (+)</td></tr>
<tr><td>6</td><td>Analog input AI2 (GND)</td></tr>
<tr><td>7</td><td>Analog input AI4 (+)</td></tr>
<tr><td>8</td><td>Analog input AI4 (GND)</td></tr>
</table>

2-, 3-, 4-wire conductor measurement

Note!

Use parameter setting to deactivate unused inputs.

If thermal detectors are connected in a 3 or 4 conductor setup, channels 3 and/or 4 must be deactivated.

The module does not provide any auxiliary supply for sensors.

The terminal assignment indicates how to connect your sensors in case of a 2, 3 or 4-wire conductor measurement.

- You can use all channels to carry out a 2-wire conductor measurement.
- A three-wire conductor measurement is only possible at the channels 1 and 2.
  - Please note that in case of a three-wire conductor measurement you always deactivate the correspondent channel by parameterisation. The correspondent channel of channel 1 is channel 3 and the one of channel 2 is channel 4.
  - Deactivate unused channels by parameterisation.
- A 4-wire conductor measurement is only possible at the channels 1 and 2.
  - The measuring current for channel 1 is output on the pins 1 and 2. The measurement for channel 1 takes place at the pins 3 and 4. The analog value for channel 1 is displayed in the input word 0.
  - The measuring current for channel 2 is output on the pins 5 and 6. The measurement for channel 2 takes place at the pins 7 and 8. The analog value for channel 2 is displayed in the input word 1.
  - Please note that in case of a four-wire conductor measurement you always deactivate the correspondent channel by parameterisation. The correspondent channel of channel 1 is channel 3 and the one of channel 2 is channel 4. Unused
  - Deactivate unused channels by parameterisation.

2.6 Counter modules EPM-S600 ... EPM-S603 – control and status words

2.6.1 Status word EPM-S600

Bit	Name	Function
0	STS_SYNC	Reset was active
1	STS_CTRL_DO	Is set if the digital output is enabled
2	STS_SW-GATE	Software gate status (set if SW gate active)
3	STS_RST	Status of the reset input
4	STS_STRT	Hardware gate status (set if HW gate active)
5	STS_GATE	Status of internal gate (set if internal gate active)
6	STS_DO	Status of digital counter output (DO)
7	STS_C_DN	Status set in the case of backward counter direction
8	STS_C_UP	Status set in the case of forward counter direction
9	STS_CMP*	Status of comparator is set if the comparison condition is met. If the comparison is deactivated, (counter mode byte 1 = 0b000), the bit has no function.
10	STS_END*	Status set if final value has been reached
11	STS_OFLW*	Status set in the case of an overflow
12	STS_UFLW*	Status set in the case of an underflow
13	STS_ZP*	Status set in the case of a zero crossing
14	STS_LTCH	Status of the latch input
15	-	Reserved

* The bits are set until reset with RES_SET (bit 6 control word)

2.6.2 Control word EPM-S600

Bit	Name	Function
0	CTRL_SYNC_SET	Activation/deactivation of the counting signal: • TRUE↘FALSE: The input for the counting signal is deactivated and the current counter content is reset to 0. • FALSE↗TRUE edge: The input for the counting signal is activated.
1	CTRL_DO_SET	Enables the digital output
2	SW_GATE_SET	Sets the software gate
3	-	Reserved
4	-	Reserved
5	COUNTERVAL_SET	Set counter temporarily to the value in the preset value
6	RES_SET	Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a rising edge
7	-	Reserved
8	CTRL_SYNC_RESET	Activation/deactivation of the zero track evaluation: • TRUE↘FALSE: The zero track evaluation is activated. • FALSE↗TRUE edge: The zero track evaluation is stopped. The counter continues to count irrespective of the zero pulse. For this purpose, bit 0 (CTRL_SYNC_SET) must be set to TRUE.
9	CTRL_DO_RESET	Inhibits the digital output
10	SW_GATE_RESET	Resets the software gate
11 ... 15	-	Reserved

2.6.3 Status word EPM-S601

Bit	Name	Function
0	-	Reserved
1	STS_CTRL_COMP	Is set if the comparison bit is enabled
2	STS_SW-GATE	Software gate status (set if SW gate active)
3	-	Reserved
4	-	Reserved
5	STS_GATE	Status of internal gate (set if internal gate active)
6	STS_COMP	Status of comparison bit
7	STS_C_DN	Status set in the case of backward counter direction
8	STS_C_UP	Status set in the case of forward counter direction
9	STS_CMP*	Status of comparator is set if the comparison condition is met. If the comparison is deactivated, (counter mode byte 1 = 0b000), the bit has no function.
10	STS_END*	Status set if final value has been reached
11	STS_OFLW*	Status set in the case of an overflow
12	STS_UFLW*	Status set in the case of an underflow
13	STS_ZP*	Status set in the case of a zero crossing
14	-	Reserved
15	-	Reserved

* The bits are set until reset with RES_SET (bit 6 control word)

2.6.4 EPM-S601 control word

Bit	Name	Function
0	-	Reserved
1	CTRL_COMP_SET	Enables the comparison bit
2	SW_GATE_SET	Sets the software gate
3	-	Reserved
4	-	Reserved
5	COUNTERVAL_SET	Set counter temporarily to the value in the preset value
6	RES_SET	Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a rising edge
7	-	Reserved
8	-	Reserved
9	CTRL_COMP_RESET	Inhibit comparison bit
10	SW_GATE_RESET	Resets the software gate
11 ... 15	-	Reserved

2.6.5 Status word EPM-S602

Bit	Name	Function
0	STS_SYNC	Reset was active
1	STS_CTRL_COMP	Is set if the comparison bit is enabled
2	STS_SW-GATE	Software gate status (set if SW gate active)
3	STS_RST	Status of the reset input
4	-	Reserved
5	STS_GATE	Status of internal gate (set if internal gate active)
6	STS_COMP	Status of comparison bit
7	STS_C_DN	Status set in the case of backward counter direction
8	STS_C_UP	Status set in the case of forward counter direction
9	STS_CMP*	Status of comparator is set if the comparison condition is met. If the comparison is deactivated, (counter mode byte 1 = 0b000), the bit has no function.
10	STS_END*	Status set if final value has been reached
11	STS_OFLW*	Status set in the case of an overflow
12	STS_UFLW*	Status set in the case of an underflow
13	STS_ZP*	Status set in the case of a zero crossing
14	-	Reserved
15	-	Reserved

* The bits are set until reset with RES_SET (bit 6 control word)

2.6.6 Control word EPM-S602

Bit	Name	Function
0	CTRL_SYNC_SET	Activation/deactivation of the counting signal: • TRUE↘FALSE: The input for the counting signal is deactivated and the current counter content is reset to 0. • FALSE↗TRUE edge: The input for the counting signal is activated.
1	CTRL_COMP_SET	Enables the comparison bit
2	SW_GATE_SET	Sets the software gate
3	-	Reserved
4	-	Reserved
5	COUNTERVAL_SET	Set counter temporarily to the value in the preset value
6	RES_SET	Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a rising edge
7	-	Reserved
8	CTRL_SYNC_RESET	Activation/deactivation of the zero track evaluation: • TRUE↘FALSE: The zero track evaluation is activated. • FALSE↗TRUE edge: The zero track evaluation is stopped. The counter continues to count irrespective of the zero pulse. For this purpose, bit 0 (CTRL_SYNC_SET) must be set to TRUE.
9	CTRL_COMP_RESET	Inhibits the comparison bit
10	SW_GATE_RESET	Resets the software gate
11 ... 15	-	Reserved

2.6.7 Status word EPM-S603

Bit	Name	Function
0	-	Reserved
1	-	Reserved
2	STS_SW-GATE	Software gate status (set if SW gate active)
3	-	Reserved
4	-	Reserved
5	STS_GATE	Status of internal gate (set if internal gate active)
6	-	Reserved
7	STS_C_DN	Status set in the case of backward counter direction
8	STS_C_UP	Status set in the case of forward counter direction
9	-	Reserved
10	-	Reserved
11	STS_OFLW*	Status set in the case of an overflow
12	STS_UFLW*	Status set in the case of an underflow
13	STS_ZP*	Status set in the case of a zero crossing
14	-	Reserved
15	-	Reserved

* The bits are set until reset with RES_SET (bit 6 control word)

2.6.8 Control word EPM-S603

Bit	Name	Function
0	-	Reserved
1	-	Reserved
2	SW_GATE_SET	Sets the software gate
3 ... 5	-	Reserved
6	RES_SET	Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a rising edge
7 ... 9	-	Reserved
10	SW_GATE_RESET	Resets the software gate
11 ... 15	-	Reserved

2.7 PWM module EPM-S620 – control and status word

Status word EPM-S620

Bit	Name	Function
0	–	Reserved
1	PWM status	0: PWM output stopped 1: PWM output active
2	Output status	0: Push/Pull output 1: Highside output
3 ... 15	–	Reserved

Control word EPM-S620

Bit	Name	Function
0 ... 1	–	Reserved
2	PWM response	• 0: Push/pull output Push/pull mode should be used if you need defined high/low levels for a rapid change. This is used with a low load especially if "highside" mode cannot move the output to low fast enough during a low status. With push/pull, the output is switched to ground with low active and to voltage with high active. • 1: Highside output In highside mode, the output switched to low remains in a state of uncertainty between ground and voltage. The load has to "pull" itself to ground. In highside mode, the switch is only made to high level active.
3 ... 7	Output status	Reserved
8	Start PWM output	0-1-edge: PWM output is starting
9	Stop PWM output	0-1 edge: PWM output stops
10 ... 15	–	Reserved

3 Configuring the I/O system in the »Engineer«

3.1 Establishing a connection between PC and I/O system 1000

The Engineering PC with installed »Engineer« and I/O system 1000 communicate via the system bus interface at the CANopen bus coupler module (EPM-S110).

- The system bus interface serves to e.g. exchange process data and parameter values between nodes.
- The system bus interface transfers CAN objects following the CANopen communication profile (CiA DS301, Version 4.02), which was created under the umbrella association of CiA (CAN in Automation) to conform with the CAL (CAN Application Layer).

The **system manual** for the I/O system 1000 contains detailed information on the communication via CANopen.

Lenze offers the following communication accessories for the connection to the PC:

Communication accessories	PC interface
PC system bus adapter 2173 incl. connection cable and voltage supply adapter • for DIN keyboard connection (EMF2173IB) • for PS/2 keyboard connection (EMF2173IBV002) • for PS/2 keyboard connection with electrical isolation (EMF2173IBV003)	Parallel interface (LPT port)
PC system bus adapter 2177 incl. connection cable (EMF2177IB)	USB (Universal Serial Bus)

Note!

- Please observe the documentation for the PC system bus adapter!
- For detailed information about the PC system bus adapter, please see the "CAN Communication Manual".

Preconditions:

- The 9-pole Sub-D terminal at the CANopen bus coupler module (EPM-S110) is connected to the LPT-Port (EMF2173IB) or to a free USB port (EMF2177IB) via the PC system bus adapter.
- The I/O system 1000 is supplied with DC 24 V for the control electronics.

3.2 Mapping the real station layout in the »Engineer«

In order to map the real station layout of the I/O system in the »Engineer«, you can either

- let the »Engineer« search for devices that can be accessed *online*
  - or -
- create a new project in the »Engineer« without going online in order to select the components manually from the catalog.

The procedure is described in the following subchapters.

3.2.1 Search for connected devices

Instead of starting with an empty project, you can let the »Engineer« search for devices that can be accessed *online*.

- After the search has been carried out, you can open the *Start-up wizard* and select the identified devices that are to be included in the project.

Note!

To read out the data, a communication link to the I/O system via system bus (CANopen) is required.

▶ Establishing a connection between PC and I/O system 1000 (📖 33)

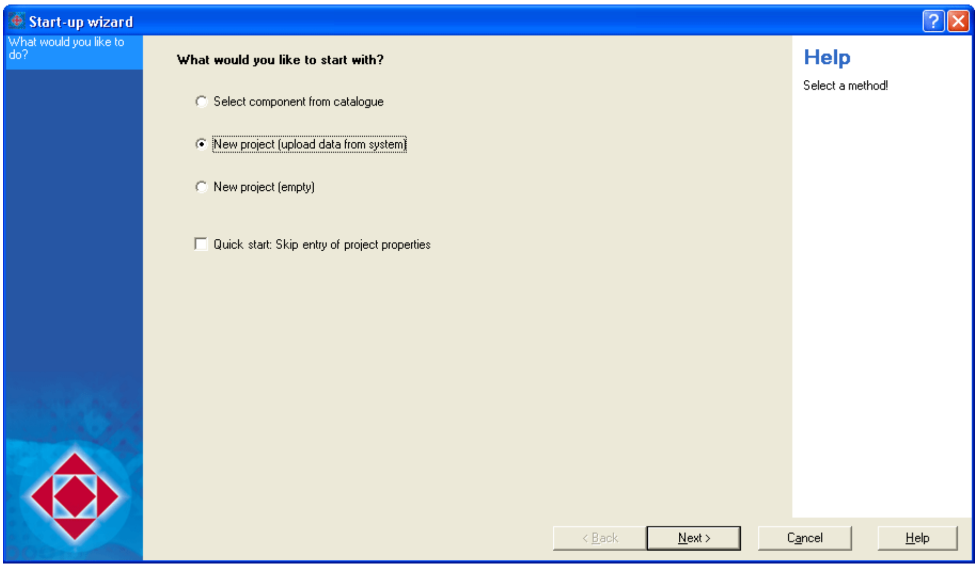
Stop!

In the fourth step "Read online data", the *start-up wizard* establishes automatically an online connection via the selected interface after the **Start Search** button has been pressed in order to be able to read out data from the connected devices.

How to create a new project for a connected device:

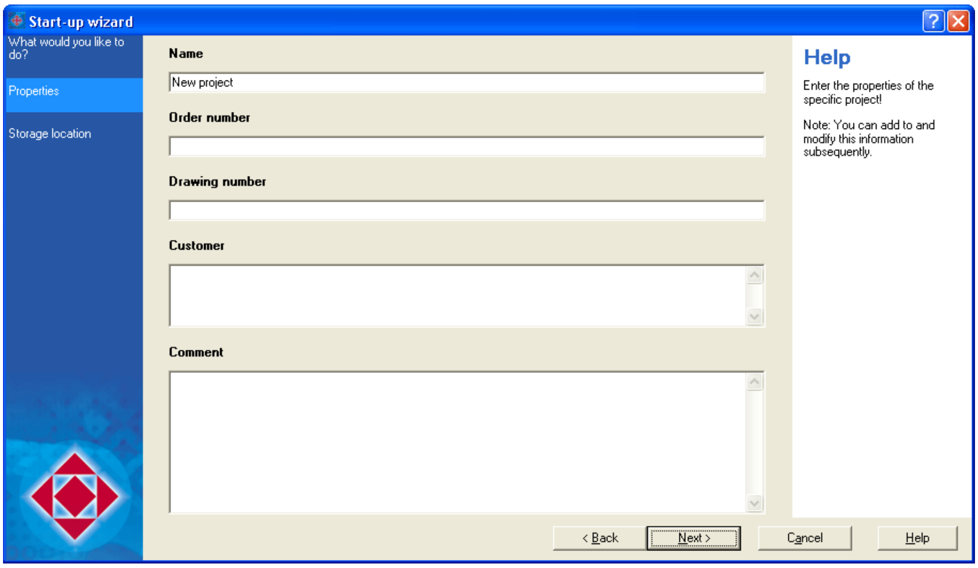
1. If the *start-up wizard* has not been displayed yet, click the icon or select the **File→New...** command to call the *start-up wizard*.

   Step 1 - What would you like to do?



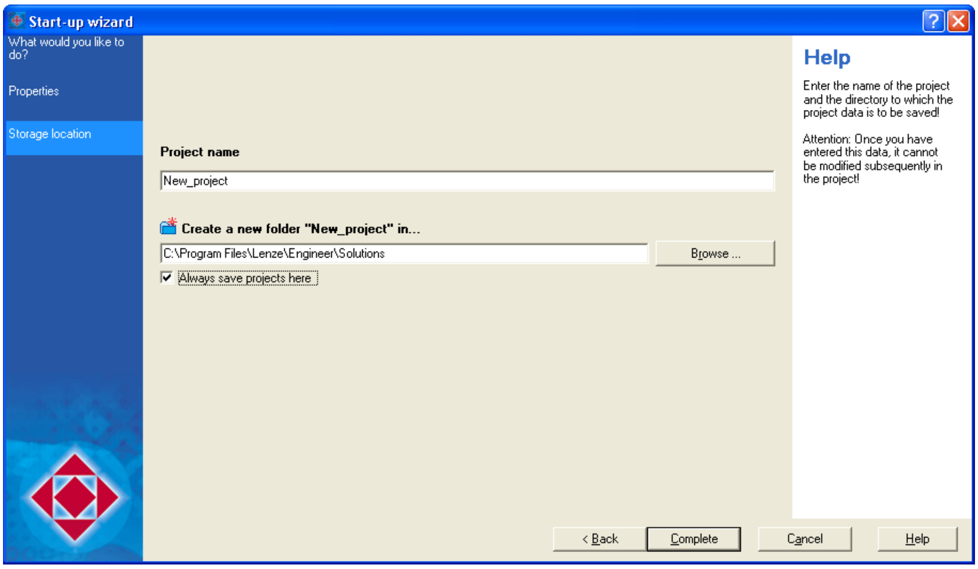
2. Select the **New project (upload data from system)** option.
3. Click the **Next** button.

   Step 2 - Properties



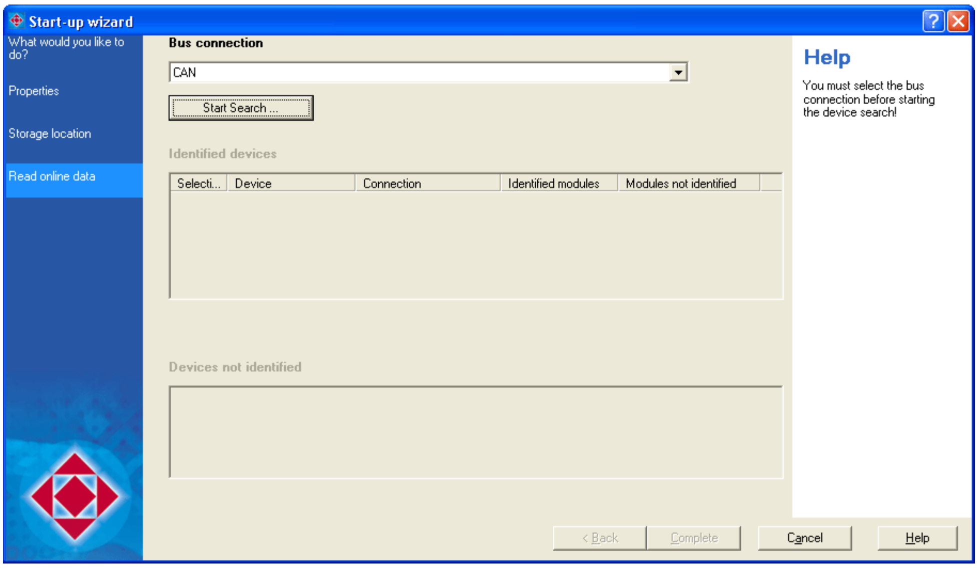
4. Enter the corresponding properties of the projects into the input fields (name, order number, etc.).
   • You can always change the properties subsequently.
   • Later you will find the name as project name in the *Project view*.
5. Click the **Next** button.

Step 3 - Storage location



6. The **Name** entered before is suggested as **project name**. If required, change the suggestion accordingly.
   - The project name is also the name for the project directory to be created.
7. Specify the storage location for the project directory in the **Directory** input field.
   - If the specified directory does not exist, it will be created.
   - As an alternative you can select an existing directory in your workspace environment via the **Browse...** button.
8. Click the **Complete** button.

Step 4 - Reading online data



9. Select the "CAN" entry from the **Bus connection** list field.

10. Click the **Start search** button to start the search for connected devices.
    - The dialog box entitled *Search for online devices* is displayed with status messages regarding the search.
    - The *Start-up wizard* automatically establishes an online connection via the selected interface in order to enable the read-out of data from the connected devices.
11. After the search has been completed, click the **Close** button to close the *Search for online devices* dialog box.
12. In the **Identified devices** list field, select the devices that are to be included in the new project to be created.
13. Click the **Complete** button.
    - The project is created with the selected settings. Below the project, all devices selected before are listed in a flat structure.
    - Now there already exists an online connection to the selected devices as you can see from the yellow highlighted device icon in the *Project view*.

Stop!

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

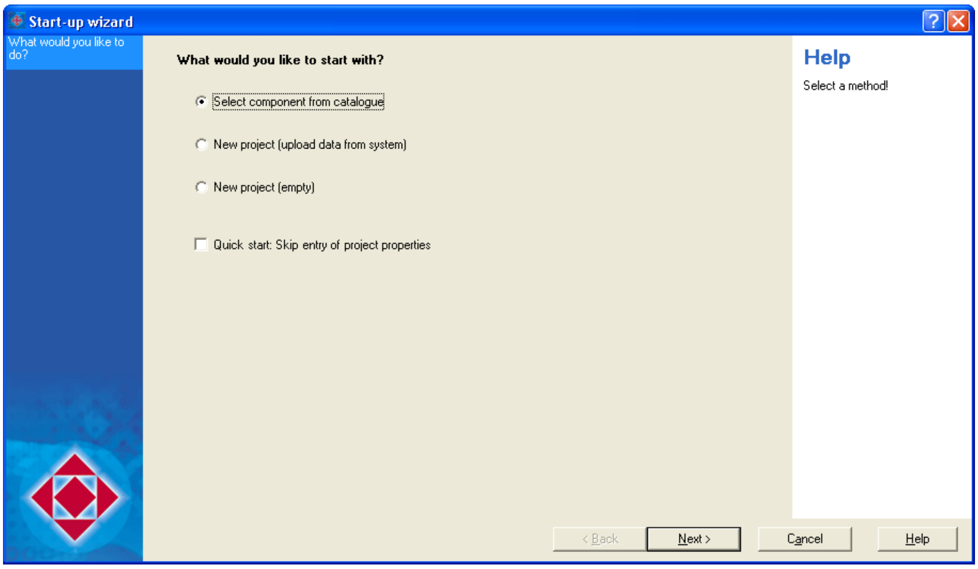
3.2.2 Creating a new project (select component from catalogue)

You can create a new project in the »Engineer« without being online. After having configured the project, go online with the I/O system 1000 to transfer parameters.

How to directly create a new project with components:

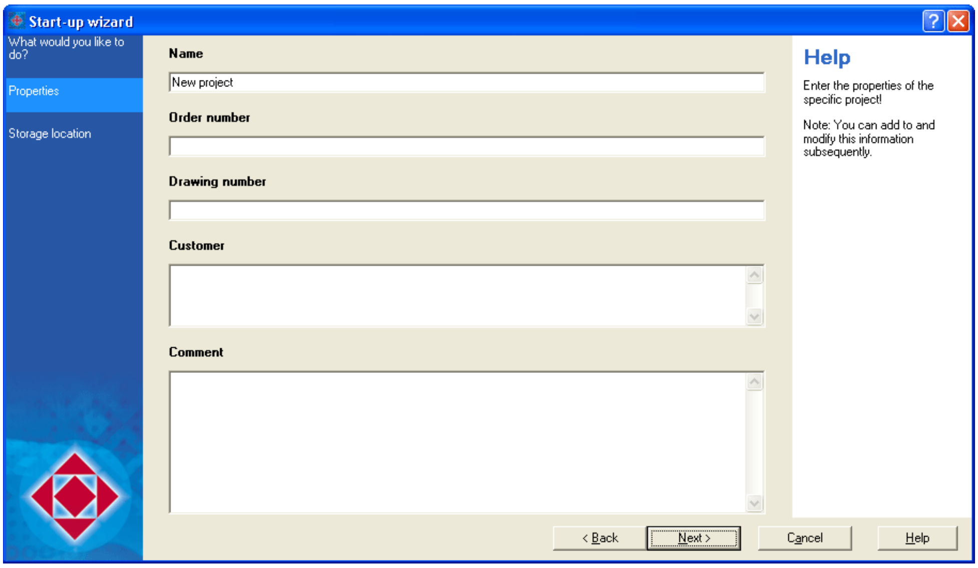
1. If the *start-up wizard* has not been displayed yet, click the icon or select the **File→New...** command to call the *start-up wizard*.

Step 1 - What would you like to do?



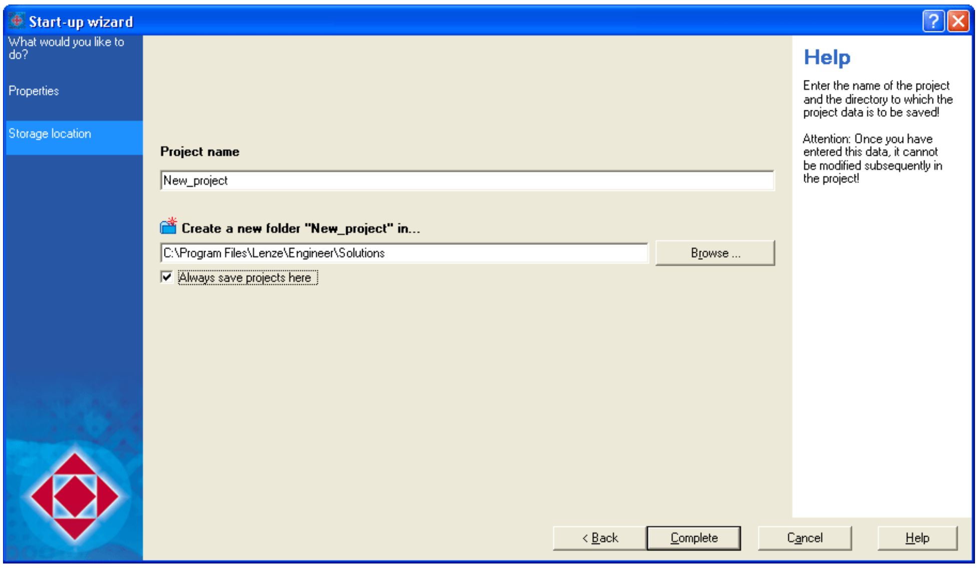
2. Select the **Select component from catalogue** option.
3. Click the **Next** button.

Step 2 - Properties



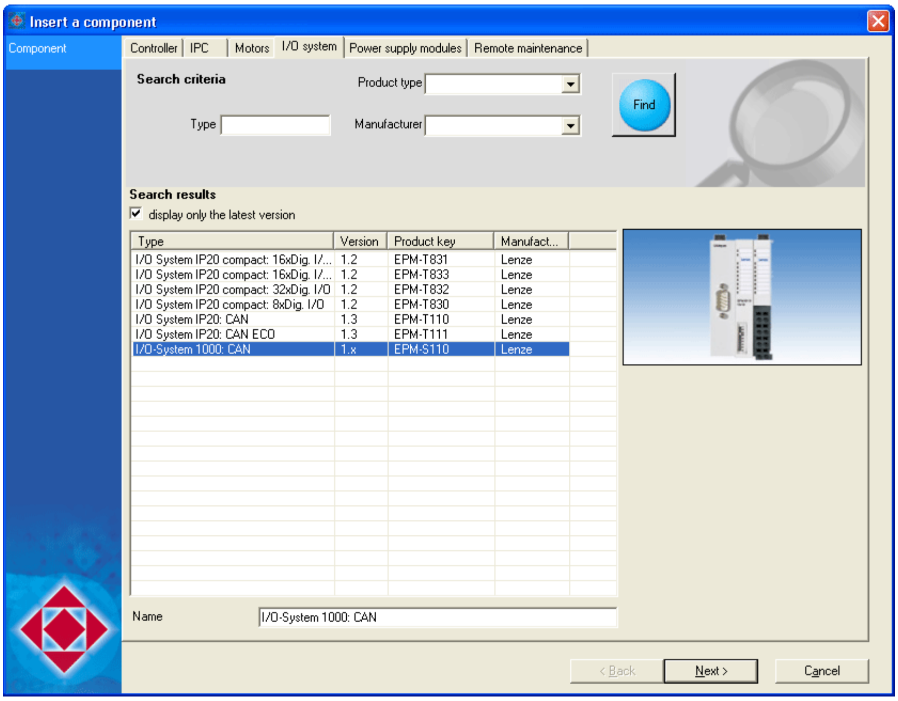
4. Enter the corresponding properties of the projects into the input fields (name, order number, etc.).
   - You can always change the properties subsequently.
   - Later you will find the name as project name in the *Project view*.
5. Click the **Next** button.

Step 3 - Storage location



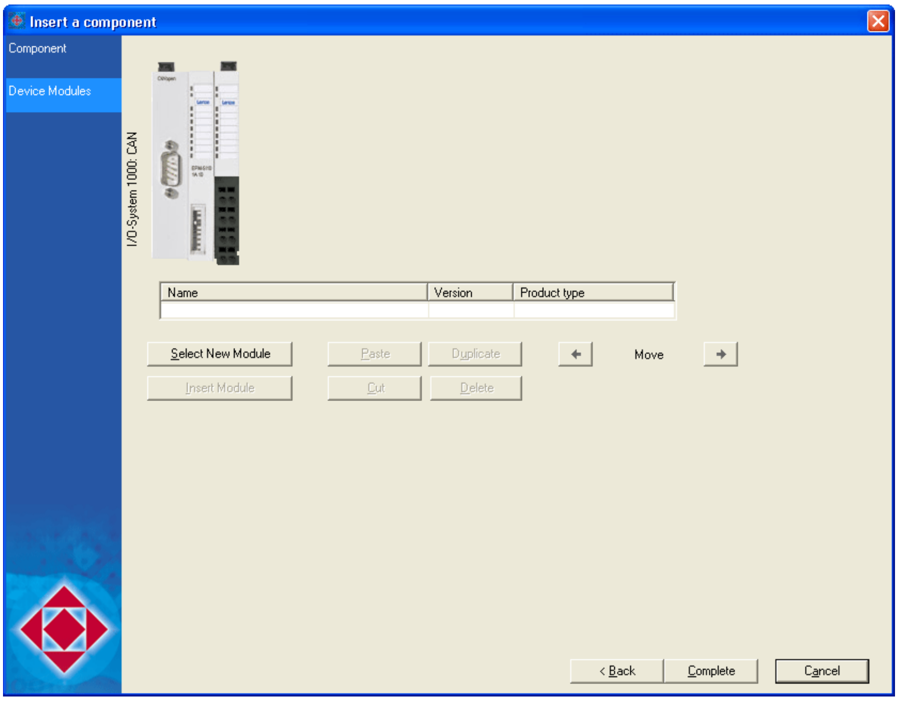
6. The **Name** entered before is suggested as **project name**. If required, change the suggestion accordingly.
   - The project name is also the name for the project directory to be created.
7. Specify the storage location for the project directory in the **Directory** input field.
   - If the specified directory does not exist, it will be created.
   - As an alternative you can select an existing directory in your workspace environment via the **Browse...** button.
   - If you select the **Always save projects here** option, the indicated directory will be the standard memory location for new projects.
8. Click the **Complete** button.
   - The **Insert a component** dialog box appears.

Step 4 - Component

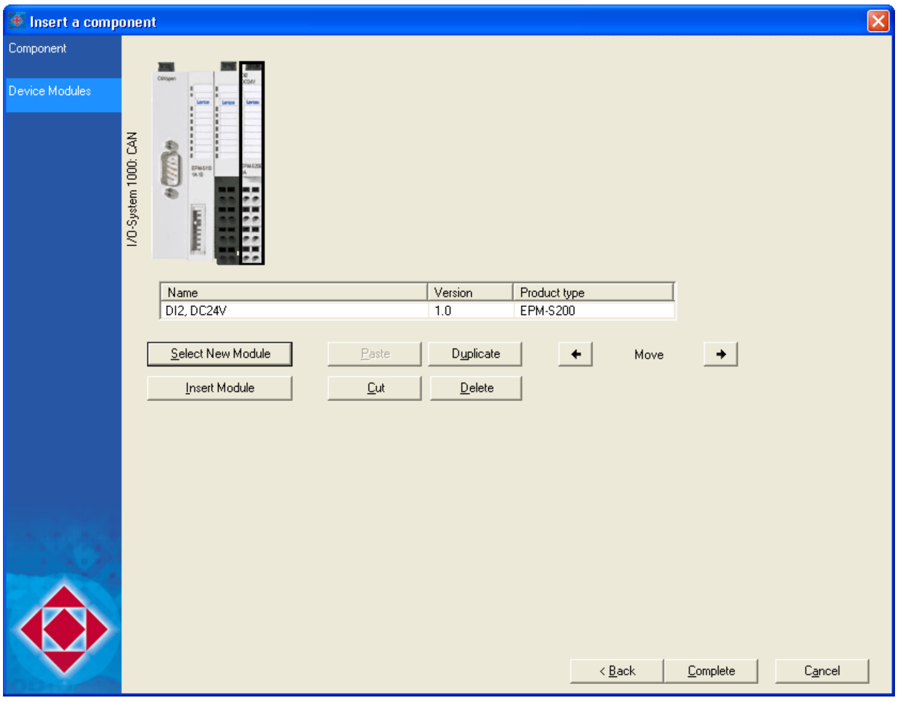


9. Select the **I/O system** tab.
   - The **Search results** list field displays all I/O systems available.
10. Select the "S110: CAN Bus coupler" module from the **Search results** list field.
    - More details and technical data of the selected module are displays on the right side.
11. Click the **Next** button.

Step 5 - Device modules

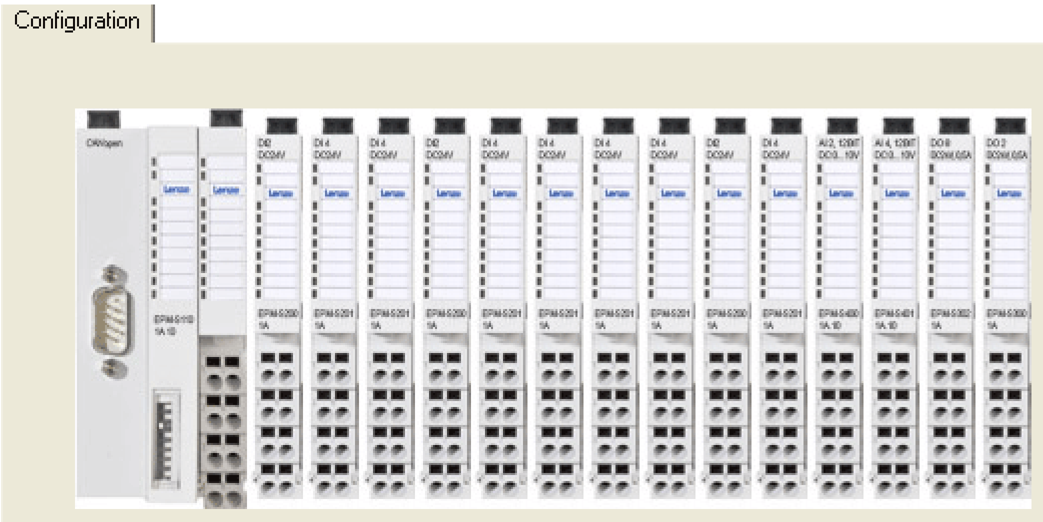


12. Use the **Select New Module** button to insert the first electronic module.



13. Add more electronic modules.
    - Use the **Select New Module** button to put the selected electronic module to the end.
    - Use the **Insert Module** button to insert the selected electronic module in front of the marked module.
    - Use the **Cut** button to remove the marked electronic module.
    - Use the **Paste** button to insert the electronic module cut before in front of the marked module.
    - Use the **Duplicate** button to copy the marked electronic module and insert it in front of it.
    - Use the **Delete** button to delete the marked electronic module.
    - Use the 🡐🡒 buttons to move the marked electronic module to the right or left.
14. Click the **Complete** button.

Then, an I/O system 1000 may look as follows:



- You can now make the required parameter settings offline.
  ▶ Parameterising I/O modules in the »Engineer« (🕮 45)
- Then, go online to transmit the parameter settings made in the »Engineer« to the I/O system 1000.
  ▶ Setting the communication path and going online (🕮 43)

3.3 Setting the communication path and going online

Note!

The communication settings need to be made only once when the connection to the device is established for the first time. If the **Go online** command is executed again, the *Communication path* dialog box will not be displayed again and an online connection will be established immediately via the bus connection set for the device.

To change a communication path that has already been set, go to the **Online** menu and execute the **Set communication path and go online** command.

How to configure an online connection via system bus (CANopen):

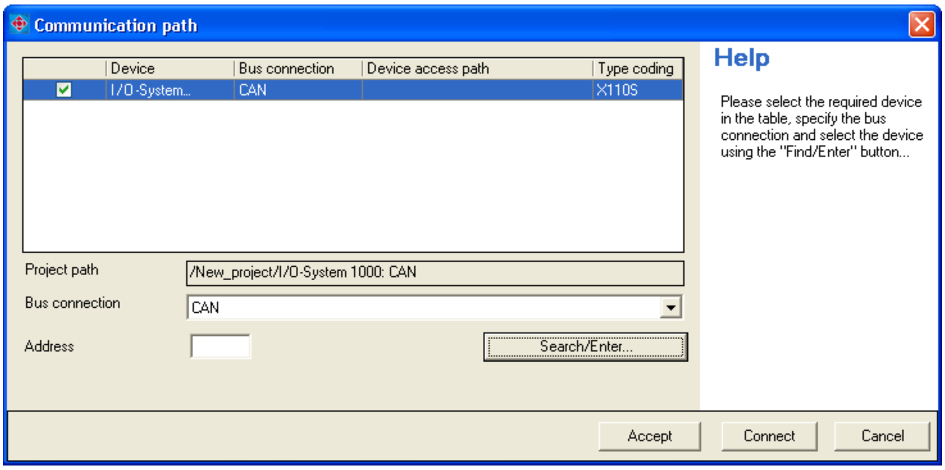
1. Go to the *Project view* and select the I/O system 1000 to be connected online:

2. Select the menu command **Online→Set communication path and go online**.

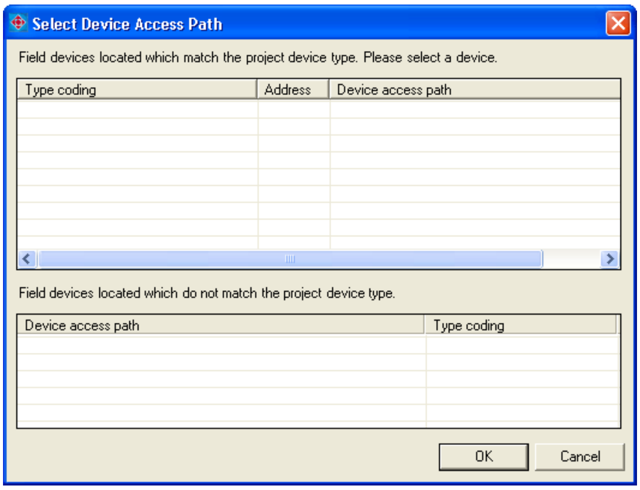
   If the changes you have made on the project have not been accepted yet, first a query on whether an update is to be carried out is effected.

   If an update is to be carried out:
   - Click **Yes** to open the *Build project* dialog box.
   - Click **Build** in the *Build project* dialog box to update the changed project elements.
   - When the update is completed, a note appears whether the update was successful.

3. In the *Communication path* dialog box, select the "CAN" entry from the **Bus connection** list field:



4. Click the **Search/Enter...** button to check the bus connection selected from the **Bus connection** list field.
   - The *Select device access path* dialog box is shown:



5. Select the corresponding device from the **Field devices located** list field.
6. Press **OK**.
   - The *Select device access path* dialog box is closed.
   - In the *Communication path* dialog box, the **Device access path** column displays the corresponding device access path (e.g. "can:/dev1/").
7. In the *Communication path* dialog box, click the **Connect** button to establish an online connection to the I/O system.

In the *Project view*, a yellow icon indicates the online connection to the I/O system 1000:

Stop!

If you change parameters in the »Engineer« while an online connection to the device is established, the changes are directly accepted in the device!

The and icons provide an easy possibility to first establish and then disconnect an online connection to the I/O system 1000.

When an online connection has been established, the »Engineer« displays the current parameter settings of the I/O system 1000 with a yellow background colour.

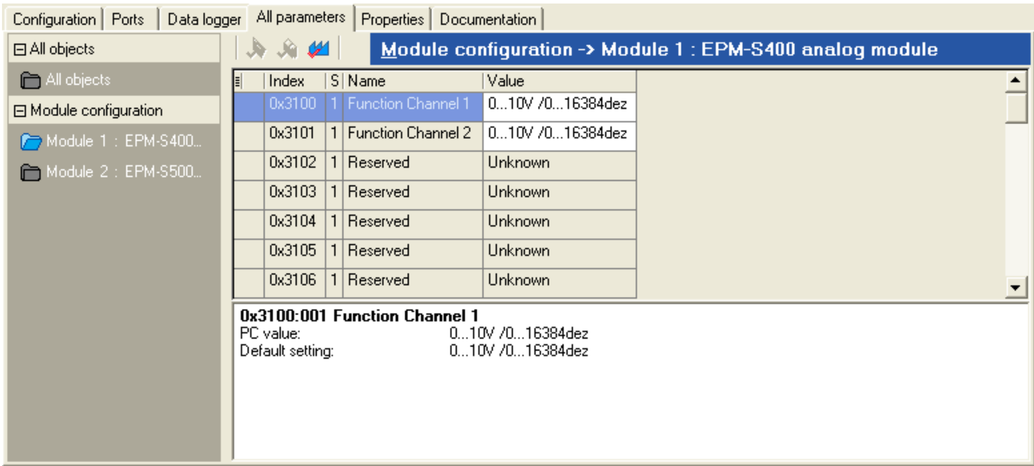
3.4 Parameterising I/O modules in the »Engineer«

This chapter uses examples to describe how to parameterise I/O compound modules in the »Engineer«.

Detailed information on the function of the single I/O modules can be found in the **system manual** for the I/O system 1000.

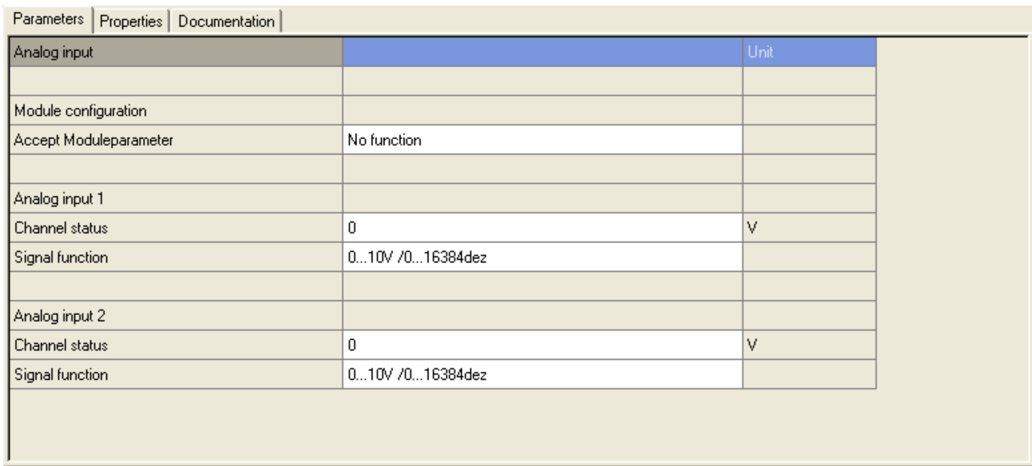
If you select the I/O system 1000 in the *project view* of the »Engineer«, the *Workspace* displays the settings and features of the I/O system.

- The **All parameters** tab displays the parameters of the bus coupler module and all attached I/O compound modules in a hierarchical structure.
- The following example shows the setting parameters of an "EPM-S400":



As an alternative, you can have the settings and features of a single I/O compound module displayed by selecting the corresponding I/O compound module in the *Project view*.

- Use the **Parameter** tab to set the parameters required for the selected I/O compound module (for this example, an "EPM-S400" has been selected):



Acceptance of parameter changes:

In order that parameter changes for parameterisable I/O compound module are accepted, they have to be confirmed. This can be done as follows:

A. Go to the **All parameters** tab of the I/O system 1000:
Set the index 0x31FF to the selection "255: Accept module parameters" in the **All objects** category.

B. Go to the **Parameters** tab of the I/O compound module:
Set the selection "255: Accept module parameters" in the **Accept module parameters** line.

3.4.1 Save parameter settings and station structure

Parameter | Name: **0×1010/1 | Save All Parameters**		
Parameter settings and station structure are saved to the EEPROM of the bus coupler.		
Selection list (Lenze setting printed in bold)		**Info**
0	**No function**	Function acc. to CANopen (communication protocol DS301/DS401)
1702257011	Save	

3.4.2 Load Lenze setting

If the read-in station structure is not identical to the saved station structure after the I/O system 1000 is switched on, the CAN bus coupler module reports an error ("SF" LED is blinking).

Parameter | Name: **0×1011/1 | Restore All Parameters**		
The parameter settings are reset to the Lenze setting in the EEPROM of the bus coupler.		
Selection list (Lenze setting printed in bold)		**Info**
0	**No function**	Function acc. to CANopen (communication protocol DS301/DS401) The Lenze setting is accepted by switching the supply voltage on and off. All settings that have been saved before are deleted from the EEPROM of the CAN bus coupler.
1684107116	LOAD	

4 Configuring the I/O system in the »PLC Designer«

This chapter describes how to set up communication with the »PLC Designer« between the Engineering PC and the I/O system 1000.

If the I/O system 1000 is used in line with the Lenze "Controller-based Automation" system, only the ...

- EtherCAT communication (📖 323),
- CANopen communication (📖 78),
- PROFIBUS communication (📖 215),
- PROFINET communication (📖 464),

are supported.

More information on the bus systems and configuration can be found in these **communication manuals**:

- Controller-based Automation EtherCAT®
- Controller-based Automation CANopen®
- Controller-based Automation PROFIBUS®
- Controller-based Automation PROFINET®

4.1 Connecting the I/O system to the Lenze Controller

- Plug the Controller and modules of the I/O system 1000 onto the fixing rail.
- Always arrange the modules from left to right starting with the Controller directly followed by a power supply module EPM-S701 to the right or an I/O bus coupler module.
- The modules must always be installed directly next to each other.
- Free slots between the modules are not permissible because otherwise the backplane bus would be interrupted.
- The side contacts of the last module always must be covered with the supplied contact cover. Otherwise, the modules may be damaged by short circuit or static discharge.

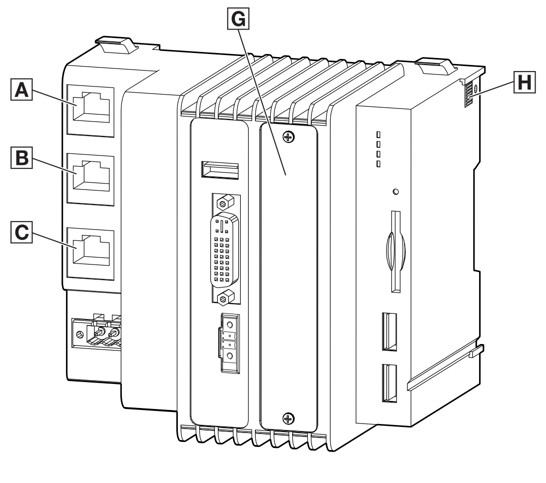
Mounting instructions of the I/O system and the Controller

Here you will find detailed information on how to install the devices.

Observe the safety instructions.

Controller reference manual - Parameterising and programming

Detailed information on the Lenze Controllers is provided here.



A	Ethernet interface "LAN 1a"
B	Ethernet interface "LAN 1b"
C	EtherCAT interface
G	Slot for optional communication card (MC-Card)
H	Contacts for backplane bus

[4-1] **Example:** Interfaces on the Controller 3231 C

Connection of the Engineering PC

Connect the Ethernet interface "LAN 1a" to the Engineering PC using a standard Ethernet network cable

Connection via EtherCAT

Connect the RJ45 port on the EtherCAT bus coupler module (EPM-S130) to the EtherCAT interface of the Controller using a standard Ethernet network cable.

Connection to an optional communication card (MC-Card)

For all other fieldbus systems, optional communication cards are available for the Controllers.

The interfaces of the communication card have to be connected to a corresponding bus coupler module.

4.2 Establishing communication with the Lenze Controller

Connect the Engineering PC with the controller via a network cable, as the »PLC Designer« accesses the controller via Ethernet.

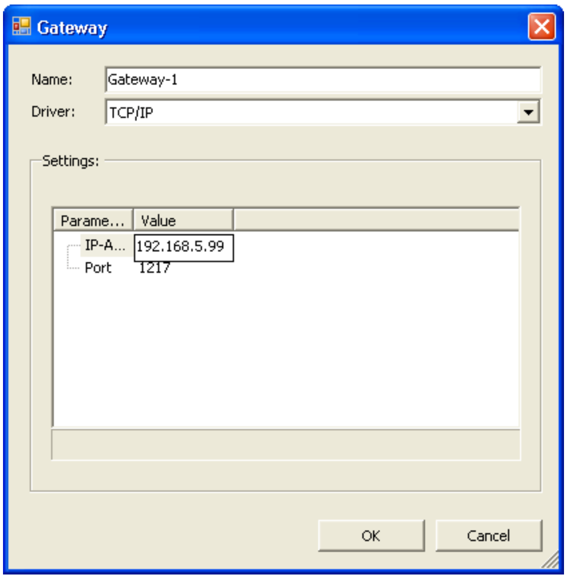
Make the IP settings with the »PLC Designer«.

Predefined IP addresses

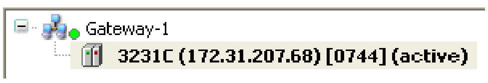
- Engineering PC: 192.168.5.100
- Controller: 192.168.5.99

How to check the communication settings:

1. Double-click the Lenze Controller in the *Device view* in order to open the settings and properties of the device in the *Editor view*.
2. Make the desired settings on the **Communication settings** tab.
   - Click the **Add gateway** button to insert a gateway.
   - Enter the desired IP address of the controller.



3. Click **OK** to add the controller as gateway.
4. Double-click the channel below the gateway in the *Device view* which is to bet set as active path for control.
   - Thus, all communication actions directly refer to this channel.
   - The currently active channel is shown in the *device view* in **bold** and is marked with the "(active)" addition:



   - A channel represented in *italics* is set as active path but has not been found during the last network scan.

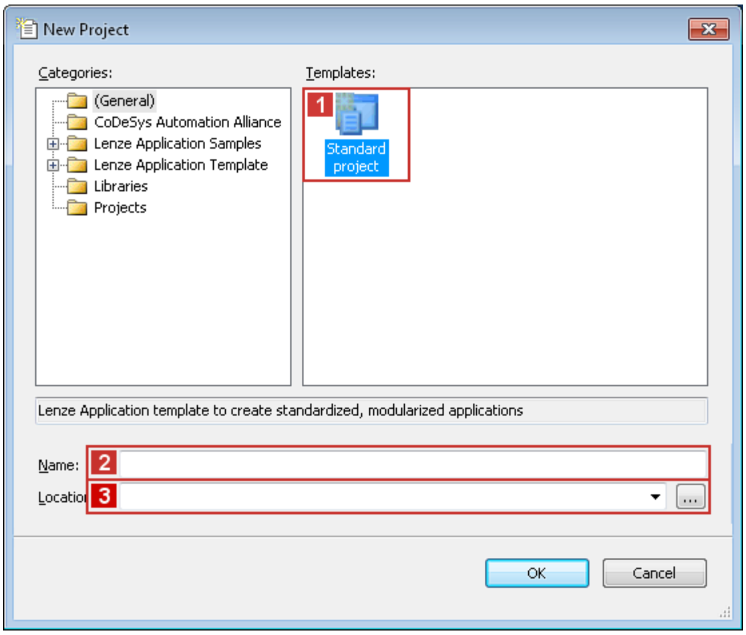
4.3 Mapping the real station structure in the »PLC Designer«

4.3.1 Create a new project in the »PLC Designer«

You can create a new project in the »PLC Designer« without having to establish an online connection. After having configured the project, go online with the I/O system 1000 to transfer parameters.

How to create a new project in the »PLC Designer«:

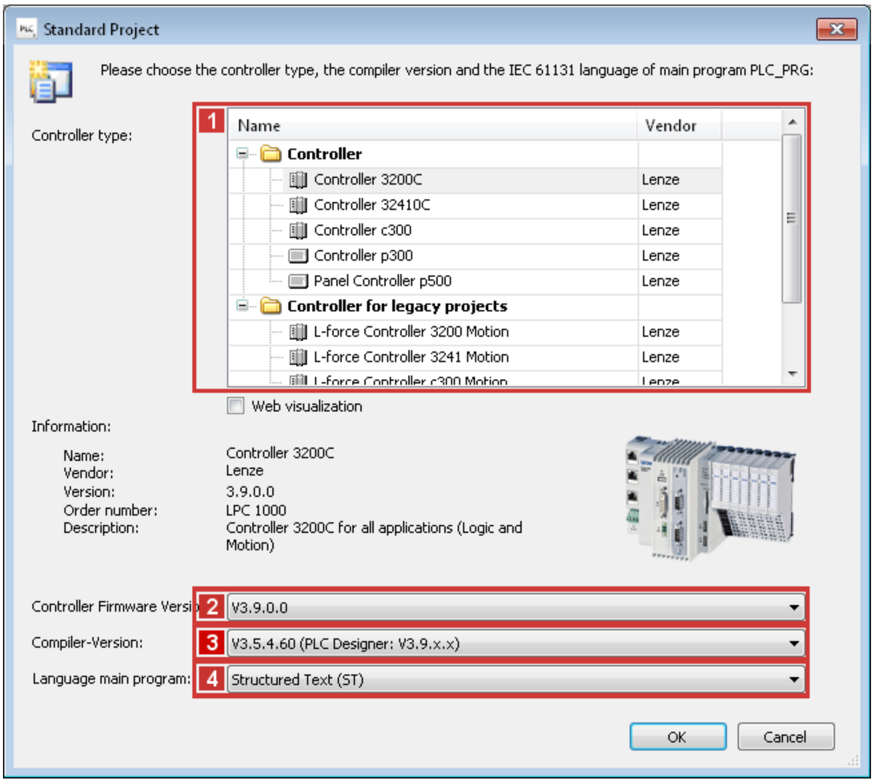
1. Select the menu command **File→New project** to create a new »PLC Designer« project.
2. Select the 1 "standard project" template in the *New project* dialog window.



   - Enter a name for the »PLC Designer« in the 2 **Name** input field.
   - Select the desired memory location under 3 **Location**.

3. Confirm the entries by clicking **OK**.

4. Go to the *Standard project* dialog window and select the target system from the **1** **Controller type** selection list.



Further optional project settings

2 Selection of the Controller firmware version

3 Selection of the compiler version

4 Selection of the programming language:
- Sequential function chart (AS)
- Instruction list (AWL)
- Continuous Function Chart (CFC)
- Function block diagram (FUP)
- Ladder diagram (KOP)
- Structured text (ST)

5. Confirm the selection by clicking **OK**.

Next steps:

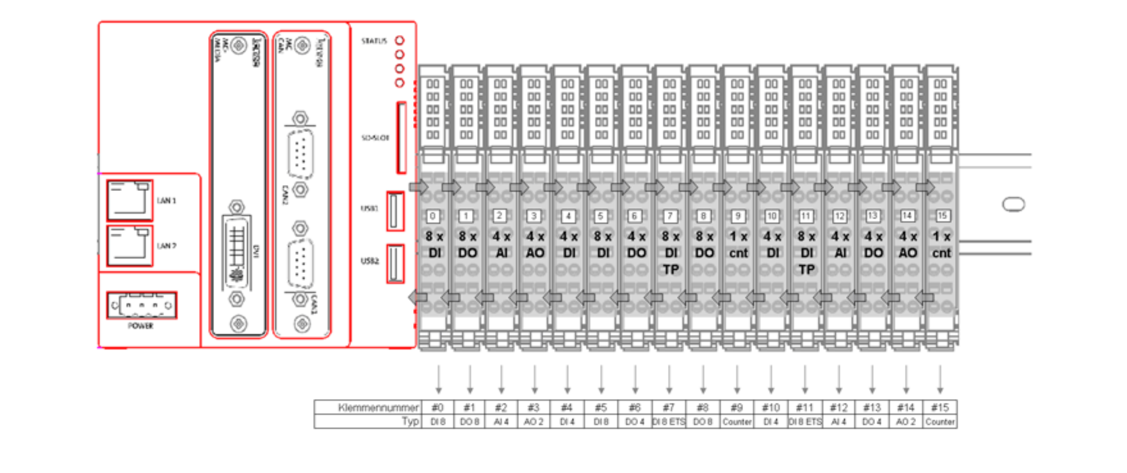
The following chapter use examples to describe how to add the available I/O system depending on the fieldbus used to the *Device view* of the »PLC Designer«:

▶ Configuring the I/O modules at the backplane bus of the Lenze Controller
▶ Configuring I/O modules on the EtherCAT bus coupler module
▶ Configuring I/O modules on the CANopen bus coupler module

4.3.2 Configuring the I/O modules at the backplane bus of the Lenze Controller

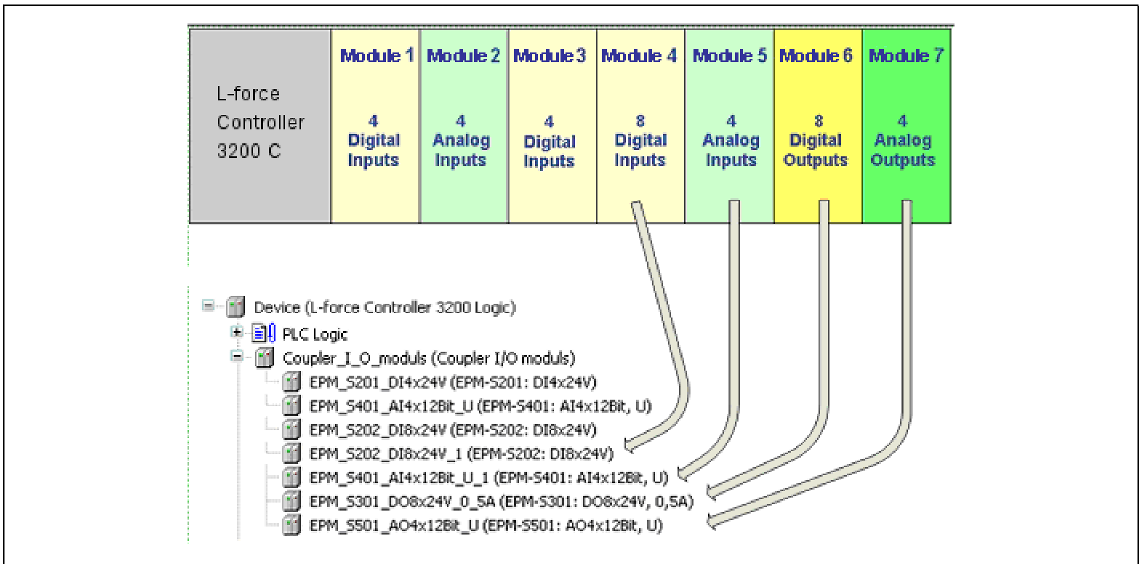
The Lenze Controller consists of the following components:

- I/O bus coupler module: controller with a PLC control in the form of a soft PLC.
- I/O compound modules: Up to 64 modules which are connected to the controller via the Lenze backplane bus.



[4-2] Schematic hardware structure of the Lenze Controller with added I/O compound modules

In order to access the modules by a PLC program (read input signals and write output signals), the real station layout, i.e. the physical arrangement of the I/O compound modules has to be mapped in the *Device view* of the »PLC Designer«:



[4-3] Example: Mapping of the physical topology (module 1 to module 7) in the device view in the »PLC Designer«

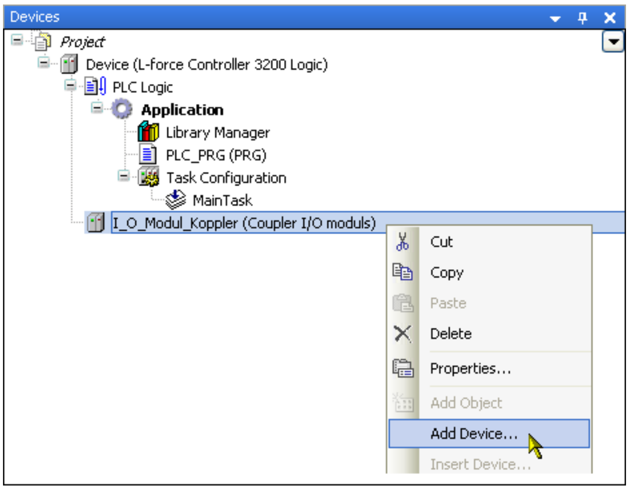
The following step-by-step instructions describe the manual mapping of the available station layout in the *device view* of the »PLC Designer«.

Tip!

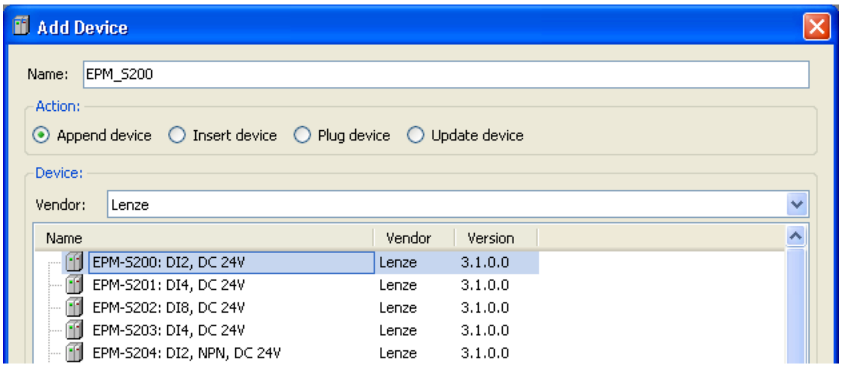
When an online connection has been established to the Lenze Controller, you can also start an automatic search for I/O compound modules available on the backplane bus of the Controller by executing the **Start Search** command in the *context menu* of the "I_O_Modul_Koppler.

How to map the station layout of the »PLC Designer«:

1. Execute the **Add device...** command in the *context menu* of the "I_O_Modul_Koppler:

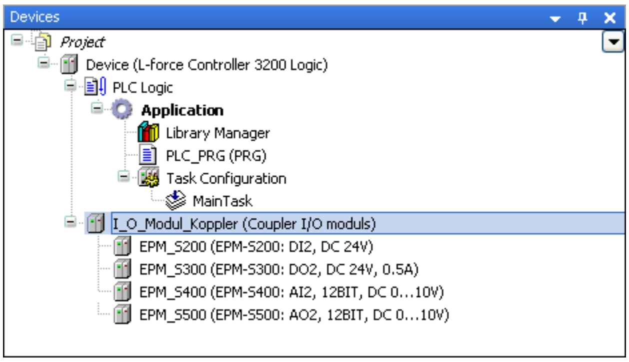


2. Select the physically first I/O compound module in the *Add device* dialog box (for this example, an "EPM-S200" has been selected):



3. Click **Add device** to add the selected device to the "I_O_Modul_Koppler".
   - The *Add Device* dialog box remains open.

4. Proceed as described before to add all other available I/O compound modules of the *Device view*.
   - Sample representation:



5. Click the **Close** button to close the *Add Device* dialog box again.

Tip!

- If configured correctly and after "going online" and starting the PLC, all modules in the *device view* are marked with the symbol.
- In the event of an error, i.e. if the configuration deviates from the physical arrangement, the modules are marked with the symbol.

4.3.3 Configuring I/O modules on the EtherCAT bus coupler module

Note!

Observe the following conditions before creating an EtherCAT configuration in the »PLC Designer«:

- Only **I/O compound modules EPM-Sxxx from hardware version 1B** are supported.
- The sequence of the EtherCAT slaves in the *device view* must correspond to the physical arrangement of the EtherCAT topology.
- In order that the system works properly, do not use any end terminals when setting up the system configuration in the *device view*.
- Select the cycle times according to the technical data, from 1 ... 10 ms.

The following step-by-step instructions describe the manual mapping of the available station layout in the *device view* of the »PLC Designer«.

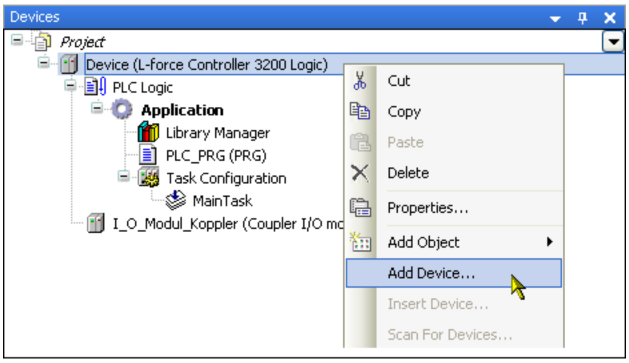
Tip!

When an online connection has been established to the Lenze Controller, you can also start an automatic search for available devices via fieldbus scan by executing the **Start Search** command in the *context menu* of the EtherCAT master after adding the EtherCAT master (step 3).

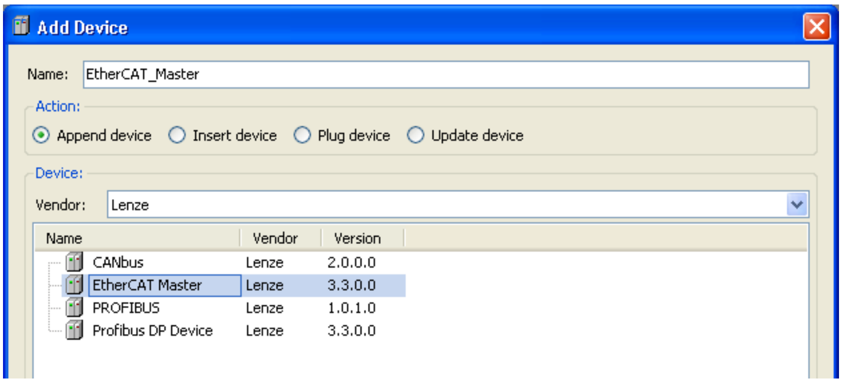
How to map the station layout of the »PLC Designer«:

Adding the EtherCAT master

1. Go to the *context menu* of the Controller and execute the **Add device...** command:



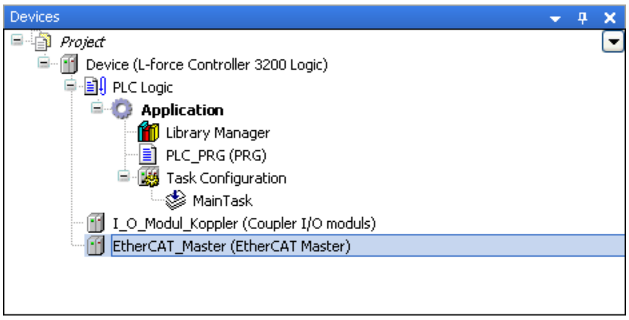
2. Select the "EtherCAT master" device in the *Add Device* dialog box:



3. Click **Add Device** to add the selected device to the Controller.
   - The *Add Device* dialog box remains open.

Adding the EtherCAT bus coupler module (slave) to the EtherCAT master

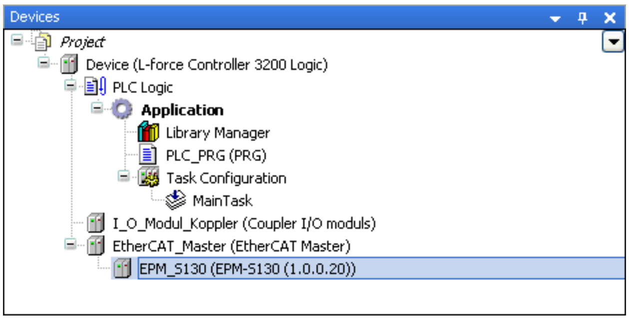
4. Select the "EtherCAT_Master" device in the *device view*:



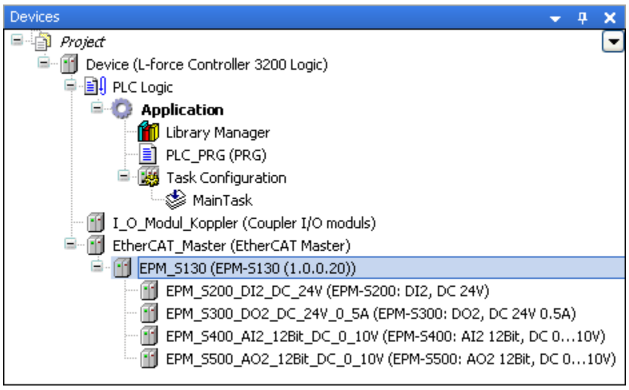
5. Now select the "EPM-S130" device (EtherCAT bus coupler module) in the still open *Add Device* dialog box.
6. Click the **Add Device** button to add the selected device to the EtherCAT master.
   - The *Add Device* dialog box remains open.

Adding I/O compound modules to the EtherCAT bus coupler module

7. Select the "EPM_S130" device in the *device view*:



8. Select the physically first I/O compound module in the *Add Device* dialog box and add it to the *device view* by clicking the **Add device** button.
9. Proceed as described before to add all other available I/O compound modules of the *Device view*.
   - Sample representation:



10. Click the **Close** button to close the *Add Device* dialog box again.

Tip!

- If configured correctly and after "going online" and starting the PLC, all modules in the *device view* are marked with the symbol.
- In the event of an error, i.e. if the configuration deviates from the physical arrangement, the modules are marked with the symbol.

4.3.4 Configuring I/O modules on the CANopen bus coupler module

Note!

The configuration of a Lenze Controller in a CANopen network must be created in the »PLC Designer«, because the complete configuration is written to the connected slaves when the Controller is started. This process overwrites the previous slave settings.

The following step-by-step instructions describe the manual mapping of the available station layout in the *device view* of the »PLC Designer«.

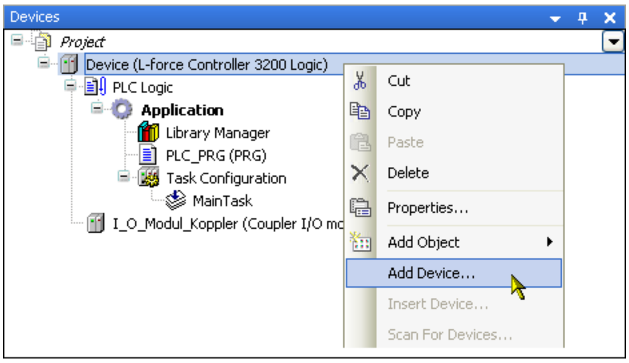
Tip!

When an online connection has been established to the Lenze Controller, you can also start an automatic search for available devices via fieldbus scan by executing the **Start Search** command in the *context menu* of the CANopen Manager after adding the CANopen Manager (step 6).

