## L-force Controls

## System Manual

$\square$


## I/O System 1000



EPM-Sxxx
Modular I/O system
1 About this documentation ..... 12
1.1 Document history ..... 14
1.2 Conventions used ..... 15
1.3 Notes used ..... 15
1.4 Terminology used ..... 16
2 Safety instructions ..... 18
2.1 General safety information ..... 18
3 Product description ..... 21
3.1 Device features ..... 21
3.2 Application as directed ..... 22
3.3 System overview ..... 23
3.3.1 System design ..... 23
3.4 Module design ..... 25
3.4.1 Bus coupler modules ..... 25
3.4.2 I/O compound modules ..... 25
3.4.3 Supply modules ..... 26
3.4.4 Power distributor module ..... 26
3.4.5 Accessories ..... 27
3.4.6 Function elements ..... 28
3.5 Bus coupler modules ..... 29
3.5.1 CANopen - EPM-S110 ..... 29
3.5.2 PROFIBUS - EPM-S120 ..... 37
3.5.3 EtherCAT - EPM-S130 ..... 44
3.5.4 PROFINET - EPM-S140 ..... 49
3.5.5 DeviceNet - EPM-S150 ..... 54
3.5.6 Modbus TCP- EPM-S160 ..... 63
3.6 I/O compound modules - digital I/O ..... 67
3.6.1 Two digital inputs - EPM-S200 and EPM-S204 (NPN) ..... 67
3.6.2 Four digital inputs - EPM-S201 and EPM-S205 (NPN) ..... 70
3.6.3 Eight digital inputs - EPM-S202 and EPM-S206 (NPN) ..... 73
3.6.4 Four digital inputs (three-wire conductor connection) - EPM-S203 ..... 76
3.6.5 2 digital inputs with time stamp function - EPM-S207 ..... 79
3.6.6 Two digital outputs 0.5 A - EPM-S300 and EPM-S303 (NPN) ..... 83
3.6.7 Four digital outputs 0.5 A - EPM-S301 and EPM-S304 (NPN) ..... 87
3.6.8 Eight digital outputs 0.5 A - EPM-S302 and EPM-S305 (NPN) ..... 91
3.6.9 Two digital outputs 2 A - EPM-S306 ..... 95
3.6.10 Four digital outputs 2 A - EPM-S309 ..... 98
3.6.11 2 digital outputs with time stamp function - EPM-S310 ..... 101
3.6.12 Two relay outputs - EPM-S308 ..... 106
3.7 I/O compound modules - analog I/O ..... 110
3.7.1 Representation of analog values ..... 110
3.7.2 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400 ..... 111
3.7.3 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401 ..... 114
3.7.4 2 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S402 ..... 117
3.7.5 4 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ ( 12 bits) - EPM-S403 ..... 121
3.7.6 2 analog inputs -10 ... +10 V ( 16 bits) - EPM-S406 ..... 125
3.7.7 2 analog inputs $0 / 4$... 20 mA ( 16 bits) - EPM-S408 ..... 128
3.7.8 2 analog outputs 0 ... 10 V ( 12 bits) - EPM-S500 ..... 131
3.7.9 4 analog outputs 0 ... 10 V ( 12 bits) - EPM-S501 ..... 135
3.7.10 2 analog outputs $0 / 4$... 20 mA (12 bits) - EPM-S502 ..... 139
3.7.11 4 analog outputs $0 / 4$... 20 mA (12 bits) - EPM-S503 ..... 143
3.8 I/O compound modules - temperature measurement ..... 147
3.8.1 Representation of analog values ..... 147
3.8.2 Four (two) analog inputs for resistance tests - EPM-S404 ..... 148
3.8.3 Two analog inputs for thermocouple measurement - EPM-S405 ..... 152
3.9 I/O compound modules - counter ..... 156
3.9.1 One counter 32 bits, 24 V DC - EPM-S600 ..... 156
3.9.2 Two counters 32 bits, 24 V DC - EPM-S601 ..... 173
3.9.3 One counter 32 bits, 5 V DC - EPM-S602 ..... 189
3.9.4 Two counters 32 bits, 24 V DC - EPM-S603 ..... 204
3.10 I/O compound modules - encoder evaluation ..... 211
3.10.1 SSI-EPM-S604 ..... 211
3.11 I/O compound module - pulse width modulation ..... 215
3.11.1 2 digital outputs with PWM functionality - EPM-S620 ..... 215
3.12 I/O compound modules - communication ..... 219
3.12.1 RS232 interface - EPM-S640 ..... 219
3.12.2 RS485 interface - EPM-S650 ..... 224
3.13 Power supply modules ..... 230
3.13.1 I/O supply - EPM-S701 ..... 230
3.13.2 I/O supply and electronic supply - EPM-S702 ..... 232
3.14 Power distributor modules ..... 235
3.14.1 8 terminals 24 V - EPM-S910 ..... 235
3.14.2 8 terminals 0 V-EPM-S911 ..... 236
3.14.3 4/4 terminals $24 \mathrm{~V} / 0 \mathrm{~V}$ - EPM-S912 ..... 237
4 Technical data238
5 Mechanical installation ..... 240
5.1 Important notes ..... 240
5.2 Dimensions ..... 241
5.3 Mounting ..... 242
5.3.1 Standard mounting ..... 242
5.3.2 Block mounting ..... 243
5.3.3 Mounting the busbar for the shield connection ..... 244
5.3.4 Dismounting the bus coupler module, the I/O compound module, and the power supply module ..... 244
6 Electrical installation ..... 245
6.1 Wiring according to EMC ..... 245
6.2 Supply voltage connection ..... 246
6.3 Wiring of the control connections ..... 247
6.4 Connecting the shield ..... 248
7 Troubleshooting and fault elimination ..... 249
7.1 Troubleshooting via RUN- and MF-LED ..... 249
8 CANopen communication ..... 250
8.1 About CANopen ..... 250
8.1.1 Structure of the CAN data telegram ..... 250
8.1.2 Identifier ..... 250
8.1.3 Saving settings ..... 251
8.2 Network management (NMT) ..... 252
8.3 Transmitting process data ..... 254
8.3.1 Process data telegram ..... 254
8.3.2 Identifier of the process data objects (PDO) ..... 255
8.3.3 Assigning individual parameters ..... 256
8.3.4 Process data transmission mode ..... 256
8.3.5 PDO mapping ..... 258
8.3.6 PDO mapping for I/O compound modules with a serial interface ..... 260
8.3.7 PDO mapping for I/O compound modules with time stamp function ..... 261
8.3.8 PDO mapping for I/O compound modules with PWM function ..... 264
8.3.9 PDO mapping for counters and encoder evaluation ..... 266
8.3.10 Data transfer between I/O system and controller ..... 272
8.3.11 Indices for setting the process data transfer ..... 273
8.4 Transmitting parameter data ..... 276
8.4.1 Telegram structure ..... 276
8.4.2 Writing a parameter (example) ..... 279
8.4.3 Reading a parameter (example) ..... 280
8.5 Response of the station after switch-on ..... 282
8.6 Baud rate and node address (node ID) setting ..... 283
8.7 General function of the parameter setting ..... 284
8.7.1 Parameterising digital I/Os ..... 284
8.7.2 Parameterising analog I/Os, counter, SSI, time stamp and PWM ..... 285
8.8 Setting the parameters of analog $1 / \mathrm{O}$ ..... 286
8.8.1 2 analog inputs $0 \ldots 10 \mathrm{~V}$ ( 12 bits) - EPM-S400 ..... 286
8.8.2 4 analog inputs $0 . . .10 \mathrm{~V}$ ( 12 bits) - EPM-S401 ..... 287
8.8.3 2 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ ( 12 bits) - EPM-S402 ..... 288
8.8.4 4 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ ( 12 bits) - EPM-S403 ..... 289
8.8.5 2 analog inputs $-10 \ldots+10 \mathrm{~V}$ ( 16 bits) - EPM-S406 ..... 290
8.8.6 2 analog inputs $0 / 4$... 20 mA ( 16 bits) - EPM-S408 ..... 292
8.8.7 2 analog outputs 0 ... 10 V (12 bits) - EPM-S500 ..... 294
8.8.8 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501 ..... 295
8.8.9 2 analog outputs $0 / 4$... 20 mA ( 12 bits) - EPM-S502 ..... 296
8.8.10 4 analog outputs $0 / 4$... 20 mA (12 bits) - EPM-S503 ..... 297
8.8.11 Error behaviour ..... 298
8.9 Parameterising the temperature measurement ..... 299
8.9.1 Four (two) analog inputs for resistance tests - EPM-S404 ..... 299
8.9.2 Two analog inputs for thermocouple measurement - EPM-S405 ..... 306
8.9.3 Error behaviour ..... 309
8.10 Parameterising the counter ..... 310
8.10.1 Commissioning examples ..... 310
8.10.2 One counter 32 bits, 24 V DC - EPM-S600 ..... 311
8.10.3 Two counters 32 bits, 24 V DC - EPM-S601 ..... 315
8.10.4 One counter 32 bits, 5 V DC - EPM-S602 ..... 318
8.10.5 Two counters 32 bits, 24 V DC - EPM-S603 ..... 321
8.11 Parameterising the encoder evaluation ..... 323
8.11.1 SSI-EPM-S604 ..... 323
8.12 Time stamp parameterising ..... 326
8.12.1 2 digital inputs with time stamp function - EPM-S207 ..... 326
8.12.2 2 digital outputs with time stamp function - EPM-S310 ..... 326
8.13 Parameterising technology modules ..... 327
8.13.1 2 digital outputs with PWM functionality - EPM-S620 ..... 327
8.13.2 RS232 interface - EPM-S640 ..... 328
8.13.3 RS485 interface - EPM-S650 ..... 334
8.14 "Store" function ..... 340
8.15 Loading the default setting ..... 341
8.16 Node Guarding ..... 342
8.17 Heartbeat ..... 344
8.18 Monitoring ..... 346
8.18.1 Time monitoring for PDO1-Rx ... PDO10-Rx ..... 346
8.18.2 Digital output monitoring ..... 347
8.18.3 Monitoring of the analog outputs ..... 348
8.19 Diagnostics ..... 349
8.19.1 Emergency telegram ..... 349
8.19.2 Reading out the module identifiers ..... 355
8.19.3 Status of the digital inputs ..... 356
8.19.4 Status of the digital outputs ..... 356
8.19.5 Status of the analog inputs ..... 356
8.19.6 Status of the analog outputs ..... 356
8.19.7 Status of the counters ..... 357
8.19.8 Status of digital inputs with time stamp function ..... 357
8.19.9 Status of digital outputs with time stamp function ..... 358
8.19.10 Status of digital outputs with PWM function ..... 359
8.20 Index table ..... 360
9 PROFIBUS communication ..... 386
9.1 Via Profibus-DP ..... 386
9.2 System configuration ..... 387
9.2.1 Types ..... 387
9.2.2 Mono-master system ..... 387
9.2.3 Multi-master system ..... 388
9.3 Communication ..... 389
9.3.1 Bus access ..... 389
9.3.2 Cyclic data transfer ..... 390
9.3.3 Acyclic data transfer ..... 391
9.3.4 Communication medium ..... 392
9.4 Project planning ..... 393
9.5 Setting the parameters of analog I/O ..... 394
9.5.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400 ..... 394
9.5.2 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401 ..... 396
9.5.3 2 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S402 ..... 398
9.5.4 4 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S403 ..... 400
9.5.5 2 analog inputs -10 ... +10 V (16 bits) - EPM-S406 ..... 402
9.5.6 2 analog inputs 0/4 ... 20 mA (16 bits) - EPM-S408 ..... 405
9.5.7 2 analog outputs 0 ... 10 V (12 bits) - EPM-S500 ..... 408
9.5.8 4 analog outputs $0 \ldots 10 \mathrm{~V}$ (12 bits) - EPM-S501 ..... 410
9.5.9 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502 ..... 412
9.5.10 4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503 ..... 414
9.6 Parameterising the temperature measurement ..... 416
9.6.1 Four (two) analog inputs for resistance tests - EPM-S404 ..... 416
9.6.2 Two analog inputs for thermocouple measurement - EPM-S405 ..... 423
9.7 Parameterising the counter ..... 428
9.7.1 One counter 32 bits, 24 V DC - EPM-S600 ..... 428
9.7.2 Two counters 32 bits, 24 V DC - EPM-S601 ..... 436
9.7.3 One counter 32 bits, 5 V DC - EPM-S602 ..... 443
9.7.4 Two counters 32 bits, 24 V DC - EPM-S603 ..... 450
9.8 Parameterising the encoder evaluation ..... 454
9.8.1 SSI - EPM-S604 ..... 454
9.9 Time stamp parameterising ..... 459
9.9.1 2 digital inputs with time stamp function - EPM-S207 ..... 459
9.9.2 2 digital outputs with time stamp function - EPM-S310 ..... 461
9.10 Parameterising technology modules ..... 464
9.10.1 2 digital outputs with PWM functionality - EPM-S620 ..... 464
9.10.2 RS232 interface - EPM-S640 ..... 467
9.10.3 RS485 interface - EPM-S650 ..... 473
10 EtherCAT communication ..... 480
10.1 About EtherCAT ..... 480
10.1.1 EtherCAT frame ..... 480
10.1.2 EtherCAT datagrams ..... 481
10.1.3 EtherCAT state machine ..... 481
10.2 Transmitting process data ..... 482
10.2.1 Access to I/O area ..... 483
10.3 Transmitting parameter data ..... 484
10.4 General function of the parameter setting ..... 485
10.5 Setting the parameters of analog I/O ..... 486
10.5.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400 ..... 486
10.5.2 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401 ..... 487
10.5.3 2 analog inputs $0 / 4$... 20 mA (12 bits) - EPM-S402 ..... 488
10.5.4 4 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S403 ..... 489
10.5.5 2 analog inputs $-10 \ldots+10 \mathrm{~V}$ (16 bits) - EPM-S406 ..... 490
10.5.6 2 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (16 bits) - EPM-S408 ..... 492
10.5.7 2 analog outputs 0 ... 10 V (12 bits) - EPM-S500 ..... 494
10.5.8 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501 ..... 495
10.5.9 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502 ..... 496
10.5.10 4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503 ..... 497
10.6 Parameterising the temperature measurement ..... 498
10.6.1 Four (two) analog inputs for resistance tests - EPM-S404 ..... 498
10.6.2 Two analog inputs for thermocouple measurement - EPM-S405 ..... 504
10.7 Parameterising the counter ..... 507
10.7.1 One counter 32 bits, 24 V DC - EPM-S600 ..... 507
10.7.2 Two counters 32 bits, 24 V DC - EPM-S601 ..... 511
10.7.3 One counter 32 bits, 5 V DC - EPM-S602 ..... 514
10.7.4 Two counters 32 bits, 24 V DC - EPM-S603 ..... 516
10.8 Parameterising the encoder evaluation ..... 518
10.8.1 SSI - EPM-S604 ..... 518
10.9 Time stamp parameterising ..... 521
10.9.1 2 digital inputs with time stamp function - EPM-S207 ..... 521
10.9.2 2 digital outputs with time stamp function - EPM-S310 ..... 521
10.10 Parameterising technology modules ..... 522
10.10.1 2 digital outputs with PWM functionality - EPM-S620 ..... 522
10.10.2 RS232 interface - EPM-S640 ..... 523
10.10.3 RS485 interface - EPM-S650 ..... 529
10.11 Monitoring ..... 535
10.12 Diagnostics ..... 535
10.12.1 SDO error codes ..... 535
10.12.2 Access to diagnostic data ..... 536
10.12.3 Standard objects ..... 548
10.12.4 Storage of the PDOs ..... 553
10.12.5 Emergency telegram ..... 554
11 DeviceNet communication ..... 557
11.1 About DeviceNet ..... 557
11.2 Access to the I/O system 1000 ..... 559
11.2.1 Access to I/O area ..... 560
11.2.2 Access to parameter data ..... 561
11.2.3 Access to diagnostic data ..... 562
11.3 Setting the parameters of analog $1 / O$ ..... 563
11.3.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400 ..... 563
11.3.2 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401 ..... 564
11.3.3 2 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S402 ..... 565
11.3.4 4 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S403 ..... 566
11.3.5 2 analog inputs $-10 \ldots+10 \mathrm{~V}$ (16 bits) - EPM-S406 ..... 567
11.3.6 2 analog inputs 0/4 ... 20 mA (16 bits) - EPM-S408 ..... 569
11.3.7 2 analog outputs $0 \ldots 10 \mathrm{~V}$ (12 bits) - EPM-S500 ..... 571
11.3.8 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501 ..... 572
11.3.9 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502 ..... 573
11.3.10 4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503 ..... 574
11.4 Parameterising the temperature measurement ..... 575
11.4.1 Four (two) analog inputs for resistance tests - EPM-S404 ..... 575
11.4.2 Two analog inputs for thermocouple measurement - EPM-S405 ..... 581
11.5 Parameterising the counter ..... 584
11.5.1 One counter 32 bits, 24 V DC - EPM-S600 ..... 584
11.5.2 Two counters 32 bits, 24 V DC - EPM-S601 ..... 588
11.5.3 One counter 32 bits, 5 V DC - EPM-S602 ..... 591
11.5.4 Two counters 32 bits, 24 V DC - EPM-S603 ..... 594
11.6 Parameterising the encoder evaluation ..... 596
11.6.1 SSI - EPM-S604 ..... 596
11.7 Time stamp parameterising ..... 599
11.7.1 2 digital inputs with time stamp function - EPM-S207 ..... 599
11.7.2 2 digital outputs with time stamp function - EPM-S310 ..... 599
11.8 Parameterising technology modules ..... 600
11.8.1 2 digital outputs with PWM functionality - EPM -S620 ..... 600
11.8.2 RS232 interface - EPM-S640 ..... 601
11.8.3 RS485 interface - EPM-S650 ..... 607
12 PROFINET communication ..... 613
12.1 About PROFINET ..... 613
12.2 Access to the I/O system 1000 ..... 616
12.2.1 Access to I/O area ..... 618
12.2.2 Access to parameter data ..... 618
12.2.3 Access to diagnostic data ..... 619
12.3 Project planning ..... 621
12.4 I\&M data ..... 623
12.5 Index table ..... 625
12.6 Setting the parameters of analog I/O ..... 626
12.6.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400 ..... 626
12.6.2 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401 ..... 628
12.6.3 2 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S402 ..... 630
12.6.4 4 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S403 ..... 632
12.6.5 2 analog inputs -10 ... +10 V (16 bits) - EPM-S406 ..... 634
12.6.6 2 analog inputs $0 / 4$... 20 mA (16 bits) - EPM-S408 ..... 637
12.6.7 2 analog outputs 0 ... 10 V (12 bits) - EPM-S500 ..... 640
12.6.8 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501 ..... 642
12.6.9 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502 ..... 644
12.6.10 4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503 ..... 646
12.7 Parameterising the temperature measurement ..... 648
12.7.1 Four (two) analog inputs for resistance tests - EPM-S404 ..... 648
12.7.2 Two analog inputs for thermocouple measurement - EPM-S405 ..... 655
12.8 Parameterising the counter ..... 660
12.8.1 One counter 32 bits, 24 V DC - EPM-S600 ..... 660
12.8.2 Two counters 32 bits, 24 V DC - EPM-S601 ..... 668
12.8.3 One counter 32 bits, 5 V DC - EPM-S602 ..... 676
12.8.4 Two counters 32 bits, 24 V DC - EPM-S603 ..... 683
12.9 Parameterising the encoder evaluation ..... 687
12.9.1 SSI - EPM-S604 ..... 687
12.10 Time stamp parameterising ..... 692
12.10.1 2 digital inputs with time stamp function - EPM-S207 ..... 692
12.10.2 2 digital outputs with time stamp function - EPM-S310 ..... 694
12.11 Parameterising technology modules ..... 697
12.11.1 2 digital outputs with PWM functionality - EPM-S620 ..... 697
12.11.2 RS232 interface - EPM-S640 ..... 700
12.11.3 RS485 interface - EPM-S650 ..... 706
13 Modbus TCP communication ..... 713
13.1 About Modbus TCP ..... 713
13.2 Access to the I/O system 1000 ..... 716
13.2.1 Access to I/O area ..... 718
13.2.2 Function codes ..... 719
13.2.3 Access to parameter data ..... 722
13.2.4 Access to diagnostic data ..... 722
13.3 Possible access operations to the EPM-S160 Modbus TCP bus coupler module ..... 724
13.3.1 Web page ..... 724
14 Maintenance ..... 725
14.1 Regular checks ..... 725
14.2 Cleaning ..... 725
14.3 Repair ..... 726
14.3.1 Replacing the electronic module/main supply ..... 726
15 Appendix ..... 727
15.1 Serial process interfacing ..... 727
15.1.1 Protocols ..... 729
15.1.2 I/O range ..... 731
15.1.3 Principle of backplane bus communication ..... 732
16 Index ..... 740

## 1 About this documentation

## 1 <br> About this documentation

## Contents

This documentation informs you about the intended use of the components of the I/O system 1000.

## Target group

This documentation is intended for all persons who design, install, set up, and adjust the I/O system 1000.


## Tip!

Information and auxiliary devices related to the Lenze products can be found in the download area at
http://www.Lenze.com

## Validity

The information given in this documentation applies to the components of the I/O system 1000 according to the following module labelling:


Fig. 1-1 Labelling of the modules
(1) Module designation according to the following table (e.g. AI 2, 12 bits, DC 0 ... 10 V )
(2) Type designation according to the following table (e.g. EPM-S400)
(3) Hardware version/software version according to the following table (e.g. 1A.10)

| Module function | Module designation | Type designation | From hardware version | From software version |
| :---: | :---: | :---: | :---: | :---: |
|  | (1) | (2) | (3) |  |
| Bus coupler module |  |  |  |  |
| CANopen | CANopen | EPM-S110 | 1D | 30 |
| PROFIBUS | PROFIBUS-DP | EPM-S120 | 1D | 30 |
| EtherCAT | EtherCAT | EPM-S130 | 1D | 28 |
| PROFINET | PROFINET | EPM-S140 | 1D | 10 |
| DeviceNet | DeviceNet | EPM-S150 | 1D | 10 |
| Modbus TCP | Modbus TCP | EPM-S160 | 1D | 10 |
| I/O compound module |  |  |  |  |
| Digital I/O |  |  |  |  |
| 2 digital inputs | DI 2, DC 24 V | EPM-S200 | 1B | - |
| 4 digital inputs | DI 4, DC 24 V | EPM-S201 | 1B | - |
| 8 digital inputs | DI 8, DC 24 V | EPM-S202 | 1B | - |
| 4 digital inputs three-wire conductor | DI 4, DC 24 V | EPM-S203 | 1B | - |
| 2 digital inputs NPN | DI 2, NPN, DC 24 V | EPM-S204 | 1B | - |
| 4 digital inputs NPN | DI 4, NPN, DC 24 V | EPM-S205 | 1B | - |
| 8 digital inputs NPN | DI 8, NPN, DC 24 V | EPM-S206 | 1B | - |
| 2 digital inputs, time stamp | DI 2, $2 \mu \mathrm{~s}$, DC 24 V | EPM-S207 | 1B | 10 |
| 2 digital outputs 0.5 A | DO 2, DC $24 \mathrm{~V}, 0.5 \mathrm{~A}$ | EPM-S300 | 1B | - |
| 4 digital outputs 0.5 A | DO 4, DC $24 \mathrm{~V}, 0.5 \mathrm{~A}$ | EPM-S301 | 1B | - |
| 8 digital outputs 0.5 A | DO 8, DC $24 \mathrm{~V}, 0.5 \mathrm{~A}$ | EPM-S302 | 1B | - |
| 2 digital outputs 0.5 A NPN | DO 2, NPN, DC $24 \mathrm{~V}, 0.5 \mathrm{~A}$ | EPM-S303 | 1B | - |
| 4 digital outputs 0.5 A NPN | DO 4, NPN, DC $24 \mathrm{~V}, 0.5 \mathrm{~A}$ | EPM-S304 | 1B | - |
| 8 digital outputs 0.5 A NPN | DO 8, NPN, DC $24 \mathrm{~V}, 0.5 \mathrm{~A}$ | EPM-S305 | 1B | - |
| 2 digital outputs 2 A | DO 2, DC $24 \mathrm{~V}, 2 \mathrm{~A}$ | EPM-S306 | 1B | - |
| 4 digital outputs 2 A | DO 4, DC $24 \mathrm{~V}, 2 \mathrm{~A}$ | EPM-S309 | 1B | - |
| 2 digital outputs, time stamp | DO 2, $1 \mu \mathrm{~s}$, DC $24 \mathrm{~V}, 0.5 \mathrm{~A}$ | EPM-S310 | 1B | 10 |
| 2 relay outputs | Relay 2, AC $230 \mathrm{~V}, 3 \mathrm{~A}$ | EPM-S308 | 1B | - |
| Analog 1/O |  |  |  |  |
| 2 analog inputs, 12 bits, $0 . . .10 \mathrm{~V}$ | Al 2, 12 bits, DC $0 . . .10 \mathrm{~V}$ | EPM-S400 | 1B | 10 |
| 4 analog inputs, 12 bits, $0 . .10 \mathrm{~V}$ | AI 4, 12 bits, DC $0 . . .10 \mathrm{~V}$ | EPM-S401 | 1B | 10 |
| 2 analog inputs, 12 bits, 0/4 ... 20 mA | Al 2, 12 bits, DC 0/4... 20 mA | EPM-S402 | 1B | 10 |
| 4 analog inputs, 12 bits, 0/4 ... 20 mA | Al 4, 12 bits, DC 0/4... 20 mA | EPM-S403 | 1B | 10 |
| 2 analog inputs, $16 \mathrm{Bit}, 0 \ldots 10 \mathrm{~V}$ | Al 2, $16 \mathrm{Bit}, \mathrm{DC} 0 . . .10 \mathrm{~V}$ | EPM-S406 | 1B | 10 |
| 2 analog inputs, 16 Bit, 0/4 ... 20 mA | Al 2, 16 Bit, DC 0/4... 20 mA | EPM-S408 | 1B | 10 |
| 2 analog outputs, 12 bits, 0 ... 10 V | AO 2, 12 bits, DC $0 . . .10 \mathrm{~V}$ | EPM-S500 | 1B | 10 |
| 4 analog outputs, 12 bits, 0 ... 10 V | AO 4, 12 bits, DC 0... 10 V | EPM-S501 | 1B | 10 |
| 2 analog outputs, 12 bits, 0/4 ... 20 mA | $\begin{gathered} \text { AO 2, } 12 \text { bits, DC } \\ 0 / 4 \ldots 20 \mathrm{~mA} \end{gathered}$ | EPM-S502 | 1B | 10 |
| 4 analog outputs, 12 bits, 0/4 ... 20 mA | $\begin{gathered} \text { AO 4, } 12 \text { bits, DC } \\ 0 / 4 \ldots . .20 \mathrm{~mA} \end{gathered}$ | EPM-S503 | 1B | 10 |
| Temperature measurement |  |  |  |  |
| 4(2) analog inputs resistor | Al 4, 16 bits, resistor | EPM-S404 | 1B | 10 |
| 2 analog inputs thermocouple | Al 2, 16 bits, thermo | EPM-S405 | 1B | 10 |


| Module function | Module designation | Type designation | From hardware version | From software version |
| :---: | :---: | :---: | :---: | :---: |
|  | (1) | (2) | (3) |  |
| Counter |  |  |  |  |
| 1 counter 32 bits, 24 V DC (reading, setting, comparing, time stamp) | Counter 1, DC 24 V | EPM-S600 | 1B | 10 |
| 2 counters, 32 bits, 24 V DC (reading, setting) | Counter 2, DC 24 V | EPM-S601 | 1B | 10 |
| 1 counter, 32 bits, 5 V DC (reading, setting, time stamp) | Counter 1, DC 5 V | EPM-S602 | 1B | 10 |
| 2 counters, 32 bits, 24 V DC (reading) | Counter 2, DC 24 V | EPM-S603 | 1B | 10 |
| Encoder evaluation |  |  |  |  |
| SSI interface | SSI | EPM-S604 | 1B | 10 |
| Technology modules |  |  |  |  |
| 2 digital outputs PWM (output of pulse width modulated signals) | PWM | EPM-S620 | 1B | 10 |
| RS232 interface | RS232 | EPM-S640 | 1B | 10 |
| RS485 interface | RS485 | EPM-S650 | 1B | 10 |
| Power supply modules |  |  |  |  |
| I/O supply | Power, DC 24 V | EPM-S701 | 1B | - |
| I/O supply and electronic supply | Power, DC $24 \mathrm{~V} / 24 \mathrm{~V}$ | EPM-S702 | 1B | - |
| Power distributor modules |  |  |  |  |
| 8 terminals DC 24 V | Supply DC 24 V | EPM-S910 | 1 A | - |
| 8 terminals DC 0 V | Supply DC 0 V | EPM-S911 | 1 A | - |
| 4/4 terminals DC $24 \mathrm{~V} / 0 \mathrm{~V}$ | Supply DC $24 \mathrm{~V} / 0 \mathrm{~V}$ | EPM-S912 | 1 A | - |

### 1.1 Document history

| Materialnummer | Version |  |  | Beschreibung |
| :---: | :---: | :---: | :---: | :---: |
| 13409969 | 7.1 | 01/2015 | TD29 | Technical data for EPM-S310 amended |
| 13409969 | 7.0 | 06/2012 | TD29 | Descriptions of I/O compound modules EPM-S406, EPM-S408 and EPM-S650 supplemented; troubleshooting |
| 13392312 | 6.1 | 02/2012 | TD29 | General revision and error recovery |
| 13376656 | 6.0 | 04/2011 | TD29 | Descriptions added: <br> Bus coupler EPM-S140, EPM-S150, EPM-S160 <br> I/O compound module EPM-S640 <br> Power distributor module EPM-S910, EPM-S911, <br> EPM-S912 |
| 13364945 | 5.0 | 11/2010 | TD29 | Descriptions of I/O compound modules EPM-S207, EPM-S310, EPM-S620 and bus coupler EPM-S130 supplemented; troubleshooting |
| 13339001 | 4.0 | 05/2010 | TD29 | Troubleshooting |
| 13321837 | 3.0 | 04/2010 | TD29 | Descriptions of the I/O compound modules EPM-S306, EPM-S308, EPM-S309, EPM-S600, EPM-S601, EPM-S602, EPM-S603, EPM-S604 supplemented; technical data for all modules revised |
| 13313389 | 2.1 | 08/2009 | TD29 | Technical data for EPM-S120 amended |
| 13313389 | 2.0 | 08/2009 | TD29 | First edition |
| 13297999 | 1.0 | 06/2009 | TD29 | Validation |

## 1.2 <br> Conventions used

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

| Spelling of numbers |  |  |
| :---: | :---: | :---: |
| Decimal separator | Point | In general, the decimal point is used. For instance: 1234.56 |
| Warnings |  |  |
| UL warnings | (11) | Are only given in English. |
| UR warnings | T |  |
| Text |  |  |
| Program name | " " | PC software <br> For example: »Engineer", „Global Drive Control" (GDC) |
| Icons |  |  |
| Page reference | $\square$ | Reference to another page with additional information <br> For instance: 16 = see page 16 |

## 1.3 <br> Notes used

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

## Safety instructions

Structure of safety instructions:


## Danger!

(characterises the type and severity of danger)

## Note

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


STOP Stop!

## Meaning <br> 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 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.

## Danger of property damage.

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

## Application notes

| Pictograph and signal word | Meaning |
| :---: | :---: |
| 1 Note! | Important note to ensure troublefree operation |
| '长’'Tip! | Useful tip for simple handling |
|  | Reference to another documentation |

### 1.4 Terminology used

$\left.\begin{array}{l|l}\text { Term } & \text { Meaning } \\ \hline \text { I/O compound module } & \begin{array}{l}\text { EPM-S2xx, EPM-S3xx, EPM-S4xx, EPM-S5xx, EPM-S6xx; module } \\ \text { of the I/O system 1000 (DI, DO, Al, AO, counter, etc.) }\end{array} \\ \hline \text { Bus coupler, bus coupler module } & \begin{array}{l}\text { EPM-S1xx; for connection of the I/O system 1000 to a fieldbus } \\ \text { system (CANopen, PROFIBUS, etc.). With an integrated DC power } \\ \text { supply unit (main supply) for supply of the bus coupler module } \\ \text { and the connected I/O compound modules via backplane bus. }\end{array} \\ \hline \text { Power supply module } & \begin{array}{l}\text { EPM-S7xx; additional DC power supply unit that is used in } \\ \text { extensive systems if the main supply of the bus coupler is not } \\ \text { sufficient to supply the I/O level and/or the electronics. }\end{array} \\ \hline \text { Power distributor module } & \begin{array}{l}\text { EPM-S9xx; power distributor for the supply of external } \\ \text { consumers via the I/O system 1000 (24 V and/or 0 V) }\end{array} \\ \hline \text { Backplane bus } & \begin{array}{l}\text { The control signals on the process level are transferred by the I/O } \\ \text { compound modules via the internal backplane bus. }\end{array} \\ \hline \text { Ohmic load } & \begin{array}{l}\text { In the technical data, the load capacity at a constant ohmic load } \\ \text { is often characterised by specifying a maximum output current } \\ \text { at signal "1". }\end{array} \\ \hline \begin{array}{l}\text { When the lamp load is specified, the fact is taken into account } \\ \text { that an incandescent lamp has the n-fold starting current } \\ \text { compared to the rated current. Only when the glow wire is } \\ \text { heated, the resistance strongly increases. In the data sheets, the } \\ \text { lamp load is characterised by specification of a power in watts } \\ \text { which is considerably lower than the product of the rated voltage } \\ \text { and the permissible output turrent. The high starting current of } \\ \text { an incandescent lamp is also the reason for the fact that the } \\ \text { maximum switching frequency is lower by a factor of }\end{array} \\ \text { approximately ten than it would be at a constant ohmic load. }\end{array}\right\}$

| Term | Meaning |
| :---: | :---: |
| $\mu \mathrm{s}$ ticker | Some I/O compound modules are provided with an integrated $\mu s$-ticker for the logging of states. The $\mu s$-ticker works with a resolution of $1 \mu \mathrm{~s}$, counts from 0 ... $65535 \mu \mathrm{~s}$ after power-on and then starts with 0 again. |
| Time stamp | By means of the time stamp function, initial states, together with a time value (see $\mu$-ticker) and a consecutive number, can be transferred to the FIFO memory as a time stamp entry. |
| PWM | Pulse width modulation |
| SSI interface | For direct connection of the I/O system to an SSI encoder (EPM-S604) |
| Basic error limit | Accuracy of analog I/Os at $25^{\circ} \mathrm{C}$ relating to the upper limit of effective range (according to EN61131); <br> example: measuring range $=0 \ldots 10 \mathrm{~V}$ or $-10 \mathrm{~V} . . .+10 \mathrm{~V}$; basic error limit $= \pm 0.2 \% \rightarrow 10 \mathrm{~V}^{*} 0.2 / 100=20 \mathrm{mV}$; this means that at $25^{\circ} \mathrm{C}$ a maximum measuring error of $\pm 20 \mathrm{mV}$ can occur. |
| Operational error limit | Accuracy of analog I/Os across the entire admissible temperature range for the module relating to the upper limit of effective range (according to EN61131). |
| Basic conversion time | Time required by an A/D converter to record the measured value. |
| PLC | Programmable Logic Controller |
| "Global Drive Control" (GDC) | Lenze Engineering tools supporting you during the whole life cycle of a machine - from planning to maintenance. |
| "Engineer" |  |
| „PLC Designer" |  |

## 2

Safety instructions

### 2.1 General safety information

## Scope

The following general safety instructions apply to all Lenze drive and automation components.
The product-specific safety and application notes given in this documentation must be observed!

## For your own safety

## Danger!

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

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

This is vital for a safe and trouble-free operation and for achieving the specified product features.
The procedural notes and circuit details provided in this document are proposals which the user must check for suitability for his application. The manufacturer does not accept any liability for the suitability of the specified procedures and circuit proposals.

- Only qualified skilled personnel are permitted to work with or on Lenze drive and automation components.
According to IEC 60364 or CENELEC HD 384, these are persons ...
... who are familiar with the installation, assembly, commissioning and operation of the product,
... possess the appropriate qualifications for their work,
... and are acquainted with and can apply all the accident prevent regulations, directives and laws applicable at the place of use.


## Transport, storage

- Transport and storage in a dry, low-vibration environment without aggressive atmosphere; preferably in the packaging provided by the manufacturer.
- Protect against dust and shocks13
- Comply with climatic conditions according to the technical data.


## Mechanical installation

- Install the product according to the regulations of the corresponding documentation. In particular observe the section "Operating conditions" in the chapter "Technical data".
- Provide for a careful handling and avoid mechanical overload. During handling neither bend components, nor change the insulation distances.
- The product contains electrostatic sensitive devices which can easily be damaged by short circuit or static discharge (ESD). Thus, electronic components and contacts must not be touched unless ESD measures are taken beforehand.


## Electrical installation

- Carry out the electrical installation according to the relevant regulations (e. g. cable cross-sections, fusing, connection to the PE conductor). Additional notes are included in the documentation.
- When working on live products, observe the applicable national regulations for the prevention of accidents (e.g. BGV 3).
- The documentation contains information about EMC-compliant installation (shielding, earthing, arrangement of filters and laying cables). The system or machine manufacturer is responsible for compliance with the limit values required by EMC legislation.
Warning: The controllers are products which can be used in category C2 drive systems as per EN 61800-3. These products may cause radio interference in residential areas. If this happens, the operator may need to take appropriate action.
- For compliance with the limit values for radio interference emission at the site of installation, the components - if specified in the technical data - have to be mounted in housings (e. g. control cabinets). The housings have to enable an EMC-compliant installation. In particular observe that for example control cabinet doors preferably have a circumferential metallic connection to the housing. Reduce openings or cutouts through the housing to a minimum.
- Only plug in or remove pluggable terminals in the deenergised state!


## Commissioning

- If required, you have to equip the system with additional monitoring and protective devices in accordance with the respective valid safety regulations (e. g. law on technical equipment, regulations for the prevention of accidents).


## Operation

- Keep all protective covers and doors closed during operation.


## Safety functions

- Without a higher-level safety system, the described product must neither be used for the protection of machines nor persons.


## Maintenance and servicing

- The components are maintenance-free if the required operating conditions are observed.
- If the cooling air is polluted, the cooling surfaces may be contaminated or the air vents may be blocked. Under these operating conditions, the cooling surfaces and air vents must be cleaned at regular intervals. Never use sharp objects for this purpose!
- After the system has been disconnected from the supply voltage, live components and power connections must not be touched immediately because capacitors may be charged. Please observe the corresponding notes on the device.


## Disposal

- Recycle metals and plastic materials. Ensure professional disposal of assembled PCBs.


## 3 Product description

### 3.1 Device features

The I/O system 1000 can be used to implement complex automation applications. It includes a bus coupler module and several I/O compound modules which use an internal backplane bus to communicate with each other within the station and exchange process data, parameter data, and diagnostic information.

- Modular system
- Mounting on standard DIN rail ( 35 mm )
- The two-piece structure (separation of electronics and process integration) enables a quick exchange of modules in the event of service
- Supply voltage of electronics and process level is separated.
- Wiring level via spring terminal
- Shield connection to standard busbar
- Individual labelling by insertable labels (item designation)
- Creation of electrical isolations by power supply modules


## 3 Product description <br> Application as directed

## 3.2 <br> Application as directed

The I/O system 1000 is applied as directed if it is only used for implementing automation tasks in common industrial and commercial areas. Any other use is not permissible.

A use contrary to the intended purposes is also the case if it bears severe risks or dangers which can cause death, injuries, or damage to material assets if an extremely high level of safety measures is not provided.

The I/O system 1000 may especially not be used ...