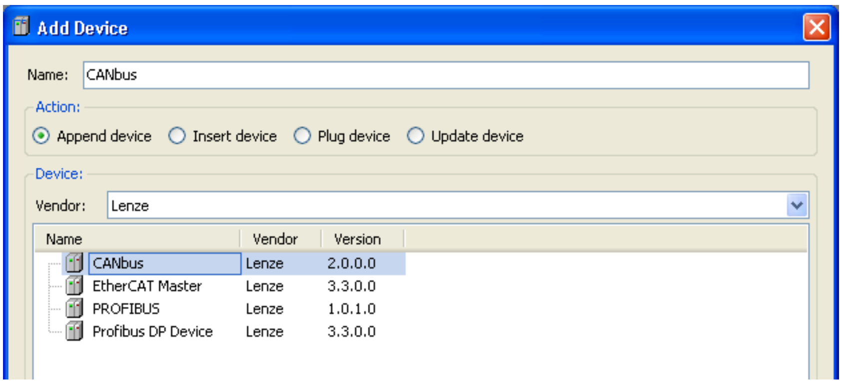
How to map the station layout of the »PLC Designer«:

Adding the CANbus

1. Go to the *context menu* of the Controller and execute the **Add device...** command:



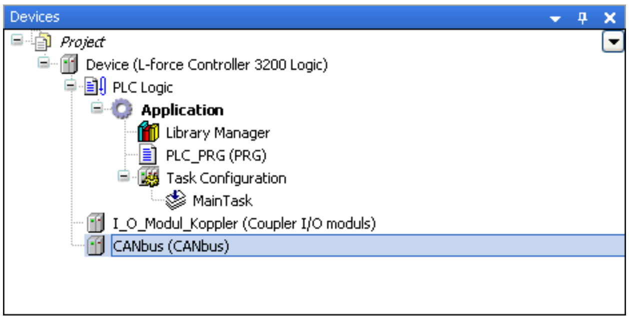
2. Select the "CANbus" device in the *Add Device* dialog box:



3. Click **Add Device** to add the selected device to the Controller.
   - The *Add Device* dialog box remains open.

Adding the CANopen Manager to the CANbus

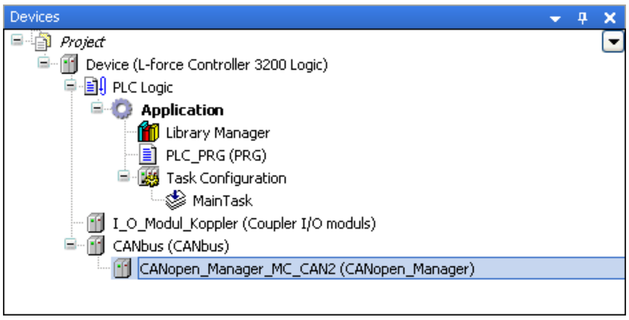
4. Select the "CANbus" device in the *device view*:



5. Now select the "CANopen Manager MC-CAN2" in the still open *Add Device* dialog box.
6. Click **Add Device** to add the selected device to the CANbus.
   - The *Add Device* dialog box remains open.

Adding the CANopen bus coupler module (slave) to the CANopen Manager

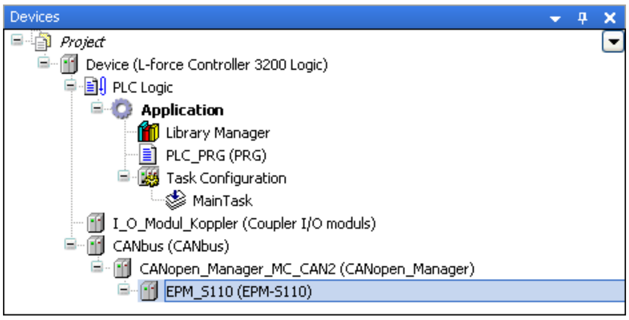
7. Select the "CANopen_Manager_MC_CAN2" device in the *device view*:



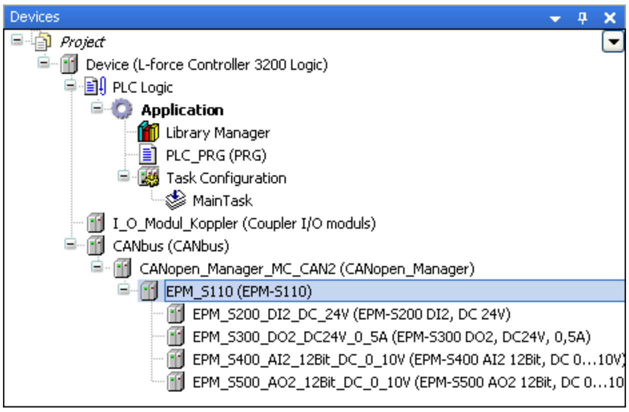
8. Now select the "EPM-S110" device (CANopen bus coupler module) in the still open *Add Device* dialog box.
9. Click **Add Device** to add the selected device to the CANopen Manager.
   - The *Add Device* dialog box remains open.

Adding I/O compound modules to the CANopen bus coupler module

10. Select the "EPM_S110" device in the *device view*:



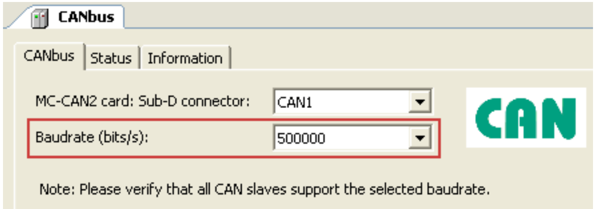
11. Select the physically first I/O compound module in the *Add Device* dialog box and add it to the *device view* by clicking the **Add device** button.
12. Proceed as described before to add all other available I/O compound modules of the *Device view*.
    - Sample representation:



13. Click the **Close** button to close the *Add Device* dialog box again.

CAN baud rate setting

14. Double-click the "CANbus" device in the *device view* to open its settings.
15. Set the baud rate in the **CANbus** tab:



Note!

- The baud rate set in the »PLC Designer« overwrites the baud rate set for the field devices via »WebConfig«/»EASY Starter«/»Engineer«/»Global Drive Control«.
- In a CANopen network, set the same baud rate for all nodes.

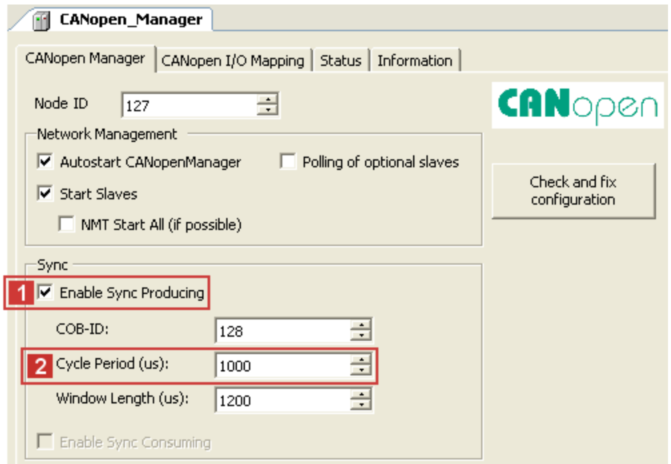
Setting sync generation

Sync generation is required if ...

- at least one PDO with sync-controlled processing is used on the bus;
- the applications are to run in synchronism in several field devices;
- Motion devices are to be operated on the fieldbus.

In order to activate the Sync generation:

16. Double-click the "CANopen_Manager_MC_CAN2" device in the *device view* to open its settings.
17. Go to the **CANopen_Manager** tab:
    - 1 Activate the **Sync generation** option.
    - 2 Set the sync cycle time in the **Cycle Period** entry field.



Tip!

- If configured correctly and after "going online" and starting the PLC, all modules in the *device view* are marked with the symbol.
- In the event of an error, i.e. if the configuration deviates from the physical arrangement, the modules are marked with the symbol.

4.3.4.1 Special features of the CANopen bus coupler module (EPM-S110)

Note!

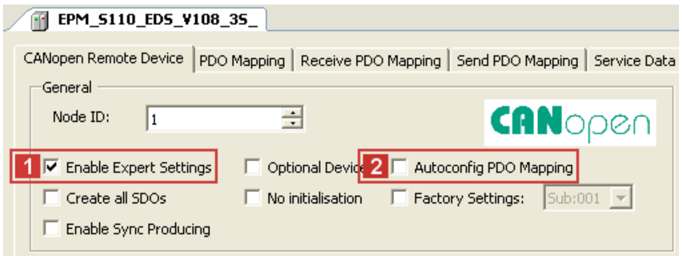
Before inserting I/O compound modules below a CANopen bus coupler module (EPM-S110) in the »PLC Designer«, close the tab for the **EPM-S110**.

- When I/O compound modules are inserted while the tab is open, no I/O image is displayed for the modules.
- When all I/O compound modules have been inserted, the tab for the **EPM-S110** can be opened again.

If the counter modules (EPM-S600 ... S604) are used at the CANopen bus coupler module (EPM-S110), the following sequence has to be complied with during the configuration.

How to configure the EPM-S600 ... S604 counter modules:

1. Add the CANopen bus coupler module (EPM-S110) to the *Device view*.
2. Add I/O discs and counter discs.
3. Double-click the "EPM_S110" device in the *device view* S110to open its settings.
4. Go to the **CANopen Remote Device** tab:
   - 1 Activate the **Expert Settings** option.
   - 2 Deactivate the **Autokonfig. PDO Mapping** option.



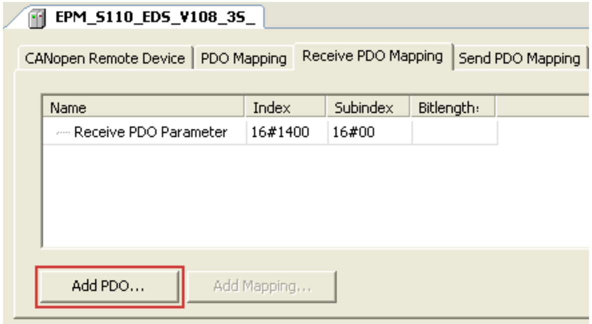
5. Execute the menu command **Window→Close All Editors**.

Adding the PDO configuration for the counter discs.

The following example uses a send PDO and a mapping element to describe the basic configuration process.

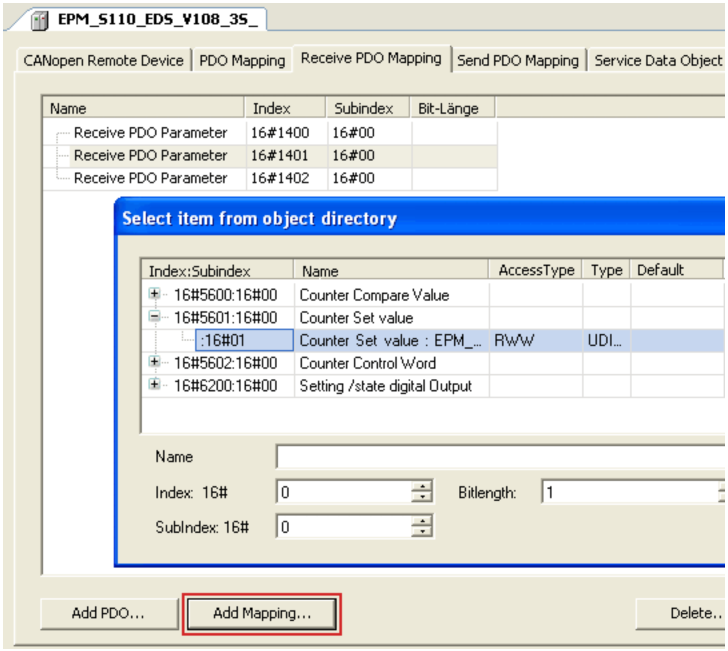
Proceed accordingly when configuring the transmit PDOs (tab **Send PDO Mapping**).

6. Double-click the "EPM_S110" device in the *device view* S110to open its settings.
7. Click **Add PDO...** on the **Receive PDO Mapping** tab to add receive PDOs.



- Depending on the counter discs used, one or more PDOs must be added in send and receive direction.
- Please refer to the module documentation to see which additional elements of the process image of the counter modules are required for operation.

8. Select added PDOs and click **Add Mapping...**.
   - Select the corresponding settings in the dialog box that appears.



Note!

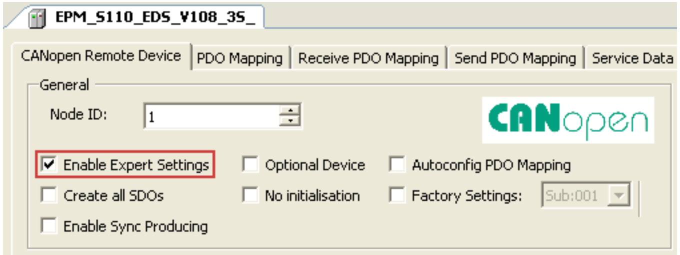
If **Autoconfig. PDO Mapping** (see step 3) is reactivated after adding the manual configuration, the entire configuration that has been added manually will be deleted.

4.3.4.2 Restart

When the program starts, the control initialises the I/O system. It changes to the "Operational" status. While the control initialises the I/O system, the »PLC Designer« must not be online on the same SDO channel.

There are different methods for restarting the I/O system: automatically with the »PLC Designer«, or automatically with the factory adjustment (see below).

For the corresponding settings, first tick the "Expert settings" on the **CANopen Remote Device** tab.



Automatic

You want the control to automatically initialise the I/O system after the device has been replaced.

- Enter all required parameter values on the **Service Data Objects** tab.
- On the **CANopen Remote Device** tab ...
  - do not tick the "Factory Settings" in order that the control does not execute a factory adjustment;
  - tick "Create all SDOs" in order that the control initialises all parameters.
- After the I/O system has been changed: Set the node address and the baud rate at the coding switch and then start the control.

4.3.4.3 Example: Commissioning of a counter (EPM-S601)

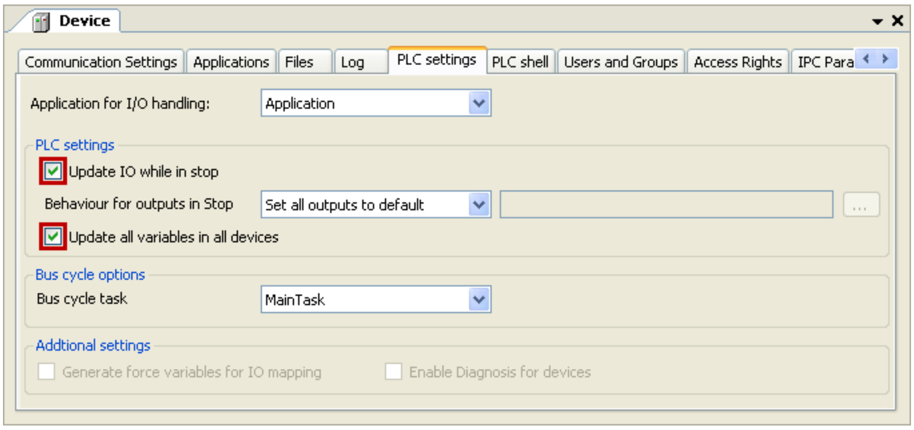
How to commission a counter using the example of an EPM-S601:

1. Create a new »PLC Designer« project.

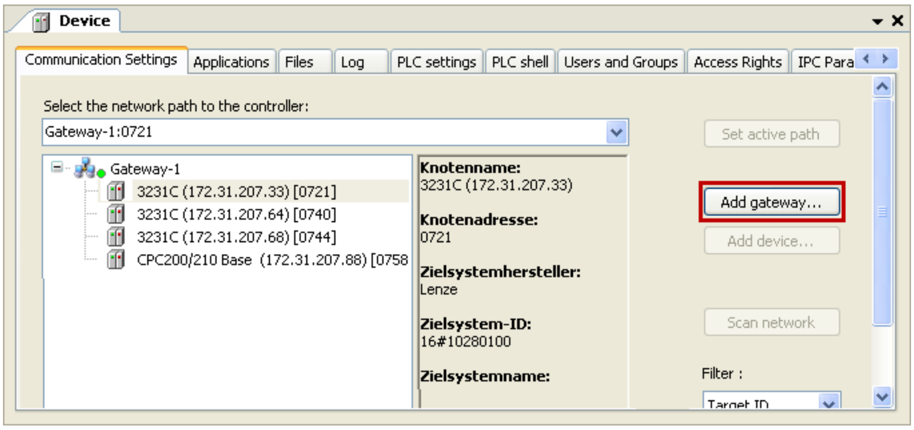
   For the general procedure, see the chapter "Create a new project in the »PLC Designer«".

   Defining the update and connection of gateway/control

2. Double-click the Lenze Controller in the *Device view* in order to open its settings and properties in the *Editor view*.
3. Tick the following on the **PLC settings** tab:



4. Add the matching gateway on the **Communication settings** tab.
   - Either via double-click or **Add gateway...** button



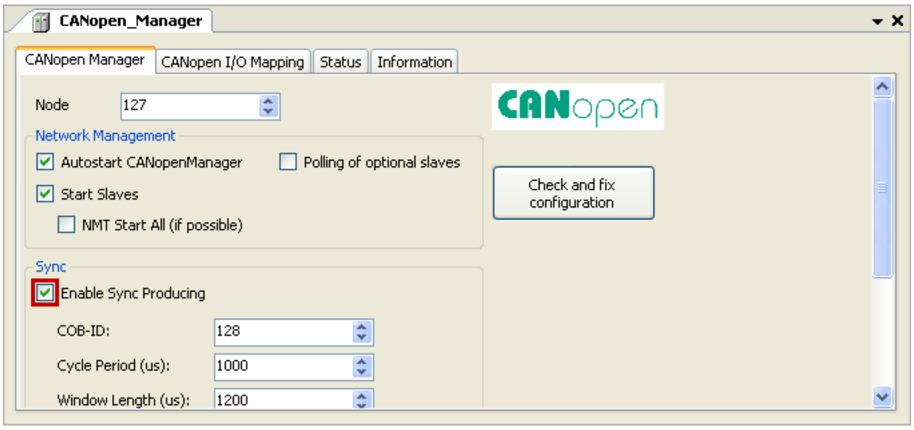
Mapping the station layout in the device view

For the general procedure, see the chapter "Configuring I/O modules on the CANopen bus coupler module".

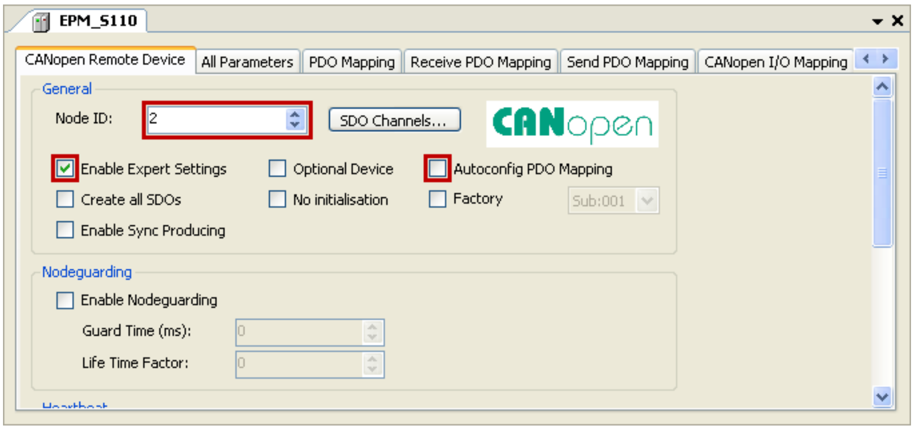
For this example: Add "EPM-S601 Counter 2xDC24V" counter to the CANopen bus coupler module (EPM-S110).

Device settings

5. Double-click the "CANopen_Manager_MC_CAN2" device in the *device view* to open its settings.
6. Activate the **sync generation** option on the **CANopen_Manager** tab:

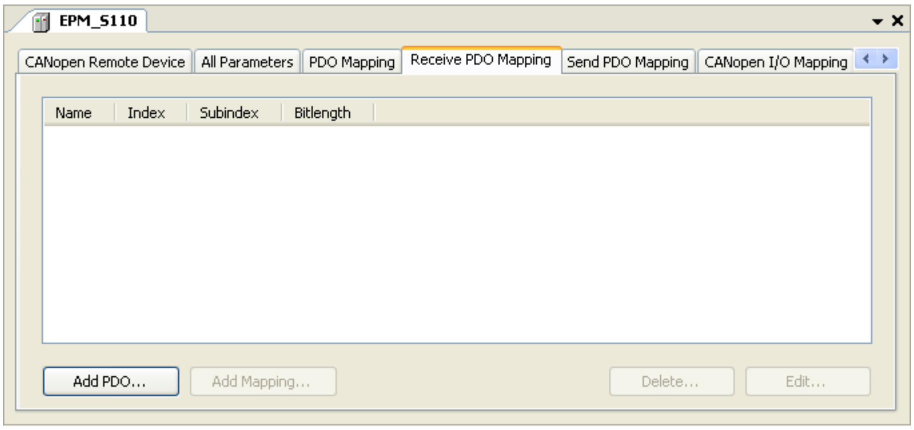


7. Double-click the "EPM_S110" device in the *device view* S110to open its settings.
8. Go to the **CANopen Remote Device** tab:
   - Set node ID "2".
   - Activate the **Expert Settings** option.
   - Deactivate the **Autoconfig. PDO Mapping** option (untick).

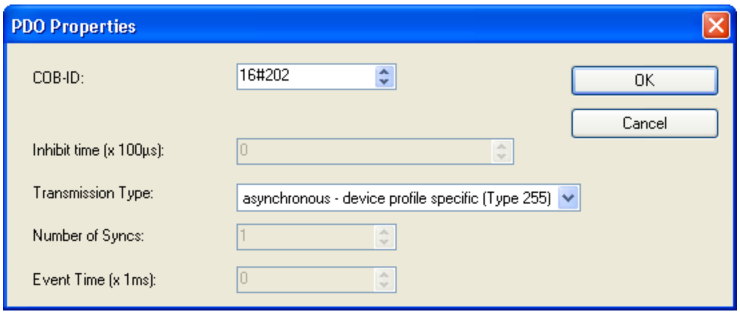


Configuring the Receive PDO Mapping

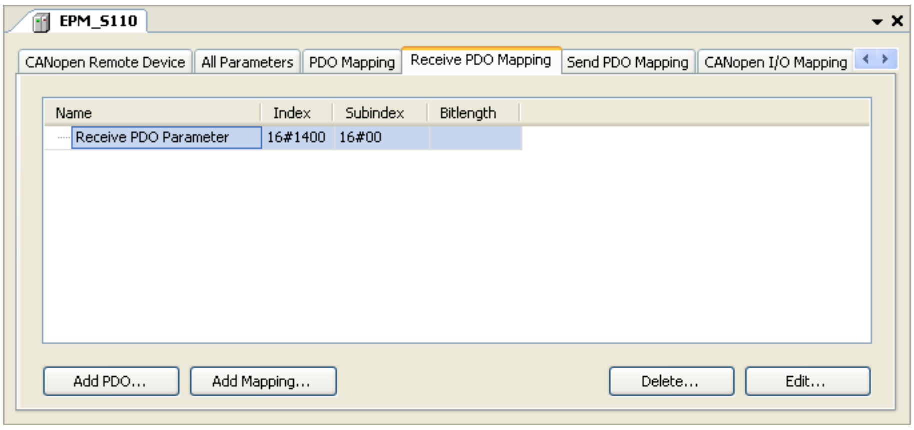
9. Change to the **Receive PDO Mapping** tab.
10. Click **Add PDO....**



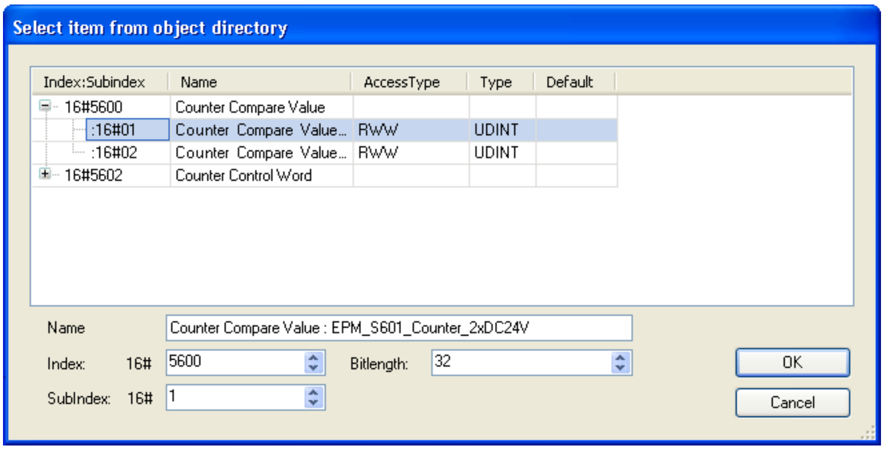
11. Enter the **COB-ID** and the **transmission type** of the PDOs in the *PDO properties* dialog box.
    - Example:



12. Click on the **OK** button to accept the entries and close the dialog box again
13. Select the added PDO and click the **Add Mapping...** button:



14. Go to the *Select element from the object directory* dialog box and select the "Counter Control Word" with the subindex 1 (":16#01") (this serves to start the counter later):



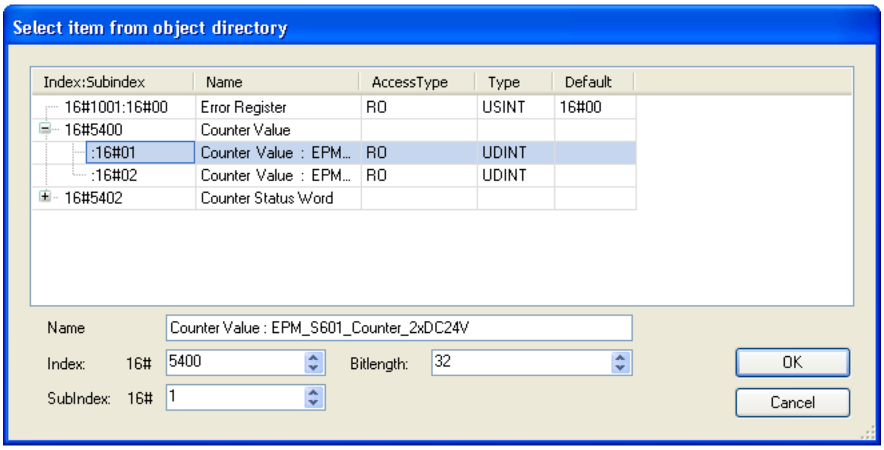
15. Click on the **OK** button to accept the selection and close the dialog box again

Configuring the Send PDO Mapping

16. Execute the "Send PDO Mapping" the same way the "Receive PDO Mapping" has been executed before.

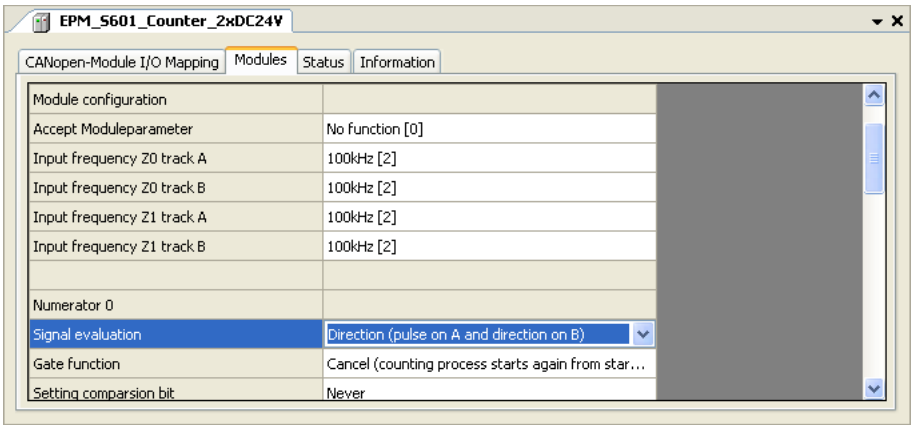
Please observe the following:

- Select "cyclic – synchronous (type 1-240)" as transmission type.
- Go to the *Select element from the object directory* dialog box and select the "Counter Value" with the subindex 1 (":16#01"):



Configuring counter 1

17. Double-click the "EPM_S601_Counter_2xDC24V" device in the *device view* to open its settings.
18. Go to the **Modules** tab and select the"Direction (pulse at A and direction at B" entry in the "Signal evaluation" table row for the counter 0:



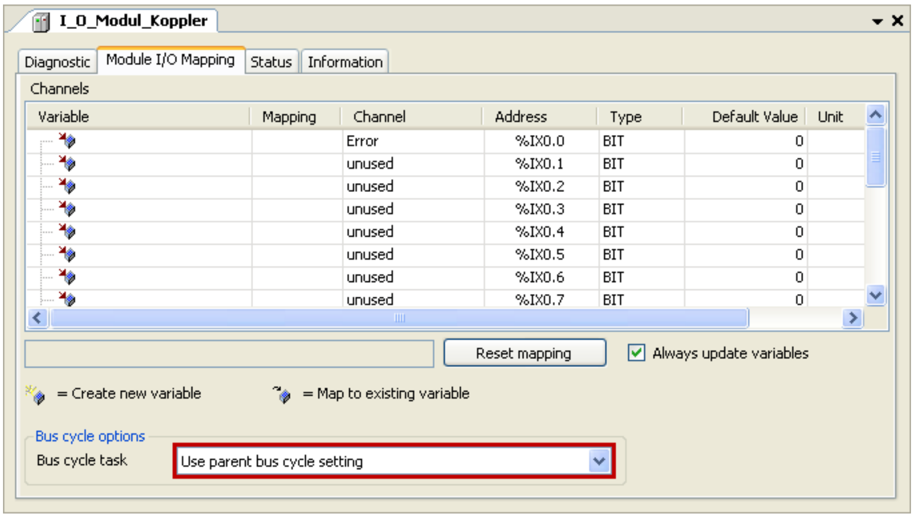
Commissioning completion

19. Establish a connection to the control.
20. Load and start the application.
21. Enter a "4" as the new value in "Counter Control Word" and transfer it via **<Ctrl>**+**<F7>** key combination to start the counting process.
    - As soon as the value has been transmitted and a countable signal is applied at the corresponding input, the counter starts to count upwards.
    - The value is transmitted cyclically via the CAN bus to the control.

4.3.5 Setting the cycle time for access to the I/O modules

In the Lenze Controller a central bus cycle task is selected, which is valid for the processing of all fieldbuses and for the I/O modules connected via the backplane bus.

- This selection is a default setting and can be overwritten in the corresponding bus master nodes (EtherCAT master, CANopen_Manager, I/O module coupler, ...).
- If you require a shorter cycle time for the backplane bus than the one selected in the Lenze Controller, you can select a faster task for I_O_Modul_Koppler on the **Module I/O Mapping** tab:



The following applies to I/O modules on the I_O_Modul_Koppler:

- At first, the »PLC Designer« defines a processing task for the PLC for generating the I/O image. An individual bus cycle task can be defined for each individual module, within which the process image for the module is then created.
- Altogether three different tasks for the I/O modules can be defined. The limitation to three tasks is determined by the system on the basis of the backplane bus specification.
- The backplane bus controller uses two internal bus cycle groups for each task; one for input signals and one for the outputs. A total of six groups is provided.

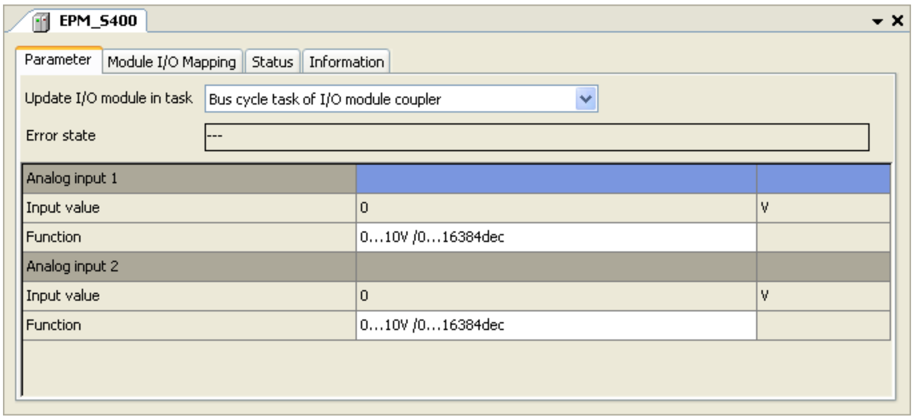
4.4 Parameterising I/O modules in the »PLC Designer«

This chapter uses examples to describe how to parameterise I/O compound modules in the »PLC Designer«.

4.4.1 Parameterising the I/O modules at the backplane bus of the Controller

If you double-click an I/O compound module in the *device view* of the »PLC Designer«, the settings and properties of the I/O compound module are displayed in the *editor view*.

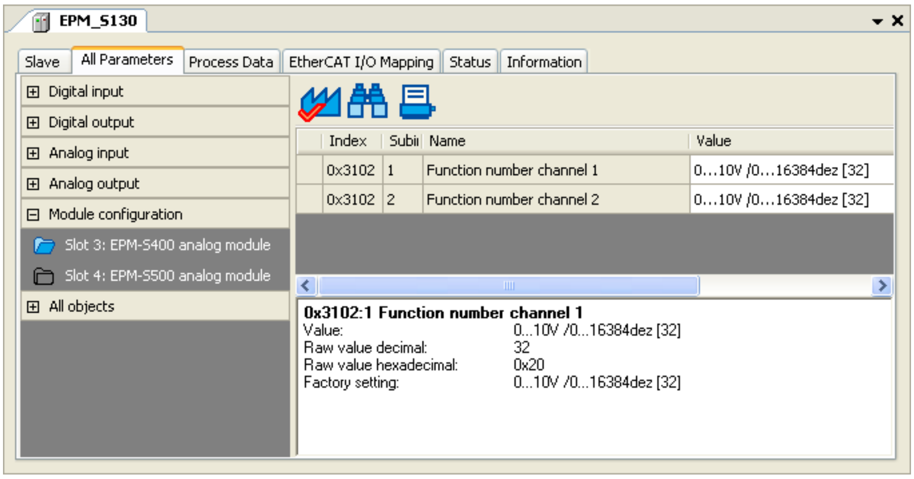
- Use the **Parameter** tab to set the parameters required for the selected I/O compound module (for this example, an "EPM-S400" has been selected):



4.4.2 Parameterising I/O modules on the EtherCAT bus coupler module

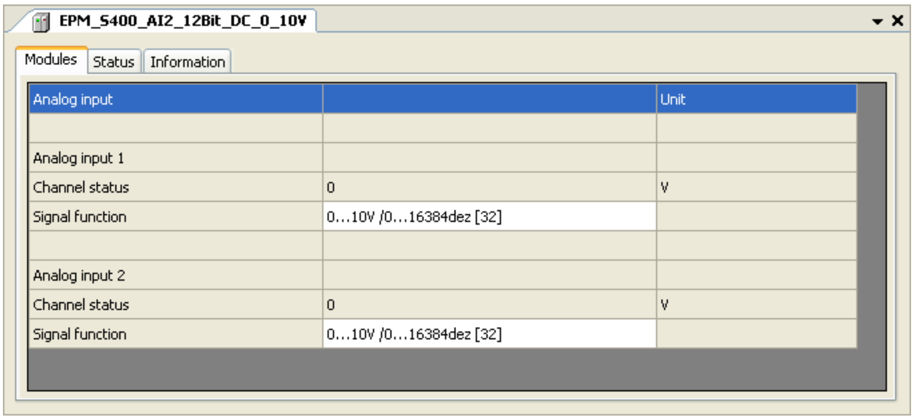
If you double-click the EtherCAT bus coupler module (EPM-S130) in the *device view* of the »PLC Designer«, the settings and properties of the bus coupler modules are displayed in the *editor view*.

- The **All parameters** tab displays the parameters of the bus coupler module and all attached I/O compound modules in a hierarchical structure.
- The following example shows the setting parameters of an "EPM-S400":



As an alternative, you can have the settings and features of a single I/O compound module displayed by double-clicking the corresponding I/O compound module in the *device view*.

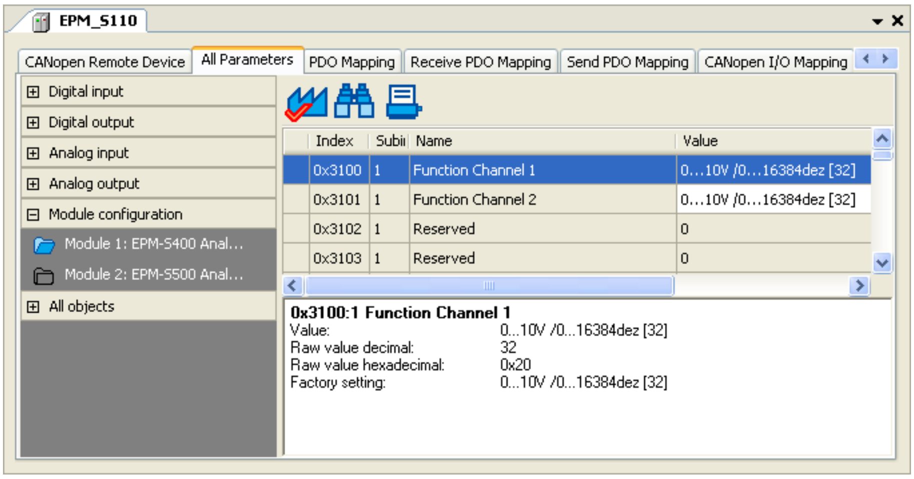
- Use the **Modules** tab to set the parameters required for the selected I/O compound module (for this example, an "EPM-S400" has been selected):



4.4.3 Parameterising I/O modules on the CANopen bus coupler module

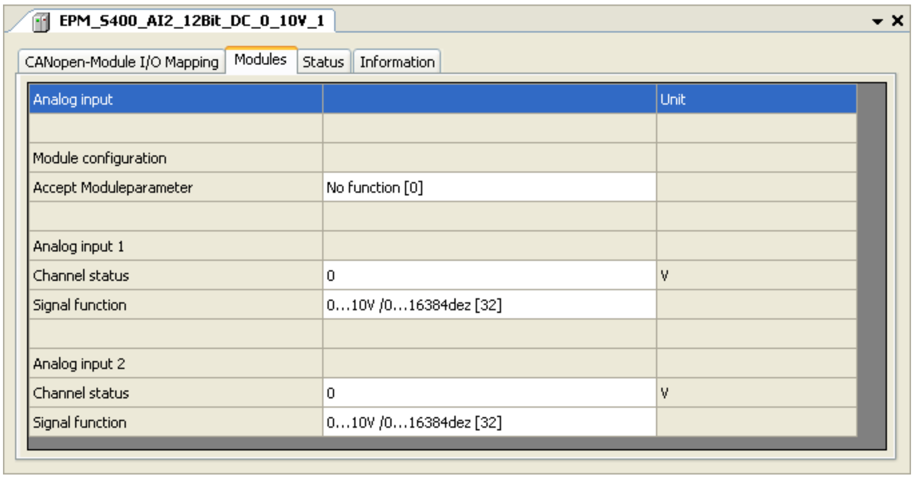
If you double-click the EtherCAT bus coupler module (EPM-S110) in the *device view* of the »PLC Designer«, the settings and properties of the bus coupler modules are displayed in the *editor view*.

- The **Parameters** tab displays the parameters of the bus coupler module and all attached I/O compound modules in a hierarchical structure.
- The following example shows the setting parameters of an "EPM-S400":



As an alternative, you can have the settings and features of a single I/O compound module displayed by double-clicking the corresponding I/O compound module in the *device view*.

- Use the **Modules** tab to set the parameters required for the selected I/O compound module (for this example, an "EPM-S400" has been selected):



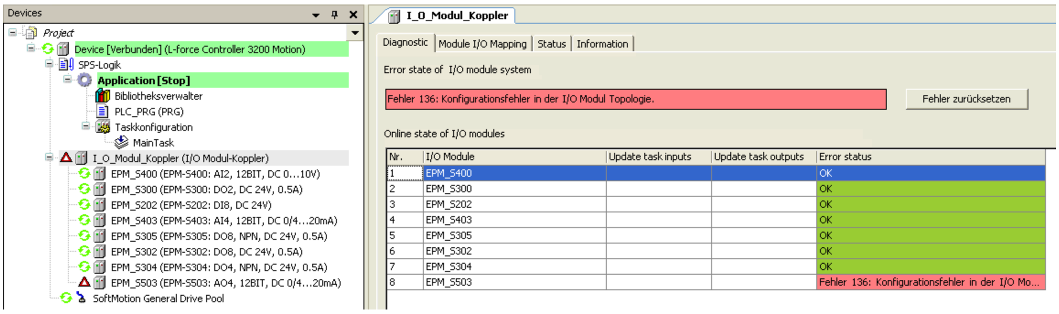
Accept parameter changes

Go to the **parameter list** of the bus coupler module in the **All objects** category and set the '0x31FF' index to the "Accept module parameters" selection.

4.5 Error messages (backplane bus)

The **Diagnostics** tab of the "I_O_Modul_Koppler" in the »PLC Designer« displays the total error status of the I/O system and the current error status of the available I/O compound modules.

- The following example shows an incorrect mapping of the station layout in the *device view* and thus the I/O compound module EPM-S503 at the backplane bus cannot be accessed:



Short overview of the possible error messages

Error number	Error text	Remedy
11	Too many I/O modules for the task cycle time selected	Increase cycle time
12	FIFO response full	Contact Lenze
32	Timeout reset	Contact Lenze
33	Timeout reset error	Contact Lenze
96	Too many I/O modules for the task cycle time selected	Increase cycle time
97	Too many I/O modules for the task cycle time selected	Increase cycle time
98	Too many I/O modules for the task cycle time selected	Increase cycle time
99	Too many I/O modules for the task cycle time selected	Increase cycle time
100	Too many I/O modules for the task cycle time selected	Increase cycle time
101	Timeout for master queue operation	Contact Lenze
102	Bus initialisation error with one I/O module	Contact Lenze
103	Master queue operation error (Not all I/O modules can be reached)	Contact Lenze
104	Read semaphore timeout for group 1	
105	Read semaphore timeout for group 2	
106	Read semaphore timeout for group 3	
107	Write semaphore timeout for group 1	
108	Write semaphore timeout for group 2	
109	Write semaphore timeout for group 3	
110	Internal backplane bus hardware error	Contact Lenze
116	Read semaphore timeout for group 1	
117	Read semaphore timeout for group 2	
118	Read semaphore timeout for group 3	
119	Write semaphore timeout for group 1	
120	Write semaphore timeout for group 2	
121	Write semaphore timeout for group 3	
136	Configuration error in the topology of the I/O modules	Check sequence of the I/O modules in the *device view*.

Error number	Error text	Remedy
137	Too many tasks for processing I/O modules	Reduce the number of tasks. Three tasks are the maximum.
138	Device description does not contain a device type	Contact Lenze
139	Error while writing the initialisation parameters	Contact Lenze
140	No definition for number of channels in device description	Contact Lenze
181	Synchronisation of the I/O system incorrect	Contact Lenze
200	SDO timeout	Contact Lenze
220	SDO communication error	Contact Lenze
221	Physical I/O module topology could not be read.	Contact Lenze
222	I/O system driver could not be opened.	Contact Lenze

5 CANopen communication

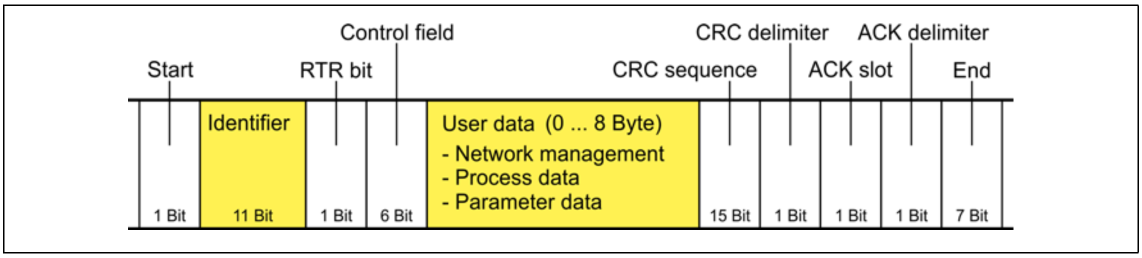
5.1 About CANopen

The I/O system supports the communication module CANopen.

The CANopen protocol is a standardised layer7 protocol for the CAN bus. This layer is based on the CAN Application Layer (CAL) which was developed as a universal protocol.

However, as the practice shows, applications with CAL were too complex for the users. CANopen provides a uniform and simple structure for connecting the CAN devices of the various manufacturers.

5.1.1 Structure of the CAN data telegram



[5-1] Basic structure of the CAN telegram

Only the identifier and the user data are relevant to the user. All other data of the CAN telegram are automatically processed by the system.

Tip!

Please visit the homepage of the CAN user organisation CiA (CAN in automation) for further information:
http://www.can-cia.org

5.1.2 Identifier

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

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

The identifier consists of a 'basic identifier' and the node address of the device to be approached:

Identifier = basic identifier + node address

- This node address is set with the coding switch at the module.
- Network management and sync telegram only require the basic identifier.
- The identifiers can also be set individually
  Assigning individual identifiers (📖 85)

5.1.3 Saving settings

The settings are permanently stored via I-1010 (communication protocol DS301/DS401).

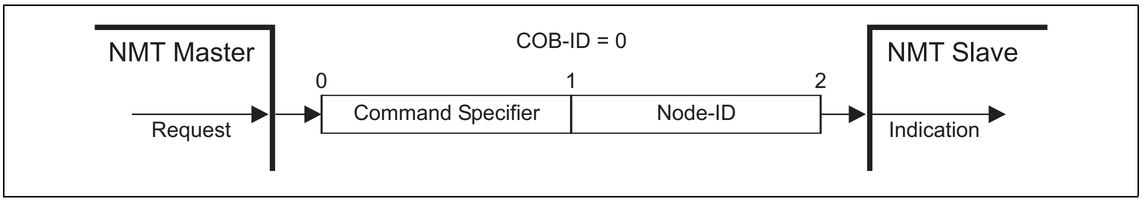
5.2 Network management (NMT)

Via network management, the master is able to make status changes for the entire CAN network.

Communication phases

Status	Explanation
"Initialisation" (Initialisation)	After the I/O system has been switched on, the initialisation is carried out. During this phase, the I/O system is not involved in the data transfer on the bus. Furthermore, it is possible in every NMT state to carry out a part of the initialisation or the entire initialisation again by transmitting various telegrams (see "State transitions"). During this process, all parameters that have already been set receive their standard values again. After the initialisation is complete, the I/O system is automatically in the "Pre-Operational" state.
"Pre-Operational" (before being ready for operation)	The I/O system is able to receive parameter data. The process data are ignored.
"Operational" (ready for operation)	The I/O system is able to receive parameter data and process data.
"Stopped" (stopped)	Only network management telegrams can be received. Parameter data and process data are not received. The module outputs switch to the configured status (see chapter "Monitoring").

Telegram structure



[5-2] Telegram for the change-over in the communication phase

The telegram for the network management contains the identifier and the command from the user data which consists of the command byte and the node address.

To be able to change over between the different communication phases, telegrams with identifier 0 and with 2 bytes of user data are used.

A change-over between communication phases throughout the network can only be carried out by the network master (e.g. inverter).

Note!

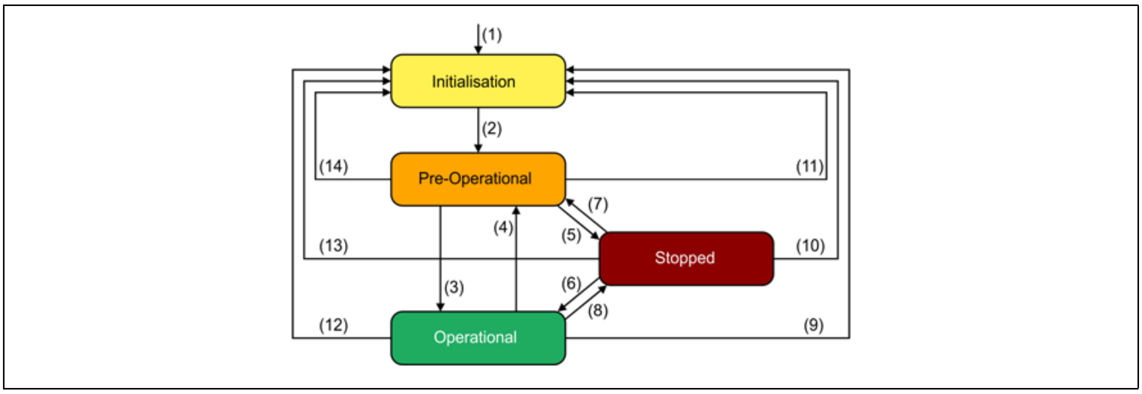
You must switch to the "Operational" state to be able to communicate via process data!

Example:

If all nodes connected to the bus are to be switched from the "Pre-Operational" communication status to the "Operational" communication status via the CAN master, the identifier and the user data must have the following values in the transmission telegram:

- Identifier: 0x00 (broadcast telegram)
- User data: 0x0100

State transitions



[5-3] NMT status transitions in the CAN network

Transition	NMT command	Status after change	Effects on process or parameter data after status change
(1)	-	Initialisation	Initialisation starts automatically when the mains is switched on. • During the initialisation phase, the I/O system is not involved in the data transfer. • After the initialisation is completed, the node sends a boot-up message with an individual identifier and automatically changes to the "pre-operational" status.
(2)	-	Pre-Operational	In this phase, the master decides in which way the I/O systems partakes in the communication.
ℹ	From here, the states are changed over by the master for the entire network. A target address included in the NMT command specifies the receiver/s.		
(3), (6)	0x01 xx Start remote node	Operational	Network management/sync/emergency telegrams as well as process data (PDO) and parameter data (SDO) are active. Optional: When the status is changed, event and time-controlled process data (PDOs) are transmitted once.
(4), (7)	0x80 xx Enter Pre-Operational	Pre-Operational	Network management/sync/emergency telegrams and parameter data (SDO) are active.
(5), (8)	0x02 xx Stop remote node	Stopped	Only network management telegrams can be received.
(9), (10), (11)	0x81 xx Reset node	Initialisation	The parameters saved last via I-1010 are loaded for all indexes. If no value has been saved yet, the Lenze setting is loaded.
(12), (13), (14)	0x82 xx Reset communication		For all communication parameters (index 0x1FFF), the parameters saved last via I-1010 are loaded. If no value has been saved yet, the Lenze setting is loaded.
ℹ	Meaning of the node address in the NMT command xx = 0x00: In case of this assignment, the telegram addresses all devices connected. The state of all devices can be changed at the same time. xx = node ID: If a node address is given, only the state of the device with the corresponding address will be changed.		

5.3 Transmitting process data

Process data are data for control-oriented purposes, e.g. setpoint and actual values.

Process data or the input / output data of the I/O system are transmitted as so-called PDOs (Process Data Objects) with high priority via the fieldbus.

5.3.1 Process data telegram

The process data telegram is structured as follows:

11 Bit	8 bytes of user data							
Identifier	1st byte	2nd byte	3rd byte	4. byte	5th byte	6. byte	7th byte	8th byte

Identifier

Information about the identifier can be found in the chapter "Structure of the CAN data telegram".

Note!

In the Lenze setting, the identifiers are defined according to CANopen (assignment in accordance with DS301). Via I-2101 the identifier calculation according to the system bus (assignment according to the Lenze system bus) can be specified.

User data

The 8 bytes of user data transmit the input signals (user data sent) and the output signals (user data received) of the modules.

Note!

Lenze inverters expect a PDO length of eight bytes even though not all of them may have assigned I/O values. The PDO length can be set via I-2100:

- 0: PDO length of eight bytes (Lenze setting)
- 1: PDO length according to process image

5.3.2 Identifiers of the process data objects (PDO)

The identifiers of process data objects PDO1 ... PDO10 consist of the so-called basic identifiers and the set node address:

Identifier = basic identifier + node address

Basic identifiers of the process data objects

Process data object	Basic identifier	
	dec	hex
PDO1		
RPDO1	512	200
TPDO1	384	180
PDO2		
RPDO2	768	300
TPDO2	640	280
PDO3		
RPDO3	1024	400
TPDO3	896	380
PDO4		
RPDO4	1280	500
TPDO4	1152	480
PDO5		
RPDO5	1920	780
TPDO5	1664	680
PDO6		
RPDO6	576	240
TPDO6	448	1C0
PDO7		
RPDO7	832	340
TPDO7	704	2C0
PDO8		
RPDO8	1088	440
TPDO8	960	3C0
PDO9		
RPDO9	1344	540
TPDO9	1216	4C0
PDO10		
RPDO10	1984	7C0
TPDO10	1728	6C0

5.3.3 Assigning individual identifiers

For larger networks with many nodes, it may be useful to set individual identifiers for process data objects PDO1 ... PDO10, that are independent of the set node address.

Process data objects for input data

Individual identifiers for input data can be set via the indices I-1400, subindex 1 ... I-1409, subindex 1.

Process data objects for output data

Individual identifiers for output data can be set via the indices I-1800, subindex 1 ... I-1809, subindex 1.

Note!

Set the value which makes the required identifier (x = corresponding process data object) in index I-140x, subindex 1 or I-180x, subindex 1.

Make a reset node so that the changes are accepted.

5.3.4 Transmission mode for process data

Transmission mode for process input data

The transmission mode is configured via index I-1400, subindex 2 (PDO1-Rx) ... I-1409, subindex 2 (PDO10-Rx):

- Sync-controlled reception
- n-sync-controlled reception
  - First, a certain number (n) of sync telegrams must be transmitted (I-140x, subindex 2 = 1 ... 240). Then the PDO telegram must be received from the master. Finally, the process input data are accepted.
- Event-controlled reception (Lenze setting)

Transmission mode for process output data

The transmission mode is configured via index I-1800, subindex 2 (PDO1-Tx) ... I-1809, subindex 2 (PDO10-Tx):

- Sync-controlled transmission
- n-sync-controlled transmission
  - First, a certain number (n) of sync telegrams must be transmitted (I-180x, Subindex 2 = 2 ... 240). Then, the PDO telegram is transmitted to the master.
- Event-controlled transmission (Lenze setting)

Note!

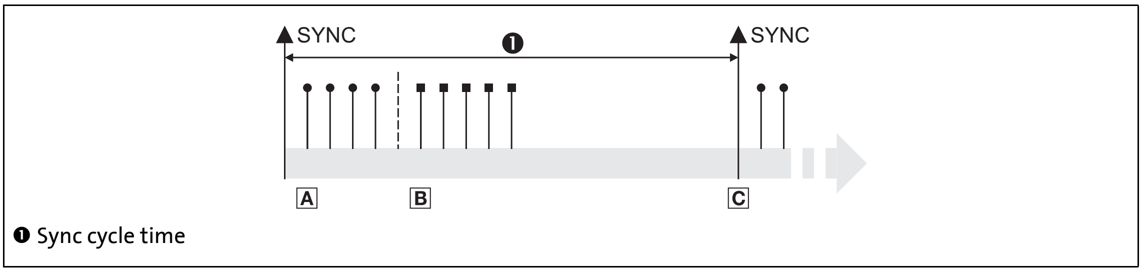
After changing to the CAN state "Operational", the current process image is transmitted from the I/O system.

Sync telegram for cyclic process data

A special telegram, the Sync telegram, is required for synchronisation when cyclic process data are transmitted.

The sync telegram must be generated by **another** node. It initiates the transmission for the cyclic process data of the I/O system and at the same time triggers data acceptance of cyclic process data received in the I/O system.

Basic workflow



[5-4] Synchronisation of the cyclic process data by a sync telegram (without considering the asynchronous data)

1. After receiving a sync telegram, the I/O system transmits the cyclic process output data (PDO1-Tx) if "sync-controlled transmission" is active.
2. Once the transmission is completed, the I/O system receives the cyclic process input data (PDO1-Rx).
3. The data is accepted by the I/O system with the next sync telegram if "sync-controlled reception" is active.
4. All other telegrams (e.g. for parameter or event-controlled process data) are accepted asynchronously by the I/O system after transmission.

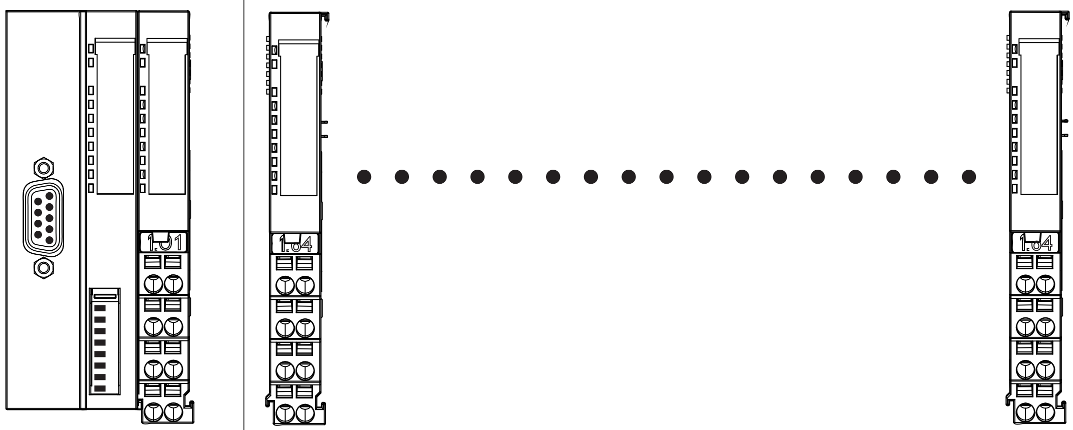
5.3.5 PDO mapping

The PDO mapping differs depending on the motion control to be configured:

Drive-based automation (decentralised, with intelligent controllers)	**Controller-based Automation** (with a central control)
• Direct data exchange between the IO system and the inverter without a master control. Configuration with Lenze »Engineer«	• Direct data exchange between the IO system and the control (IPC). Configuration via Lenze »PLC Designer«
• Static PDO mapping (default setting)	• Free mapping by means of the bus configurator (e.g. »PLC Designer«)
• Up to 10 RPDOs and TPDOs can be used	• Up to 16 RPDOs and TPDOs can be used

The following representation describes the static PDO mapping (Drive-based automation) by means of an example station structure:

Setup



CANopen	2 x DI	8 x DI	4 × DI	2 × DI	8 × DI	8 × DI	8 × DI	8 × DI	2 × DI	8 x DI	2 x AI	4 x AI	8xDO	2 x DO
M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14

- n x DI: I/O compound module with n digital inputs
- n x DO: I/O compound module with n digital outputs
- n x AI: I/O compound module with n analog inputs
- n x AO: I/O compound module with n analog outputs

Process image			Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
PDO1	Permanent for the first DI or DO	RPDO1	M13	M14	–	–	–	–	–	–
		TPDO1	M1	M2	M3 M4	M5	M6	M7	M8	M9
PDO2	Permanent for the first AI or AO	RPDO2	–	–	–	–	–	–	–	–
		TPDO2	M11	M11	M11	M11	M12	M12	M12	M12
PDO3	DI/DO, DI/DO time stamp, DO PWM, AI/AO, counter, SSI, RS232, RS422/RS485	RPDO3	–	–	–	–	–	–	–	–
		TPDO3	M12	M12	M12	M12	–	–	–	–
PDO4	DI/DO, DI/DO time stamp, DO PWM, AI/AO, counter, SSI, RS232, RS422/RS485	RPDO4	–	–	–	–	–	–	–	–
		TPDO4	M10	–	–	–	–	–	–	–
...	...	...								
PDO10	DI/DO, DI/DO time stamp, DO PWM, AI/AO, counter, SSI, RS232, RS422/RS485	RPDO10	–	–	–	–	–	–	–	–
		TPDO10	–	–	–	–	–	–	–	–

The following rules for assigning the PDO apply in accordance with the CANopen communication profile DS401:

- RPDO1 is reserved for the first I/O compound modules with digital inputs.
- TPDO1 is reserved for the first I/O compound modules with digital outputs.
- RPDO2 is reserved for the first I/O compound modules with analog inputs.
- TPDO2 is reserved for the first I/O compound modules with analog outputs.
- As of PDO3: a PDO can only be occupied by electronic modules of one module class. Free bytes are reserved for modules of the same module class.
  Sorting sequence for module classes:
  - Digital inputs and outputs (EPM-S200 to EPM-S206, EPM-S300 to EPM-S309)
  - Analog inputs and outputs (EPM-S400 to EPM-S503)
  - Counter (EPM-S600 to EPM-S603)
  - SSI encoder (EPM-S604)
  - Digital outputs with pulse width modulation (EPM-S620)
  - RS232/RS422/RS485 interfaces (EPM-S640/EPM-S650)
  - Digital outputs with time stamp functionality (EPM-S310)
  - Digital inputs with time stamp functionality (EPM-S207)

Note!

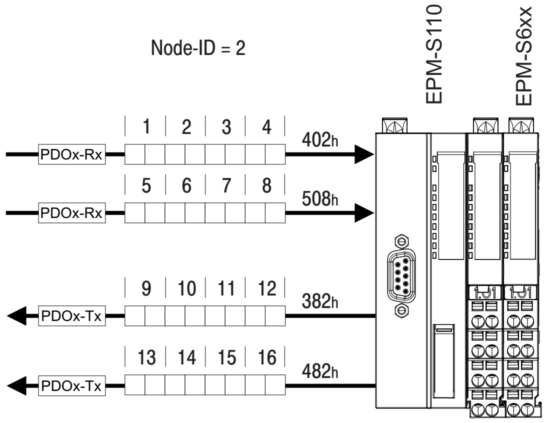
In order to ensure data consistency, the data in the PDO are mapped as follows:

- Digital I/O:
  With digital values, an I/O compound module is always mapped in one byte. If the byte in a PDO has not enough free bits, the I/O compound module is mapped in the next byte.
- Analog I/O:
  With analog values the data lengths of which exceed on byte, the data consistency is extended. Since one channel assigns two bytes, each channel of an I/O compound module is mapped in two successive PDOs.

5.3.6 PDO mapping for I/O compound modules with a serial interface

Mapping for I/O compound modules with a serial interface (EPM-S640, EPM-S650) starts at PDO 3.

Default mapping



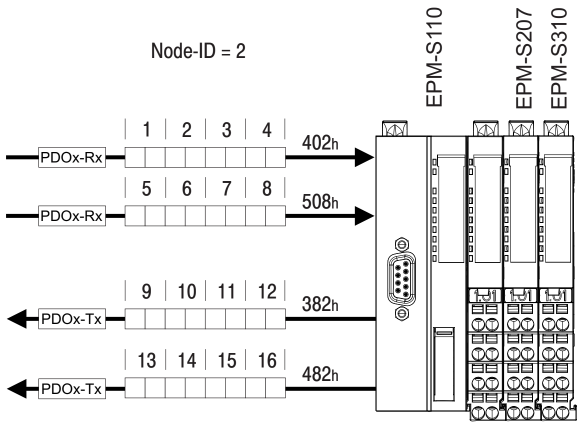
Word	EPM-S640, EPM-S650
1	Rx byte 0 ... 1
2	Rx byte 2 ... 3
3	Rx byte 4 ... 5
4	Rx byte 6 ... 7
5	–
6	–
7	–
8	–
9	Tx byte 0 ... 1
10	Tx byte 2 ... 3
11	Tx byte 4 ... 5
12	Tx byte 6 ... 7
13 ... 16	–

[5-5] Mapping contents

5.3.7 PDO mapping for I/O compound modules with time stamp function

Mapping for I/O compound modules with time stamp function begins at PDO 3.

Default mapping



Word	EPM-S207	EPM-S310
1	–	Ticker value DO1
2	–	Running number Channel status DO1
3	–	Ticker value DO2
4	–	Running number Channel status DO2
5	–	–
6	–	–
7	–	–
8	–	–
9	Ticker value DI1	FIFO memory status
10	Running number Channel status DI1	
11	Ticker value DI2	–
12	Running number Channel status DI2	–
13 ... 16	–	–

[5-6] Mapping contents

Ticker values EPM-S207 and EPM-S310		
Bit	**Name**	**Function**
0 ... 15	Ticker value	After mains connection, a timer (µs ticker) is started, which after 65535 µs starts with 0 again.

Running number, channel status EPM-S207		
Bit	**Name**	**Function**
0 ... 7	Running number (RN)	Counter counting from 0 ... 63 and recommencing at 0. In the first run, the "running number" must start with 1.
8	Channel status DI1	0: FALSE 1: TRUE
9	Channel status DI2	0: FALSE 1: TRUE
10 ... 15	–	Reserved (fix 0)

Running number, channel status EPM-S310		
Bit	**Name**	**Function**
0 ... 7	Running number (RN)	Counter counting from 0 ... 63 and recommencing at 0. In the first run, the "running number" must start with 1.
8 ... 11	–	Reserved (fix 0)
12	Enable DO1	0: Inhibit 1: Enable
13	Enable DO2	
14	Channel status DO1	0: FALSE 1: TRUE
15	Channel status DO2	0: FALSE 1: TRUE

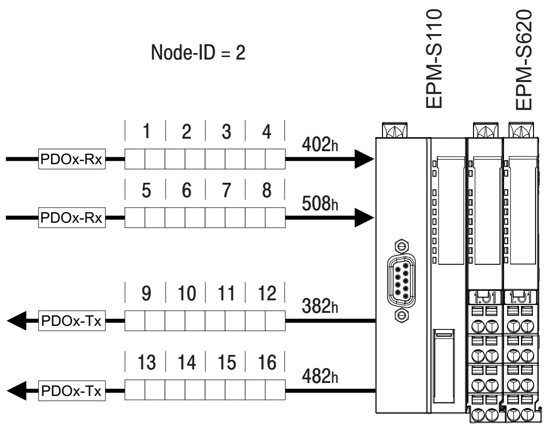
FIFO memory status EPM-S310		
Bit	**Name**	**Function**
0 ... 5	RN-LAST	Running number of the time stamp entry that was recognised as valid last by the module and written into the FIFIO memory.
6		1 (fix); is used for identification in the process image
7		0 (fix); is used for identification in the process image
8 ... 13	RN_NEXT	Running number of the time stamp entry which is processed next in the FIFO memory.
14		1 (fix); is used for identification in the process image
15		1 (fix); is used for identification in the process image
16 ... 23	STS_FIFO	FIFO memory state information • 0x00 or 0x80: Everything is OK; you will receive this message directly after the transfer to the FIFO memory of the module. • 0x01 or 0x81: No follow-up entry available, the running number does not correspond to the expected running number. Check the running number in the output area. • 0x02 or 0x82: No new entries available in the FIFO memory. • 0x03 or 0x83: FIFIO memory full. No new entries possible. A full memory will not accept any more time stamp entries. Perform a status query to establish the FIFO memory's status before transferring more time stamp entries. **Note:** If less than possible time stamp entries are written, bit 6 must be set in the "running number" of the last time stamp entry. This is required in order to not write the follow-up entries as "invalid" since the module ignores all time stamp entries after an entry with a set RN bit 6. If a time stamp entry with a running number and a set bit 6 is in the FIFO memory, STS_FIFO with 0x80 is returned in an OR-ed manner.
24 ... 31	–	Current number of time stamp entries in FIFO memory.

▶ Digital-I/O EPM-S207, EPM-S310 – time stamp function (example) (📖 20)

5.3.8 PDO mapping for I/O compound modules with PWM function

Mapping for I/O compound modules with a PWM function begins at PDO 3.

Default mapping



Word	EPM-S620
1	Pulse duration DO1
2	
3	Pulse duration DO2
4	
5	Control word DO1
6	Control word DO2
7	–
8	–
9	Status word DO1
10	Status word DO2
11 ... 16	–

[5-7] Mapping contents

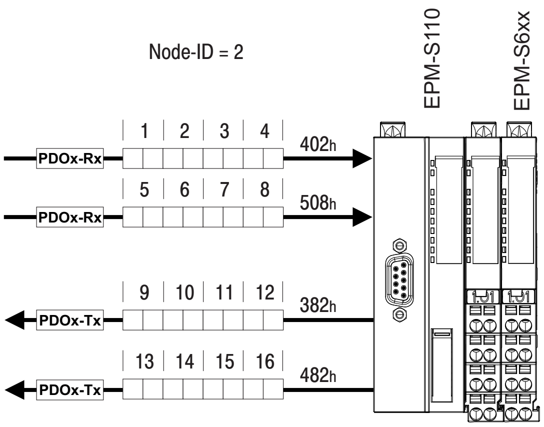
▸ PWM module EPM-S620 – control and status word (📖 32)

5.3.9 PDO mapping for counters and encoder evaluation

The mapping for counters starts at PDO 3, since the first two PDOs are reserved for digital and analog modules.

The following table describes the indexes for the PDO mapping for counters.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-5400		Counter Value		0x00000000	{0x1}	0xFFFFFFFF	Counter value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: Subindex is increased by 2 for each counter
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5401		Latch value		0x00000000	{1}	0xFFFFFFFF	Latch value
	1	DWord 1					EPM-S600: Subindex is increased by 1 for each counter EPM-S601/-S602/-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5402		Status word		0x0000	{1}	0xFFFF	Status word
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: Subindex is increased by 2 for each counter
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5403		Counter ticker value		0x0000	{1}	0xFFFF	Ticker value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5600		Counter Compare Value		0x00000000	{1}	0xFFFFFFFF	Comparison value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601: Subindex is increased by 2 for each counter EPM-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5601		Counter Set Value		0x00000000	{1}	0xFFFFFFFF	Set value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5602		Counter Control Word		0x00000x0000	{1}	0xFFFF	Control word
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: Subindex is increased by 2 for each counter
	2	DWord 2					
	...	...					
	64	DWord 64					

<table>
<tr><th colspan="6">Default mapping</th></tr>
<tr><td colspan="6">
</td></tr>
<tr><th>Word</th><th>EPM-S600</th><th>EPM-S601</th><th>EPM-S602</th><th>EPM-S603</th><th>EPM-S604</th></tr>
<tr><td>1</td><td rowspan="2">Comparison value</td><td rowspan="2">Comparison value C1</td><td rowspan="2">Comparison value</td><td>Control word C1</td><td rowspan="8">-</td></tr>
<tr><td>2</td><td>Control word C2</td></tr>
<tr><td>3</td><td rowspan="2">Set value</td><td rowspan="2">Comparison value C2</td><td rowspan="2">Set value</td><td rowspan="6">-</td></tr>
<tr><td>4</td></tr>
<tr><td>5</td><td>Control word</td><td>Control word C1</td><td>Control word</td></tr>
<tr><td>6</td><td rowspan="3">-</td><td>Control word C2</td><td rowspan="3">-</td></tr>
<tr><td>7</td><td rowspan="2">-</td></tr>
<tr><td>8</td></tr>
<tr><td>9</td><td rowspan="2">Counter value</td><td rowspan="2">Count value C1</td><td rowspan="2">Counter value</td><td rowspan="2">Count value C1</td><td rowspan="2">Encoder value</td></tr>
<tr><td>10</td></tr>
<tr><td>11</td><td rowspan="2">Latch value</td><td rowspan="2">Count value C2</td><td>Status word</td><td rowspan="2">Count value C2</td><td rowspan="2">Ticker value</td></tr>
<tr><td>12</td><td>Ticker value</td></tr>
<tr><td>13</td><td>Status word</td><td>Status word C1</td><td rowspan="4">-</td><td>Status word C1</td><td rowspan="4">-</td></tr>
<tr><td>14</td><td>Ticker value</td><td>Status word C2</td><td>Status word C2</td></tr>
<tr><td>15</td><td rowspan="2">-</td><td rowspan="2">-</td><td rowspan="2">-</td></tr>
<tr><td>16</td></tr>
</table>

[5-8] Mapping contents

Comparison value: Here you can specify a value which, by comparison with the current counter content, is able to influence the counter output or trigger a process alarm. The response of the output or of the process alarm can be parameterised.

Set value: With an edge change 0-1 of *COUNTERVAL_SET* in the control word, the set value is accepted in the counter.

Count value: Current counter content

Latch value: If there is a positive edge at the latch input, the count value is stored here.

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the count value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

▸ Counter modules EPM-S600 ... EPM-S603 – control and status words (📖 27)

5.3.10 Data transfer between I/O system and inverter

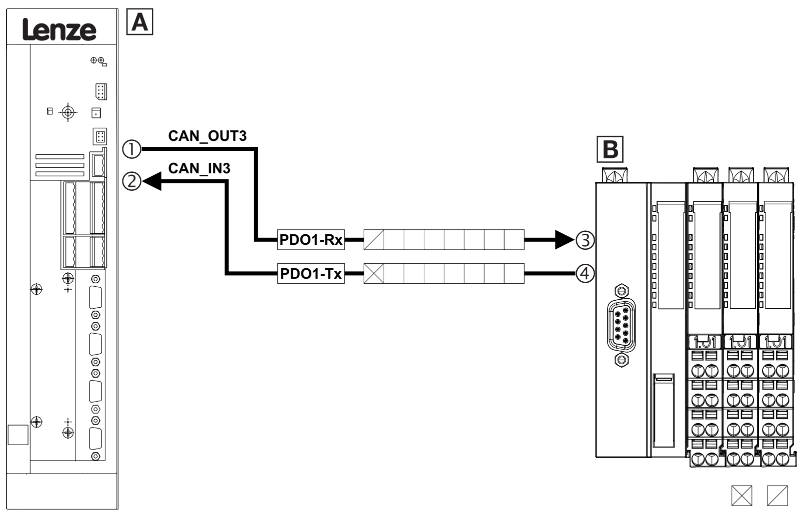
In the Lenze setting of the I/O system 1000, the basic identifiers of the PDOs are set according to CANopen.