- in private areas
- in potentially explosive atmospheres
- in areas with harmful gases, oils, acids, radiation, etc.
- in applications where vibration and impact loads occur which exceed the requirements of the EN 60068-2-6 / EN 60068-2-27
- to execute safety functions, as for example
- in the air-traffic control / in flight control systems
- for monitoring/controlling nuclear reactions
- for monitoring/controlling mass transportation
- for monitoring/controlling medical systems
- for monitoring/controlling weapons systems

In order to ensure the protection of persons and material assets, higher-level safety systems must be used!

## 3.3

## System overview

### 3.3.1 System design

An I/O system 1000 consists of the following modules:

- An EPM-S1xx bus coupler module
- Connection of the aystem to a fieldbus system (CANopen, PROFIBUS, etc.)
- Integrated DC power supply unit for the supply of the bus coupler module and the connected I/O compound modules via the backplane bus.
- Up to 64 EPM-S2xx ... EPM-S6xx I/O compound modules (DI, DO, AI, AO, counter, etc.)


SLIOO10
Fig. 3-1 Standard design
*) If no UL conformity is required, the maximum permissible load for the I/O supply is 10 A .
Electronic supply: 5 V voltage for the supply of the bus coupler module electronics and the I/O-compound modules connected.
I/O supply: 24 V voltage for the power supply of the I/O-compound modules.

## 3 <br> Product description

System overview
System design

If, in the case of great station designs, the power of the main bus coupler supply does not suffice to supply the I/O level and/or the electronics, power supply modules can be used. Each supply provides an individual separate potential area.


Fig. 3-2 Grouping through power supply modules
${ }^{*}$ ) If no UL conformity is required, the maximum permissible load for the I/O supply is 10 A .
For the supply of external consumers via the I/O system 1000 you can use power distributor modules (EPM-S9xx) which provide the 24 V and/or 0 V voltage of the I/O supply via their terminals.

## 3.4

 Module design
### 3.4.1 Bus coupler modules



Fig. 3-3 Components of an EPM-S1xx bus coupler module
(A) Bus coupler with firmly attached base module (inseparable)

B Electronic module (main supply with fuse; spare part order designation: EPM-S700)
C Contact cover for the last module of an I/O system

### 3.4.2 I/O compound modules



Fig. 3-4 Components of an EPM-S2xx ... EPM-S6xx I/O compound module
(A) Base module

B Electronic module

## 3 Product description

Module design
Supply modules

### 3.4.3 Supply modules



Fig. 3-5 Components of an EPM-S7xx power supply module
(A) Base module

B Electronic module

### 3.4.4 Power distributor module



Fig. 3-6 EPM-S9xx power distributor module

### 3.4.5 Accessories



Fig. 3-7 I/O system 1000 accessories
A 35 mm DIN rail (available in specialist shops)
B EPM-S900 busbar support
C $10 \times 3 \mathrm{~mm}$ busbar for shield connection via shield terminals (available in specialist shops)

## 3 Product description

Module design
Function elements

### 3.4.6 Function elements



Fig. 3-8 Standard function elements (on the left bus coupler module, on the rechts I/O compound module)
(A) Base module $\leftrightarrow$ DIN rail locking lever

B Contacts - backplane bus
C Contacts, I/O supply
© Terminals
E Electronic module $\leftrightarrow$ base module locking button
F Status displays (LEDs) with labelling strips

## 1 Note!

The description of the module-specific function elements can be found in the following module descriptions.

### 3.5 Bus coupler modules

### 3.5.1 CANopen - EPM-S110

The bus coupler module represents the interface between the process level (I/O level) and the higher-level fieldbus. The control signals on the process level are transmitted by the I/O compound modules via the internal backplane bus.

## Features

- Up to 64 I/O compound modules can be connected to a CANopen bus coupler module
- Integrated power supply unit for the internal voltage supply and the voltage supply of the connected I/O compound modules
- Power supply unit is fed via an external DC voltage source
- Connected to the CAN bus via a 9-pole Sub-D plug
- Setting of the CAN address and baud ratevia coding switch
- LEDs for status display


## Overview



Fig. 3-9
Elements and circuit diagram of voltage supply
© Displays for station and fieldbus status
B Displays for electronics and I/O supply status
[] Terminals for the voltage supply
© Coding switch for setting the CAN address and baud rate
E Sub-D plug for connection to the fieldbus
F Electronic supply
(G) I/O supply

## 3 Product description

Bus coupler modules
CANopen - EPM-S110

## Status displays

| Fieldbus status LEDs ${ }_{\text {A }}$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | PWR | Green | On: Bus coupler is supplied with voltage |
|  | 2 | SF | Red | On: Station error, station structure does not comply with configuration. |
| $\begin{gathered} 1- \\ 0 \\ 0 \\ 0 \\ \square \end{gathered}$ | 3 | BA | Green | On: Operating mode "Operational" (ready for data exchange) <br> Blinking: Operating mode "Pre-operational" (waiting for parameters) |
| \| | 4 | IF | Red | On: Internal error is pending |
| $\square$ | 5 |  |  |  |
|  | 6 |  |  |  |
| $10-\square$ | 7 |  |  |  |
|  | 8 | - | - | Not assigned |
|  | 9 |  |  |  |
| SL10001 | 10 |  |  |  |

Module status LEDs A


## Control elements

The CAN node address and the baud rate are set via the coding switch.
Setting the baud rate:

| ing switch | N | ress and ba | rate (addr |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Valency | Baud rate [kbps] | Example |  |
|  |  |  |  | Switching status | Baud rate |
| SLIOOO4 | 1 | not assigned | - | - |  |
|  | 2 | 1 | $\begin{aligned} & 0=1000 \\ & 1=500 \\ & 2=250 \\ & 3=125 \end{aligned}$ | 1 | $1+2=3 \rightarrow$ baud rate 125 kbps |
|  | 3 | 2 |  | 1 |  |
|  | 4 | 4 |  | 0 |  |
|  | 5 | 8 | $\begin{aligned} & 4=100 \\ & 5=50 \end{aligned}$ | 0 |  |
|  | 6 | 16 |  | 0 |  |
|  | 7 | 32 | $6=20$ | 0 |  |
|  | 8 | 64 | $8=800$ | 0 |  |

1. Switch off the voltage supply for the I/O system.
2. Set all switches at the coding switch to " 0 ".
3. Switch on the voltage supply for the I/O system.

The LEDs SF, IF and CAN-RUN are blinking with a frequency of 1 Hz .
4. Set the desired baud rate with the coding switch. You have 10 seconds to do this. The IF LED goes off after 10 seconds, and the set baud rate is saved.
You have a further 10 seconds time to set the node address.

## 3 Product description

Bus coupler modules
CANopen - EPM-S110

Setting the node address:
Coding switch - CAN address and baud rate (addr.) $\square$

| View | Pos. | Valency | Example |  |
| :---: | :---: | :---: | :---: | :---: |
|  |  |  | Switching status | Node address |
| SLIOOO4 | 1 | not assigned | - |  |
|  | 2 | 1 | 1 | $1+2+16=19 \rightarrow$ address 19 |
|  | 3 | 2 | 1 |  |
|  | 4 | 4 | 0 |  |
|  | 5 | 8 | 0 |  |
|  | 6 | 16 | 1 |  |
|  | 7 | 32 | 0 |  |
|  | 8 | 64 | 0 |  |

5. Set the node address for the module with the coding switch.

- Device addresses permitted are 1 ... 127.
- Each node address must be assigned only once.


## 1 Note!

The node address can be changed any time by means of the coding switch. The setting is accepted once the supply voltage is switched on.

## Terminals

Module terminals, spring terminals C

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Not assigned | $\xrightarrow[(10 \mathrm{~mm}]{(\text { AWG } 28 \ldots 16)}$ |
|  | 2 | I/O supply +24 V DC |  |
|  | 3 | I/O supply 0 V |  |
| - | 4 | Electronic supply +24 V DC |  |
| $3^{3}$ | 5 | Not assigned |  |
|  | 6 | I/O supply +24 V DC |  |
| $0^{8}{ }^{8}$ | 7 | I/O supply 0 V |  |
| SL10002 | 8 | Electronic supply 0 V |  |

## 1 Note!

- Terminals 2 and 6 as well as 3 and 7 are bridged internally. Please note that the max. permissible bridge current is 5 A .
- Both the I/O supply and the electronic supply are protected against overload internally by a fuse. When the fuses have been tripped, the main supply of the bus coupler (EPM-S700) must be replaced ( $\mathbb{\square}$ 726).

System bus (CAN) / CANopen ("CAN"), 9-pole Sub-D plug F

| View | Pin | Assignment | Explanation |
| :---: | :---: | :---: | :---: |
|  | 1 | - | Not assigned |
| -61 | 2 | CAN-LOW | Data line |
| - 7 | 3 | CAN-GND | Data ground |
| 8 | 4 | - | Not assigned |
| 9 | 5 | - | Not assigned |
| - | 6 | - | Not assigned |
|  | 7 | CAN-HIGH | Data line |
|  | 8 | - | Not assigned |
| epm-to23 | 9 | - | Not assigned |

## STOP Stop!

Housing breakage in the case of a too high tightening torque of the securing screws
If the connector securing screws are tightened too firmly, the housing may break.
Possible consequences:

- The plug connection is no longer secured against tension.
- Enclosure IP20 of the module which is warranted can no longer be guaranteed.


## Protective measures:

- Tighten securing screws without using force (max. 40 Nm ).


## 3 Product description

Bus coupler modules
CANopen - EPM-S110

## Wiring



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

| CAN cable in accordance with ISO 11898-2 |  |
| :---: | :---: |
| Cable type | Paired with shielding |
| Impedance | $120 \Omega$ (95 ... $140 \Omega$ ) |
| Cable resistance/cross-section |  |
| Cable length $\leq 300 \mathrm{~m}$ | $\leq 70 \mathrm{~m} \Omega / \mathrm{m} / 0.25 \ldots 0.34 \mathrm{~mm}^{2}$ (AWG22) |
| Cable length 301 ... 1000 m | $\leq 40 \mathrm{~m} \Omega / \mathrm{m} / 0.5 \mathrm{~mm}^{2}$ (AWG20) |
| Signal propagation delay | $\leq 5 \mathrm{~ns} / \mathrm{m}$ |

More information on the system bus (CAN) / CANopen can be found in the chapter "CANopen communication" ( $\square 250)$.

## Accessories

- CAN bus plug "Node" - EPM-T950
- SUB-D, $90^{\circ}$
- Screw terminals
- CAN bus plug "Termination" - EPM-T951
- SUB-D, $90^{\circ}$
- Screw terminals
- Integrated terminating resistor
- CAN bus plug "Even" - EPM-T952
- SUB-D, $180^{\circ}$
- Screw terminals
- Switched terminating resistor
- CAN bus plug "Switch" - EWZ0046
- SUB-D, $90^{\circ}$
- Spring-type terminals
- Switched terminating resistor

Technical data

| EPM-S110: Rated data |  |
| :---: | :---: |
| Electrical data |  |
| Supply voltage |  |
| Nominal value | DC 24 V |
| Permissible range | DC 20.4 ... 28.8 V |
| Current consumption |  |
| Nominal value | 0.95 A |
| In idle state | 0.09 A |
| Starting current | 2.8 A |
| ${ }^{2} \mathrm{t}$ | $0.25 \mathrm{~A}^{2} \mathrm{~s}$ |
| Current output, max. |  |
| At the backplane bus | 3 A |
| Load supply | 7 A (if no UL conformity is required: max. 10 A ) |
| Polarity reversal protection | Yes |
| Power loss | 3 W |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | No |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes, parameterisable |
| Diagnostic information can be read out | Possible |
| Supply voltage display | Green LED |
| Group error display | Red LED |
| Channel error display | None |
| System limits |  |
| Mounting racks, max. | 1 |
| Modules per mounting rack | 64 |
| Number, max. |  |
| I/O compound modules | 64 (depends on current consumption) |
| Inputs/outputs | 128 bytes /128 bytes ( 128 bytes = 16 PDOs of 8 bytes each) |
| Digital inputs/outputs | 512 |
| Analog inputs/outputs | 36 |
| Counter | 4 |
| SSI | 8 |
| Digital inputs, time stamp | 4 |
| Digital outputs, time stamp | 4 |
| Digital outputs, PWM | 4 |


| EPM-S110: Rated data |
| :--- |
| Communication <br> Fieldbus <br> Physics <br> Connection <br> Topology <br> Electrical isolation <br> Nodes <br> Number, max. CAN <br> Address Linear bus with bus termination at both ends <br> Transmission speed 127 <br> Min. $1 \ldots 127$ <br> Max. 10 kbps <br> Address range 1 Mbps <br> $\frac{\text { Inputs, max. }}{}$ 128 bytes <br> Outputs, max. 128 bytes <br> Number of TxPDOs/RxPDO, max. $16 / 16$ <br> for controller-based automation $10 / 10$ |

## Calculation example for determining the maximum number of modules

## 1 Note!

If the CANopen bus coupler is used in conjunction with Lenze drives as part of Drive-based Automation (i.e. as a "terminal extension" to the drive), the PDO configuration of the factory adjustment is used (see examples below):

- PDO1 only purely digital I/O compound modules
- PDO2 only purely analog I/O compound modules
- as of PDO3 counter, rapid I/Os with time stamp function etc.

With Controller-based Automation however all 16 PDOs are available in any distribution.

Example 1: Maximum configuration with digital inputs/outputs
In the default setting, 9 Rx-/Tx-PDOs are provided for digital inputs/outputs (DI/DO).
$9 \times 8$ bytes $=72$ bytes $=576$ bits
Since only 64 I/O compound modules are permissible, max. $64 \times 8$ bits (DI8/DO8) $=512$ channels can be used.
Example 2: Maximum configuration with analog inputs/outputs
In the default setting, 9 Rx-/Tx-PDOs are provided for analog inputs/outputs (AI/AO).
$9 \times 8$ bytes $=72$ bytes $=576$ bits
Since 2 bytes are always occupied by an analog channel, max. $72: 2=36 \mathrm{Al} / \mathrm{AO}$ channels can be used.

### 3.5.2 <br> PROFIBUS - EPM-S120

The bus coupler module represents the interface between the process level (I/O level) and the higher-level fieldbus. The control signals on the process level are transmitted by the I/O compound modules via the internal backplane bus.

## Features

- PROFIBUS-DP slave; supports PROFIBUS-DP-V1
- Up to 64 I/O compound modules can be connected to a PROFIBUS bus coupler module
- Integrated power supply unit for the internal voltage supply and the voltage supply of the connected I/O compound modules
- Power supply unit is fed via an external DC voltage source
- Connection to the PROFIBUS via 9-pin Sub-D socket
- Coding switch for setting the PROFIBUS address
- LEDs for status display


## Overview



Fig. 3-10 Elements and circuit diagram of voltage supply
(A) Displays for station and fieldbus status

B Displays for electronics and I/O supply status
[ Terminals for the voltage supply
© Coding switch for setting the PROFIBUS address
E Sub-D socket for connection to the fieldbus
F Electronic supply
(G) I/O supply

## 3 Product description

Bus coupler modules
PROFIBUS - EPM-S120

## Status displays

| Fieldbus status LEDs $\mathbb{A}$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | PWR | Green | On: Bus coupler is supplied with voltage |
|  | 2 | SF | Red | On: Station error, station structure does not comply with configuration. |
| $\left.1-1 \begin{array}{l} \square \\ 0 \\ 0 \end{array}\right]=$ | 3 | DE | Green | On: "Data Exchange" state Blinking: Bus coupler waiting for parameters |
| $\left\lvert\, \begin{aligned} & 0 \\ & \square \end{aligned}\right.$ | 4 | IF | Red | On: Internal error is pending |
| - | 5 |  |  |  |
| $\left.\left\lvert\, \begin{array}{ll} \square \\ 0 \end{array}\right.\right]$ | 6 |  |  |  |
| $10$ | 7 | - | - | Not assigned |
|  | 8 |  | - | Not assigned |
|  | 9 |  |  |  |
| SL10001 | 10 |  |  |  |

Module status LEDs A


## Control elements

Via coding switch $\square$ the PROFIBUS node address is set. The setting is permanently stored in EEPROM.

| Coding switch - PROFIBUS address (addr.) $\square$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Valency | Example |  |
|  |  |  | Switching status | Node address |
| SLIO004 | 1 | Not assigned | - |  |
|  | 2 | 1 | 1 | 19 dec $\square \mathrm{m} \square$ address: 19 |
|  | 3 | 2 | 1 |  |
|  | 4 | 4 | 0 |  |
|  | 5 | 8 | 0 |  |
|  | 6 | 16 | 1 |  |
|  | 7 | 32 | 0 |  |
|  | 8 | 64 | 0 |  |

How to proceed:

1. Switch off the voltage supply for the I/O system.
2. Set the node address with the coding switch $\mathbb{\square}$.

- Permitted addresses: 1 ... 125
- Each node address within a fieldbus system must be non-ambiguous.

3. Switch on the voltage supply for the I/O system.

## 3 Product description

Bus coupler modules
PROFIBUS - EPM-S120

## Terminals

Module terminals, spring terminals ©

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Not assigned |  |
|  | 2 | I/O supply +24 V DC |  |
|  | 3 | I/O supply 0 V |  |
|  | 4 | Electronic supply +24 V DC |  |
|  | 5 | Not assigned |  |
|  | 6 | I/O supply +24 V DC |  |
|  | 7 | I/O supply 0 V |  |
|  | 8 | Electronic supply 0 V |  |

## (i) Note!

- Terminals 2 and 6 as well as 3 and 7 are bridged internally. Please note that the max. permissible bridge current is 5 A .
- Both the I/O supply and the electronic supply are protected against overload internally by a fuse. When the fuses have been tripped, the main supply of the bus coupler (EPM-S700) must be replaced ( $₫ 7$ 726).

Profibus, 9-pole Sub-D socket E

| View | Pin | Assignment | Explanation |
| :---: | :---: | :---: | :---: |
| EPM-T223 | 1 | - | Not assigned |
|  | 2 | - | Not assigned |
|  | 3 | RxD/TxD-P | Data line B (received / transmitted data plus) |
|  | 4 | RTS | Request To Send (received / transmitted data, no differential signal) |
|  | 5 | M5V2 | Data ground (ground at 5 V ) |
|  | 6 | P5V2 | DC $5 \mathrm{~V} / 30 \mathrm{~mA}$ (bus termination) |
|  | 7 | - | Not assigned |
|  | 8 | RxD/TxD-N | Data line A (received / transmitted data minus) |
|  | 9 | - | Not assigned |

STOP Stop!
Housing breakage in the case of a too high tightening torque of the securing screws
If the connector securing screws are tightened too firmly, the housing may break.
Possible consequences:

- The plug connection is no longer secured against tension.
- Enclosure IP20 of the module which is warranted can no longer be guaranteed.


## Protective measures:

- Tighten securing screws without using force (max. 40 Nm ).


## Wiring



## 1 Note!

The PROFIBUS cable must be terminated with its surge impedance.
(19) More information on the PROFIBUS can be found in the chapter "PROFIBUS communication" (■386).