For communicating with Lenze inverters, the basic identifiers for the process data object 1 must be adapted.

1. Set RPDO1 to 770 via index I-1400, subindex 1.
2. Set TPDO1 to 769 via index I-1800, subindex 1.

The input data are accepted on sync telegram transmission.

Data transfer between I/O system and inverter



PDO-Rx		The I/O system receives the status information from the inverter
PDO-Tx		The I/O system transmits the status information to the inverter
A	Inverter with node address 1 (C0350 = 1)	
	①	768 (basic identifier) + 1 (node address) = 769 (identifier)
	②	769 (basic identifier) + 1 (node address) = 770 (identifier)
B	CAN bus coupler module with node address 2	
	③	767 (basic identifier) + 2 (node address) = 769 (identifier)
	④	768 (basic identifier) + 2 (node address) = 770 (identifier)

[5-9] Data transfer between I/O system and inverter

5.3.11 Indices for setting the process data transfer

Process data objects for input data

COB-IDs can be changed as follows:

The COB-ID is set via a 32 bit value.

Bit 0 ... 11	Bit 12 ... 29	Bit 30	Bit 31
COB-ID	Reserved	RTR allowed	PDO not valid: 1 PDO valid: 0

Example: Changing the COB-ID from 201 to 202

1. Enter the COB-ID value and bit 31 = 1.
   0xC0000202
2. Bit 31 = 0
   0x40000202
   The PDO is activated with a new identifier. The changes will be accepted in the Pre-Operational or Operational status.

Index	Possible settings			Info
Settings for RxPDO 1 ...				
I-1400/1	385		2047	COB-ID used by RxPDO 1 • Defining the individual identifiers for process data object 1 • Lenze setting: 513 + node ID
I-1400/2				Transmission type • Defining the transmission mode. • Lenze setting: 255
	0 ... 240	Process data update on every sync telegram transmission		The input data are accepted on sync telegram transmission.
	241 ... 254	Reserved		
	255	Process data update on every occurrence of an event		Every received value is accepted.
... RxPDO 10				
I-1409/1	385		2047	COB-ID used by RxPDO 10 • Defining the individual identifiers for process data object 1 • Lenze setting: 1984 + node ID
I-1409/2				Transmission type • Defining the transmission mode. • Lenze setting: 255
	0 ... 240	Process data update on every sync telegram transmission		The input data are accepted on sync telegram transmission.
	241 ... 254	Reserved		
	255	Process data update on every occurrence of an event		Every received value is accepted.

Process data objects for output data

COB-IDs can be changed as follows:

The COB-ID is set via a 32 bit value.

Bit 0 ... 11	Bit 12 ... 29	Bit 30	Bit 31
COB-ID	Reserved	RTR allowed	PDO not valid: 1 PDO valid: 0

Example: A COB-ID value of 182 is to be set

1. Enter the COB-ID value and bit 31 = 1.
   0x80000182
2. Bit 31 = 0
   0x00000182

The PDO is activated with a new identifier. The changes will be accepted in the Pre-Operational or Operational status.

Index	Possible settings			Info
Settings for TxPDO 1 ...				
I-1800/1	385		2047	COB-ID used by TxPDO 1 • Defining the individual identifiers for process data object 1 • Lenze setting: 384 + node ID
I-1800/2				Transmission type • Defining the transmission mode. • Lenze setting: 255
	0	Function deactivated		The output data is accepted when the sync telegram is transmitted.
	1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240		The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted.
	254	Time-controlled process data transfer		Only if a cycle time is set in I-1800/5.
	255	Event-controlled process data transfer		
		Event-controlled process data transfer with cyclic overlay		Only if a cycle time is set in I-1800/5.
I-1800/3	0	ms	65535	Inhibit time • Lenze setting: 0 ms
I-1800/5	0	ms	65535	Event time (cycle time) • Lenze setting: 100 ms
... TxPDO 10				
I-1809/1	385		2047	COB-ID used by TxPDO 10 • Defining the individual identifiers for process data object 1 • Lenze setting: 1728 + node ID

Index	Possible settings			Info
I-1809/2				Transmission type • Defining the transmission mode. • Lenze setting: 255
	0	Function deactivated		The output data is accepted when the sync telegram is transmitted.
	1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240		The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted.
	254	Time-controlled process data transfer		Only if a cycle time is set in I-1809/5.
	255	Event-controlled process data transfer		
		Event-controlled process data transfer with cyclic overlay		Only if a cycle time is set in I-1809/5.
I-1809/3	0	ms	65535	Inhibit time • Lenze setting: 0 ms
I-1809/5	0	ms	65535	Event time (cycle time) • Lenze setting: 100 ms

5.4 Transmitting parameter data

Parameter data are the indices.

The parameters are mostly set once during commissioning.

Parameter data are transmitted as SDOs (*Service Data Objects*) via the system bus and confirmed by the receiver, i.e., the transmitter receives a feedback whether the transmission was successful.

5.4.1 Telegram structure

Structure of the telegram for parameter data:

11 Bit	8 bytes of user data							
Identifier	Command code	Index		Subindex	Data 1	Data 2	Data 3	Data 4
		Low byte	High byte					

- The subchapters below explain the individual telegram components in detail.
- Moreover, you will find examples of how to write and read a parameter
  ▸ Writing a parameter (example) (🕮 103)
  ▸ Reading a parameter (example) (🕮 105)

Identifier

One parameter channel is available for parameter data transmission, which is addressed via the identifier.

Identifier =			Basic identifier		+ node address of the device
			dec	hex	
SDO	Parameter channel 1				
		Output (transmit)	1408	580	+ value set with coding switch
		Input (receive)	1536	600	

5.4.1.1 Command code

The instruction code contains the command to be executed and information about the parameter data length. It is structured as follows:

	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Command	**Command Specifier (cs)**				**Length**		**e**	**s**
Write request	0	0	1	0	00 = 4 bytes 01 = 3 bytes 10 = 2 bytes 11 = 1 byte		1	1
Write Response 0	0	1	1	0			00 = 4 bytes 01 = 3 bytes 10 = 2 bytes 11 = 1 byte	0
Read Request 0	0	1	0	0				0
Read Response 1	0	1	0	0				1
Error response	1	0	0	0	0	0	0	0

Instruction code for parameters with 4 bytes of data length:

Command	4 byte data		Info
	hex	dec	
Write request	23	35	Transmitting parameters to a node
Write response	60	96	Node response to the Write Request (acknowledgement)
Read request	40	64	Request to read a parameter from a node
Read response	43	67	Response to the read request with the actual value
Error response	80	128	Node reports a communication error

"Error Response" command

If an error occurs, the addressed node generates an "Error Response".

In data 4, this telegram always contains the value "6", in data 3 it contains an error code:

Command code Error Response	Data 3	Data 4	Error message
0x80	3	6	Access denied
	5		Wrong subindex
	6		Wrong index

5.4.1.2 Addressing the parameter (index / subindex)

The index of the telegram is used to address the index to be read or written:

- The index value must be entered in flush-left Intel format and divided into Low byte and High byte (see example).
- For subindices, the number of the associated subindex must be entered into the telegram's subindex.
- For indices without subindex, the subindex always has the value "0'.

Example

The subindex 1 of index I-2400 (monitoring time for PDO1) is to be addressed:

11 Bit	8 bytes of user data							
Identifier	Command code	Index		Subindex	Data 1	Data 2	Data 3	Data 4
		Low byte	High byte					
		0x00	0x24	1				

5.4.1.3 Data of the parameter (data 1 ... 4)

Up to 4 bytes (data 1 ... 4) are available for parameter data.

Data are entered in left-justified Intel format with data 1 as LSB and data 4 as MSB (see example).

Example

The value "1 s" is to be transmitted for the index I-2400 (monitoring time):

- Data 1 ... 4 = 1 × 1000 = 1000 = 00 00 03 E8

11 Bit	8 bytes of user data							
Identifier	Command code	Index		Subindex	Data 1	Data 2	Data 3	Data 4
		Low byte	High byte					
					0xE8 (LSB)	0x03	0x00	0x00 (MSB)

5.4.2 Writing a parameter (example)

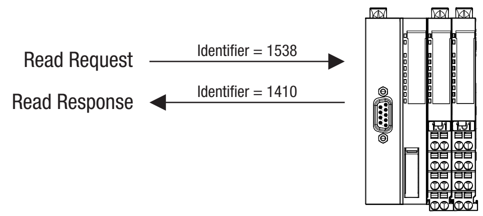
Task

The I/O system 1000 has been assigned node address 2. The function of the first channel (voltage signal 0 ... +10 V, 0 ... 27648) is to be output at the first analog I/O compound module (EPM-S500, 2 analog outputs 0 ... 10 V).

Telegram to the I/O system

	Formula	Info
Identifier	= Basic identifier + node address = 1536 + 2 =1538 = 0x602	• Basic identifier for parameter channel 1 (output) = 1536 • Node address of the I/O system = 2
Command code	= 0x23	• Write "Request command (transmitting parameters to the I/O system)
Index	= I-3100	• First channel of the analog module
Subindex	= 1	• First analog module
Data 1 **Data 2 ... 4**	= 0x0F = 0x00	• 0 ... 10 V, 0 ... 07648dec • Without function

11 Bit	**8 bytes of user data**							
Identifier	**Command code**	**Index**		**Subindex**	**Data 1**	**Data 2**	**Data 3**	**Data 4**
		LOW byte	**HIGH byte**					
0x602	0x23	0x00	0x31	0x0F	0x00 (LSB)	0x00	0x00	0x00 (MSB)



[5-10] Write parameters

Telegram from the I/O system (acknowledgement when being executed faultlessly)

	Formula	Info
Identifier	= Basic identifier + node address = 1408 + 2 = 1410	• Basic identifier for parameter channel 1 (input) = 1408 • Node address of the I/O system = 2
Command code	= 0x60	• Command "Write Response ' (acknowledgement from the I/O system)
Index	= Index of the read request	•
Subindex	= Subindex of the read request	•
Data 1 ... 4	= 0	• Acknowledgement only

11 Bit	**8 bytes of user data**							
Identifier	**Command code**	**Index**		**Subindex**	**Data 1**	**Data 2**	**Data 3**	**Data 4**
		LOW byte	**HIGH byte**					
1410	0x60	0x01	0x30	0	0	0	0	0

5.4.3 Reading a parameter (example)

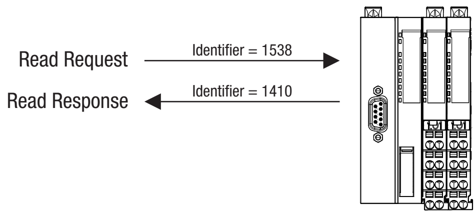
Task

An I/O system 1000 has been assigned node address 21. The function of the first channel is to be read at the first analog I/O compound module (EPM-S500, 2 analog outputs 0 ... 10 V).

Telegram to the I/O system

	Formula	Info
Identifier	= Basic identifier + node address = 1536 + 2 =1538 = 0x602	• Basic identifier for parameter channel 1 (output) = 1536 • Node address of the I/O system = 2
Command code	= 0x40	• "Read Request" command (request for reading a parameter of the I/O system)
Index	= I-3100	• First channel of the analog module
Subindex	= 1	• First analog module
Data 1 ... 4: • Data 1 • Data 2 • Data 3 • Data 4	 = 0x00 = 0x00 = 0x00 = 0x00	• Read request only

11 Bit	8 bytes of user data							
Identifier	Command code	Index		Subindex	Data 1	Data 2	Data 3	Data 4
		Low byte	High byte					
0x602	0x40	0x00	0x31	1	0x0F (LSB)	0x00	0x00	0x00 (MSB)



[5-11] Read parameters

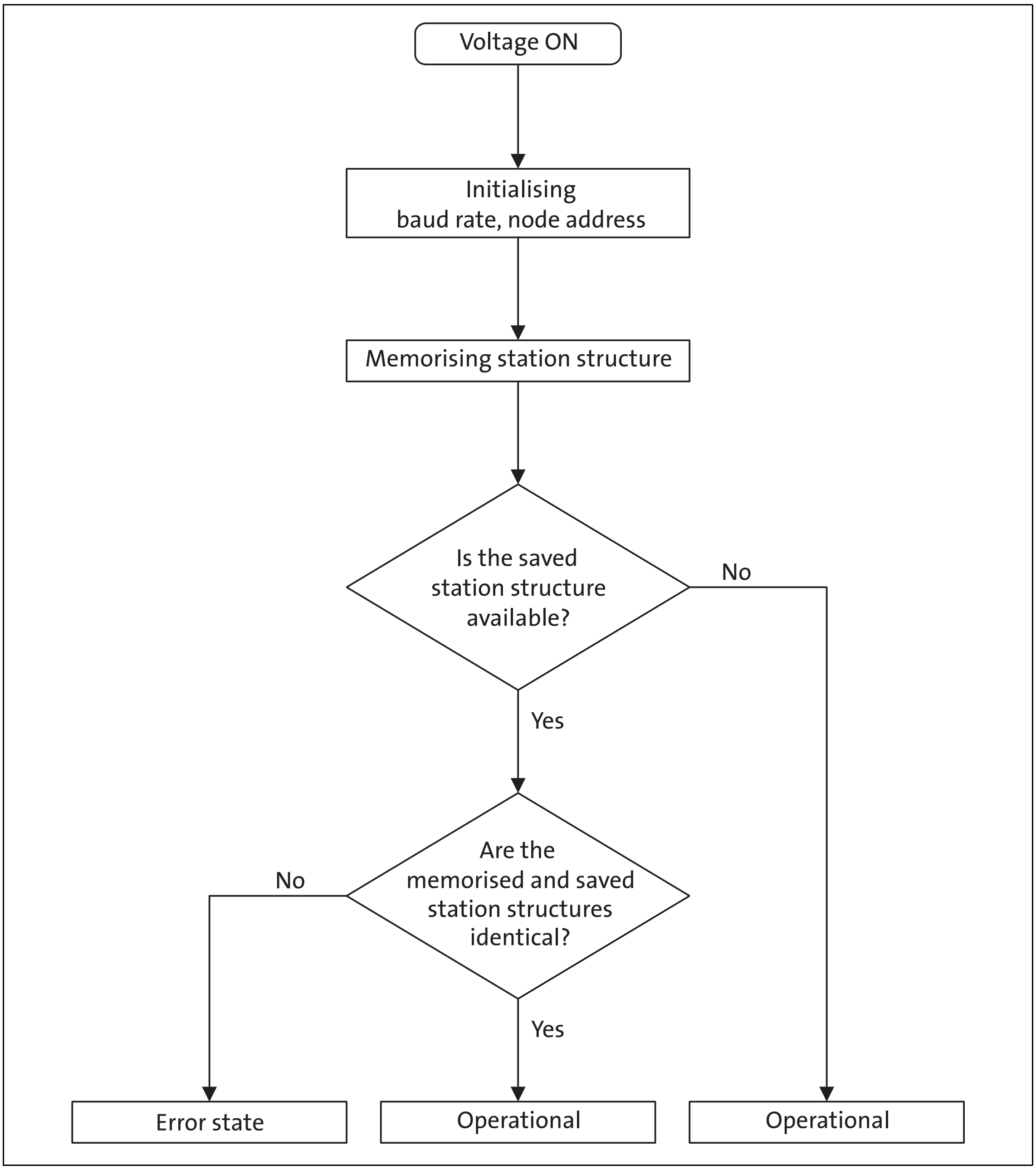
Telegram from the I/O system (value of the requested parameter)

	Formula	Info
Identifier	= Basic identifier + node address = 1408 + 2 = 1410	• Basic identifier for parameter channel 1 (input) = 1408 • Node address of the I/O system = 2
Command code	= 0x43	• "Read Response" command (response to the read request with the current value)
Index	= Index of the read request	
Subindex	= Subindex of the read request	
Data 1 ... 4: • Data 1 • Data 2 • Data 3 • Data 4	 = 0x00 = 0x00 = 0x05 = 0x3B	• Assumption: The first channel of the first analog module outputs a voltage signal 0 ... +10 V with a resolution of 0 ... 07648dec.

11 Bit	8 bytes of user data							
Identifier	Command code	Index		Subindex	Data 1	Data 2	Data 3	Data 4
		LOW byte	HIGH byte					
1410	0x43	0x00	0x31	0	0x00	0x00	0x05	0x3B

5.5 Station behaviour after switch-on

The sequence diagram shows the test routine of the I/O system after every switch-on of the supply voltage.



5.6 Baud rate and node address (node ID) setting

Baud rate

For establishing a communication, all devices must use the same baud rate for the data transfer.

- Use the coding switch to set the baud rate at the CANopen bus coupler module.

Node address

Each node of the network must be assigned to a node address, also called *Node-ID* within a range of 1 ... 127 for clear identification.

- A node address in a network may be used only once.
- Use the coding switch to set the node address of the I/O system at the CANopen bus coupler module.
- The set node address can be read via the index I-100B.
  ▶ Index table (📖 193)

5.7 General function of the parameter setting

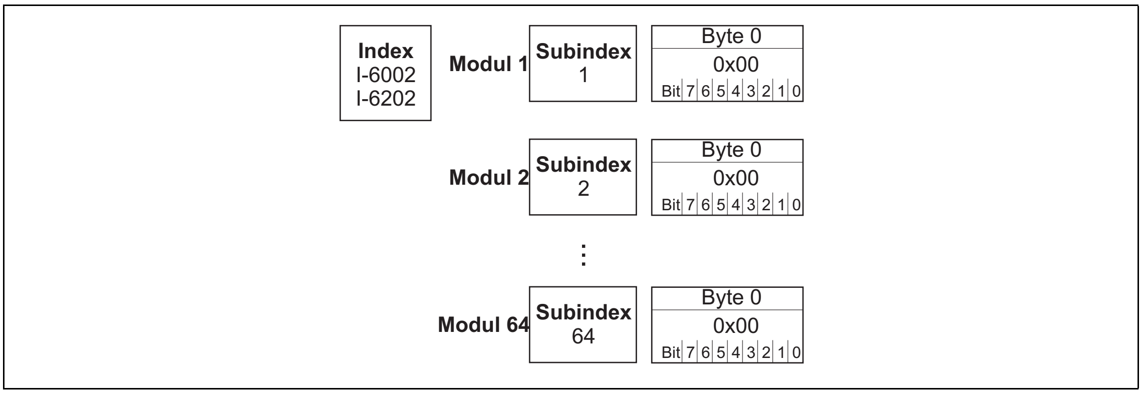
5.7.1 Parameterising digital I/Os

The parameter data of the digital I/O define if the control signals are to be transmitted with original or inverted polarity.

For parameter data, 1 byte is available which is assigned via SDO.

- Digital inputs are parameterised via the index I-6002.
- Digital outputs are parameterised via the index I-6202.

The subindex depends on the slot.



[5-12] Display of the parameter date of digital I/Os

Byte	Assignment				Lenze
0	Polarity of the transmitted signals	Bit 0	0	Signal is transmitted	0x00
			1	Signal is transmitted in inverse form	
		Bit 1...7		Reserve	

Tip!

- Store changed parameters in the EEPROM via index I-1010. The settings are maintained after the supply voltage is switched off.

5.7.2 Parameterising analog I/Os, counter, SSI, time stamp and PWM

The parameter data of each parameterisable I/O compound module (e.g. analog I/O, counter, SSI, digital I/O time stamp or PWM) are values in the index range 0x3100 ... 0x3129.

The parameter data of the 1st parameterisable I/O compound module are in subindex 1 of this range, the parameter data of the 2nd parameterisable I/O compound module in subindex 2, etc.

Index range for	#1 (analog I/O)	#2 (analog I/O)	#3 (counter)
Signal functions	0x3100/1 : 0x3129/1	0x3100/2 : 0x3129/2	0x3100/3 : 0x3129/3

5.8 Parameterising analog I/Os

5.8.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
I-3101/x	Function channel 2		0x20
I-3102/x	Reserved	0	
...	...	...	
I-311D/x	Reserved	0	

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

5.8.2 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
I-3101/x	Function channel 2		0x20
I-3102/x	Function channel 3		0x20
I-3103/x	Function channel 4		0x20
I-3104/x	Reserved	0	
...	...	...	
I-311D/x	Reserved	0	

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

5.8.3 2 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S402

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
I-3101/x	Function channel 2		0x31
I-3102/x	Reserved	0	
...	...	...	
I-311D/x	Reserved	0	

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

5.8.4 4 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S403

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
I-3101/x	Function channel 2		0x31
I-3102/x	Function channel 3		0x31
I-3103/x	Function channel 4		0x31
I-3104/x	Reserved	0	
...	...	...	
I-311D/x	Reserved	0	

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

5.8.5 2 analog inputs -10 ... +10 V (16 bits) - EPM-S406

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Diagnostics	Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
I-3101/x	Reserved	0	
I-3102/x	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 (0 = inhibited; 1 = enabled) Bits 7 ... 2: reserved	0x00
I-3103/x	Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
I-3104/x	Function channel 1	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
I-3105/x	Reserved	0	
I-3106/x I-3107/x	Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is enabled, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
I-3108/x I-3109/x	Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is enabled, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
I-310A/x	Function channel 2	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
I-310B/x	Reserved	0	
I-310C/x I-310D/x	Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is enabled, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
I-310E/x I-310F/x	Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is enabled, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
-10 ... +10 V (0x12)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-5	-13824	CA00		
	-10	-27648	9400		
	-11.76	-32512	8100	Underflow	
-10 ... +10 V (0x22)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-5	-8192	E000		
	-10	-16384	C000		
	-12.5	-20480	B000	Underflow	
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-1.76	-4864	ED00	Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-2	-3277	F333	Underflow	

5.8.6 2 analog inputs 0/4 ... 20 mA (16 bits) - EPM-S408

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Diagnostics	Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
I-3101/x	Reserved	0	
I-3102/x	Limit value monitoring	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 (0 = deactivated; 1 = activated) Bits 7 ... 2: reserved	0x00
I-3103/x	Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
I-3104/x	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
I-3105/x	Reserved	0	
I-3106/x I-3107/x	Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
I-3108/x I-3109/x	Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
I-310A/x	Function channel 2	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
I-310B/x	Reserved	0	
I-310C/x I-310D/x	Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
I-310E/x I-310F/x	Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	**Current (I)** [mA]	**dec**	**hex**	**Range**	**Conversion**
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

5.8.7 2 analog outputs 0 ... 10 V (12 bits) - EPM-S500

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Reserved	0	
I-3101/x	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
I-3102/x	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
I-3103/x	Function channel 2		0x20
I-3104/x	Reserved		
...	...	...	
I-311D/x	Reserved	0	

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

5.8.8 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Reserved	0	
I-3101/x	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
I-3102/x	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
I-3103/x	Function channel 2		0x20
I-3104/x	Function channel 3		0x20
I-3105/x	Function channel 4		0x20
I-3106/x	Reserved	0	
...	...	...	
I-311D/x	Reserved	0	

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

5.8.9 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Reserved	0	
I-3101/x	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
I-3102/x	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
I-3103/x	Function channel 2		0x31
I-3104/x	Reserved		
...	...	...	
I-311D/x	Reserved	0	

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

5.8.10 4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Reserved	0	
I-3101/x	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
I-3102/x	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
I-3103/x	Function channel 2		0x31
I-3104/x	Function channel 3		0x31
I-3105/x	Function channel 4		0x31
I-3106/x	Reserved	0	
...	...	...	
I-311D/x	Reserved	0	

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

5.8.11 Error behaviour

Index		Name	Possible settings		Info
			Lenze	Selection	
I-6443		Error mode analog output		0 {1} 255 0: All analog outputs retain the last value output 255: Response from I-6444	Configures analog output monitoring
	1	Channel 1	0		
	2	Channel 2	0		
	...	...	...		
	36	Channel 36	0		
I-6444		Error value analog output		32768 {1} 32767	Configures the individual analog output responses The analog outputs provide the set value
	1	Channel 1	0		
	2	Channel 2	0		
	...	...	...		
	36	Channel 36	0		

5.9 Parameterising temperature measurement

5.9.1 4(2) analog input for resistance measurement - EPM-S404

Note!

Use parameter setting to deactivate unused inputs.

If thermal detectors are connected in a 3 or 4 conductor setup, channels 3 and/or 4 must be deactivated.

▸ 2-, 3-, 4-wire conductor measurement (📖 26)

The module does not provide any auxiliary supply for sensors.

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
I-3101/x	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
I-3102/x	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
I-3103/x	Reserved		
I-3104/x	Temperature system	Bit 0, 1: 0b00 = °C; 0b01= °F; 0b10 = K Bit 2 ... 7: Reserved	0x00
I-3105/x	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02

Index/subindex	Name	Description/value	Lenze
Channel 1			
I-3106/x	Function channel 1	Thermal detector: 80 (0x50): PT100 2-wire conductor -200°C ... +850°C / -2000 ... +8500 81 (0x51): PT1000 2-wire conductor -200°C ... +850°C / -2000 ... +8500 82 (0x52): Ni100 2-wire conductor -60°C ... +250°C / -600 ... +2500 83 (0x53): Ni1000 2-wire conductor -60°C ... +250°C / -600 ... +2500 88 (0x58): PT100 3-wire conductor -200°C ... +850°C / -2000 ... +8500 89 (0x59): PT1000 3-wire conductor -200°C ... +850°C / -2000 ... +8500 90 (0x5A): Ni100 3-wire conductor -60°C ... +250°C / -600 ... +2500 91 (0x5B): Ni1000 3-wire conductor -60°C ... +250°C / -600 ... +2500 96 (0x60): PT100 4-wire conductor -200°C ... +850°C / -2000 ... +8500 97 (0x61): PT1000 4-wire conductor -200°C ... +850°C / -2000 ... +8500 98 (0x62): Ni100 4-wire conductor -60°C ... +250°C / -600 ... +2500 99 (0x63): Ni1000 4-wire conductor -60°C ... +250°C / -600 ... +2500 Resistor: 112 (0x70): R 60-Ω 2-wire conductor 0.00 ... +60.00 / 0 ... +32767 113 (0x71): R 600-Ω 2-wire conductor 0.00 ... +600.00 / 0 ... +32767 114 (0x72): R 3000-Ω 2-wire conductor 0.00 ... +3000.00 / 0 ... +32767 115 (0x73): R 6000-Ω 2-wire conductor 0.00 ... +6000.00 / 0 ... +32767 128 (0x80): R 60-Ω 4-wire conductor 0.00 ... +60.00 / 0 ... +32767 129 (0x81): R 600-Ω 4-wire conductor 0.00 ... +600.00 / 0 ... +32767 130 (0x82): R 3000-Ω 4-wire conductor 0.00 ... +3000.00 / 0 ... +32767 144 (0x90): R 60-Ω 2-wire conductor 0.00 ... +60.00 / 0 ... +6000 145 (0x91): R 600-Ω 2-wire conductor 0.00 ... +600.00 / 0 ... +6000 146 (0x92): R 3000-Ω 2-wire conductor 0.00 ... +3000.00 / 0 ... +30000 160 (0xA0): R 60-Ω 2-wire conductor 0.00 ... +60.00 / 0 ... +6000 161 (0xA1): R 600-Ω 2-wire conductor 0.00 ... +600.00 / 0 ... +6000 162 (0xA2): R 3000-Ω 2-wire conductor 0.00 ... +3000.00 / 0 ... +30000 255 (0xFF): Channel deactivated	0x50

Index/subindex	Name	Description/value	Lenze
I-3107/x	Conversion time Channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At 50 Hz: 44.1 ms/channel 16 bits; at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x00
I-3108/x	Upper limit value (HIGH byte) channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. If your measured value is outside a limit value and you have activated the limit value monitoring, a process alarm is triggered.	0x7F
I-3109/x	Upper limit value channel 1 (LOW byte)		0xFF
I-310A/x	Lower limit value (HIGH byte) channel 1		0x80
I-310B/x	Lower limit value channel 1 (LOW byte)		0x00
Channel 2			
I-310C/x	Function channel 2	See channel 1	0x50
I-310D/x	Conversion time channel 2	See channel 1	0x00
I-310E/x	Upper limit value channel 2 (HIGH byte)	See channel 1	0x7F
I-310F/x	Upper limit value (LOW byte) channel 2		0xFF
I-3110/x	Lower limit value channel 2 (HIGH byte)		0x80
I-3111/x	Lower limit value (LOW byte) channel 2		0x00
Channel 3 (for two-wire conductor connections only)			
I-3112/x	Function channel 3	See channel 1	0x50
I-3113/x	Conversion time channel 3	See channel 1	0x00

Index/subindex	Name	Description/value	Lenze
I-3114/x	Upper limit value channel 3 (HIGH byte)	See channel 1	0x7F
I-3115/x	Upper limit value channel 3 (LOW byte)		0xFF
I-3116/x	Lower limit value channel 3 (HIGH byte)		0x80
I-3117/x	Lower limit value channel 3 (LOW byte)		0x00
Channel 4 (for two-wire conductor connections only)			
I-3118/x	Function channel 4	See channel 1	0x50
I-3119/x	Conversion time channel 4	See channel 1	0x00
I-311A/x	Upper limit value channel 4 (HIGH byte)	See channel 1	0x7F
I-311B/x	Upper limit value channel 4 (LOW byte)		0xFF
I-311C/x	Lower limit value channel 4 (HIGH byte)		0x80
I-311D/x	Lower limit value channel 4 (LOW byte)		0x00

Measuring range

Measuring range (Fct. no.)	Measured value	Signal range	Range
2-wire: PT100 (0x50)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: PT1000 (0x51)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: NI100 (0x52)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: NI1000 (0x53)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: PT100 (0x58)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: PT1000 (0x59)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: NI100 (0x5A)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: NI1000 (0x5B)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: PT100 (0x60)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
4-wire: PT1000 (0x61)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
4-wire: NI100 (0x62)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: NI1000 (0x63)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: 0 ... 60 Ω (0x70)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x71)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
2-wire: 0 ... 3000 Ω (0x72)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0x78)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x79)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x7A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0x80)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0x81)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0x82)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0x90)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x91)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0x92)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0x98)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x99)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x9A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xA0)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
4-wire: 0 ... 600 Ω (0xA1)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0xA2)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0xD0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0xD1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0xD2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0xD8)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0xD9)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0xDA)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xE0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0xE1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0xE2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow

5.9.2 2 analog inputs for thermocouple measurement - EPM-S405

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
I-3101/x	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
I-3102/x	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
I-3103/x	Reserved	0	
I-3104/x	Temperature system	Bit 0, 1: 0b00 = °C; 0b10 = °F; 0b11 = K Bit 2 ... 7: Reserved	0x00
I-3105/x	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02
Channel 1			
I-3106/x	Function channel 1	External temperature compensation: 176 (0x60): type J, -210.0 ... +1200.0 °C / -2100 ... +12000 177 (0x61): type K, -270.0 ... +1372.0 °C / -2700 ... +13720 178 (0x62): type N -270.0 ... +1300.0 °C / -2700 ... +13000 179 (0x63): type R, -50.0 ... +1769.0 °C / -500 ... +17690 180 (0x64): type S, -50.0 ... +1769.0 °C / -500 ... +17690 181 (0x65): type T, -270.0 ... +400.0 °C / -2700 ... +4000 182 (0x66): type B, 0.0 ... +1820.0 °C / 0 ... +18200 183 (0x67): type C, 0.0 ... +2315.0 °C / 0 ... +23150 184 (0x68): type E, -270.0 ... +1000.0 °C / -2700 ... +10000 185 (0x69): type L, -200.0 ... +900.0 °C / -2000 ... +9000 Internal temperature compensation: 192 (0xC0): type J, -210.0 ... +1200.0 °C / -2100 ... +12000 193 (0xC1): type K, -270.0 ... +1372.0 °C / -2700 ... +13720 194 (0xC2): type N -270.0 ... +1300.0 °C / -2700 ... +13000 195 (0xC3): type R, -50.0 ... +1769.0 °C / -500 ... +17690 196 (0xC4): type S, -50.0 ... +1769.0 °C / -500 ... +17690 197 (0xC5): type T, -270.0 ... +400.0 °C / -2700 ... +4000 198 (0xC6): type B, 0.0 ... +1820.0 °C / 0 ... +18200 199 (0xC7): type C, 0.0 ... +2315.0 °C / 0 ... +23150 200 (0xC8): type E, -270.0 ... +1000.0 °C / -2700 ... +10000 201 (0xC9): type L, -200.0 ... +900.0 °C / -2000 ... +9000 255 (0xFF): Channel deactivated	0xC1

Index/subindex	Name	Description/value	Lenze
I-3107/x	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At Hz: 44.1 ms/channel 16 bits at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x02
I-3108/x	Upper limit value (HIGH byte) channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. If your measured value is outside a limit value and you have activated the limit value monitoring, a process alarm is triggered.	0x7F
I-3109/x	Upper limit value channel 1 (LOW byte)		0xFF
I-310A/x	Lower limit value (HIGH byte) channel 1		0x80
I-310B/x	Lower limit value channel 1 (LOW byte)		0x00
Channel 2			
I-310C/x	Function channel 2	See channel 1	0xC1
I-310D/x	Conversion time channel 2	See channel 1	0x02
I-310E/x	Upper limit value channel 2 (HIGH byte)		0x7F
I-310F/x	Upper limit value (LOW byte) channel 2		0xFF
I-3110/x	Lower limit value channel 2 (HIGH byte)		0x80
I-3111/x	Lower limit value (LOW byte) channel 2		0x00
I-3112/x	Reserved	0	
...	...	...	
I-311D/x	Reserved	0	

Measuring range

Measuring range (Fct. no.)	Measured value [°C]	[°F]	[K]	Range
Type J: -210 ... +1200 °C -346 ... 2192 °F 63.2 ... 1473.2 K (0xB0: ext. comp. 0 °C) (0xC0: int. comp. 0 °C)	+14500	26420	17232	Overflow
	-2100 ... +12000	-3460 ... +21920	632 ... 14732	Nominal range
	-	-	-	Underflow
Type K: -210 ... +1372 °C -454 ... 2501.6 °F 0 ... 1645.2 K (0xB1: ext. comp. 0 °C) (0xC1: int. comp. 0 °C)	+16220	29516	18952	Overflow
	-2700 ... +13720	-4540 ... 25016	0 ... 16452	Nominal range
	-	-	-	Underflow
Type N: -270 ... +1300 °C -454 ... 2372 °F 0 ... 1573.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+15500	28220	18232	Overflow
	-2700 ... +13000	-4540 ... 23720	0 ... 15732	Nominal range
	-	-	-	Underflow
Type R: -50 ... +1769 °C -58 ... 3216.2 °F 223.2 ... 2042.2 K (0xB3: ext. comp. 0 °C) (0xC3: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type S: -50 ... +1769 °C -58 ... 3216.2 °F 223.2 ... 2042.2 K (0xB4: ext. comp. 0 °C) (0xC4: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type T: -270 ... +440 °C -454 ... 752 °F 3.2 ... 673.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+5400	10040	8132	Overflow
	-2700 ... +4000	-4540 ... 7520	32 ... 6732	Nominal range
	-	-	-	Underflow
Type B: 0 ... +1820 °C 32 ... 2786.5 °F 273.2 ... 2093.2 K (0xB6: ext. comp. 0 °C) (0xC6: int. comp. 0 °C)	+20700	32766	23432	Overflow
	0 ... +18200	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow

Measuring range (Fct. no.)	Measured value [°C]	[°F]	[K]	Range
Type C: 0 ... +2315 °C 32 ... 2786.5 °F 273.2 ... 2093.2 K (0xB7: ext. comp. 0 °C) (0xC7: int. comp. 0 °C)	+25000	32766	23432	Overflow
	0 ... +23150	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow
Type E: -270 ... +1000 °C -454 ... 1832 °F 0 ... 1273.2 K (0xB8: ext. comp. 0 °C) (0xC8: int. comp. 0 °C)	+12000	21920	14732	Overflow
	-2700 ... +10000	-4540 ... 18320	0 ... 12732	Nominal range
	-	-	-	Underflow
Type L: -200 ... +900 °C -328 ... 1652 °F 73.2 ... 1173.2 K (0xB9: ext. comp. 0 °C) (0xC9: int. comp. 0 °C)	+11500	21020	14232	Overflow
	-2000 ... +9000	-3280 ... 16520	732 ... 11732	Nominal range
	-	-	-	Underflow

5.9.3 Error behaviour

Index		Name	Possible settings				Info
			Lenze	Selection			
I-6443		Error mode analog output		0	{1}	255	Configures analog output monitoring
				I-6443	0 All analog outputs retain the last value output		
					255 Response from I-6444		In I-6444, the response can be configured individually for each analog output
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				
I-6444		Error value analog output		-32768	{1}	32767	Configures the individual analog output responses The analog outputs provide the set value
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				

5.10 Parameterising the counter

5.10.1 Commissioning examples

The indices for signal setting in the following examples refer to the EPM-S600.
▸ Parameter data (📖 140)

In the case of different counters, the function selection can be assigned to a different index.

Example 1: Counting upwards

Step	Setting	Comment
1	Set signal evaluation: Index 0x310A/x, bit 2 ... 0 = 0b100 (direction)	A signal evaluation has to be specified, otherwise the counting process is not started.
2	Accept parameter setting: 0x31FF = 255	No storage into the EEPROM of the bus coupler.
3	Enable of the software gates via control word : Bit 2 (*SW_GATE_SET*) = HIGH	Transfer control word to the counter via PDO.

Example 2: Accept set value

Step	Setting	Comment
1	Set signal evaluation: Index 0x310A/x, bit 2 ... 0 = 0b100 (direction)	A signal evaluation has to be specified, otherwise the counting process is not started.
2	Accept parameter setting: 0x31FF = 255	No storage into the EEPROM of the bus coupler.
3	Transmit set value to the counter via PDO: e.g. 2147483583→Tx-PDO= 7F|FF|FF|BF	
4	Activating the set value via control word: Bit 5 (*COUNTERval_SET*) = HIGH	Transfer control word to the counter via PDO.

Example 3: Set comparison bit

If the count value is 255 (= comparison value), the digital output is set.

Step	Setting	Comment
1	Set signal evaluation: Index 0x310A/x, bit 2 ... 0 = 0b100 (direction)	A signal evaluation has to be specified, otherwise the counting process is not started.
2	Set comparator: Index 0x3109/x, bit 2 ... 0 = 0b010 (count value ≤ comparison value)	
3	Accept parameter setting: 0x31FF = 255	No storage into the EEPROM of the bus coupler.
4	Transfer the comparison value to the counter via PDO: e.g. 255→Tx-PDO= FF|00|00|00|00	
5	Enable digital output via control word : Bit 1 (*CTRL_DO_SET*) = HIGH	Transfer control word to the counter via PDO.

5.10.2 Rotary transducer signal evaluation

Depending on the edge of the channel that is evaluated, the following pulse trains and the connected pulse multiplication can be realised.

Pulse trains	
Channel A 	
Channel B 	
Single evaluation 	A response to the falling edges of channel A takes place. The number of pulses has not increased.
Double evaluation 	A response to the rising and falling edges of channel A takes place. The number of pulses has doubled and is symmetrical.
Four-fold evaluation 	The rising and falling edges of the channels A and B are evaluated. The number of pulses has quadrupled and is symmetrical.

5.10.3 1 counter 32 bits, 24 V DC - EPM-S600

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter a differentiation between the internal gate (I-gate), hardware gate (HW gate), and software gate (SW gate) is made. • The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). • The SW gate is controlled via your user program (status word in the output area). • The HW gate is controlled via the digital gate input. The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Latch function	If a positive edge occurs at the latch input, the current count value is stored in the latch register. The latch register is accessed via the input area. After a STOP-RUN transition, latch is always 0.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Further information: ▶ Product description (🕮 15)

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
I-3101/x	Input frequency Track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
I-3102/x	Input frequency Track B		0x02
I-3103/x	Input frequency Latch		0x02
I-3104/x	Input frequency Gate		0x02
I-3105/x	Input frequency Reset		0x00
I-3106/x	Reserved		
I-3107/x	Alarm response	Setting activates process alarm Bit 0: Proc. alarm HW gate open Bit 1: Proc. alarm HW gate closed Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 6: Proc. alarm latch value Bit 7: Reserved	0x80
I-3108/x	Numerator function	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = single counting, main counting direction is forward 0b000010 = single counting, main counting direction is backward 0b000100 = single counting, no main counting direction 0b001000 = periodic counting, main counting direction is forward 0b010000 = periodic counting, main counting direction is backward 0b100000 = periodic counting, no main counting direction Bit 7 ... 6: Reserved	0x40
I-3109/x	Comparator	Bit 2 ... 0: output switches (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 ... 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00

Index/subindex	Name	Description/value	Lenze
I-310A/x	Signal evaluation	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Hardware gate (HW gate) • 0b000 = deactivated (counter starts by setting SW gate) • 0b001 = activated (HIGH level at gate activates the HW gate. Counter starts if HW and SW gate are set.) Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	
I-310B/x ... I-310E/x	Final value	Upper limitation of the counting range Counting method: • 0x310B: byte 3 (high byte) • 0x310C: byte 2 • 0x310D: byte 1 • 0x310E: byte 0 (low byte)	0x00
I-310F/x ... I-3112/x	Start value	Lower limitation of the counting range Counting method: • 0x310B: byte 3 (high byte) • 0x3110: byte 2 • 0x3111: byte 1 • 0x3112: byte 0 (low byte)	0x00
I-3113/x	Hysteresis	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
I-3114/x	Pulse	The pulse duration indicates for how long the output is to be set if the parameterised comparison criterion is reached or exceeded. The pulse duration can be specified in steps of 2.048 ms between 0 and 522.24 ms. If the pulse duration is = 0, the output is set until the comparison condition is no longer met.	0x00
I-3115/x ... I-3129/x	Reserved		

5.10.4 2 counters 32 bits, 24 V DC - EPM-S601

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Further information: ▸ Product description (📖 15)

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
I-3101/x	Input frequency Counter 1, track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
I-3102/x	Input frequency Counter 1, track B		0x02
I-3103/x	Input frequency Counter 2, track A		0x02
I-3104/x	Input frequency Counter 2, track B		0x02
I-3105/x	Alarm response counter 1	Setting activates process alarm Bit 1 ... 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 ... 6: Reserved	0x00
I-3106/x	Counter function counter 1	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 ... 6: Reserved	0x00
I-3107/x	Comparator counter 1	Bit 2 ... 0: Comparison bit is set (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00

Index/subindex	Name	Description/value	Lenze
I-3108/x	Signal evaluation counter 1	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
I-3109/x ... I-310C/x	Set value counter 1	When a set value is given, the counter can be loaded with the set value. With an edge 0-1 at COUNTERVAL_SET in the control word, the set value is accepted in the counter. Counting method: • 0x3109: byte 3 (high byte) • 0x310A: byte 2 • 0x310B: byte 1 • 0x310C: byte 0 (low byte)	0x00
I-310D/x ... I-3110/x	Final value counter 1	Upper limitation of the counting range Counting method: • 0x310D: byte 3 (high byte) • 0x310E: byte 2 • 0x310F: byte 1 • 0x3110: byte 0 (low byte)	0x00
I-3111/x ... I-3114/x	Loading value counter 1	Lower limitation of the counting range Counting method: • 0x3111: byte 3 (high byte) • 0x3112: byte 2 • 0x3113: byte 1 • 0x3114: byte 0 (low byte)	0x00
I-3115/x	Hysteresis counter 1	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
I-3116/x	Reserved		
I-3117/x	Alarm response counter 2	See counter 1	0x00
I-3118/x	Counter function counter 2	See counter 1	0x00
I-3119/x	Comparator counter 2	See counter 1	0x00
I-311A/x	Signal evaluation counter 2	See counter 1	0x00
I-311B/x ... I-311E/x	Set value counter 2	See counter 1	0x00

Index/subindex	Name	Description/value	Lenze
I-311F/x ... I-3122/x	Final value counter 2	See counter 1	0x00
I-3123/x ... I-3126/x	Loading value counter 2	See counter 1	0x00
I-3127/x	Hysteresis counter 2	See counter 1	0x00
I-3128/x I-3129/x	Reserved		
I-310A/x	Signal evaluation	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Hardware gate (HW gate) • 0b000 = deactivated (counter starts by setting SW gate) • 0b001 = activated (HIGH level at gate activates the HW gate. Counter starts if HW and SW gate are set.) Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	
I-310B/x ... I-310E/x	Final value counter 1	Upper limitation of the counting range Counting method: • 0x310D: byte 3 (high byte) • 0x310E: byte 2 • 0x310F: byte 1 • 0x3110: byte 0 (low byte)	0x00
I-310F/x ... I-3112/x	Loading value counter 1	Lower limitation of the counting range Counting method: • 0x3111: byte 3 (high byte) • 0x3112: byte 2 • 0x3113: byte 1 • 0x3114: byte 0 (low byte)	0x00
I-3113/x	Hysteresis	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
I-3114/x	Pulse	The pulse duration indicates for how long the output is to be set if the parameterised comparison criterion is reached or exceeded. The pulse duration can be specified in steps of 2.048 ms between 0 and 522.24 ms. If the pulse duration is = 0, the output is set until the comparison condition is no longer met.	0x00
I-3115/x ... I-3129/x	Reserved		

5.10.5 1 counter 32 bits, 5 V DC - EPM-S602

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Further information: ▶ Product description (📖 15)

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
I-3101/x	Input frequency Track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
I-3102/x	Input frequency Track B		0x02
I-3103/x	Input frequency Reset		0x02
I-3104/x	Reserved		
I-3105/x	Alarm response	Setting activates process alarm Bit 1 ... 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 ... 6: Reserved	0x00
I-3106/x	Numerator function	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 ... 6: Reserved	0x00
I-3107/x	Comparator	Bit 2 ... 0: Comparison bit is set (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 ... 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00

Index/subindex	Name	Description/value	Lenze
I-3108/x	Signal evaluation	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
I-3109/x ... I-310C/x	Final value counter 1	Upper limitation of the counting range Counting method: • 0x3109: byte 3 (high byte) • 0x310A: byte 2 • 0x310B: byte 1 • 0x310C: byte 0 (low byte)	0x00
I-310D/x ... I-3110/x	Loading value counter 1	Lower limitation of the counting range Counting method: • 0x310D: byte 3 (high byte) • 0x310E: byte 2 • 0x310F: byte 1 • 0x3110: byte 0 (low byte)	0x00
I-3111/x	Hysteresis		0x00
I-3112/x ... I-3129/x	Reserved		0x00

5.10.6 2 counters 32 bits, 24 V DC - EPM-S603

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from 0 to the counting limit, then skips to the opposite counting limit and continues to count from there.

<table>
<tr><th>Signal evaluation</th><th>Description</th></tr>
<tr><td>Single rotary transducer</td><td rowspan="2">Connection to "A/pulse" input</td></tr>
<tr><td>Double rotary transducer</td></tr>
<tr><td>Quadruple rotary transducer</td><td>Connection to input "A/pulse" and "B/direction"</td></tr>
<tr><td>Direction</td><td>Pulse at "A/pulse" input and direction at "B/direction" input</td></tr>
</table>

Additional functions	Description
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area).

Further information: ▶ Product description (📖 15)

Parameter data

<table>
<tr><th>Index/subindex</th><th>Name</th><th>Description/value</th><th>Lenze</th></tr>
<tr><td>I-3100/x</td><td>Input frequency
Counter 1, track A</td><td rowspan="4">Filters for instance serve to filter signal peaks in the case of an unclean input signal.
0 (0x00): 500 kHz
1 (0x01): 300 kHz
2 (0x02): 100 kHz
3 (0x03): 60 kHz
4 (0x04): 30 kHz
6 (0x06): 10 kHz
7 (0x07): 5 kHz
8 (0x08): 2 kHz
9 (0x09): 1 kHz
Other values are not permissible!</td><td>0x02</td></tr>
<tr><td>I-3101/x</td><td>Input frequency
Counter 1, track B</td><td>0x02</td></tr>
<tr><td>I-3102/x</td><td>Input frequency
Counter 2, track A</td><td>0x02</td></tr>
<tr><td>I-3103/x</td><td>Input frequency
Counter 2, track B</td><td>0x02</td></tr>
<tr><td>I-3104/x</td><td>Counting direction counter 1, track B</td><td>Bit 2 ... 0: Reserved
Bit 3: Invert counting direction track B
0 = no (do not invert)
1 = yes (invert)
Bit 7 ... 4: Reserved</td><td>0x00</td></tr>
<tr><td>I-3105/x</td><td>Signal evaluation counter 1</td><td>Bit 2 ... 0: Signal evaluation
• 0b000 = counter deactivated (the other parameter data details for the counter are ignored)
• 0b001 = rotary transducer 1-fold (connection to "A/pulse" input)
• 0b010 = rotary transducer 2-fold (connection to "A/pulse" input)
• 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction")
• 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input)
Bit 7 ... 3: Reserved</td><td>0x00</td></tr>
</table>

Index/subindex	Name	Description/value	Lenze
I-3106/x	Counting direction counter 2, track B	Bit 2 ... 0: Reserved Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
I-3107/x	Signal evaluation counter 2	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 7 ... 3: Reserved	0x00
I-3108/x ... I-3129/x	Reserved		

5.11 Parameterising the encoder evaluation

5.11.1 SSI - EPM-S604

Further information: Product description (15)

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
I-3101/x	Dead time HIGH byte	The dead time, also called tbs (time between sends), defines the waiting time between two encoder values to be observed by the module so that the encoder is able to process its value. This data can be found in the data sheet for your encoder. HIGH LOW 0x00 0x30: 1 µs 0x00 0x60: 2 µs 0x00 0xC0: 4 µs 0x01 0x80: 8 µs 0x03 0x00: 16 µs 0x06 0x00: 32 µs 0x09 0x00: 48 µs 0x0C 0x00: 64 µs	0x0C
I-3102/x	Dead time LOW byte		0x00
I-3103/x	Baud rate HIGH byte	In "monitoring operation" operating mode, the baud rate is irrelevant. Enter the baud rate here. This corresponds to the clock frequency via which the encoder connected communicates. Information on this can be found in the data sheet for your encoder. HIGH LOW 0x00 0x18: 2 MHz 0x00 0x20: 1.5 MHz 0x00 0x30: 1 MHz 0x00 0x60: 500 kHz 0x00 0xC0: 250 kHz 0x01 0x80: 125 kHz	0x01
I-3104/x	Baud rate LOW byte		0x80
I-3105/x	Reserved		
I-3106/x	Standardisation	Depending on the encoder, further bits are transmitted in addition to the encoder value. Scaling serves to determine how many bits post-positioned to the encoder value will be removed by shifting the encoder value to the right. The encoder value is scaled by the module only after a Gray-binary conversion. More information can be found in the data sheet for your encoder. Value range: 0x00 ... 0x0F = 0 bit ... 15 bits	0x00

Index/subindex	Name	Description/value	Lenze
I-3107/x	Bit length of encoder data	Enter the bit length of the encoder data here. Depending on the encoder, the encoder data consist of the current encoder value with subsequent bits. The total length has to be specified here. More information on this can be found in the data sheet for your encoder. 7 (0x07) = "8 bits" 8 (0x08) = "9 bits" 9 (0x09) = "10 bits" 10 (0x0A) = "11 bits" 11 (0x0B) = "12 bits" 12 (0x0C) = "13 bits" 13 (0x0D) = "14 bits" 14 0x0E) = "15 bits" 15 (0x0F) = "16 bits" 16 (0x10) = "17 bits" 17 (0x11) = "18 bits" 18 (0x12) = "19 bits" 19 (0x13) = "20 bits" 20 (0x14) = "21 bits" 21 (0x15) = "22 bits" 22 (0x16) = "23 bits" 23 (0x17) = "24 bits" 24 (0x18) = "25 bits" 25 (0x19) = "26 bits" 26 (0x1A) = "27 bits" 27 (0x1B) = "28 bits" 28 (0x1C) = "29 bits" 29 (0x1D) = "30 bits" 30 (0x1E) = "31 bits" 31 (0x1F) = "32 bits"	0x18

Index/subindex	Name	Description/value	Lenze
I-3107/x	Bit length of encoder data	Enter the bit length of the encoder data here. Depending on the encoder, the encoder data consist of the current encoder value with subsequent bits. The total length has to be specified here. More information on this can be found in the data sheet for your encoder. 7 (0x07) = "8 bits" 8 (0x08) = "9 bits" 9 (0x09) = "10 bits" 10 (0x0A) = "11 bits" 11 (0x0B) = "12 bits" 12 (0x0C) = "13 bits" 13 (0x0D) = "14 bits" 14 0x0E) = "15 bits" 15 (0x0F) = "16 bits" 16 (0x10) = "17 bits" 17 (0x11) = "18 bits" 18 (0x12) = "19 bits" 19 (0x13) = "20 bits" 20 (0x14) = "21 bits" 21 (0x15) = "22 bits" 22 (0x16) = "23 bits" 23 (0x17) = "24 bits" 24 (0x18) = "25 bits" 25 (0x19) = "26 bits" 26 (0x1A) = "27 bits" 27 (0x1B) = "28 bits" 28 (0x1C) = "29 bits" 29 (0x1D) = "30 bits" 30 (0x1E) = "31 bits" 31 (0x1F) = "32 bits"	0x18

Index/subindex	Name	Description/value	Lenze
I-3108/x		Bit 1 ... 0: Ready for operation During "Monitoring operation" the module serves to monitor the data exchange between an SSI master and an SSI encoder. It receives the cycle by the master and the data flow by the SSI encoder. In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder. 0b01 = monitoring operation 0b10 = master mode Bit 2: Shifting direction Specify the orientation of the encoder data here. More information can be found in the data sheet for your encoder. Usually, the SSI encoder uses "MSB first". 0 = LSB first (LSB is transmitted first) 1 = MSB first (MSB is transmitted first) Bit 3: Edge clock signal Here you can specify the edge type of the clock signal, in the case of which the encoder supplies data. Information on this can be found in the data sheet for your encoder. Usually the SSI encoders respond to rising edges. 0 = falling edge 1 = rising edge **Master mode:** Connect clock output signal (ClockOut) to the EPM-S604. **Monitoring mode:** Connect clock input signal (ClockIn) to the EPM-S604. Bit 4: Coding In the "Binary code" setting, the provided encoder value remains unchanged. In the "Gray code" setting, the Gray-coded value provided by the encoder is converted into a binary value. Only after this conversion, the received encoder value is scaled, if required. The Gray code is another form of representation of the binary code. It is based on the fact that two adjacent Gray numbers differ from each other in exactly one bit. If the Gray code is used, transmission errors can be easily detected, since adjacent characters must only differ from each other in one digit. Information on this can be found in the data sheet for your encoder. 0 = standard code 1 = Gray code Bit 7 ... 5: Reserved	0x1E
I-3109/x ... I-310B/x	Reserved		
I-310C/x	SSI function	By enabling the SSI function the module starts with the cycle output and the evaluation of the encoder data. In the "monitor operation" mode, the module starts with the encoder evaluation. 0 (0x00) = inhibited 1 (0x01) = enabled	0x00
I-310D/x ... I-3129/x	Reserved		

5.12 Parameterising the time stamp

5.12.1 2 digital inputs with time stamp function - EPM-S207

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C (fix)
I-3101/x	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)
I-3102/x	Input delay DI1	0x00 = 1 μs 0x02 = 3 μs 0x04 = 10 μs 0x07 = 86 μs 0x09 = 342 μs 0x0C = 273 μs No other values are permissible.	0x02
I-3103/x	Input delay DI2		0x02
I-3104/x	Edge 0-1 at DIx	Time stamp entry on rising edge Bit 0: DI1 (0: inhibit, 1 = enable) Bit 1: DI2 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00
I-3105/x	Edge 1-0 at DIx	Time stamp entry on falling edge Bit 0: DI1 (0: inhibit, 1 = enable) Bit 1: DI2 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00
I-3106/x ... I-3129/x	Reserved		

5.12.2 2 digital outputs with time stamp function - EPM-S310

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C (fix)
I-3101/x	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)
I-3102/x ... I-3129/x	Reserved		

5.13 Parameterising technology modules

5.13.1 2 digital outputs with PWM functionality - EPM-S620

Parameter data

Index/subindex	Name	Description/value	Lenze
I-3100/x	PWM 1: Period byte 3	Set parameters here for the total time for pulse duration and pulse pause. The time should be selected as a factor for the 20.83 ns basis. Values below 25 µs are ignored. If the pulse duration is higher or equal to the period, the DO output is set permanently. Value range: 1200 ... 8388607 (25 µs ... approx. 175 ms)	0x1F40
I-3101/x	PWM 1: Period byte 2		
I-3102/x	PWM 1: Period byte 1		
I-3103/x	PWM 1: Period byte 0		
I-3104/x	PWM 2: Period byte 3		0x1F40
I-3105/x	PWM 2: Period byte 2		
I-3106/x	PWM 2: Period byte 1		
I-3107/x	PWM 2: Period byte 0		
I-3108/x ... I-3129/x	Reserved		
I-5620/x	PWM pulse duration	Selection of pulse duration Permissible values: 48 ... 8388607 (corresponds to 1 ... 175000 µs	

Tip!

The pulse/pause ratio is determined by specifying the period (0x3100 ... 0x3107) and pulse duration (0x5620).

5.13.2 RS232 interface - EPM-S640

Parameter data - ASCII protocol

Index/subindex	Name	Description/value	Lenze
I-3100/x	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3101/x	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3102/x	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
I-3103/x	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
I-3104/x	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x01: ASCII	0x01
Option 1, drawing frame			
I-3105/x	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			

Index/subindex	Name	Description/value	Lenze
I-3106/x	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
I-3107/x	ZNA (LOW byte)		0x00
Option 4 / 5, character delay time			
I-3108/x	Character delay time (HIGH byte)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
I-3109/x	Character delay time (LOW byte)		0xFA
Option 6, number of receive buffers			
I-310A/x	Number of receive buffers	Defines the number of receive buffers. As long as only one receive buffer is used and is assigned, no more data can be received. If up to 250 receive buffers are connected, the received data can be redirected to a still free receive buffer. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7 ... 12, reserved			
I-310B/x ... I-3110/x	Reserved		0x00

Parameter data STX/ETX protocol

Index/subindex	Name	Description/value	Lenze
I-3100/x	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3101/x	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3102/x	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
I-3103/x	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
I-3104/x	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame			
I-3105/x	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			
I-3106/x	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
I-3107/x	ZNA (LOW byte)		0x00
Option 4 / 5, TMO			

Index/subindex	Name	Description/value	Lenze
I-3108/x	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
I-3109/x	TMO (LOW byte)		0xFA
Option 6, number of start identifiers			
I-310A/x	Number of start identifiers	0x00: 1 start identifier (2nd start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1			
I-310B/x	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2			
I-310C/x	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers			
I-310D/x	Number of end identifiers	0x00: 1 end identifier (2nd end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1			
I-310E/x	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2			
I-310F/x	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved			
I-3110/x	Reserved		0x00

Parameter data 3964(R) protocol

Index/subindex	Name	Description/value	Lenze
I-3100/x	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3101/x	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3102/x	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
I-3103/x	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
I-3104/x	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame			
I-3105/x	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)			
I-3106/x	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00

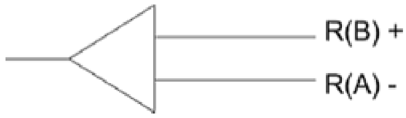
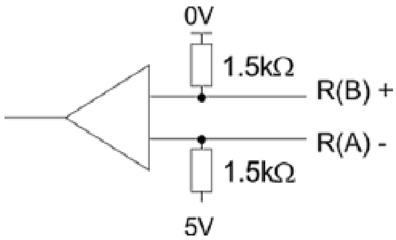
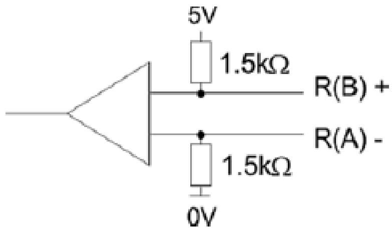
Index/subindex	Name	Description/value	Lenze
Option 3, character delay time (x 20 ms)			
I-3107/x	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)			
I-3108/x	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)			
I-3109/x	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repetitions			
I-310A/x	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL			
I-310B/x	DBL	If the block wait time is exceeded, you can select the maximum number of repetitions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority			
I-310C/x	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved			
I-310D/x ... I-3110/x	Reserved		0x00

5.13.3 RS422/RS485 interface - EPM-S650

Parameter data - ASCII protocol

Index/subindex	Name	Description/value	Lenze
I-3100/x	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3101/x	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3102/x	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
I-3103/x	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
I-3104/x	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x01: ASCII	0x01
Option 1, drawing frame			
I-3105/x	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			

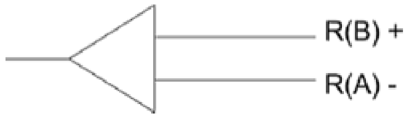
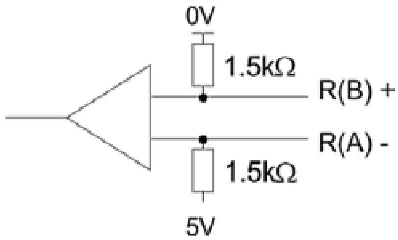
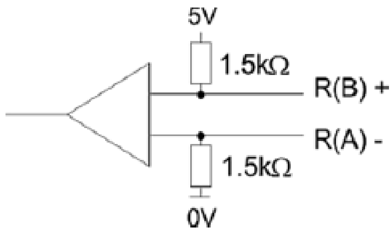
Index/subindex	Name	Description/value	Lenze
I-3106/x	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
I-3107/x	ZNA (LOW byte)		0x00
Option 4 / 5, character delay time			
I-3108/x	Character delay time (HIGH byte)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
I-3109/x	Character delay time (LOW byte)		0xFA
Option 6, number of receive buffers			
I-310A/x	Number of receive buffers	Defines the number of receive buffers. As long as only one receive buffer is used and is assigned, no more data can be received. If up to 250 receive buffers are connected, the received data can be redirected to a still free receive buffer. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7 ... 12, reserved			
I-310B/x ... I-3110/x	Reserved		0x00
Option 13, operating mode			
I-3111/x	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parameter setting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

Index/subindex	Name	Description/value	Lenze
Option 14, cable assignment			
I-3112/x	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible. 	0x00

Parameter data STX/ETX protocol

Index/subindex	Name	Description/value	Lenze
I-3100/x	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3101/x	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3102/x	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
I-3103/x	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
I-3104/x	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame			
I-3105/x	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			
I-3106/x	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
I-3107/x	ZNA (LOW byte)		0x00
Option 4 / 5, TMO			

Index/subindex	Name	Description/value	Lenze
I-3108/x	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
I-3109/x	TMO (LOW byte)		0xFA
Option 6, number of start identifiers			
I-310A/x	Number of start identifiers	0x00: 1 start identifier (2. start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1			
I-310B/x	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2			
I-310C/x	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers			
I-310D/x	Number of end identifiers	0x00: 1 end identifier (2. end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1			
I-310E/x	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2			
I-310F/x	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved			
I-3110/x	Reserved		0x00
Option 13, operating mode			
I-3111/x	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

Index/subindex	Name	Description/value	Lenze
Option 12, cable assignment			
I-3112/x	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible. 	0x00

Parameter data 3964(R) protocol

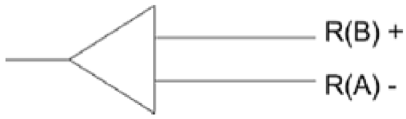
Index/subindex	Name	Description/value	Lenze
I-3100/x	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3101/x	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
I-3102/x	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
I-3103/x	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
I-3104/x	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame			
I-3105/x	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)			
I-3106/x	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00

Index/subindex	Name	Description/value	Lenze
Option 3, character delay time (x 20 ms)			
I-3107/x	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)			
I-3108/x	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)			
I-3109/x	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repititons			
I-310A/x	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL			
I-310B/x	DBL	If the block wait time is exceeded, you can select the maximum number of repititions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority			
I-310C/x	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved			
I-310D/x ... I-3110/x	Reserved		0x00
Option 13, operating mode			

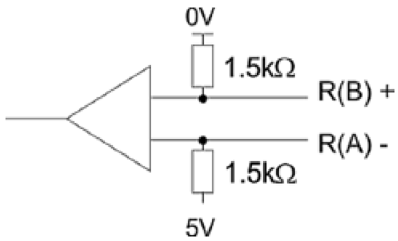
<table>
<tr><th>Index/subindex</th><th>Name</th><th>Description/value</th><th>Lenze</th></tr>
<tr><td>I-3111/x</td><td>Operating mode</td><td>The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422).

0x00: Half duplex - two-wire operation (RS485)
Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions.
In case of the half duplex parametersetting under RS485, no software data flow control is possible.
0x01: Full duplex - four-wire operation (RS422)
The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.
</td><td>0x01</td></tr>
<tr><td colspan="4">Option 12, cable assignment</td></tr>
<tr><td>I-3112/x</td><td>Cable assignment</td><td>For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level.

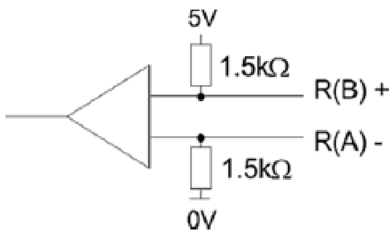
0x00:
No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.



0x01:
Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V
In full duplex operation under RS422, open-circuit monitoring is possible.



0x02:
Signal R(A) 0V, signal R(B) 5V
This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485.
Open-circuit monitoring is not possible.



</td><td>0x00</td></tr>
</table>

5.14 "Save" function

All parameter changes and the structure of the station are permanently stored via index I-1010 (Store parameter).

Note!

If the real structure of the station does not correspond with its stored structure anymore after a restart, the bus coupler module reports an error.

- The LED SF (red) is lit permanently.
- The LED PWR (green) is lit permanently.
- The LED CAN-RUN (green) is blinking.

The stored structure of the station is deleted via index I-1011 (Restore all parameters).

Index	Name	Possible settings		Info
		Lenze	Selection	
I-1010	Save All Parameters	0		Saving the parameter settings and the station structure in the EEPPROM of the bus coupler. Function in accordance with CANopen (DS301/DS401 communication protocol).
0			Number of subindexes assigned	Display of the subindices user by the index I-1010
1	Save		0 = no function 1702257011 = save parameters	The numerical value is ASCII-coded and complies with: 65 76 61 0x73 = "E" "V" "A" "S"

5.15 Loading the default setting

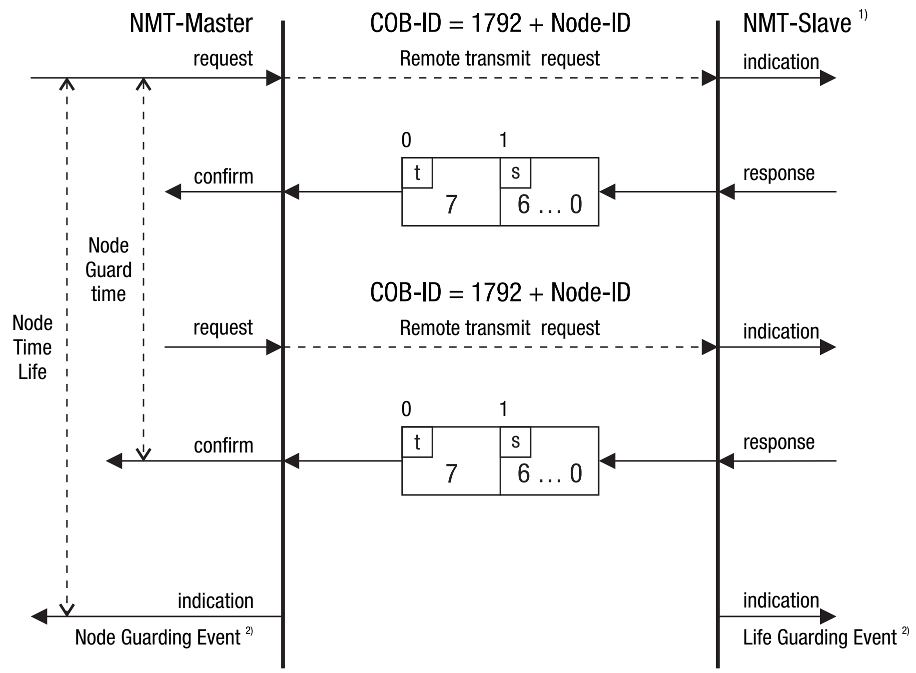
All parameter changes are reset to the default setting via index I-1011.

The default setting is accepted by switching the supply off and on again. Changes that you may have carried out before will be deleted from the EEPROM of the I/O system.

Index	Name	Possible settings		Info
		Lenze	Selection	
I-1011	Restore All Parameters	0		Resetting the parameter setting in the EEPPROM of the bus coupler to the Lenze setting. Function in accordance with CANopen (DS301/DS401 communication protocol).
0			Number of subindexes assigned	Read only Number of the subindices used by the index I-1011
1	Load		0 = no function 1684107116 = load Lenze setting	The numerical value is ASCII-coded and complies with: 64 61 6F 0x6C = "D" "A" "O" "L" After executing "Load", acceptance is effected by • Voltage off/on, or • Transmission of a reset node telegram (00 82 xx, with xx = node address)

5.16 Node guarding

Node Guarding Protocol



1)	I/O system
s	Status of the I/O system
t	Toggle bit

[5-13] Node Guarding Protocol

The Node Guarding Protocol monitors the connection between master and slave.

Via index I-100C, "Guard time", a time [ms] can be set and in index , Life Time Factor, a factor can be set. If both indexes are multiplied by each other, you get a monitoring time in which the master must send a Node Guarding telegram to the slave. If one of the two indexes is set to zero, the monitoring time is also zero and hence deactivated. The slave sends a telegram with its current status to the master.

With event-controlled process data transmission, Node Guarding ensures cyclical node monitoring.

- The master starts the Node Guarding by sending the Node Guarding telegram.
- If the slave (I/O system) does not receive a telegram within the monitoring time, a Node Guarding event will be triggered. The I/O system switches to the status set in I-1029. The outputs switch to defined states).
- A reset takes place by a change to Operational.

Status telegram

11 Bit	1 byte of user data	
Identifier	Device status (bits 0 ... 6)	Toggle bit
1792 (0x700)		

Identifier:

	Formula	Info
Identifier	= Basic identifier + node address = 1792 + xx	The basic identifier for node guarding is permanently set to 1792 (0x700) xx = Node address of the I/O system

Device status (bit 0 ... 6) of the slave (I/O system):

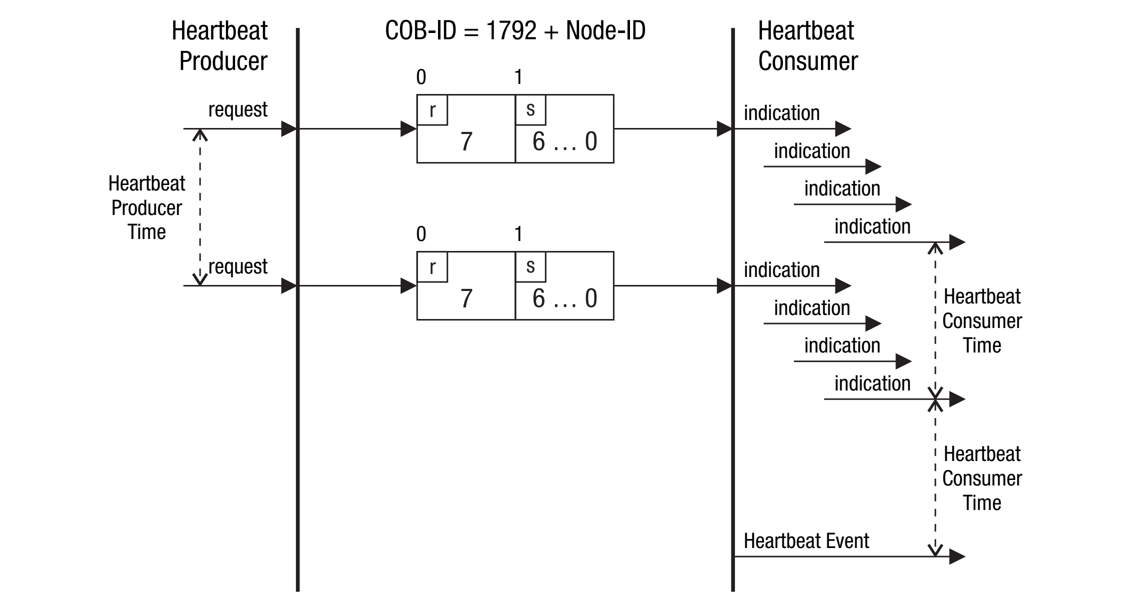
Command (hex)	Formula	Info
04	Stopped	
05	Operational	
7F	Pre-Operational	

Indices for setting

Index	Name	Possible settings				Info
		Lenze	Selection			
I-100C *	Guard time	0	0	{1 ms}	65535	Node guarding Monitoring time 0 = monitoring not active
I-100D *	Life time factor	0	0	{1}	255	Node guarding Response time computation factor 0 = monitoring not active The response time is computed as: Monitoring time × factor
I-100E	Node Guarding identifier					Read only Identifier = basic identifier + node address (basic identifier cannot be modified)

5.17 Heartbeat

Heartbeat protocol



r	Reserved
s	Heartbeat producer status

[5-14] Heartbeat protocol

Heartbeat producer

The I/O system transmits a status telegram to the fieldbus and can thus be monitored by other nodes.

Settings are made in index I-1017.

- Producer heartbeat is automatically started if a time > 0 is entered in the index I-1017 and the I/O system changes to the status "Operational".
- After the cycle time has elapsed, the status telegram is transferred from the I/O system to the fieldbus.
- A reset takes place by a change to Operational.

Status telegram

11 Bit	1 byte of user data	
Identifier	Device status (bits 0 ... 6)	Toggle bit
1792 (0x700)	Reserved	

Identifier:

	Formula	Info
Identifier	= Basic identifier + node address = 1792 + xx	The basic identifier for node guarding is permanently set to 1792 (0x700) xx = Node address of the I/O system

Device status (bit 1 ... 6) of the heartbeat producer:

Command (hex)	Formula	Info
00	Boot-up	
04	Stopped	
05	Operational	
7F	Pre-Operational	

5.18 Monitoring

5.18.1 Time monitoring for PDO1-Rx ... PDO10-Rx

A time monitoring can be configured for the inputs of the process data objects PDO1-Rx ... PDO10-Rx via index I-2400.

If no PDO is received within the time set in I-2400, the outputs will switch to their defined error status (see following sections).

Index		Name	Possible settings				Info
			Lenze	Selection			
I-2400		Timer value		0	{1 ms}	65535	Monitoring time for process data input objects
	1	Lenze-PDO Control 1	0				
	2	Lenze-PDO Control 2	0				
	3	Lenze-PDO Control 3	0				
	4	Lenze-PDO Control 4	0				
	5	Lenze-PDO Control 5	0				
	6	Lenze-PDO Control 6	0				
	7	Lenze-PDO Control 7	0				
	8	Lenze-PDO Control 8	0				
	9	Lenze-PDO Control 9	0				
	10	Lenze PDO control 10	0				

5.18.2 Monitoring of the digital outputs

Use index I-6206I-6206 to configure the response of the digital outputs if no telegram, node guarding event, or heartbeat have been received within the defined monitoring time.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-6206		Error mode - digital output		0	{1}	255	Configures digital output monitoring
				I-6206	0 All digital outputs retain the last status output.		
					255 Response from I-6207		In I-6207, the response can be configured individually for each digital output
	1	Byte 1	0				
	2	Byte 2	0				
	...	...	...				
	80	Byte 80	0				

Individual response setting

Via index I-6207, the response can be individually configured for each digital output.

Index		Name	Possible settings						Info
			Lenze	Selection					
I-6207 Configures the individual digital output responses		Error value - digital output	0	0		{1}		255	Configures the individual digital output responses
				I-6207 8 bits of information					
				Bit value 0	Output switches to LOW				
				Bit value 1	Output retains last status output				
	1	Byte 1	0						
	2	Byte 2	0						
	...	...	...						
	64	Byte 80	0						

5.18.3 Monitoring of the analog outputs

Use index I-6443 to configure the response of the analog outputs if no telegram, node guarding event, or heartbeat has been received within the defined monitoring time.

- Monitoring is started if the next PDO telegram is received after the settings.
- If a telegram is not transmitted within the time set, the modules switches to the Pre-Operational state. Process data are not transmitted anymore.
- A reset takes place by a change to Operational.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-6443		Error mode analog output		0	{1}	255	Configures analog output monitoring
				I-6443	0 All analog outputs retain the last value output		
					255 Response from I-6444		In I-6444, the response can be configured individually for each analog output
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				

Individual response setting

Via index I-6444, the response can be individually configured for each analog output.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-6444		Error value analog output		-32768	{1}	32767	Configures the individual analog output responses The analog outputs provide the set value
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				

5.19 Diagnostics

Note!

The diagnostic function is only supported by I/O compound modules from HW version 1B and by bus coupler modules from HW version 1A.

If a module in the system complies with an earlier HW version, the diagnostic function is deactivated for all modules.

The following indices can be used for purposes of diagnostics. They show operating states. Settings cannot be made.

Index	Information displayed
I-1014	Emergency telegram
I-1027	Reading out the module IDs
I-6000	Status of digital inputs
I-6200	Status of digital outputs
I-6401	Status of analog inputs
I-6411	Status of analog outputs
I-5400 ... I-5403	Counter status
I-1003	Current errors

▸ Index table (📖 193)

5.19.1 Emergency telegram

By means of the emergency telegram, the I/O system communicates internal device errors to other system bus nodes with high priority. 8 bytes of user data are available.

Index		Name	Possible settings		Info
			Lenze	Selection	
I-1003					Read only, fault memory
	1	Actual errors			
I-1014		COB ID emergency			Emergency telegram After the boot-up, the identifier 0x80 + node address are displayed.

Emergency telegram structure

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
LOW byte	HIGH byte	Error register I-1001	Error information				
Error code	Error code	The error code 0x81 (device error) is displayed in the index I-1001 (error register).	1	2	3	4	5

5.19.1.1 Error codes

Error code	Meaning	Error information				
		1	2	3	4	5
0x0000	Reset Emergency	0x00	0x00	0x00	0x00	0x00
0x8100	Heartbeat consumer	Node ID	LOW byte timer value	HIGH byte timer value	0x00	0x00
0x8130	Node guarding error	LOW byte guard time	HIGH byte guard time	LifeTime	0x00	0x00
0x8157	Module removed from slot [n]; no communication	0x05	0x[n]	0x00	0x00	0x00
0x8210	PDO not executed due to length error	PDO number	Wrong length	PDO length	0x00	0x00
0x8220	PDO length exceeded	PDO number	0x1000	PDO number	0x00	0x00
0x1000	Module configuration was changed	0x01	0x00	0x00	0x00	0x00
0x1000	Error during initialisation of the backplane modules	0x02	Module number	LOW byte Error register	HIGH byte Error register	0x00
0x1000	Diagnosealarm	0x40 + module number	Diagnostic byte 1	Diagnostic byte 2	Diagnostic byte 3	Diagnostic byte 4
0x0000	Process alarm	0x80 + module number	Diagnostic byte 1	Diagnostic byte 2	Diagnostic byte 3	Diagnostic byte 4
0x1000	Backplane bus: Initialisation error	E0	Module number	LOW byte Error bitfield	HIGH byte Error bitfield	0x00
0x1000	Backplane bus: Initialisation error Pre-Operational → Operational	E0	0x00	0x00	0x00	0x00
0x1000	Backplane bus: Bus error	E1	0x00	0x00	0x00	0x00
0x1001	Lenze PDO control, monitoring time exceeded	FF	0x10	PDO number	Monitoring time that has been set in [ms]	
0x2000	Description of the process data width not permissible for modules with time stamp functionality or serial interfaces (I-3100/x, I-3101/x)	E2	0x00	0x00	0x00	0x00

5.19.1.2 Example: Error information "46 0D 15 00 00"

The error information "46 0D 15 00 00" is contained in the Emergency telegram (181) in the bytes 3 ... 7.

Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Error information				
1	2 (Diagnostic byte 1)	3 (Diagnostic byte 2)	4 (Diagnostic byte 3)	5 (Diagnostic byte 4)
0x46 (0x40 + slot no. '6')	0x0D	0x15	0x00	0x00

- Byte 4: 0x0D = 0b00001101
  Meaning: Module fault, external error, channel error available.

Byte 4 (Diagnostic byte 1)								Bit information
0	0	0	0	1	1	0	1	
								Bit 0: Module fault
								Bit 1: Internal error
								Bit 2: External error
								Bit 3: Channel error
								Bit 4: No external supply voltage
								Bit 5: Reserved
								Bit 6: Reserved
								Bit 7: Parameterisation error

- Byte 5: 0x15 = 0b00010101
  Meaning: Module class "analog module", channel information (channel error) available.

Byte 5 (Diagnostic byte 2)								Bit information
0	0	0	0	0	1	0	1	
								Bit 0 ... 3: Module class • 0b0101 = analog module • 0b1000 = function module • 0b1100 = communication module
								Bit 4: channel information available
								Bit 5: Reserved
								Bit 6: Reserved
								Bit 7: Reserved

The module-specific bit information of the "Diagnostic Bytes 1 ... 4" (process/diagnostic alarms) are described in the following sections.

5.19.1.3 EPM-S404 - process alarm

Diagnostic byte	Bit 7 ... 0
1	Bit 0: Limit value exceeded channel 1 Bit 1: Limit value exceeded channel 2 Bit 7 ... 2: 0 (fixed)
2	Bit 0: Limit value not reached, channel 1 Bit 1: Limit value not reached, channel 2 Bit 7 ... 2: 0 (fixed)
3/4	Ticker value at the time of the alarm

5.19.1.4 EPM-S406, EPM-S408 - process alarm

Diagnostic byte	Bit 7 ... 0
1	Bit 0: Limit value exceeded channel 1 Bit 1: Limit value exceeded channel 2 Bit 7 ... 2: 0 (fixed)
2	Bit 0: Limit value not reached, channel 1 Bit 1: Limit value not reached, channel 2 Bit 7 ... 2: 0 (fixed)
3/4	μs ticker value at the time of the alarm The I/O compound module features an integrated 32-bit timer (μs-Ticker) which is started at switch-on and restarts at 0 after 232 - 1 μs. These two bytes represent the lower two bytes of the μs ticker (0 ... 216 - 1)

5.19.1.5 EPM-S600 - process alarm

Diagnostic byte	Bit 7 ... 0
1	Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bit 7 ... 5: 0 (fixed)
2	State of the inputs at the time of the alarm Bit 0: A/pulse Bit 1: B/direction Bit 2: Latch Bit 3: Hardware gate Bit 4: Reset Bit 7 ... 5: 0 (fixed)
3/4	Ticker value at the time of the alarm

5.19.1.6 EPM-S601, EPM-S602 - process alarm

Diagnostic byte	Bit 7 ... 0
1	Bit 0: 0 Bit 1: 0 Bit 2: Counter 1, overflow/underflow/final value Bit 3: Counter 1, comparison value reached Bit 4: 0 Bit 5: 0 Bit 6: Counter 2, overflow/underflow/final value Bit 7: Counter 2, comparison value reached
2	State of the inputs at the time of the alarm Bit 0: Counter 1, A/pulse Bit 1: Counter 1, B/direction Bit 2: Counter 2, A/pulse Bit 3: Counter 2, B/direction Bit 7 ... 4: 0 (fixed)
3/4	16 bit μs value at the time of the alarm

5.19.1.7 EPM-S600 - diagnostic alarm

Diagnostic byte	Bit 7 ... 0
1	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
2	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)

5.19.1.8 EPM-S601 - diagnostic alarm

Diagnostic byte	Bit 7 ... 0
1	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
2	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)

5.19.1.9 EPM-S603 - diagnostic alarm

Diagnostic byte	Bit 7 ... 0
1	0 (fixed)
2	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
3	0 (fixed)
4	0 (fixed)

5.19.1.10 EPM-S604 - diagnostic alarm

Diagnostic byte	Bit 7 ... 0
1	Bit 0: Set in case of module fault Bit 1: Set in case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of missing external supply voltage Bit 6 ... 5: 0 (fixed) Bit 7: Parameterisation error
2	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)

5.19.1.11 EPM-S640 - diagnostic alarm

Diagnostic byte	Bit 7 ... 0
1	Bit 0: set in the case of module fault Bit 1: set in the case of internal error Bit 2: set in the case of external error (cable break only for RS422) Bit 3: 0 (fixed) Bit 4: set in the case of missing external supply voltage Bit 5, 6: 0 (fixed) Bit 7: set in the case of parameterisation error
2	Bit 3 ... 0: Module class, 0b1100: Communication module Bit 4: Set if channel information available Bit 7 ... 5: 0 (fix)
3	Bits 3 ... 0: 0 (fixed) Bit 4: set in the case of internal communication error Bits 7 ... 5: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x60: Communication processor Bit 7: 0 (fix)

5.19.1.12 EPM-S650 - diagnostic alarm

Diagnostic byte	Bit 7 ... 0
1	Bit 0: set in the case of module fault Bit 1: set in the case of internal error Bit 2: set in the case of external error (cable break only for RS422) Bit 3: 0 (fixed) Bit 4: set in the case of missing external supply voltage Bit 5, 6: 0 (fixed) Bit 7: set in the case of parameterisation error
2	Bit 3 ... 0: Module class, 0b1100: Communication module Bit 4: Set if channel information available Bit 7 ... 5: 0 (fix)
3	Bits 3 ... 0: 0 (fixed) Bit 4: set in the case of internal communication error Bits 7 ... 5: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x60: Communication processor Bit 7: 0 (fix)

5.19.2 Reading out the module IDs

The number of I/O compound modules connected and the module types used can be read out via index I-1027. Each module type can be identified unambiguously via a hex value.

Index	Subindex	Read out...	Module type	Module ID	
				dec	hex
I-1027	0	... the number of plugged modules (0 ... 64)	–	23	017
	1 ... 64	... the module type in slots 1 ... 64	EPM-S200	1	001
			EPM-S201	3	003
			EPM-S202	5	005
			EPM-S203	8	008
			EPM-S204	2	002
			EPM-S205	4	004
			EPM-S206	7	007
			EPM-S207	3841	F01
			EPM-S300	257	101
			EPM-S301	260	104
			EPM-S302	262	106
			EPM-S303	259	103
			EPM-S304	261	105
			EPM-S305	263	107
			EPM-S306	258	102
			EPM-S308	265	109
			EPM-S309	264	108
			EPM-S310	3905	F41
			EPM-S400	1025	401
			EPM-S401	1028	404
			EPM-S402	1026	402
			EPM-S403	1029	405
			EPM-S404	1030	406
			EPM-S405	1027	403
			EPM-S406	1036	40C
			EPM-S408	1035	40B
			EPM-S500	1281	501
			EPM-S501	1283	503
			EPM-S502	1282	502
			EPM-S503	1284	504
			EPM-S600	2241	8C1
			EPM-S601	2243	8C3
			EPM-S602	2242	8C2
			EPM-S603	2244	8C4
			EPM-S604	2497	9C1
			EPM-S620	2305	901
			EPM-S640	3585	E01
			EPM-S650	2625	A41

5.19.3 Status of the digital inputs

Use index I-6000 to have the status of the digital inputs displayed.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-6000		Digital input		0	{1}	255	Read only, digital input status
	1	Byte 1					
	2	Byte 2					
	...	...					
	80	Byte 80					

5.19.4 Status of the digital outputs

Use index I-6200 to have the status of the digital outputs displayed:

Index		Name	Possible settings				Info
			Lenze	Selection			
I-6200 *		Digital output		0	{1}	255	• Digital output status • The outputs can be set manually (forcing): • Depending on the CAN status and on I-2360
	1	Byte 1					
	2	Byte 2					
	...	...					
	80	Byte 80					

5.19.5 Status of the analog inputs

Use index I-6401 to have the status of the analog inputs displayed.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-6401		Analog input		-32768	{1}	32767	Read only, analog input status
	1	Channel 1					
	2	Channel 2					
	...	...					
	36	Channel 36					

5.19.6 Status of the analog outputs

Use index I-6411 to have the status of the analog outputs displayed:

Index		Name	Possible settings				Info
			Lenze	Selection			
I-6411 *		Analog output		-32768	{1}	32767	• Analog output status • The outputs can be set manually (forcing): • Depending on the CAN status and on I-2360
	1	Channel 1					
	2	Channel 2					
	...	...					
	36	Channel 36					

5.19.7 Counter status

Via the following indexes you can have the status of the counters displayed:

Index		Name	Possible settings				Info
			Lenze	Selection			
I-5400		Counter Value		0x00000000	{0x1}	0xFFFFFFFF	Counter value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: Subindex is increased by 2 for each counter
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5401		Latch value		0x00000000	{1}	0xFFFFFFFF	Latch value
	1	DWord 1					EPM-S600: Subindex is increased by 1 for each counter EPM-S601/-S602/-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5402		Status word		0x0000	{1}	0xFFFF	Status word
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: Subindex is increased by 2 for each counter
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5403		Counter ticker value		0x0000	{1}	0xFFFF	Ticker value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					

5.19.8 Status of the digital inputs with time stamp function

Use index I-5430 to display the status of the digital inputs with time stamp function.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-5430		DI time stamp state		0x0000	{1}	0xFFFF	Read only, time stamp status 15 entries per module Bit 0: channel status DI1 (0: FALSE; 1: TRUE) Bit 1: channel status DI2 (0: FALSE; 1: TRUE) Bit 2 ... 7: Reserved Bit 8 ... 15: Counter which counts from 0 ... 127 and restarts at 0. Bits 16 ... 32: ticker value
	1	DWORD 1					
	2	DWORD 2					
	...	...					
	80	DWORD 64					

5.19.9 Status of the digital outputs with time stamp function

The following index can be used to display the status of the digital outputs with time stamp function.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-5440		DO time stamp state		0x0000	{1}	0xFFFF	Read only, FIFO memory status 1 entry per module Bits 0 ... 5: Running number of time stamp entry last written to the FIFO memory. Bits 6 ... 7: reserved Bits 8 ... 13: Running number of time stamp entry to be processed next. Bits 14 ... 15: reserved Bits 16 ... 23: • 0x00 or 0x80: OK • 0x01 or 0x81: No following entry available • 0x02 or 0x82: No new entries available • 0x03 or 0x83: FIFIO memory full. No new entries possible. A full memory will not accept any more time stamp entries. Perform a status query to establish the FIFO memory's status before transferring more time stamp entries. Bit 24 ... 31: Number of the time stamp entries in the FIFO memory.
	1	DWORD 1					
	2	DWORD 2					
	...	...					
	80	DWORD 64					

You can specify values using the following index.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-5640		DO time stamp control		0x0000	{1}	0xFFFF	15 entries per module Bit 0 ... 5: Reserved Bit 6: channel status D01 (0: FALSE; 1: TRUE) Bit 7: channel status D02 (0: FALSE; 1: TRUE) Bit 8 ... 15: Counter which counts from 0 ... 127 and restarts at 0. Bits 16 ... 32: ticker value
	1	DWORD 1					
	2	DWORD 2					
	...	...					
	80	DWORD 64					

5.19.10 Status of the digital outputs with PWM function

You can use the following indexes to display the status of the digital outputs with PWM function.

Index		Name	Possible settings				Info
			Lenze	Selection			
I-5420		PWM state		0x0000	{1}	0xFFFF	Read only, PWM status 2 entries per module Bit 0: Reserved Bit 1: PWM status (0: PWM output stopped; 1: PWM output active) Bit 2: output status (0: push/pull output; 1: highside output) Bits 3 ... 15: reserved
	1	WORD 1					
	2	WORD 2					
	...	...					
	80	WORD 64					
I-5620		PWM pulse duration		0x0000	{1}	0xFFFF	Status PWM 2 entries per module Specification of pulse duration in [μs].
	1	DWORD 1					
	2	DWORD 2					
	...	...					
	80	DWORD 64					
I-5621		PMW control		0x0000	{1}	0xFFFF	Control word PWM 2 entries per module Bits 0 ... 1: reserved Bit 2: • 0: Push/Pull output The output signal is switched active on HIGH level and on LOW level. • 1: Highside output The output signal is only switched active on HIGH level. Bits 3 ... 7: reserved Bit 8: 0-1 edge: PWM output starts Bit 9: 1-0 edge: PWM output stops Bits 10 ... 15: reserved
	1	WORD 1					
	2	WORD 2					
	...	...					
	80	WORD 64					

5.20 Index table

- The indices are numbered in ascending order for reference purposes.
- How to read the index table:

Column	Abbreviation				Meaning
Index	I-xxxx				Index I-xxxx
		1			Subindex 1 of I-xxxx
		2			Subindex 2 of I-xxxx
	I-xxxx ↲				After entry, the index parameter value is stored in the EEPROM
	I-xxxx *				Parameter value of the index is stored in the EEPROM with I-2003 = 1
Name					Index name
Lenze					Lenze setting, setting on delivery
Selection	1		{%}	99	Min. value {unit} max. value
Info					Short, important explanation
	Page x				Reference to detailed explanations

Index		Name	Possible settings: Lenze	Selection					Info
I-1000		Device Type							Read only, device type
I-1001		Error register							Read only
				I-1001	Bit 0: Generic An unspecified error has occurred (flag set on each error message)				
					Bit 1: Reserved				
					Bit 2: Reserved				
					Bit 3: Reserved				
					Bit 4: Comm. Communication error (Overrun CAN)				
					Bit 5: Reserved				
					Bit 6: Reserved				
					Bit 7: ManSpec. Manufacturer-specific error				Shown in detail in I-1003
I-1003									Read only, fault memory
	1	Actual errors							
I-1004		Number of supported PDOs							Read only
	1	Number of synchronous PDOs supported							
	2	Number of synchronous PDOs supported							
I-1005 *		Sync COB-ID	128	128		{1}		2047	

Index	Name	Possible settings				Info
		Lenze	Selection			
I-1006 *	Sync interval (µs)	0	0	{1 µs}	4294967295	The I/O system acts as sync consumer: • In I-1005, bit 30 must be set = 0 • After the time set in I-1006 has elapsed, the I/O system changes to the communication status set in I-1029 • A reset will be carried out with the next sync telegram • If I-1006 = 0, monitoring is switched off
I-1007	Synchronous window length	0	0	{1 µs}	4294967295	The length of the time slot for synchronous PDOs in µs.
I-1008	DIS: Device name					Read only, device name
I-1009	DIS: Hardware version					Read only, hardware version
I-100A	DIS: Software version					Read only, software version
I-100B	Node ID		1	{1}	127	Read only, CANopen node address
I-100C *	Guard time	0	0	{1 ms}	65535	Node guarding Monitoring time 0 = monitoring not active
I-100D *	Life time factor	0	0	{1}	255	Node guarding Response time computation factor 0 = monitoring not active The response time is computed as: Monitoring time × factor
I-100E	Node Guarding identifier					Read only Identifier = basic identifier + node address (basic identifier cannot be modified)
I-1010 ↲	Save All Parameters	0				Saving the parameter settings and the station structure in the EEPPROM of the bus coupler. Function in accordance with CANopen (DS301/DS401 communication protocol).
0			Number of subindexes assigned			Read only Number of the subindices used by the index I-1010
1	Save		0 = no function 1702257011 = save parameters			The numerical value is ASCII-coded and complies with: 65 76 61 0x73 = "E" "V" "A" "S"
I-1011 ↲	Restore All Parameters	0				Resetting the parameter setting in the EEPPROM of the bus coupler to the Lenze setting. Function in accordance with CANopen (DS301/DS401 communication protocol).
0			Number of subindexes assigned			Read only Number of the subindices used by the index I-1011

Index		Name	Possible settings: Lenze	Possible settings: Selection	Info
	1	Load		0 = no function 1684107116 = load Lenze setting	The numerical value is ASCII-coded and complies with: 64 61 6F 0x6C = "D" "A" "O" "L" After executing "Load", acceptance is effected by • Voltage off/on, or • Transmission of a reset node telegram (00 82 xx, with xx = node address)
I-1014		COB ID emergency			Emergency telegram After the boot-up, the identifier 0x80 + node address are displayed.
I-1017 ↲		Heartbeat producer time	0	0 … {1 ms} … 65535 I-1017: 0 Function is not active	I/O system can be monitored by other nodes. Within this time the device status of I/O system is transmitted to the fieldbus.
I-1018					Read only, device identification
	1	Vendor ID			
	2	Product Code			
	3	Revision number			
I-1027		Type of			Read only
	0				Number of plugged-in modules
	1	Module no. 1			Module list Subindexes 1 ... 64 Module identifiers of the plugged-in modules
	2	Module no. 2			
	...	...			
	64	Module no. 64			
I-1029 *		Error behaviour		0 Pre-Operational 1 No state changed 2 Stopped 3 Reset	Error behaviour
	1	Communication error	0		I/O system switches to the status set if the communication with the master fails or "node guarding", "heartbeat", or the output monitoring have been activated.
	2	Manufacturer-specific error	0		Without function
I-1200		Server SDO parameter 1			Read only, current identifier for SDO communication
	1	SDO1-Rx		1536 (basic identifier) + node address	
	2	SDO1-Tx		1408 (basic identifier) + node address	
I-1201		Server SDO parameter 2			Current identifiers for SDO communication SDO channel deactivated: bit 31 = 1 SDO channel activated: bit 31 = 0 + identifier from the SDO area
	1	SDO2-Rx	0	COB-ID client -> server (Rx)	
	2	SDO2-Tx	0	COB-ID server -> client (Tx)	
I-1400 ↲					

Index		Name	Possible settings: Lenze	Selection					Info
	1	COB-ID used by RxPDO 1	513 + NID	385		{1}		2047	Definition of specific identifiers for process data object 1 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission				The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved				
				255	Process data update on every occurrence of an event				Every received value is accepted
I-1401 ↲									
	1	COB-ID used by RxPDO 2	768 + NID	385		{1}		2047	Definition of specific identifiers for process data object 2 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission				The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved				
				255	Process data update on every occurrence of an event				Every received value is accepted
I-1402 ↲									
	1	COB-ID used by RxPDO 3	1024+ NID	385		{1}		2047	Definition of specific identifiers for process data object 3 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission				The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved				
				255	Process data update on every occurrence of an event				Every received value is accepted
I-1403 ↲									
	1	COB-ID used by RxPDO 4	1280+ NID	385		{1}		2047	Definition of specific identifiers for process data object 4 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission				The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved				
				255	Process data update on every occurrence of an event				Every received value is accepted
I-1404 ↲									

Index		Name	Possible settings						Info
			Lenze	Selection					
	1	COB-ID used by RxPDO 5	1920+ NID	385		{1}		2047	Definition of specific identifiers for process data object 5 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission				The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved				
				255	Process data update on every occurrence of an event				Every received value is accepted
I-1405 ↲									
	1	COB-ID used by RxPDO 6	576 + NID	385		{1}		2047	Definition of specific identifiers for process data object 6 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission				The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved				
				255	Process data update on every occurrence of an event				Every received value is accepted
I-1406 ↲									
	1	COB-ID used by RxPDO 7	832 + NID	385		{1}		2047	Definition of specific identifiers for process data object 7 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission				The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved				
				255	Process data update on every occurrence of an event				Every received value is accepted
I-1407 ↲		I-1408							
	1	COB-ID used by RxPDO 8	1088+ NID	385		{1}		2047	Definition of specific identifiers for process data object 8 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission				The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved				
				255	Process data update on every occurrence of an event				Every received value is accepted
I-1408 ↲									

Index		Name	Possible settings				Info
			Lenze	Selection			
	1	COB-ID used by RxPDO 9	1344+ NID	385	{1}	2047	Definition of specific identifiers for process data object 9 (NID= Node ID / node address)
	2	Transmission type	255	0	{1}	255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission		The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved		
				255	Process data update on every occurrence of an event		Every received value is accepted
I-1409 ↲							
	1	COB-ID used by RxPDO 10	1984+ NID	385	{1}	2047	Definition of specific identifiers for process data object 10 (NID= Node ID / node address)
	2	Transmission type	255	0	{1}	255	Defining the transmission mode
				0 ... 240	Process data update on every sync telegram transmission		The input data are accepted on sync telegram transmission.
				241 ... 254	Reserved		
				255	Process data update on every occurrence of an event		Every received value is accepted
I-1600 ↲							Mapping parameters for receive PDOs
	0	Number of mapped RxPDO1		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1601 ↲							Mapping parameters for receive PDOs
	0	Number of mapped RxPDO2		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1602 ↲							Mapping parameters for receive PDOs

Index		Name	Possible settings				Info
			Lenze	Selection			
	0	Number of mapped RxPDO3		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1603 ↲							Mapping parameters for receive PDOs
	0	Number of mapped RxPDO4		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1604 ↲							Mapping parameters for receive PDOs
	0	Number of mapped RxPDO5		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1605 ↲							Mapping parameters for receive PDOs
	0	Number of mapped RxPDO6		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value

Index		Name	Possible settings				Info
			Lenze	Selection			
I-1606 ↲							Mapping parameters for receive PDOs
	0	Number of mapped RxPDO7		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1607 ↲							Mapping parameters for receive PDOs
	0	Number of mapped RxPDO8		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1608 ↲							Mapping parameters for receive PDOs
	0	Number of mapped RxPDO9		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1609 ↲							Mapping parameters for receive PDOs

Index		Name	Possible settings						Info
			Lenze	Selection					
	0	Number of mapped RxPDO10		0		{1}		255	8 bit value
	1	1st mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	2	2nd mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	3	3rd mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	4	4th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	5	5th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	6	6th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	7	7th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	8	8th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
I-1800 ↲									
	1	COB-ID used by TxPDO 1	384+N ID	385		{1}		2047	Definition of specific identifiers for process data object 1 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	100	0		{1 ms}		65535	Cycle time
I-1801 ↲									
	1	COB-ID used by TxPDO 2	640 + NID	385		{1}		2047	Definition of specific identifiers for process data object 2 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	0	0		{1 ms}		65535	Cycle time
I-1802 ↲									

Index		Name	Possible settings						Info
			Lenze	Selection					
	1	COB-ID used by TxPDO 3	896 + NID	385		{1}		2047	Definition of specific identifiers for process data object 3 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	0	0		{1 ms}		65535	Cycle time
I-1803 ↲									
	1	COB-ID used by TxPDO 4	1152+ NID	385		{1}		2047	Definition of specific identifiers for process data object 4 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	0	0		{1 ms}		65535	Cycle time
I-1804 ↲									

Index		Name	Possible settings						Info
			Lenze	Selection					
	1	COB-ID used by TxPDO 5	1664+ NID	385		{1}		2047	Definition of specific identifiers for process data object 5 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	0	0		{1 ms}		65535	Cycle time
I-1805 ↲									
	1	COB-ID used by TxPDO 6	448 + NID	385		{1}		2047	Definition of specific identifiers for process data object 6 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	0	0		{1 ms}		65535	Cycle time
I-1806 ↲									

Index		Name	Possible settings						Info
			Lenze	Selection					
	1	COB-ID used by TxPDO 7	704 + NID	385		{1}		2047	Definition of specific identifiers for process data object 7 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	0	0		{1 ms}		65535	Cycle time
I-1807 ↲									
	1	COB-ID used by TxPDO 8	960 + NID	385		{1}		2047	Definition of specific identifiers for process data object 8 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	0	0		{1 ms}		65535	Cycle time
I-1808 ↲									

Index		Name	Possible settings						Info
			Lenze	Selection					
	1	COB-ID used by TxPDO 9	1216+ NID	385		{1}		2047	Definition of specific identifiers for process data object 9 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	0	0		{1 ms}		65535	Cycle time
I-1809 ↲									
	1	COB-ID used by TxPDO 10	1728+ NID	385		{1}		2047	Definition of specific identifiers for process data object 10 (NID= Node ID / node address)
	2	Transmission type	255	0		{1}		255	Defining the transmission mode
				0	Function deactivated				The output data are accepted on sync telegram transmission.
				1 ... 240	Process data transfer after sync no. 1 ... Process data transfer according to sync no. 240				The output data is accepted after the set number of sync telegrams (1 ... 240) has been transmitted
				254	Time-controlled process data transfer				Only if a cycle time is set in I-180x, subindex 5
				255	Event-controlled process data transfer				
				255	Event-controlled process data transfer with cyclic overlay				Only if a cycle time is set in I-180x, subindex 5
	3	Inhibit time	0	0		{1 ms}		65535	Blocking time
	5	Event time	0	0		{1 ms}		65535	Cycle time
I-1A00 ↲									Mapping parameters for transmit PDOs
	0	Number of mapped TxPDO1		0		{1}		255	8 bit value
	1	1st mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	2	2nd mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	3	3rd mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	4	4th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	5	5th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	6	6th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	7	7th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
	8	8th mapped object		0x00000000		{1}	0xFFFFFFFF		32 bit value
I-1A01 ↲									Mapping parameters for transmit PDOs

Index		Name	Possible settings				Info
			Lenze	Selection			
	0	Number of mapped TxPDO2		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1A02 ↲							Mapping parameters for transmit PDOs
	0	Number of mapped TxPDO3		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1A03 ↲							Mapping parameters for transmit PDOs
	0	Number of mapped TxPDO4		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1A04 ↲							Mapping parameters for transmit PDOs
	0	Number of mapped TxPDO5		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value

Index		Name	Possible settings				Info
			Lenze	Selection			
I-1A05 ↲							Mapping parameters for transmit PDOs
	0	Number of mapped TxPDO6		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1A06 ↲							Mapping parameters for transmit PDOs
	0	Number of mapped TxPDO7		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1A07 ↲							Mapping parameters for transmit PDOs
	0	Number of mapped TxPDO8		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1A08 ↲							Mapping parameters for transmit PDOs

Index		Name	Possible settings				Info
			Lenze	Selection			
	0	Number of mapped TxPDO9		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-1A09 ↲							Mapping parameters for transmit PDOs
	0	Number of mapped TxPDO10		0	{1}	255	8 bit value
	1	1st mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	2	2nd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	3	3rd mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	4	4th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	5	5th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	6	6th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	7	7th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
	8	8th mapped object		0x00000000	{1}	0xFFFFFFFF	32 bit value
I-2100		PDO length	0	0	{1}	255	Setting of the user data width Note: Lenze inverters expect a PDO length of eight bytes even though not all of them may have assigned I/O values.
				I-2100	0 PDO length 8 bytes		
					1 PDO length according to process image		
I-2101		Identifier assignment		0	CANopen (assignment in accordance with DS301)		
			I-2101		1 Assignment according to the Lenze system bus		
I-2359		Bus state		0 1	Operational Pre-Operational		Current bus state
I-2361		Mode		0 1	CANopen Lenze system bus		
I-2400 ↲		Timer value		0	{1 ms}	65535	Monitoring time for process data input objects

Index		Name	Lenze	Selection			Info
	1	Lenze-PDO Control 1	0				
	2	Lenze-PDO Control 2	0				
	3	Lenze-PDO Control 3	0				
	4	Lenze-PDO Control 4	0				
	5	Lenze-PDO Control 5	0				
	6	Lenze-PDO Control 6	0				
	7	Lenze-PDO Control 7	0				
	8	Lenze-PDO Control 8	0				
	9	Lenze-PDO Control 9	0				
	10	Lenze PDO control 10	0				
I-3100 ... I-311D		Parameter data byte		0	{0x1}	255	Contents are written by the parameterisable module
	1	Module 1					
	2	Module 2					
	...	...					
	64	Module 64					
I-31FE ↲		Accept Module Parameter (single)	0	0	{1}	255	Index for accepting parameter settings one module at a time. Is used, for instance, in the »PLC Designer«
	1	DWord 1		255	Accept module parameter		Each subindex complies with a parameterisable module (sequence according to the slot, from left to right). The value changes are only accepted after writing the corresponding subindex with 255
	2	DWord 2					
	...	...					
	64	DWord 64					
I-31FF		Accept module parameter	0	0	{1}	255	After describing the index with 255, the parameter changes are transferred by all modules
				I-31FF	0		
					255 Accept module parameter		
I-5400 ↲		Counter Value		0x00000000	{0x1}	0xFFFFFFFF	Counter value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: Subindex is increased by 2 for each counter
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5401 ↲		Latch value		0x00000000	{1}	0xFFFFFFFF	Latch value

Index		Name	Possible settings				Info
			Lenze	Selection			
	1	DWord 1					EPM-S600: Subindex is increased by 1 for each counter EPM-S601/-S602/-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5402 ↲		Status word		0x0000	{1}	0xFFFF	Status word
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: Subindex is increased by 2 for each counter
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5403 ↲		Counter ticker value		0x0000	{1}	0xFFFF	Ticker value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5420		PWM state		0x0000	{1}	0xFFFF	Read only, PWM status 2 entries per module Bit 0: Reserved Bit 1: PWM status (0: PWM output stopped; 1: PWM output active) Bit 2: output status (0: push/pull output; 1: highside output) Bits 3 ... 15: reserved
	1	WORD 1					
	2	WORD 2					
	...	...					
	80	WORD 64					
I-5430		DI time stamp state		0x0000	{1}	0xFFFF	Read only, time stamp status 15 entries per module Bit 0: channel status DI1 (0: FALSE; 1: TRUE) Bit 1: channel status DI2 (0: FALSE; 1: TRUE) Bit 2 ... 7: Reserved Bit 8 ... 15: Counter which counts from 0 ... 127 and restarts at 0. Bits 16 ... 32: ticker value
	1	DWORD 1					
	2	DWORD 2					
	...	...					
	80	DWORD 64					

Index		Name	Possible settings				Info
			Lenze	Selection			
I-5440		DO time stamp state		0x0000	{1}	0xFFFF	Read only, FIFO memory status 1 entry per module Bits 0 ... 5: Running number of time stamp entry last written to the FIFO memory. Bits 6 ... 7: reserved Bits 8 ... 13: Running number of time stamp entry to be processed next. Bits 14 ... 15: reserved Bits 16 ... 23: • 0x00 or 0x80: OK • 0x01 or 0x81: No following entry available • 0x02 or 0x82: No new entries available • 0x03 or 0x83: FIFIO memory full. No new entries possible. A full memory will not accept any more time stamp entries. Perform a status query to establish the FIFO memory's status before transferring more time stamp entries. Bit 24 ... 31: Number of the time stamp entries in the FIFO memory.
	1	DWORD 1					
	2	DWORD 2					
	...	...					
	80	DWORD 64					
I-5600 ↲		Counter Compare Value		0x00000000	{1}	0xFFFFFFFF	Comparison value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601: Subindex is increased by 2 for each counter EPM-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5601 ↲		Counter Set Value		0x00000000	{1}	0xFFFFFFFF	Set value
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: without function
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5602 ↲		Counter Control Word		0x0000	{1}	0xFFFF	Control word
	1	DWord 1					EPM-S600/-S602: Subindex is increased by 1 for each counter EPM-S601/-S603: Subindex is increased by 2 for each counter
	2	DWord 2					
	...	...					
	64	DWord 64					
I-5620		PWM pulse duration		0x0000	{1}	0xFFFF	Status PWM 2 entries per module Specification of pulse duration in [μs].
	1	DWORD 1					
	2	DWORD 2					
	...	...					
	80	DWORD 64					

Index		Name	Possible settings				Info
			Lenze	Selection			
I-5621		PMW control		0x0000	{1}	0xFFFF	Control word PWM 2 entries per module Bits 0 ... 1: reserved Bit 2: • 0: Push/Pull output The output signal is switched active on HIGH level and on LOW level. • 1: Highside output The output signal is only switched active on HIGH level. Bits 3 ... 7: reserved Bit 8: 0-1 edge: PWM output starts Bit 9: 1-0 edge: PWM output stops Bits 10 ... 15: reserved
	1	WORD 1					
	2	WORD 2					
	...	...					
	80	WORD 64					
I-5640		DO time stamp control		0x0000	{1}	0xFFFF	15 entries per module Bit 0 ... 5: Reserved Bit 6: channel status D01 (0: FALSE; 1: TRUE) Bit 7: channel status D02 (0: FALSE; 1: TRUE) Bit 8 ... 15: Counter which counts from 0 ... 127 and restarts at 0. Bits 16 ... 32: ticker value
	1	DWORD 1					
	2	DWORD 2					
	...	...					
	80	DWORD 64					
I-6000		Digital input		0	{1}	255	Read only, digital input status
	1	Byte 1					
	2	Byte 2					
	...	...					
	80	Byte 80					
I-6002 ↲		Change polarity - digital input		0	{1}	255	Inverts digital input signals
	1	Byte 1	0				
	2	Byte 2	0				
	...	...	...				
	80	Byte 80	0				
I-6200 *		Digital output		0	{1}	255	• Digital output status • The outputs can be set manually (forcing): • Depending on the CAN status and on I-2360
•	1	Byte 1					
	2	Byte 2					
	...	...					
	80	Byte 80					
I-6202 ↲		Change polarity - digital output		0	{1}	255	Inverts digital output signals
	1	Byte 1	0				
	2	Byte 2	0				
	...	...	...				
	80	Byte 80	0				

Index		Name	Lenze	Selection			Info
I-6206 ↲		Error mode - digital output		0	{1}	255	Configures digital output monitoring
				I-6206	0 All digital outputs retain the last status output.		
					255 Response from I-6207		In I-6207, the response can be configured individually for each digital output
	1	Byte 1	0				
	2	Byte 2	0				
	...	...	...				
	80	Byte 80	0				
I-6207 ↲		Error value - digital output	0	0	{1}	255	Configures the individual digital output responses
				I-6207 8 bits of information			
				Bit value 0	Output switches to LOW		
				Bit value 1	Output retains last status output		
	1	Byte 1	0				
	2	Byte 2	0				
	...	...	...				
	64	Byte 80	0				
I-6401		Analog input		-32768	{1}	32767	Read only, analog input status
	1	Channel 1					
	2	Channel 2					
	...	...					
	36	Channel 36					
I-6411 *		Analog output		-32768	{1}	32767	• Analog output status • The outputs can be set manually (forcing): • Depending on the CAN status and on I-2360
•	1	Channel 1					
	2	Channel 2					
	...	...					
	36	Channel 36					
I-6421 ↲		Trigger selection		0	{1}	255	Enables interrupt for analog inputs/outputs
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				

I-6423 ↲		Global interrupt enable		0	{1}	255	Global activation/deactivation of the event-controlled process data transfer of the analog input signals.
							The setting in I-6423 has a higher priority than the settings in the TxPDOs.
							• Lenze setting:
							  • System bus (CAN): I-6423 = 255
							  • CANopen: I-6423 = 0

				I-6423	0 Event-controlled process data transfer deactivated		
					255 Event-controlled process data transfer activated		

Index		Name	Possible settings				Info
			Lenze	Selection			
I-6424 ↲		Upper limit analog input		0x00000000	{1}	0xFFFFFFFF	
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				
I-6425 ↲		Lower limit analog input		0x00000000	{1}	0xFFFFFFFF	
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				
I-6426 ↲		Delta limit analog input		0x00000000	{1}	0xFFFFFFFF	
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				
I-6443 ↲		Error mode analog output		0	{1}	255	Configures analog output monitoring
				I-6443	0 All analog outputs retain the last value output		
					255 Response from I-6444		In I-6444, the response can be configured individually for each analog output
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				
I-6444 ↲		Error value analog output		-32768	{1}	32767	Configures the individual analog output responses The analog outputs provide the set value
	1	Channel 1	0				
	2	Channel 2	0				
	...	...	...				
	36	Channel 36	0				

6 PROFIBUS communication

6.1 PROFIBUS / PROFIBUS-DP

PROFIBUS is an integrated, open, digital communication system with a wide application range mainly in manufacturing and process automation. PROFIBUS is suitable for fast, time-critical and complex communication tasks.

PROFIBUS-DP can be used for manufacturing automation. It provides for an easy, fast, cyclic and deterministic process data exchange between a master and the assigned slaves. Power section DP-V0 is provided with these basic functions. Power section DP-V1 was enhanced by an acyclic data exchange between master and slave.

Power section DP-V0

DP-V0 (Decentralised Peripherals) provides the basic functions of DP.

Power section DP-V1

The power section DP-V1 contains supplements for DP-V0 with the focus on process automation: Simultaneously to the cyclic process data transfer, an acyclic data link to the slaves is built up in order to parameterise the slaves.

Power section DP-V1 can only be used if it is supported by the master and the slaves.

6.2 System configuration

6.2.1 Device types

PROFIBUS differentiates between active nodes (master) and passive nodes (slave).

Class 1 master (DPM 1)

A class 1 master (DPM 1) is a central control which exchanges data with the slaves in a fixed cycle. Typical DPM 1 are, for example, PLC or PC. Via an active bus access, measured data is read cyclically from the input modules of the slaves and setpoints are written to the output modules of the slaves.

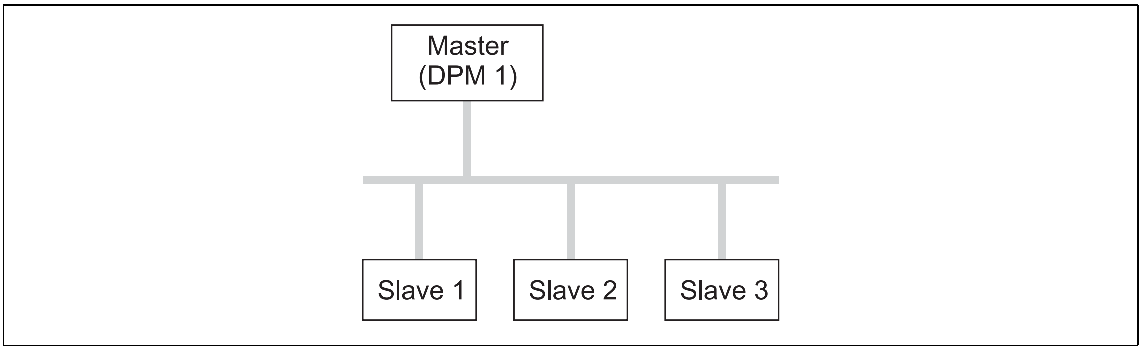
Class 2 master (DPM 2)

Class 2 masters (DPM 2) are used for engineering, configuration or operation. During commissioning, maintenance and diagnostics, for example, DPM 2 can be used to configure the connected slaves, evaluate measured values and parameters and query the status of the slaves. The data is transmitted acyclically. DPM 2 do not have to be permanently connected to the bus. DPM 2 are provided with active bus access.

Slave

Slaves are peripherals (PROFIBUS bus couplers) making process information (input data and output data) available. Slaves only respond to direct requests by the master.

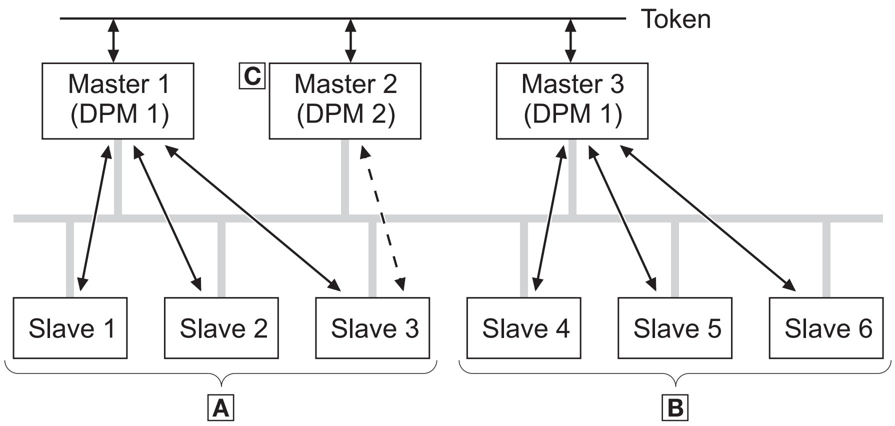
6.2.2 Mono master system



[6-1] PROFIBUS-DP Mono master system

In case of mono master systems, only one master is active on the bus during the operating phase of the bus system. The slaves are connected in a decentralised way to the master via the transmission medium. This system configuration serves to achieve the shortest bus cycle time.

6.2.3 Multi-master system



A Subsystem from master 1 and slaves 1 ... 3 with cyclic data transmission
B Subsystem from master 3 and slaves 4 ... 6 with cyclic data transmission.
C For the purpose of project planning and diagnostics, the master 2 can communicate with slaves 1 ... 6. The data is transferred acyclically.

[6-2] PROFIBUS-DP multi-master system

In the multi-master operation, several masters are connected to one bus. They either form independent subsystems consisting of one master of class 1 (DPM 1) each and the respective slaves or additional masters of class 2 (DPM 2) for project planning and diagnostics. The input images and output images of the slaves can be read by all masters. Writing the outputs can only be done by the corresponding master of class 1 (DPM 1).

6.3 Communication

6.3.1 Bus access

The transmission protocol offers two bus access procedures.

Master ↔ Master

The master communication is also referred to as token passing procedure. The token passing procedure makes sure that the bus access authorisation is assigned. The bus access authorisation is given by means of a "token". The token is a special telegram transmitted via the bus.

If a master has a token, it can communicate with all of the other bus nodes. The token hold time is defined during system configuration. Once the token hold time has elapsed, the token is passed on to the next master which is then in possession of the bus access authorisation and can communicate with all of the other nodes.

The data transfer between the master and the slaves assigned to it is automatically controlled by the master and takes place in a fixed and recurring sequence. The slaves are assigned to a master during configuration. In addition, it can be defined which slaves participate in the cyclic process data transfer.

Master ↔ Slave

Before master and slave can communicate, the configuration and the parameter setting are checked for errors after startup.

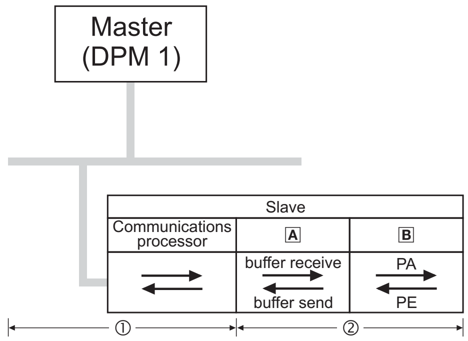
The following are checked: type, format information, length information and the number of inputs and outputs.

If the parameters are valid, the slave changes over to the DataExchange (DE) state. The master can now transmit output data to the slave and receive the current input data from the slave.

While the process data transfer is in progress, the master can transmit new parameter data to the slave.

6.3.2 Cyclic data transmission

The data communication with PROFIBUS-DP-V0 includes cyclic diagnostics as well as cyclic process data and parameter data transfer.



A Backplane bus with transmit and receive buffer
B Input / output modules
- PA: Process image of the outputs,
- PE: Process image of the inputs

C Status C
① PROFIBUS cycle
② Backplane bus cycle

[6-3] DP cycle and cycle of the backplane bus

Backplane bus cycle

In a backplane bus cycle,

- the input data (PE) are collected at the inputs and transmitted to the transmit buffer (buffer send),
- the output data (PA) is written into the outputs by the receive buffer (buffer receive).

PROFIBUS cycle

In a PROFIBUS cycle, the master addresses all assigned slaves sequentially with a DataExchange. In case of DataExchange, the memory areas assigned to the PROFIBUS are transmitted.

- The data of the PROFIBUS input range is transmitted to the receive buffer (buffer receive).
- The data of the transmit buffer (buffer send) is transmitted to the PROFIBUS output range.

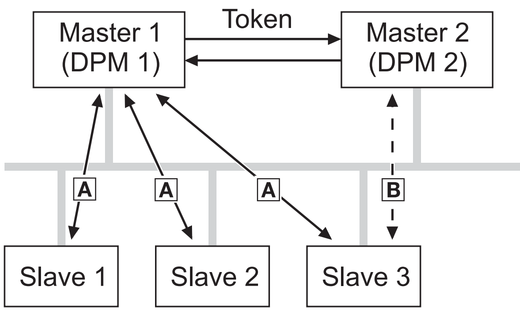
6.3.3 Acyclic data transfer

An optional service expansion is the acyclic parameter data transfer of PROFIBUS-DP-V1. PROFIBUS-DP-V0 and PROFIBUS-DP-V1 can be operated simultaneously in a network.

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

- With configuration, a time slot is reserved.
- Without configuration the acyclic service is appended when a DPM 2 acyclically accesses a DP-V1 slave.
- The acyclic service is always carried out with a lower priority.

Parameter data transfer between DPM 1 and slaves



A Cyclic process data transfer between DPM 1 and slave 1 ... 3
B Acyclic parameter data transfer between DPM 2 and slave 3

[6-4] Acyclic data transfer

1. DPM 1 has the transmission authorisation (token) and communicates with the slave 1, then with slave 2 etc., until the last slave of the current list via the MS0 channel in a fixed sequence.
2. DPM 1 transfers the token to DPM 2.
3. In the remaining cycle time (time slot), DPM 2 establishes an acyclic connection to any slave to transfer parameter data via the MS2 channel.
4. At the end of the running cycle time, DPM 2 returns the token to DPM 1.
   Depending on the remaining cycle time, several time slots may be required for the non-cyclical transmission of the data records.
5. When all data records have been transferred, DPM 2 cancels the connection within one time slot.

Note!

DPM 1 can exchange parameter data acyclically with the slaves via the MS1 channel.

6.3.3.1 Services for the acyclic parameter data transfer

Data transfer between DPM 1 and slaves

The connection is established by DPM 1 via the MS1 channel. The connection to the slave can only be established by the master that has parameterised and configured the slave.

Service	Description
Read	The master reads a data block from the slave.
Write	The master writes a data block to the slave.
Alarm	The slave transmits an alarm message to the master. The master acknowledges receipt. To prevent alarm messages from being overwritten, the slave can only transmit a new alarm message if it has received the acknowledgement.
Alarm_Acknowledge	The master transmits an acknowledgement to the slave, confirming that it has received an alarm message.
Status	The slave transmits a status message to the master. The master does not acknowledge receipt.

Data transfer between DPM 2 and slaves

The connection is established by DPM 2 via the MS1 channel using the "Initiate" service. One slave can maintain several active connections at the same time. The number of connections is limited depending on the resources available in the slave.

Service	Description
Initiate / Abort	Establishing or terminating a connection for the acyclic data transfer between DPM 2 and a slave.
Read	The master reads a data block from the slave.
Write	The master writes a data block to the slave.
Data_Transport	The master writes user-specific data (defined in profiles) to the slave and, if required, it reads data from the slave in the same cycle.

Note!

For further information on the services and communication with DP-V0 and DP-V1, refer to standard IEC 61158.

6.3.4 Communication medium

- The communication medium is an RS485 interface.
- The bus can be configured as line or tree topology.
- The bus structure under RS485 enables the reactionless connection and disconnection of stations as well as the gradual commissioning of the system. Subsequent enhancements do not affect the stations already in operation. It is automatically detected whether a node has failed or just been connected to the mains.
- The PROFIBUS bus coupler is provided with a 9-pole Sub-D socket for connecting them to the bus.

6.4 Project planning

The I/O system is configured via the master. The following work steps must be carried out:

- Import the GSE file (device description) of the I/O system into the project planning tool.
  - PROFIBUS bus coupler: LENZ0C3A.gse
  - The description or linkage of the GSE file can be found in the project planning tool.
- Address nodes
  - Every node at the PROFIBUS is identified by an address.
  - Each address may only be assigned once in a bus system.
  - Addresses between 1 ... 125 can be assigned.
  - At the PROFIBUS bus coupler (slave), the address with the front DIP switch is set
  - At the master, the address is set during the configuration.
- Setting the baud rate
  - The baud rate is set in the configuration tool.
  - Adapt the baud rate to the length of the bus cable.
- Parameterise slaves

Note!

The diagnostic function is only supported by I/O compound modules from HW version 1B and by bus coupler modules from HW version 1A.

If a module in the system complies with an earlier HW version, the diagnostic function is deactivated for all modules.

6.5 Parameterising analog I/Os

6.5.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
128	0	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
129	0	Function channel 2		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x02))
6	Number of channels of a module (here 0x02)
7	Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10 ... 15	0 (fixed)

6.5.2 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
128	0	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
129	0	Function channel 2		0x20
130	0	Function channel 3		0x20
131	0	Function channel 4		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
12 ... 15	0 (fixed)

6.5.3 2 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S402

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set No.	Byte	Name	Description/value	Lenze
128	0	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
129	0	Function channel 2		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x02))
6	Number of channels of a module (here 0x02)
7	Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10 ... 15	0 (fixed)

6.5.4 4 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S403

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
128	0	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
129	0	Function channel 2		0x31
130	0	Function channel 3		0x31
131	0	Function channel 4		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
12 ... 15	0 (fixed)

6.5.5 2 analog inputs -10 ... +10 V (16 bits) - EPM-S406

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	1	Reserved	0	
	2	Limit value monitoring	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 (0 = deactivated; 1 = activated) Bits 7 ... 2: reserved	0x00
1	0	Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
128	0	Function channel 1	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
	1	Reserved	0	
	2	Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
	3	Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
129	0	Function channel 2	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
	1	Reserved	0	
	2	Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
	3	Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
-10 ... +10 V (0x12)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-5	-13824	CA00		
	-10	-27648	9400		
	-11.76	-32512	8100	Underflow	
-10 ... +10 V (0x22)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-5	-8192	E000		
	-10	-16384	C000		
	-12.5	-20480	B000	Underflow	
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-1.76	-4864	ED00	Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-2	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
12 ... 15	0 (fixed)

6.5.6 2 analog inputs 0/4 ... 20 mA (16 bits) - EPM-S408

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0		Diagnostics	Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
		Reserved	0	
		Limit value monitoring	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 (0 = deactivated; 1 = activated) Bits 7 ... 2: reserved	0x00
1		Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
128		Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
		Reserved	0	
		Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
		Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
129		Function channel 2	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
		Reserved	0	
		Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
		Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	**Current (I)** [mA]	**dec**	**hex**	**Range**	**Conversion**
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
12 ... 15	0 (fixed)

6.5.7 2 analog outputs 0 ... 10 V (12 bits) - EPM-S500

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	1	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
128	0	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
129	0	Function channel 2		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Output area (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Short circuit/overload (if parameterised)

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x73: Analog output Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x02))
6	Number of channels of a module (here 0x02)
7	Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
10 ... 15	0 (fixed)

6.5.8 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	1	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
128	0	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
129	0	Function channel 2		0x20
130	0	Function channel 3		0x20
131	0	Function channel 4		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Output area (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Short circuit/overload (if parameterised)

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x73: Analog output Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
12 ... 15	0 (fixed)

6.5.9 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	1	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
128	0	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
129	0	Function channel 2		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Open circuit (if parameterised)

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x73: Analog output Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x02))
6	Number of channels of a module (here 0x02)
7	Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
10 ... 15	0 (fixed)

6.5.10 4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	1	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
128	0	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
129	0	Function channel 2		0x31
130	0	Function channel 3		0x31
131	0	Function channel 4		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Open circuit (if parameterised)

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x73: Analog output Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
12 ... 15	0 (fixed)

6.6 Parameterising temperature measurement

6.6.1 4(2) analog input for resistance measurement - EPM-S404

Note!

Use parameter setting to deactivate unused inputs.

If thermal detectors are connected in a 3 or 4 conductor setup, channels 3 and/or 4 must be deactivated.

▶ 2-, 3-, 4-wire conductor measurement (📖 26)

The module does not provide any auxiliary supply for sensors.

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
	1	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) : Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
	2	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) : Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
	3	Reserved		
1	0	Temperature system	Bit 0, 1: 0b00 = °C; 0b01= °F; 0b10 = K Bit 2 ... 7: Reserved	0x00
	1	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02

Data set		Name	Description/value	Lenze
No.	Byte			
Channel 1				
128	0	Function channel 1	80 (0x50) ... 162 (0xA2): see "measuring range" 255 (0xFF): Channel deactivated	0x50
	1	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At 50 Hz: 44.1 ms/channel 16 bits; at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x00
	2, 3	Upper limit value channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. As soon as the measured value is outside a limit value and limit value monitoring is activated, a process alarm is triggered.	0x7FFF
	4,5	Lower limit value channel 1		0x8000
Channel 2				
129	0	Function channel 2	See channel 1	0x50
	1	Conversion time channel 2	See channel 1	0x00
	2, 3	Upper limit value channel 2	See channel 1	0x7FFF
	4, 5	Lower limit value channel 2 (HIGH byte)		0x8000
Channel 3 (for two-wire conductor connections only)				
130	0	Function channel 3	See channel 1	0x50
	1	Conversion time channel 3	See channel 1	0x00
	2, 3	Upper limit value channel 3	See channel 1	0x7FFF
	4, 5	Lower limit value channel 3		0x8000

Data set No.	Data set Byte	Name	Description/value	Lenze
Channel 4 (for two-wire conductor connections only)				
131	0	Function channel 4	See channel 1	0x50
	1	Conversion time channel 4	See channel 1	0x00
	2, 3	Upper limit value channel 4	See channel 1	0x7FFF
	4, 5	Lower limit value channel 4		0x8000

Measuring range

Measuring range (Fct. no.)	Measured value	Signal range	Range
2-wire: PT100 (0x50)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: PT1000 (0x51)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: NI100 (0x52)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: NI1000 (0x53)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: PT100 (0x58)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: PT1000 (0x59)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: NI100 (0x5A)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: NI1000 (0x5B)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: PT100 (0x60)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
4-wire: PT1000 (0x61)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
4-wire: NI100 (0x62)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: NI1000 (0x63)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: 0 ... 60 Ω (0x70)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x71)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0x72)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0x78)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x79)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x7A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0x80)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0x81)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0x82)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0x90)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x91)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0x92)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
3-wire: 0 ... 60 Ω (0x98)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x99)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x9A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xA0)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0xA1)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0xA2)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0xD0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0xD1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0xD2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0xD8)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0xD9)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0xDA)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xE0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0xE1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
4-wire: 0 ... 3000 Ω (0xE2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow

Diagnostics and alarm

Trigger	Process alarm	Diagnosealarm	parameterisable
Configuration/parameterisation errors	-	X	-
Open circuit detected	-	X	X
Measuring range exceeded	-	X	-
Measuring range not reached	-	X	-
Limit value exceeded	X	-	X
Limit value not reached	X	-	X
Process alarm lost	-	X	-

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: Limit value exceeded channel 1 Bit 1: Limit value exceeded channel 2 Bit 2: Limit value exceeded channel 3 Bit 3: Limit value exceeded channel 4 Bit 7 ... 4: 0 (fixed)
2	Bit 0: Limit value not reached channel 1 Bit 1: Limit value not reached channel 2 Bit 2: Limit value not reached, channel 3 Bit 3: Limit value not reached, channel 4 Bit 7 ... 4: 0 (fixed)
3/4	Ticker value at the time of the alarm After mains connection, a timer (µs ticker) is started, which after 65535 µs starts with 0 again.

Diagnosealarm

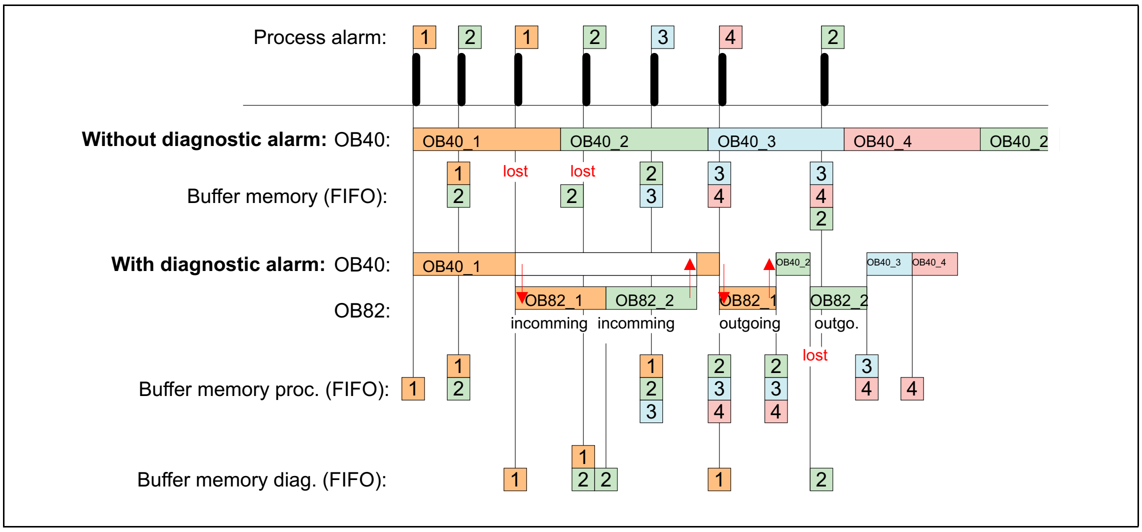
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x01)
7	Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed)
8	Diagnostic alarm due to process alarm lost to ... Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bit 7 ... 5: 0 (fixed)
9 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

6.6.2 2 analog inputs for thermocouple measurement - EPM-S405

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
	1	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
	2	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
	3	Reserved	0	
1	0	Temperature system	Bit 0, 1: 0b00 = °C; 0b10 = °F; 0b11 = K Bit 2 ... 7: Reserved	0x00
	1	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02
Channel 1				
128	0	Function channel 1	176 (0x60) ... 201 (0xC9): see "measuring range" External Temperature compensation: 176 (0x60) ... 185 (0x69) Internal temperature compensation: 192 (0xC0): ... 201 (0xC9) 255 (0xFF): Channel deactivated	0xC1
	1	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At Hz: 44.1 ms/channel 16 bits at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x02
	2, 3	Upper limit value channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. If the measured value is outside a limit value and the limit value monitoring is activated, a process alarm is triggered.	0x7FFF
	4, 5	Lower limit value channel 1		0x8000

Data set		Name	Description/value	Lenze
No.	Byte			
Channel 2				
129	0	Function channel 2	See channel 1	0xC1
	1	Conversion time channel 2	See channel 1	0x02
	2, 3	Upper limit value channel 2	See channel 1	0x7FFF
	3, 4	Lower limit value channel 2		0x8000

Measuring range

Measuring range (Fct. no.)	Measured value [°C]	[°F]	[K]	Range
Type J: -210 ... +1200 °C -346 ... 2192 °F 63.2 ... 1473.2 K (0xB0: ext. comp. 0 °C) (0xC0: int. comp. 0 °C)	+14500	26420	17232	Overflow
	-2100 ... +12000	-3460 ... +21920	632 ... 14732	Nominal range
	-	-	-	Underflow
Type K: -210 ... +1372 °C -454 ... 2501.6 °F 0 ... 1645.2 K (0xB1: ext. comp. 0 °C) (0xC1: int. comp. 0 °C)	+16220	29516	18952	Overflow
	-2700 ... +13720	-4540 ... 25016	0 ... 16452	Nominal range
	-	-	-	Underflow
Type N: -270 ... +1300 °C -454 ... 2372 °F 0 ... 1573.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+15500	28220	18232	Overflow
	-2700 ... +13000	-4540 ... 23720	0 ... 15732	Nominal range
	-	-	-	Underflow
Type R: -50 ... +1769 °C -58 ... 3216.2 °F 223.2 ... 2042.2 K (0xB3: ext. comp. 0 °C) (0xC3: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type S: -50 ... +1769 °C -58 ... 3216.2 °F 223.2 ... 2042.2 K (0xB4: ext. comp. 0 °C) (0xC4: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type T: -270 ... +440 °C -454 ... 752 °F 3.2 ... 673.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+5400	10040	8132	Overflow
	-2700 ... +4000	-4540 ... 7520	32 ... 6732	Nominal range
	-	-	-	Underflow
Type B: 0 ... +1820 °C 32 ... 2786.5 °F 273.2 ... 2093.2 K (0xB6: ext. comp. 0 °C) (0xC6: int. comp. 0 °C)	+20700	32766	23432	Overflow
	0 ... +18200	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow

Measuring range (Fct. no.)	Measured value			Range
	[°C]	[°F]	[K]	
Type C: 0 ... +2315 °C 32 ... 2786.5 °F 273.2 ... 2093.2 K (0xB7: ext. comp. 0 °C) (0xC7: int. comp. 0 °C)	+25000	32766	23432	Overflow
	0 ... +23150	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow
Type E: -270 ... +1000 °C -454 ... 1832 °F 0 ... 1273.2 K (0xB8: ext. comp. 0 °C) (0xC8: int. comp. 0 °C)	+12000	21920	14732	Overflow
	-2700 ... +10000	-4540 ... 18320	0 ... 12732	Nominal range
	-	-	-	Underflow
Type L: -200 ... +900 °C -328 ... 1652 °F 73.2 ... 1173.2 K (0xB9: ext. comp. 0 °C) (0xC9: int. comp. 0 °C)	+11500	21020	14232	Overflow
	-2000 ... +9000	-3280 ... 16520	732 ... 11732	Nominal range
	-	-	-	Underflow

Diagnostics and alarm

Trigger	Process alarm	Diagnosealarm	parameterisable
Configuration/parameterisation errors	-	X	-
Open circuit detected	-	X	X
Measuring range exceeded	-	X	-
Measuring range not reached	-	X	-
Limit value exceeded	X	-	X
Limit value not reached	X	-	X
Process alarm lost	-	X	-

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: Limit value exceeded channel 1 Bit 1: Limit value exceeded channel 2 Bit 7 ... 2: 0 (fixed)
2	Bit 0: Limit value not reached, channel 1 Bit 1: Limit value not reached, channel 2 Bit 7 ... 2: 0 (fixed)
3/4	Ticker value at the time of the alarm After mains connection, a timer (μs ticker) is started, which after 65535 μs starts with 0 again.

Diagnosealarm

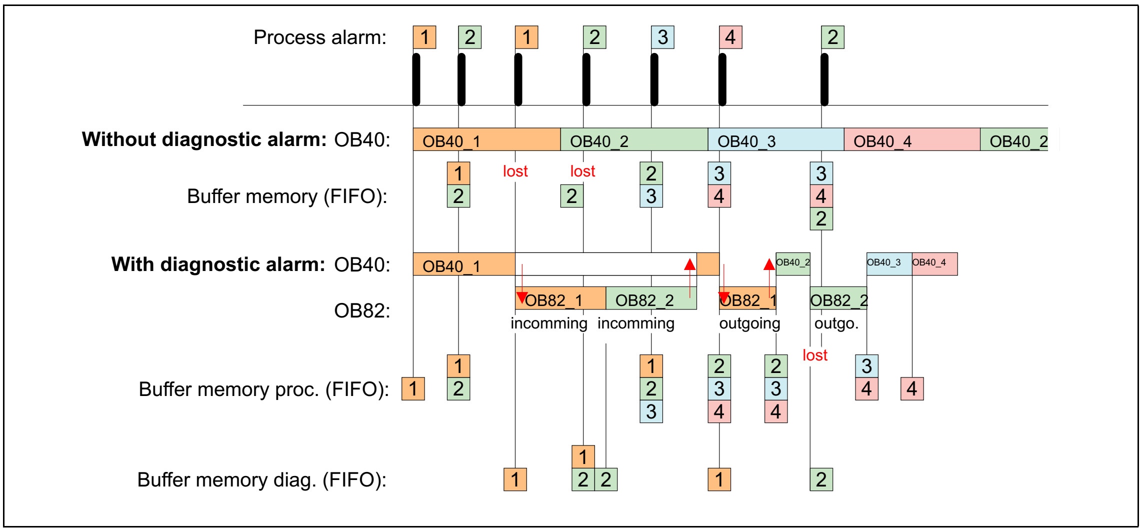
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x01)
7	Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed)
8	Diagnostic alarm due to process alarm lost to ... Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bit 7 ... 5: 0 (fixed)
9 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

6.7 Parameterising the counter

6.7.1 1 counter 32 bits, 24 V DC - EPM-S600

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter a differentiation between the internal gate (I-gate), hardware gate (HW gate), and software gate (SW gate) is made. • The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). • The SW gate is controlled via your user program (status word in the output area). • The HW gate is controlled via the digital gate input. The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Latch function	If a positive edge occurs at the latch input, the current count value is stored in the latch register. The latch register is accessed via the input area. After a STOP-RUN transition, latch is always 0.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.

Additional functions	Description
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Read data: 12 bytes

Input area		
Addr.	**Access**	**Assignment**
+0	Double word	Counter value
+4	Double word	Latch value
+8	Word	Status word
+10	Word	Ticker value

Count value: Current counter content

Latch value: If there is a positive edge at the latch input, the count value is stored here.