## 3 Product description

Bus coupler modules
PROFIBUS - EPM-S120

Technical data

| EPM-S120: Rated data |  |
| :---: | :---: |
| Electrical data |  |
| Supply voltage |  |
| Nominal value | DC 24 V |
| Permissible range | DC 20.4 ... 28.8 V |
| Current consumption |  |
| Nominal value | 0.95 A |
| In idle state | 0.09 A |
| Starting current | 2.8 A |
| $1^{2} \mathrm{t}$ | $0.25 \mathrm{~A}^{2} \mathrm{~s}$ |
| Current output, max. |  |
| At the backplane bus | 3 A |
| Load supply | 7 A (if no UL conformity is required: max. 10 A ) |
| Polarity reversal protection | Yes |
| Power loss | 3 W |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes, parameterisable |
| Diagnostic information can be read out | Possible |
| Supply voltage display | Green LED |
| Group error display | Red LED |
| Channel error display | None |
| System limits |  |
| Mounting racks, max. | 1 |
| Modules per mounting rack | 64 |

EPM-S120: Rated data

| Communication |  |
| :---: | :---: |
| Fieldbus | PROFIBUS-DP in accordance with EN 50170 |
| Physics | RS485 insulated |
| Connection | 9-pole Sub-D socket |
| Topology | Linear bus with bus termination at both ends |
| Electrical isolation | Yes |
| Nodes |  |
| Number, max. | 125 |
| Address | 1... 125 |
| Transmission speed | 9.6 kbps ... 12 Mbps |
| Process data for PROFIBUS-DP-V0 |  |
| Input data, max. | 244 bytes |
| Output data, max. | 244 bytes |
| Process data for PROFIBUS-DP-V1 |  |
| Input data, max. | 240 bytes |
| Output data, max. | 240 bytes |
| Parameter data, max. length * | 224 bytes |
| the amount required by an I/O module with ... |  |
| digital inputs or outputs (EPM-S200 ... EPM-S305) | 0 bytes |
| 2 analog inputs (EPM-S400/-S402) | 6 bytes |
| 4 analog inputs (EPM-S401/-S403) | 8 bytes |
| 2 analog outputs (EPM-S500/-S502) | 8 bytes |
| 4 analog outputs (EPM-S501/-S503) | 10 bytes |
| 4 analog inputs res. (EPM-S404) | 34 bytes |
| 2 analog inputs TE (EPM-S405) | 22 bytes |
| 1 counter 32 bits, 24 V DC (EPM-S600) | 21 bytes |
| 2 counters 32 bits, 24 V DC (EPM-S601) | 42 bytes |
| 1 counter 32 bits, 5 V DC (EPM-S602) | 22 bytes |
| 2 counters 32 bits, 24 V DC (EPM-S603) | 8 bytes |
| SSI (EPM-S604) | 33 bytes |
| 2 dig. inputs, time stamp (EPM-S207) | 6 bytes |
| 2 dig. outputs, time stamp (EPM-S310) | 2 bytes |
| 2 dig. outputs, PWM (EPM-S620) | 8 bytes |
| RS232 interface (EPM-S640) | 8/20/60 bytes (parameterisable) |

[^0]
## 3 Product description

Bus coupler modules
EtherCAT - EPM-S130

### 3.5.3 EtherCAT - EPM-S130

The bus coupler module represents the interface between the process level (l/O level) and the higher-level fieldbus. The control signals on the process level are transmitted by the I/O compound modules via the internal backplane bus.

## Features

- EtherCAT bus coupler module for up to 64 peripheral modules
- Ethernet-based fieldbus system with high real-time capability
- Online project planning using Mater system
- Extensive diagnostic functions
- Integrated power supply unit for the internal voltage supply and the voltage supply of the connected I/O compound modules
- Power supply unit supplied via an external DC voltage source
- Connected to fieldbus via RJ45 socket 100BaseTX, 10BaseT
- LEDs for status display
- "Distributed Clock" and "Station Alias" are not supported


## Overview



Fig. 3-11 Elements and circuit diagram of voltage supply
(A)Displays for station and fieldbus status

B Displays for electronics and I/O supply status
(C) Terminals for the voltage supply
(D) Bus interface

E RJ45 socket OUT
F RJ45 socket IN
G Electronic supply
(H) I/O supply

## i] Note!

EtherCAT uses the Ethernet as its transmission medium. Only EtherCAT components may be used in an EtherCAT network. To produce topologies deviating from the line topology, you will need the corresponding EtherCAT components which support such deviations. Hubs cannot be used.
An EtherCAT network always consists of a master and any number of EtherCAT slaves (bus couplers). Each EtherCAT slave has an "IN" and "OUT" RJ45 socket. The EtherCAT cable from the master should be plugged into the socket marked as "IN". The socket marked as "OUT" should be connected to the following node. The "OUT" socket on the last node is free.

## 3 Product description

Bus coupler modules
EtherCAT - EPM-S130

Status displays

| Fieldbus status LEDs $\begin{aligned} & \text { A }\end{aligned}$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | PWR | Green | On: bus coupler is supplied with voltage |
|  | 2 | SF | Red | Flashes at 0.5 Hz : EtherCAT timeout (watchdog) Flashes at 1 Hz : status change due to error Flashes at 2 Hz : configuration error |
|  | 3 | RUN | Green | On: bus coupler in operational status Flashes at 1 Hz : bus coupler in pre-operational status <br> Flashes at 2 Hz : bus coupler in safe operational status <br> Off: bus coupler in initialisation status |
|  | 4 | L/A1 | Green | Off: no communication with preceding EtherCAT node <br> On: preceding EtherCAT node is connected |
|  | 5 | IF1 | Red | On: internal error in communication with preceding EtherCAT node |
|  | 6 | L/A2 | Green | Off: no communication with following EtherCAT node <br> On: following EtherCAT node is connected |
|  | 7 | IF2 | Red | On: internal error in communication with following EtherCAT node |
|  | 8 | - | - | Not assigned |
|  | 9 |  |  |  |
|  | 10 |  |  |  |

Module status LEDs A


## Terminals

## Module terminals, spring terminals C

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Not assigned | 0.08 ... $1.5 \mathrm{~mm}^{2}$ |
|  | 2 | I/O supply +24 V DC |  |
|  | 3 | I/O supply 0 V |  |
| ${ }^{2} 0^{6}$ | 4 | Electronic supply +24 V DC |  |
|  | 5 | Not assigned | (AWG 28 ... 16) |
|  | 6 | I/O supply +24 V DC |  |
|  | 7 | I/O supply 0 V |  |
| SL10002 | 8 | Electronic supply 0 V |  |

## 1 Note!

- Terminals 2 and 6 as well as 3 and 7 are bridged internally. Please note that the max. permissible bridge current is 5 A .
- Both the I/O supply and the electronic supply are protected against overload internally by a fuse. When the fuses have been tripped, the main supply of the bus coupler (EPM-S700) must be replaced ( $\square 726$ ).

EtherCAT, RJ45 socket E, [

| View | Pin | Assignment | Explanation |
| :---: | :---: | :---: | :---: |
| $\square \square$ | 1 | Transmit + | Transmitted data plus |
| $\square$ | 2 | Transmit - | Transmitted data minus |
|  | 3 | Receive + | Received data plus |
|  | 4 | - | Not assigned |
|  | 5 | - | Not assigned |
|  | 6 | Receive - | Received data minus |
|  | 7 | - | Not assigned |
|  | 8 | - | Not assigned |
| SLU0065 | 9 | - | Not assigned |

## Technical data

| EPM-S130: rated data |
| :--- |
| Electrical data <br> Supply voltage  <br> Nominal value DC 24 V <br> Permissible range DC $20.4 \ldots 28.8 \mathrm{~V}$ <br> Current consumption 0.95 A <br> Nominal value 0.095 A <br> In idle state 2.8 A <br> Starting current $0.25 \mathrm{~A}^{2} \mathrm{~s}$ <br> I2t 3 A <br> Current output, max. 7 A (if no UL conformity is required: max. 10 A ) <br> At the backplane bus No <br> Load supply 3 W <br> Polarity reversal protection  <br> Power loss  |

## 3 Product description

Bus coupler modules
EtherCAT - EPM-S130

| EPM-S130: rated data |  |
| :--- | :--- |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes, parameterisable |
| Diagnostic information can be read out | Possible |
| Supply voltage display | Green LED |
| Group error display | Red SF-LED |
| Channel error display | None |
| System limits | 1 |
| Mounting racks, max. | 64 |
| Modules per mounting rack |  |
| Communication | EtherCAT |
| Fieldbus | Ethernet 100 Mbits |
| Physics | RJ45 |
| Connection | Line topology with branches and spur lines |
| Topology | No |
| Electrical isolation | 6553 |
| Nodes | None |
| Number, max. | 100 Mbps |
| Address | 4 kB |
| Transmission speed | 4 kB |
| Address range | 1024 bytes |
| Inputs, max. | 1024 bytes |
| Outputs, max. |  |
| Process data | Input data, max. |
| Output data, max. |  |

### 3.5.4 <br> PROFINET - EPM-S140

The bus coupler module represents the interface between the process level (l/O level) and the higher-level fieldbus. The control signals on the process level are transmitted by the I/O compound modules via the internal backplane bus.

## Features

- PROFINET I/O-Device according to IEC 61158
- Up to 64 I/O compound modules can be connected to a PROFINET bus coupler module
- Integrated power supply unit for the internal voltage supply and the voltage supply of the connected I/O compound modules
- Power supply unit supplied via an external DC voltage source
- Integrated 2-port switch
- Ethernet connection via 2 RJ45 sockets (P1, P2)
- Auto negotiation (negotiating the transmission parameters)
- Auto crossover (transmit and receive path are automatically crossed if required)
- Setting of the PROFINET address via coding switch
- LEDs for status display


## Overview



Fig. 3-12 Elements and circuit diagram of voltage supply
(A) Displays for station and fieldbus status

B Displays for electronics and I/O supply status
C Terminals for the voltage supply
© Coding switch for setting the PROFINET address
E RJ45 socket for connection to the fieldbus (P1)
F RJ45 socket for connection to the fieldbus (P2)
G Electronic supply
H I/O supply

## Status displays

| Fieldbus status LEDs ® $^{\text {a }}$ |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| View |  | Pos. | Designation |  | Colour |  | Explanation |  |
|  |  | 1 | PWR |  | Green |  | On: bus coupler is supplied with voltage |  |
| $1-$ |  | 2 | SF |  | Red |  | On: System error; error at PROFINET or at backplane bus |  |
| 口 |  | 3 | BS |  | Green |  | Bus error; error in PROFINET communication |  |
| $\square$ |  | 4 | MT |  | Yellow |  | Maintenance: PROFINET maintenance request |  |
| - |  | 5 | LINK1 |  | Green |  | Physical connection to PROFINET (P1) |  |
| d |  | 6 | ACT1 |  | Green |  | Communication via PROFINET (P1) |  |
| - |  | 7 | LINK2 |  | Green |  | Physical connection to PROFINET (P2) |  |
| 10- |  | 8 | ACT2 |  | Green |  | Communication via PROFINET (P2) |  |
|  | st10001 | 9 10 |  |  |  |  | Not assigned |  |
| PWR | SF | BS | MT | LINK1 | ACT1 | LINK2 | ACT2 | Status |
| Green | Red | Green | Yellow | Green | Green | Green | Green |  |
| On | - | - | - | - | - | - | - | The PROFINET bus coupler is supplied with voltage. |
| On | Off | 0.5 Hz | - | [On] | - | [On] | - | It is not possible to establish a connection with the PROFINET I/O controller. A connection to the switch, however, exists (no AR is active) LNK1 or LNK2 is on. |
| On | Off | On | - | Off | Off | Off | Off | There is no physical connection to the Ethernet. LNK1 and LNK2 are off. |
| On | - | Off | - | [On] | Pulse | [On] | Pulse | A connection to a PROFINET I/O controller has been established (at least one AR is active) LNK1 or LNK2 is on. |
| On | On | - | - | - | - | - | - | - A diagnostic message not yet acknowledged is available. <br> - Error at the backplane bus (e.g. module defective, bus disturbed). <br> - Error at firmware update (only visible for a short time, afterwards restart). |
| On | 2 Hz | On | - | On | - | On | - | IP address error <br> - No valid IP address has been assigned. <br> - The assigned IP address already exists in the system. |
| On | - | 1 Hz | 1 Hz | - | - | - | - | A firmware update is currently being executed. Here, BS and MT are blinking alternately. |
| On | - | - | - | 2 Hz | - | 2 Hz | - | Identification via DCP. Depending on the connection, LINK1 or LINK2 is blinking for 3 seconds with 2 Hz . |
| On | On | - | On | - | - | - | - | Maintenance request (Maintenance demanded/requested) <br> - After the coupler has been parameterised, no sync frame has been received. <br> - Jitter is outside the limits (renewed synchronisation). <br> - Switch has rejected 10 frames (network overloaded). <br> - Error at the system SLIO bus (version error). |

[^1]

## Control elements

## Coding switch PROFINET address $\square$

| View | Pos. | Valency | Example |  |
| :---: | :---: | :---: | :---: | :---: |
|  |  |  | Switching status | Node address |
| SLIO004 | 1 | - | - | No function |
|  | 2 | 1 | 1 | Profinet name: "EPM-S140-xxx" <br> with $x x x=$ decimal value of position 2 ... 8 ; for example: $19_{\mathrm{dec}} \rightarrow \mathrm{xxx}=19$ |
|  | 3 | 2 | 1 |  |
|  | 4 | 4 | 0 |  |
|  | 5 | 8 | 0 |  |
|  | 6 | 16 | 1 |  |
|  | 7 | 32 | 0 |  |
|  | 8 | 64 | 0 |  |
| Important switch positions |  |  |  |  |
| Pos. |  | Status | Behaviour at restart |  |
| 2 ... 8 |  | 0 | Profinet-compliant (IEC 61158-6-10, IEC 61784-2) <br> - IP address./subnet mask comes from flash memory. <br> - Profinet name comes from flash memory. |  |
| $2 . .8$ |  | [1 ... 127] | - IP address./subnet mask comes from flash memory. <br> - Profinet name: EPM-S140-xxx (with xxx = decimal value of position 2...8): Profinet name with I/O controller cannot be changed. |  |

## 3 Product description

Bus coupler modules
PROFINET - EPM-S140

Terminals
Module terminals, spring terminals ©

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Not assigned | - $0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
|  | 2 | I/O supply +24 V DC |  |
|  | 3 | I/O supply 0 V |  |
|  | 4 | Electronic supply +24 V DC |  |
|  | 5 | Not assigned | (AWG 28 ... 16) |
|  | 6 | I/O supply +24 V DC |  |
| $4-4$ | 7 | I/O supply 0 V |  |
| su10002 | 8 | Electronic supply 0 V |  |

1 Note!

- Terminals 2 and 6 as well as 3 and 7 are bridged internally. Please note that the max. permissible bridge current is 5 A .
- Both the I/O supply and the electronic supply are protected against overload internally by a fuse. When the fuses have been tripped, the main supply of the bus coupler (EPM-S700) must be replaced ( $■ 726$ ).

PROFINET, RJ45 socket ${ }^{\text {D }}$

| View | Pin | Assignment | Explanation |
| :---: | :---: | :---: | :---: |
|  | 1 | Transmit + | Transmitted data plus |
|  | 2 | Transmit - | Transmitted data minus |
|  | 3 | Receive + | Received data plus |
| SL10065 | 4 | GND | Ground |
|  | 5 | GND | Ground |
|  | 6 | Receive - | Received data minus |
|  | 7 | GND | Ground |
|  | 8 | GND | Ground |

## Technical data

## Rated data EPM-S140

## Electrical data

| Supply voltage |  |
| :--- | :--- |
| Nominal value | DC 24 V |
| Permissible range | DC $20.4 \ldots 28.8 \mathrm{~V}$ |
| Current consumption | 0.95 A |
| Nominal value | 0.095 A |
| In idle state | 2.8 A |
| Starting current | $0.25 \mathrm{~A}^{2} \mathrm{~s}$ |
| It | 3 A |
| Current output, max. | 7 A (if no UL conformity is required: max. 10 A$)$ |
| At the backplane bus | Yes |
| Load supply | 3 W |
| Polarity reversal protection |  |
| Power loss |  |


| Rated data EPM-S140 |  |
| :--- | :--- |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes, parameterisable |
| Diagnostic information can be read out | Possible |
| Supply voltage display | Green LED |
| Maintenance display | Yellow LED |
| Group error display | Red LED |
| Channel error display | None |
| System limits |  |
| Mounting racks, max. |  |
| Modules per mounting rack | 64 |
| Communication | PROFINET-IO |
| Fieldbus | Ethernet 100 Mbits |
| Physics | $2 \times$ RJ45 |
| Connection | Yes |
| Electrical isolation | 100 Mbps |
| Transmission speed | 512 bytes |
| Address range | 512 bytes |
| Inputs, max. |  |
| Outputs, max. |  |

## 3 Product description

Bus coupler modules
DeviceNet - EPM-S150

### 3.5.5 <br> DeviceNet - EPM-S150

The bus coupler module represents the interface between the process level (l/O level) and the higher-level fieldbus. The control signals on the process level are transmitted by the I/O compound modules via the internal backplane bus.

## Features

- DeviceNet coupler
- Up to 64 I/O compound modules can be connected to a DeviceNet bus coupler module
- Group 2 only Device uses Predefined Connection Set
- Poll only Device
- No operating mode BIT STROBE
- NO operating mode CHANGE OF STATE
- Profile Generic Device
- Support of all baud rates: 125, 250 und 500 kbps
- Max. 255 bytes input/output data
- Integrated power supply unit for the internal voltage supply and the voltage supply of the connected I/O compound modules
- Power supply unit supplied via an external DC voltage source
- Connection to the DeviceNet via "Open Style Connector"
- Setting of the DeviceNet address and the baud rate via coding switch
- LEDs for status display


## Overview



Fig. 3-13 Elements and circuit diagram of voltage supply
(A) Displays for station and fieldbus status

B Displays for electronics and I/O supply status
C Terminals for the voltage supply
(D Coding switch for setting the DeviceNet address
E Fieldbus connection
F Electronic supply
( I/O supply

## Status displays

Fieldbus status LEDs $\AA$


## 3 Product description

Bus coupler modules
DeviceNet - EPM-S150

| SF | RD | BA | State |
| :---: | :---: | :---: | :---: |
| Red | Green | Yellow |  |
| On | Off | Off | Invalid switch position at the address switch |
| Off | On | Off | Baud rate has been accepted successfully |
| 1 Hz | 1 Hz | 1 Hz | Firmware update is running |
| On | On | On | Firmware update completed successfully |
| 2 Hz | On | On | Firmware update completed with errors <br> - EDS file defective <br> - Transmission error <br> - Flash error |
| - | - | Off | Fieldbus is offline <br> - No DC 24 V at the fieldbus plug <br> - No further node at the fieldbus |
| - | - | 1 Hz | Fieldbus is ready <br> - No connection established |
| - | - | On | Fieldbus connected |
| On | - | Off | Fieldbus error <br> - Fieldbus address already exists <br> - Communication error at the fieldbus |
| On | - | 1 Hz | Inactivity monitoring <br> - Connection has been terminated due to inactivity |
| - | 1 Hz | - | System SLIO Bus ready, outputs inactive |
| - | On | - | System SLIO bus active |
| On | 2 Hz | - | Access error to system SLIO bus |
| 2 Hz | 2 Hz | - | Configuration error of system SLIO bus <br> - The numbers of plugged and projected modules are different. <br> - The module type of at least one plugged module differs from the parameterised module type. |
| 1 Hz | 2 Hz | 2 Hz | Device error / internal error Please contact the Lenze service! |

"-": Not relevant; "x Hz": Blinking with x Hz
Module status LEDs ©


## Control elements

The coding switch serves for the following settings:

- Baud rate
- DeviceNet address
- Update mode for firmware update


## 1 Note!

Changes at the coding switch get only effective after PowerON or automatic reset. Changes during operation will not be recognised!

Setting the baud rate:
Every station on the DeviceNet communicates with the same transfer rate. Proceed as follows to define the baud rate:

1. Switch off the voltage supply.
2. Set the switch 1 to " 1 " (= baud rate).
3. Set the baud rate via switches 2 ... 4 according to the following table.

| Baud rate D |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Description | Example |  |
|  |  |  | Switching status | Result |
| SLIOOO4 | 1 | Configuration | 1 | 1: Setting of baud rate/update mode |
|  | 2 | Baud rate 125 kbps | 0 | Baud rate: 250 kbps |
|  | 3 | Baud rate 250 kbps | 1 |  |
|  | 4 | Baud rate 500 kbps | 0 |  |
|  | 5 | Not assigned | - |  |
|  | 6 | Not assigned | - |  |
|  | 7 | Update mode | 0 |  |
|  | 8 | Not assigned | - |  |

4. Switch on the voltage supply.

The set baud rate is stored in the EEPROM.