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the count value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

Write data: 10 bytes

Output area		
Addr.	**Access**	**Assignment**
+0	Double word	Comparison value
+4	Double word	Set value
+8	Word	Control word

Comparison value: Here you can specify a value which, by comparison with the current counter content, is able to influence the counter output or trigger a process alarm. The response of the output or of the process alarm can be parameterised.

Set value: With an edge change 0-1 of *COUNTERVAL_SET* in the control word, the set value is accepted in the counter.

▸ Counter modules EPM-S600 ... EPM-S603 – control and status words (📖 27)

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
0x01	0	Input frequency Track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
	1	Input frequency Track B		0x02
	2	Input frequency Latch		0x02
	3	Input frequency Gate		0x02
	4	Input frequency Reset		0x00
	5	Reserved		

Data set		**Name**	**Description/value**	**Lenze**
No.	Byte			
0x80	0	Alarm response	Setting activates process alarm Bit 0: Proc. alarm HW gate open Bit 1: Proc. alarm HW gate closed Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 6: Proc. alarm latch value Bit 7: Reserved	0x80
	1	Numerator function	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = single counting, main counting direction is forward 0b000010 = single counting, main counting direction is backward 0b000100 = single counting, no main counting direction 0b001000 = periodic counting, main counting direction is forward 0b010000 = periodic counting, main counting direction is backward 0b100000 = periodic counting, no main counting direction Bit 7 ... 6: Reserved	0x40
	2	Comparator	Bit 2 ... 0: output switches (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 ... 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00
	3	Signal evaluation	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Hardware gate (HW gate) • 0b000 = deactivated (counter starts by setting SW gate) • 0b001 = activated (HIGH level at gate activates the HW gate. Counter starts if HW and SW gate are set.) Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	

Data set		Name	Description/value	Lenze
No.	Byte			
0x81	0	Final value	Upper limitation of the counting range	0x00
	1	Start value	Lower limitation of the counting range	0x00
	2	Hysteresis	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
	3	Pulse	The pulse duration indicates for how long the output is to be set if the parameterised comparison criterion is reached or exceeded. The pulse duration can be specified in steps of 2.048 ms between 0 and 522.24 ms. If the pulse duration is = 0, the output is set until the comparison condition is no longer met.	0x00

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bit 7 ... 5: 0 (fixed)
2	State of the inputs at the time of the alarm Bit 0: A/pulse Bit 1: B/direction Bit 2: Latch Bit 3: Hardware gate Bit 4: Reset Bit 7 ... 5: 0 (fixed)
3/4	Ticker value at the time of the alarm

Gate counter open/closed: Bit 0 is set if the HW gate is activated while the SW gate is active. Bit 1 is set if the HW gate is deactivated while the SW gate is active.

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the count value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

Diagnosealarm

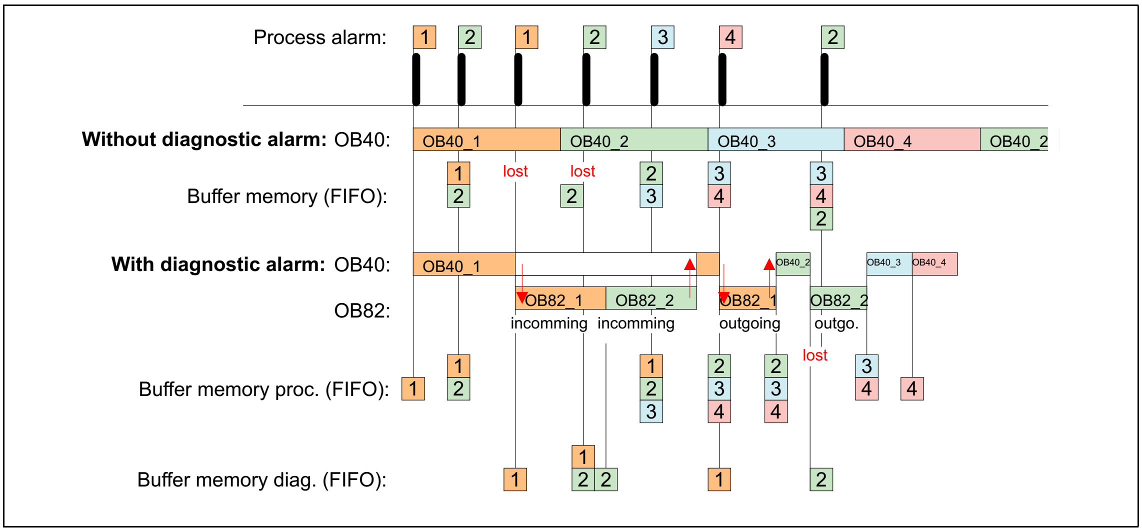
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x01)
7	Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed)
8	Diagnostic alarm due to process alarm lost to ... Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bit 7 ... 5: 0 (fixed)
9 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

6.7.2 2 counters 32 bits, 24 V DC - EPM-S601

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.

Additional functions	Description
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Read data: 12 bytes

Input area in the process image		
Addr.	**Access**	**Assignment**
+0	Double word	Counter 1: count value
+4	Double word	Counter 2: count value
+8	Word	Counter 1: Status word
+10	Word	Counter 2: Status word

Count value: Current counter content

Write data: 12 bytes

Output area in the process image		
Addr.	**Access**	**Assignment**
+0	Double word	Counter 1: Comparison value
+4	Double word	Counter 2: Comparison value
+8	Word	Counter 1: Control word
+10	Word	Counter 2: Control word

Comparison value: With the comparison value you can specify a value which, by comparison with the current counter content, can impact the counter output or trigger a process alarm. The response of the comparison bit *STS_COMP* in the counter status or the process alarm is to be specified via data record 0x80 for counter 1 and 0x82 for counter 2.

▸ Counter modules EPM-S600 ... EPM-S603 – control and status words (📖 27)

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x01	0	Input frequency Counter 1, track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
	1	Input frequency Counter 1, track B		0x02
	2	Input frequency Counter 2, track A		0x02
	3	Input frequency Counter 2, track B		0x02

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	0	Alarm response counter 1	Setting activates process alarm Bit 1 ... 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 ... 6: Reserved	0x00
	1	Counter function counter 1	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 ... 6: Reserved	0x00
	2	Comparator counter 1	Bit 2 ... 0: Comparison bit is set (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
	3	Signal evaluation counter 1	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
0x81	0...3	Set value counter 1	When a set value is given, the counter can be loaded with the set value. With an edge 0-1 at COUNTERVAL_SET in the control word, the set value is accepted in the counter.	0x00
	4...7	Final value counter 1	Upper limitation of the counting range	0x00
	8...11	Loading value counter 1	Lower limitation of the counting range	0x00
	12	Hysteresis counter 1	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
	13	Reserved		

Data set		Name	Description/value	Lenze
No.	Byte			
0x82	0	Alarm response counter 2	See counter 1	0x00
	1	Counter function counter 2	See counter 1	0x00
	2	Comparator counter 2	See counter 1	0x00
	3	Signal evaluation counter 2	See counter 1	0x00
0x83	0...3	Set value counter 2	See counter 1	0x00
	4...7	Final value counter 2	See counter 1	0x00
	8...11	Loading value counter 2	See counter 1	0x00
	12	Hysteresis counter 2	See counter 1	0x00

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: 0 Bit 1: 0 Bit 2: Counter 1, overflow/underflow/final value Bit 3: Counter 1, comparison value reached Bit 4: 0 Bit 5: 0 Bit 6: Counter 2, overflow/underflow/final value Bit 7: Counter 2, comparison value reached
2	State of the inputs at the time of the alarm Bit 0: Counter 1, A/pulse Bit 1: Counter 1, B/direction Bit 2: Counter 2, A/pulse Bit 3: Counter 2, B/direction Bit 7 ... 4: 0 (fixed)
3/4	16 bit µs value at the time of the alarm

Ticker value: After mains connection, a timer (µs ticker) is started which restarts at 0 after 65535 µs. With every change of the count value, the time value of the timer is stored as 16-bit-µs value together with the count value in the input area.

Diagnosealarm

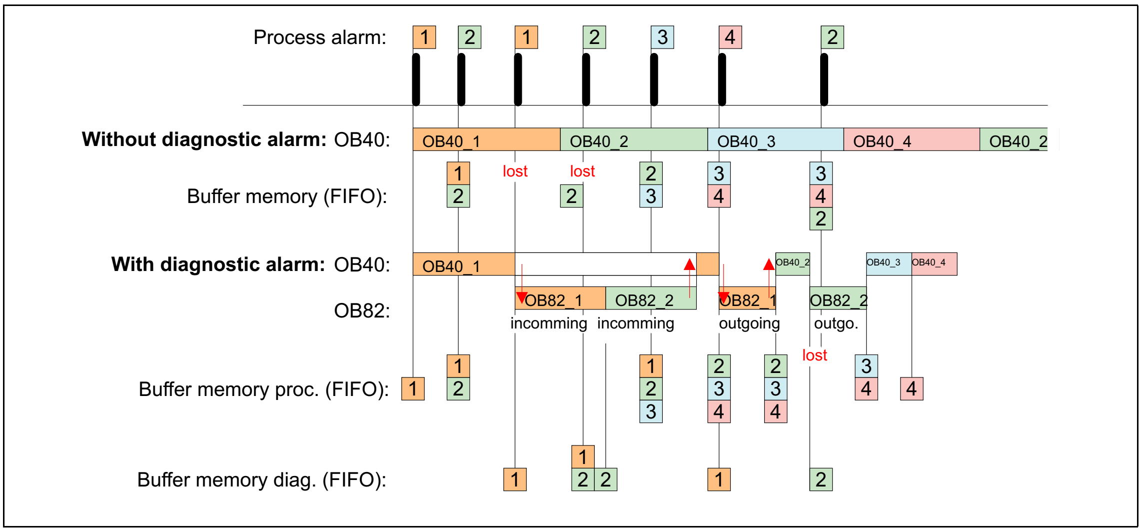
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x02)
7	Bit 0: Error in channel group 0 (counter 1) Bit 1: Error in channel group 1 (counter 2) Bit 7 ... 2: 0 (fixed)
8	Channel group 0: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fixed) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fixed)
9	Channel group 1: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fixed) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fixed)
10 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

6.7.3 1 counter 32 bits, 5 V DC - EPM-S602

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.

Additional functions	Description
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Read data: 8 bytes

Input area		
Addr.	**Access**	**Assignment**
+0	Double word	Counter value
+4	Word	Status word
+6	Word	Ticker value

Count value: Current counter content

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the count value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

Write data: 10 bytes

Output area		
Addr.	**Access**	**Assignment**
+0	Double word	Comparison value
+4	Double word	Set value
+8	Word	Control word

Comparison value: Here you can specify a value which, by comparison with the current counter content, is able to influence the counter output or trigger a process alarm. The response of the output or of the process alarm can be parameterised.

Set value: With an edge change 0-1 of *COUNTERVAL_SET* in the control word, the set value is accepted in the counter.

▸ Counter modules EPM-S600 ... EPM-S603 – control and status words (📖 27)

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x01	0	Input frequency Track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
	1	Input frequency Track B		0x02
	2	Input frequency Reset		0x02
	3	Reserved		

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	0	Alarm response	Setting activates process alarm Bit 1 ... 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 ... 6: Reserved	0x00
	1	Numerator function	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 ... 6: Reserved	0x00
	2	Comparator	Bit 2 ... 0: Comparison bit is set (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 ... 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00
	3	Signal evaluation	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
0x81	4...7	Final value	Upper limitation of the counting range	0x00
	8...11	Start value	Lower limitation of the counting range	0x00
	12	Hysteresis		0x00

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: 0 Bit 1: 0 Bit 2: Counter 1, overflow/underflow/final value Bit 3: Counter 1, comparison value reached Bit 4: 0 Bit 5: 0 Bit 6: Counter 2, overflow/underflow/final value Bit 7: Counter 2, comparison value reached
2	State of the inputs at the time of the alarm Bit 0: Counter 1, A/pulse Bit 1: Counter 1, B/direction Bit 2: Counter 2, A/pulse Bit 3: Counter 2, B/direction Bit 7 ... 4: 0 (fixed)
3/4	16 bit μs value at the time of the alarm

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the count value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

Diagnosealarm

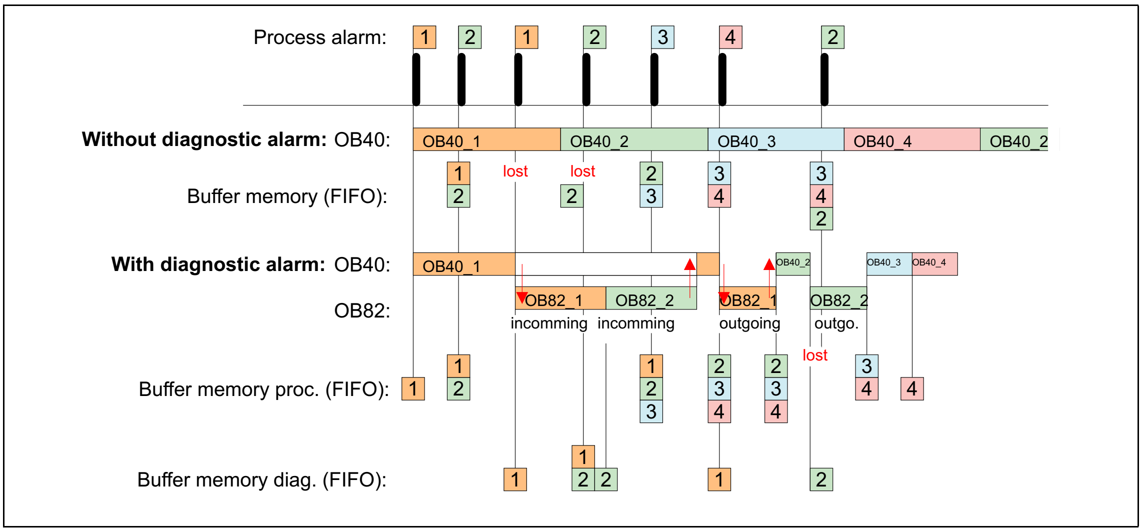
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{\text{incoming}}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{\text{incoming}}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{\text{outgoing}}$.

All events of a channel between the diagnostic alarm$_{\text{incoming}}$ and diagnostic alarm$_{\text{outgoing}}$ are not buffered and get lost. During this time (first diagnostic alarm$_{\text{incoming}}$ to the last diagnostic alarm$_{\text{outgoing}}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{\text{incoming/outgoing}}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x02)
7	Bit 0: Error in channel group 0 (counter 1) Bit 1: Error in channel group 1 (counter 2) Bit 7 ... 2: 0 (fixed)
8	Channel group 0: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fixed) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fixed)
9	Channel group 1: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fixed) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fixed)
10 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

6.7.4 2 counters 32 bits, 24 V DC - EPM-S603

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from 0 to the counting limit, then skips to the opposite counting limit and continues to count from there.

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area).

Read data: 12 bytes

Input area in the process image		
Addr.	**Access**	**Assignment**
+0	Double word	Counter 1: count value
+4	Double word	Counter 2: count value
+8	Word	Counter 1: Status word
+10	Word	Counter 2: Status word

Write data: 4 bytes

Output area in the process image		
Addr.	**Access**	**Assignment**
+0	Word	Counter 1: Control word
+2	Word	Counter 2: Control word

▶ Counter modules EPM-S600 ... EPM-S603 – control and status words (📖 27)

Parameter data

Data set No.	Data set Byte	Name	Description/value	Lenze
0x01	0	Input frequency Counter 1, track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
	1	Input frequency Counter 1, track B		0x02
	2	Input frequency Counter 2, track A		0x02
	3	Input frequency Counter 2, track B		0x02
0x80	0	Counting direction counter 1, track B	Bit 2 ... 0: Reserved Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
	1	Signal evaluation counter 1	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 7 ... 3: Reserved	0x00
0x82	0	Counting direction counter 2, track B	Bit 2 ... 0: Reserved Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
	1	Signal evaluation counter 2	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 7 ... 3: Reserved	0x00

Diagnostic data

Using the SFB 52 you can access the diagnostic data of the module any time. Since this module does not support a process alarm, the diagnostic data serve to provide information on this module.

Data set 1 is structured as follows:

Byte	Bit 7 ... 0
0	0 (fixed)
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x76: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x00))
6	Number of channels of a module (here 0x02)
7 ... 15	0 (fixed)

6.8 Parameterising the encoder evaluation

6.8.1 SSI - EPM-S604

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Functions	Description
SSI encoder parameters	According to encoder data sheet
Operating mode	Master mode or monitoring operation
Alarm response	With definition of the comparison and limit values

Read data: 6 bytes

Input area		
Addr.	**Access**	**Assignment**
+0	Double word	Encoder value
+4	Word	Ticker value

Encoder value: current encoder value

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the encoder value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

Parameter data

Data set No.	Data set Byte	Name	Description/value	Lenze
0x00	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x80	0	Idle time	The dead time, also called tbs (time between sends), defines the waiting time between two encoder values to be observed by the module so that the encoder is able to process its value. This data can be found in the data sheet for your encoder. HIGH LOW 0x00 0x30: 1 μs 0x00 0x60: 2 μs 0x00 0xC0: 4 μs 0x01 0x80: 8 μs 0x03 0x00: 16 μs 0x06 0x00: 32 μs 0x09 0x00: 48 μs 0x0C 0x00: 64 μs	0x0C00
	1	Baud rate	In "monitoring operation" operating mode, the baud rate is irrelevant. Enter the baud rate here. This corresponds to the clock frequency via which the encoder connected communicates. Information on this can be found in the data sheet for your encoder. HIGH LOW 0x00 0x18: 2 MHz 0x00 0x20: 1.5 MHz 0x00 0x30: 1 MHz 0x00 0x60: 500 kHz 0x00 0xC0: 250 kHz 0x01 0x80: 125 kHz	0x0180
	2	Reserved		
	3	Standardisation	Depending on the encoder, further bits are transmitted in addition to the encoder value. Scaling serves to determine how many bits post-positioned to the encoder value will be removed by shifting the encoder value to the right. The encoder value is scaled by the module only after a Gray-binary conversion. More information can be found in the data sheet for your encoder. Value range: 0x00 ... 0x0F = 0 bit ... 15 bits	0x00

<table>
<tr><th colspan="2">Data set</th><th rowspan="2">Name</th><th rowspan="2">Description/value</th><th rowspan="2">Lenze</th></tr>
<tr><th>No.</th><th>Byte</th></tr>
<tr><td rowspan="2">0x80
(Continuation)</td><td>4</td><td>Bit length of encoder data</td><td>Enter the bit length of the encoder data here. Depending on the encoder, the encoder data consist of the current encoder value with subsequent bits. The total length has to be specified here. More information on this can be found in the data sheet for your encoder.
7 (0x07) = "8 bits"
8 (0x08) = "9 bits"
...
24 (0x18) = "25 bits"
...
31 (0x1F) = "32 bits"</td><td>0x18</td></tr>
<tr><td>5</td><td></td><td>Bit 1 ... 0: Ready for operation
During "Monitoring operation" the module serves to monitor the data exchange between an SSI master and an SSI encoder. It receives the cycle by the master and the data flow by the SSI encoder.
In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder.
0b01 = monitoring operation
0b10 = master mode

Bit 2: Shifting direction
Specify the orientation of the encoder data here. More information can be found in the data sheet for your encoder. Usually, the SSI encoder uses "MSB first".
0 = LSB first (LSB is transmitted first)
1 = MSB first (MSB is transmitted first)

Bit 3: Edge clock signal
Here you can specify the edge type of the clock signal, in the case of which the encoder supplies data. Information on this can be found in the data sheet for your encoder. Usually the SSI encoders respond to rising edges.
0 = falling edge
1 = rising edge
Master mode: Connect clock output signal (ClockOut) to the EPM-S604.
Monitoring mode: Connect clock input signal (ClockIn) to the EPM-S604.

Bit 4: Coding
In the "Binary code" setting, the provided encoder value remains unchanged. In the "Gray code" setting, the Gray-coded value provided by the encoder is converted into a binary value. Only after this conversion, the received encoder value is scaled, if required. The Gray code is another form of representation of the binary code. It is based on the fact that two adjacent Gray numbers differ from each other in exactly one bit. If the Gray code is used, transmission errors can be easily detected, since adjacent characters must only differ from each other in one digit. Information on this can be found in the data sheet for your encoder.
0 = standard code
1 = Gray code

Bit 7 ... 5: Reserved</td><td>0x1E</td></tr>
</table>

Data set		Name	Description/value	Lenze
No.	Byte			
	6	Reserved		
	7	SSI function	By enabling the SSI function the module starts with the cycle output and the evaluation of the encoder data. In the "monitor operation" mode, the module starts with the encoder evaluation. 0 (0x00) = inhibited 1 (0x01) = enabled	0x00

Diagnosealarm

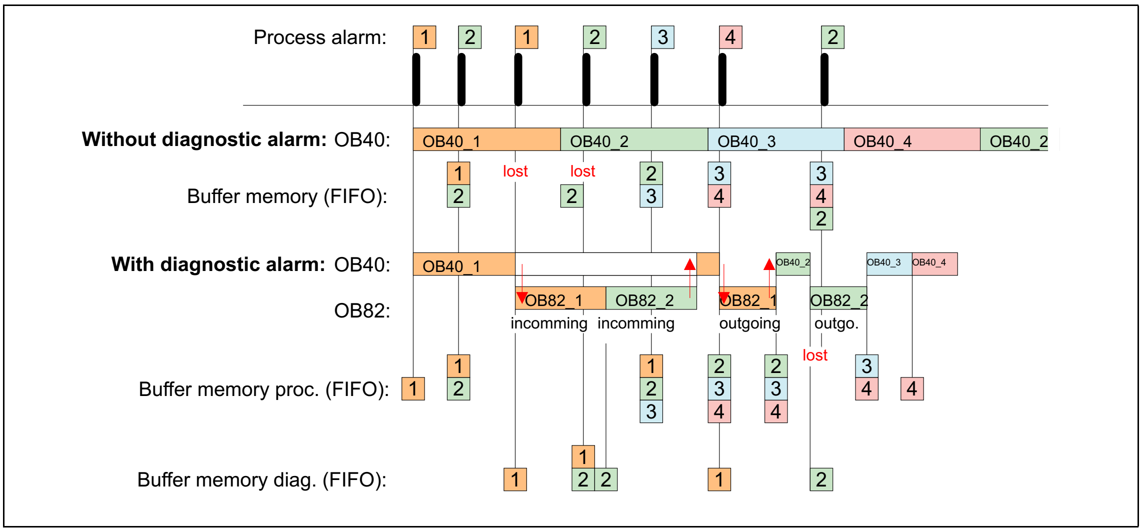
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of missing external supply voltage Bit 6 ... 5: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x01)
7	Bit 0: Error in channel group 0
8 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$	
Byte	**Bit 7 ... 0**
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.	

6.9 Parameterising the time stamp

6.9.1 2 digital inputs with time stamp function - EPM-S207

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Functions	Description
Input delay	For example, signal peaks can be filtered in the event of an unclean input signal.
Edge selection	Specification of signal edge for input signal to produce a time stamp entry.

Read data: 6 bytes

Input area		
Addr.	**Access**	**Assignment**
+0	Byte	Status of inputs (PAE)
+1	Byte	Running number (RN)
+2	Word	Ticker value

Status of inputs: The status of the inputs after edge change is saved here. Parameters can be set for the following variants by integrating the gse file LE010C3A.gse:

20 bytes, 5 time stamp entries:

Addr.	+0	+1	+2	+3
+0	PAE	RN	16-bit µs value	
+4	PAE	RN-1	16-bit µs value	
+8	PAE	RN-2	16-bit µs value	
+12	PAE	RN-3	16-bit µs value	
+16	PAE	RN-4	16-bit µs value	

60 bytes, 15 time stamp entries:

Addr.	+0	+1	+2	+3
+0	PAE	RN	16-bit µs value	
+4	PAE	RN-1	16-bit µs value	
+8	PAE	RN-2	16-bit µs value	
+12	PAE	RN-3	16-bit µs value	
...	...	...	...	
+56	PAE	RN-14	16-bit µs value	

Running number (RN): The "running number" is a consecutive number between 0 ... 63, which always starts afresh from 0. You use the "running number" to determine the time sequence of entries. It must be incremented with every time stamp entry.

In the first run, the "running number" must start with 1.

Ticker value: After mains connection, a timer (µs ticker) is started which restarts at 0 after 65535 µs. With every change of the encoder value, the time value of the timer is stored as 16-bit-µs value together with the count value in the input area.

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C (fix)
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)
0x01	0	Input delay DI 1	0x00 = 1 µs 0x02 = 3 µs 0x04 = 10 µs 0x07 = 86 µs 0x09 = 342 µs 0x0C = 273 µs No other values are permissible.	0x02
	1	Input delay DI 2		0x02
0x80	0	Edge 0-1 at DI x	Time stamp entry on rising edge Bit 0: DI 1 (0: inhibit, 1 = enable) Bit 1: DI 2 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00
	1	Edge 1-0 on DI x	Time stamp entry on falling edge Bit 0: DI 1 (0: inhibit, 1 = enable) Bit 1: DI 2 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00

Diagnostic data

Using SFB 52, you can read out the diagnostic bytes which provide information about the module. With SFB 52 you can also read out data set 1 which contains further information.

Data set 1 is structured as follows:

Data set 1, diagnostics	
Byte	**Bit 7 ... 0**
0	0 (fixed)
1	Bit 3 ... 0: Module class, 0b1111: Digital module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x70: Digital module Bit 7: more channel types available (0: yes; 1: no)
5	Number of diagnostic bits output by the module per channel (here 0x00))
6	Number of channels of a module (here 0x02)
7 ... 15	0 (fixed)

6.9.2 2 digital outputs with time stamp function - EPM-S310

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The module has an FIFO (first-in-first-out) memory for 15 time stamp entries. Depending on parameter setting, you can use the output area to transfer up to 15 time stamp entries to the FIFO memory. The input process image provides information on the status of the FIFO memory and the status of processing.

Input area		
Addr.	**Access**	**Assignment**
+0	Byte	Bit 5 ... 0: Running number (RN = Running Number) of the last FIFO entry Bit 6: 1 (fixed) Bit 7: 0 (fixed)
+1	Byte	Bits 5 ... 0: running number of the next FIFO entry Bit 6: 1 (fixed) Bit 7: 1 (fixed)
+2	Byte	Status
+3	Byte	Number of the time stamp entries in the FIFO memory

Running number (RN): Here you will find the "running number" of the time stamp entry last/next written to the FIFO.

Status: The status informs you of the status of the FIFO memory:

Code 0x00/0x80: Everything is OK

Code 0x01/0x81: No following time stamp entry available

Code 0x02/0x82: No new time stamp entries available.

Code 0x03/0x83: FIFO memory is full. No new time stamp entry can be accepted.

If bit 6 of the last processed "running number" was set, the code is returned at 0x80 OR-ed.

Note!

Note that no more time stamp entries can be accepted once the FIFO memory is full. You should always establish the status of the FIFO memory before the transfer to ensure that your entries are accepted.

Write data: 20 bytes/60 bytes

Depending on project planning, the output area can be used to write up to 15 time stamp entries. 4 bytes in the process image are intended for each time stamp entry:

Output area		
Addr.	**Access**	**Assignment**
+0	Byte	Bits 3 ... 0: 0 (fixed) Bit 4: Enable of DO 1 (0: Disable, 1: Enable) Bit 5: Enable of DO 0 (0: Disable, 1: Enable) Bit 6: DO 1 status Bit 7: DO 0 status
+1	Byte	Running number (RN)
+2	Word	Ticker value

Status of outputs:the status of the outputs for the time required is stated here. You can project plan the following variants by incorporating the GSE file LE010C3A.gse.gse:

20 bytes, 5 time stamp entries:

Addr.	+0	+1	+2	+3
+0	PAA	RN	16-bit µs value	
+4	PAA	RN-1	16-bit µs value	
+8	PAA	RN-2	16-bit µs value	
+12	PAA	RN-3	16-bit µs value	
+16	PAA	RN-4	16-bit µs value	

60 bytes, 15 time stamp entries:

Addr.	+0	+1	+2	+3
+0	PAA	RN	16-bit µs value	
+4	PAA	RN-1	16-bit µs value	
+8	PAA	RN-2	16-bit µs value	
+12	PAA	RN-3	16-bit µs value	
...	...	...	...	
+56	PAA	RN-14	16-bit µs value	

Running number (RN): The "running number" is a consecutive number between 0 ... 63, which always starts afresh from 0. The "running number" reflects the time sequence of the edges

In the first run, the "running number" must start with 1.

Note!

If using SFC 15 to write consistent user data, up to 15 time stamp entries can be written. If less than 15 time stamp entries are written, bit 6 must also be set for the last RN. This has to be done to ensure that the following entries don't have to be written in an "invalid" way. The module ignores all time stamp entries after an entry with a set bit 6.

Ticker value: Specify a time here in µs at which the status of the outputs is to be accepted (value range: 0 ... 65535).

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C (fix)
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)

Diagnostic data

Using SFB 52, you can read out the diagnostic bytes which provide information about the module. With SFB 52 you can also read out data set 1 which contains further information.

Data set 1 is structured as follows:

Data set 1, diagnostics	
Byte	**Bit 7 ... 0**
0	0 (fixed)
1	Bit 3 ... 0: Module class, 0b1111: Digital module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x70: Digital module Bit 7: more channel types available (0: yes; 1: no)
5	Number of diagnostic bits output by the module per channel (here 0x00))
6	Number of channels of a module (here 0x02)
7 ... 15	0 (fixed)

6.10 Parameterising technology modules

6.10.1 2 digital outputs with PWM functionality - EPM-S620

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

Read data: 4 bytes

Input area			
Addr.	**Name**	**Byte**	**Function**
+0	PWMSTS_I	2	PWM 1: Status word
+2	PWMSTS_II	2	PWM 2: Status word

Status PVMx		
Bit	**Name**	**Function**
0	-	Reserved
1	STS_PVM	Status PWM 0: PWM output stopped 1: PWM output active
2	STS_OUTBV	Output status 0: Push/Pull output 1: Highside output
3 ... 15	-	Reserved

Write data: 12 bytes

Output area			
Addr.	**Name**	**Byte**	**Function**
+0	PWMPD_I	4	PWM 1: pulse duration
+4	PWMPD_II	4	PWM 2: pulse duration
+8	PWMCTRL_I	2	PWM 1: Control word
+10	PWMCTRL_II	2	PWM 2: Control word

- PWMPD_I, PWMPD_II (pulse duration):
  Determine the scanning ratio for the parameterised period by stating the duration for the HIGH level for the corresponding PWM channel. The pulse duration should be chosen as factor for the 20.83 ns basis.
  Value range: 48 ... 8388607 (1µs ... approx. 175ms)
- PWMPD_I, PWMPD_II (control word):
  Here you can define the PWM output behaviour for the corresponding channel and start or stop the PWM output.

▸ PWM module EPM-S620 – control and status word (📖 32)

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	0	PWM 1: period	Set parameters here for the total time for pulse duration and pulse pause. The time should be selected as a factor for the 20.83 ns basis. Values below 25 µs are ignored. If the pulse duration is higher or equal to the period, the DO output is set permanently. Value range: 1200 ... 8388607 (25 µs ... approx. 175 ms)	0x1F40
0x81	0	PWM 2: Period		0x1F40

Diagnostic data

Since this module does not support alarms, the diagnostic data provides information about this module.

Diagnostic data record - data record 0x01			
Name	**Byte**	**Function**	**Default**
ERR_A	1	Reserved	0x00
MODTYP	1	Module information Byte 0: Bit 3 ... 0: Module class (0b1111: Digital module) Bit 4: channel information available Bits 7 ... 4: 0 reserved	0x15
ERR_C	1	Reserved	0x00
ERR_D	1	Reserved	0x00
CHTYP	1	Channel type Byte 0: Bit 6 ... 0: Channel type (0x72: Digital output) Bit 7: Reserved	0x72
NUMBIT	1	Number of diagnostic bits per channel Byte 0: Here 0x00	0x00
NUMCH	1	Number of channels in module Byte 0: Here 0x02	0x02
CHERR	1	Reserved	0x00
CH0ERR ... CH7ERR	6	Reserved	0x00
DIAG_US	4	Value of µs ticker when diagnostics occur Bytes 0 ...3	0

6.10.2 RS232 interface - EPM-S640

Parameter data - ASCII protocol

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x01: ASCII	0x01
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA				

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	3	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	4	ZNA (LOW byte)		0x00
Option 4 / 5, character delay time				
0x80	5	Character delay time (HIGH byte)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	6	Character delay time (LOW byte)		0xFA
Option 6, number of receive buffers				
0x80	7	Number of receive buffers	Defines the number of receive buffers. As long as only one receive buffer is used and is assigned, no more data can be received. If up to 250 receive buffers are connected, the received data can be redirected to a still free receive buffer. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7 ... 12, reserved				
0x80	8 ... 13	Reserved		0x00

Parameter data STX/ETX protocol

Data set No.	Data set Byte	Name	Description/value	Lenze
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA				
0x80	3	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	4	ZNA (LOW byte)		0x00

Data set		Name	Description/value	Lenze
No.	Byte			
Option 4 / 5, TMO				
0x80	5	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	6	TMO (LOW byte)		0xFA
Option 6, number of start identifiers				
0x80	7	Number of start identifiers	0x00: 1 start identifier (2. start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1				
0x80	8	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2				
0x80	9	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers				
0x80	10	Number of end identifiers	0x00: 1 end identifier (2. end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1				
0x80	11	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2				
0x80	12	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved				
0x80	13	Reserved		0x00

Parameter data 3964(R) protocol

Data set No.	Byte	Name	Description/value	Lenze
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)				

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	3	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 3, character delay time (x 20 ms)				
0x80	4	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)				
0x80	5	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)				
0x80	6	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repititons				
0x80	7	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL				
0x80	8	DBL	If the block wait time is exceeded, you can select the maximum number of repititions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority				
0x80	9	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved				
0x80	10 ... 13	Reserved		0x00

Diagnostic data

In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

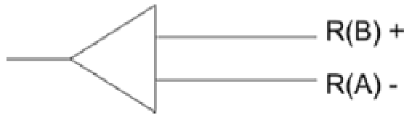
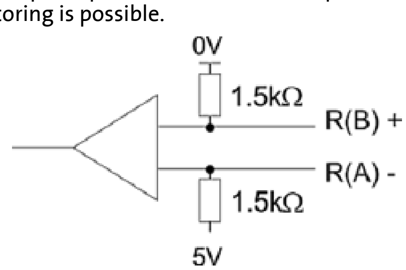
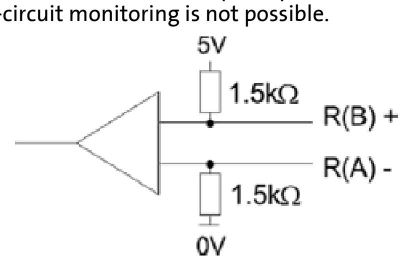
Diagnostic data record - data record 0x01			
Name	**Byte**	**Function**	**Default**
ERR_A	1	ERR_A-diagnostics Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error (cable break) Bit 3: Reserved Bit 4: Set in the case of a missing external supply voltage Bit 5, 6: Reserved Bit 7: Set in the case of parameterisation error	0x00
MODTYP	1	Module information Byte 0: Bit 3 ... 0: Module class (0b0111: Gateway module) Bit 4: channel information available Bits 7 ... 4: 0 reserved	0x17
ERR_C	1	ERR_A-diagnostics Bit 7 ... 0: Reserved	0x00
ERR_D	1	ERR_D diagnostics Bit 3 ... 0: Reserved Bit 4: Set in the case of internal communication error Bit 7 ... 5: Reserved	0x00
CHTYP	1	Channel type Bit 7 ... 0: Reserved	0x00
NUMBIT	1	Number of diagnostic bits of the module per channel (here 0x08)	0x08
NUMCH	1	Number of channels in module Bit 7 ... 0: Reserved	0x00
CHERR	1	Bit 7 ... 0: Reserved	0x00
CH0ERR ... CH7ERR	8	Bit 7 ... 0: Reserved	0x00
DIAG_US	4	Value of µs ticker when diagnostics occur Bytes 0 ...3	0x00

6.10.3 RS422/RS485 interface - EPM-S650

Parameter data - ASCII protocol

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x01: ASCII	0x01
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA				

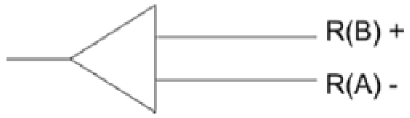
Data set		Name	Description/value	Lenze
No.	Byte			
0x80	3	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	4	ZNA (LOW byte)		0x00
Option 4 / 5, character delay time				
0x80	5	Character delay time (HIGH byte)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	6	Character delay time (LOW byte)		0xFA
Option 6, number of receive buffers				
0x80	7	Number of receive buffers	Defines the number of receive buffers. As long as only one receive buffer is used and is assigned, no more data can be received. If up to 250 receive buffers are connected, the received data can be redirected to a still free receive buffer. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7 ... 12, reserved				
0x80	8 ... 13	Reserved		0x00
Option 13, operating mode				
0x80	14	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

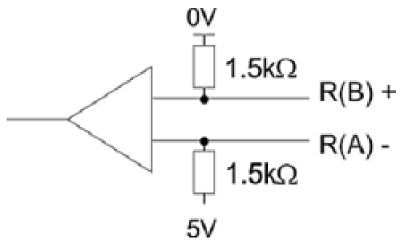
Data set		Name	Description/value	Lenze
No.	Byte			
Option 14, cable assignment				
0x80	15	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible.  	0x00

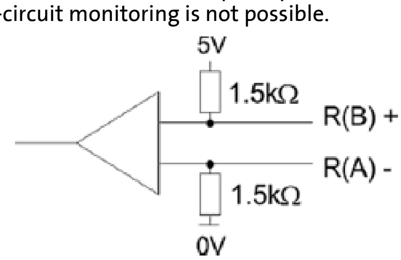
Parameter data STX/ETX protocol

Data set No.	Data set Byte	Name	Description/value	Lenze
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA				
0x80	3	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	4	ZNA (LOW byte)		0x00

Data set No.	Data set Byte	Name	Description/value	Lenze
Option 4 / 5, TMO				
0x80	5	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	6	TMO (LOW byte)		0xFA
Option 6, number of start identifiers				
0x80	7	Number of start identifiers	0x00: 1 start identifier (2. start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1				
0x80	8	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2				
0x80	9	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers				
0x80	10	Number of end identifiers	0x00: 1 end identifier (2. end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1				
0x80	11	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2				
0x80	12	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved				
0x80	13	Reserved		0x00
Option 13, operating mode				
0x80	14	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

<table>
<tr><th colspan="2">Data set</th><th rowspan="2">Name</th><th rowspan="2">Description/value</th><th rowspan="2">Lenze</th></tr>
<tr><th>No.</th><th>Byte</th></tr>
<tr><td colspan="5">Option 14, cable assignment</td></tr>
<tr><td>0x80</td><td>15</td><td>Cable assignment</td><td>For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level.
• 0x00:
No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.


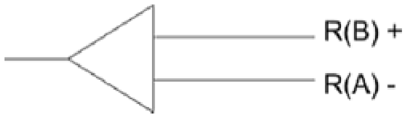
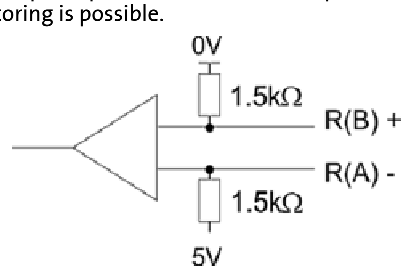
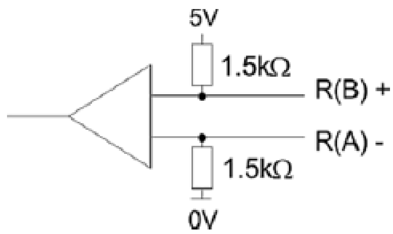
• 0x01:
Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V
In full duplex operation under RS422, open-circuit monitoring is possible.


• 0x02:
Signal R(A) 0V, signal R(B) 5V
This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible.

</td><td>0x00</td></tr>
</table>

Parameter data 3964(R) protocol

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)				

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	3	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 3, character delay time (x 20 ms)				
0x80	4	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)				
0x80	5	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)				
0x80	6	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repititons				
0x80	7	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL				
0x80	8	DBL	If the block wait time is exceeded, you can select the maximum number of repititions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority				
0x80	9	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved				
0x80	10 ... 13	Reserved		0x00
Option 13, operating mode				

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	14	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01
Option 14, cable assignment				
0x80	15	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible.  	0x00

Diagnostic data

In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

Diagnostic data - data record 0x01			
Name	**Byte**	**Function**	**Default**
ERR_A	1	ERR_A-diagnostics Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error (cable break) Bit 3: Reserved Bit 4: Set in the case of a missing external supply voltage Bit 5, 6: Reserved Bit 7: Set in the case of parameterisation error	0x00
MODTYP	1	Module information Byte 0: Bit 3 ... 0: Module class (0b0111: Gateway module) Bit 4: channel information available Bits 7 ... 4: 0 reserved	0x17
ERR_C	1	ERR_A-diagnostics Bit 7 ... 0: Reserved	0x00
ERR_D	1	ERR_D diagnostics Bit 3 ... 0: Reserved Bit 4: Set in the case of internal communication error Bit 7 ... 5: Reserved	0x00
CHTYP	1	Channel type Bit 7 ... 0: Reserved	0x00
NUMBIT	1	Number of diagnostic bits of the module per channel (here 0x08)	0x08
NUMCH	1	Number of channels in module Bit 7 ... 0: Reserved	0x00
CHERR	1	Bit 0: set in the event of an error of channel group 1 Bits 7 ... 1 0 (fixed)	0x00
CH0ERR	8	Channel-specific error: channel x: Bits 3 ... 0: 0 (fixed) Bit 4 : set in the case of open circuit (only possible for RS422) Bits 7 ... 5: 0 (fixed)	0x00
CH1ERR ... CH7ERR	8	Bit 7 ... 0: Reserved	0x00
DIAG_US	4	Value of μs ticker when diagnostics occur Bytes 0 ...3	0x00

6.11 Diagnostics

The extensive diagnostic functions under PROFIBUS-DP enable a quick error localisation. The diagnostic data are transmitted via the bus and summarised in the master. There, you can, for instance, access the diagnostic data with your project planning tool. The diagnostic message created by the PROFIBUS slave, have a maximum length of 122 bytes, depending on the parameterisation. As soon as the slave sends a diagnostic message to the master, the max. 122 bytes of diagnostic data are prefixed with 6 bytes of slave standard diagnostic data.

Structure of the diagnostic data	
Byte	**Function**
0 ... 5	Standard diagnostic data The master is only prefixed in case the transfer is executed via PROFIBUS.
x ... x + 8	ID-related diagnostics (can be disabled/enabled via parameterisation)
x ... x + 19	Module status (can be disabled/enabled via parameterisation)
max. 21 × (x ... x + 2)	Channel-related diagnostics (can be disabled/enabled via parameterisation)
x ... x + 20	Alarm (can be disabled/enabled via parameterisation)

6.11.1 Slave standard diagnostic data

If a diagnostic message is transmitted to the master, the diagnostic bytes are prefixed with the standard diagnostic data.

Standard diagnostic data	
Byte	**Function**
0	Bit 0: 0 (fix) Bit 1: Slave is not ready for data exchange Bit 2: Configuration data do not match with each other Bit 3: Slave has external diagnostic data Bit 4: Slave does not support the requested function Bit 5: 0 (fix) Bit 6: Wrong parameterisation Bit 7: 0 (fixed)
1	Bit 0: Slave has to be re-parameterised Bit 1: Statistical diagnostics Bit 2: 1 (fix) Bit 3: Response monitoring is active Bit 4: "FREEZE" command received Bit 5: "SYNC" command received Bit 6: Reserved Bit 7: 0 (fixed)
2	Bit 6 ... 0: Reserved Bit 7: Diagnostic data overflow
3	Master address after parameterisation • 0xFF: Slave is without parameterisation
4	ID number High Byte
5	ID number Low Byte

More information on the structure of the standard diagnostic data can be found in the documentation of the PROFIBUS user organisation.

6.11.2 ID-related diagnostics

The ID-related diagnostics serves to inform in which PROFIBUS slot (module) an error has occurred. More information on the error can be obtained with the module status and the channel-related diagnostics. The ID-related diagnostics can be activated via the parameterisation.

Diagnostic data	
Byte	**Function**
x	Bit 5 ... 0: • 0b001001 (fix): Length of the ID-related diagnostics Bit 7 ... 6: • 0b01 (fix): Code for the ID-related diagnostics
x + 1	The bits of the modules per PROFIBUS slot are set if: • the module is removed; • a non-projected module is plugged-in; • a module cannot be accessed; • a module reports a diagnostic alarm. Bit 0: Entry of module in PROFIBUS slot 1 ... Bit 7: Entry of module in PROFIBUS slot 8
x + 2	Bit 0: Entry of module in PROFIBUS slot 9 ... Bit 7: Entry of module in PROFIBUS slot 16
x + 3	Bit 0: Entry of module in PROFIBUS slot 17 ... Bit 7: Entry of module in PROFIBUS slot 24
x + 4	Bit 0: Entry of module in PROFIBUS slot 25 ... Bit 7: Entry of module in PROFIBUS slot 32
x + 5	Bit 0: Entry of module in PROFIBUS slot 33 ... Bit 7: Entry of module in PROFIBUS slot 40
x + 6	Bit 0: Entry of module in PROFIBUS slot 41 ... Bit 7: Entry of module in PROFIBUS slot 48
x + 7	Bit 0: Entry of module in PROFIBUS slot 49 ... Bit 7: Entry of module in PROFIBUS slot 56
x + 8	Bit 0: Entry of module in PROFIBUS slot 57 ... Bit 7: Entry of module in PROFIBUS slot 64

6.11.3 Module status

The module status serves to provide more information on the error occurred in a module. The module status can be activated via the parameterisation.

Diagnostic data	
Byte	**Function**
x	Bit 5 ... 0: • 0b001001 (fix): Length of the module status Bit 7 ... 6: • 0b01 (fix): Code for the module status
x + 1	0x82 (fix): Status type of the module status
x + 2	0x00 (fix)
x + 3	0x00 (fix)
x + 4	For PROFIBUS slot 1 ... 64, the following errors are specified: • 0b00: Module has valid data • 0b01: Module error - invalid data (module defective) • 0b10: Wrong module - invalid data • 0b11: No module plugged-in - invalid data Bit 1, 0: Module status module in PROFIBUS slot 1 Bit 3, 2: Module status module in PROFIBUS slot 2 Bit 5, 4: Module status module in PROFIBUS slot 3 Bit 7, 6: Module status module in PROFIBUS slot 4
x + 5	Bit 1, 0: Module status module in PROFIBUS slot 5 Bit 3, 2: Module status module in PROFIBUS slot 6 Bit 5, 4: Module status module in PROFIBUS slot 7 Bit 7, 6: Module status module in PROFIBUS slot 8
x + 6	Bit 1, 0: Module status module in PROFIBUS slot 9 Bit 3, 2: Module status module in PROFIBUS slot 10 Bit 5, 4: Module status module in PROFIBUS slot 11 Bit 7, 6: Module status module in PROFIBUS slot 12
x + 7	Bit 1, 0: Module status module in PROFIBUS slot 13 Bit 3, 2: Module status module in PROFIBUS slot 14 Bit 5, 4: Module status module in PROFIBUS slot 15 Bit 7, 6: Module status module in PROFIBUS slot 16
x + 8	Bit 1, 0: Module status module in PROFIBUS slot 17 Bit 3, 2: Module status module in PROFIBUS slot 18 Bit 5, 4: Module status module in PROFIBUS slot 19 Bit 7, 6: Module status module in PROFIBUS slot 20
x + 9	Bit 1, 0: Module status module in PROFIBUS slot 21 Bit 3, 2: Module status module in PROFIBUS slot 22 Bit 5, 4: Module status module in PROFIBUS slot 23 Bit 7, 6: Module status module in PROFIBUS slot 24
x + 10	Bit 1, 0: Module status module in PROFIBUS slot 25 Bit 3, 2: Module status module in PROFIBUS slot 26 Bit 5, 4: Module status module in PROFIBUS slot 27 Bit 7, 6: Module status module in PROFIBUS slot 28
x + 11	Bit 1, 0: Module status module in PROFIBUS slot 29 Bit 3, 2: Module status module in PROFIBUS slot 30 Bit 5, 4: Module status module in PROFIBUS slot 31 Bit 7, 6: Module status module in PROFIBUS slot 32
x + 12	Bit 1, 0: Module status module in PROFIBUS slot 33 Bit 3, 2: Module status module in PROFIBUS slot 34 Bit 5, 4: Module status module in PROFIBUS slot 35 Bit 7, 6: Module status module in PROFIBUS slot 36

Diagnostic data	
Byte	**Function**
x + 13	Bit 1, 0: Module status module in PROFIBUS slot 37 Bit 3, 2: Module status module in PROFIBUS slot 38 Bit 5, 4: Module status module in PROFIBUS slot 39 Bit 7, 6: Module status module in PROFIBUS slot 40
x + 14	Bit 1, 0: Module status module in PROFIBUS slot 41 Bit 3, 2: Module status module in PROFIBUS slot 42 Bit 5, 4: Module status module in PROFIBUS slot 43 Bit 7, 6: Module status module in PROFIBUS slot 44
x + 15	Bit 1, 0: Module status module in PROFIBUS slot 45 Bit 3, 2: Module status module in PROFIBUS slot 46 Bit 5, 4: Module status module in PROFIBUS slot 47 Bit 7, 6: Module status module in PROFIBUS slot 48
x + 16	Bit 1, 0: Module status module in PROFIBUS slot 49 Bit 3, 2: Module status module in PROFIBUS slot 50 Bit 5, 4: Module status module in PROFIBUS slot 51 Bit 7, 6: Module status module in PROFIBUS slot 52
x + 17	Bit 1, 0: Module status module in PROFIBUS slot 53 Bit 3, 2: Module status module in PROFIBUS slot 54 Bit 5, 4: Module status module in PROFIBUS slot 55 Bit 7, 6: Module status module in PROFIBUS slot 56
x + 18	Bit 1, 0: Module status module in PROFIBUS slot 57 Bit 3, 2: Module status module in PROFIBUS slot 58 Bit 5, 4: Module status module in PROFIBUS slot 59 Bit 7, 6: Module status module in PROFIBUS slot 60
x + 19	Bit 1, 0: Module status module in PROFIBUS slot 61 Bit 3, 2: Module status module in PROFIBUS slot 62 Bit 5, 4: Module status module in PROFIBUS slot 63 Bit 7, 6: Module status module in PROFIBUS slot 64

6.11.4 Channel-related diagnostics

The channel-related diagnostics serves to obtain information about channel errors inside a module. For using the channel-related diagnostics, the diagnostic alarm must be enabled for each module via parameterisation. The channel-related diagnostics can be activated via parameterisation.

Note!

The maximum number of channel-related diagnostic messages is limited by the max. total length of 122 bytes. By deactivating other diagnostic areas, you can enable these areas for further channel-related diagnostic messages. 3 bytes are always used per channel.

Diagnostic data for one channel	
Byte	**Function**
x	Bit 5 ... 0: • 0b000000 ... 0b111111: ID number of the module that provides the channel-related diagnostics • PROFIBUS slot 1 has the ID number ’0’ ... • PROFIBUS slot 64 has the ID number ’63’ Bit 7, 6: • 0b10 (fix): Code for the channel-related diagnostics
x + 1	Bit 5 ... 0: • 0b000000 .... 0b111111: Number of the channel or channel group that provides the diagnostics Bit 7, 6: Module type • 0b01: Input module • 0b10: Output module • 0b11: Input/output module
x + 2	Bit 4 ... 0: **Error type according to PROFIBUS standard** • 0b00001: Short circuit • 0b00010: Undervoltage (supply voltage) • 0b00011: Overvoltage (supply voltage) • 0b00100: Output module is overloaded • 0b00101: Overtemperature of output module • 0b00110: Cable break of the sensor or actuator • 0b00111: Upper limit value exceeded • 0b01000: Lower limit value exceeded • 0b01001: Error (load voltage at the output, encoder supply, hardware error of the module) Bit 4 ... 0: **Manufacturer-specific error type** • 0b10000: Parameterisation error • 0b10001: Module-specific error • 0b10010: Defective fuse • 0b10100: Earthing error • 0b10101: Reference channel error • 0b10110: Process alarm lost • 0b11001: Safety-oriented switch-off • 0b11010: External error • 0b11010: Non-specific error Bit 7 ... 5: Channel type • 0b001: Bit • 0b010: 2 bits • 0b011: 4 bits • 0b100: Byte • 0b101: Word • 0b110: 2 words

6.11.5 Alarms

The alarm part of the slave diagnostic message informs about the alarm type and the cause that has triggered an alarm. The alarm part consists of maximally 24 bytes. Maximally 1 alarm can be reported per slave diagnostic message. The alarm part is always added as last part to the diagnostic message if it is activated in the parameterisation.

Structure of the alarm part, depending on the alarm type	
Byte	**Function**
x ... x + 3	**Alarm status** Contains information on the alarm type.
x + 4 ... x + 20	**Diagnosealarm** The 20 bytes correspond to the data set 1 of the CPU diagnostics.
x + 4 ... x + 7	**Process alarm** The 4 bytes are module-specific and are described in the respective module chapters.

Alarm status

If there is a diagnostic event for channel/channel group of a module, a module error may exist in addition to a channel error. In this case, an entry is made even you have not enabled the diagnostics for channel/channel group 0 of the module.

Structure of alarm status – byte x ... x + 3	
Byte	**Function**
x	Bit 5 ... 0: • 0b010100: Length of the alarm part incl. byte x Bit 7 ... 6: • 0b00 (fix): Code for device-related diagnostics
x + 1	Bit 6 ... 0: Alarm type • 0b0000001: Diagnostic alarm • 0b0000010: Process alarm Bit 7: Code for alarm
x + 2	Bit 7 ... 0: • 0b00000000 ... 0b00111111: ID number of the module that provides the alarm • PROFIBUS slot 1 has the ID number ’0’ ... • PROFIBUS slot 64 has the ID number ’63’
x + 3	Bit 1, 0: Alarm type • 0b00: Process alarm • 0b01: Diagnostic alarm incoming • 0b10: Diagnostic alarm outgoing • 0b11: Reserved Bit 2: • 0 (fixed) Bit 7 ... 3: Alarm sequence number 0 ... 31 • 0b00000 ... 0b11111

Diagnosealarm

Structure of diagnostic alarm – byte x + 4 ... x + 20	
Byte	**Function**
x + 4	• Bit 0: Module fault, i.e. an error has been detected • Bit 1: Internal error in the module • Bit 2: External error (module is not accessible anymore) • Bit 3: Channel error in the module • Bit 4: External supply voltage is missing • Bit 5, 6: Reserved • Bit 7: Parameterisation error
x + 5	Bit 3 ... 0: Module class • 0b1111: Digital module • 0b0101: Analog module • 0b1000: FM • 0b0111: ETS, CP Bit 4: channel information available Bit 7 ... 5: • 0 (fixed)
x + 6	see module description
x + 7	• Bit 5 ... 0: Reserved • Bit 6: Process alarm lost • Bit 7: Reserved
x + 8	Channel type • 0x70: Module with digital inputs • 0x71: Module with analog inputs • 0x72: Module with digital outputs • 0x73: Module with analog outputs • 0x74: Module with analog inputs and outputs • 0x76: Counter
x + 9	Number of diagnostic bits per channel
x + 10	Number of channels per module
x + 11	Position (channel) of the diagnostic event
x + 12	Diagnostic event for channel/channel group 0 For assignment, see module description
...	...
x + 19	Diagnostic event for channel/channel group 7 For assignment, see module description
x + 20	µs ticker (4 bytes) Value of the µs ticker when the diagnostic alarm occurs

Process alarm

More information on the diagnostic data of a process alarm (bytes x + 4 to x + 7) can be found in the respective module description.

7 EtherCAT communication

Note!

Bus coupler module EPM-S130 (EtherCAT)

Only **I/O compound modules EPM-Sxxx from hardware version 1B** are supported.

7.1 About EtherCAT

EtherCAT (**Ether**net for **C**ontroller and **A**utomation **T**echnology) is an Ethernet-based fieldbus system which fulfills the user profile for the part of industrial real time systems. Unlike the classic Ethernet communication, the I/O data is exchanged in full duplex operation with 100 Mbps. This delays the telegrams in the μs range.

The EtherCAT protocol optimised for process data is directly transported in the Ethernet frame. This can consist of several subframes which serve as a memory area each of the process image.

As each EtherCAT gateway is assigned a clear address (MAC address), the data-related sequence is independent of the physical sequence of the EtherCAT gateways in the network.

7.1.1 EtherCAT frame

EtherCAT frames have the following structure:

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

Ethernet header

The Ethernet header contains the following information:

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

Ethernet data

The Ethernet data contain the following information:

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

FCS

- Checksum of the EtherCAT frame

7.1.2 EtherCAT datagrams

EtherCAT datagrams have the following structure:

header	Data	WKC
10 bits	Max. 1486 bytes 2 bytes WKC = Working Counter	

7.1.3 EtherCAT state machine

Each fieldbus node is lead through a status machine by the master. The changes of the bus states are displayed in the following illustration.

Status	Description
Init	• No communication on the "Application Layer" • The master can access the "DL-Information Register".
Pre-Operational	• Mailbox communication on the "Application Layer" • No process data communication
Safe-operational	• Mailbox communication on the "Application Layer" • Process data communication (only the inputs are evaluated, the outputs are in the "safe" status.)
Operational	• Inputs and outputs are evaluated.

7.2 Transmitting process data

EtherCAT transfers parameter data and process data between the master and the slaves, which, depending on their time-critical behaviour, are divided into corresponding communication channels.

The process data are transmitted by means of "datagrams" via the process data channel.

- The process data controls the I/O channels.
- The transmission of the process data is time-critical.
- Process data are transmitted cyclically between the host system and the I/O system (permanent exchange of current input and output data).
- The master can directly access the process data. In the PLC for instance, the data are directly stored in the I/O area.
- Process data are not stored in the I/O system.
- Process data are e.g. input and output data of the I/O system.

Note!

The process data size of the I/O area is described by XML device description files.

Import the following two files via an EtherCAT configurator for this purpose:
- EtherCAT bus coupler module: Lenze_IOSystem1000_EPM_S130.xml
- I/O compound module: Lenze_IOSystem1000_EPM_S130_Modules.xml

7.2.1 Access to the I/O area

SDO access can be used for read-only access to the object directory's input and output data.

Input data

When accessing the input area of an I/O compound module, addressing takes place via index I-6000 + EtherCAT slot no. Subindices give you access to the corresponding input data. The relevant module description contains the subindex assignment.

Index	Subindex	Name	Type	Attr.	Default	Meaning
I-6000 ... I-603F	0x00	Input data	Unsigned8	ro		Number of input data subindexes for the corresponding EtherCAT slot no.
	0x01 0x02 ...			ro ro		Input data (see module description)

Output data

During read-only access to the output area of an I/O compound module, addressing takes place via index I-7000 + EtherCAT slot no. Subindices give you read-only access to the corresponding output data. The relevant module description contains the subindex assignment.

Index	Subindex	Name	Type	Attr.	Default	Meaning
I-7000 ... I-703F	0x00	Output data	Unsigned8	ro		Number of output data subindexes for the corresponding EtherCAT slot no.
	0x01 0x02 ...			ro ro		Output data (see module description)

7.3 Transmitting parameter data

Note!

Parameter data can be transferred from the EtherCAT master to the EtherCAT bus coupler module EPM-S130 (slave). Once the EtherCAT master has been started, the data is transferred to the bus coupler module where the settings are accepted after the change from "Pre-operational" operating mode to "Operational".

EtherCAT transfers parameter data and process data between the master and the bus coupler modules, which, depending on their time-critical behaviour, are divided into corresponding communication channels.

Parameter data (SDOs, service data objects) are transmitted via the SDO channel.

- Access to all indexes with CoE (CAN over EtherCAT) is enabled via the SDO channel.
- The transfer of parameter data is not normally time-critical.
- Parameter data corresponds to the indexes of the index list for the EtherCAT bus coupler module

Establishment of connection between the master and slave

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

Acyclic data transfer

Parameters ...

- are values which are stored under an index in the Lenze I/O system.
- are, for instance, used for one-off plant settings or a change of material in a machine.
- are transmitted with a low priority.

7.4 General function of the parameter setting

Parameterise I/O compound modules using SDO transfer.

Addressing takes place via index I-3100 + EtherCAT slot no. Subindices give you access to the corresponding parameters. The relevant module description contains the subindex assignment.

Index	Subindex	Name	Type	Attr.	Meaning
I-3100 ... I-313F	0x00	Parameter	Unsigned8	ro	Number of input data subindexes for the corresponding EtherCAT slot no.
	0x01 0x02 ...	Param1 Param2		ro/rw ro/rw	Parameter data (see module description)

If the module is parameterisable, the following applies:

- Index I-3100: Access to EtherCAT slot no. 1
- Index I-3101: Access to EtherCAT slot no. 2
- ...
- Index I-313F: Access to EtherCAT slot no. 64

The following example shows access to the parameters of the module in slot 4 via index I-3103.

Phy. slot	1	2	3	4
Module	DI	DI	DO	AI
Index	0x3100*	03101*	0x3102*	0x3103
EtherCAT slot no.	1	-	2	3

* This entry is not executed because the module is not parameterisable.

7.5 Parameterising analog I/Os

7.5.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400

Parameter data

Subindex	Name	Description/value	Lenze
01	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
02	Function channel 2		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

7.5.2 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401

Parameter data

Subindex	Name	Description/value	Lenze
01	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
02	Function channel 2		0x20
03	Function channel 3		0x20
04	Function channel 4		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

7.5.3 2 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S402

Parameter data

Subindex	Name	Description/value	Lenze
01	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
02	Function channel 2		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

7.5.4 4 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S403

Parameter data

Subindex	Name	Description/value	Lenze
01	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
02	Function channel 2		0x31
03	Function channel 3		0x31
04	Function channel 4		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

7.5.5 2 analog inputs -10 ... +10 V (16 bits) - EPM-S406

Parameter data

Subindex	Name	Description/value	Lenze
01	Diagnostics	Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
02	Reserved	0	
03	Limit value monitoring	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 (0 = deactivated; 1 = activated) Bits 7 ... 2: reserved	0x00
04	Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
05	Function channel 1	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
06	Reserved	0	
07	Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
08	Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
09	Function channel 2	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
0A	Reserved	0	
0B	Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
0C	Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
-10 ... +10 V (0x12)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-5	-13824	CA00		
	-10	-27648	9400		
	-11.76	-32512	8100	Underflow	
-10 ... +10 V (0x22)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-5	-8192	E000		
	-10	-16384	C000		
	-12.5	-20480	B000	Underflow	
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-1.76	-4864	ED00	Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-2	-3277	F333	Underflow	

7.5.6 2 analog inputs 0/4 ... 20 mA (16 bits) - EPM-S408

Parameter data

Subindex	Name	Description/value	Lenze
01	Diagnostics	Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
02	Reserved	0	
03	Limit value monitoring	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 (0 = deactivated; 1 = activated) Bits 7 ... 2: reserved	0x00
04	Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
05	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
06	Reserved	0	
07	Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
08	Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
09	Function channel 2	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
0A	Reserved	0	
0B	Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
0C	Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

7.5.7 2 analog outputs 0 ... 10 V (12 bits) - EPM-S500

Parameter data

Subindex	Name	Description/value	Lenze
01	Reserved	0	
02	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
03	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
04	Function channel 2		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Output area (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

7.5.8 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501

Parameter data

Subindex	Name	Description/value	Lenze
01	Reserved	0	
02	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
03	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
04	Function channel 2		0x20
05	Function channel 3		0x20
06	Function channel 4		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Output area (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

7.5.9 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502

Parameter data

Subindex	Name	Description/value	Lenze
01	Reserved	0	
02	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
03	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
04	Function channel 2		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

7.5.10 4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503

Parameter data

Subindex	Name	Description/value	Lenze
01	Reserved	0	
02	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
03	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
04	Function channel 2		0x31
05	Function channel 3		0x31
06	Function channel 4		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

7.6 Parameterising temperature measurement

7.6.1 4(2) analog input for resistance measurement - EPM-S404

Note!

Use parameter setting to deactivate unused inputs.

If thermal detectors are connected in a 3 or 4 conductor setup, channels 3 and/or 4 must be deactivated.

▸ 2-, 3-, 4-wire conductor measurement (📖 26)

The module does not provide any auxiliary supply for sensors.

Parameter data

Subindex	Name	Description/value	Lenze
01	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
02	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) : Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
03	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) : Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
04	Reserved		
05	Temperature system	Bit 0, 1: 0b00 = °C; 0b01= °F; 0b10 = K Bit 2 ... 7: Reserved	0x00
06	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02
Channel 1			

Subindex	Name	Description/value	Lenze
07	Function channel 1	Thermal detector: 80 (0x50): PT100 2-wire conductor -200°C ... +850°C / -2000 ... +8500 81 (0x51): PT1000 2-wire conductor -200°C ... +850°C / -2000 ... +8500 82 (0x52): Ni100 2-wire conductor -60°C ... +250°C / -600 ... +2500 83 (0x53): Ni1000 2-wire conductor -60°C ... +250°C / -600 ... +2500 88 (0x58): PT100 3-wire conductor -200°C ... +850°C / -2000 ... +8500 89 (0x59): PT1000 3-wire conductor -200°C ... +850°C / -2000 ... +8500 90 (0x5A): Ni100 3-wire conductor -60°C ... +250°C / -600 ... +2500 91 (0x5B): Ni1000 3-wire conductor -60°C ... +250°C / -600 ... +2500 96 (0x60): PT100 4-wire conductor -200°C ... +850°C / -2000 ... +8500 97 (0x61): PT1000 4-wire conductor -200°C ... +850°C / -2000 ... +8500 98 (0x62): Ni100 4-wire conductor -60°C ... +250°C / -600 ... +2500 99 (0x63): Ni1000 4-wire conductor -60°C ... +250°C / -600 ... +2500 Resistor: 112 (0x70): R 60-Ω 2-wire conductor 0.00 ... +60.00 / 0 ... +32767 113 (0x71): R 600-Ω 2-wire conductor 0.00 ... +600.00 / 0 ... +32767 114 (0x72): R 3000-Ω 2-wire conductor 0.00 ... +3000.00 / 0 ... +32767 115 (0x73): R 6000-Ω 2-wire conductor 0.00 ... +6000.00 / 0 ... +32767 128 (0x80): R 60-Ω 4-wire conductor 0.00 ... +60.00 / 0 ... +32767 129 (0x81): R 600-Ω 4-wire conductor 0.00 ... +600.00 / 0 ... +32767 130 (0x82): R 3000-Ω 4-wire conductor 0.00 ... +3000.00 / 0 ... +32767 144 (0x90): R 60-Ω 2-wire conductor 0.00 ... +60.00 / 0 ... +6000 145 (0x91): R 600-Ω 2-wire conductor 0.00 ... +600.00 / 0 ... +6000 146 (0x92): R 3000-Ω 2-wire conductor 0.00 ... +3000.00 / 0 ... +30000 160 (0xA0): R 60-Ω 2-wire conductor 0.00 ... +60.00 / 0 ... +6000 161 (0xA1): R 600-Ω 2-wire conductor 0.00 ... +600.00 / 0 ... +6000 162 (0xA2): R 3000-Ω 2-wire conductor 0.00 ... +3000.00 / 0 ... +30000 255 (0xFF): Channel deactivated	0x50

Subindex	Name	Description/value	Lenze
08	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At 50 Hz: 44.1 ms/channel 16 bits; at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x00
09	Upper limit value channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. If your measured value is outside a limit value and you have activated the limit value monitoring, a process alarm is triggered.	0x7FFF
0A	Lower limit value channel 1		0x8000
Channel 2			
0B	Function channel 2	See channel 1	0x50
0C	Conversion time channel 2	See channel 1	0x00
0D	Upper limit value channel 2	See channel 1	0x7FFF
0E	Lower limit value channel 2		0x8000
Channel 3 (for two-wire conductor connections only)			
0F	Function channel 3	See channel 1	0x50
10	Conversion time channel 3	See channel 1	0x00
11	Upper limit value channel 3	See channel 1	0x7FFF
12	Lower limit value channel 3		0x8000
Channel 4 (for two-wire conductor connections only)			
13	Function channel 4	See channel 1	0x50
14	Conversion time channel 4	See channel 1	0x00
15	Upper limit value channel 4	See channel 1	0x7FFF
16	Lower limit value channel 4		0x8000

Measuring range

Measuring range (Fct. no.)	Measured value	Signal range	Range
2-wire: PT100 (0x50)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: PT1000 (0x51)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: NI100 (0x52)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: NI1000 (0x53)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: PT100 (0x58)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: PT1000 (0x59)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: NI100 (0x5A)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: NI1000 (0x5B)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: PT100 (0x60)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
4-wire: PT1000 (0x61)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
4-wire: NI100 (0x62)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: NI1000 (0x63)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: 0 ... 60 Ω (0x70)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x71)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
2-wire: 0 ... 3000 Ω (0x72)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0x78)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x79)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x7A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0x80)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0x81)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0x82)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0x90)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x91)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0x92)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0x98)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x99)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x9A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xA0)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
4-wire: 0 ... 600 Ω (0xA1)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0xA2)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0xD0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0xD1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0xD2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0xD8)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0xD9)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0xDA)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xE0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0xE1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0xE2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow

7.6.2 2 analog inputs for thermocouple measurement - EPM-S405

Parameter data

Subindex	Name	Description/value	Lenze
01	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
02	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
03	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
04	Reserved	0	
05	Temperature system	Bit 0, 1: 0b00 = °C; 0b10 = °F; 0b11 = K Bit 2 ... 7: Reserved	0x00
06	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02
Channel 1			
07	Function channel 1	External temperature compensation: 176 (0x60): type J, -210.0 ... +1200.0 °C / -2100 ... +12000 177 (0x61): type K, -270.0 ... +1372.0 °C / -2700 ... +13720 178 (0x62): type N -270.0 ... +1300.0 °C / -2700 ... +13000 179 (0x63): type R, -50.0 ... +1769.0 °C / -500 ... +17690 180 (0x64): type S, -50.0 ... +1769.0 °C / -500 ... +17690 181 (0x65): type T, -270.0 ... +400.0 °C / -2700 ... +4000 182 (0x66): type B, 0.0 ... +1820.0 °C / 0 ... +18200 183 (0x67): type C, 0.0 ... +2315.0 °C / 0 ... +23150 184 (0x68): type E, -270.0 ... +1000.0 °C / -2700 ... +10000 185 (0x69): type L, -200.0 ... +900.0 °C / -2000 ... +9000 Internal temperature compensation: 192 (0xC0): type J, -210.0 ... +1200.0 °C / -2100 ... +12000 193 (0xC1): type K, -270.0 ... +1372.0 °C / -2700 ... +13720 194 (0xC2): type N -270.0 ... +1300.0 °C / -2700 ... +13000 195 (0xC3): type R, -50.0 ... +1769.0 °C / -500 ... +17690 196 (0xC4): type S, -50.0 ... +1769.0 °C / -500 ... +17690 197 (0xC5): type T, -270.0 ... +400.0 °C / -2700 ... +4000 198 (0xC6): type B, 0.0 ... +1820.0 °C / 0 ... +18200 199 (0xC7): type C, 0.0 ... +2315.0 °C / 0 ... +23150 200 (0xC8): type E, -270.0 ... +1000.0 °C / -2700 ... +10000 201 (0xC9): type L, -200.0 ... +900.0 °C / -2000 ... +9000 255 (0xFF): Channel deactivated	0xC1

Subindex	Name	Description/value	Lenze
08	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/ channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/ channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/ channel 16 bits 3 (0x03): At Hz: 44.1 ms/channel 16 bits at 60 Hz: 37.2 ms/ channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/ channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/ channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/ channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/ channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/ channel 13 bits	0x02
09	Upper limit value channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. If your measured value is outside a limit value and you have activated the limit value monitoring, a process alarm is triggered.	0x7FFF
0A	Lower limit value channel 1		0x8000
Channel 2			
0B	Function channel 2	See channel 1	0xC1
0C	Conversion time channel 2	See channel 1	0x02
0D	Upper limit value channel 2	See channel 1	0x7FFF
0E	Lower limit value channel 2		0x8000

Measuring range

Measuring range (Fct. no.)	Measured value [°C]	[°F]	[K]	Range
Type J: -210 ... +1200 °C -346 ... 2192 °F 63.2 ... 1473.2 K (0xB0: ext. comp. 0 °C) (0xC0: int. comp. 0 °C)	+14500	26420	17232	Overflow
	-2100 ... +12000	-3460 ... +21920	632 ... 14732	Nominal range
	-	-	-	Underflow
Type K: -210 ... +1372 °C -454 ... 2501.6 °F 0 ... 1645.2 K (0xB1: ext. comp. 0 °C) (0xC1: int. comp. 0 °C)	+16220	29516	18952	Overflow
	-2700 ... +13720	-4540 ... 25016	0 ... 16452	Nominal range
	-	-	-	Underflow

Measuring range (Fct. no.)	Measured value [°C]	[°F]	[K]	Range
Type N: -270 ... +1300 °C -454 ... 2372 °F 0 ... 1573.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+15500	28220	18232	Overflow
	-2700 ... +13000	-4540 ... 23720	0 ... 15732	Nominal range
	-	-	-	Underflow
Type R: -50 ... +1769 °C -58 ... 3216.2 °F 223.2 ... 2042.2 K (0xB3: ext. comp. 0 °C) (0xC3: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type S: -50 ... +1769 °C -58 ... 3216.2 °F 223.2 ... 2042.2 K (0xB4: ext. comp. 0 °C) (0xC4: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type T: -270 ... +440 °C -454 ... 752 °F 3.2 ... 673.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+5400	10040	8132	Overflow
	-2700 ... +4000	-4540 ... 7520	32 ... 6732	Nominal range
	-	-	-	Underflow
Type B: 0 ... +1820 °C 32 ... 2786.5 °F 273.2 ... 2093.2 K (0xB6: ext. comp. 0 °C) (0xC6: int. comp. 0 °C)	+20700	32766	23432	Overflow
	0 ... +18200	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow
Type C: 0 ... +2315 °C 32 ... 2786.5 °F 273.2 ... 2093.2 K (0xB7: ext. comp. 0 °C) (0xC7: int. comp. 0 °C)	+25000	32766	23432	Overflow
	0 ... +23150	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow
Type E: -270 ... +1000 °C -454 ... 1832 °F 0 ... 1273.2 K (0xB8: ext. comp. 0 °C) (0xC8: int. comp. 0 °C)	+12000	21920	14732	Overflow
	-2700 ... +10000	-4540 ... 18320	0 ... 12732	Nominal range
	-	-	-	Underflow
Type L: -200 ... +900 °C -328 ... 1652 °F 73.2 ... 1173.2 K (0xB9: ext. comp. 0 °C) (0xC9: int. comp. 0 °C)	+11500	21020	14232	Overflow
	-2000 ... +9000	-3280 ... 16520	732 ... 11732	Nominal range
	-	-	-	Underflow

7.7 Parameterising the counter

7.7.1 1 counter 32 bits, 24 V DC - EPM-S600

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter a differentiation between the internal gate (I-gate), hardware gate (HW gate), and software gate (SW gate) is made. • The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). • The SW gate is controlled via your user program (status word in the output area). • The HW gate is controlled via the digital gate input. The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Latch function	If a positive edge occurs at the latch input, the current count value is stored in the latch register. The latch register is accessed via the input area. After a STOP-RUN transition, latch is always 0.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.

Additional functions	Description
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Further information: ▶ Product description (📖 15)

Parameter data

<table>
<tr><th>Subindex</th><th>Name</th><th>Description/value</th><th>Lenze</th></tr>
<tr><td>01</td><td>Diagnostics*</td><td>A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing.
Bit 5 ... 0: Reserved
Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled)
Bit 7: Reserved</td><td>0x00</td></tr>
<tr><td>02</td><td>Input frequency Track A</td><td rowspan="4">Filters for instance serve to filter signal peaks in the case of an unclean input signal.
0 (0x00): 500 kHz
1 (0x01): 300 kHz
2 (0x02): 100 kHz
3 (0x03): 60 kHz
4 (0x04): 30 kHz
6 (0x06): 10 kHz
7 (0x07): 5 kHz
8 (0x08): 2 kHz
9 (0x09): 1 kHz
Other values are not permissible!</td><td>0x02</td></tr>
<tr><td>03</td><td>Input frequency Track B</td><td>0x02</td></tr>
<tr><td>04</td><td>Input frequency Latch</td><td>0x02</td></tr>
<tr><td>05</td><td>Input frequency Gate</td><td>0x02</td></tr>
<tr><td>06</td><td>Input frequency Reset</td><td></td><td>0x00</td></tr>
<tr><td>07</td><td>Reserved</td><td></td><td></td></tr>
<tr><td>08</td><td>Alarm response*</td><td>Setting activates process alarm
Bit 0: Proc. alarm HW gate open
Bit 1: Proc. alarm HW gate closed
Bit 2: Proc. alarm overflow
Bit 3: Proc. alarm underflow
Bit 4: Proc. alarm comparison value
Bit 5: Proc. alarm final value
Bit 6: Proc. alarm latch value
Bit 7: Reserved</td><td>0x80</td></tr>
</table>

Subindex	Name	Description/value	Lenze
09	Counter function*	Bit 5 … 0: 0b000000 = continuous counting 0b000001 = single counting, main counting direction is forward 0b000010 = single counting, main counting direction is backward 0b000100 = single counting, no main counting direction 0b001000 = periodic counting, main counting direction is forward 0b010000 = periodic counting, main counting direction is backward 0b100000 = periodic counting, no main counting direction Bit 7 … 6: Reserved	0x40
0A	Comparator*	Bit 2 … 0: output switches (… if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 … 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00
0B	Signal evaluation*	Bit 2 … 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 … 3: Hardware gate (HW gate) • 0b000 = deactivated (counter starts by setting SW gate) • 0b001 = activated (HIGH level at gate activates the HW gate. Counter starts if HW and SW gate are set.) Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	
0C	Final value	Upper limitation of the counting range	0x00
0D	Start value	Lower limitation of the counting range	0x00
0E	Hysteresis	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
0F	Pulse	The pulse duration indicates for how long the output is to be set if the parameterised comparison criterion is reached or exceeded. The pulse duration can be specified in steps of 2.048 ms between 0 and 522.24 ms. If the pulse duration is = 0, the output is set until the comparison condition is no longer met.	0x00

* The values will only be accepted after the state has changed from "Pre-Operational" to "Operational".

7.7.2 2 counters 32 bits, 24 V DC - EPM-S601

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Further information: ▶ Product description (📖 15)

Parameter data

Subindex	Name	Description/value	Lenze
01	Diagnostics*	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 … 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
02	Input frequency Counter 1, track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
03	Input frequency Counter 1, track B		0x02
04	Input frequency Counter 2, track A		0x02
05	Input frequency Counter 2, track B		0x02
06	Alarm response counter 1*	Setting activates process alarm Bit 1 … 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 … 6: Reserved	0x00
07	Counter function counter 1*	Bit 5 … 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 … 6: Reserved	0x00
08	Comparator counter 1*	Bit 2 … 0: Comparison bit is set (… if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 … 4: Reserved	0x00

Subindex	Name	Description/value	Lenze
09	Signal evaluation counter 1*	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
0A	Set value counter 1	When a set value is given, the counter can be loaded with the set value. With an edge 0-1 at COUNTERVAL_SET in the control word, the set value is accepted in the counter.	0x00
0B	Final value counter 1	Upper limitation of the counting range	0x00
0C	Loading value counter 1	Lower limitation of the counting range	0x00
0D	Hysteresis counter 1	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
0E	Reserved		
0F	Alarm response counter 2*	See counter 1	0x00
10	Counter function counter 2*	See counter 1	0x00
11	Comparator counter 2*	See counter 1	0x00
12	Signal evaluation counter 2*	See counter 1	0x00
13	Set value counter 2	See counter 1	0x00
14	Final value counter 2	See counter 1	0x00
15	Loading value counter 2	See counter 1	0x00
16	Hysteresis counter 2	See counter 1	0x00

* The values will only be accepted after the state has changed from "Pre-Operational" to "Operational".

7.7.3 1 counter 32 bits, 5 V DC - EPM-S602

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Further information: ▶ Product description (📖 15)

Parameter data

Subindex	Name	Description/value	Lenze
01	Diagnostics*	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
02	Input frequency Track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
03	Input frequency Track B		0x02
04	Input frequency Reset		0x02
05	Reserved		
06	Alarm response*	Setting activates process alarm Bit 1 ... 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 ... 6: Reserved	0x00
07	Counter function*	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 ... 6: Reserved	0x00
08	Comparator*	Bit 2 ... 0: Comparison bit is set (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 ... 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00

Subindex	Name	Description/value	Lenze
09	Signal evaluation*	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
0A	Final value	Upper limitation of the counting range	0x00
0B	Start value	Lower limitation of the counting range	0x00
0C	Hysteresis		0x00

* The values will only be accepted after the state has changed from "Pre-Operational" to "Operational".

7.7.4 2 counters 32 bits, 24 V DC - EPM-S603

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from 0 to the counting limit, then skips to the opposite counting limit and continues to count from there.

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area).

Further information: ▶ Product description (📖 15)

Parameter data

Subindex	Name	Description/value	Lenze
01	Input frequency Counter 1, track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
02	Input frequency Counter 1, track B		0x02
03	Input frequency Counter 2, track A		0x02
04	Input frequency Counter 2, track B		0x02
05	Counting direction counter 1, track B	Bit 2 ... 0: Reserved Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
06	Signal evaluation counter 1*	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 7 ... 3: Reserved	0x00

Subindex	Name	Description/value	Lenze
07	Counting direction counter 1, track B	Bit 2 ... 0: Reserved Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
08	Signal evaluation counter 1*	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 7 ... 3: Reserved	0x00

* The values will only be accepted after the state has changed from "Pre-Operational" to "Operational".

7.8 Parameterising the encoder evaluation

7.8.1 SSI - EPM-S604

Parameter data

Subindex	Name	Description/value	Lenze
01	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
02	Idle time	The dead time, also called tbs (time between sends), defines the waiting time between two encoder values to be observed by the module so that the encoder is able to process its value. This data can be found in the data sheet for your encoder. HIGH LOW 0x00 0x30: 1 µs 0x00 0x60: 2 µs 0x00 0xC0: 4 µs 0x01 0x80: 8 µs 0x03 0x00: 16 µs 0x06 0x00: 32 µs 0x09 0x00: 48 µs 0x0C 0x00: 64 µs	0x0C00
03	Baud rate	In "monitoring operation" operating mode, the baud rate is irrelevant. Enter the baud rate here. This corresponds to the clock frequency via which the encoder connected communicates. Information on this can be found in the data sheet for your encoder. HIGH LOW 0x00 0x18: 2 MHz 0x00 0x20: 1.5 MHz 0x00 0x30: 1 MHz 0x00 0x60: 500 kHz 0x00 0xC0: 250 kHz 0x01 0x80: 125 kHz	0x0180
04	Reserved		
05	Standardisation	Depending on the encoder, further bits are transmitted in addition to the encoder value. Scaling serves to determine how many bits post-positioned to the encoder value will be removed by shifting the encoder value to the right. The encoder value is scaled by the module only after a Gray-binary conversion. More information can be found in the data sheet for your encoder. Value range: 0x00 ... 0x0F = 0 bit ... 15 bits	0x00

Subindex	Name	Description/value	Lenze
06	Bit length of encoder data	Enter the bit length of the encoder data here. Depending on the encoder, the encoder data consist of the current encoder value with subsequent bits. The total length has to be specified here. More information on this can be found in the data sheet for your encoder. 7 (0x07) = "8 bits" 8 (0x08) = "9 bits" 9 (0x09) = "10 bits" 10 (0x0A) = "11 bits" 11 (0x0B) = "12 bits" 12 (0x0C) = "13 bits" 13 (0x0D) = "14 bits" 14 0x0E) = "15 bits" 15 (0x0F) = "16 bits" 16 (0x10) = "17 bits" 17 (0x11) = "18 bits" 18 (0x12) = "19 bits" 19 (0x13) = "20 bits" 20 (0x14) = "21 bits" 21 (0x15) = "22 bits" 22 (0x16) = "23 bits" 23 (0x17) = "24 bits" 24 (0x18) = "25 bits" 25 (0x19) = "26 bits" 26 (0x1A) = "27 bits" 27 (0x1B) = "28 bits" 28 (0x1C) = "29 bits" 29 (0x1D) = "30 bits" 30 (0x1E) = "31 bits" 31 (0x1F) = "32 bits"	0x18

Subindex	Name	Description/value	Lenze
07		Bit 1 ... 0: Ready for operation During "Monitoring operation" the module serves to monitor the data exchange between an SSI master and an SSI encoder. It receives the cycle by the master and the data flow by the SSI encoder. In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder. 0b01 = monitoring operation 0b10 = master mode Bit 2: Shifting direction Specify the orientation of the encoder data here. More information can be found in the data sheet for your encoder. Usually, the SSI encoder uses "MSB first". 0 = LSB first (LSB is transmitted first) 1 = MSB first (MSB is transmitted first) Bit 3: Edge clock signal Here you can specify the edge type of the clock signal, in the case of which the encoder supplies data. Information on this can be found in the data sheet for your encoder. Usually the SSI encoders respond to rising edges. 0 = falling edge 1 = rising edge **Master mode:** Connect clock output signal (ClockOut) to the EPM-S604. **Monitoring mode:** Connect clock input signal (ClockIn) to the EPM-S604. Bit 4: Coding In the "Binary code" setting, the provided encoder value remains unchanged. In the "Gray code" setting, the Gray-coded value provided by the encoder is converted into a binary value. Only after this conversion, the received encoder value is scaled, if required. The Gray code is another form of representation of the binary code. It is based on the fact that two adjacent Gray numbers differ from each other in exactly one bit. If the Gray code is used, transmission errors can be easily detected, since adjacent characters must only differ from each other in one digit. Information on this can be found in the data sheet for your encoder. 0 = standard code 1 = Gray code Bit 7 ... 5: Reserved	0x1E
08	Reserved		
09	SSI function	By enabling the SSI function the module starts with the cycle output and the evaluation of the encoder data. In the "monitor operation" mode, the module starts with the encoder evaluation. 0 (0x00) = inhibited 1 (0x01) = enabled	0x00

7.9 Parameterising the time stamp

7.9.1 2 digital inputs with time stamp function - EPM-S207

Parameter data

Subindex	Name	Description/value	Lenze
01	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C(fix)
02	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)
03	Input delay DI 0	0x00 = 1 μs 0x02 = 3 μs 0x04 = 10 μs 0x07 = 86 μs 0x09 = 342 μs 0x0C = 273 μs No other values are permissible.	0x02
04	Input delay DI 1		0x02
05	Edge 0-1 at DI x	Time stamp entry on rising edge Bit 0: DI 0 (0: inhibit, 1 = enable) Bit 1: DI 1 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00
06	Edge 1-0 on DI x	Time stamp entry on falling edge Bit 0: DI 0 (0: inhibit, 1 = enable) Bit 1: DI 1 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00

7.9.2 2 digital outputs with time stamp function - EPM-S310

Parameter data

Subindex	Name	Description/value	Lenze
01	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C(fix)
02	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)

7.10 Parameterising technology modules

7.10.1 2 digital outputs with PWM functionality - EPM-S620

Parameter data

<table>
<tr><th>Subindex</th><th>Name</th><th>Description/value</th><th>Lenze</th></tr>
<tr><td>01</td><td>PWM 0: period</td><td rowspan="2">Set parameters here for the total time for pulse duration and pulse pause. The time should be selected as a factor for the 20.83 ns basis.
Values below 25 μs are ignored. If the pulse duration is higher or equal to the period, the DO output is set permanently.
Value range: 1200 ... 8388607 (25 μs ... approx. 175 ms)</td><td>0x1F40</td></tr>
<tr><td>02</td><td>PWM 1: period</td><td>0x1F40</td></tr>
</table>

7.10.2 RS232 interface - EPM-S640

Parameter data - ASCII protocol

Subindex	Name	Description/value	Lenze
0x01	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x02	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x03	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x04	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x05	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x01: ASCII	0x01
Option 1, drawing frame			
0x06	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			

Subindex	Name	Description/value	Lenze
0x07	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
0x08	ZNA (LOW byte)		0x00
Option 4 / 5, character delay time			
0x09	Character delay time (HIGH byte)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
0x0A	Character delay time (LOW byte)		0xFA
Option 6, number of receive buffers			
0x0B	Number of receive buffers	Defines the number of receive buffers. As long as only one receive buffer is used and is assigned, no more data can be received. If up to 250 receive buffers are connected, the received data can be redirected to a still free receive buffer. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7 ... 12, reserved			
0x0C ... 0x11	Reserved		0x00

Parameter data STX/ETX protocol

Subindex	Name	Description/value	Lenze
0x01	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x02	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x03	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x04	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x05	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame			
0x06	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			
0x07	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
0x08	ZNA (LOW byte)		0x00
Option 4 / 5, TMO			

Subindex	Name	Description/value	Lenze
0x09	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
0x0A	TMO (LOW byte)		0xFA
Option 6, number of start identifiers			
0x0B	Number of start identifiers	0x00: 1 start identifier (2. start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1			
0x0C	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2			
0x0D	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers			
0x0E	Number of end identifiers	0x00: 1 end identifier (2. end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1			
0x0F	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2			
0x10	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved			
0x11	Reserved		0x00

Parameter data 3964(R) protocol

Index/subindex	Name	Description/value	Lenze
0x01	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x02	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x03	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x04	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x05	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame			
0x06	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)			
0x07	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00

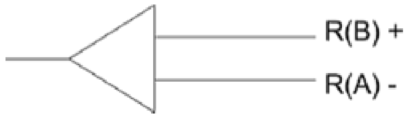
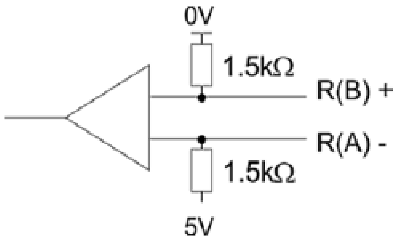
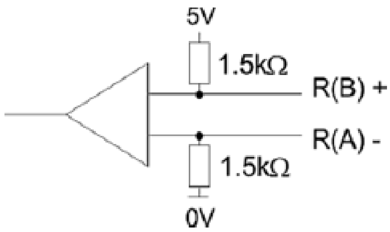
Index/subindex	Name	Description/value	Lenze
Option 3, character delay time (x 20 ms)			
0x08	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)			
0x09	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)			
0x0A	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repititons			
0x0B	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL			
0x0C	DBL	If the block wait time is exceeded, you can select the maximum number of repititions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority			
0x0D	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved			
0x0E ... 0x11	Reserved		0x00

7.10.3 RS422/RS485 interface - EPM-S650

Parameter data - ASCII protocol

Subindex	Name	Description/value	Lenze
0x01	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x02	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x03	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x04	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x05	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x01: ASCII	0x01
Option 1, drawing frame			
0x06	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			

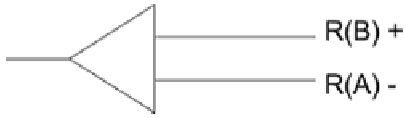
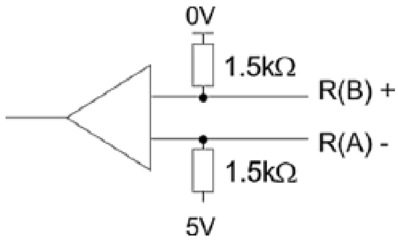
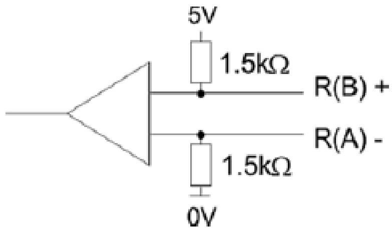
Subindex	Name	Description/value	Lenze
0x07	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
0x08	ZNA (LOW byte)		0x00
Option 4 / 5, character delay time			
0x09	Character delay time (HIGH byte)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
0x0A	Character delay time (LOW byte)		0xFA
Option 6, number of receive buffers			
0x0B	Number of receive buffers	Defines the number of receive buffers. As long as only one receive buffer is used and is assigned, no more data can be received. If up to 250 receive buffers are connected, the received data can be redirected to a still free receive buffer. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7 ... 12, reserved			
0x0C ... 0x11	Reserved		0x00
Option 13, operating mode			
0x12	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

Subindex	Name	Description/value	Lenze
Option 14, cable assignment			
0x13	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible. 	0x00

Parameter data STX/ETX protocol

Subindex	Name	Description/value	Lenze
0x01	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x02	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x03	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x04	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x05	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame			
0x06	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			
0x07	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
0x08	ZNA (LOW byte)		0x00
Option 4 / 5, TMO			

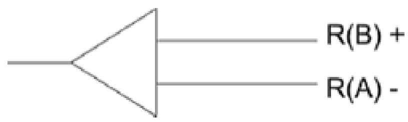
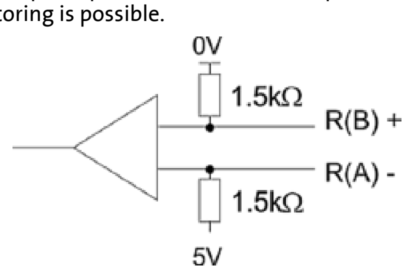
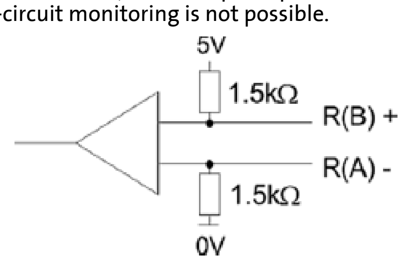
Subindex	Name	Description/value	Lenze
0x09	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
0x0A	TMO (LOW byte)		0xFA
Option 6, number of start identifiers			
0x0B	Number of start identifiers	0x00: 1 start identifier (2. start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1			
0x0C	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2			
0x0D	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers			
0x0E	Number of end identifiers	0x00: 1 end identifier (2. end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1			
0x0F	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2			
0x10	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved			
0x11	Reserved		0x00
Option 13, operating mode			
0x12	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

Subindex	Name	Description/value	Lenze
Option 14, cable assignment			
0x13	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible.  	0x00

Parameter data 3964(R) protocol

Index/subindex	Name	Description/value	Lenze
0x01	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x02	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x03	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x04	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x05	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame			
0x06	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)			
0x07	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00

Index/subindex	Name	Description/value	Lenze
Option 3, character delay time (x 20 ms)			
0x08	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)			
0x09	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)			
0x0A	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repititons			
0x0B	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL			
0x0C	DBL	If the block wait time is exceeded, you can select the maximum number of repititions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority			
0x0D	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved			
0x0E ... 0x11	Reserved		0x00
Option 13, operating mode			

Index/subindex	Name	Description/value	Lenze
0x12	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01
Option 14, cable assignment			
0x13	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible. 	0x00

7.11 Monitoring

In the event of an error, the digital and analog outputs are switched to the FALSE state or 0 V.

Exception: Digital outputs with time stamp functionality retain the last set value.

7.12 Diagnostics

Access to diagnostic data

I/O compound modules capable of alarms automatically transmit process alarm and/or diagnostic alarm data via the emergency telegram, provided that the alarm is activated by the parameter setting. You can however also request diagnostic data via SDO.

Alarm status

The alarm status contains a counter for the process alarm and a counter for the diagnostic alarm for alarm signalling. These counters are input data for the EtherCAT slave and are transferred along with the process data.

Index	Subindex	Name	Type	Attr.	Default	Meaning
I-F100	0x00	Interrupt status	Unsigned8	ro	2	
	0x01	Hardware interrupt counter	Unsigned32	ro	0x00000000	Counter for process alarm
	0x02	Diagnostic interrupt counter	Unsigned32	ro	0x00000000	Counter for diagnostic alarm

When auto-acknowledge is deactivated for the EtherCAT bus coupler module, the corresponding counter is set to 1 until you acknowledge it. To do this, write any value to subindex 0x06 using the index assigned.

When auto-acknowledge is activated for the EtherCAT bus coupler module, here you will find the number of process and/or diagnostic alarms which have been triggered since the last alarm reset. To reset the corresponding counter, write any value to subindex 0x06 using the index assigned.

The following index assignment applies:

- Writing to 0x06 of index I-5000: Resets counter for process alarms
- Writing to 0x06 of index I-5002: Resets counter for diagnostics alarm

7.12.1 Process alarm data

If the alarm status indicates a process alarm, index I-5000 provides access to the current process alarm data.

Index	Subindex	Name	Type	Attr.	Default	Meaning
0x5000	0x00	Hardware interrupt data	Unsigned8	ro	0x00	Current process alarm data
	0x01	Slot number	Unsigned8	ro	0x00	EtherCAT slot no. of module on which the alarm has occurred
	0x02	Diagnostic byte 1	Unsigned8	ro	0x00	Process alarm data (see tables below)
	0x03	Diagnostic byte 2	Unsigned8	ro	0x00	
	0x04	Diagnostic byte 3	Unsigned8	ro	0x00	
	0x05	Diagnostic byte 4	Unsigned8	ro	0x00	
	0x06	Acknowledge	Unsigned8	rw	0x00	Writing any value resets the diagnostic alarm counter and if necessary acknowledges the alarm

EPM-S404 - process alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Exceedance of limit value, channel 1 Bit 1: Limit value exceedance, channel 2 Bit 7 ... 2: 0 (fix)
2	Bit 0: Limit value underflow, channel 1 Bit 1: Limit value underflow, channel 2 Bit 7 ... 2: 0 (fix)
3/4	Ticker value at the time of the alarm

EPM-S405 - process alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Exceedance of limit value, channel 1 Bit 1: Limit value exceedance, channel 2 Bit 2: Limit value exceedance, channel 3 Bit 3: Limit value exceedance, channel 4 Bit 7 ... 4: 0 (fix)
2	Bit 0: Limit value underflow, channel 1 Bit 1: Limit value underflow, channel 2 Bit 2: Limit value underflow, channel 3 Bit 3: Limit value underflow, channel 4 Bit 7 ... 4: 0 (fix)
3/4	Ticker value at the time of the alarm

EPM-S406, EPM-S408 - process alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Exceedance of limit value, channel 1 Bit 1: Limit value exceedance, channel 2 Bit 7 ... 2: 0 (fix)
2	Bit 0: Limit value underflow, channel 1 Bit 1: Limit value underflow, channel 2 Bit 7 ... 2: 0 (fix)
3/4	μs ticker value at the time of the alarm The I/O compound module features an integrated 32-bit timer (μs-Ticker) which is started at switch-on and restarts at 0 after 232 - 1 μs. These two bytes represent the lower two bytes of the μs ticker (0 ... 216 - 1)

EPM-S600 - process alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bits 7 ... 5: 0 (fixed)
2	State of the inputs at the time of the alarm Bit 0: A/pulse Bit 1: B/direction Bit 2: Latch Bit 3: Hardware gate Bit 4: Reset Bits 7 ... 5: 0 (fixed)
3/4	Ticker value at the time of the alarm

EPM-S601, EPM-S602 - process alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: 0 Bit 1: 0 Bit 2: Counter 1, overflow/underflow/final value Bit 3: Counter 1, comparison value reached Bit 4: 0 Bit 5: 0 Bit 6: Counter 2, overflow/underflow/final value Bit 7: Counter 2, comparison value reached
2	State of the inputs at the time of the alarm Bit 0: Counter 1, A/pulse Bit 1: Counter 1, B/direction Bit 2: Counter 2, A/pulse Bit 3: Counter 2, B/direction Bit 7 ... 4: 0 (fix)
3/4	16 bit μs value at the time of the alarm

7.12.2 Diagnostic alarm data

Diagnostic data (bytes 1 ... 4)

If the alarm status indicates a diagnostic alarm, index I-5002 provides access to the current diagnostic alarm data.

Index	Subindex	Name	Type	Attr.	Default	Meaning
0x5002	0x00	Diagnostic data	Unsigned8	ro	6	Current diagnostic data
	0x01	Slot number	Unsigned8	ro	0x00	EtherCAT slot no. of module on which the alarm has occurred
	0x02	Diagnostic byte 1	Unsigned8	ro	0x00	Diagnostic data (see tables below)
	0x03	Diagnostic byte 2	Unsigned8	ro	0x00	
	0x04	Diagnostic byte 3	Unsigned8	ro	0x00	
	0x05	Diagnostic byte 4	Unsigned8	ro	0x00	
	0x06	Acknowledge	Unsigned8	rw	0x00	Writing any value resets the diagnostic alarm counter and if necessary acknowledges the alarm

EPM-S404, EMP-S405 - diagnostic alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Set in case of module fault Bit 1: Set in case of internal error Bit 2: Set in case of external error Bit 3: Set in case a channel error is active Bit 4: Set in case the external supply voltage is missing Bit 6: 0 (fix) Bit 7: Set in case of parameterisation error
2	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Set if channel information is available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bits 3 ... 0: 0 (fixed) Bit 4: Set in case of internal communication error Bit 5: Reserved Bit 6: Set if process alarm is lost Bit 7: 0 (fixed)

EPM-S600 - diagnostic alarm	
Diagnostic byte	Bit 7 ... 0
1	Bit 0: Set in case of module fault Bit 1: 0 (fix) Bit 2: Set in case of external error Bit 3: Set in case a channel error is active Bit 6 ... 4: 0 (fix) Bit 7: Parameterisation error
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: channel information available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fix) Bit 6: Process alarm lost Bit 7: 0 (fixed)

Counting functions	Description
1	Bit 0: Set in case of module fault Bit 1: 0 (fix) Bit 2: Set in case of external error Bit 3: Set in case a channel error is active Bit 6 ... 4: 0 (fix) Bit 7: Parameterisation error
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: channel information available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fix) Bit 6: Process alarm lost Bit 7: 0 (fixed)

EPM-S602 diagnostic alarm	
Diagnostic byte	Bit 7 ... 0
1	Bit 0: Set in case of module fault Bit 1: Set in case of internal error Bit 2: Set in case of external error Bit 3: Set in case a channel error is active Bit 4: Set in case the external supply voltage is missing Bit 7 .... 5: 0 (fix)
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: Set if channel information is available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fix) Bit 6: Set if process alarm is lost Bit 7: 0 (fixed)

EPM-S603 - diagnostic alarm	
Diagnostic byte	**Bit 7 ... 0**
1	0 (fixed)
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: channel information available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	0 (fixed)

EPM-S604 - diagnostic alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Set in case of module fault Bit 1: Set in case of internal error Bit 2: Set in case of external error Bit 3: Set in case a channel error is active Bit 4: Set in case the external supply voltage is missing Bit 6 ... 5: 0 (fix) Bit 7: Parameterisation error
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: channel information available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fix) Bit 6: Process alarm lost Bit 7: 0 (fixed)

Diagnostic data (bytes 1 ... n)

This object gives you access to all a module's diagnostic data. You can either call up the current diagnostic data or a module's diagnostic data on any EtherCAT slot number.

Index	Subindex	Name	Type	Attr.	Default	Meaning
0x5005	0x00	Diagnostic data	Unsigned8	ro	18	
	0x01	Slot number	Unsigned8	rw	0	During read access, here you will find the EtherCAT slot no. of the module from which the following diagnostic data originates. By writing an EtherCAT slot no., you can query the diagnostic data of any module.
	0x02		Unsigned8	ro	0	Diagnostic alarm data (see module description)
	0x03		Unsigned8	ro	0	
	0x04		Unsigned8	ro	0	
	0x05		Unsigned8	ro	0	
	0x06		Unsigned8	ro	0	
	0x07		Unsigned8	ro	0	
	0x08		Unsigned8	ro	0	
	0x09		Unsigned8	ro	0	
	0x0A		Unsigned8	ro	0	
	0x0B		Unsigned8	ro	0	
	0x0C		Unsigned8	ro	0	
	0x0D		Unsigned8	ro	0	
	0x0E		Unsigned8	ro	0	
	0x0F		Unsigned8	ro	0	
	0x10		Unsigned8	ro	0	
	0x11		Unsigned8	ro	0	
	0x12		Unsigned32	ro	0	

EPM-S404, EMP-S405 - diagnostic alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Set in case of module fault Bit 1: Set in case of internal error Bit 2: Set in case of external error Bit 3: Set in case a channel error is active Bit 4: Set in case the external supply voltage is missing Bit 6: 0 (fix) Bit 7: Set in case of parameterisation error
2	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Set if channel information is available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bits 3 ... 0: 0 (fixed) Bit 4: Set in case of internal communication error Bit 5: Reserved Bit 6: Set if process alarm is lost Bit 7: 0 (fixed)
5	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fixed)
6	Number of diagnostic bits output by the module per channel (here 0x08))
7	Number of channels of a module (here 0x04)
8	Bit 0: Channel error channel group 0 Bit 1: Channel error channel group 1 Bit 7 ... 2: 0 (fix)
9	Channel-specific error: Channel 0: Bit 0: Set in case of project planning/parameterisation error Bit 3 ... 1: 0 (fix) Bit 4: Set in case of open circuit Bit5: Set if process alarm is lost Bit 6: Set in case of measuring range underflow Bit 7: Set in case of measuring range exceedance
10	Channel-specific error: Channel 1: Bit 0: Set in case of project planning/paramerisation error Bit 3 ... 1: 0 (fix) Bit 4: Set in case of open circuit Bit5: Set if process alarm is lost Bit 6: Set in case of measuring range underflow Bit 7: Set in case of measuring range exceedance
11 ... 16	0 (fixed)

EPM-S600 - diagnostic alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Set in case of module fault Bit 1: 0 (fix) Bit 2: Set in case of external error Bit 3: Set in case a channel error is active Bit 6 ... 4: 0 (fix) Bit 7: Parameterisation error
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: channel information available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fix) Bit 6: Process alarm lost Bit 7: 0 (fixed)
5	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
6	Number of diagnostic bits output by the module per channel (here 0x08))
7	Number of channels of a module (here 0x01)
8	Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fix)
9	Diagnostic alarm due to lost process alarm to ... Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bits 7 ... 5: 0 (fixed)
10 ... 16	0 (fixed)

EPM-S601 - diagnostic alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Set in case of module fault Bit 1: 0 (fix) Bit 2: Set in case of external error Bit 3: Set in case a channel error is active it 6 ... 4: 0 (fix) Bit 7: Parameterisation error
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: channel information available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fix) Bit 6: Process alarm lost Bit 7: 0 (fixed)
5	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
6	Number of diagnostic bits output by the module per channel (here 0x08))
7	Number of channels of a module (here 0x02)
8	Bit 0: Error in channel group 0 (counter 1) Bit 1: Error in channel group 1 (counter 2) Bit 7 ... 2: 0 (fix)
9	Channel group 0: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fix) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fix)
10	Channel group 1: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fix) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fix)
11 ... 16	0 (fixed)

EPM-S602 diagnostic alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Set in case of module fault Bit 1: Set in case of internal error Bit 2: Set in case of external error Bit 3: Set in case a channel error is active Bit 4: Set in case the external supply voltage is missing Bit 7 .... 5: 0 (fix)
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: Set if channel information is available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fix) Bit 6: Set if process alarm is lost Bit 7: 0 (fixed)
5	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
6	Number of diagnostic bits output by the module per channel (here 0x08))
7	Number of channels of a module (here 0x02)
8	Bit 0: Error in channel group 0 (counter 1) Bit 1: Error in channel group 1 (counter 2) Bit 7 ... 2: 0 (fix)
9	Channel group 0: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fix) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fix)
10	Channel group 1: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fix) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fix)
11 ... 16	0 (fixed)

EPM-S603 - diagnostic alarm	
Diagnostic byte	**Bit 7 ... 0**
1	0 (fixed)
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: channel information available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	0 (fixed)
5	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
6	Number of diagnostic bits output by the module per channel (here 0x00))
7	Number of channels of a module (here 0x02)
8 ... 16	0 (fixed)

EPM-S604 - diagnostic alarm	
Diagnostic byte	**Bit 7 ... 0**
1	Bit 0: Set in case of module fault Bit 1: Set in case of internal error Bit 2: Set in case of external error Bit 3: Set in case a channel error is active Bit 4: Set in case the external supply voltage is missing Bit 6 ... 5: 0 (fix) Bit 7: Parameterisation error
2	Bit 3 ... 0: Module class, 0b1000: function module Bit 4: channel information available Bits 7 ... 5: 0 (fixed)
3	0 (fixed)
4	Bit 5 ... 0: 0 (fix) Bit 6: Process alarm lost Bit 7: 0 (fixed)
5	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: 0 (fixed)
6	Number of diagnostic bits output by the module per channel (here 0x08))
7	Number of channels of a module (here 0x01)
8	Bit 0: Error in channel group 0
9 ... 16	0 (fixed)

7.12.3 Standard objects

The following indices can be used for purposes of diagnostics. They show operating states. Settings cannot be made.

Index		Name	INFO
I-1000		Device Type	Device type of the module Read only
I-1003		Last error	Current error, the fault memory is cleared by a reset or power cycle. Read only
I-1008		Device name	Device name of the module Read only
I-1009		Hardware version	Hardware version of the module Read only
I-100A		Software version	Software version of the module Read only
I-100B		System version	Delivery status, depending on FPGA versions of bus coupler and modules, at least system version 2 Read only
I-1018		Identity	General data of the EtherCAT bus coupler module Read only
	0	Identity object	
	1	Vendor ID	
	2	Product code	
	3	Revision number	
	4	Serial number	
I-1600 ... I-163F		Output mapping modules	
	0	RxPDO Map	Number of outputs at this slot Entry only exists for slots with parameterisable modules
	1	Output mapping	E.g.: 0x7000:01, 1 > the first output on slot 0 is 1 bit long
	2	Output mapping	
	...	...	
I-1A00 ... I-1A3F		Input mapping modules	
	0	TxPDO map	Number of inputs at this slot Entry only exists for slots with parameterisable modules
	1	Input mapping	E.g.: 0x6000:01, 8 > the first output on slot 0 is 8 bits long
	2	Input mapping	
	...	...	
I-1AFF ... I-1A3F		Input mapping coupler	Due to the system design, the mapping in the project planning tool must not be changed otherwise errors may arise in the process image!
	0	Status PDO	Mapping for coupler's alarm counter
	1	Input mapping	Mapping for process alarm counter
	2	Input mapping	
	...	...	Mapping for the diagnostic alarm counter

Index		Name	INFO
I-1C00		Sync manager type	Use of the sync manager channels Read only
	0	Sync manager type	
	1	Subindex 01	Write mailbox from view of master
	2	Subindex 02	Read mailbox from view of master
	3	Subindex 03	Process input data from view of master
	4	Subindex 04	Process output data from view of master
I-1C12		RxPDO assign	Mapping of the digital output modules Read only
	0	RxPDO assign	
	1	Subindex 001	Output module at slot 1
	...	...	...
	32	Subindex 032	Output module at slot 32
I-1C13		TxPDO assign	Mapping of the digital input modules Read only
	0	TxPDO assign	
	1	Subindex 001	Input module at slot 1
	...	...	...
	32	Subindex 032	Input module at slot 32
I-1C32		SM output parameter	Read only
	0	SM output parameter	
	1	SYNC MODE	
	2	Cycle time	
	3	Shift time	
	4	Sync modes supported	
	5	Minimum cycle time	
	6	Maximum shift time	
I-1C33		SM input parameter	Read only
	0	SM input parameter	
	1	SYNC MODE	
	2	Cycle time	
	3	Shift time	
	4	Sync modes supported	
	5	Minimum cycle time	
	6	Maximum shift time	
I-3000		Parameter EtherCAT coupler	
	0	Coupler parameter	
	1	Auto-acknowledge	States how the EtherCAT coupler is to respond to alarms. • When auto-acknowledge = 0, you are responsible for acknowledgement. You are therefore informed of every alarm. If an alarm is not acknowledged, other alarms are inhibited. • When auto-acknowledge = 1, each alarm is automatically acknowledged by the EtherCAT coupler. In this mode, diagnostic data is overwritten by new alarms. Auto-acknowledge = 1 by default. Auto-acknowledge should be activated for continuous use.

Index		Name	INFO
I-3100 ... I-313F		Parameter EtherCAT coupler	This object provides access to the parameters of an I/O compound module. The EtherCAT slot is addressed via the index. Subindices provide access to the corresponding parameter. The respective module description contains the subindex assignment. Again here, power and terminal modules are not detected by the EtherCAT coupler and are therefore not taken into account when listing and/or assigning slots.
	0	Parameter	Number of parameters
	1	Param	Module parameter data
	2	Param	
	...	...	
I-4000		Clear IO counter	Writing any value to the corresponding subindex clears the counter
	0	Clear master counter	
	1	Clear module counter	
I-4001			
	0	Master counter	
	1	Expected length error	
	2	Timeout error	
	3	Stop bit error	
	4	FCS error	
	5	Telegram length error	
	6	Telegram type error	
	7	Alarm retry error	
	8	Bus idle time error	
	9	Wrong node address	
	A	Telegram valid	
	B	Master load	
I-4002			
	0	Module MDL counter	
	1	Slot 1	
	2	Slot 2	
	...	...	
	40	Slot 64	
I-4003			
	0	Module NDL counter	
	1	Slot 1	
	2	Slot 2	
	...	...	
	40	Slot 64	
I-4100			
	0	System version	Version details of coupler components
	1	Master FPGA	FPGA version
	2	I/O bus	Backplane bus version
	3	Firmware package	Version package
	4	Mx file	Name and version of coupler's Mx file

Index		Name	INFO
I-4101			
	0	Module FPGA version	FPGA versions of modules
	1	Slot 1	EtherCAT slot
	2	Slot 1	
	...	...	
	64	Slot 64	
I-4102			
	0	Module firmware version	Firmware versions of modules
	1	Slot 1	EtherCAT slot
	2	Slot 1	
	...	...	
	64	Slot 64	
I-4103			
	0	Module serial number	Serial numbers of modules
	1	Slot 1	EtherCAT slot
	2	Slot 1	
	...	...	
	64	Slot 64	
I-5000			If object I-F100 indicates that a process alarm has occurred, you can access the current process alarm data here. The corresponding module description contains the process alarm data assignment.
	0	Hardware interrupt data	Current process alarm data
	1	Slot number	EtherCAT slot no. of module on which the alarm has occurred
	2	Hardware interrupt data 00	Process alarm data
	3	Hardware interrupt data 01	
	4	Hardware interrupt data 02	
	5	Hardware interrupt data 03	
	6	Acknowledge	Writing any value resets the process alarm counter and if necessary acknowledges the alarm If auto-acknowledge is deactivated for the EtherCAT coupler, you can reset the hardware interrupt counter of object 0xF100 and acknowledge the process alarm by writing any value to subindex 0x06 of index I-5000.

7.12.4 Storage of the PDOs

Index	Description
I-6000	Input PDO (1. slot)
I-6001	Input PDO (2. slot)
...	...
I-601F	Input PDO (32. slot)
I-7000	Output PDO (1. slot)
I-7001	Output PDO (1. slot)
...	...
I-701F	Output PDO (32. slot)

7.12.5 Emergency telegram

Note!

Emergency frames can only be sent by modules that support alarm functions such as "process alarm" or "diagnostic alarm". These are **EPM-S404**, **EPM-S405**, **EPM-S406**, **EPM-S408** and the counter modules **EPM-S6xx**.

The module-specific alarm structure is described in the following sections:

- Process alarm data (📖 384)
- Diagnostic alarm data (📖 386)

Emergency messages are triggered by device-internal mechanisms and reported to the master by the mailbox service of EtherCAT. If a state change cannot be executed for an EtherCAT gateway, it is reported by a corresponding emergency message.

For a more detailed error description, the EtherCAT master reads out the **AL status code** (AL = application layer). This is located in the EtherCAT coupler in the register '0x0134'. Here you can find the EtherCAT-specific error codes (📖 401) and Manufacturer-specific error codes (📖 402).

Structure of the emergency message

Bytes							
0	1	2	3	4	5	6	7
Emergency Error Lenze = 0xFF00		State 1: Init 2: Pre-Operational 3: Save-Operational 4: Operational	Data				
			Slot Slot of the error-reporting module.	Type 1: Process alarm 2: Diagnostic alarm	Code Error code described for the corresponding module.		Reserved (any assignment)

The emergency messages are displayed as raw data in the »PLC Designer« and entered in the logbook. Within this telegram, the reported error is entered as 5-digit hex value starting from the "data:" information. The structure of the hex value corresponds to the bytes 3 ... 7 in the above table.

Example of an error message

```
07174 2013-06-04 10:12:04  _02M6D601 (1008): CoE emergency request.
id=0x0, len=8, ErrCode=0x0, ErrReg=0x8, data: 0x5 0x2 0x0 0x15 0x0.
```

Error information "data":

- Slot: 5
- Type: Diagnostic alarm
- Code: Error message '0x0015' (Invalid mailbox configuration)

7.12.6 EtherCAT-specific error codes

Code	Description	Current State	Resulting State
0x0000	No error	Any	Current State
0x0001	Unspecified error	Any	Any + E
0x0002	No Memory	Any	Any + E
0x0011	Invalid requested state change	I → S, I → O, P → O, O → B, S → B, P → B	Current state + E
0x0012	Unknown requested state	Any	Current state + E
0x0013	Bootstrap not supported	I → B	I + E
0x0014	No valid firmware	I → P	I + E
0x0015	Invalid mailbox configuration	I → B	I + E
0x0016	Invalid mailbox configuration	I → P	I + E
0x0017	Invalid SyncManager configuration	P → S, S → O	Current state + E
0x0018	No valid inputs available	O, S → O	S + E
0x0019	No valid outputs	O, S → O	S + E
0x001A	Synchronisation error	O, S → O	S + E
0x001B	SyncManager watchdog	O, S	S + E
0x001C	Invalid SyncManager Types	O, S, P → S	S + E
0x001D	Invalid Output Configuration	O, S, P → S	S + E
0x001E	Invalid Input Configuration	O, S, P → S	P + E
0x001F	Invalid Watchdog Configuration	O, S, P → S	P + E
0x0020	Coupler needs cold start	Any	Current state + E
0x0021	Coupler needs INIT	B, P, S, O	Current state + E
0x0022	Coupler needs PREOP	S, O	S + E, O + E
0x0023	Coupler needs SAFEOP	O	O + E
0x0024	Invalid Input Mapping	P → S	0x0019
0x0025	Invalid Output Mapping	P → S	P + E
0x0026	Inconsistent Settings	P → S	P + E
0x0027	Free-run not supported	P → S	P + E
0x0028	Synchronisation not supported	P → S	P + E
0x0029	Free-run needs 3 Buffer Mode	P → S	P + E
0x002A	Background Watchdog	S, O	P + E
0x002B	No Valid Inputs and Outputs	O, S → O	S + E
0x002C	Fatal Sync Error	O	S + E
0x002D	No Sync Error	S → O	S + E
0x0030	Invalid DC SYNC Configuration	O, S → O, P → S	P + E, S + E
0x0031	Invalid DC Latch Configuration	O, S → O, P → S	P + E, S + E
0x0032	PLL Error	O, S → O	S + E

Code	Description	Current State	Resulting State
0x0033	DC Sync IO Error	O, S ➜ O	S + E
0x0034	DC Sync Timeout Error	O, S ➜ O	S + E
0x0035	DC Invalid Sync Cycle Time	P ➜ S	P + E
0x0036	DC Sync0 Cycle Time	P ➜ S	P + E
0x0037	DC Sync1 Cycle Time	P ➜ S	P + E
0x0041	MBX_AOE	B, P, S, O	Current state + E
0x0042	MBX_EOE	B, P, S, O	Current state + E
0x0043	MBX_COE	B, P, S, O	Current state + E
0x0044	MBX_FOE	B, P, S, O	Current state + E
0x0045	MBX_SOE	B, P, S, O	Current state + E
0x004F	MBX_VOE	B, P, S, O	Current state + E
0x0050	EEPROM No Access	Any	Any + E
0x0051	EEPROM Error	Any	Any + E
0x0060	Coupler Restarted Locally	Any	I
< x8000	Reserved		

I: Init
P: Pre-Operational
S: Safe-Operational
O: Operational
B: Bootstrap
E: Error

7.12.7 Manufacturer-specific error codes

Code	Description	Current State	Resulting State
0x8000	no module recognised or present	I	I + E
0x8001	Module at system bus needs update	P > S	P + E
0x8002	Init error	P > S	P + E
0x8003	unexpected restart (Watchdog)	P > S	P + E
0x8004	Error reading EEPROM	P > S	P + E
0x8005	EPM-S130 module area too big or small	P > S	P + E

I: Init
P: Pre-Operational
S: Safe-Operational
O: Operational
B: Bootstrap
E: Error

7.12.8 SDO error codes

The SDO service and the object directory serve to access all parameters of the EtherCAT slave.

If an SDO request is evaluated negatively, a corresponding error code is output in the Abort SDO Transfer Protocol. The following table shows the possible error codes:

Code	Description
0x05030000	Toggle bit not alternated
0x05040000	SDO protocol timed out
0x05040001	Client/server command specifier not valid or unknown
0x05040002	Invalid block size (block mode only)
0x05040003	Invalid sequence number (block mode only)
0x05040004	CRC error (block mode only)
0x05040005	Out of memory
0x06010000	Unsupported access to an object
0x06010001	Attempt to read a write only object
0x06010002	Attempt to write a read only object
0x06020000	Object does not exist in the object dictionary
0x06040041	Object cannot be mapped to the PDO
0x06040042	The number and length of the objects to be mapped would exceed PDO length
0x06040043	General parameter incompatibility reason
0x06040047	General internal incompatibility in the device
0x06060000	Access failed due to an hardware error
0x06070010	Data type does not match, length of service parameter does not match
0x06070012	Data type does not match, length of service parameter too high
0x06070013	Data type does not match, length of service parameter too low
0x06090011	Sub-index does not exist
0x06090030	Value range of parameter exceeded (only for write access)
0x06090031	Value of parameter written too high
0x06090032	Value of parameter written too low
0x06090036	Maximum value is less than minimum value
0x08000000	General error
0x08000020	Data cannot be transferred or stored to the application
0x08000021	Data cannot be transferred or stored to the application because of local control
0x08000022	Data cannot be transferred or stored to the application because of the present device state
0x08000023	Object directory dynamic generation fails or no object directory is present (e.g. object directory is generated from file and generation fails because of an file error)

8 DeviceNet communication

8.1 About DeviceNet

DeviceNet is an open device net standard that satisfies the user profile for industrial real-time system applications. The DeviceNet protocol has an open specification that is the property of and administered by the independent vendor organization "Open DeviceNet Vendor Association" ODVA. This is where standardised device profiles are created to provide compatibility and exchangeability on logical level for simple devices of the same type.

In contrast to the classical source–destination model, DeviceNet uses a modern producer/consumer model that requires data packets with identifier fields for the identification of the data. This approach caters for multiple priority levels, more efficient transfers of I/O data and multiple consumers for the data.

A device that has data to send produces the data on the network together with an identifier. All devices requiring data listen for messages. When devices recognize a suitable identifier, they act and consume the respective data.

DeviceNet carries two types of messages:

- I/O messages
  Messages that are subject to critical timing constraints and contain data for control purposes that can be exchanged by means of single or multiple connections and that employ identifiers with a high priority.
- Explicit messages
  These are used to establish multi-purpose point-to-point communication paths between two devices, which are used for the configuration of network couplers and for diagnostic purposes. These functions usually employ identifiers of a low priority.

Messages that are longer than 8 bytes are subject to the fragmentation service. A set of rules for master/slave, peer-to-peer- and multi-master connections is also available.

Transmission medium

DeviceNet employs a screened five-core cable as data communication medium. DeviceNet uses voltage differences and for this reason it exhibits less sensitivity to interference than a voltage or current based interface.

Signals and power supply conductors are included in the same network cable. It is therefore possible to connect devices that obtain the operating voltage via the network as well as devices with an integrated power supply. Furthermore it is possible to connect redundant power supplies to the network that guarantees the power supply when required.

DeviceNet employs a master line/tap line topology with up to 64 network nodes. The maximum distance is either 500m at a rate of 125kbit/s, 250m at a rate of 250kbit/s or 100m at a rate of 500kbit/s.

The length of the tap lines can be up to 6m while the total length of all tap lines depends on the baud rate.

Network nodes can be removed from or inserted into the network without interruption of the network operation. New and failed stations are detected automatically.

Bus access procedure

DeviceNet operates according to the Carrier-Sense Multiple Access (CSMA) principle, i.e. every station on the network may access the bus when it is not occupied (random access).

The exchange of messages is message orientated and not station orientated.

Each message is provided with a unique and prioritising identifier. At any time only one station is able to occupy the bus with its messages.

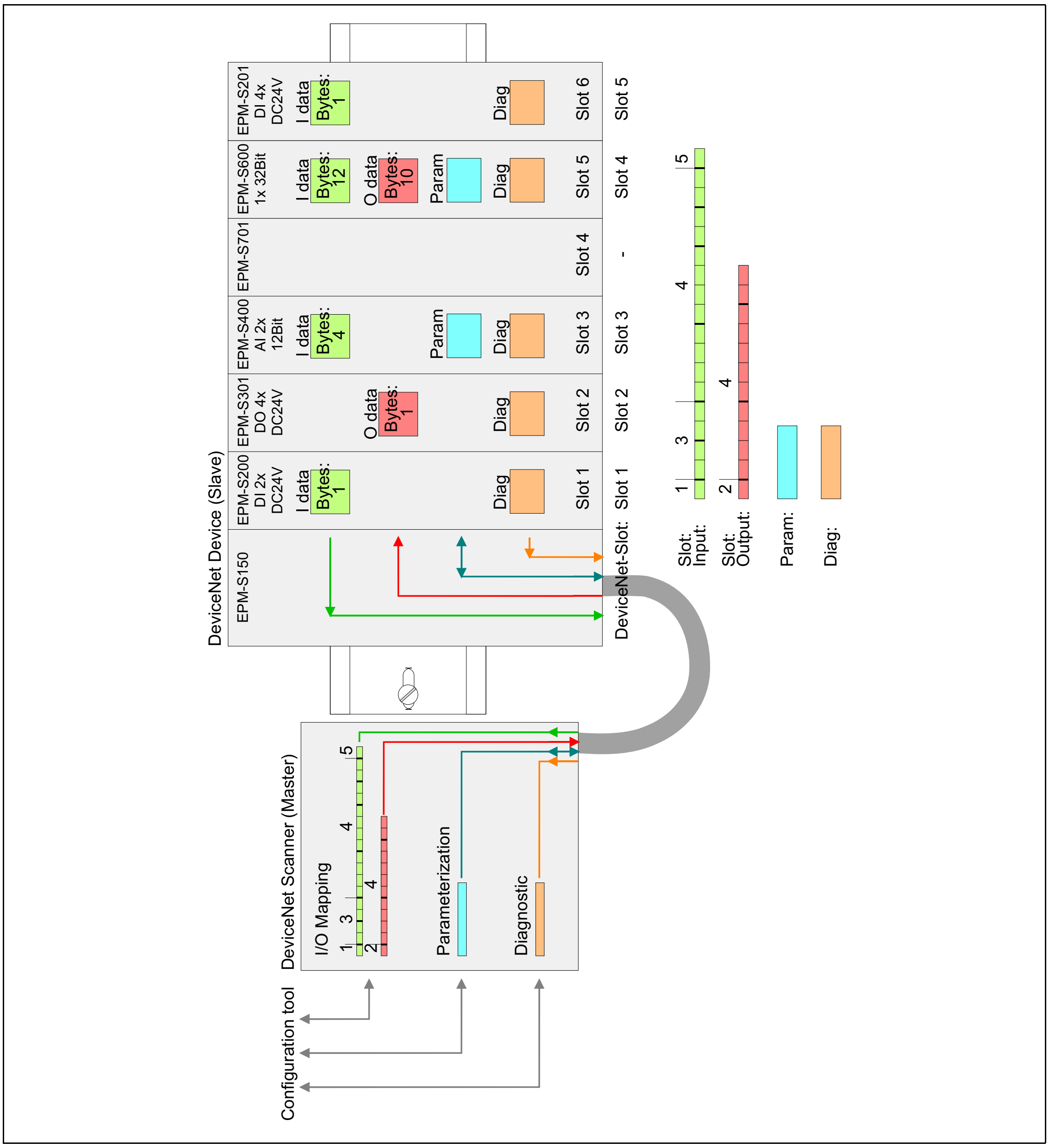
The DeviceNet bus access control is subject to non-destructive, bit-wise arbitration. In this case non-destructive means that the successful station participating in the arbitration does not need to re-send its message. The most important station is selected automatically when multiple stations access the bus simultaneously. If a station that is ready to send recognises that the bus is occupied, its send request is delayed until the current transfer has been completed.

Addressing

All stations on the bus must be uniquely identified by means of an ID address. Every DeviceNet device has addressing facilities.

8.2 Access to the I/O system 1000

The following illustration shows the access under DeviceNet to the areas "I/O", "parameters" and "diagnostics".



> **Note!**
>
> Please note that the supply and terminal modules do not have any type identification and thus cannot be recognised by the DeviceNet bus coupler.
>
> In the following, slots within the DeviceNet are called DeviceNet slots. Counting always starts at 1.

EDS file (Electronic Data Sheet)

For the DeviceNet bus coupler module EPM-S150, the EDS file "Lenze-EPM-S150_64_10.eds" is available in the download area of www.Lenze.de. Install this EDS file in your project planning tool. More information on how to install the EDS file can be found in the manual of your project planning tool.

8.2.1 Access to the I/O area

The DeviceNet bus coupler automatically detects the I/O compound modules plugged at the backplane bus and generates the number of the input and output bytes. (Supply and power distributing modules are not considered.)

When the DeviceNet scanner is configured, the respective total length of the input or output data must be indicated. Information on I/O assignment of a module can be found in the "Product description" chapter in the prevailing descriptions of the I/O compound modules.

The position (offset) of the input and output bytes within the input and output data results from the sequence of the modules (DeviceNet slot 1 ... 64). Use the basic address set for the bus coupler in the DeviceNet scanner to access the input or output data via the corresponding offset.

During operation, the DeviceNet bus coupler cyclically reads the input data of the peripheral modules and always provides the current state for the DeviceNet scanner. Output data, the DeviceNet bus coupler has directly received by the DeviceNet scanner will be directly forwarded to the modules as soon as they have been received via DeviceNet.

How to configure the DeviceNet scanner (master):

1. Switch off the voltage supply of the DeviceNet bus coupler.

   Set the baud rate and the DeviceNet address.
2. Start your configuration tool for the DeviceNet scanner.
3. Set the "POLL IO" connection type in the DeviceNet scanner.
4. Enter the number of input and output data:
5. Number of input data: Produced connection size
6. Number of output data: Consumed connection size
7. Enter a basic address for the input and output data (mapping).
8. Activate the DeviceNet bus coupler EPM-S150 in the scan list.
9. Start the DeviceNet scanner.

After the DeviceNet scanner has been configured, the input and output modules can be triggered under the configured addresses.

8.2.2 Access to parameter data

Your configuration tool also enables you to parameterise your I/O compound modules. For this purpose, the DeviceNet bus coupler must be located actively at the bus.

Your configuration tool serves to parameterise your modules in string form via the corresponding DeviceNet slot. you can also transfer the current parameters from the modules to the configuration tool, adapt and rewrite them.

Rules for the parameters

- Each DeviceNet must be parameterised with a string.
- Within the string, each parameter consists of a type with attached value.
- The parameters must be separated by a space.
- Only parameters written in small letters are supported.

Depending on the type, you can indicate hexadecimal, decimal or binary values as parameters:

Type	Meaning	Value as	Example string
x	1 byte	Hexadecimal	xhh
2x	2 bytes	Hexadecimal	2xhhhh
4x	4 bytes	Hexadecimal	4xhhhhhhhh
+	1 byte	decimal positive	+ddd
2+	2 bytes	decimal positive	2+ddddd
4+	4 bytes	decimal positive	4+dddddddddd
-	1 byte	decimal negative	-ddd
2-	2 bytes	decimal negative	2-ddddd
4-	4 bytes	decimal negative	4-dddddddddd
b	1 byte	Binary	xbbbbbbbb

Example: The I/O compound module EPM-S405 has 18 bytes of parameter data. For parameterisation with the default values, the following string occurs:

x00 x00 x00 x00 x00 x02 xC1 x02 2x7FFF 2x8000 xC1 x02 2x7FFF 2x8000

Note!

If the parameter data does not match the hardware structure, the DeviceNet bus coupler changes to an error status and signalises it via its status LEDs.

Modules that have not yet been parameterised will be automatically supplied with their default values as soon as a read access from the configuration tool is executed.

8.2.3 Access to diagnostic data

The DeviceNet bus coupler exclusively supports passive diagnostics, i.e. no alarm activation on the module side is required for diagnostics. You have to request the diagnostics by yourself.

For this purpose, go to your configuration tool and select the diagnostic data of the corresponding DeviceNet slot. All diagnostic data of the module will be displayed as byte sequence.

DeviceNet bus coupler

Class code: 100 (0x64)

No.	Name	Information	Format	Example
1	DeviceName	Device name	String	EPM-S150
2	HwVersion	HW output version	String	02
3	SwVersion	SW output version	String	V101
4	SerialNumber	Serial number	Unsigned16, String	00000205
5	FpgaVersion	FPGA version	Unsigned16, String	V208
6	MxFile	Mx file	String	MX000053.101
7	ProductVersion	Product version	String	01V01.00
8	OrderCode	Order No.	String	(8-digit Lenze material number)

I/O compound module

Class code:

Slot 1: 101 (0x65)

Slot 2: 102 (0x66)

...

Slot 64: 164 (0xA4)

No.	Name	Information	Format	Example
1	DeviceName	Device name	String	EPM-S403
2	HwVersion	HW output version	String	21
3	SwVersion	SW output version	String	V202
4	SerialNumber	Serial number	Unsigned32, String	00001143
5	FpgaVersion	FPGA version	Unsigned16, String	V208
6	MxFile	Mx file	String	MX000028.130
7	ProductVersion	Product version	String	01V31.001
8	OrderCode	Order No.	String	(8-digit Lenze material number)
20	Parameter	Parameter data	String	x00 x00 x31 x31 x31 x31
21	Diagnostics	Diagnostic data	String	x00 x15 x00 x00 x73 x08 x04 x00 x00 x00 x00 x00 x00 x00 x00 x00 4x000020EB

8.3 Parameterising analog I/Os

8.3.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
2	Function channel 2		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

8.3.2 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
2	Function channel 2		0x20
3	Function channel 3		0x20
4	Function channel 4		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

8.3.3 2 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S402

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
2	Function channel 2		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

8.3.4 4 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S403

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
2	Function channel 2		0x31
3	Function channel 3		0x31
4	Function channel 4		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

8.3.5 2 analog inputs -10 ... +10 V (16 bits) - EPM-S406

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Diagnostics	Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
2	Reserved	0	
3	Limit value monitoring	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 (0 = deactivated; 1 = activated) Bits 7 ... 2: reserved	0x00
4	Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
5	Function channel 1	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
6	Reserved	0	
7	Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
8	Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
9	Function channel 2	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
10	Reserved	0	
11	Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
12	Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	**Voltage (U)** [V]	**dec**	**hex**	**Range**	**Conversion**
-10 ... +10 V (0x12)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-5	-13824	CA00		
	-10	-27648	9400		
	-11.76	-32512	8100	Underflow	
-10 ... +10 V (0x22)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-5	-8192	E000		
	-10	-16384	C000		
	-12.5	-20480	B000	Underflow	
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-1.76	-4864	ED00	Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-2	-3277	F333	Underflow	

8.3.6 2 analog inputs 0/4 ... 20 mA (16 bits) - EPM-S408

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Diagnostics	Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
2	Reserved	0	
3	Limit value monitoring	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 (0 = deactivated; 1 = activated) Bits 7 ... 2: reserved	0x00
4	Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
5	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
6	Reserved	0	
7	Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
8	Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
9	Function channel 2	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
10	Reserved	0	
11	Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
12	Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

8.3.7 2 analog outputs 0 ... 10 V (12 bits) - EPM-S500

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Reserved	0	
2	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
3	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x10
4	Function channel 2		0x10

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Output area (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

8.3.8 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Reserved	0	
2	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
3	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
4	Function channel 2		0x20
5	Function channel 3		0x20
6	Function channel 4		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Output area (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

8.3.9 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Reserved	0	
2	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
3	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
4	Function channel 2		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

8.3.10 4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Reserved	0	
2	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
3	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
4	Function channel 2		0x31
5	Function channel 3		0x31
6	Function channel 4		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

8.4 Parameterising temperature measurement

8.4.1 4(2) analog input for resistance measurement - EPM-S404

Note!

Use parameter setting to deactivate unused inputs.

If thermal detectors are connected in a 3 or 4 conductor setup, channels 3 and/or 4 must be deactivated.

2-, 3-, 4-wire conductor measurement (26)

The module does not provide any auxiliary supply for sensors.

Parameter data

Position in string No.	**Name**	**Description/value**	**Lenze**
1	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
2	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) : Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
3	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) : Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
4	Reserved		
5	Temperature system	Bit 0, 1: 0b00 = °C; 0b01= °F; 0b10 = K Bit 2 ... 7: Reserved	0x00
6	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02

Position in string No.	Name	Description/value	Lenze
Channel 1			
7	Function channel 1	Thermal detector: 80 (0x50): PT100 2-wire conductor -200°C ... +850°C / -2000 ... +8500 81 (0x51): PT1000 2-wire conductor -200°C ... +850°C / -2000 ... +8500 82 (0x52): Ni100 2-wire conductor -60°C ... +250°C / -600 ... +2500 83 (0x53): Ni1000 2-wire conductor -60°C ... +250°C / -600 ... +2500 88 (0x58): PT100 3-wire conductor -200°C ... +850°C / -2000 ... +8500 89 (0x59): PT1000 3-wire conductor -200°C ... +850°C / -2000 ... +8500 90 (0x5A): Ni100 3-wire conductor -60°C ... +250°C / -600 ... +2500 91 (0x5B): Ni1000 3-wire conductor -60°C ... +250°C / -600 ... +2500 96 (0x60): PT100 4-wire conductor -200°C ... +850°C / -2000 ... +8500 97 (0x61): PT1000 4-wire conductor -200°C ... +850°C / -2000 ... +8500 98 (0x62): Ni100 4-wire conductor -60°C ... +250°C / -600 ... +2500 99 (0x63): Ni1000 4-wire conductor -60°C ... +250°C / -600 ... +2500 Resistor: 112 (0x70): R 60-Ω 2-wire conductor 0.00 ... +60.00 / 0 ... +32767 113 (0x71): R 600-Ω 2-wire conductor 0.00 ... +600.00 / 0 ... +32767 114 (0x72): R 3000-Ω 2-wire conductor 0.00 ... +3000.00 / 0 ... +32767 115 (0x73): R 6000-Ω 2-wire conductor 0.00 ... +6000.00 / 0 ... +32767 128 (0x80): R 60-Ω 4-wire conductor 0.00 ... +60.00 / 0 ... +32767 129 (0x81): R 600-Ω 4-wire conductor 0.00 ... +600.00 / 0 ... +32767 130 (0x82): R 3000-Ω 4-wire conductor 0.00 ... +3000.00 / 0 ... +32767 144 (0x90): R 60-Ω 2-wire conductor 0.00 ... +60.00 / 0 ... +6000 145 (0x91): R 600-Ω 2-wire conductor 0.00 ... +600.00 / 0 ... +6000 146 (0x92): R 3000-Ω 2-wire conductor 0.00 ... +3000.00 / 0 ... +30000 160 (0xA0): R 60-Ω 2-wire conductor 0.00 ... +60.00 / 0 ... +6000 161 (0xA1): R 600-Ω 2-wire conductor 0.00 ... +600.00 / 0 ... +6000 162 (0xA2): R 3000-Ω 2-wire conductor 0.00 ... +3000.00 / 0 ... +30000 255 (0xFF): Channel deactivated	0x50

Position in string No.	**Name**	**Description/value**	**Lenze**
8	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At 50 Hz: 44.1 ms/channel 16 bits; at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x00
9	Upper limit value channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. If your measured value is outside a limit value and you have activated the limit value monitoring, a process alarm is triggered.	0x7FFF
10	Lower limit value channel 1		0x8000
Channel 2			
11	Function channel 2	See channel 1	0x50
12	Conversion time channel 2	See channel 1	0x00
13	Upper limit value channel 2	See channel 1	0x7FFF
14	Lower limit value channel 2		0x8000
Channel 3 (for two-wire conductor connections only)			
15	Function channel 3	See channel 1	0x50
16	Conversion time channel 3	See channel 1	0x00
17	Upper limit value channel 3	See channel 1	0x7FFF
18	Lower limit value channel 3		0x8000
Channel 4 (for two-wire conductor connections only)			
19	Function channel 4	See channel 1	0x50
20	Conversion time channel 4	See channel 1	0x00

Position in string No.	Name	Description/value	Lenze
21	Upper limit value channel 4	See channel 1	0x7FFF
22	Lower limit value channel 4		0x8000

Measuring range

Measuring range (Fct. no.)	Measured value	Signal range	Range
2-wire: PT100 (0x50)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: PT1000 (0x51)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: NI100 (0x52)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: NI1000 (0x53)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: PT100 (0x58)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: PT1000 (0x59)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: NI100 (0x5A)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: NI1000 (0x5B)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: PT100 (0x60)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
4-wire: PT1000 (0x61)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
4-wire: NI100 (0x62)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: NI1000 (0x63)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: 0 ... 60 Ω (0x70)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x71)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
2-wire: 0 ... 3000 Ω (0x72)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0x78)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x79)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x7A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0x80)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0x81)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0x82)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0x90)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x91)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0x92)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0x98)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x99)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x9A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xA0)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
4-wire: 0 ... 600 Ω (0xA1)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0xA2)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0xD0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0xD1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0xD2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0xD8)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0xD9)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0xDA)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xE0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0xE1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0xE2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow

8.4.2 2 analog inputs for thermocouple measurement - EPM-S405

Parameter data

Position in string No.	Name	Description/value	Lenze
01	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
02	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
03	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
04	Reserved	0	
05	Temperature system	Bit 0, 1: 0b00 = °C; 0b10 = °F; 0b11 = K Bit 2 ... 7: Reserved	0x00
06	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02
Channel 1			
07	Function channel 1	External temperature compensation: 176 (0x60): type J, -210.0 ... +1200.0 °C / -2100 ... +12000 177 (0x61): type K, -270.0 ... +1372.0 °C / -2700 ... +13720 178 (0x62): type N -270.0 ... +1300.0 °C / -2700 ... +13000 179 (0x63): type R, -50.0 ... +1769.0 °C / -500 ... +17690 180 (0x64): type S, -50.0 ... +1769.0 °C / -500 ... +17690 181 (0x65): type T, -270.0 ... +400.0 °C / -2700 ... +4000 182 (0x66): type B, 0.0 ... +1820.0 °C / 0 ... +18200 183 (0x67): type C, 0.0 ... +2315.0 °C / 0 ... +23150 184 (0x68): type E, -270.0 ... +1000.0 °C / -2700 ... +10000 185 (0x69): type L, -200.0 ... +900.0 °C / -2000 ... +9000 Internal temperature compensation: 192 (0xC0): type J, -210.0 ... +1200.0 °C / -2100 ... +12000 193 (0xC1): type K, -270.0 ... +1372.0 °C / -2700 ... +13720 194 (0xC2): type N -270.0 ... +1300.0 °C / -2700 ... +13000 195 (0xC3): type R, -50.0 ... +1769.0 °C / -500 ... +17690 196 (0xC4): type S, -50.0 ... +1769.0 °C / -500 ... +17690 197 (0xC5): type T, -270.0 ... +400.0 °C / -2700 ... +4000 198 (0xC6): type B, 0.0 ... +1820.0 °C / 0 ... +18200 199 (0xC7): type C, 0.0 ... +2315.0 °C / 0 ... +23150 200 (0xC8): type E, -270.0 ... +1000.0 °C / -2700 ... +10000 201 (0xC9): type L, -200.0 ... +900.0 °C / -2000 ... +9000 255 (0xFF): Channel deactivated	0xC1

Position in string No.	Name	Description/value	Lenze
08	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At Hz: 44.1 ms/channel 16 bits at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x02
09	Upper limit value channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. If your measured value is outside a limit value and you have activated the limit value monitoring, a process alarm is triggered.	0x7FFF
0A	Lower limit value channel 1		0x8000
Channel 2			
0B	Function channel 2	See channel 1	0xC1
0C	Conversion time channel 2	See channel 1	0x02
0D	Upper limit value channel 2	See channel 1	0x7FFF
0E	Lower limit value channel 2		0x8000

Measuring range

Measuring range (Fct. no.)	Measured value [°C]	[°F]	[°K]	Range
Type J: -210 ... +1200 °C -346 ... 2192 °F 63.2 ... 1473.2 K (0xB0: ext. comp. 0 °C) (0xC0: int. comp. 0 °C)	+14500	26420	17232	Overflow
	-2100 ... +12000	-3460 ... +21920	632 ... 14732	Nominal range
	-	-	-	Underflow
Type K: -210 ... +1372 °C -454 ... 2501.6 °F 0 ... 1645.2 K (0xB1: ext. comp. 0 °C) (0xC1: int. comp. 0 °C)	+16220	29516	18952	Overflow
	-2700 ... +13720	-4540 ... 25016	0 ... 16452	Nominal range
	-	-	-	Underflow
Type N: -270 ... +1300 °C -454 ... 2372 °F 0 ... 1573.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+15500	28220	18232	Overflow
	-2700 ... +13000	-4540 ... 23720	0 ... 15732	Nominal range
	-	-	-	Underflow
Type R: -50 ... +1769 °C -58 ... 3216.2 °F 223.2 ... 2042.2 K (0xB3: ext. comp. 0 °C) (0xC3: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type S: -50 ... +1769 °C -58 ... 3216.2 °F 223.2 ... 2042.2 K (0xB4: ext. comp. 0 °C) (0xC4: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type T: -270 ... +440 °C -454 ... 752 °F 3.2 ... 673.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+5400	10040	8132	Overflow
	-2700 ... +4000	-4540 ... 7520	32 ... 6732	Nominal range
	-	-	-	Underflow
Type B: 0 ... +1820 °C 32 ... 2786.5 °F 273.2 ... 2093.2 K (0xB6: ext. comp. 0 °C) (0xC6: int. comp. 0 °C)	+20700	32766	23432	Overflow
	0 ... +18200	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow
Type C: 0 ... +2315 °C 32 ... 2786.5 °F 273.2 ... 2093.2 K (0xB7: ext. comp. 0 °C) (0xC7: int. comp. 0 °C)	+25000	32766	23432	Overflow
	0 ... +23150	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow
Type E: -270 ... +1000 °C -454 ... 1832 °F 0 ... 1273.2 K (0xB8: ext. comp. 0 °C) (0xC8: int. comp. 0 °C)	+12000	21920	14732	Overflow
	-2700 ... +10000	-4540 ... 18320	0 ... 12732	Nominal range
	-	-	-	Underflow

Measuring range	Measured value			Range
(Fct. no.)	[°C]	[°F]	[°K]	
Type L: -200 ... +900 °C -328 ... 1652 °F 73.2 ... 1173.2 K (0xB9: ext. comp. 0 °C) (0xC9: int. comp. 0 °C)	+11500	21020	14232	Overflow
	-2000 ... +9000	-3280 ... 16520	732 ... 11732	Nominal range
	-	-	-	Underflow

8.5 Parameterising the counter

8.5.1 1 counter 32 bits, 24 V DC - EPM-S600

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter a differentiation between the internal gate (I-gate), hardware gate (HW gate), and software gate (SW gate) is made. • The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). • The SW gate is controlled via your user program (status word in the output area). • The HW gate is controlled via the digital gate input. The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Latch function	If a positive edge occurs at the latch input, the current count value is stored in the latch register. The latch register is accessed via the input area. After a STOP-RUN transition, latch is always 0.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.