Setting the DeviceNet address:
All stations on the bus must be uniquely identified by means of a DeviceNet address which is between 0 and 63. Proceed as follows to define the baud rate:

1. Switch off the voltage supply.
2. Set the switch 1 to " 0 " (address).
3. Set the address via switches $2 . . .4$ according to the following table.

| DeviceNet address $\square$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Valency | Example |  |
|  |  |  | Switching status | Result |
| SLIO004 | 1 | Configuration | 0 | 0: Address setting |
|  | 2 | 1 | 1 | $1+2+16=19 \square \mathrm{~m} \square$ address: 19 |
|  | 3 | 2 | 1 |  |
|  | 4 | 4 | 0 |  |
|  | 5 | 8 | 0 |  |
|  | 6 | 16 | 1 |  |
|  | 7 | 32 | 0 |  |
|  | 8 | Not assigned | - |  |

4. Switch on the voltage supply.

The set node address is saved in the EEPROM. In case of a wrong or already available address the SF-LED (red) is lit after PowerON.

Setting the update mode:
Set the update mode to transmit a firmware to the bus coupler.
How to proceed:

1. Switch off the voltage supply.
2. Set the switch 1 to " 1 " (= baud rate/update mode).
3. Set the update mode via switches $2 . . .4$ according to the following table.

| Baud rate ${ }^{\text {D }}$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Description | Example |  |
|  |  |  | Switching status | Result |
| SLIOOO4 | 1 | Configuration | 1 | 1: Setting of baud rate/update mode |
|  | 2 | Baud rate 125 kbps | 0 | Update mode active |
|  | 3 | Baud rate 250 kbps | 0 |  |
|  | 4 | Baud rate 500 kbps | 0 |  |
|  | 5 | Not assigned | - |  |
|  | 6 | Not assigned | - |  |
|  | 7 | Update mode | 1 |  |
|  | 8 | Not assigned | - |  |

4. Switch on the voltage supply.

The firmware is copied to the bus coupler.

## 3 Product description

Bus coupler modules
DeviceNet - EPM-S150

## Terminals

Module terminals, spring terminals $\mathbf{C}$

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Not assigned |  |
|  | 2 | I/O supply +24 V DC |  |
|  | 3 | I/O supply 0 V |  |
|  | 4 | Electronic supply +24 V DC |  |
|  | 5 | Not assigned |  |
|  | 6 | I/O supply +24 V DC |  |
|  | 7 | I/O supply 0 V |  |
|  | 8 | Electronic supply 0 V |  |

(i) Note!

- Terminals 2 and 6 as well as 3 and 7 are bridged internally. Please note that the max. permissible bridge current is 5 A .
- Both the I/O supply and the electronic supply are protected against overload internally by a fuse. When the fuses have been tripped, the main supply of the bus coupler (EPM-S700) must be replaced ( $\subseteq$ 726).

DeviceNet, 5-pole socket E

| View | Pin | Assignment | Explanation |
| :---: | :---: | :---: | :---: |
| SLIO066 | 1 | V- | GND, optional voltage supply |
|  | 2 | CL | CAN Low |
|  | 3 | DR | DRAIN / shield |
|  | 4 | CH | CAN High |
|  | 9 | V+ | DC 24 V , optional voltage supply |

## STOP Stop!

Housing breakage in the case of a too high tightening torque of the securing screws
If the connector securing screws are tightened too firmly, the housing may break.
Possible consequences:

- The plug connection is no longer secured against tension.
- Enclosure IP20 of the module which is warranted can no longer be guaranteed.


## Protective measures:

- Tighten securing screws without using force (max. 40 Nm ).


## Wiring



More information on the DeviceNet can be found in the chapter "DeviceNet communication" (■557).

Technical data
EPM-S150: Rated data

## Electrical data

| Supply voltage |  |
| :--- | :--- |
| Nominal value | DC 24 V |
| Permissible range | DC $20.4 \ldots 28.8 \mathrm{~V}$ |
| Current consumption 0.95 A <br> Nominal value 0.09 A <br> In idle state 2.8 A <br> Starting current 0.25 A$^{2} \mathrm{~s}$ <br> $\mathrm{I}^{2} \mathrm{t}$ 3 A <br> Current output, max. 7 A (if no UL conformity is required: max. 10 A) <br> At the backplane bus Yes <br> Load supply 3 W <br> Polarity reversal protection  <br> Power loss Yes <br> Status, alarm, diagnostics None <br> Status display None <br> Alarms None <br> Process alarm None <br> Diagnostic alarm Possible <br> Diagnostic function Green LED <br> Diagnostic information can be read out Red SF-LED <br> Supply voltage display None <br> Group error display  <br> Channel error display  |  |

## 3 Product description

Bus coupler modules
DeviceNet - EPM-S150

| EPM-S150: Rated data |
| :--- |
| System limits <br> Mounting racks, max.  <br> Modules per mounting rack 1 <br> Communication 64 <br> Fieldbus DeviceNet <br> Physics CAN <br> Connection 5 -pole open-style connector <br> Topology Line with bus termination at both ends <br> Electrical isolation Yes <br> Nodes 64 <br> Number, max. $0 \ldots 63$ <br> Address 125 ... 500 kbps <br> Transmission speed 256 bytes <br> Process data 256 bytes <br> Address range input data, max.  <br> Address range output data, max.  |

### 3.5.6 <br> Modbus TCP- EPM-S160

The bus coupler module represents the interface between the process level (l/O level) and the higher-level fieldbus. The control signals on the process level are transmitted by the I/O compound modules via the internal backplane bus.

## Features

- Ethernet slave with Modbus TCP protocol
- Up to 64 I/O compound modules can be connected to an bus coupler module
- Integrated power supply unit for the internal voltage supply and the voltage supply of the connected I/O compound modules
- Power supply unit supplied via an external DC voltage source
- I/O access of up to 8 stations
- Online parameter setting via integrated web server
- RJ45 socket 100BaseTX, 10BaseTX
- Automatic polarity and speed detection (auto negotiation)
- Automatic detection of parallel or crossed cables (auto crossover)
- LEDs for status display


## Overview



Fig. 3-14 Elements and circuit diagram of voltage supply
(A) Displays for station and fieldbus status

B Displays for electronics and I/O supply status
(C) Terminals for the voltage supply
(D Fieldbus connection
E Electronic supply
F I/O supply

## 3 Product description

Bus coupler modules
Modbus TCP- EPM-S160

## Status displays

Fieldbus status LEDs $\AA$

| View | Pos. | Designation | Colour | Explanation |
| :---: | :---: | :---: | :---: | :---: |
|  | 1 | PWR | Green | On: Bus coupler is supplied with voltage |
|  | 2 | SF | Red | On: System error; error at Ethernet or at backplane bus |
| $1$ | 3 | RUN | Green | Bus coupler status |
|  | 4 | MT | Yellow | Bus coupler is localised |
|  | 5 | L/A | Green | On: Ethernet is connected physically Blinking: Bus activity |
| $\begin{aligned} & \square \\ & \square \\ & \square \end{aligned}$ | 6 | SPD | Green | On: Speed 100 Mbps Off: Speed 10 Mbps |
| 10- | 7 | - | - | Not assigned |
|  | 8 |  |  |  |
|  | 9 |  |  |  |
| SLO001 | 10 |  |  |  |


| PWR | SF | RUN | MT | L/A | SPD | State |
| :---: | :---: | :---: | :---: | :---: | :---: | :--- | :--- |
| Green | Red | Green | Yellow | Green | Green |  |
| On | - | - | - | - | - | Bus coupler is supplied with voltage. |
| On | Off | On | - | On | - | Bus coupler is communicating via Ethernet; <br> no errors are pending. |
| On | - | - | - | Off | Off | There is no physical connection to the <br> Ethernet. |
| On | On | Off | - | - | - | Error of Ethernet communication <br> IP address error <br> ( Error in DHCP setting <br> - Defective module connected |
| On | B2 | Off | - | - | - | Backplane bus error; module is not <br> supported |
| On | B3 | Off | - | - | - | Backplane bus error; parameterisation error |
| On | - | - | B1 | - | - | Bus coupler is localised; identification. <br> Blinking lasts for 10 s. |

"-":not relevant
"B1": Blinking code with a period of 1 s "off-off-on-on"
"B2": Blinking code with a period of 1 s "on-on-on-on-off"
"B3": Blinking code with a period of 1 s "off-off-off-off-on"

## Module status LEDs ©

| View |  |  |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |

## Terminals

Module terminals, spring terminals ©

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Not assigned | $(/ \mathrm{ll/}$ |
|  | 2 | I/O supply +24 V DC |  |
|  | 3 | I/O supply 0 V |  |
|  | 4 | Electronic supply +24 V DC |  |
|  | 5 | Not assigned |  |
|  | 6 | I/O supply +24 V DC |  |
|  | 7 | I/O supply 0 V |  |
|  | 8 | Electronic supply 0 V |  |

1 Note!

- Terminals 2 and 6 as well as 3 and 7 are bridged internally. Please note that the max. permissible bridge current is 5 A .
- Both the I/O supply and the electronic supply are protected against overload internally by a fuse. When the fuses have been tripped, the main supply of the bus coupler (EPM-S700) must be replaced ( $\square 726$ ).


## Modbus TCP, 5-pole socket $\square$

| View | Pin | Assignment | Explanation |
| :---: | :---: | :---: | :---: |
|  | 1 | Transmit + | Transmitted data plus |
| memman | 2 | Transmit - | Transmitted data minus |
|  | 3 | Receive + | Received data plus |
|  | 4 | - | Not assigned |
|  | 5 | - | Not assigned |
|  | 6 | Receive - | Received data minus |
|  | 7 | - | Not assigned |
| SLIO065 | 8 |  |  |

## Technical data

## EPM-S160: Rated data

## Electrical data

| Supply voltage |  |
| :---: | :---: |
| Nominal value | DC 24 V |
| Permissible range | DC 20.4 ... 28.8 V |
| Current consumption |  |
| Nominal value | 0.95 A |
| In idle state | 0.095 A |
| Starting current | 2.8 A |
| 12 t | $0.25 \mathrm{~A}^{2} \mathrm{~s}$ |
| Current output, max. |  |
| At the backplane bus | 3 A |
| Load supply | 7 A (if no UL conformity is required: max. 10 A ) |
| Polarity reversal protection | Yes |
| Power loss | 3 W |

## 3 Product description

## Bus coupler modules

Modbus TCP- EPM-S160

| Status, alarm, diagnostics |  |
| :---: | :---: |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes, parameterisable |
| Diagnostic information can be read out | Possible |
| Supply voltage display | Green LED |
| Maintenance display | Yellow LED |
| Group error display | Red LED |
| Channel error display | None |
| System limits |  |
| Mounting racks, max. |  |
| Modules per mounting rack | 64 |
| Communication |  |
| Fieldbus | Modbus / TCP/IP |
| Physics | Ethernet 10/100 Mbps |
| Connection | RJ45 |
| Electrical isolation | Yes |
| Transmission speed | 10 ... 100 Mbps |
| Address range |  |
| Inputs, max. | 1 kB |
| Outputs, max. | 1 kB |

$3.6 \quad$ I/O compound modules - digital I/O

### 3.6.1 Two digital inputs - EPM-S200 and EPM-S204 (NPN)

This module detects up to two binary control signals from the process level and transmits them to the higher-level bus system.

## Features

- 2 digital inputs (EPM-S204: N-switching)
- Suitable for switches and proximity switches
- LEDs display the switching states of the digital inputs


## Overview



Fig. 3-15 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number
(0) ... (7) Bit number in bit presentation

## 3 Product description

I/O compound modules - digital I/O
Two digital inputs - EPM-S200 and EPM-S204 (NPN)

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
| $1-$ | 2 | MF | Red | On: Module error (see table below) |
| $\square$ | 3 | DI1 |  |  |
| $\square-$ | 4 | DI2 | Green | On: Digital input triggered |
|  | 5 |  |  |  |
| $\square$ | 6 |  |  |  |
|  | 7 |  |  |  |
| $10-\square$ | 8 | - | - | Not assigned |
|  | 9 |  |  |  |
| su0001 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■) 249) |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
|  | 1 | Digital input DI1 |  |
|  | 2 | 24 V DC |  |
|  | 3 | GND |  |
|  | 4 | Not assigned |  |
|  | 5 | Digital input DI2 |  |
|  | 6 | 24 V DC |  |
|  | 7 | GND |  |
|  | 8 | Not assigned |  |

## Technical data

| EPM-S200 / EPM-S204 (NPN): Rated data |  |
| :---: | :---: |
| Module identifier | EPM-S200: $1_{\text {dec }} ;$ EPM-S204: $2_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | EPM-S200: 55 mA ; EPM-S204: 60 mA |
| Power loss | 0.5 W |
| Digital inputs |  |
| Number of inputs | 2 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Input voltage |  |
| Nominal value | DC 24 (DC 20.4... 28.8 V) |
| for signal "0" | EPM-S200: DC 0 ... 5 V ; EPM-S204: DC 15 ... 28.8 V |
| for signal "1" | EPM-S200: DC 15 ... 28.8 V; EPM-S204: DC 0 ... 5 V |
| Input current |  |
| for signal "1" | 3 mA |
| 2-wire BERO |  |
| Connection possible | Yes |
| Max. permissible closed-circuit current | 0.5 mA |
| Input delay |  |
| from "0" to "1" | 3 ms |
| from " 1 " to "0" | 3 ms |
| Number of inputs which can be used simultaneously |  |
| horizontal structure | 2 |
| vertical structure | 2 |
| Input characteristic | EPM-S200: IEC 61131, type 1; EPM-S204: none |
| Input data size | 8 bits (with EPM-S110: 2 bits) |
| Status, alarm, diagnostics |  |
| Status display | Green LEDs per channel |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | None |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound modules - digital I/O
Four digital inputs - EPM-S201 and EPM-S205 (NPN)

### 3.6.2 Four digital inputs - EPM-S201 and EPM-S205 (NPN)

This module detects up to four binary control signals from the process level and transmits them to the higher-level bus system.

## Features

- 4 digital inputs (EPM-S205: N-switching)
- Suitable for switches and proximity switches
- LEDs display the switching states of the digital inputs


## Overview



Fig. 3-16 Elements and circuit diagram

$$
\begin{array}{ll}
\text { A } & \text { Displays for module status } \\
\text { B } & \text { Terminals } \\
1 \ldots 8 & \text { Connection number } \\
\text { © ... (8) } & \text { Bit number in bit presentation }
\end{array}
$$

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
| 1 | 2 | MF | Red | On: Module error (see table below) |
| - - | 3 | DI1 | Green | On: Digital input triggered |
| $\square-$ | 4 | DI2 |  |  |
| $\begin{aligned} & \square \\ & 0 \\ & \square \end{aligned}$ | 5 | DI3 |  |  |
|  | 6 | DI4 |  |  |
|  | 7 | - | - | Not assigned |
| $10-\square$ | 8 |  |  |  |
|  | 9 |  |  |  |
| SLIOOO1 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (Cl 249) |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
|  | 1 | Digital input DI1 | - $0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
|  | 2 | 24 V DC |  |
|  | 3 | Digital input DI3 |  |
|  | 4 | 24 V DC |  |
|  | 5 | Digital input DI2 | $\xrightarrow[(10 \mathrm{~mm}]{(\text { AWG } 28 \ldots 16)}$ |
|  | 6 | 24 V DC |  |
|  | 7 | Digital input DI4 |  |
| sulouer | 8 | 24 V DC |  |

## 3 Product description

I/O compound modules - digital I/O
Four digital inputs - EPM-S201 and EPM-S205 (NPN)

## Technical data

| EPM-S201 / EPM-S205 (NPN): rated data |  |
| :---: | :---: |
| Module identifier | EPM-S201: $3_{\text {dec }}$; EPM-S205: $4_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | EPM-S201: 55 mA ; EPM-S205: 65 mA ; |
| Power loss | 0.6 W |
| Digital inputs |  |
| Number of inputs | 4 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Input voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V) |
| for signal "0" | EPM-S201: DC 0 ... 5 V ; EPM-S205: DC 15 ... 28.8 V |
| for signal "1" | EPM-S201: DC 15 ... 28.8 V; EPM-S205: DC 0 ... 5 V |
| Input current |  |
| for signal "1" | 3 mA |
| 2-wire BERO |  |
| Connection possible | Yes |
| Max. permissible closed-circuit current | 0.5 mA |
| Input delay |  |
| from "0" to "1" | 3 ms |
| from " 1 " to "0" | 3 ms |
| Number of inputs which can be used simultaneously |  |
| horizontal structure | 4 |
| vertical structure | 4 |
| Input characteristic | EPM-S201: IEC 61131, type 1; EPM-S205: none |
| Input data size | 8 bits (with EPM-S110: 4 bits) |
| Status, alarm, diagnostics |  |
| Status display | Green LEDs per channel |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | None |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

### 3.6.3 Eight digital inputs - EPM-S202 and EPM-S206 (NPN)

This module detects up to eight binary control signals from the process level and transmits them to the higher-level bus system.

## Features

- 8 digital inputs (EPM-S206: N-switching)
- Suitable for switches and proximity switches
- LEDs display the switching states of the digital inputs


## Overview



Fig. 3-17 Elements and circuit diagram

$$
\begin{array}{ll}
\text { A } & \text { Displays for module status } \\
\text { B } & \text { Terminals } \\
1 \ldots 8 & \text { Connection number } \\
\text { (0) ... © } & \text { Bit number in bit presentation }
\end{array}
$$

## 3 Product description

I/O compound modules - digital I/O
Eight digital inputs - EPM-S202 and EPM-S206 (NPN)

## Status displays

| Module status LEDs A $^{\text {a }}$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| - - | 3 | DI1 | Green | On: Digital input triggered |
| $\square-$ | 4 | DI2 |  |  |
| - | 5 | DI3 |  |  |
| $\begin{aligned} & \square \\ & \square \\ & \hline \end{aligned}$ | 6 | DI4 |  |  |
| $\square$ | 7 | DI5 |  |  |
| 10- | 8 | DI6 |  |  |
|  | 9 | DI7 |  |  |
| SL10001 | 10 | DI8 |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■249) |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
|  | 1 | Digital input DI1 | - $0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
|  | 2 | Digital input DI3 |  |
|  | 3 | Digital input DI5 |  |
| ${ }^{\square}{ }^{\circ}{ }^{6}$ | 4 | Digital input DI7 |  |
|  | 5 | Digital input DI2 | $\xrightarrow[(10 \mathrm{~mm}]{(\text { AWG } 28 \ldots 16)}$ |
|  | 6 | Digital input DI4 |  |
|  | 7 | Digital input DI6 |  |
| SLu0002 | 8 | Digital input DI8 |  |

## Technical data

| EPM-S202 / EPM-S206 (NPN): rated data |  |
| :---: | :---: |
| Module identifier | EPM-S202: $5_{\text {dec }}$; EPM-S206: $7_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | EPM-S202: 60 mA ; EPM-S206: 65 mA ; |
| Power loss | 0.9 W |
| Digital inputs |  |
| Number of inputs | 8 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Input voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V) |
| for signal "0" | EPM-S202: DC 0 ... 5 V ; EPM-S206: DC 15 ... 28.8 V |
| for signal "1" | EPM-S202: DC 15 ... 28.8 V; EPM-S206: DC 0 ... 5 V |
| Input current |  |
| for signal "1" | 3 mA |
| 2-wire BERO |  |
| Connection possible | Yes |
| Max. permissible closed-circuit current | 0.5 mA |
| Input delay |  |
| from "0" to "1" | 3 ms |
| from " 1 " to "0" | 3 ms |
| Number of inputs which can be used simultaneously |  |
| horizontal structure | 8 |
| vertical structure | 8 |
| Input characteristic | EPM-S202: IEC 61131, type 1; EPM-S206: none |
| Input data size | 8 bits |
| Status, alarm, diagnostics |  |
| Status display | Green LEDs per channel |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | None |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound modules - digital I/O
Four digital inputs (three-wire conductor connection) - EPM-S203

### 3.6.4 Four digital inputs (three-wire conductor connection) - EPM-S203

This module detects up to four binary control signals from the process level and transmits them to the higher-level bus system.