Additional functions	Description
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Further information: Product description (15)

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
2	Input frequency Track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
3	Input frequency Track B		0x02
4	Input frequency Latch		0x02
5	Input frequency Gate		0x02
6	Input frequency Reset		0x00
7	Reserved		
8	Alarm response	Setting activates process alarm Bit 0: Proc. alarm HW gate open Bit 1: Proc. alarm HW gate closed Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 6: Proc. alarm latch value Bit 7: Reserved	0x80
9	Numerator function	Bit 5 ... 0: 0b0000000 = continuous counting 0b000001 = single counting, main counting direction is forward 0b000010 = single counting, main counting direction is backward 0b000100 = single counting, no main counting direction 0b001000 = periodic counting, main counting direction is forward 0b010000 = periodic counting, main counting direction is backward 0b100000 = periodic counting, no main counting direction Bit 7 ... 6: Reserved	0x40

Position in string No.	Name	Description/value	Lenze
10	Comparator	Bit 2 ... 0: output switches (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 ... 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00
11	Signal evaluation	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Hardware gate (HW gate) • 0b000 = deactivated (counter starts by setting SW gate) • 0b001 = activated (HIGH level at gate activates the HW gate. Counter starts if HW and SW gate are set.) Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	
12	Final value	Upper limitation of the counting range	0x00
13	Start value	Lower limitation of the counting range	0x00
14	Hysteresis	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
15	Pulse	The pulse duration indicates for how long the output is to be set if the parameterised comparison criterion is reached or exceeded. The pulse duration can be specified in steps of 2.048 ms between 0 and 522.24 ms. If the pulse duration is = 0, the output is set until the comparison condition is no longer met.	0x00

8.5.2 2 counters 32 bits, 24 V DC - EPM-S601

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Additional functions	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Signal evaluation	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Further information: ▶ Product description (📖 15)

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
2	Input frequency Counter 1, track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
3	Input frequency Counter 1, track B		0x02
4	Input frequency Counter 2, track A		0x02
5	Input frequency Counter 2, track B		0x02
6	Alarm response counter 1	Setting activates process alarm Bit 1 ... 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 ... 6: Reserved	0x00
7	Counter function counter 1	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 ... 6: Reserved	0x00
8	Comparator counter 1	Bit 2 ... 0: Comparison bit is set (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00

Position in string No.	Name	Description/value	Lenze
9	Signal evaluation counter 1	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
10	Set value counter 1	When a set value is given, the counter can be loaded with the set value. With an edge 0-1 at COUNTERVAL_SET in the control word, the set value is accepted in the counter.	0x00
11	Final value counter 1	Upper limitation of the counting range	0x00
12	Loading value counter 1	Lower limitation of the counting range	0x00
13	Hysteresis counter 1	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
14	Reserved		
15	Alarm response counter 2	See counter 1	0x00
16	Counter function counter 2	See counter 1	0x00
17	Comparator counter 2	See counter 1	0x00
18	Signal evaluation counter 2	See counter 1	0x00
19	Set value counter 2	See counter 1	0x00
20	Final value counter 2	See counter 1	0x00
21	Loading value counter 2	See counter 1	0x00
22	Hysteresis counter 2	See counter 1	0x00

8.5.3 1 counter 32 bits, 5 V DC - EPM-S602

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Further information: ▶ Product description (📖 15)

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
2	Input frequency Track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
3	Input frequency Track B		0x02
4	Input frequency Reset		0x02
5	Reserved		
6	Alarm response	Setting activates process alarm Bit 1 ... 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 ... 6: Reserved	0x00
7	Numerator function	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 ... 6: Reserved	0x00
8	Comparator	Bit 2 ... 0: Comparison bit is set (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 ... 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00

Position in string No.	Name	Description/value	Lenze
9	Signal evaluation	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
10	Final value	Upper limitation of the counting range	0x00
11	Start value	Lower limitation of the counting range	0x00
12	Hysteresis		0x00

8.5.4 2 counters 32 bits, 24 V DC - EPM-S603

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from 0 to the counting limit, then skips to the opposite counting limit and continues to count from there.

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area).

Further information: ▶ Product description (📖 15)

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Input frequency Counter 1, track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
2	Input frequency Counter 1, track B		0x02
3	Input frequency Counter 2, track A		0x02
4	Input frequency Counter 2, track B		0x02
5	Counting direction counter 1, track B	Bit 2 ... 0: Reserved Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
6	Signal evaluation counter 1	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 7 ... 3: Reserved	0x00

Position in string No.	Name	Description/value	Lenze
7	Counting direction counter 1, track B	Bit 2 ... 0: Reserved Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
8	Signal evaluation counter 1	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 7 ... 3: Reserved	0x00

8.6 Parameterising the encoder evaluation

8.6.1 SSI - EPM-S604

Parameter data

Position in string No.	Name	Description/value	Lenze
1	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
2	Idle time	The dead time, also called tbs (time between sends), defines the waiting time between two encoder values to be observed by the module so that the encoder is able to process its value. This data can be found in the data sheet for your encoder. HIGH LOW 0x00 0x30: 1 μs 0x00 0x60: 2 μs 0x00 0xC0: 4 μs 0x01 0x80: 8 μs 0x03 0x00: 16 μs 0x06 0x00: 32 μs 0x09 0x00: 48 μs 0x0C 0x00: 64 μs	0x0C00
3	Baud rate	In "monitoring operation" operating mode, the baud rate is irrelevant. Enter the baud rate here. This corresponds to the clock frequency via which the encoder connected communicates. Information on this can be found in the data sheet for your encoder. HIGH LOW 0x00 0x18: 2 MHz 0x00 0x20: 1.5 MHz 0x00 0x30: 1 MHz 0x00 0x60: 500 kHz 0x00 0xC0: 250 kHz 0x01 0x80: 125 kHz	0x0180
4	Reserved		
5	Standardisation	Depending on the encoder, further bits are transmitted in addition to the encoder value. Scaling serves to determine how many bits post-positioned to the encoder value will be removed by shifting the encoder value to the right. The encoder value is scaled by the module only after a Gray-binary conversion. More information can be found in the data sheet for your encoder. Value range: 0x00 ... 0x0F = 0 bit ... 15 bits	0x00

Position in string No.	Name	Description/value	Lenze
6	Bit length of encoder data	Enter the bit length of the encoder data here. Depending on the encoder, the encoder data consist of the current encoder value with subsequent bits. The total length has to be specified here. More information on this can be found in the data sheet for your encoder. 7 (0x07) = "8 bits" 8 (0x08) = "9 bits" 9 (0x09) = "10 bits" 10 (0x0A) = "11 bits" 11 (0x0B) = "12 bits" 12 (0x0C) = "13 bits" 13 (0x0D) = "14 bits" 14 0x0E) = "15 bits" 15 (0x0F) = "16 bits" 16 (0x10) = "17 bits" 17 (0x11) = "18 bits" 18 (0x12) = "19 bits" 19 (0x13) = "20 bits" 20 (0x14) = "21 bits" 21 (0x15) = "22 bits" 22 (0x16) = "23 bits" 23 (0x17) = "24 bits" 24 (0x18) = "25 bits" 25 (0x19) = "26 bits" 26 (0x1A) = "27 bits" 27 (0x1B) = "28 bits" 28 (0x1C) = "29 bits" 29 (0x1D) = "30 bits" 30 (0x1E) = "31 bits" 31 (0x1F) = "32 bits"	0x18

Position in string No.	Name	Description/value	Lenze
7		Bit 1 ... 0: Ready for operation During "Monitoring operation" the module serves to monitor the data exchange between an SSI master and an SSI encoder. It receives the cycle by the master and the data flow by the SSI encoder. In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder. 0b01 = monitoring operation 0b10 = master mode Bit 2: Shifting direction Specify the orientation of the encoder data here. More information can be found in the data sheet for your encoder. Usually, the SSI encoder uses "MSB first". 0 = LSB first (LSB is transmitted first) 1 = MSB first (MSB is transmitted first) Bit 3: Edge clock signal Here you can specify the edge type of the clock signal, in the case of which the encoder supplies data. Information on this can be found in the data sheet for your encoder. Usually the SSI encoders respond to rising edges. 0 = falling edge 1 = rising edge **Master mode:** Connect clock output signal (ClockOut) to the EPM-S604. **Monitoring mode:** Connect clock input signal (ClockIn) to the EPM-S604. Bit 4: Coding In the "Binary code" setting, the provided encoder value remains unchanged. In the "Gray code" setting, the Gray-coded value provided by the encoder is converted into a binary value. Only after this conversion, the received encoder value is scaled, if required. The Gray code is another form of representation of the binary code. It is based on the fact that two adjacent Gray numbers differ from each other in exactly one bit. If the Gray code is used, transmission errors can be easily detected, since adjacent characters must only differ from each other in one digit. Information on this can be found in the data sheet for your encoder. 0 = standard code 1 = Gray code Bit 7 ... 5: Reserved	0x1E
8	Reserved		
9	SSI function	By enabling the SSI function the module starts with the cycle output and the evaluation of the encoder data. In the "monitor operation" mode, the module starts with the encoder evaluation. 0 (0x00) = inhibited 1 (0x01) = enabled	0x00

Further information: ▸ Product description (📖 15)

8.7 Parameterising the time stamp

8.7.1 2 digital inputs with time stamp function - EPM-S207

Parameter data

Position in string No.	**Name**	**Description/value**	**Lenze**
1	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C (fix)
2	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)
3	Input delay DI1	0x00 = 1 µs 0x02 = 3 µs 0x04 = 10 µs 0x07 = 86 µs 0x09 = 342 µs 0x0C = 273 µs No other values are permissible.	0x02
4	Input delay DI2		0x02
5	Edge 0-1 at DI x	Time stamp entry on rising edge Bit 0: DI1 (0: inhibit, 1 = enable) Bit 1: DI2 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00
6	Edge 1-0 on DI x	Time stamp entry on falling edge Bit 0: DI1 (0: inhibit, 1 = enable) Bit 1: DI2 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00

8.7.2 2 digital outputs with time stamp function - EPM-S310

Parameter data

Position in string No.	**Name**	**Description/value**	**Lenze**
1	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C (fix)
2	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)

8.8 Parameterising technology modules

8.8.1 2 digital outputs with PWM functionality - EPM-S620

Parameter data

Position in string No.	Name	Description/value	Lenze
1	PWM 1: period	Set parameters here for the total time for pulse duration and pulse pause. The time should be selected as a factor for the 20.83 ns basis. Values below 25 µs are ignored. If the pulse duration is higher or equal to the period, the DO output is set permanently. Value range: 1200 ... 8388607 (25 µs ... approx. 175 ms)	0x1F40
2	PWM 2: Period		0x1F40

8.8.2 RS232 interface - EPM-S640

Parameter data - ASCII protocol

Position in string No.	Name	Description/value	Lenze
1	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
2	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
3	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
4	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
5	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x01: ASCII	0x01
Option 1, drawing frame			

Option 1, drawing frame			

Parameter data STX/ETX protocol

Position in string No.	Name	Description/value	Lenze
1	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
2	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
3	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
4	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
5	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame			
6	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			
7	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
8	ZNA (LOW byte)		0x00

Position in string No.	Name	Description/value	Lenze
Option 4 / 5, TMO			
9	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
10	TMO (LOW byte)		0xFA
Option 6, number of start identifiers			
10	Number of start identifiers	0x00: 1 start identifier (2. start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1			
11	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2			
12	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers			
13	Number of end identifiers	0x00: 1 end identifier (2. end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1			
14	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2			
15	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved			
16	Reserved		0x00

Parameter data 3964(R) protocol

Position in string No.	Name	Description/value	Lenze
1	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
2	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
3	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
4	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
5	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame			
6	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)			

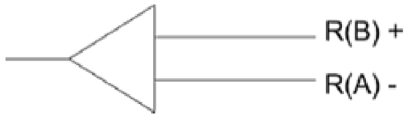
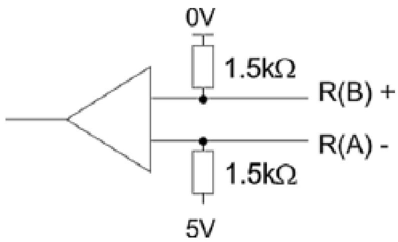
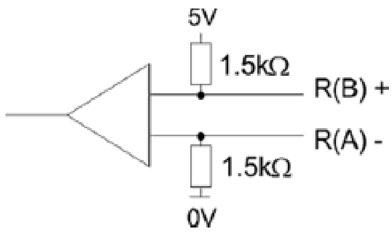
Position in string No.	Name	Description/value	Lenze
7	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 3, character delay time (x 20 ms)			
8	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)			
9	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)			
10	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repititons			
11	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL			
12	DBL	If the block wait time is exceeded, you can select the maximum number of repititions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority			
13	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved			
14 ... 17	Reserved		0x00

8.8.3 RS422/RS485 interface - EPM-S650

Parameter data - ASCII protocol

Position in string No.	Name	Description/value	Lenze
1	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
2	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
3	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
4	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
5	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x01: ASCII	0x01
Option 1, drawing frame			
6	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			

Position in string No.	**Name**	**Description/value**	**Lenze**
7	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
8	ZNA (LOW byte)		0x00
Option 4 / 5, character delay time			
9	Character delay time (HIGH byte)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
10	Character delay time (LOW byte)		0xFA
Option 6, number of receive buffers			
11	Number of receive buffers	Defines the number of receive buffers. As long as only one receive buffer is used and is assigned, no more data can be received. If up to 250 receive buffers are connected, the received data can be redirected to a still free receive buffer. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7 ... 12, reserved			
12 ... 17	Reserved		0x00
Option 13, operating mode			
18	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

Position in string No.	Name	Description/value	Lenze
Option 14, cable assignment			
19	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible. 	0x00

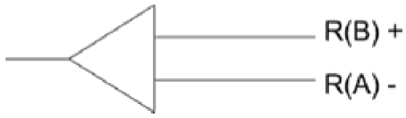
Parameter data STX/ETX protocol

Position in string No.	Name	Description/value	Lenze
1	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
2	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
3	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
4	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
5	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame			
6	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA			
7	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
8	ZNA (LOW byte)		0x00

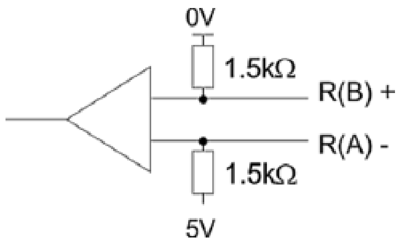
Position in string No.	Name	Description/value	Lenze
Option 4 / 5, TMO			
9	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
10	TMO (LOW byte)		0xFA
Option 6, number of start identifiers			
10	Number of start identifiers	0x00: 1 start identifier (2. start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1			
11	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2			
12	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers			
13	Number of end identifiers	0x00: 1 end identifier (2. end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1			
14	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2			
15	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved			
16	Reserved		0x00
Option 13, operating mode			
17	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

<table>
<tr><th>Position in string
No.</th><th>Name</th><th>Description/value</th><th>Lenze</th></tr>
<tr><td colspan="4">Option 14, cable assignment</td></tr>
<tr><td>18</td><td>Cable assignment</td><td>
For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level.

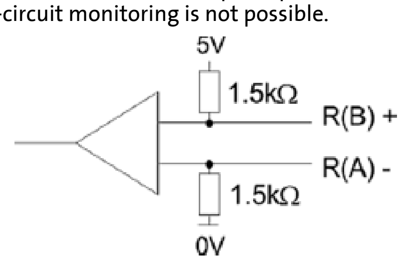
- 0x00:
  No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.



- 0x01:
  Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V
  In full duplex operation under RS422, open-circuit monitoring is possible.



- 0x02:
  Signal R(A) 0V, signal R(B) 5V
  This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485.
  Open-circuit monitoring is not possible.

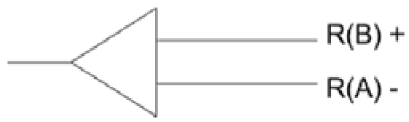
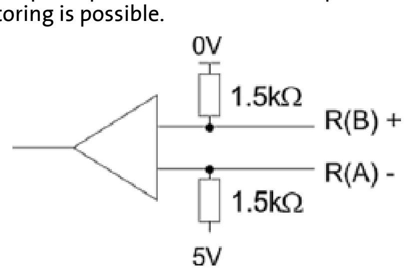
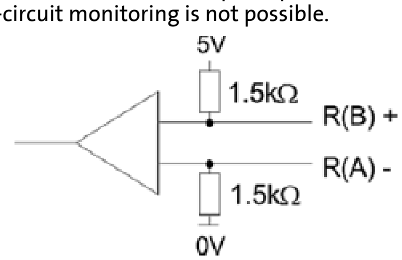


</td><td>0x00</td></tr>
</table>

Parameter data 3964(R) protocol

Position in string No.	Name	Description/value	Lenze
1	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
2	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
3	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
4	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
5	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame			
6	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
		Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
		Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
		Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)			

Position in string No.	Name	Description/value	Lenze
7	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 3, character delay time (x 20 ms)			
8	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)			
9	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)			
10	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repititons			
11	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL			
12	DBL	If the block wait time is exceeded, you can select the maximum number of repititions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority			
13	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved			
14 ... 17	Reserved		0x00
Option 13, operating mode			

Position in string No.	**Name**	**Description/value**	**Lenze**
18	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01
Option 14, cable assignment			
19	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible. 	0x00

9 PROFINET communication

9.1 About PROFINET

PROFINET is an open industrial Ethernet standard by PROFIBUS & PROFINET International (PI) for automation technology. PROFINET is standardised in IEC 61158.

PROFINET uses TCP/IP and IT standards and complements the PROFIBUS technology for applications where fast data communication in combination with industrial IT functions is required.

There are two PROFINET command classes which can be implemented in three performance steps:

- PROFINET IO
  - RT communication
  - IRT communication
- PROFINET CBA (is not supported by the EPM-S140 bus coupler)
  - TCP/IP communication

PROFINET IO

PROFINET IO describes an I/O data view on decentralised peripherals. PROFINET IO describes the entire data exchange between the I/O controller and the I/O device. In the configuration, PROFINET IO is based on PROFIBUS.

PROFINET IO always features the real time concept. PROFINET IO uses a Provider/Consumer model in contrast to the master/slave method under PROFIBUS. This supports the communication relations (AR = Application Relation) between the equal nodes at the Ethernet. Here, the provider transmits its data without a request of the communication partner. In addition to the user data exchange, functions for parameterisation and diagnostics are supported as well.

RT communication

RT stands for Real Time. RT communication is the basis for data exchange with PROFINET IO. Here, RT data is treated with higher priority.

IRT communication

IRT stands for Isochronous Real Time. With IRT communication, the bus cycle starts in a true-to-cycle mode, i.e. with a max. permissible deviation and is consistently synchronised. This ensures the time-controlled and cycle-synchronous transfer of data. Here, sync frames of a sync master in the network provide for synchronisation.

PROFINET performance features

PROFINET according to IEC 61158 has the following performance features:

- Full duplex transmission with 100 Mbit/s via copper cables or optical fibre
- Switched Ethernet
- Auto negotiation (negotiating the transmission parameters)
- Auto crossover (transmit and receive path are automatically crossed if required)
- Wireless communication via Bluetooth or WLAN
- UDP/IP is used as higher-level protocol. UDP stands for User Datagram Protocol and comprises the unsecured connectionless broadcast communication in combination with IP.

PROFINET devices

As with PROFIBUS-DP, the following devices are classified according to their tasks with PROFINET IO as well:

- IO-Controller
- IO-Device
- IO-Supervisor

The IO-Controller is equivalent to the master under PROFIBUS. Here, it is the PLC with PROFINET connection in which the automation program is executed.

An IO-Device is a decentralised I/O field device which is connected via PROFINET. The IO-Device is equivalent to the slave under PROFIBUS.

An IO-Supervisor is an engineering station as for example a programming device, a PC or a control panel for commissioning and diagnostics.

Industrial Ethernet

Due to the openness of PROFINET standard, you can use standard Ethernet components. For industrial environments and due to the high baud rate of 100 Mbps, the PROFINET system should consist of industrial Ethernet components. All devices connected via switches are located in one network and can communicate directly with each other. A network is physically limited by a router. For communication via network limits, you must program your routers in such a way that they permit this communication.

Topology

Line: For the line structure, all nodes are connected in series. The line structure is achieved via switches which have already been implemented in the PROFINET devices. If a node fails, communication over the failed node is not possible.

Star: When nodes are connected to a switch with more than 2 PROFINET interfaces, a star-shaped network topology is automatically formed. If a single PROFINET device fails, this does not cause a complete network failure as in case of the other structures. The switch failure only causes a failure of the sub-network.

Ring: In order to increase the availability, you can connect both open ends of a line structure via a switch. When you parameterise the switch as redundancy manager, it will make the data be transmitted via an intact network connection in the case of power failure.

Tree: When several star-shaped structures are interconnected, a tree-shaped network topology is formed.

GSDML file

Lenze provides you with a GSDML file for your IO-Device. This file is either located on the enclosed data medium or in the download area of www.lenze.de.

Install the GSDML file in your project planning tool. More information on how to install the GSDML file can be found in the manual of your project planning tool. For configuration in your project planning tool, the GSDML file contains all system modules as XML data.

Addressing

In contrast to the PROFIBUS address, each device in PROFINET can be identified non-ambiguously via its PROFINET interface:

- IP address or MAC address
- Device name

Transmission medium

PROFINET is Ethernet-compatible according to the IEEE standards. The PROFINET IO field devices are exclusively connected via switches as network components. It is either carried out as a star-shaped structure via multiple port switches or line-shaped by means of switched implemented in the field device.

9.2 Access to the I/O system 1000

In the following, the access under PROFINET to the following ranges of the I/O system 1000 are displayed:

- I/O area
- Parameter data
- Diagnostic data

Note!

Please note that the supply and terminal modules do not have any type identification. They cannot be identified by the PROFINET coupler and are thus not considered in the listing or assignment of the slots.

In the following, slots within PROFINET are called PROFINET slots. Counting always starts at 0.

GSDML file

For configuring a Device-I/O interface connection in your own project planning tool, you get the performance features of the PROFINET components in the form of a GSDML file.

This file is either located on the enclosed data medium or in the download area of www.lenze.de. Install this GSDML file in your project planning tool.

More information on how to install the GSDML file can be found in the manual for your project planning tool. Structure and contents of the GSDML file are defined by the IEC 61158 standard.

Handling blocks

For accepting or changing data records during runtime, two corresponding handling blocks for reading/writing data records are required. For CPUs to be programmed with STEP7 by Siemens, the following handling blocks are available:

- SFB 52 Read data record
- SFB 53 Write data record
- SFB 54 Read diagnostic data

With "slot" you address the module and via "index" you address the data area related to a module.

Acyclic access to the I/O system

For an acyclic write and read access, PROFINET uses appropriate frames. Here, the PROFINET coupler or the module is addressed via slot (0 ... 64) and the corresponding data area within the module via index. Subslot is always 1.

- Read access

Request frame (ReadRequest)							
0x0009 +0	...	API +24	Slot +28	Subslot +30	Index +34	Length +36	... ... +64

Response with data (ReadResponse)							
0x8008 +0	...	API +24	Slot +28	Subslot +30	Index +34	Length +36	... ... +63

- Write access

Request frame (WriteRequest)								
0x8008 +0	...	API +24	Slot +28	Subslot +30	Index +34	Length +36	... ... +63	Data +64 ...

Response with length (WriteResponse)							
0x0008 +0	...	API +24	Slot +28	Subslot +30	Index +34	Length +36	... ... +64

9.2.1 Access to the I/O area

With PROFINET, the input/output range is automatically shown in the corresponding address range of the master system. The following index no. also provides access to the I/O ranges:

- Index = 0x8028: Reading of the input data (slot 1 ... 64)
- Index = 0x8029: Reading of the output data (slot 1 ... 64)

9.2.2 Access to parameter data

The GSDML file serves to set parameter data for the corresponding modules via the hardware configuration. When the I/O device starts, the parameter data is given once to the modules via the I/O controller. After writing, the parameter data in the module is active.

Access to	Slot	Index
All parameters of the PROFINET coupler incl. header (4 bytes)	0	0x007D
All parameters of the PROFINET coupler	0	0x007E
All module parameters incl. header (4 bytes)	1 ... 64	0x007D
Data record DS 0x00 of the module parameters	1 ... 64	0x007E
Data record DS 0x01 of the module parameters	1 ... 64	0x007F
Data record DS 0x80 ... 0x90 of the module parameters	1 ... 64	0x0080 ...0x0090

9.2.3 Access to diagnostic data

Alarm-capable I/O compound modules automatically send process alarm data or diagnostic data via the diagnostics frame if the alarm has been activated via parameterisation.

Another option is to request the diagnostic data. In this case, address the PROFINET bus coupler or the module via slot (0 ... 64) and the corresponding data area via the index.

Diagnostic data of PROFINET bus coupler

Slot = 0 / Subslot = 1 serves to access the PROFINET coupler. Depending on the index, you receive the following data:

Index = 0x0000: 4Byte: Byte 0: Diagnostic byte, byte 1 ... 3: 0 (fix)
Index = 0x0001: 20Byte: Byte 0: Diagnostics byte, byte 1 ... 19: 0 (fix)

Structure of diagnostic data of PROFINET bus coupler		
Byte	**Bit 7 ... 0**	**Lenze**
0	Diagnostic byte Bit 0: Error at backplane bus Bit 1: Parameters have been rejected by the addressed module or coupler (error in data consistency) Bit 2: General bus coupler parameter error (data could not be saved) Bit 3: Version error at the backplane bus (at least one module at the backplane bus is not supported) Bit 5, 4: 0 (fix) Bit 6: Port error with activated port monitoring Bit 7: Configuration error backplane bus (actual configuration unequal to setpoint configuration)	0x00
2 ... 3 (19)	0x00 (fix)	0x00

Diagnostic data of I/O compound module

Slot = 1 ... 64 / Subslot = 1 serves to access the corresponding I/O compound module. Depending on the index, you receive the following data:

- Index = 0x0000: Data record DS 0x00 of the diagnostic data
- Index = 0x0001: Data record DS 0x01 of the diagnostic data

Note!

Information on how to assign the ranges can be found in the descriptions of the corresponding I/O compound module.

Structure of diagnostic data of I/O compound module		Lenze
Byte	**Bit 7 ... 0**	
0	Bit 0: Module fault, i.e. an error has been detected Bit 1: Internal error in the module Bit 2: External error - module cannot be addressed anymore Bit 3: Channel error in module Bit 4: External supply voltage is missing Bit 5, 6: Reserved Bit 7: Parameterisation error	
1	Bit 3 ... 0: Module class 1111: Digital module 0101: Analog module 1000: Counter module, SSI module 0111: Time stamp module, gateway module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)	
2	see module description	
3	Bit 5 ... 0: Reserved Bit 6: Process alarm lost Bit 7: Reserved	
4	Channel type 0x70: Module with digital inputs 0x71: Module with analog inputs 0x72: Module with digital outputs 0x73: Module with analog outputs 0x74: Module with analog inputs/outputs 0x77: Counter	
5	Number of diagnostic bits per channel	
6	Number of channels per module	
7	Position (channel) of the diagnostic event	
8	Diagnostic event for channel/channel group 0 For assignment see module description	
9	Diagnostic event for channel/channel group 1 For assignment see module description	
...		
15	Diagnostic event for channel/channel group 7 For assignment see module description	
16 ... 19	Value of the μs ticker when diagnostic data is generated (The I/O compound module features a timer that is started with mains ON and restarts with 0 after 232-1 μs.)	

9.3 Project planning

Project planning is carried out as hardware configuration in your PROFINET configuring tool as for example the Siemens SIMATIC Manager. Here, you assign your I/O controller to the appropriate I/O device. A direct assignment is made via the PROFINET address that can be set at the I/O device via the address switch and the in I/O device properties.

By implementing the corresponding GSDML file, the PROFINET I/O bus coupler EPM-S140 is specified under:

PROFINET IO -> More field devices -> I/O > I/O system 1000

GSDML file

Lenze provides you with a GSDML file for the I/O device. This file is either located on the enclosed data medium or in the download area of www.lenze.de. Install the GSDML file in your configuring tool. More information on how to install the GSDML file can be found in the manual of your configuring tool.

For operating with your configuring tool, the GSDML file contains all I/O compound modules as XML files.

After installing the GSDML file, you can find the I/O system 1000 in the hardware catalogue of Siemens under:

PROFINET IO > More field devices > I/O > I/O system 1000

Commissioning

- Set up your PROFINET system.
- Start your configuring tool with a new project.
- Configure a master system and create a new
- PROFINET subnetwork.
- For configuring the bus coupler, select the "EPM-S140" from the hardware catalogue and drag it to the PROFINET subnetwork.
- As soon as all switches of the address switch have the 0 status, the "DeviceName" can be assigned freely via the properties of the PROFINET bus coupler. Otherwise assign a PROFINET name via the switch position of the address switch.
- If required, parameterise the I/O device.
- Transfer your project into the PLC.

Parameter data of PROFINET bus coupler

Parameter data of PROFINET bus coupler		Lenze
Byte	**Bit 7 ... 0**	

0	Bit 0: Process alarm
	0 = inhibit
	1 = enable
	Bit 1: Diagnostic alarm
	0 = inhibit
	1 = enable
	Bit 2: #Diagnostic alarm type
	0 = manufacturer-specific data
	1 = channel-specific data
	Bit 3 ... 6: Reserved
	Bit 7: Data format
	0 = Data format Motorola
	1 = Data format Intel
		0x00

2 ... 6	0x00 (fix)	0x00

Diagnostic alarm type Here you can determine the structure of the diagnostic alarm data that are sent via the diagnostic frame in the event of an error or be used to access standard PROFINET index numbers.

Mnufacturer-specific data: You always obtain the data record DS 0x01 of the diagnostic data of a module.

Channel-specific data: You always obtain the data record DS 0x00 of the diagnostic data of a module.

Data format Motorola/Intel: This parameter refers to how a value is stored in the CPU address range.

In the Motorola format (default), the bytes are stored in descending order, i.e. the first byte contains the High byte and the second byte contains the Low byte.

In the Intel format, the bytes are stored in ascending order, i.e. the first byte contains the Low byte and the second byte contains the High byte.

9.4 I&M data

Identification and maintenance data (I&M) are information stored in a module which supports you in:

- Checking the system configuration
- Locating hardware changes in a system
- Eliminating errors in a system

Identification data (I data)are information on the module, as for instance order number and serial number, which are partly printed on the module housing. I data is manufacturer information on the module and can only be read.

Maintenance data (M data)are system-dependent information, as for instance mounting place and date of installation. M data is created during the project planning phase and written onto the module.

The I&M data serves to unambiguously identify modules online.

Via "read data record" you can access certain identification data. Here, you address parts of the identification data via the corresponding index.

The data records have the following structure:

Contents	Length (byte)	Coding (hex)
Head information		
- BlockType	2	I&M0: 0020 I&M1: 0021 I&M2: 0022 I&M3: 0023
- BlockLength	2	I&M0: 0038 I&M1: 0038 I&M2: 0012 I&M3: 0038
- BlockVersionHigh	1	01
- BlockVersionLow	4	00
Identification data (see the following table)	I&M0 / Index 0xAFF0: 0x54 I&M1 / Index 0xAFF1: 0x54 I&M2 / Index 0xAFF2: 0x16 I&M3 / Index 0xAFF3: 0x54	

I&M data for PROFINET-IO

Identification data	Access	Lenze	Description
Identification data 0: (index 0xAFF0)			
VendorIDHigh	Read (1 byte)	0x02	Name of the manufacturer
VendorIDLow	Read (1 byte)	0x2B	
Order_ID	Read (20 bytes)		Order number
IM_SERIAL_NUMBER	Read (16 bytes)	-	Serial number
IM_HARDWARE_REVISION	Read (2 bytes)	1	HW output version
IM_SOFTWARE_REVISION • SWRevisionPrefix • IM_SWRevision_Functional_Enhancement • IM_SWRevision_Bug_Fix • IM_SWRevision_Internal_Change	read (1 byte) (1 byte) (1 byte) (1 byte)	 V, R, P, U, T 0x00 ... 0xFF 0x00 ... 0xFF 0x00 ... 0xFF Firmware version	
IM_REVISION_COUNTER	Read (2 bytes)	0x0000	for internal use
IM_PROFILE_ID	Read (2 bytes)	0x0000	for internal use
IM_PROFILE_SPECIFIC_TYPE	Read (2 bytes)	0x0005	for internal use
IM_VERSION • IM_Version_Major • IM_Version_Minor	read (1 byte) (1 byte)	0x0101 Version of the I&M data (z.B. 0x0101 = version 1.1)	
IM_SUPPORTED	Read (2 bytes)	0x000E	I&M1 ... I&M3 are available
Maintenance data 1: (index 0xAFF1)			
IM_TAG_FUNCTION	Read/write (32 bytes)	-	Selection of a clear identification throughout the system
IM_TAG_LOCATION	Read/write (22 bytes)	-	Selection of the mounting place
Maintenance data 2: (index 0xAFF2)			
IM_DATE	Read/write (16 bytes)	YYYY-MMDD HH:MM	Selection of an date
Maintenance data 3: (index 0xAFF3)			
IM_DESCRIPTOR	Read/write (54 bytes)	-	Selection of a comment

9.5 Index table

Within a module you can access the I/O data, parameter data and diagnostic data via index numbers. Under PROFINET the index numbers are summarised in the following areas:

0x0000 ... 0x7FFF: Manufacturer-specific index numbers
0x8000 ... 0xF7FF: Standard index numbers of PROFINET.

Information on this can be found in the PROFINET specification. There, the "index" is also called "data record". In the following, all supported index numbers are listed.

Index	Description
Readable index numbers	
0x0000	Read DS 0x00 diagnostic data
0x0001	Read DS 0x01 diagnostic data
0x007D	Read all parameter data
0x007E	Read DS 0x00 of the parameter data
0x007F (only I/O compound modules)	Read DS 0x01 of the parameter data
0x0080 . 0x0090 (only I/O compound modules)	DS 0x80 . Read DS 0x90 of the parameter data
0x8000 / 0x8001 / 0x800A / 0x800B / 0x800C / 0x8010 / 0x8011 / 0x8012 / 0x8013 / 0x801E / 0x802A / 0x802B / 0x802C / 0x802D / 0x802F / 0x8030 / 0x8031 / 0x8050 / 0x8051 / 0x8052 / 0x8053 / 0x8054 / 0x8060 / 0x8061 / 0x8062 / 0x8070 / 0x8080 / 0x8090	See PROFINET specification
0x8028 (only I/O compound modules)	Read input data of a sub-slot
0x8029 (only I/O compound modules)	Read output data of a sub-slot
0xAFF0	Read I&M 0 (serial no., name, SW/HW version)
0xAFF1 (only PROFINET bus coupler9)	Read I&M 1 (designation and mounting place)
0xAFF2 (only PROFINET bus Buskoppler9)	Read I&M 2 (date of installation)
0xAFF3 (only PROFINET bus Buskoppler9)	Read I&M 3 (comment)
0xC000 / 0xC001 / 0xC00A / 0xC00B / 0xC00C / 0xC010 / 0xC011 / 0xC012 / 0xC013 / 0xE000 / 0xE001 / 0xE002 / 0xE00A / 0xE00B / 0xE00C / 0xE010 / 0xE011 / 0xE012 / 0xE013 / 0xE030 / 0xE040 / 0xE050 / 0xF000 / 0xF001 / 0xF00A / 0xF00B / 0xF00C / 0xF010 / 0xF011 / 0xF012 / 0xF013 / 0xF020 / 0xF80C / 0xF820 / 0xF821 / 0xF830 / 0xF831 / 0xF840 / 0x8041 / 0xF842	See PROFINET specification
Writable index numbers	
0x007D	Write all parameter data
0x007E	Write DS 0x00 of the parameter data
0x007F (only I/O compound modules)	Write DS 0x01 of the parameter data
0x0080 . 0x0090 (only I/O compound modules)	Write DS 0x80 .DS 0x90 of the parameter data
0xAFF1 (only PROFINET bus coupler9)	Write I&M 1 (designation and mounting place)
0xAFF2 (only PROFINET bus Buskoppler9)	Write I&M 2 (date of installation)
0xAFF3 (only PROFINET bus Buskoppler9)	Write I&M 3 (comment)

9.6 Parameterising analog I/Os

9.6.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
128	0	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
129	0	Function channel 2		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x02))
6	Number of channels of a module (here 0x02)
7	Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10 ... 15	0 (fixed)

9.6.2 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
128	0	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
129	0	Function channel 2		0x20
130	0	Function channel 3		0x20
131	0	Function channel 4		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
12 ... 15	0 (fixed)

9.6.3 2 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S402

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set No.	Byte	Name	Description/value	Lenze
128	0	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
129	0	Function channel 2		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x02))
6	Number of channels of a module (here 0x02)
7	Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10 ... 15	0 (fixed)

9.6.4 4 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S403

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set No.	Byte	Name	Description/value	Lenze
128	0	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
129	0	Function channel 2		0x31
130	0	Function channel 3		0x31
131	0	Function channel 4		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded

9.6.5 2 analog inputs -10 ... +10 V (16 bits) - EPM-S406

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	1	Reserved	0	
	2	Limit value monitoring	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 (0 = deactivated; 1 = activated) Bits 7 ... 2: reserved	0x00
1	0	Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
128	0	Function channel 1	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
	1	Reserved	0	
	2	Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
	3	Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
129	0	Function channel 2	12 (0x12): -10 ... 10 V / -27648 ... 27648 34 (0x22): -10 ... 10 V / -16384 ... 16384 16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
	1	Reserved	0	
	2	Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
	3	Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
-10 ... +10 V (0x12)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-5	-13824	CA00		
	-10	-27648	9400		
	-11.76	-32512	8100	Underflow	
-10 ... +10 V (0x22)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-5	-8192	E000		
	-10	-16384	C000		
	-12.5	-20480	B000	Underflow	
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	-1.76	-4864	ED00	Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	-2	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
12 ... 15	0 (fixed)

9.6.6 2 analog inputs 0/4 ... 20 mA (16 bits) - EPM-S408

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set No.	Data set Byte	Name	Description/value	Lenze
0		Diagnostics	Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
		Reserved	0	
		Limit value monitoring	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 (0 = deactivated; 1 = activated) Bits 7 ... 2: reserved	0x00
1		Interference frequency suppression	Bit 1, 0: Channel 1 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 3, 2: Channel 2 00: Deactivated 01: 60 Hz 10: 50 Hz Bit 7 ... 4: Reserved	0x00
128		Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
		Reserved	0	
		Upper limit value channel 1	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
		Lower limit value channel 1	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000
129		Function channel 2	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x41
		Reserved	0	
		Upper limit value channel 2	Value from the rated range; if this value is exceeded and the limit value monitoring function is active, a process alarm is triggered. 0x7FFF: Limit value alarm deactivated	0x7FFF
		Lower limit value channel 2	Value from the rated range; if the actual value falls below this value and the limit value monitoring function is active, a process alarm is triggered. 0x8000: Limit value alarm deactivated	0x8000

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	**Current (I)** [mA]	**dec**	**hex**	**Range**	**Conversion**
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Measuring range exceeded
- Measuring range not reached

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x71: Analog input Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 5 ... 1: 0 (fixed) Bit 6: Measuring range not reached Bit 7: Measuring range exceeded
12 ... 15	0 (fixed)

9.6.7 2 analog outputs 0 ... 10 V (12 bits) - EPM-S500

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	1	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
128	0	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
129	0	Function channel 2		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Output area (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Short circuit/overload (if parameterised)

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x73: Analog output Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x02))
6	Number of channels of a module (here 0x02)
7	Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
10 ... 15	0 (fixed)

9.6.8 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set No.	Data set Byte	Name	Description/value	Lenze
0	1	Short-circuit detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
128	0	Function channel 1	16 (0x10): 0 ... 10 V / 0 ... 27648 32 (0x20): 0 ... 10 V / 0 ... 16384 255 (0xFF): Channel deactivated	0x20
129	0	Function channel 2		0x20
130	0	Function channel 3		0x20
131	0	Function channel 4		0x20

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Output area (Fct. no.)	Voltage (U) [V]	dec	hex	Range	Conversion
0 ... 10 V (0x10)	11.76	32511	7EFF	Overflow	U = D * 10 / 27648 D = 27648 * U / 10
	10	27648	6C00	Nominal range	
	5	13824	3600		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	
0 ... 10 V (0x20)	12.5	20480	5000	Overflow	U = D * 10 / 16384 D = 16384 * U / 10
	10	16384	4000	Nominal range	
	5	8192	2000		
	0	0	0000		
	Not possible, is limited to 0 V			Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Short circuit/overload (if parameterised)

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x73: Analog output Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Short circuit after M Bit 7 ... 4: 0 (fixed)
12 ... 15	0 (fixed)

9.6.9 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	1	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
128	0	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
129	0	Function channel 2		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Open circuit (if parameterised)

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x73: Analog output Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x02))
6	Number of channels of a module (here 0x02)
7	Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
10 ... 15	0 (fixed)

9.6.10 4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503

Parameter data

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	1	Wire-breakage detection	Bit 0: Channel 1 (0 = deactivated; 1 = activated) Bit 1: Channel 2 Bit 2: Channel 3 Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
128	0	Function channel 1	48 (0x30): 4 ... 20 mA / -4864 ... 32511 64 (0x40): 4 ... 20 mA / -3277 ... 20480 49 (0x31): 0 ... 20 mA / -4864 ... 32511 65 (0x41): 0 ... 20 mA / -3277 ... 20480 255 (0xFF): Channel deactivated	0x31
129	0	Function channel 2		0x31
130	0	Function channel 3		0x31
131	0	Function channel 4		0x31

With the formulas listed in the following you can convert a digital value into an analog output value and vice versa.

Measuring range (Fct. no.)	Current (I) [mA]	dec	hex	Range	Conversion
0 ... 20 mA (0x31)	23.52	32511	7EFF	Overflow	I = D * 20 / 27648 D = 27648 * I / 20
	20	27648	6C00	Nominal range	
	10	13824	3600		
	0	0	0000		
	-3.52	-4864	ED00	Underflow	
0 ... 20 mA (0x41)	25	20480	5000	Overflow	I = D * 20 / 16384 D = 16384 * I / 20
	20	16384	4000	Nominal range	
	10	8192	2000		
	0	0	0000		
	-4.00	-3277	F333	Underflow	
4 ... 20 mA (0x30)	22.81	32511	7EFF	Overflow	I = D * 16 / 27648 + 4 D = 27648 * (I-4) / 16
	20	27648	6C00	Nominal range	
	12	13824	3600		
	4	0	0000		
	1.19	-4864	ED00	Underflow	
4 ... 20 mA (0x40)	24	20480	5000	Overflow	I = D * 16 / 16384 + 4 D = 16384 * (I-4) / 16
	20	16384	4000	Nominal range	
	12	8192	2000		
	4	0	0000		
	0.8	-3277	F333	Underflow	

Diagnostic data

Since this module does not support an alarm, the diagnostic data serve to provide information on this module. In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

The following errors are registered in the diagnostic data:

- Configuration/parameterisation errors
- Open circuit (if parameterised)

Using SFB 52 you can access the diagnostic data of the module any time. Data set 1 has the following structure:

Byte	Bit 7 ... 0
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of a missing external supply voltage Bit 6: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b0101: Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x73: Analog output Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x04)
7	Bit 0: Channel error channel 1 Bit 1: Channel error channel 2 Bit 2: Channel error channel 3 Bit 3: Channel error channel 4 Bit 7 ... 4: 0 (fixed)
8	Channel-specific errors: channel 1: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
9	Channel-specific errors: channel 2: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
10	Channel-specific errors: channel 3: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
11	Channel-specific errors: channel 4: Bit 0: Configuration/parameterisation error Bit 2 ... 1: 0 (fixed) Bit 3: Open circuit Bit 7 ... 4: 0 (fixed)
12 ... 15	0 (fixed)

9.7 Parameterising temperature measurement

9.7.1 4(2) analog input for resistance measurement - EPM-S404

During the execution time you can access the parameter data via the following data sets:

Data set No.	Byte	Name	Description/value	Lenze
0	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
	1	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) : Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
	2	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) : Bit 3: Channel 4 Bit 4 ... 7: Reserved	0x00
	3	Reserved		
1	0	Temperature system	Bit 0, 1: 0b00 = °C; 0b01= °F; 0b10 = K Bit 2 ... 7: Reserved	0x00
	1	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02
Channel 1				
128	0	Function channel 1	80 (0x50) ... 162 (0xA2): see "measuring range" 255 (0xFF): Channel deactivated	0x50
	1	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At 50 Hz: 44.1 ms/channel 16 bits; at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x00
	2, 3	Upper limit value channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. As soon as the measured value is outside a limit value and limit value monitoring is activated, a process alarm is triggered.	0x7FFF
	4,5	Lower limit value channel 1		0x8000

Channel 1				
128	0	Function channel 1	80 (0x50) ... 162 (0xA2): see "measuring range" 255 (0xFF): Channel deactivated	0x50
	1	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At 50 Hz: 44.1 ms/channel 16 bits; at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x00
	2, 3	Upper limit value channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. As soon as the measured value is outside a limit value and limit value monitoring is activated, a process alarm is triggered.	0x7FFF
	4,5	Lower limit value channel 1		0x8000

Data set No.	Data set Byte	Name	Description/value	Lenze
Channel 4 (for two-wire conductor connections only)				
131	0	Function channel 4	See channel 1	0x50
	1	Conversion time channel 4	See channel 1	0x00
	2, 3	Upper limit value channel 4	See channel 1	0x7FFF
	4, 5	Lower limit value channel 4		0x8000

Measuring range

Measuring range (Fct. no.)	Measured value	Signal range	Range
2-wire: PT100 (0x50)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: PT1000 (0x51)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
2-wire: NI100 (0x52)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: NI1000 (0x53)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: PT100 (0x58)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: PT1000 (0x59)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
3-wire: NI100 (0x5A)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
3-wire: NI1000 (0x5B)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: PT100 (0x60)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow
4-wire: PT1000 (0x61)	+1000 °C	+10000	Overflow
	-200 ... +850 °C	-2000 ... +8500	Nominal range
	-243 °C	-2430	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
4-wire: NI100 (0x62)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
4-wire: NI1000 (0x63)	+295 °C	+2950	Overflow
	-60 ... +250 °C	-600 ... +2500	Nominal range
	-105 °C	-1050	Underflow
2-wire: 0 ... 60 Ω (0x70)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x71)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0x72)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0x78)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x79)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x7A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0x80)	-	-	Overflow
	0 ... 60 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0x81)	-	-	Overflow
	0 ... 600 Ω	0 ... 32767	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0x82)	-	-	Overflow
	0 ... 3000 Ω	0 ... 32767	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0x90)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0x91)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0x92)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
3-wire: 0 ... 60 Ω (0x98)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0x99)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0x9A)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xA0)	-	-	Overflow
	0 ... 60 Ω	0 ... 6000	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0xA1)	-	-	Overflow
	0 ... 600 Ω	0 ... 6000	Nominal range
	-	-	Underflow
4-wire: 0 ... 3000 Ω (0xA2)	-	-	Overflow
	0 ... 3000 Ω	0 ... 30000	Nominal range
	-	-	Underflow
2-wire: 0 ... 60 Ω (0xD0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 600 Ω (0xD1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
2-wire: 0 ... 3000 Ω (0xD2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 60 Ω (0xD8)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 600 Ω (0xD9)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow
3-wire: 0 ... 3000 Ω (0xDA)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 60 Ω (0xE0)	70.55 Ω	32511	Overflow
	0 ... 60 Ω	0 ... 27648	Nominal range
	-	-	Underflow
4-wire: 0 ... 600 Ω (0xE1)	705.5 Ω	32511	Overflow
	0 ... 600 Ω	0 ... 27648	Nominal range
	-	-	Underflow

Measuring range (Fct. no.)	Measured value	Signal range	Range
4-wire: 0 ... 3000 Ω (0xE2)	3528 Ω	32511	Overflow
	0 ... 3000 Ω	0 ... 27648	Nominal range
	-	-	Underflow

Diagnostics and alarm

Trigger	Process alarm	Diagnosealarm	parameterisable
Configuration/parameterisation errors	-	X	-
Open circuit detected	-	X	X
Measuring range exceeded	-	X	-
Measuring range not reached	-	X	-
Limit value exceeded	X	-	X
Limit value not reached	X	-	X
Process alarm lost	-	X	-

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: Limit value exceeded channel 1 Bit 1: Limit value exceeded channel 2 Bit 2: Limit value exceeded channel 3 Bit 3: Limit value exceeded channel 4 Bit 7 ... 4: 0 (fixed)
2	Bit 0: Limit value not reached channel 1 Bit 1: Limit value not reached channel 2 Bit 2: Limit value not reached, channel 3 Bit 3: Limit value not reached, channel 4 Bit 7 ... 4: 0 (fixed)
3/4	Ticker value at the time of the alarm After mains connection, a timer (μs ticker) is started, which after 65535 μs starts with 0 again.

Diagnosealarm

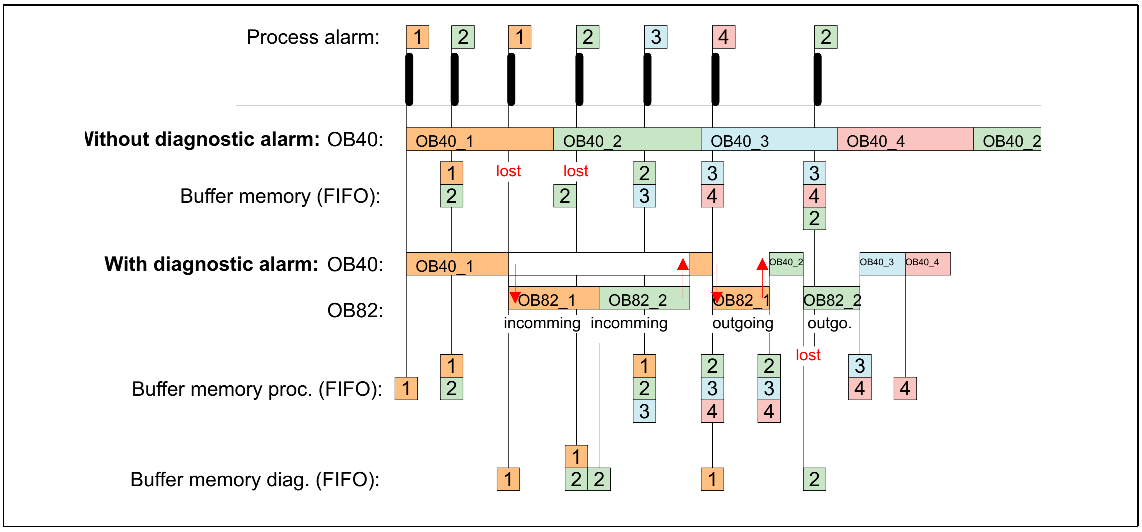
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x01)
7	Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed)
8	Diagnostic alarm due to process alarm lost to ... Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bit 7 ... 5: 0 (fixed)
9 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

9.7.2 2 analog inputs for thermocouple measurement - EPM-S405

During the execution time you can access the parameter data via the following data sets:

Data set		Name	Description/value	Lenze
No.	Byte			
0	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 0 ... 5: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
	1	Wire-breakage detection	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
	2	Limit value monitoring	Bit 0: Channel 1 (0 = inhibited; 1 = enabled) Bit 1: Channel 2 Bit 2 ... 7: Reserved	0x00
	3	Reserved	0	
1	0	Temperature system	Bit 0, 1: 0b00 = °C; 0b10 = °F; 0b11 = K Bit 2 ... 7: Reserved	0x00
	1	Interference frequency suppression	Bit 0, 1: 0b01 = 60 Hz; 0b10 = 50 Hz Bit 2 ... 7: Reserved	0x02
Channel 1				
128	0	Function channel 1	176 (0x60) ... 201 (0xC9): see "measuring range" External Temperature compensation: 176 (0x60) ... 185 (0x69) Internal temperature compensation: 192 (0xC0): ... 201 (0xC9) 255 (0xFF): Channel deactivated	0xC1
	1	Conversion time channel 1	You can set the transformer speed as a function of the interference frequency suppression (see 0x3105/x) for each channel. 0 (0x00): At 50 Hz: 324.1 ms/channel 16 bits; at 60 Hz: 270.5 ms/channel 16 bits 1 (0x01): at 50 Hz: 164.2 ms/channel 16 bits; at 60 Hz: 137.2 ms/channel 16 bits 2 (0x02): At 50 Hz: 84.2 ms/channel 16 bits; at 60 Hz: 70.5 ms/channel 16 bits 3 (0x03): At Hz: 44.1 ms/channel 16 bits at 60 Hz: 37.2 ms/channel 16 bits 4 (0x04): At 50 Hz: 24.2 ms/channel 16 bits; at 60 Hz: 20.5 ms/channel 16 bits 5 (0x05): At 50 Hz: 14.2 ms/channel 16 bits; at 60 Hz: 12.2 ms/channel 16 bits 6 (0x06): At 50 Hz: 9.2 ms/channel 16 bits; at 60 Hz: 8.0 ms/channel 16 bits 7 (0x07): At 50 Hz: 6.6 ms/channel 15 bits; at 60 Hz: 5.9 ms/channel 15 bits 8 (0x08): At 50 Hz: 4.2 ms/channel 13 bits; at 60 Hz: 3.8 ms/channel 13 bits	0x02
	2, 3	Upper limit value channel 1	You can define an upper or lower limit value for each channel. For this you can only specify values from the nominal range, otherwise you'll receive a parameterisation error. By specification of 0x7FFF for the upper or 0x8000 for the lower limit value, the corresponding limit value is deactivated. If the measured value is outside a limit value and the limit value monitoring is activated, a process alarm is triggered.	0x7FFF
	4, 5	Lower limit value channel 1		0x8000

Data set No.	Data set Byte	Name	Description/value	Lenze
Channel 2				
129	0	Function channel 2	See channel 1	0xC1
	1	Conversion time channel 2	See channel 1	0x02
	2, 3	Upper limit value channel 2	See channel 1	0x7FFF
	3, 4	Lower limit value channel 2		0x8000

Measuring range

Measuring range (Fct. no.)	Measured value [°C]	Measured value [°F]	Measured value [K]	Range
Type J: -210 ... +1200 °C, -346 ... 2192 °F, 63.2 ... 1473.2 K (0xB0: ext. comp. 0 °C) (0xC0: int. comp. 0 °C)	+14500	26420	17232	Overflow
	-2100 ... +12000	-3460 ... +21920	632 ... 14732	Nominal range
	-	-	-	Underflow
Type K: -210 ... +1372 °C, -454 ... 2501.6 °F, 0 ... 1645.2 K (0xB1: ext. comp. 0 °C) (0xC1: int. comp. 0 °C)	+16220	29516	18952	Overflow
	-2700 ... +13720	-4540 ... 25016	0 ... 16452	Nominal range
	-	-	-	Underflow
Type N: -270 ... +1300 °C, -454 ... 2372 °F, 0 ... 1573.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+15500	28220	18232	Overflow
	-2700 ... +13000	-4540 ... 23720	0 ... 15732	Nominal range
	-	-	-	Underflow
Type R: -50 ... +1769 °C, -58 ... 3216.2 °F, 223.2 ... 2042.2 K (0xB3: ext. comp. 0 °C) (0xC3: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type S: -50 ... +1769 °C, -58 ... 3216.2 °F, 223.2 ... 2042.2 K (0xB4: ext. comp. 0 °C) (0xC4: int. comp. 0 °C)	+20190	32766	22922	Overflow
	-500 ... +17690	-580 ... 32162	2232 ... 20422	Nominal range
	-1700	-2740	1032	Underflow
Type T: -270 ... +440 °C, -454 ... 752 °F, 3.2 ... 673.2 K (0xB2: ext. comp. 0 °C) (0xC2: int. comp. 0 °C)	+5400	10040	8132	Overflow
	-2700 ... +4000	-4540 ... 7520	32 ... 6732	Nominal range
	-	-	-	Underflow
Type B: 0 ... +1820 °C, 32 ... 2786.5 °F, 273.2 ... 2093.2 K (0xB6: ext. comp. 0 °C) (0xC6: int. comp. 0 °C)	+20700	32766	23432	Overflow
	0 ... +18200	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow

Measuring range (Fct. no.)	Measured value [°C]	[°F]	[K]	Range
Type C: 0 ... +2315 °C 32 ... 2786.5 °F 273.2 ... 2093.2 K (0xB7: ext. comp. 0 °C) (0xC7: int. comp. 0 °C)	+25000	32766	23432	Overflow
	0 ... +23150	320 ... 27865	2732 ... 20932	Nominal range
	-1200	-1840	1532	Underflow
Type E: -270 ... +1000 °C -454 ... 1832 °F 0 ... 1273.2 K (0xB8: ext. comp. 0 °C) (0xC8: int. comp. 0 °C)	+12000	21920	14732	Overflow
	-2700 ... +10000	-4540 ... 18320	0 ... 12732	Nominal range
	-	-	-	Underflow
Type L: -200 ... +900 °C -328 ... 1652 °F 73.2 ... 1173.2 K (0xB9: ext. comp. 0 °C) (0xC9: int. comp. 0 °C)	+11500	21020	14232	Overflow
	-2000 ... +9000	-3280 ... 16520	732 ... 11732	Nominal range
	-	-	-	Underflow

Diagnostics and alarm

Trigger	Process alarm	Diagnosealarm	parameterisable
Configuration/parameterisation errors	-	X	-
Open circuit detected	-	X	X
Measuring range exceeded	-	X	-
Measuring range not reached	-	X	-
Limit value exceeded	X	-	X
Limit value not reached	X	-	X
Process alarm lost	-	X	-

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: Limit value exceeded channel 1 Bit 1: Limit value exceeded channel 2 Bit 7 ... 2: 0 (fixed)
2	Bit 0: Limit value not reached, channel 1 Bit 1: Limit value not reached, channel 2 Bit 7 ... 2: 0 (fixed)
3/4	Ticker value at the time of the alarm After mains connection, a timer (µs ticker) is started, which after 65535 µs starts with 0 again.

Diagnosealarm

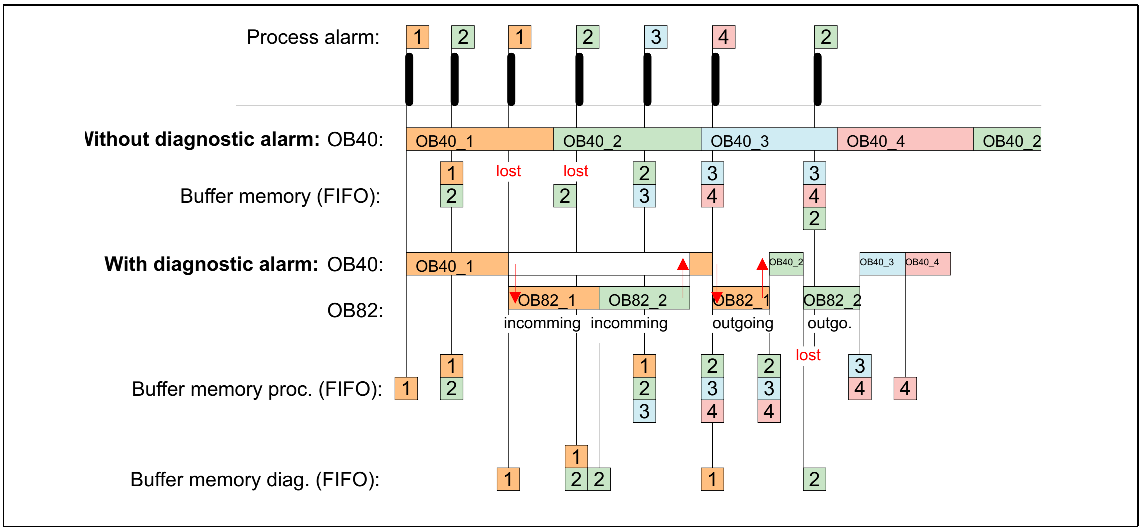
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x01)
7	Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed)
8	Diagnostic alarm due to process alarm lost to ... Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bit 7 ... 5: 0 (fixed)
9 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

9.8 Parameterising the counter

9.8.1 1 counter 32 bits, 24 V DC - EPM-S600

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter a differentiation between the internal gate (I-gate), hardware gate (HW gate), and software gate (SW gate) is made. • The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). • The SW gate is controlled via your user program (status word in the output area). • The HW gate is controlled via the digital gate input. The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Latch function	If a positive edge occurs at the latch input, the current count value is stored in the latch register. The latch register is accessed via the input area. After a STOP-RUN transition, latch is always 0.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.

Additional functions	Description
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Read data: 12 bytes

Input area		
Addr.	**Access**	**Assignment**
+0	Double word	Counter value
+4	Double word	Latch value
+8	Word	Status word
+10	Word	Ticker value

Count value: Current counter content

Latch value: If there is a positive edge at the latch input, the count value is stored here.

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the count value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

Write data: 10 bytes

Output area		
Addr.	**Access**	**Assignment**
+0	Double word	Comparison value
+4	Double word	Set value
+8	Word	Control word

Comparison value: Here you can specify a value which, by comparison with the current counter content, is able to influence the counter output or trigger a process alarm. The response of the output or of the process alarm can be parameterised.

Set value: With an edge change 0-1 of *COUNTERVAL_SET* in the control word, the set value is accepted in the counter.

▸ Counter modules EPM-S600 ... EPM-S603 – control and status words (📖 27)

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved	0x00
0x01	0	Input frequency Track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
	1	Input frequency Track B		0x02
	2	Input frequency Latch		0x02
	3	Input frequency Gate		0x02
	4	Input frequency Reset		0x00
	5	Reserved		

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	0	Alarm response	Setting activates process alarm Bit 0: Proc. alarm HW gate open Bit 1: Proc. alarm HW gate closed Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 6: Proc. alarm latch value Bit 7: Reserved	0x80
	1	Numerator function	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = single counting, main counting direction is forward 0b000010 = single counting, main counting direction is backward 0b000100 = single counting, no main counting direction 0b001000 = periodic counting, main counting direction is forward 0b010000 = periodic counting, main counting direction is backward 0b100000 = periodic counting, no main counting direction Bit 7 ... 6: Reserved	0x40
	2	Comparator	Bit 2 ... 0: output switches (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 ... 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00
	3	Signal evaluation	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Hardware gate (HW gate) • 0b000 = deactivated (counter starts by setting SW gate) • 0b001 = activated (HIGH level at gate activates the HW gate. Counter starts if HW and SW gate are set.) Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	

Data set		Name	Description/value	Lenze
No.	Byte			
0x81	0	Final value	Upper limitation of the counting range	0x00
	1	Start value	Lower limitation of the counting range	0x00
	2	Hysteresis	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
	3	Pulse	The pulse duration indicates for how long the output is to be set if the parameterised comparison criterion is reached or exceeded. The pulse duration can be specified in steps of 2.048 ms between 0 and 522.24 ms. If the pulse duration is = 0, the output is set until the comparison condition is no longer met.	0x00

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bit 7 ... 5: 0 (fixed)
2	State of the inputs at the time of the alarm Bit 0: A/pulse Bit 1: B/direction Bit 2: Latch Bit 3: Hardware gate Bit 4: Reset Bit 7 ... 5: 0 (fixed)
3/4	Ticker value at the time of the alarm

Gate counter open/closed: Bit 0 is set if the HW gate is activated while the SW gate is active. Bit 1 is set if the HW gate is deactivated while the SW gate is active.

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the count value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

Diagnosealarm

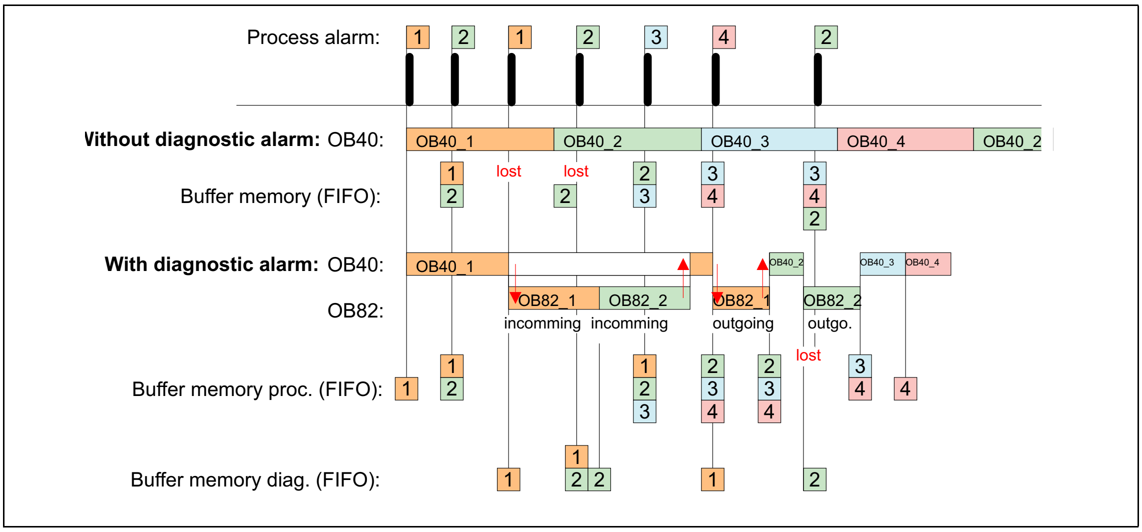
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{\text{incoming}}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{\text{incoming}}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{\text{outgoing}}$.

All events of a channel between the diagnostic alarm$_{\text{incoming}}$ and diagnostic alarm$_{\text{outgoing}}$ are not buffered and get lost. During this time (first diagnostic alarm$_{\text{incoming}}$ to the last diagnostic alarm$_{\text{outgoing}}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{\text{incoming/outgoing}}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x01)
7	Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed)
8	Diagnostic alarm due to process alarm lost to ... Bit 0: Hardware gate open Bit 1: Hardware gate closed Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 4: Latch value Bit 7 ... 5: 0 (fixed)
9 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

9.8.2 2 counters 32 bits, 24 V DC - EPM-S601

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.

Additional functions	Description
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Read data: 12 bytes

Input area in the process image		
Addr.	**Access**	**Assignment**
+0	Double word	Counter 1: count value
+4	Double word	Counter 2: count value
+8	Word	Counter 1: Status word
+10	Word	Counter 2: Status word

Count value: Current counter content

Write data: 12 bytes

Output area in the process image		
Addr.	**Access**	**Assignment**
+0	Double word	Counter 1: Comparison value
+4	Double word	Counter 2: Comparison value
+8	Word	Counter 1: Control word
+10	Word	Counter 2: Control word

Comparison value: With the comparison value you can specify a value which, by comparison with the current counter content, can impact the counter output or trigger a process alarm. The response of the comparison bit *STS_COMP* in the counter status or the process alarm is to be specified via data record 0x80 for counter 1 and 0x82 for counter 2.

▸ Counter modules EPM-S600 ... EPM-S603 – control and status words (📖 27)

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x01	0	Input frequency Counter 1, track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
	1	Input frequency Counter 1, track B		0x02
	2	Input frequency Counter 2, track A		0x02
	3	Input frequency Counter 2, track B		0x02

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	0	Alarm response counter 1	Setting activates process alarm Bit 1 ... 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 ... 6: Reserved	0x00
	1	Counter function counter 1	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 ... 6: Reserved	0x00
	2	Comparator counter 1	Bit 2 ... 0: Comparison bit is set (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
	3	Signal evaluation counter 1	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
0x81	0...3	Set value counter 1	When a set value is given, the counter can be loaded with the set value. With an edge 0-1 at COUNTERVAL_SET in the control word, the set value is accepted in the counter.	0x00
	4...7	Final value counter 1	Upper limitation of the counting range	0x00
	8...11	Loading value counter 1	Lower limitation of the counting range	0x00
	12	Hysteresis counter 1	The hysteresis for instance serves to avoid frequent switching operations of the output and/or triggering of the alarm when the count value is within the range of the comparison value. For the hysteresis you can select a range between 0 and 255. With the settings 0 and 1 the hysteresis is switched off. The hysteresis has an effect on the zero crossing, comparison, overflow/underflow.	0x00
	13	Reserved		

Data set		Name	Description/value	Lenze
No.	Byte			
0x82	0	Alarm response counter 2	See counter 1	0x00
	1	Counter function counter 2	See counter 1	0x00
	2	Comparator counter 2	See counter 1	0x00
	3	Signal evaluation counter 2	See counter 1	0x00
0x83	0...3	Set value counter 2	See counter 1	0x00
	4...7	Final value counter 2	See counter 1	0x00
	8...11	Loading value counter 2	See counter 1	0x00
	12	Hysteresis counter 2	See counter 1	0x00

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: 0 Bit 1: 0 Bit 2: Counter 1, overflow/underflow/final value Bit 3: Counter 1, comparison value reached Bit 4: 0 Bit 5: 0 Bit 6: Counter 2, overflow/underflow/final value Bit 7: Counter 2, comparison value reached
2	State of the inputs at the time of the alarm Bit 0: Counter 1, A/pulse Bit 1: Counter 1, B/direction Bit 2: Counter 2, A/pulse Bit 3: Counter 2, B/direction Bit 7 ... 4: 0 (fixed)
3/4	16 bit μs value at the time of the alarm

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the count value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

Diagnosealarm

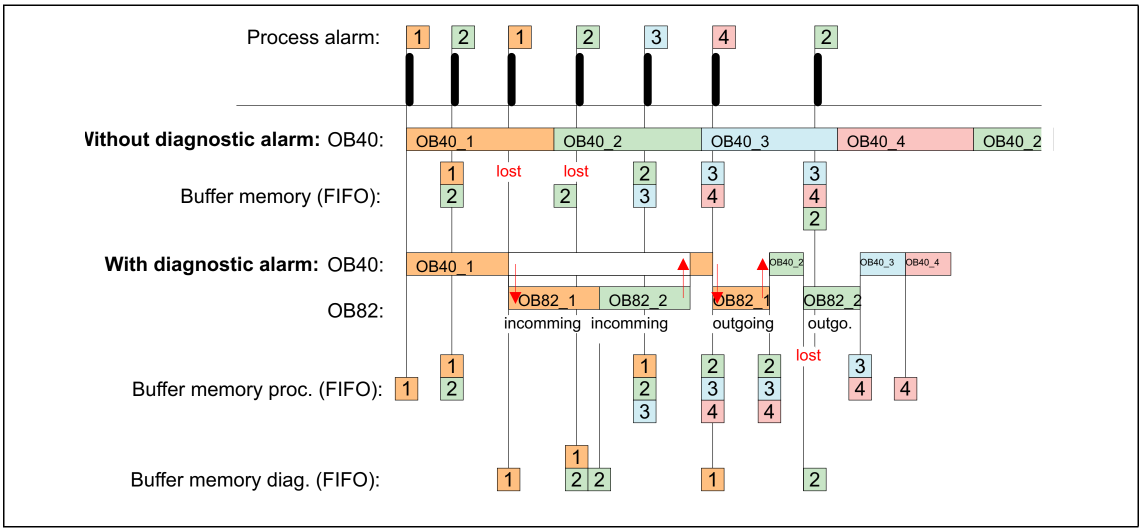
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x02)
7	Bit 0: Error in channel group 0 (counter 1) Bit 1: Error in channel group 1 (counter 2) Bit 7 ... 2: 0 (fixed)
8	Channel group 0: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fixed) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fixed)
9	Channel group 1: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fixed) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fixed)
10 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

9.8.3 1 counter 32 bits, 5 V DC - EPM-S602

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.
Counting once	The counter counts once/periodically from the loading value in the specified counting range.
Counting periodically	

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Main counting direction	The main counting direction can be parameterised: **None:** The entire counting range is available. **Forwards:** Limitation of the counting range upwards. The counter counts from the parameterised loading value in the positive direction to the parameterised final value -1 and then skips to the loading value again with the encoder pulse that is following. **Backwards:** Limitation of the counting range downwards. The counter counts from the parameterised loading value in the negative direction to the parameterised final value +1 and then skips to the loading value again with the encoder pulse that is following.
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). The following response can be parameterised: **Cancelling gate function:** After closing the gate and opening it again, the counting process continues from the loading value again. **Interrupting gate function:** After closing the gate and opening it again, the counting process continues with the last current counter content.
Comparator	You can specify a comparison value which, depending on the count value, activates the digital output or triggers a process alarm.
Hysteresis	By specification of a hysteresis you can for instance prevent the output from being permanently switched if the value of an encoder signal fluctuates around the comparison value.
Process alarm	The activation of a process alarm can be parameterised. A process alarm can be triggered in the case of the following events: • Open hardware gate • Closed hardware gate • Counter limit overflow • Counter limit underflow • Comparison value reached • Final value reached • Latch value reached
Diagnosealarm	If the diagnostic alarm is enabled in the parameter setting, it occurs if another process alarm is triggered for the same event during a process alarm processing.

Additional functions	Description
Diagnostic function	Diagnostic functions are provided by the modules.
Diagnostic information	The diagnostic information describes the diagnostic contents that can be read out of a module. Diagnostic information is not automatically sent by the module but must be read out actively via SDO access.

Read data: 8 bytes

Input area		
Addr.	**Access**	**Assignment**
+0	Double word	Counter value
+4	Word	Status word
+6	Word	Ticker value

Count value: Current counter content

Ticker value: After mains connection, a timer (µs ticker) is started which restarts at 0 after 65535 µs. With every change of the count value, the time value of the timer is stored as 16-bit-µs value together with the count value in the input area.

Write data: 10 bytes

Output area		
Addr.	**Access**	**Assignment**
+0	Double word	Comparison value
+4	Double word	Set value
+8	Word	Control word

Comparison value: Here you can specify a value which, by comparison with the current counter content, is able to influence the counter output or trigger a process alarm. The response of the output or of the process alarm can be parameterised.

Set value: With an edge change 0-1 of *COUNTERVAL_SET* in the control word, the set value is accepted in the counter.

▸ Counter modules EPM-S600 ... EPM-S603 – control and status words (📖 27)

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x01	0	Input frequency Track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
	1	Input frequency Track B		0x02
	2	Input frequency Reset		0x02
	3	Reserved		

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	0	Alarm response	Setting activates process alarm Bit 1 ... 0: Reserved Bit 2: Proc. alarm overflow Bit 3: Proc. alarm underflow Bit 4: Proc. alarm comparison value Bit 5: Proc. alarm final value Bit 7 ... 6: Reserved	0x00
	1	Numerator function	Bit 5 ... 0: 0b000000 = continuous counting 0b000001 = one-time: Forward 0b000010 = one-time: Backward 0b000100 = one-time: No main counting direction 0b001000 = periodical: Forward 0b010000 = periodical: backward 0b100000 = periodical: No main counting direction Bit 7 ... 6: Reserved	0x00
	2	Comparator	Bit 2 ... 0: Comparison bit is set (... if condition is met) 0b000 = never 0b001 = count value ÷ comparison value 0b010 = count value ≤ comparison value 0b100 = count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 6 ... 4: Reset 0b000 = deactivated 0b001 = HIGH level 0b011 = rising edge 0b101 = one-time rising edge Bit 7: Reserved	0x00
	3	Signal evaluation	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 6 ... 3: Reserved Bit 7: Gate function (internal gate) • 0= abort (counting process starts again from loading value) • 1 = interrupt (counting process is continued with counter content)	0x00
0x81	4...7	Final value	Upper limitation of the counting range	0x00
	8...11	Start value	Lower limitation of the counting range	0x00
	12	Hysteresis		0x00

Process alarm

A process alarm causes a call of the OB 40. Within the OB 40 you have the possibility of determining the logic basic address of the module that has triggered the process alarm via the local word 6. Further information on the triggering event can be found in the "Local double word 8".

Local double word 8 of OB 40:

Local byte	Bit 7 ... 0
1	Bit 0: 0 Bit 1: 0 Bit 2: Counter 1, overflow/underflow/final value Bit 3: Counter 1, comparison value reached Bit 4: 0 Bit 5: 0 Bit 6: Counter 2, overflow/underflow/final value Bit 7: Counter 2, comparison value reached
2	State of the inputs at the time of the alarm Bit 0: Counter 1, A/pulse Bit 1: Counter 1, B/direction Bit 2: Counter 2, A/pulse Bit 3: Counter 2, B/direction Bit 7 ... 4: 0 (fixed)
3/4	16 bit µs value at the time of the alarm

Ticker value: After mains connection, a timer (µs ticker) is started which restarts at 0 after 65535 µs. With every change of the count value, the time value of the timer is stored as 16-bit-µs value together with the count value in the input area.

Diagnosealarm

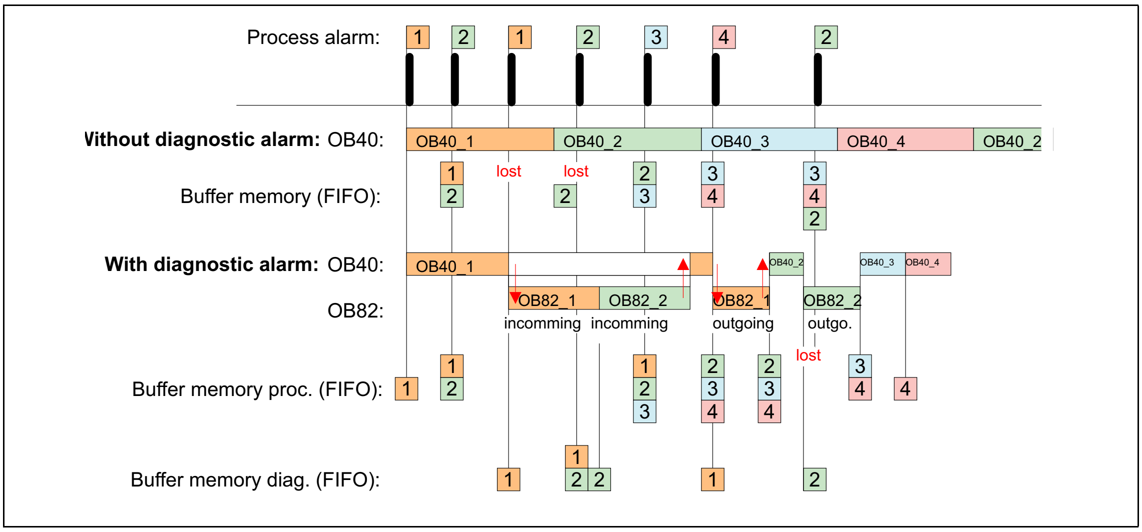
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set if module fault Bit 1: 0 (fixed) Bit 2: Set in the case of an external error Bit 3: Set in the case of a channel error Bit 6 ... 4: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x02)
7	Bit 0: Error in channel group 0 (counter 1) Bit 1: Error in channel group 1 (counter 2) Bit 7 ... 2: 0 (fixed)
8	Channel group 0: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fixed) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fixed)
9	Channel group 1: Diagnostic alarm due to lost process alarm to ... Bit 1 ... 0: 0 (fixed) Bit 2: Overflow/underflow/final value Bit 3: Comparison value reached Bit 7 ... 4: 0 (fixed)
10 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

9.8.4 2 counters 32 bits, 24 V DC - EPM-S603

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Counting functions	Description
Continuous counting	The counter counts from 0 to the counting limit, then skips to the opposite counting limit and continues to count from there.

Signal evaluation	Description
Single rotary transducer	Connection to "A/pulse" input
Double rotary transducer	
Quadruple rotary transducer	Connection to input "A/pulse" and "B/direction"
Direction	Pulse at "A/pulse" input and direction at "B/direction" input

Additional functions	Description
Gate function	The gate function serves to start, stop, and interrupt a counting function. In the case of this counter the internal gate (I-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area).

Read data: 12 bytes

Input area in the process image		
Addr.	**Access**	**Assignment**
+0	Double word	Counter 1: count value
+4	Double word	Counter 2: count value
+8	Word	Counter 1: Status word
+10	Word	Counter 2: Status word

Write data: 4 bytes

Output area in the process image		
Addr.	**Access**	**Assignment**
+0	Word	Counter 1: Control word
+2	Word	Counter 2: Control word

▸ Counter modules EPM-S600 ... EPM-S603 – control and status words (📖 27)

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x01	0	Input frequency Counter 1, track A	Filters for instance serve to filter signal peaks in the case of an unclean input signal. 0 (0x00): 500 kHz 1 (0x01): 300 kHz 2 (0x02): 100 kHz 3 (0x03): 60 kHz 4 (0x04): 30 kHz 6 (0x06): 10 kHz 7 (0x07): 5 kHz 8 (0x08): 2 kHz 9 (0x09): 1 kHz Other values are not permissible!	0x02
	1	Input frequency Counter 1, track B		0x02
	2	Input frequency Counter 2, track A		0x02
	3	Input frequency Counter 2, track B		0x02
0x80	0	Counting direction counter 1, track B	Bit 2 ... 0: Reserved Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
	1	Signal evaluation counter 1	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 7 ... 3: Reserved	0x00
0x82	0	Counting direction counter 2, track B	Bit 2 ... 0: Reserved Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bit 7 ... 4: Reserved	0x00
	1	Signal evaluation counter 2	Bit 2 ... 0: Signal evaluation • 0b000 = counter deactivated (the other parameter data details for the counter are ignored) • 0b001 = rotary transducer 1-fold (connection to "A/pulse" input) • 0b010 = rotary transducer 2-fold (connection to "A/pulse" input) • 0b011 = rotary transducer 4-fold (connection to "A/pulse" input and "B/direction") • 0b100 = direction (pulse at "A/pulse" input and direction at "B/direction" input) Bit 7 ... 3: Reserved	0x00

Diagnostic data

Using the SFB 52 you can access the diagnostic data of the module any time. Since this module does not support a process alarm, the diagnostic data serve to provide information on this module.

Data set 1 is structured as follows:

Byte	Bit 7 ... 0
0	0 (fixed)
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: More channel types available, 0: no, 1: yes
5	Number of diagnostic bits output by the module per channel (here 0x00))
6	Number of channels of a module (here 0x02)
7 ... 15	0 (fixed)

9.9 Parameterising the encoder evaluation

9.9.1 SSI - EPM-S604

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Functions	Description
SSI encoder parameters	According to encoder data sheet
Operating mode	Master mode or monitoring operation
Alarm response	With definition of the comparison and limit values

Read data: 6 bytes

Input area		
Addr.	**Access**	**Assignment**
+0	Double word	Encoder value
+4	Word	Ticker value

Encoder value: current encoder value

Ticker value: After mains connection, a timer (μs ticker) is started which restarts at 0 after 65535 μs. With every change of the encoder value, the time value of the timer is stored as 16-bit-μs value together with the count value in the input area.

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x80	0	Idle time	The dead time, also called tbs (time between sends), defines the waiting time between two encoder values to be observed by the module so that the encoder is able to process its value. This data can be found in the data sheet for your encoder. HIGH LOW 0x00 0x30: 1 µs 0x00 0x60: 2 µs 0x00 0xC0: 4 µs 0x01 0x80: 8 µs 0x03 0x00: 16 µs 0x06 0x00: 32 µs 0x09 0x00: 48 µs 0x0C 0x00: 64 µs	0x0C00
	1	Baud rate	In "monitoring operation" operating mode, the baud rate is irrelevant. Enter the baud rate here. This corresponds to the clock frequency via which the encoder connected communicates. Information on this can be found in the data sheet for your encoder. HIGH LOW 0x00 0x18: 2 MHz 0x00 0x20: 1.5 MHz 0x00 0x30: 1 MHz 0x00 0x60: 500 kHz 0x00 0xC0: 250 kHz 0x01 0x80: 125 kHz	0x0180
	2	Reserved		
	3	Standardisation	Depending on the encoder, further bits are transmitted in addition to the encoder value. Scaling serves to determine how many bits post-positioned to the encoder value will be removed by shifting the encoder value to the right. The encoder value is scaled by the module only after a Gray-binary conversion. More information can be found in the data sheet for your encoder. Value range: 0x00 ... 0x0F = 0 bit ... 15 bits	0x00

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	4	Bit length of encoder data	Enter the bit length of the encoder data here. Depending on the encoder, the encoder data consist of the current encoder value with subsequent bits. The total length has to be specified here. More information on this can be found in the data sheet for your encoder. 7 (0x07) = "8 bits" 8 (0x08) = "9 bits" ... 24 (0x18) = "25 bits" ... 31 (0x1F) = "32 bits"	0x18
	5		Bit 1 ... 0: Ready for operation During "Monitoring operation" the module serves to monitor the data exchange between an SSI master and an SSI encoder. It receives the cycle by the master and the data flow by the SSI encoder. In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder. 0b01 = monitoring operation 0b10 = master mode Bit 2: Shifting direction Specify the orientation of the encoder data here. More information can be found in the data sheet for your encoder. Usually, the SSI encoder uses "MSB first". 0 = LSB first (LSB is transmitted first) 1 = MSB first (MSB is transmitted first) Bit 3: Edge clock signal Here you can specify the edge type of the clock signal, in the case of which the encoder supplies data. Information on this can be found in the data sheet for your encoder. Usually the SSI encoders respond to rising edges. 0 = falling edge 1 = rising edge **Master mode:** Connect clock output signal (ClockOut) to the EPM-S604. **Monitoring mode:** Connect clock input signal (ClockIn) to the EPM-S604. Bit 4: Coding In the "Binary code" setting, the provided encoder value remains unchanged. In the "Gray code" setting, the Gray-coded value provided by the encoder is converted into a binary value. Only after this conversion, the received encoder value is scaled, if required. The Gray code is another form of representation of the binary code. It is based on the fact that two adjacent Gray numbers differ from each other in exactly one bit. If the Gray code is used, transmission errors can be easily detected, since adjacent characters must only differ from each other in one digit. Information on this can be found in the data sheet for your encoder. 0 = standard code 1 = Gray code Bit 7 ... 5: Reserved	0x1E

Data set		**Name**	**Description/value**	**Lenze**
No.	Byte			
0x80	6	Reserved		
	7	SSI function	By enabling the SSI function the module starts with the cycle output and the evaluation of the encoder data. In the "monitor operation" mode, the module starts with the encoder evaluation. 0 (0x00) = inhibited 1 (0x01) = enabled	0x00

Diagnosealarm

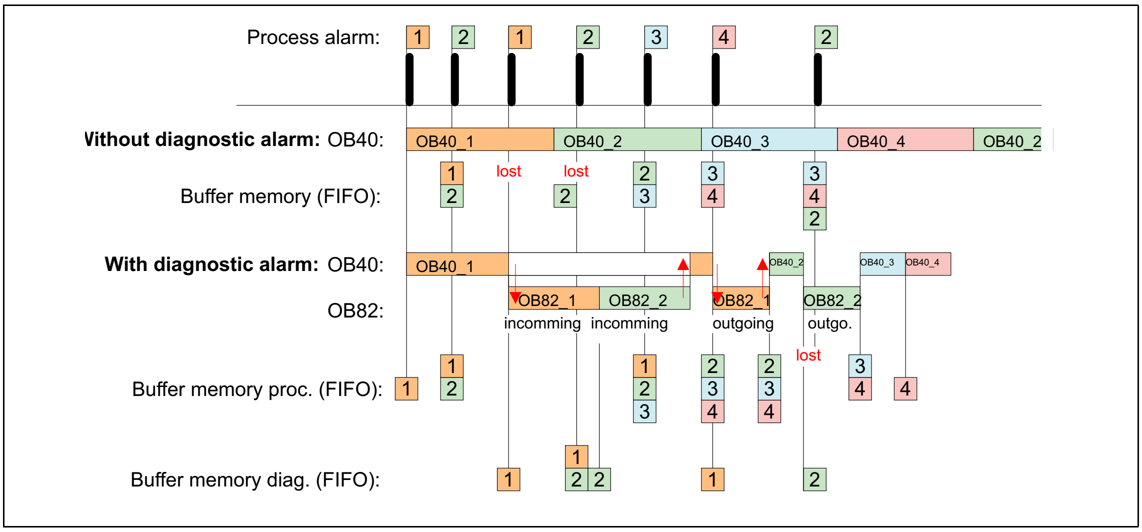
You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 0x00). A diagnostic alarm occurs if a further process alarm is triggered for the same event during a process alarm processing in the OB 40.

By activation of a diagnostic alarm the current process alarm processing in the OB 40 is interrupted and branched to the diagnostic alarm processing in OB 82. If further events occur on other channels during the diagnostic alarm processing, which can trigger a process or diagnostic alarm, they are buffered. At the end of the diagnostic alarm processing, first all buffered diagnostic alarms are processed in order of their occurrence, and afterwards all process alarms.

If further process alarms occur on a channel for which a diagnostic alarm$_{incoming}$ is currently processed or buffered, they are lost. If a process alarm for which a diagnostic alarm$_{incoming}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarm$_{outgoing}$.

All events of a channel between the diagnostic alarm$_{incoming}$ and diagnostic alarm$_{outgoing}$ are not buffered and get lost. During this time (first diagnostic alarm$_{incoming}$ to the last diagnostic alarm$_{outgoing}$), the MF-LED of the module is lit. Additionally, an entry in the diagnostic buffer of the CPU is made for every diagnostic alarm$_{incoming/outgoing}$.

Example:



Diagnostic alarm processing

Using the SFB 52 you can read out the diagnostic bytes. If the diagnostic alarm is deactivated, you can access the last diagnostic event in each case.

If you have activated the diagnostic function in your hardware configuration, OB 82 is called automatically. Here you can react on the diagnostics accordingly. With the SFB 52 you can additionally read out data set 1 which contains further information.

After exiting OB 82, the data can no longer be clearly assigned to the last diagnostic alarm.

Data set 1 is structured as follows:

Data set 1, diagnostic alarm$_{incoming}$	
Byte	**Bit 7 ... 0**
0	Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error Bit 3: Set if channel error available Bit 4: Set in the case of missing external supply voltage Bit 6 ... 5: 0 (fixed) Bit 7: Parameterisation error
1	Bit 3 ... 0: Module class, 0b1000: Function module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed)
4	Bit 6 ... 0: Channel type, 0x77: Counter module Bit 7: 0 (fix)
5	Number of diagnostic bits output by the module per channel (here 0x08))
6	Number of channels of a module (here 0x01)
7	Bit 0: Error in channel group 0
8 ... 15	0 (fixed)

Data set 1, diagnostic alarm$_{outgoing}$
After the error correction, a diagnostic alarm$_{outgoing}$ takes place.

9.10 Parameterising the time stamp

9.10.1 2 digital inputs with time stamp function - EPM-S207

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The following functions can be parameterised:

Functions	Description
Input delay	For example, signal peaks can be filtered in the event of an unclean input signal.
Edge selection	Specification of signal edge for input signal to produce a time stamp entry.

Read data: 6 bytes

Input area		
Addr.	**Access**	**Assignment**
+0	Byte	Status of inputs (PAE)
+1	Byte	Running number (RN)
+2	Word	Ticker value

Status of inputs: The status of the inputs after edge change is saved here. Parameters can be set for the following variants by integrating the gse file LE010C3A.gse:

20 bytes, 5 time stamp entries:

Addr.	+0	+1	+2	+3
+0	PAE	RN	16-bit µs value	
+4	PAE	RN-1	16-bit µs value	
+8	PAE	RN-2	16-bit µs value	
+12	PAE	RN-3	16-bit µs value	
+16	PAE	RN-4	16-bit µs value	

60 bytes, 15 time stamp entries:

Addr.	+0	+1	+2	+3
+0	PAE	RN	16-bit µs value	
+4	PAE	RN-1	16-bit µs value	
+8	PAE	RN-2	16-bit µs value	
+12	PAE	RN-3	16-bit µs value	
...	...	...	...	
+56	PAE	RN-14	16-bit µs value	

Running number (RN): The "running number" is a consecutive number between 0 ... 63, which always starts afresh from 0. You use the "running number" to determine the time sequence of entries. It must be incremented with every time stamp entry.

In the first run, the "running number" must start with 1.

Ticker value: After mains connection, a timer (µs ticker) is started which restarts at 0 after 65535 µs. With every change of the encoder value, the time value of the timer is stored as 16-bit-µs value together with the count value in the input area.

Parameter data

Data set No.	Data set Byte	Name	Description/value	Lenze
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C (fix)
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)
0x01	0	Input delay DI 1	0x00 = 1 µs 0x02 = 3 µs 0x04 = 10 µs 0x07 = 86 µs 0x09 = 342 µs 0x0C = 273 µs No other values are permissible.	0x02
	1	Input delay DI 2		0x02
0x80	0	Edge 0-1 at DI x	Time stamp entry on rising edge Bit 0: DI 1 (0: inhibit, 1 = enable) Bit 1: DI 2 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00
	1	Edge 1-0 on DI x	Time stamp entry on falling edge Bit 0: DI 1 (0: inhibit, 1 = enable) Bit 1: DI 2 (0: inhibit, 1 = enable) Bits 7 ... 2: reserved	0x00

Diagnostic data

Using SFB 52, you can read out the diagnostic bytes which provide information about the module. With SFB 52 you can also read out data set 1 which contains further information.

Data set 1 is structured as follows:

Data set 1, diagnostics	
Byte	**Bit 7 ... 0**
0	0 (fixed)
1	Bit 3 ... 0: Module class, 0b1111: Digital module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x70: Digital module Bit 7: more channel types available (0: yes; 1: no)
5	Number of diagnostic bits output by the module per channel (here 0x00))
6	Number of channels of a module (here 0x02)
7 ... 15	0 (fixed)

9.10.2 2 digital outputs with time stamp function - EPM-S310

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

The module has an FIFO (first-in-first-out) memory for 15 time stamp entries. Depending on parameter setting, you can use the output area to transfer up to 15 time stamp entries to the FIFO memory. The input process image provides information on the status of the FIFO memory and the status of processing.

Read data: 4 bytes

Input area		
Addr.	**Access**	**Assignment**
+0	Byte	Bit 5 ... 0: Running number (RN = Running Number) of the last FIFO entry Bit 6: 1 (fixed) Bit 7: 0 (fixed)
+1	Byte	Bits 5 ... 0: running number of the next FIFO entry Bit 6: 1 (fixed) Bit 7: 1 (fixed)
+2	Byte	Status
+3	Byte	Number of the time stamp entries in the FIFO memory

Running number (RN): Here you will find the "running number" of the time stamp entry last/next written to the FIFO.

Status: The status informs you of the status of the FIFO memory:

Code 0x00/0x80: Everything is OK

Code 0x01/0x81: No following time stamp entry available

Code 0x02/0x82: No new time stamp entries available.

Code 0x03/0x83: FIFO memory is full. No new time stamp entry can be accepted.

If bit 6 of the last processed "running number" was set, the code is returned at 0x80 OR-ed.

Note!

Note that no more time stamp entries can be accepted once the FIFO memory is full. You should always establish the status of the FIFO memory before the transfer to ensure that your entries are accepted.

Write data: 20 bytes/60 bytes

Depending on project planning, the output area can be used to write up to 15 time stamp entries. 4 bytes in the process image are intended for each time stamp entry:

Output area		
Addr.	**Access**	**Assignment**
+0	Byte	Bits 3 ... 0: 0 (fixed) Bit 4: Enable of DO 1 (0: Disable, 1: Enable) Bit 5: Enable of DO 0 (0: Disable, 1: Enable) Bit 6: DO 1 status Bit 7: DO 0 status
+1	Byte	Running number (RN)
+2	Word	Ticker value

Status of outputs:the status of the outputs for the time required is stated here. You can project plan the following variants by incorporating the GSE file LE010C3A.gse.gse:

20 bytes, 5 time stamp entries:

Addr.	+0	+1	+2	+3
+0	PAA	RN	16-bit µs value	
+4	PAA	RN-1	16-bit µs value	
+8	PAA	RN-2	16-bit µs value	
+12	PAA	RN-3	16-bit µs value	
+16	PAA	RN-4	16-bit µs value	

60 bytes, 15 time stamp entries:

Addr.	+0	+1	+2	+3
+0	PAA	RN	16-bit µs value	
+4	PAA	RN-1	16-bit µs value	
+8	PAA	RN-2	16-bit µs value	
+12	PAA	RN-3	16-bit µs value	
...	...	...	...	
+56	PAA	RN-14	16-bit µs value	

Running number (RN): The "running number" is a consecutive number between 0 ... 63, which always starts afresh from 0. You use the "running number" to determine the time sequence of entries. It must be incremented with every time stamp entry.

In the first run, the "running number" must start with 1.

Note!

If using SFC 15 to write consistent user data, up to 15 time stamp entries can be written. If less than 15 time stamp entries are written, bit 6 must also be set for the last RN. This has to be done to ensure that the following entries don't have to be written in an "invalid" way. The module ignores all time stamp entries after an entry with a set bit 6.

Ticker value: Specify a time here in µs at which the status of the outputs is to be accepted (value range: 0 ... 65535).

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x14 or 0x3C (fix)
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	0x00 (fix)

Diagnostic data

Using SFB 52, you can read out the diagnostic bytes which provide information about the module. With SFB 52 you can also read out data set 1 which contains further information.

Data set 1 is structured as follows:

Data set 1, diagnostics	
Byte	**Bit 7 ... 0**
0	0 (fixed)
1	Bit 3 ... 0: Module class, 0b1111: Digital module Bit 4: Channel information available Bit 7 ... 5: 0 (fix)
2	0 (fixed)
3	0 (fixed)
4	Bit 6 ... 0: Channel type, 0x70: Digital module Bit 7: more channel types available (0: yes; 1: no)
5	Number of diagnostic bits output by the module per channel (here 0x00))
6	Number of channels of a module (here 0x02)
7 ... 15	0 (fixed)

9.11 Parameterising technology modules

9.11.1 2 digital outputs with PWM functionality - EPM-S620

By integration of the GSE file VI010C19.gse you can specify all counter parameters via a hardware configuration.

Read data: 4 bytes

Input area			
Addr.	**Name**	**Byte**	**Function**
+0	PWMSTS_I	2	PWM 1: Status word
+2	PWMSTS_II	2	PWM 2: Status word

Input area		
Bit	**Name**	**Function**
0	-	Reserved
1	STS_PVM	Status PWM 0: PWM output stopped 1: PWM output active
2	STS_OUTBV	Output status 0: Push/Pull output 1: Highside output
3 ... 15	-	Reserved

Write data: 12 bytes

Output area			
Addr.	**Name**	**Byte**	**Function**
+0	PWMPD_I	4	PWM 1: pulse duration
+4	PWMSTS_II	4	PWM 2: pulse duration
+8	PWMCTRL_I	2	PWM 1: Control word
+10	PWMSTS_II	2	PWM 2: Control word

- PWMPD_I, PWMPD_II (pulse duration):
  Determine the scanning ratio for the parameterised period by stating the duration for the HIGH level for the corresponding PWM channel. The pulse duration should be chosen as factor for the 20.83 ns basis.
  Value range: 48 ... 8388607 (1μs ... approx. 175ms)
- PWMPD_I, PWMPD_II (control word):
  Here you can define the PWM output behaviour for the corresponding channel and start or stp the PWM output.

▸ PWM module EPM-S620 – control and status word (📖 32)

Parameter data

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	0	PWM 1: period	Set parameters here for the total time for pulse duration and pulse pause. The time should be selected as a factor for the 20.83 ns basis. Values below 25 µs are ignored. If the pulse duration is higher or equal to the period, the DO output is set permanently. Value range: 1200 ... 8388607 (25 µs ... approx. 175 ms)	0x1F40
0x81	0	PWM 2: Period		0x1F40

Diagnostic data

Since this module does not support alarms, the diagnostic data provides information about this module.

Diagnostic data record - data record 0x01			
Name	Byte	**Function**	**Default**
ERR_A	1	Reserved	0x00
MODTYP	1	Module information Byte 0: Bit 3 ... 0: Module class (0b1111: Digital module) Bit 4: channel information available Bits 7 ... 4: 0 reserved	0x15
ERR_C	1	Reserved	0x00
ERR_D	1	Reserved	0x00
CHTYP	1	Channel type Byte 0: Bit 6 ... 0: Channel type (0x72: Digital output) Bit 7: Reserved	0x72
NUMBIT	1	Number of diagnostic bits per channel Byte 0: Here 0x00	0x00
NUMCH	1	Number of channels in module Byte 0: Here 0x02	0x02
CHERR	1	Reserved	0x00
CH0ERR ... CH7ERR	6	Reserved	0x00
DIAG_US	4	Value of µs ticker when diagnostics occur Bytes 0 ...3	0

9.11.2 RS232 interface - EPM-S640

Parameter data - ASCII protocol

<table>
<tr><th colspan="2">Data set</th><th rowspan="2">Name</th><th rowspan="2">Description/value</th><th rowspan="2">Lenze</th></tr>
<tr><th>No.</th><th>Byte</th></tr>
<tr><td>0x00</td><td>0</td><td>Diagnostics</td><td>Bit 5 ... 0: Reserved
Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled)
Bit 7: Reserved
Other values are not permissible!</td><td>0x00</td></tr>
<tr><td rowspan="2">0x02</td><td>0</td><td>Length - process image input data</td><td>Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.</td><td></td></tr>
<tr><td>1</td><td>Length - process image output data</td><td>Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.</td><td></td></tr>
<tr><td>0x80</td><td>0</td><td>Baud rate</td><td>0x00: 9600 Baud
0x01: 150 Baud
0x02: 300 Baud
0x03: 600 Baud
0x04: 1200 Baud
0x05: 1800 Baud
0x06: 2400 Baud
0x07: 4800 Baud
0x08: 7200 Baud
0x09: 9600 Baud
0x0A: 14400 Baud
0x0B: 19200 Baud
0x0C: 38400 Baud
0x0D: 57600 Baud
0x0E: 76800 Baud
0x0F: 115200 Baud
0x10: 109700 Baud</td><td>0x00</td></tr>
<tr><td>0x80</td><td>1</td><td>Protocol</td><td>The protocol to be used. This setting influences the structure of the parameter data.
0x01: ASCII</td><td>0x01</td></tr>
<tr><td colspan="5">Option 1, drawing frame</td></tr>
<tr><td rowspan="4">0x80</td><td rowspan="4">2</td><td rowspan="4">Data format</td><td>Bit 1/0: Number of data bits
• 0b00: 5
• 0b01: 6
• 0b10: 7
• 0b11: 8</td><td>0b11</td></tr>
<tr><td>Bit 3/2: Parity
• 0b00: none
• 0b01: odd
• 0b10: even
• 0b11: even</td><td>0b00</td></tr>
<tr><td>Bit 5/4: Number of stop bits
• 0b01: 1
• 0b10: 1.5
• 0b11: 2</td><td>0b01</td></tr>
<tr><td>Bit 7/6: Flow control
• 0b00: None
• 0b01: Hardware
• 0b10: XON/XOFF</td><td>0b00</td></tr>
<tr><td colspan="5">Option 2 / 3, ZNA</td></tr>
</table>

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	3	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	4	ZNA (LOW byte)		0x00
Option 4 / 5, character delay time				
0x80	5	Character delay time (HIGH byte)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	6	Character delay time (LOW byte)		0xFA
Option 6, number of receive buffers				
0x80	7	Number of receive buffers	Defines the number of receive buffers. As long as only one receive buffer is used and is assigned, no more data can be received. If up to 250 receive buffers are connected, the received data can be redirected to a still free receive buffer. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7 ... 12, reserved				
0x80	8 ... 13	Reserved		0x00

Parameter data STX/ETX protocol

Data set No.	Data set Byte	Name	Description/value	Lenze
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA				
0x80	3	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	4	ZNA (LOW byte)		0x00

Data set No.	Byte	Name	Description/value	Lenze
Option 4 / 5, TMO				
0x80	5	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	6	TMO (LOW byte)		0xFA
Option 6, number of start identifiers				
0x80	7	Number of start identifiers	0x00: 1 start identifier (2. start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1				
0x80	8	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2				
0x80	9	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers				
0x80	10	Number of end identifiers	0x00: 1 end identifier (2. end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1				
0x80	11	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2				
0x80	12	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved				
0x80	13	Reserved		0x00

Parameter data 3964(R) protocol

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)				

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	3	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 3, character delay time (x 20 ms)				
0x80	4	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)				
0x80	5	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)				
0x80	6	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repititons				
0x80	7	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL				
0x80	8	DBL	If the block wait time is exceeded, you can select the maximum number of repititions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority				
0x80	9	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved				
0x80	10 ... 13	Reserved		0x00

Diagnostic data

Since this module does not support alarms, the diagnostic data provides information about this module.

In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

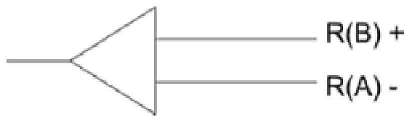
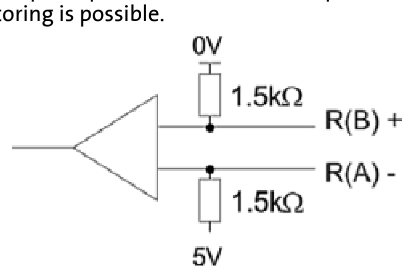
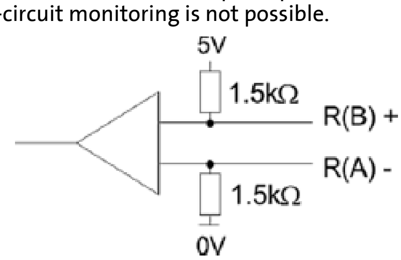
Diagnostic data - data record 0x01			
Name	**Byte**	**Function**	**Default**
ERR_A	1	ERR_A-diagnostics Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error (cable break) Bit 3: Reserved Bit 4: Set in the case of a missing external supply voltage Bit 5, 6: Reserved Bit 7: Set in the case of parameterisation error	0x00
MODTYP	1	Module information Byte 0: Bit 3 ... 0: Module class (0b0111: Gateway module) Bit 4: channel information available Bits 7 ... 4: 0 reserved	0x17
ERR_C	1	ERR_A-diagnostics Bit 7 ... 0: Reserved	0x00
ERR_D	1	ERR_D diagnostics Bit 3 ... 0: Reserved Bit 4: Set in the case of internal communication error Bit 7 ... 5: Reserved	0x00
CHTYP	1	Channel type Bit 7 ... 0: Reserved	0x00
NUMBIT	1	Number of diagnostic bits of the module per channel (here 0x08)	0x08
NUMCH	1	Number of channels in module Bit 7 ... 0: Reserved	0x00
CHERR	1	Bit 7 ... 0: Reserved	0x00
CH0ERR ... CH7ERR	8	Bit 7 ... 0: Reserved	0x00
DIAG_US	4	Value of µs ticker when diagnostics occur Bytes 0 ...3	0x00

9.11.3 RS422/RS485 interface - EPM-S650

Parameter data - ASCII protocol

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x01: ASCII	0x01
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA				

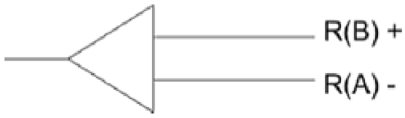
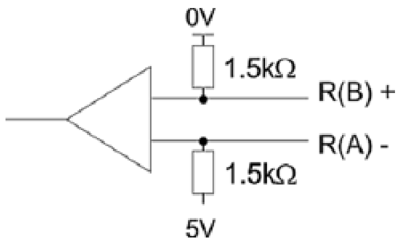
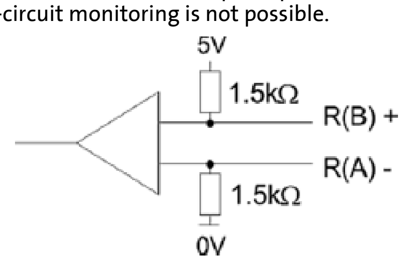
Data set		Name	Description/value	Lenze
No.	Byte			
0x80	3	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	4	ZNA (LOW byte)		0x00
Option 4 / 5, character delay time				
0x80	5	Character delay time (HIGH byte)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	6	Character delay time (LOW byte)		0xFA
Option 6, number of receive buffers				
0x80	7	Number of receive buffers	Defines the number of receive buffers. As long as only one receive buffer is used and is assigned, no more data can be received. If up to 250 receive buffers are connected, the received data can be redirected to a still free receive buffer. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7 ... 12, reserved				
0x80	8 ... 13	Reserved		0x00
Option 13, operating mode				
0x80	14	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

Data set		Name	Description/value	Lenze
No.	Byte			
Option 14, cable assignment				
0x80	15	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible.  	0x00

Parameter data STX/ETX protocol

Data set No.	Data set Byte	Name	Description/value	Lenze
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x02: STX/ETX	0x02
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2 / 3, ZNA				
0x80	3	ZNA (HIGH byte)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	4	ZNA (LOW byte)		0x00

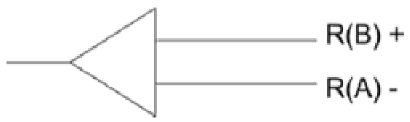
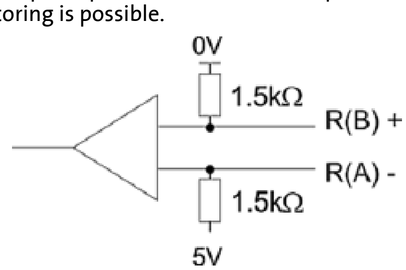
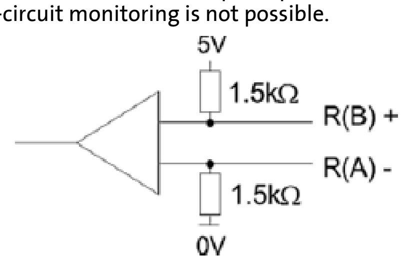
Data set No.	Byte	Name	Description/value	Lenze
Option 4 / 5, TMO				
0x80	5	TMO (HIGH byte)	TMO serves to define the maximally permissible interval between two frames. 0 ... 65535 [ms] (0x0000 ... 0xFFFF)	0x00
	6	TMO (LOW byte)		0xFA
Option 6, number of start identifiers				
0x80	7	Number of start identifiers	0x00: 1 start identifier (2. start identifier is ignored) 0x01: 2 start identifiers	0x01
Option 7, start identifier 1				
0x80	8	Start identifier 1	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 8, start identifier 2				
0x80	9	Start identifier 2	ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission.	0x00
Option 9, number of end identifiers				
0x80	10	Number of end identifiers	0x00: 1 end identifier (2. end identifier (0x310D/x) is ignored) 0x01: 2 end identifiers	0x00
Option 10, end identifier 1				
0x80	11	End identifier 1	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 11, end identifier 2				
0x80	12	End identifier 2	ASCII value of the end character that is sent after a frame and marks the end of a transmission. 0 ... 255 (0x00 ... 0xFF)	0x00
Option 12, reserved				
0x80	13	Reserved		0x00
Option 13, operating mode				
0x80	14	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01

Data set		Name	Description/value	Lenze
No.	Byte			
Option 14, cable assignment				
0x80	15	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible.  	0x00

Parameter data 3964(R) protocol

Data set		Name	Description/value	Lenze
No.	Byte			
0x00	0	Diagnostics	Bit 5 ... 0: Reserved Bit 6: Diagnostic alarm(0 = inhibited; 1 = enabled) Bit 7: Reserved Other values are not permissible!	0x00
0x02	0	Length - process image input data	Length of input data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
	1	Length - process image output data	Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible.	
0x80	0	Baud rate	0x00: 9600 Baud 0x01: 150 Baud 0x02: 300 Baud 0x03: 600 Baud 0x04: 1200 Baud 0x05: 1800 Baud 0x06: 2400 Baud 0x07: 4800 Baud 0x08: 7200 Baud 0x09: 9600 Baud 0x0A: 14400 Baud 0x0B: 19200 Baud 0x0C: 38400 Baud 0x0D: 57600 Baud 0x0E: 76800 Baud 0x0F: 115200 Baud 0x10: 109700 Baud	0x00
0x80	1	Protocol	The protocol to be used. This setting influences the structure of the parameter data. 0x03: 3964 0x04: 3964R	
Option 1, drawing frame				
0x80	2	Data format	Bit 1/0: Number of data bits • 0b00: 5 • 0b01: 6 • 0b10: 7 • 0b11: 8	0b11
			Bit 3/2: Parity • 0b00: none • 0b01: odd • 0b10: even • 0b11: even	0b00
			Bit 5/4: Number of stop bits • 0b01: 1 • 0b10: 1.5 • 0b11: 2	0b01
			Bit 7/6: Flow control • 0b00: None • 0b01: Hardware • 0b10: XON/XOFF	0b00
Option 2, ZNA (x 20 ms)				

Data set		Name	Description/value	Lenze
No.	Byte			
0x80	3	ZNA (x 20 ms)	Time after request (ZNA) Waiting time to be complied with until the next transmit request is executed. The ZNA is given as a factor of 20 ms steps. 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 3, character delay time (x 20 ms)				
0x80	4	Character delay time (x 20 ms)	Character delay time; the character delay time defines the max. permissible interval between two received characters within a frame. The character delay time is given as a factor of 20ms steps. If the character delay time is 0, the module independently calculates the character delay time on the basis of the baud rate (approx. the double character time). 0 ... 255 [ms] (0x00 ... 0xFA)	0x00
Option 4, acknowledgement time (x 20 ms)				
0x80	5	Acknowledgement time (x 20 ms)	The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. The acknowledgement time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 5, block wait time (x 20 ms)				
0x80	6	Block wait time (x 20 ms)	The block wait time is the maximum time period between confirming a request frame (DLE) and STX of the response frame. The block wait time is given as a factor of 20ms steps. 1 ... 255 (0x01 ... 0xFF)	0xFA
Option 6, STX repititons				
0x80	7	STX repetitions	Maximum number of times the module attempts to establish a connection. 1 ... 255 (0x01 ... 0xFF)	0x01
Option 7, DBL				
0x80	8	DBL	If the block wait time is exceeded, you can select the maximum number of repititions for the request frame via the DBL parameter. If these trials are not successful, the transfer is aborted. 1 ... 255 (0x01 ... 0xFF)	0x00
Option 8, priority				
0x80	9	Priority	A communication partner has high priority if the transmission attempt has priority over the partner's request to transmit. With a low priority, the partner's request to transmit comes first. In case of 3964(R), both partners must have different priorities. You have the following setting options: 0x00: Low 0x01: High	0x00
Option 9 ... 12, reserved				
0x80	10 ... 13	Reserved		0x00
Option 13, operating mode				

Data set No.	Data set Byte	Name	Description/value	Lenze
0x80	14	Operating mode	The operating mode determines if the interface is to be operated half duplex (RS485) or full duplex (RS422). • 0x00: Half duplex - two-wire operation (RS485) Half duplex operation means that either transmission or reception can take place at a time. The data are transmitted between the communication partners alternately in both directions. In case of the half duplex parametersetting under RS485, no software data flow control is possible. • 0x01: Full duplex - four-wire operation (RS422) The data are exchanged simultaneously between the communication partners; transmission and reception can take place at the same time. Each communication partner must simultaneously actuate a receive path.	0x01
Option 14, cable assignment				
0x80	15	Cable assignment	For a connection with minimum reflections and the open circuit detection in RS422/485 operation, the cables can be pre-assigned via parameters with a defined idle level. • 0x00: No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability.  • 0x01: Signal R(A) 5V (open-circuit monitoring), signal R(B) 0V In full duplex operation under RS422, open-circuit monitoring is possible.  • 0x02: Signal R(A) 0V, signal R(B) 5V This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open-circuit monitoring is not possible. 	0x00

Diagnostic data

Since this module does not support alarms, the diagnostic data provides information about this module.

In the event of an error the corresponding channel LED of the module is lit and the error is entered in the diagnostic data.

Diagnostic data - data record 0x01			
Name	**Byte**	**Function**	**Default**
ERR_A	1	ERR_A-diagnostics Bit 0: Set in the case of module fault Bit 1: Set in the case of internal error Bit 2: Set in the case of external error (cable break) Bit 3: Reserved Bit 4: Set in the case of a missing external supply voltage Bit 5, 6: Reserved Bit 7: Set in the case of parameterisation error	0x00
MODTYP	1	Module information Byte 0: Bit 3 ... 0: Module class (0b0111: Gateway module) Bit 4: channel information available Bits 7 ... 4: 0 reserved	0x17
ERR_C	1	ERR_A-diagnostics Bit 7 ... 0: Reserved	0x00
ERR_D	1	ERR_D diagnostics Bit 3 ... 0: Reserved Bit 4: Set in the case of internal communication error Bit 7 ... 5: Reserved	0x00
CHTYP	1	Channel type Bit 7 ... 0: Reserved	0x00
NUMBIT	1	Number of diagnostic bits of the module per channel (here 0x08)	0x08
NUMCH	1	Number of channels in module Bit 7 ... 0: Reserved	0x00
CHERR	1	Bit 0: set in the event of an error of channel group 1 Bits 7 ... 1 0 (fixed)	0x00
CH0ERR	8	Channel-specific error: channel x: Bits 3 ... 0: 0 (fixed) Bit 4 : set in the case of open circuit (only possible for RS422) Bits 7 ... 5: 0 (fixed)	0x00
CH1ERR ... CH7ERR	8	Bit 7 ... 0: Reserved	0x00
DIAG_US	4	Value of µs ticker when diagnostics occur Bytes 0 ...3	0x00

10 Modbus TCP communication

10.1 About Modbus TCP

Typical fieldbus systems are divided into master and slave systems. Master systems are CPs coupled to the CPU that which allow remote programming and visualisation of the corresponding CPU and a data exchange between several TCP/IP nodes.

Slave systems are "data collectors" which provide the requesting master with the I/O data of the connected modules.

The Modbus TCP bus coupler module described here is a slave system. Since, however, communication is executed out via TCP/IP, the slave system is called server and the master is called client.

The Modbus TCP bus coupler module serves to connect up to 64 I/O compound modules via Ethernet. Up to 8 clients can communicate simultaneously with the bus coupler.

Automatic address mapping

After switch-on, the bus coupler module identifies the I/O compound modules connected via the backplane bus and adds them to the address range. Address mapping provides one area for input data and one area for output data. The integrated web server provides access to the current mapping. Here, you can also parameterise your modules.

Communication

The Modbus TCP bus coupler module is connected to the I/O compound modules via the backplane bus. It collects their data and makes them available as "server" (slave) to a higher-level "client" (master system).

Communication is executed via TCP/IP Modbus TCP protocol where TCP/IP packets are transmitted. In reverse, the bus coupler module receives the data addressed via IP address and port and transmits them to its output peripherals.

Protocol

Protocols define regulations or standards for communication. A generally accepted model for the standardisation of the complete computer communication is the ISO/OSI layer model consisting of seven layers which manage the use of hardware and software.

Layer	Function	Protocol
7	Application layer	Modbus TCP
6	Presentation layer	
5	Session layer	
4	Transport layer	TCP
3	Network layer	IP
2	Data link layer (protection)	
1	Physical layer (bit transmission)	

Telegram structure

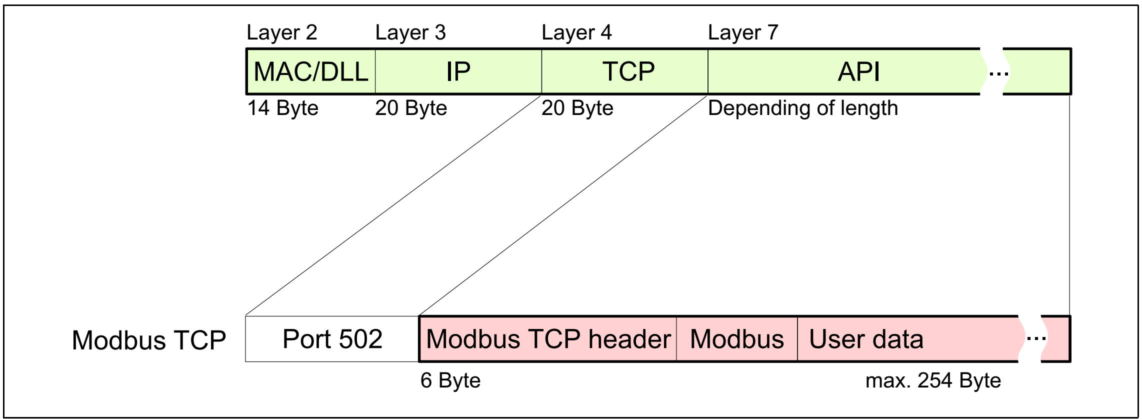
Layer 2	Layer 3	Layer 4	Layer 7	
MAC/DLL	**IP**	**TCP**	**API**	**...**
14 bytes	20 bytes	20 bytes	Length depends on protocol	

MAC/DLL: While the Ethernet physics with its standardised signal levels covers layer 1, MAC/DLL complies with the specifications for the data link layer (layer 2). With MAC (Medium Access Control) / DLL (Data Link Layer), communication takes place on the lowest Ethernet level using MAC addresses. Every Ethernet-capable station has a non-ambiguous MAC address that may only exist once. When MAC addresses are used, source and target are clearly specified.

IP: The internet protocol covers the network layer (layer 3) of the SO/OSI layer model. The task of the IP is to send data packets from one computer to the receiver via several computers. These data packets are datagrams. The IP neither ensures the correct order of datagrams nor the delivery at the receiver. For a clear distinction between sender and receiver, 32-bit addresses (IP addresses) are used that usually are written in four octets (exactly 8 bits), e.g. 172.16.192.11. In an octet, figures between 0 and 255 can be displayed. A part of the address specifies the network, the rest serves to identify the computer in the network. The transition between network part and host part is smooth and depends on the network size.

TCP: The TCP (Transmission Control Protocol) is directly based on the IP and thus covers the transport layer (layer 4) on the OSI layer model. TCP is a connection-oriented end-to-end protocol and serves as a logical connection between two partners. TCP ensures a logical and reliable data transfer. Each datagram is provided with a header of min. 20 bytes which, among other things, contains a sequence number for the correct order. Thus, the single datagrams in a network are able to reach the target in various ways.

API: API stands for Application Programming Interface. API fulfills the requirements for the application layer (layer 7). Here, header and user data of the corresponding protocols are stored. In the ModbusTCP bus coupler, the Modbus TCP protocol is used which will be explained in detail in the following section.



[10-1] API structure

Modbus TCP: Modbus TCP is a Modbus RTU protocol where TCP/IP packets are transmitted. The Modbus protocol is a communication protocol that supports a hierarchical structure with one master an several slaves.

Modbus TCP expands Modbus by a client/server communication where several clients can access a server. Since addressing is made via IP addresses, the address embedded in the Modbus frame is irrelevant. The CRC checksum is not required either since protection takes place via TCP/IP. After a request from a client, it waits for the response of the server until an adjustable waiting time has elapsed.

With Modbus TCP, the RTU format is used exclusively: Here, each byte is transmitted as a character. Thus, you have a higher data throughput as in the Modbus-ASCII format. RTU time monitoring does not occur since the header includes the size of the frame length to be received.

Data that is transferred with Modbus TCP may contain bit and word information. Here, in case of bit chains, the most significant bit is transmitted first, which means it has the left-most position in a word. In case of words, the most significant bit is transmitted first.

A Modbus slave is accessed via function codes

▶ Function codes (575)

10.2 Access to the I/O system 1000

In the following, access under Modbus TCP to the following areas of the I/O system are displayed:

- I/O area
- Parameter data
- Diagnostic data

Information on how to assign the ranges can be found in the descriptions of the corresponding I/O compound module.

Note!

Please note that the supply and terminal modules do not have any type identification. They cannot be identified by the bus coupler and are thus not considered in the listing or assignment of the slots.

In the following, slots within Modbus TCP will be referred to as Modbus TCP slots. Counting always starts at 0.

Addressing

In order that the plugged-in I/O compound modules can be addressed individually, specific addresses in the bus coupler must be assigned to them. The Modbus TCP bus coupler module provides an address range of 1024 bytes for input and output, respectively.

Address allocation (also called mapping) takes place automatically and cannot be influenced. The mapping can be output via the web page of the bus coupler.

During acceleration, the bus coupler automatically allocates addresses for its I/O compound modules according to the following rules:

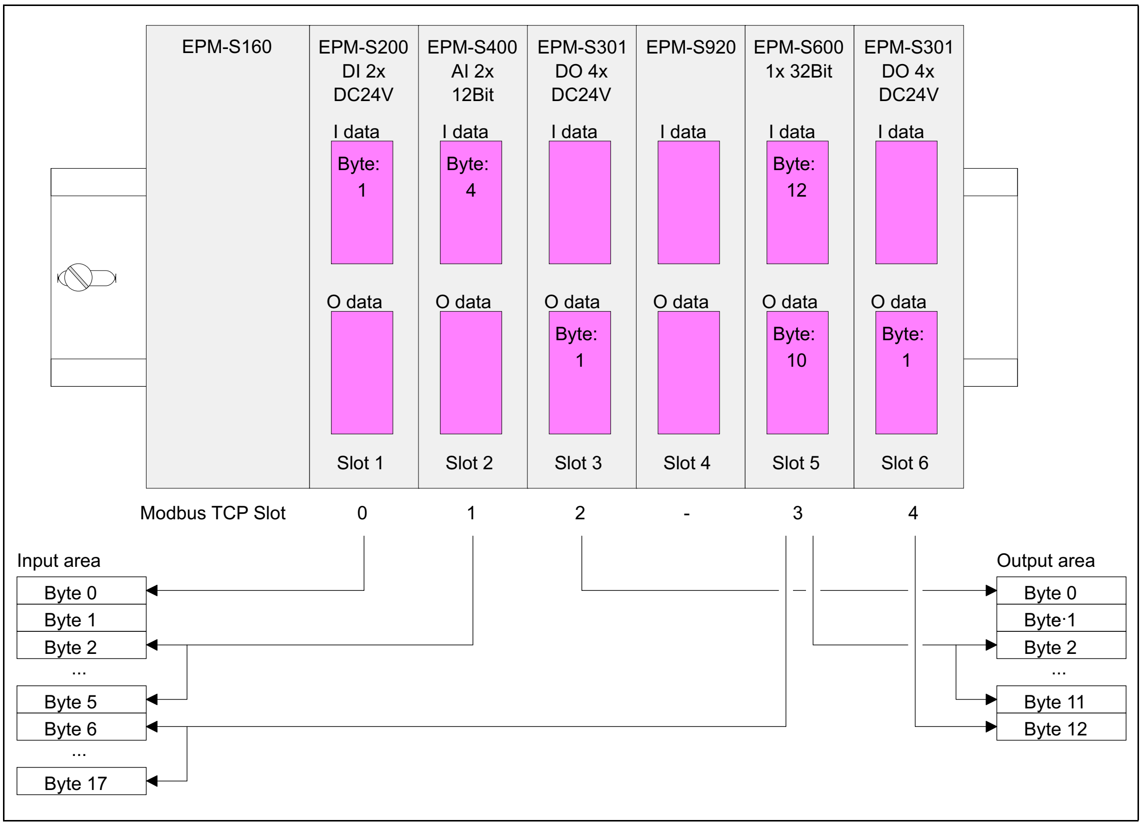
- As of address 0, all modules are mapped from the left (bus coupler) to the right in ascending order.
- It is distinguished between input and output range (if, for instance, a module has input and output data, they can be filed on different addresses).

There is no distinction made between digital and analog data. The Modbus TCP bus coupler generates one coherent area for input and output data, respectively, from all modules.

Note!

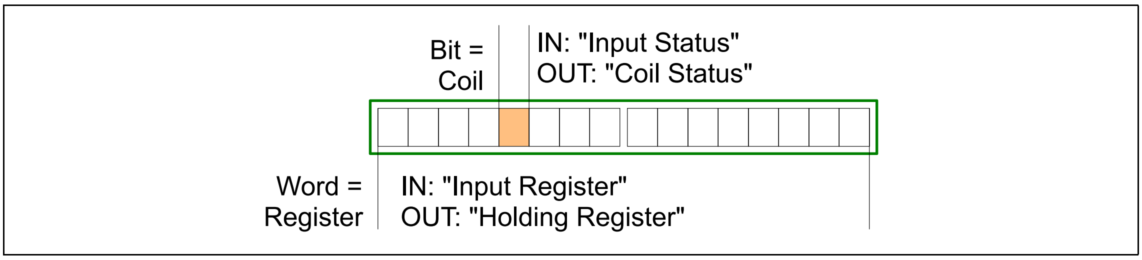
A description of the input and output ranges assigned to a module can be found in the "Product description" chapter in the respective module description.

Please observe that modules assigning more than 1 byte such as analog modules are stored from of a straight address onwards. Otherwise word access errors will be caused for Modbus TCP.



[10-2] Example for addressing

10.2.1 Access to the I/O area



Conventions:

- Modbus distinguishes between bit access and word access; bits = "coils" and words = "register".
- Bit inputs are called "input status" and bit outputs are called "coil status".
- Word inputs are called "input register" and word outputs are called "holding register"

Range definitions

Usually, the access under Modbus takes place via the ranges 0x, 1x, 3x and 4x.

0x And 1x serve to access digital ranges and 3x and 4x serve to access word ranges.

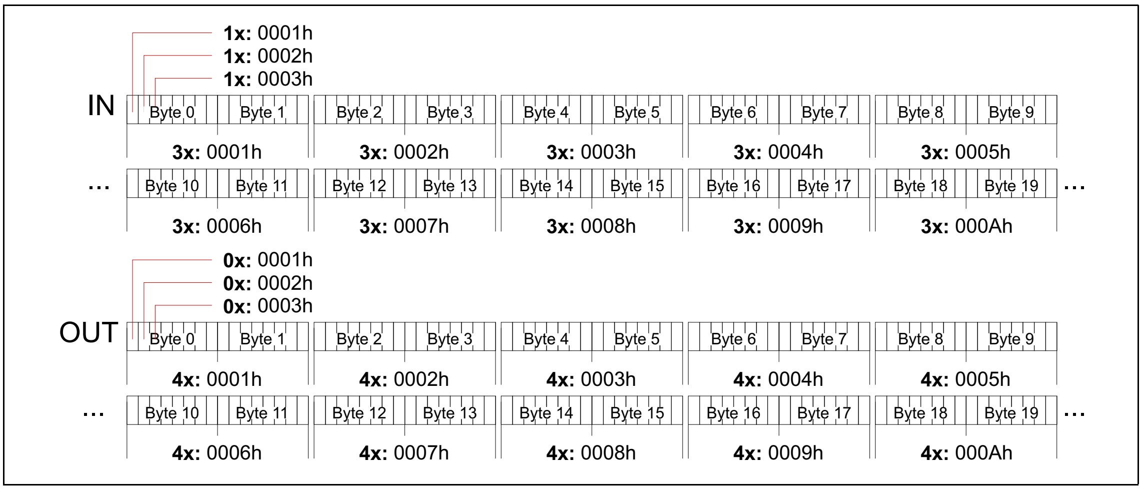
Since, however, no distinction is made between digital and analog data for the Lenze Modbus TCP bus coupler, the following assignment applies:

0x: Bit area for master output; access via function code 0x01, 0x05, 0x0F

1x: Bit area for master input; access via function code 0x02

3x: Word range for master input; access via function code 0x04, 0x17

4x: Word range for master output; access via function code 0x03, 0x06, 0x10, 0x16, 0x17



10.2.2 Function codes

The following function codes can be used to access a slave by a Modbus master. The description is always made from the master's view:

Code	Command	Description
0x01	Read n bits	Read n bits from the master output range 0x
0x02	Read n bits	Read n bits from the master input range 1x
0x03	Read n words	Read n words from the master output range 4x
0x04	Read n words	Read n words from the master input range 3x
0x05	Write 1 bit	Write 1 bit in the master output range 0x
0x06	Write 1 word	Write 1 word in the master output range 4x
0x0F	Write n bits	Write n bits in the master output range 0x
0x10	Write n words	Write n words in the master output range 4x
0x16	Mask 1 word	Mask 1 word in the master output range 4x
0x17	Write n words and read m words	Write n words in the master output range 4x and the response contains m read words of the master input range 3x

The following always applies to the byte order in the word: HIGH byte | LOW byte

Response of the bus coupler

If the slave returns an error, the function code is returned in an OR-ed manner with 0x80. If no error has occurred, the function code is returned.

Coupler response:

Function code OR 0x80 → error & error number

Function code → OK

In the event of an error you will receive additionally an error number in another byte. Here, the following error numbers exist:

0x01: Function number is not supported
0x02: Faulty addressing
0x03: Faulty data
0x04: System SLIO bus is not initialised
0x07: General error

Read n bits

Code 0x01: Read n bits of master output range 0x.
Code 0x02: Read n bits of master input range 1x.

Command frame:

Modbus TCP Header						Slave Address	Function code	Address 1. bit	Number Bits	
x	x	0	0	0	6					
6 byte						1 byte	1 byte	1 word	1 word	

Response message:

Modbus TCP Header						Slave Address	Function code	Number of read bytes	Data 1st byte	Data 2nd byte	...
x	x	0	0	0							
6 byte						1 byte	1 byte	1 byte	1 byte	1 byte	
									max. 252 bytes		

Read n words

0x03: Read n words from the master output range 4x.

0x04: Read n words from the master input range 3x.

Command frame:

Modbus TCP Header x x 0 0 0 6	Slave Address	Function code	Address Word	Number Words	
6 byte	1 byte	1 byte	1 word	1 word	

Response message:

Modbus TCP Header x x 0 0 0	Slave Address	Function code	Number of read bytes	Data 1st word	Data 2. word	...
6 byte	1 byte	1 byte	1 word	1 word		
				max. 126 words		

Write 1 bit

Code 0x05: Write 1 bit in the master output range 0x.

A status is changed under "Status bit" with the following values:

"Status bit" = 0x0000 → bit = 0

"Status bit" = 0xFF00 → bit = 1

Command frame:

Modbus TCP Header x x 0 0 0 6	Slave Address	Function code	Address Bit	Status Bit	
6 byte	1 byte	1 byte	1 word	1 word	

Response message:

Modbus TCP Header x x 0 0 0 6	Slave Address	Function code	Address Bit	Status Bit	
6 byte	1 byte	1 byte	1 word	1 word	

Write 1 word

Code 0x06: Write 1 word in the master output range 4x.

Command frame:

Modbus TCP Header x x 0 0 0 6	Slave Address	Function code	Address Word	Value Word	
6 byte	1 byte	1 byte	1 word	1 word	

Response message:

Modbus TCP Header x x 0 0 0 6	Slave Address	Function code	Address Word	Value Word	
6 byte	1 byte	1 byte	1 word	1 word	

Write n bits

Code 0x0F: Write n bits in the master output range 0x

Please observe that the number of bits must be additionally given in bits.

Command frame:

Modbus TCP Header x x 0 0 0	Slave Address	Function code	Address 1. bit	Number Bits	Number Bytes	Data 1st byte	Data 2nd byte	...
6 byte	1 byte	1 byte	1 word	1 word	1 byte	1 byte	1 byte	1 byte
						max. 248 bytes		

Response message:

Modbus TCP Header x x 0 0 0 6	Slave Address	Function code	Address 1. bit	Number Bits	
6 byte	1 byte	1 byte	1 word	1 word	

Write n words

Code 0x10: Write n words in the master output range.

Command frame:

Modbus TCP Header x x 0 0 0	Slave Address	Function code	Address 1st word	Number Words	Number Bytes	Data 1st word	Data 2. word	...
6 byte	1 byte	1 byte	1 word	1 word	1 byte	1 word	1 word	1 word
						max. 124 words		

Response message:

Modbus TCP Header x x 0 0 0 6	Slave Address	Function code	Address 1st word	Number Words	
6 byte	1 byte	1 byte	1 word	1 word	

Mask 1 word

Code 0x16: This function serves to mask a word in the master output range 4x.

Command frame:

Modbus TCP Header x x 0 0 0 8	Slave Address	Function code	Address Word	AND Mask	OR Mask	
6 byte	1 byte	1 byte	1 word	1 word	1 word	

Response message:

Modbus TCP Header x x 0 0 0 8	Slave Address	Function code	Address Word	AND Mask	OR Mask	
6 byte	1 byte	1 byte	1 word	1 word	1 word	

Write n words and read n words

Code 0x17: This function serves to write n words in the master output range 4x via a request and read n words of the master input range 3x.

Command frame:

Modbus TCP Header x x 0 0 0	Slave Address	Function code	Read address	Read Number Words	Write address	Write Number of bytes	Write Data 1st byte	Write data 2nd byte	...
6 byte	1 byte	1 byte	1 word	1 word	1 word	1 byte	1 word	1 word	
							max. 122 words		

Response message:

Modbus TCP Header x x 0 0 0	Slave Address	Function code	Read Number of bytes	Read Data 1. word	Read Data 2. word	...	
6 byte	1 byte	1 byte	1 word	1 word	1 word		
				max. 126 words			

10.2.3 Access to parameter data

At initial start, parameterisable modules are operated with their default parameters. If you want to set parameters, you can parameterise the Modbus TCP bus coupler or the plugged I/O compound modules via a web page.

10.2.4 Access to diagnostic data

In the event of an error, I/O compound modules can provide alarm data. As soon as one or several I/O compound modules report an alarm, the alarm data of the respective slot are received and acknowledged by the Modbus TCP bus coupler which then sets a bit assigned to the Modbus TCP slot in its internal "Alarm Information Image" and saves the corresponding alarm data.

In the I/O system 1000, it is made a distinction between diagnostic alarm and process alarm. For this purpose, the diagnostic image has one 64 bit wide field each (bit 0 = Modbus TCP slot 0 to bit 63 = Modbus TCP slot 63) for process alarm and diagnostic alarm. They are followed by a 16 byte slot for process alarm data and 32 byte slot for diagnostic alarm data.

For acknowledgement, you also have writing access to diagnostic and process alarm status. The alarm data can only be accessed by reading.

Register assignment

Address	Access to
0x/1x: 0x4000 ... 403F	Bit access to process alarm status: 0x/1x: 0x4000: Process alarm status Modbus TCP-Slot 0 ... 0x/1x: 0x403F: Process alarm status Modbus TCP-Slot 63
0x/1x: 0x5000 ... 0x503F	Bit access to diagnostic alarm status: 0x/1x: 0x5000: Diagnostic alarm status Modbus TCP slot 0 ... 0x/1x: 0x503F: Diagnostic alarm status Modbus TCP slot 63
3x: 0x4000 ... 0x41FF	Word access to process alarm data 3x: 0x4000 ... 0x4007: Modbus TCP slot 0 ... 3x: 0x41F8 ... 0x41FF: Modbus TCP slot 63
3x: 0x4000 ... 0x4007	16 bytes of process alarm data of Modbus TCP slot 0
...	...
3x: 0x41F8 ... 0x41FF	16 bytes of process alarm data of Modbus TCP slot 63
3x: 0x5000 ... 0x500F	32 bytes of diagnostic alarm data of Modbus TCP slot 0
...	...
3x: 0x53F0 ... 0x53FF	32 bytes of diagnostic alarm data of Modbus TCP slot 63

10.3 Possible access operations to the EPM-S160 Modbus TCP bus coupler module

10.3.1 Web page

The integrated HTTP web server is accessed via port 80, IP address 10.0.0.1 (default setting). The web page is built dynamically and is based on the number of modules connected to the Modbus TCP coupler.

Note!

Please note that the supply and terminal modules do not have any type identification. They cannot be identified by the bus coupler and are thus not considered in the listing or assignment of the slots.

In the following, slots within Modbus TCP will be referred to as Modbus TCP slots. Counting always starts at 0.

Structure of the web page

Data of the Modbus TCP bus coupler (serial number and firmware version)

Modbus TCP slot	Module designation	Link
0	Module on 1. slot	Information Data Parameter
1	Module on 2. slot	Information Data Parameter
...		
n	Last module	Information Data Parameter
		Communication settings Security settings Software update

Information: Here, the product name, order no., serial no., software version and hardware version number are listed.

Data: Here, you obtain information on the input/output status. Moreover, you can directly control the outputs of the corresponding module.

Parameter: If available, you can output and change the parameters of the corresponding parameters.

Communication Settings: Here you can define a timeout value in ms. If the waiting time of the Ethernet bus coupler exceeds the set timeout value, the Ethernet coupler stops communication and deactivates all modules. With a timeout value < 500 ms the timeout function is deactivated.

Security Settings: All functions for the writing access to the Modbus TCP bus coupler can be protected with a password query.

Software Update: This link is intended for future firmware updates.

Index

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.

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

feedback-docu@lenze.com

Thank you very much for your support.

Your Lenze documentation team

Lenze Automation GmbH
Postfach 10 13 52, 31763 Hameln
Hans-Lenze-Straße 1, 31855 Aerzen
GERMANY
HR Hannover B 205381
+49 5154 82-0
+49 5154 82-2800
lenze@lenze.com
www.lenze.com

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

Lenze