## Features

- 4 digital inputs in three-wire conductor technique
- Suitable for switches and proximity switches
- LEDs display the switching states of the digital inputs


## Overview



Fig. 3-18 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number
© ... (7) Bit number in bit presentation

## Status displays

| Module status LEDs ${ }_{\text {A }}$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| $10$ | 3 | DI1 | Green | On: Digital input triggered |
| $\square-$ | 4 | DI2 |  |  |
| O- | 5 | DI3 |  |  |
| - $\qquad$ | 6 | DI4 |  |  |
| - | 7 | - | - | Not assigned |
| $10-\square$ | 8 |  |  |  |
|  | 9 |  |  |  |
| SLu0001 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | Module reports error <br> Bus communication is OK |  |
| Off | On | Module reports error <br> Bus communication not possible |
| Off | Off | Error in the bus supply voltage |
| Blinking | Configuration error (■) 249) |  |

## Terminals

## Module terminals, spring terminals

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
| $1-\longdiv { \square _ { 5 } } - 5$ | 1 | Digital input DI1 | (AWG $0.08 .5 \mathrm{~mm}^{2}$ |
|  | 2 | 24 V DC |  |
|  | 3 | GND |  |
|  | 4 | Digital input DI3 |  |
|  | 5 | Digital input DI2 |  |
|  | 6 | 24 V DC |  |
| $4^{4} \bigcirc 0^{8-8}$ | 7 | GND |  |
| SL10002 | 8 | Digital input DI4 |  |

## 3 Product description

I/O compound modules - digital I/O
Four digital inputs (three-wire conductor connection) - EPM-S203

## Technical data

| EPM-S203: Rated data |  |
| :---: | :---: |
| Module identifier | 8 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 55 mA |
| Power loss | 0.6 W |
| Digital inputs |  |
| Number of inputs | 4 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Input voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V) |
| for signal "0" | DC 0 ... 5 V |
| for signal "1" | DC $15 . . .28 .8 \mathrm{~V}$ |
| Input current |  |
| for signal "1" | 3 mA |
| 2-wire BERO |  |
| Connection possible | Yes |
| Max. permissible closed-circuit current | 0.5 mA |
| Input delay |  |
| from "0" to "1" | 3 ms |
| from " 1 " to "0" | 3 ms |
| Number of inputs which can be used simultaneously |  |
| horizontal structure | 4 |
| vertical structure | 4 |
| Input characteristic | IEC 61131 |
| Input data size | 8 bits (with EPM-S110: 4 bits) |
| Status, alarm, diagnostics |  |
| Status display | Green LEDs per channel |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | None |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

### 3.6.5 $\quad 2$ digital inputs with time stamp function - EPM-S207

This module detects up to two binary control signals from the process level and transmits them to the higher-level bus system.
If the time stamp function is parameterised, in the case of a specified edge (rising or falling), the current time of the system-internal $\mu s$-ticker is stored as a time stamp entry in the process image along with the status of the inputs and a consecutive number.

Depending on the configuration, up to 15 time successive stamp entries can be recorded in the process image.

## Features

- 2 digital inputs
- up to 15 time stamp entries can be recorded
- Suitable for switches and proximity switches
- LEDs display the switching states of the digital inputs (even when electronic supply is switched off)


## Overview



Fig. 3-19 Elements and circuit diagram
(A) Displays for module status

B Terminals
1... 8 Connection number
(0) ... (7) Bit number in bit presentation

## 3 Product description

I/O compound modules - digital I/O
2 digital inputs with time stamp function - EPM-S207

## Status displays

## Module status LEDs A

| View | Pos. | Designation | Colour | Explanation |
| :---: | :---: | :---: | :---: | :---: |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| - | 3 | DI1 | Green | On: Digital input triggered |
| $\square$ | 4 | DI2 |  |  |
| $\left\lvert\, \begin{gathered} \square \\ \square \end{gathered}\right.$ | 5 | - | - | Not assigned |
| $\square$ | 6 |  |  |  |
| $\square$ | 7 |  |  |  |
| $10-$ | 8 |  |  |  |
|  | 9 |  |  |  |
| SLOOO1 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :---: |
| On | Off | Module status OK Bus communication is OK |
| On | On | Module reports error Bus communication is OK |
| Off | On | Module reports error Bus communication not possible |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error ( $\square$ 249) |

## Terminals

| Module terminals, spring terminals ${ }^{\text {B }}$ |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
| $\square_{5} 5$ | 1 | Digital input DI1 |  |
|  | 2 | 24 V DC |  |
| $\square_{2} \square_{6}$ | 3 | GND |  |
|  | 4 | Not assigned |  |
| ${ }^{3} 10^{1}$ | 5 | Digital input DI2 |  |
| Ш\# | 6 | 24 V DC |  |
| $7^{4} \bigcirc 0^{8}$ | 7 | GND |  |
| su0002 | 8 | Not assigned |  |

## Technical data

| EPM-S207: rated data |  |
| :---: | :---: |
| Module identifier | $3841_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 85 mA |
| Power loss | 0.9 W |
| Digital inputs |  |
| Number of inputs | 2 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumption from load voltage L+ | 10 mA (without load) |
| Input voltage |  |
| Nominal value | DC 20.4 ... 28.8 V |
| for signal "0" | DC $0 . . .5 \mathrm{~V}$ |
| for signal "1" | DC $15 . . .28 .8 \mathrm{~V}$ |
| Input current |  |
| for signal "1" | 3 mA |
| 2-wire BERO |  |
| Connection possible | Yes |
| Max. permissible closed-circuit current | 0.5 mA |
| Input delay |  |
| from "0" to "1" | Parameterisable |
| from " 1 " to "0" | Parameterisable |
| Number of inputs which can be used simultaneously |  |
| horizontal structure | 2 |
| vertical structure | 2 |
| Input characteristic | IEC 61131, type 1 |
| Input data size | 60 bytes |
| Status, alarm, diagnostics |  |
| Status display | Green LEDs per channel |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| zwischen Kanälen in Gruppen zu | 2 |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound modules - digital I/O
2 digital inputs with time stamp function - EPM-S207

## Functional principle taking the example of the system bus CAN

Long cycle times or fieldbus cycle times which fluctuate depending on bus load inevitably result in unacceptable inaccuracy when precise switching times are needed. The time stamp function can be used to calculate switching times for outputs accurate to $1 \mu \mathrm{~s}$.
An I/O compound module with time stamp function is fitted with an internal ticker. The tickers within a station are all synchronised via the backplane bus to ensure the same time base.

- The ticker has a resolution of $1 \mu \mathrm{~s}$. After power-on, it counts from 0 ... $65535 \mu \mathrm{~s}$ and then goes back to 0 .
- If using I/O compound modules with the time stamp function, when the signal undergoes a edge change, the ticker value is saved to the process image along with the channel status.
- Up to 15 DI switching orders (sub-index entries) can be transmitted with the time stamp function.


Fig. 3-20 Saving the time stamp entries
DI1 Digital input 1
The time stamp entry for the edge evaluation is generated with a rising edge.
DI2 Digital input 2
The time stamp entry for the edge evaluation is generated with a falling edge.
A Channel status
B Running number (RN)
Counts from 0 ... 127 and then starts with 0 again. The consecutive number reflects the time sequence of the edges.
(C) Ticker value

### 3.6.6 Two digital outputs 0.5 A - EPM-S300 and EPM-S303 (NPN)

This module detects up to two binary control signals from the higher-level bus system and transmits them to the process level.

## Features

- 2 digital outputs (EPM-S303 N-switching)
- LEDs display the switching states of the digital outputs


## Overview



Fig. 3-21 Elements and circuit diagram
(A) Displays for module status

B Terminals
1... 8 Connection number
(0) ... (7) Bit number in bit presentation

## 3 Product description

I/O compound modules - digital I/O
Two digital outputs 0.5 A - EPM-S300 and EPM-S303 (NPN)

Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
| $1-\square \square$ | 2 | MF | Red | On: Module error and error at overload, short circuit, overtemperature (see table below) |
| [ | 3 | DO1 |  |  |
|  | 4 | DO2 | Green | On: Digital output triggered |
| $0$ | 5 |  |  |  |
| $\square$ | 6 |  |  |  |
| $\left.\begin{array}{\|c} 0 \\ 0 \end{array} \right\rvert\,$ | 7 |  |  |  |
| $10-\square$ | 8 | - | - | Not assigned |
|  | 9 |  |  |  |
| SL10001 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error ( $\mathbb{C} \mathbf{2 4 9 )}$ |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
|  | 1 | Digital output DO1 | $\xrightarrow\left[\left(1 / / / \mathrm{mm}^{(\text {AWG } 28 \ldots 16)}\right]{\left.\frac{0.08 \ldots 1.5 \mathrm{~mm}^{2}}{(10 \mathrm{~mm}}\right)}\right.$ |
|  | 2 | 24 V DC |  |
| $\square \square_{6} \square^{\circ}$ | 3 | GND |  |
|  | 4 | Not assigned |  |
|  | 5 | Digital output DO2 |  |
| \#\#-8 | 6 | 24 V DC |  |
|  | 7 | GND |  |
| SL10002 | 8 | Not assigned |  |

## Technical data

| EPM-S300 / EPM-S303 (NPN): rated data |  |
| :---: | :---: |
| Module identifier | EPM-S300: $\mathbf{2 5 7}_{\text {dec }}$; EPM-S303: $259{ }_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | EPM-S300: 55 mA ; EPM-S303: 60 mA |
| Power loss | 0.4 W |
| Digital outputs |  |
| Number of outputs | 2 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Load voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V) |
| Current consumption from load voltage L+ | EPM-S300: 5 mA (without load); EPM-S303: 2.5 mA (without load) |
| Total current |  |
| for each group, horizontal structure, $40^{\circ} \mathrm{C}$ | 1 A |
| for each group, horizontal structure, $60^{\circ} \mathrm{C}$ | 1 A |
| for each group, vertical structure | 1 A |
| Output voltage |  |
| for "1" signal, max. current | Only for EPM-S303: M (+250 mV) |
| for "1" signal, min. current | Only for EPM-S303: $M$ (+0 V) |
| Output current |  |
| for "1" signal, nominal value | 0.5 A |
| Output delay |  |
| from "0" to "1" | $30 \mu \mathrm{~s}$ |
| from " 1 " to "0" | EPM-S300: $175 \mu \mathrm{~s}$; EPM-S303: $100 \mu \mathrm{~s}$ |
| Lamp load | 10 W |
| Parallel switching of outputs |  |
| for redundant control | Not possible |
| for power increase | Not possible |
| Control of a digital input | Possible |
| Switching frequencies |  |
| for ohmic loads | Max. 1000 Hz |
| for inductive loads | Max. 0.5 Hz |
| for lamp loads | Max. 10 Hz |
| Limitation (internal) of the inductive breaking voltage | EPM-S300: L+ (-52 V); EPM-S303: +45 V |
| Short circuit protection of the output | Electronically |
| Operating threshold of the protection system | EPM-S300: 1 A; EPM-S303: 1.7 A |
| Output data size | 8 bits (with EPM-S110: 2 bits) |

## 3 Product description

I/O compound modules - digital I/O
Two digital outputs 0.5 A - EPM-S300 and EPM-S303 (NPN)

EPM-S300 / EPM-S303 (NPN): rated data
Status, alarm, diagnostics

| Status display | Green LEDs per channel |
| :--- | :--- |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | None |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation <br> Between the channels and the backplane <br> bus | Yes |
| Insulation checked with | DC 500 V |

### 3.6.7 Four digital outputs 0.5 A - EPM-S301 and EPM-S304 (NPN)

This module detects up to four binary control signals from the higher-level bus system and transmits them to the process level.

## Features

- 4 digital outputs (EPM-S304 N-switching)
- LEDs display the switching states of the digital outputs


## Overview



Fig. 3-22 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number
(0) ... (7) Bit number in bit presentation

## 3 Product description

I/O compound modules - digital I/O
Four digital outputs 0.5 A - EPM-S301 and EPM-S304 (NPN)

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
| 1 - | 2 | MF | Red | On: Module error and error at overload, short circuit, overtemperature (see table below) |
|  | 3 | DO1 | Green | On: Digital output triggered |
|  | 4 | DO2 |  |  |
| $0$ | 5 | DO3 |  |  |
| 0 | 6 | DO4 |  |  |
|  | 7 | - | - | Not assigned |
| $10-\square$ | 8 |  |  |  |
|  | 9 |  |  |  |
| SLO001 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error ( $\mathbb{C}$ 249) |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
| SLIOOO2 | 1 | Digital output DO1 |  |
|  | 2 | EPM-S301: GND EPM-S304: 24 V DC |  |
|  | 3 | Digital output DO3 |  |
|  | 4 | EPM-S301: GND EPM-S304: 24 V DC |  |
|  | 5 | Digital output DO2 |  |
|  | 6 | EPM-S301: GND EPM-S304: 24 V DC |  |
|  | 7 | Digital output DO4 |  |
|  | 8 | EPM-S301: GND EPM-S304: 24 V DC |  |

## Technical data

| EPM-S301 / EPM-S304 (NPN): rated data |  |
| :---: | :---: |
| Module identifier | EPM-S301: $260_{\text {dec }}$; EPM-S304: $261_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | EPM-S301: 55 mA ; EPM-S304: 65 mA |
| Power loss | 0.5 W |
| Digital outputs |  |
| Number of outputs | 4 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Load voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V) |
| Current consumption from load voltage L+ | EPM-S301: 10 mA (without load); EPM-S304: 5 mA (without load) |
| Total current |  |
| for each group, horizontal structure, $40^{\circ} \mathrm{C}$ | 2 A |
| for each group, horizontal structure, $60^{\circ} \mathrm{C}$ | 2 A |
| for each group, vertical structure | 2 A |
| Output voltage |  |
| for "1" signal, max. current | Only for EPM-S304: M (+250 mV) |
| for "1" signal, min. current | Only for EPM-S304: M (+0 V) |
| Output current |  |
| for "1" signal, nominal value | 0.5 A |
| Output delay |  |
| from "0" to "1" | $30 \mu \mathrm{~s}$ |
| from " 1 " to "0" | EPM-S301: $175 \mu s ;$ EPM-S304: $100 \mu s$ |
| Lamp load | 10 W |
| Parallel switching of outputs |  |
| for redundant control | Not possible |
| for power increase | Not possible |
| Control of a digital input | Possible |
| Switching frequencies |  |
| for ohmic loads | Max. 1000 Hz |
| for inductive loads | Max. 0.5 Hz |
| for lamp loads | Max. 10 Hz |
| Limitation (internal) of the inductive breaking voltage | EPM-S301: L+ (-52 V); EPM-S304: +45 V |
| Short circuit protection of the output | Electronically |
| Operating threshold of the protection system | EPM-S301: 1 A; EPM-S304: 1.7 A |
| Output data size | 8 bits (with EPM-S110: 4 bits) |

## 3 Product description

I/O compound modules - digital I/O
Four digital outputs 0.5 A - EPM-S301 and EPM-S304 (NPN)

EPM-S301 / EPM-S304 (NPN): rated data
Status, alarm, diagnostics

| Status display | Green LEDs per channel |
| :--- | :--- |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | None |
| Module status | Green LED |
| Module error display | Red SF-LED |
| Channel error display | None |
| Electrical isolation <br> Between the channels and the backplane <br> bus | Yes |
| Insulation checked with | DC 500 V |

### 3.6.8 Eight digital outputs 0.5 A - EPM-S302 and EPM-S305 (NPN)

This module detects up to eight binary control signals from the higher-level bus system and transmits them to the process level.

## Features

- 8 digital outputs (EPM-S305 N-switching)
- LEDs show the switching states of the digital outputs


## Overview



Fig. 3-23
Elements and circuit diagram
A Displays for module status
B Terminals
1 ... 8 Connection number
(0) ... (7) Bit number in bit presentation

## 3 Product description

I/O compound modules - digital I/O
Eight digital outputs 0.5 A - EPM-S302 and EPM-S305 (NPN)

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
| 1 - | 2 | MF | Red | On: Module error and error at overload, short circuit, overtemperature (see table below) |
| $\square-$ | 3 | DO1 | Green | On: Digital output triggered |
|  | 4 | DO2 |  |  |
| $\square$ | 5 | DO3 |  |  |
|  $\qquad$ | 6 | DO4 |  |  |
| $\left.\begin{array}{\|l\|l} 0 \\ 0 \end{array} \right\rvert\,$ | 7 | DO5 |  |  |
| $10-\square$ | 8 | DO6 |  |  |
|  | 9 | DO7 |  |  |
| SL10001 | 10 | DO8 |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error ( $\mathbb{C}$ 249) |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
|  | 1 | Digital output DO1 |  |
|  | 2 | Digital output DO3 |  |
|  | 3 | Digital output DO5 |  |
|  | 4 | Digital output DO7 |  |
|  | 5 | Digital output DO2 |  |
|  | 6 | Digital output DO4 |  |
|  | 7 | Digital output DO6 |  |
|  | 8 | Digital output DO8 |  |

## Technical data

| EPM-S302 / EPM-S305 (NPN): rated data |  |
| :---: | :---: |
| Module identifier | EPM-S302: $\mathbf{2 6 2}_{\text {dec }}$; EPM-S305: $263_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | EPM-S302: 65 mA ; EPM-S305: 70 mA |
| Power loss | EPM-S302: 0.7 W; EPM-S305: 0.6 W |
| Digital outputs |  |
| Number of outputs | 8 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Load voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V) |
| Current consumption from load voltage L+ | EPM-S302: 15 mA (without load); EPM-S305: 10 mA (without load) |
| Total current |  |
| for each group, horizontal structure, $40^{\circ} \mathrm{C}$ | EPM-S302: 4 A; EPM-S305: 2.5 A |
| for each group, horizontal structure, $60^{\circ} \mathrm{C}$ | EPM-S302: 4 A; EPM-S305: 2.5 A |
| for each group, vertical structure | EPM-S302: 4 A; EPM-S305: 2.5 A |
| Output voltage |  |
| for "1" signal, max. current | Only for EPM-S305: $M(+250 \mathrm{mV})$ |
| for "1" signal, min. current | Only for EPM-S305: M (+0 V) |
| Output current |  |
| for "1" signal, nominal value | 0.5 A |
| Output delay |  |
| from "0" to " 1 " | $30 \mu \mathrm{~s}$ |
| from " 1 " to "0" | EPM-S302: $175 \mu \mathrm{~s}$; EPM-S305: $100 \mu \mathrm{~s}$ |
| Lamp load | 10 W |
| Parallel switching of outputs |  |
| for redundant control | Not possible |
| for power increase | Not possible |
| Control of a digital input | Possible |
| Switching frequencies |  |
| for ohmic loads | Max. 1000 Hz |
| for inductive loads | Max. 0.5 Hz |
| for lamp loads | Max. 10 Hz |
| Limitation (internal) of the inductive breaking voltage | EPM-S302: L+ (-52 V); EPM-S305: +45 V |
| Short circuit protection of the output | Electronically |
| Operating threshold of the protection system | EPM-S302: 1 A; EPM-S305: 1.7 A |
| Output data size | 8 bits |

## 3 Product description

I/O compound modules - digital I/O
Eight digital outputs 0.5 A - EPM-S302 and EPM-S305 (NPN)

| EPM-S302 / EPM-S305 (NPN): rated data |
| :--- |
| Status, alarm, diagnostics <br> Status display Green LEDs per channel <br> Alarms No <br> Process alarm No <br> Diagnostic alarm No <br> Diagnostic function No <br> Diagnostic information can be read out None <br> Module status Green LED <br> Module error display Red LED <br> Channel error display None <br> Electrical isolation  <br> Between the channels and the backplane Yes <br> bus DC 500 V <br> Insulation checked with  |

### 3.6.9 Two digital outputs 2 A-EPM-S306

This module detects up to two binary control signals from the higher-level bus system and transmits them to the process level.

## Features

- 2 digital outputs which each can be loaded with 2 A
- LEDs show the switching states of the digital outputs
- Overload protection (outputs switch off at overload)


## Overview



Fig. 3-24 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number
(0) ... (7) Bit number in bit presentation

## 3 Product description

I/O compound modules - digital I/O
Two digital outputs 2 A - EPM-S306

Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
| $1-\square \square$ | 2 | MF | Red | On: Module error and error at overload, short circuit, overtemperature (see table below) |
| [ | 3 | DO1 |  |  |
|  | 4 | DO2 | Green | On: Digital output triggered |
| $0$ | 5 |  |  |  |
| $\square$ | 6 |  |  |  |
| $\left.\begin{array}{\|c} 0 \\ 0 \end{array} \right\rvert\,$ | 7 |  |  |  |
| $10-\square$ | 8 | - | - | Not assigned |
|  | 9 |  |  |  |
| SL10001 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (Cl 249) |

## Terminals

Module terminals, spring terminals

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Digital output DO1 | $\underset{\substack{10 \mathrm{~mm}}}{\substack{0.08 \ldots 1.5 \mathrm{~mm}^{2} \\(\text { AWG } 28 \ldots 16)}}$ |
|  | 2 | 24 V DC |  |
|  | 3 | GND |  |
|  | 4 | Not assigned |  |
|  | 5 | Digital output DO2 |  |
|  | 6 | 24 V DC |  |
|  | 7 | GND |  |
| SLu0002 | 8 | Not assigned |  |

## Technical data

| EPM-S306: rated data |  |  |  |
| :--- | :--- | :---: | :---: |
| Module identifier | $258_{\text {dec }}$ |  |  |
| Current consumption/power loss |  |  |  |
| Current consumption from backplane bus |  |  | 60 mA |
| Power loss | 0.55 W |  |  |


| Digital outputs |  |
| :---: | :---: |
| Number of outputs | 2 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Load voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V ) |
| Current consumption from load voltage L+ | 10 mA (without load) |
| Total current |  |
| for each group, horizontal structure, $40^{\circ} \mathrm{C}$ | 4 A |
| for each group, horizontal structure, $60^{\circ} \mathrm{C}$ | 4 A |
| for each group, vertical structure | 4 A |
| Output current |  |
| for "1" signal, nominal value | 2 A |
| Output delay |  |
| from "0" to " 1 " | $100 \mu \mathrm{~s}$ |
| from " 1 " to "0" | 250 us |
| Lamp load | 10 W |
| Parallel switching of outputs |  |
| for redundant control | Not possible |
| for power increase | Not possible |
| Control of a digital input | Possible |
| Switching frequencies |  |
| for ohmic loads | Max. 1000 Hz |
| for inductive loads | Max. 0.5 Hz |
| for lamp loads | Max. 10 Hz |
| Limitation (internal) of the inductive breaking voltage | L+ (-52 V) |
| Short circuit protection of the output | Electronically |
| Operating threshold of the protection system | 2.7 A |
| Output data size | 8 bits (with EPM-S110: 2 bits) |
| Status, alarm, diagnostics |  |
| Status display | Green LEDs per channel |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | None |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound modules - digital I/O
Four digital outputs 2 A - EPM-S309
3.6.10

## Four digital outputs 2 A - EPM-S309

This module detects up to four binary control signals from the higher-level bus system and transmits them to the process level.

## Features

- 4 digital outputs, each can be loaded with 2 A (permanent total current max. 4 A )
- LEDs show the switching states of the digital outputs
- Overload protection (outputs switch off at overload)


## Overview

EPM-S309

Fig. 3-25 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number
(0) ... (7) Bit number in bit presentation

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
| 1 | 2 | MF | Red | On: Module error and error at overload, short circuit, overtemperature (see table below) |
| $\square-$ | 3 | DO1 | Green | On: Digital output triggered |
| $\left.\begin{array}{ll} 0 \\ 0 \end{array}\right]$ | 4 | DO2 |  |  |
|  | 5 | DO3 |  |  |
| $\left[\begin{array}{l} u \\ 0 \end{array}\right.$ | 6 | DO4 |  |  |
|  | 7 | - | - | Not assigned |
| $10-\square$ | 8 |  |  |  |
|  | 9 |  |  |  |
| SL0001 | 10 |  |  |  |


| Messages of the status LEDs RUN and MF |  |  |
| :---: | :---: | :--- |
| RUN | MF | Meaning |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | On | Module reports error <br> Bus communication not possible |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (C) 249) |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
|  | 1 | Digital output DO1 | $\xrightarrow\left[\left(1 / / / \mathrm{mm}^{(\text {AWG } 28 \ldots 16)}\right]{\left.\frac{0.08 \ldots 1.5 \mathrm{~mm}^{2}}{(10 \mathrm{~mm}}\right)}\right.$ |
| - | 2 | GND |  |
| $\square \square_{6}^{6}$ | 3 | Digital output DO3 |  |
|  | 4 | GND |  |
| $3^{\square} \square^{7}$ | 5 | Digital output DO2 |  |
| $\square_{4} \square_{8}-8$ | 6 | GND |  |
|  | 7 | Digital output DO4 |  |
| SLO002 | 8 | GND |  |

## Technical data

| EPM-S309: rated data | $264_{\mathrm{dec}}$ |
| :--- | :--- |
| Module identifier |  |
| Current consumption/power loss | 65 mA |
| Current consumption from backplane bus | 0.8 W |
| Power loss |  |

## 3 Product description

I/O compound modules - digital I/O
Four digital outputs 2 A - EPM-S309

| EPM-S309: rated data |  |
| :---: | :---: |
| Digital outputs |  |
| Number of outputs | 4 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Load voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V ) |
| Current consumption from load voltage L+ | 20 mA (without load) |
| Total current |  |
| for each group, horizontal structure, $40^{\circ} \mathrm{C}$ | 4 A |
| for each group, horizontal structure, $60^{\circ} \mathrm{C}$ | 4 A |
| for each group, vertical structure | 4 A |
| Output current |  |
| for "1" signal, nominal value | 2 A |
| Output delay |  |
| from "0" to "1" | $100 \mu \mathrm{~s}$ |
| from " 1 " to "0" | 250 \% |
| Lamp load | 10 W |
| Parallel switching of outputs |  |
| for redundant control | Not possible |
| for power increase | Not possible |
| Control of a digital input | Not possible |
| Switching frequencies |  |
| for ohmic loads | Max. 1000 Hz |
| for inductive loads | Max. 0.5 Hz |
| for lamp loads | Max. 10 Hz |
| Limitation (internal) of the inductive breaking voltage | $\mathrm{L}+(-52 \mathrm{~V})$ |
| Short circuit protection of the output | Electronically |
| Operating threshold of the protection system | 2.7 A |
| Output data size | 8 bits (with EPM-S110: 4 bits) |
| Status, alarm, diagnostics |  |
| Status display | Green LEDs per channel |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | None |
| Module status | Green LED |
| Module error display | Red SF-LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

### 3.6.11 2 digital outputs with time stamp function - EPM-S310

This module detects up to two binary control signals from the higher-level bus system and transmits them to the process level with time control using the time stamp function.
If the time stamp function has been parameterised, 15 initial states can be transferred to the FIFO memory as time stamp entry together with a time value of the $\mu \mathrm{s}$ ticker and a consecutive number. The FIFO memory offers space for max. 31 time stamp entries.

## Features

- Two digital outputs
- FIFO memory for up to 15 time stamp entries
- Activation via process image
- LEDs show the switching states of the digital outputs


## Overview



Fig. 3-26 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number
(0) ... (7) Bit number in bit presentation

## 3 Product description

I/O compound modules - digital I/O
2 digital outputs with time stamp function - EPM-S310

Status displays


Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (Cl 249) |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
|  | 1 | Digital output DO1 |  |
|  | 2 | 24 V DC |  |
|  | 3 | GND |  |
|  | 4 | Not assigned |  |
|  | 5 | Digital output DO2 |  |
|  | 6 | 24 V DC |  |
|  | 7 | GND |  |
|  | 8 | Not assigned |  |

## Technical data

| EPM-S310: rated data |  |
| :---: | :---: |
| Module identifier | $3905_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 85 mA |
| Power loss | 0.95 W |
| Digital outputs |  |
| Number of outputs | 2 |
| Cable length |  |
| shielded | 1000 m |
| unshielded | 600 m |
| Load voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V) |
| Current consumption from load voltage L+ | 15 mA (without load) |
| Total current |  |
| for each group, horizontal structure, $40^{\circ} \mathrm{C}$ | 1 A |
| for each group, horizontal structure, $60^{\circ} \mathrm{C}$ | 1 A |
| for each group, vertical structure | 1 A |
| Output current |  |
| for "1" signal, nominal value | 0.5 A |
| Output delay |  |
| from "0" to " 1 " | Max. 100 ns |
| from "1" to "0" | Max. 100 ns |
| Lamp load | 10 W |
| Parallel switching of outputs |  |
| for redundant control | Not possible |
| for power increase | Not possible |
| Control of a digital input | Possible |
| Switching frequencies |  |
| for ohmic loads | Max. 40 kHz |
| for inductive loads | Max. 40 kHz |
| for lamp loads | Max. 40 kHz |
| Limitation (internal) of the inductive breaking voltage | L+ (-52 V) |
| Short circuit protection of the output | Electronically; only highside |
| Operating threshold of the protection system | 2.5 A |
| Output data size | 60 bits |

## 3 Product description

I/O compound modules - digital I/O
2 digital outputs with time stamp function - EPM-S310

| Status, alarm, diagnostics |  |
| :---: | :---: |
| Status display | Green LEDs per channel |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels in groups of | 2 |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## Functional principle taking the example of the system bus CAN

Long cycle times or fieldbus cycle times which fluctuate depending on bus load inevitably result in unacceptable inaccuracy when precise switching times are needed. The time stamp function can be used to calculate switching times for outputs accurate to $1 \mu \mathrm{~s}$.
An I/O compound module with time stamp function is fitted with an internal ticker. The tickers within a station are all synchronised via the backplane bus to ensure the same time base.

- The ticker has a resolution of $1 \mu$ s. After power-on, it counts from 0 ... $65535 \mu$ s and then goes back to 0 .
- If using I/O compound modules with the time stamp function, when the signal undergoes a edge change, the ticker value is saved to the process image along with the channel status.
- Up to 15 DO switching orders can be received with the time stamp function which allows an output to be activated several times within a cycle.


Fig. 3-27 Output of time stamp entries
DO1 Digital output 1
The output signal is switched according to the time stamp entry.
DO2 Digital output 2
The output signal is switched according to the time stamp entry.
A Status of digital outputs
B Running number (RN)
Counts from 0 ... 127 and then starts with 0 again. The consecutive number is used to determine the time sequence of time stamp entries. The consecutive number is incremented with each time stamp entry.
C Ticker value

## 3 Product description

I/O compound modules - digital I/O
Two relay outputs - EPM-S308

### 3.6.12 Two relay outputs - EPM-S308

This module detects up to two binary control signals from the higher-level bus system and transmits them to the process level via relay outputs (switches).

## Features

- 2 relay outputs (switches), potential-free
- LEDs show the switching states of the outputs


## Overview

EPM-S308

Fig. 3-28 Elements and circuit diagram
(A) Displays for module status

B Terminals
1... 8 Connection number
(0) ... (7) Bit number in bit presentation

## Status displays

| Module status LEDs $\mathbf{A}$ |
| :--- |
| View |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■) 249) |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
|  | 1 | Relay output DO1 | $\square 0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
|  | 2 | Not assigned |  |
|  | 3 | Relay output DO2 |  |
|  | 4 | Not assigned |  |
|  | 5 | Relay output DO1 | $\xrightarrow[\substack{ \\10 \mathrm{~mm}}]{\substack{ \\(\text { AWG } 28 \ldots 16)}}$ |
|  | 6 | Not assigned |  |
|  | 7 | Relay output DO2 |  |
| SLu0002 | 8 | Not assigned |  |

## 3 Product description

I/O compound modules - digital I/O
Two relay outputs - EPM-S308

## Technical data

| EPM-S308: rated data |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Module identifier | $265{ }_{\text {dec }}$ |  |  |  |
| Current consumption/power loss |  |  |  |  |
| Current consumption from backplane bus | 130 mA |  |  |  |
| Power loss | 0.7 W |  |  |  |
| Relay outputs |  |  |  |  |
| Number of outputs | 2 |  |  |  |
| Load voltage |  |  |  |  |
| Nominal value | DC $30 \mathrm{~V} / \mathrm{AC} 230 \mathrm{~V}$ |  |  |  |
| Total current |  |  |  |  |
| for each group, horizontal structure, $40^{\circ} \mathrm{C}$ | 3 A |  |  |  |
| for each group, horizontal structure, $60^{\circ} \mathrm{C}$ | 3 A |  |  |  |
| for each group, vertical structure | 3 A |  |  |  |
| Output current |  |  |  |  |
| for "1" signal, nominal value | 3 A |  |  |  |
| Maximum switching capacity |  |  | V |  <br> 350 |
| Service life |  |  | N |  |
| Parallel switching of outputs |  |  |  |  |
| for redundant control | Not possible |  |  |  |
| for power increase | Not possible |  |  |  |
| Control of a digital input | Not possible |  |  |  |
| Switching frequencies |  |  |  |  |
| for ohmic loads | Max. 100 Hz |  |  |  |
| Output data size | 8 bits (with EPM-S110: 2 bits) |  |  |  |


| EPM-S308: rated data |  |
| :--- | :--- |
| Status, alarm, diagnostics | Green LEDs per channel |
| Status display | No |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | None |
| Diagnostic information can be read out | Green LED |
| Module status | Red LED |
| Module error display | None |
| Channel error display | Yes |
| Electrical isolation | Yes |
| between the channels | DC 500 V |
| Between the channels and the backplane |  |
| bus |  |
| Insulation checked with |  |

## 3 Product description

I/O compound modules - analog I/O
Representation of analog values
3.7 I/O compound modules - analog I/O

### 3.7.1 Representation of analog values

Analog values can only be processed in a binary form. For this, the analog module converts each process signal into a digital form and passes it on as a word.

| Resolution | Analog value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | HIGH byte (byte 0) |  |  |  |  |  |  |  | LOW byte (byte 1) |  |  |  |  |  |  |  |
| Bit number | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Valency | Sign bit | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^{9}$ | $2^{8}$ | $2^{7}$ | $2^{6}$ | $2^{5}$ | $2^{4}$ | $2^{3}$ | $2^{2}$ | $2^{1}$ | $2^{0}$ |
| 12 bits + sign bit | Sign bit | Measured value |  |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 0 |
| 15 bits + sign bit | Sign bit | Measured value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Resolution: In the case of a resolution of 12 bits plus sign bit, the lower-order digits that are not used ( 3 bits) are written to with " 0 ".
Sign bit: Bit $15=" 0 " \rightarrow$ positive value; bit $15=" 1 " \rightarrow$ negative value.
Response in the event of an error: If a measured value exceeds the overrange or falls below the lower range, the following value is output:
Measured value > overrange $\rightarrow 32767$ ( $7 \mathrm{FFF}_{\mathrm{h}}$ )
Measured value < lower range $\rightarrow-32768\left(8000_{h}\right)$
If a parameterisation error occurs, the measured value $32767\left(7 \mathrm{FFF}_{\mathrm{h}}\right)$ is output.

### 3.7.2 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400

This module detects up to two analog control signals from the process level and transmits them to the higher-level bus system.

## Features

- 2 analog inputs
- Voltage range 0 ... 10 V
- 12-bit resolution
- Signal function is parameterisable
- An LED indicates if an input signal is outside the permissible measuring range


## STop Stop!

Overvoltage at the inputs
The electronics of the electronic module are not protected against too high input signals.
Possible consequences:

- The module is destroyed

Protective measures:

- Make sure that the signals and encoders connected match the measuring range parameterised.


## Overview



Fig. 3-29 Elements and circuit diagram

| A | Displays for module status |
| :--- | :--- |
| B | Terminals |
| $1 \ldots 8$ | Connection number |

## 3 Product description

I/O compound modules - analog I/O
2 analog inputs 0 ... 10 V (12 bits) - EPM-S400

## Status displays

## Module status LEDs A

| View | Pos. | Designation | Colour | Explanation |
| :---: | :---: | :---: | :---: | :---: |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| $1-$ | 3 | Al1 | Red | On: Channel 1, signal outside the measuring range, error in parameter setting |
|  | 4 | Al2 | Red | On: Channel 2 , signal outside the measuring range, error in parameter setting |
|  | 5 | - | - | Not assigned |
|  | 6 |  |  |  |
|  | 7 |  |  |  |
|  | 8 |  |  |  |
|  | 9 |  |  |  |
|  | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■) 249) |

## Terminals

Module terminals, spring terminals ©

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
| $1-\square_{5}^{\square}-5$ | 1 | Analog input AI1 (+) |  |
|  | 2 | Analog input AI1 (GND) |  |
|  | 3 | Not assigned |  |
|  | 4 | Not assigned |  |
|  | 5 | Analog input Al2 (+) |  |
|  | 6 | Analog input AI2 (GND) |  |
| ${ }^{\circ}$ | 7 | Not assigned |  |
| stu0002 | 8 | Not assigned |  |

## 1 Note!

- Use parameter setting to deactivate unused inputs.
- The module does not provide any auxiliary supply for sensors.

Technical data

| EPM-S400: Rated data |  |
| :---: | :---: |
| Module identifier | $1025_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 70 mA |
| Power loss | 0.7 W |
| Analog inputs |  |
| Number of inputs | 2 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumpt. from load voltage L+ | 15 mA (without load) |
| Voltage inputs |  |
| Input voltage ranges | 0 V ... +10 V |
| Destruction limit (input voltage) | 30 V |
| Min. input resistance | $100 \mathrm{k} \Omega$ |
| Operational error limit | +/-0.3 \% |
| Basic error limit | +/- 0.2 \% |
| Measuring principle | Gradual approximation |
| Resolution | 12 bits |
| Basic conversion time | 2 ms all channels |
| Interference voltage suppression for a frequency of | > 50 dB at 50 Hz (UCM < 2 V ) |
| Temperature error (relating to input range) | $\pm 0.005 \% / \mathrm{K}$ |
| Linearity distortion (relating to input range) | $\pm 0.02$ \% |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the input range) | $\pm 0.05$ \% |
| Input data size | 4 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Between the channels and the voltage supply | Yes |
| Max. potential difference between inputs ( $\mathrm{U}_{\mathrm{cm}}$ ) | DC 2 V |
| Max. potential difference between inputs and $M_{\text {internal }}\left(\mathrm{U}_{\text {iso }}\right)$ | DC $75 \mathrm{~V} / \mathrm{AC} 60 \mathrm{~V}$ |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound modules - analog I/O
4 analog inputs $0 \ldots 10 \mathrm{~V}$ ( 12 bits) - EPM-S401
3.7.3 4 analog inputs 0 ... 10 V (12 bits) - EPM-S401

This module detects up to four analog control signals from the process level and transmits them to the higher-level bus system.

## Features

- 4 analog inputs
- Voltage range 0 ... 10 V
- 12-bit resolution
- Signal function is parameterisable
- An LED indicates if an input signal is outside the permissible measuring range


## STop Stop!

## Overvoltage at the inputs

The electronics of the electronic module are not protected against too high input signals.
Possible consequences:

- The module is destroyed

Protective measures:

- Make sure that the signals and encoders connected match the measuring range parameterised.


## Overview



Fig. 3-30 Elements and circuit diagram

[^2]
## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| 1 | 3 | Al1 | Red | On: Channel 1, signal outside the measuring range, error in parameter setting |
| $\square$ | 4 | Al2 | Red | On: Channel 2 , signal outside the measuring range, error in parameter setting |
| $10-$ | 5 | Al3 | Red | On: Channel 3, signal outside the measuring range, error in parameter setting |
|  | 6 | Al4 | Red | On: Channel 4, signal outside the measuring range, error in parameter setting |
|  | 7 |  |  |  |
|  | 8 | - | - | Not assigned |
|  | 9 | - | - | Not assigned |
|  | 10 |  |  |  |


| Messages of the status LEDs RUN and MF |  |  |
| :---: | :---: | :--- |
| RUN | MF | Meaning |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | On | Module reports error <br> Bus communication not possible |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (Cl 249) |

## Terminals

Module terminals, spring terminals ${ }^{\text {B }}$

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Analog input AI1 (+) | $(1 / \mathrm{ll}$ |
|  | 2 | Analog input AI1 (GND) |  |
|  | 3 | Analog input AI3 (+) |  |
|  | 4 | Analog input AI3 (GND) |  |
|  | 5 | Analog input Al2 (+) |  |
|  | 6 | Analog input Al2 (GND) |  |
|  | 7 | Analog input Al4 (+) |  |
| SL10002 | 8 | Analog input Al4 (GND) |  |

## 1 Note!

- Use parameter setting to deactivate unused inputs.
- The module does not provide any auxiliary supply for sensors.


## 3 Product description

I/O compound modules - analog I/O
4 analog inputs 0 ... 10 V (12 bits) - EPM-S401

Technical data

| EPM-S401: Rated data |  |
| :---: | :---: |
| Module identifier | 1028 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 70 mA |
| Power loss | 0.7 W |
| Analog inputs |  |
| Number of inputs | 4 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumpt. from load voltage L+ | 15 mA (without load) |
| Voltage inputs |  |
| Input voltage ranges | $0 \mathrm{~V} . . .+10 \mathrm{~V}$ |
| Destruction limit (input voltage) | 30 V |
| Min. input resistance | $100 \mathrm{k} \Omega$ |
| Operational error limit | +/-0.3 \% |
| Basic error limit | +/-0.2 \% |
| Measuring principle | Gradual approximation |
| Resolution | 12 bits |
| Basic conversion time | 4 ms all channels |
| Interference voltage suppression for a frequency of | $>50 \mathrm{~dB}$ at $50 \mathrm{~Hz}($ UCM < 2 V ) |
| Temperature error (relating to input range) | + 70.005 \%/K |
| Linearity distortion (relating to input range) | $\pm \square 0.02$ \% |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the input range) | $\pm 70.05$ \% |
| Input data size | 8 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Between the channels and the voltage supply | Yes |
| Max. potential difference between inputs $\left(U_{\mathrm{cm}}\right)$ | DC 2 V |
| Max. potential difference between inputs and $M_{\text {internal }}\left(U_{\text {iso }}\right)$ | DC $75 \mathrm{~V} / \mathrm{AC} 60 \mathrm{~V}$ |
| Insulation checked with | DC 500 V |

### 3.7.4 2 analog inputs $0 / 4$... 20 mA (12 bits) - EPM-S402

This module detects up to four analog control signals from the process level and transmits them to the higher-level bus system.

## Features

- 2 analog inputs
- Current range 0/4 ... 20 mA
- 12-bit resolution
- Signal function is parameterisable
- An LED indicates if an input signal is outside the permissible measuring range


## STop Stop!

Overvoltage at the inputs
The electronics of the electronic module are not protected against too high input signals.
Possible consequences:

- The module is destroyed

Protective measures:

- Make sure that the signals and encoders connected match the measuring range parameterised.


## Overview

(A) EPM-S402

Fig. 3-31 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number

## 3 Product description

I/O compound modules - analog I/O
2 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S402

## Status displays

## Module status LEDs A



Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■) 249) |

## Terminals

Module terminals, spring terminals B

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Analog input AI1 (+) |  |
|  | 2 | Analog input AI1 (GND) |  |
|  | 3 | Not assigned |  |
|  | 4 |  |  |
|  | 5 | Analog input Al2 (+) |  |
|  | 6 | Analog input AI2 (GND) |  |
|  | 7 |  |  |
| SL10002 | 8 | Not assigned |  |

## 1 Note!

- Use parameter setting to deactivate unused inputs.
- The module does not provide any auxiliary supply for sensors.

Technical data

| EPM-S402: Rated data |  |
| :---: | :---: |
| Module identifier | 1026 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 70 mA |
| Power loss | 0.7 W |
| Analog inputs |  |
| Number of inputs | 2 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumpt. from load voltage L+ | 15 mA (without load) |
| Current inputs |  |
| Input current ranges | $\begin{aligned} & 0 \mathrm{~mA} \ldots+20 \mathrm{~mA} \\ & +4 \mathrm{~mA} \ldots+20 \mathrm{~mA} \end{aligned}$ |
| Destruction limit (input current) | 40 mA |
| Max. input resistance | $110 \Omega$ |
| Operational error limit | +/- 0.3 \% ... +/- 0.5 \% |
| Basic error limit | +/- 0.2 \% ... +/- 0.3 \% |
| Measuring principle | Gradual approximation |
| Resolution | 12 bits |
| Basic conversion time | 2 ms all channels |
| Interference voltage suppression for a frequency of | > 50 dB at 50 Hz (UCM < 2 V ) |
| Temperature error (relating to input range) | $\pm \square 0.005$ \%/K |
| Linearity distortion (relating to input range) | $\pm \square 0.02$ \% |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the input range) | $\pm \square 0.05$ \% |
| Input data size | 4 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |

## 3 Product description

I/O compound modules - analog I/O
2 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S402
EPM-S402: Rated data

| Electrical isolation |
| :--- |
| Between the channels and the backplane <br> bus |
| Between the channels and the voltage <br> supply |
| Max. potential difference between inputs <br> $\left(U_{\mathrm{cm}}\right)$ |
| Max. potential difference between inputs <br> and $M_{\text {internal }}\left(\mathrm{U}_{\text {iso }}\right)$ |
| DC 2 V |
| Insulation checked with |

### 3.7.5 4 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S403

This module detects up to four analog control signals from the process level and transmits them to the higher-level bus system.

## Features

- 4 analog inputs
- Current range 0/4 ... 20 mA
- 12-bit resolution
- Signal function is parameterisable
- An LED indicates if an input signal is outside the permissible measuring range


## stop Stop!

Overvoltage at the inputs
The electronics of the electronic module are not protected against too high input signals.
Possible consequences:

- The module is destroyed

Protective measures:

- Make sure that the signals and encoders connected match the measuring range parameterised.


## Overview



Fig. 3-32 Elements and circuit diagram

| A | Displays for module status |
| :--- | :--- |
| B | Terminals |
| $1 \ldots 8$ | Connection number |

## 3 Product description

I/O compound modules - analog I/O
4 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S403

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
|  | 3 | Al1 | Red | On: Channel 1, signal outside the measuring range, error in parameter setting |
|  | 4 | Al2 | Red | On: Channel 2, signal outside the measuring range, error in parameter setting |
|  | 5 | Al3 | Red | On: Channel 3 , signal outside the measuring range, error in parameter setting |
|  | 6 | Al4 | Red | On: Channel 4, signal outside the measuring range, error in parameter setting |
|  | 7 |  |  |  |
| Su0001 | 8 |  | - |  |
|  | 9 | - | - | Not assigned |
|  | 10 |  |  |  |


| Messages of the status LEDs RUN and MF |  |  |
| :---: | :---: | :--- |
| RUN | MF | Meaning |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | On | Module reports error <br> Bus communication not possible |
| Off | Blinking | Crror in the bus supply voltage |
| Blinking | Configuration error (■) 249) |  |

## Terminals

Module terminals, spring terminals ${ }^{\text {B }}$

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
| $1-\longdiv { \square _ { 5 } } - 5$ | 1 | Analog input AI1 (+) | $0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
|  | 2 | Analog input AI1 (GND) |  |
|  | 3 | Analog input AI3 (+) |  |
|  | 4 | Analog input AI3 (GND) |  |
| $\bigcirc$ | 5 | Analog input Al2 (+) | (AWG 28 ... 16) <br> 10 mm |
|  | 6 | Analog input Al2 (GND) |  |
|  | 7 | Analog input Al4 (+) |  |
| SL10002 | 8 | Analog input Al4 (GND) |  |

## Note!

- Use parameter setting to deactivate unused inputs.
- The module does not provide any auxiliary supply for sensors.

Technical data

| EPM-S403: Rated data |  |
| :---: | :---: |
| Module identifier | 1029 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 70 mA |
| Power loss | 0.7 W |
| Analog inputs |  |
| Number of inputs | 4 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumpt. from load voltage L+ | 15 mA (without load) |
| Current inputs |  |
| Input current ranges | $\begin{aligned} & 0 \mathrm{~mA} \ldots+20 \mathrm{~mA} \\ & +4 \mathrm{~mA} \ldots+20 \mathrm{~mA} \end{aligned}$ |
| Destruction limit (input current) | 40 mA |
| Max. input resistance | $110 \Omega$ |
| Operational error limit | +/- 0.3 \% ... +/- 0.5 \% |
| Basic error limit | +/- 0.2 \% ... +/- 0.3 \% |
| Measuring principle | Gradual approximation |
| Resolution | 12 bits |
| Basic conversion time | 4 ms all channels |
| Interference voltage suppression for a frequency of | > 50 dB at 50 Hz (UCM < 2 V ) |
| Temperature error (relating to input range) | $\pm \square 0.005$ \%/K |
| Linearity distortion (relating to input range) | $\pm \square 0.02$ \% |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the input range) | $\pm \square 0.05$ \% |
| Input data size | 8 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |

## 3 Product description

I/O compound modules - analog I/O
4 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S403

| EPM-S403: Rated data |
| :--- |
| Electrical isolation <br> Between the channels and the backplane <br> bus <br> Between the channels and the voltage <br> supply <br> Max. potential difference between inputs <br> $\left(U_{\text {cm }}\right)$ <br> Max. potential difference between inputs <br> and $M_{\text {internal }}\left(U_{\text {iso }}\right)$ <br> DC 2 V <br> Insulation checked with |

### 3.7.6 2 analog inputs $\mathbf{- 1 0} \ldots+10 \mathrm{~V}$ (16 bits) - EPM-S406

This module detects up to two analog control signals from the process level and transmits them to the higher-level bus system.

## 1 Note!

The UL approval for this module is in preparation.

## Features

- 2 analog inputs
- Voltage range -10 ... +10 V
- 16-bit resolution
- Signal function is parameterisable
- An LED indicates if an input signal is outside the permissible measuring range


## STOP Stop!

## Overvoltage at the inputs

The electronics of the electronic module are not protected against too high input signals.

## Possible consequences:

- The module is destroyed


## Protective measures:

- Make sure that the signals and encoders connected match the measuring range parameterised.


## Overview



Fig. 3-33 Elements and circuit diagram

| A | Displays for module status |
| :--- | :--- |
| B | Terminals |
| $1 \ldots 8$ | Connection number |

## 3 Product description

I/O compound modules - analog I/O
2 analog inputs -10 ... +10 V (16 bits) - EPM-S406

## Status displays

## Module status LEDs A

| View | Pos. | Designation | Colour | Explanation |
| :---: | :---: | :---: | :---: | :---: |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| $1-0$ | 3 | Al1 | Red | On: Channel 1, signal outside the measuring range, error in parameter setting |
|  | 4 | Al2 | Red | On: Channel 2, signal outside the measuring range, error in parameter setting |
|  | 5 | - | - | Not assigned |
|  | 6 |  |  |  |
|  | 7 |  |  |  |
|  | 8 |  |  |  |
|  | 9 |  |  |  |
|  | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■24) $\mathbb{C}$ |

## Terminals

Module terminals, spring terminals

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Analog input AI1 (+) |  |
|  | 2 | Analog input AII (GND) |  |
|  | 3 | Not assigned |  |
|  | 4 | Not assigned |  |
|  | 5 | Analog input Al2 (+) |  |
|  | 6 | Analog input Al2 (GND) |  |
|  | 7 | Not assigned |  |
| su0002 | 8 | Not assigned |  |

## 1 Note!

- Use parameter setting to deactivate unused inputs.
- The module does not provide any auxiliary supply for sensors.

Technical data

| EPM-S406: Rated data |  |
| :---: | :---: |
| Module identifier | 1036 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 60 mA |
| Power loss | 0.8 W |
| Analog inputs |  |
| Number of inputs | 2 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumption from load voltage L+ | 20 mA (without load) |
| Voltage inputs |  |
| Input voltage ranges | -10 V ... +10 V |
| Min. input resistance | $200 \mathrm{k} \Omega$ |
| Operational error limit | +/- 0.2 \% |
| Basic error limit | +/- 0.1 \% |
| Measuring principle | Gradual approximation |
| Resolution | 16 bits |
| Basic conversion time | $240 \mu \mathrm{~s}$ all channels |
| Interference voltage suppression for a frequency of | > 80 dB at 50 Hz (UCM < 9 V ) |
| Input data size | 4 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Between the channels and the voltage supply | Yes |
| Max. potential difference between inputs ( $U_{\mathrm{cm}}$ ) | DC 9 V |
| Max. potential difference between inputs and $M_{\text {internal }}\left(\mathrm{U}_{\text {iso }}\right)$ | DC $75 \mathrm{~V} / \mathrm{AC} 60 \mathrm{~V}$ |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound modules - analog I/O
2 analog inputs 0/4 ... 20 mA (16 bits) - EPM-S408

### 3.7.7 2 analog inputs $0 / 4$... 20 mA (16 bits) - EPM-S408

This module detects up to four analog control signals from the process level and transmits them to the higher-level bus system.

## 1 Note!

The UL approval for this module is in preparation.

## Features

- 2 analog inputs
- Current range 0/4 ... 20 mA
- 16-bit resolution
- Signal function is parameterisable
- An LED indicates if an input signal is outside the permissible measuring range


## STOP Stop!

Overvoltage at the inputs
The electronics of the electronic module are not protected against too high input signals.

## Possible consequences:

- The module is destroyed

Protective measures:

- Make sure that the signals and encoders connected match the measuring range parameterised.


## Overview



Fig. 3-34 Elements and circuit diagram

| (A | Displays for module status |
| :--- | :--- |
| B | Terminals |
| $1 \ldots 8$ | Connection number |

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| $1-0 \square$ | 3 | Al1 | Red | On: Channel 1, signal outside the measuring range, error in parameter setting |
| $\square$ $\qquad$ | 4 | Al2 | Red | On: Channel 2, signal outside the measuring range, error in parameter setting |
| $\left\lvert\, \begin{aligned} & u \\ & \square \end{aligned}\right.$ $\qquad$ | 5 |  |  |  |
| $\left.\left\lvert\, \begin{array}{ll} \square \\ 0 \end{array}\right.\right)$ | 6 |  |  |  |
| $10 \left\lvert\, \begin{gathered} 0 \\ 0 \end{gathered}\right.$ | 7 |  | - |  |
|  | 8 | - | - | Not assigned |
|  | 9 |  |  |  |
| SL10001 | 10 |  |  |  |

Messages of the status LEDS RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■24) $\mathbb{C}$ |

## Terminals

Module terminals, spring terminals B

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Analog input AI1 (+) |  |
|  | 2 | Analog input AI1 (GND) |  |
|  | 3 | Not assigned |  |
|  | 4 |  |  |
|  | 5 | Analog input Al2 (+) |  |
|  | 6 | Analog input AI2 (GND) |  |
|  | 7 |  |  |
| SL10002 | 8 | Not assigned |  |

## 1 Note!

- Use parameter setting to deactivate unused inputs.
- The module does not provide any auxiliary supply for sensors.


## 3 Product description

I/O compound modules - analog I/O
2 analog inputs $0 / 4$... 20 mA (16 bits) - EPM-S408

Technical data

| EPM-S408: Rated data |  |
| :---: | :---: |
| Module identifier | 1035 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 60 mA |
| Power loss | 0.7 W |
| Analog inputs |  |
| Number of inputs | 2 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumption from load voltage L+ | 15 mA (without load) |
| Current inputs |  |
| Input current ranges | $\begin{aligned} & 0 \mathrm{~mA} \ldots+20 \mathrm{~mA} \\ & +4 \mathrm{~mA} \ldots+20 \mathrm{~mA} \end{aligned}$ |
| Max. input resistance | $60 \Omega$ |
| Operational error limit | +/- 0.2 \% |
| Basic error limit | +/- 0.1 \% |
| Measuring principle | Gradual approximation |
| Resolution | 16 bits |
| Basic conversion time | $240 \mu \mathrm{~s}$ all channels |
| Interference voltage suppression for a frequency of | > $80 \mathrm{~dB}(\mathrm{UCM}<4 \mathrm{~V})$ |
| Input data size | 4 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Between the channels and the voltage supply | Yes |
| Max. potential difference between inputs ( $\mathrm{U}_{\mathrm{cm}}$ ) | DC 4 V |
| Max. potential difference between inputs and $M_{\text {internal }}\left(U_{\text {iso }}\right)$ | DC $75 \mathrm{~V} / \mathrm{AC} 60 \mathrm{~V}$ |
| Insulation checked with | DC 500 V |

3.7.8 2 analog outputs $0 \ldots 10 \mathrm{~V}$ (12 bits) - EPM-S500

This module detects up to two analog control signals from the higher-level bus system and transmits them to the process level.

## Features

- 2 analog outputs
- Voltage output $0 \ldots+10 \mathrm{~V}$
- 12-bit resolution
- Signal function is parameterisable
- 24 V DC supply voltage
- A reference potential for all outputs
- An LED indicates if an output signal is outside the permissible measuring range


## Overview



Fig. 3-35 Elements and circuit diagram
A Displays for module status
B Terminals
1 ... 8 Connection number

## 3 Product description

I/O compound modules - analog I/O
2 analog outputs $0 \ldots 10 \mathrm{~V}$ ( 12 bits) - EPM-S500

## Status displays

## Module status LEDs A

| View | Pos. | Designation | Colour | Explanation |
| :---: | :---: | :---: | :---: | :---: |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| $1-$ | 3 | AO1 | Red | On: Channel 1, overload, short circuit, error in parameter setting |
|  | 4 | AO2 | Red | On: Channel 2, overload, short-circuit, error in parameter setting |
|  | 5 | - | - | Not assigned |
|  | 6 |  |  |  |
|  | 7 |  |  |  |
|  | 8 |  |  |  |
|  | 9 |  |  |  |
|  | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK <br> On |
| On | Module reports error <br> Bus communication is OK |  |
| Off | On | Module reports error <br> Bus communication not possible |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■249) |

## Terminals

Module terminals, spring terminals B

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Analog output AO1 (+) | $\xrightarrow[10 \mathrm{~mm}]{(\text { AWG } 28 \ldots 16)}$ |
|  | 2 | Analog output AO1 (GND) |  |
|  | 3 | Not assigned |  |
|  | 4 |  |  |
|  | 5 | Analog output AO2 (+) |  |
|  | 6 | Analog output AO2 (GND) |  |
| $4^{4}$ | 7 |  |  |
| SL10002 | 8 | Not assigned |  |

## 1 Note!

- When connecting the actuators, make sure that the polarity is correct.
- Outputs that are not used are not connected.
- The module does not provide any auxiliary supply for actuators.

Technical data

| EPM-S500: Rated data |  |
| :---: | :---: |
| Module identifier | 1281 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 80 mA |
| Power loss | 1.2 W |
| Analog outputs |  |
| Number of outputs | 2 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumpt. from load voltage L+ | 35 mA (without load) |
| Voltage outputs |  |
| Output voltage ranges | 0 V ... +10 V |
| Min. load impedance | $5 \mathrm{k} \Omega$ |
| Max. capacitive load | $1 \mu \mathrm{~F}$ |
| Operational error limit | +/-0.3 \% |
| Basic error limit | +/- 0.2 \% |
| Short circuit protection | Yes |
| Voltage at the outputs | 15 V |
| Current | 30 mA |
| Interference suppression (cross-talk between the outputs) | $>40 \mathrm{~dB}$ |
| Resolution | 12 bits |
| Basic conversion time | 2 ms all channels |
| Substitute values can be applied | Yes |
| Temperature error (relating to the output range) |  |
| Linearity distortion (relating to the output range) | + $\square 0.1$ \% |
| Output ripple; bandwidth 0 to 50 kHz (relating to the output range) | $\pm \square 0.05$ \% |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the output range) | $\pm \square 0.05$ \% |
| Dwell time |  |
| for ohmic loads | 1.5 ms |
| for capacitive loads | 2 ms |
| Output data size | 4 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |

## 3 Product description

I/O compound modules - analog I/O
2 analog outputs $0 \ldots 10 \mathrm{~V}$ ( 12 bits) - EPM-S500

| Electrical isolation |  |
| :---: | :---: |
| Between the channels and the backplane bus | Yes |
| Between the channels and the voltage supply | Yes |
| Max. potential difference between inputs and Minternal ( $\mathrm{U}_{\text {iso }}$ ) | DC $75 \mathrm{~V} / \mathrm{AC} 60 \mathrm{~V}$ |
| Insulation checked with | DC 500 V |

### 3.7.9 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501

This module detects up to four analog control signals from the higher-level bus system and transmits them to the process level.

## Features

- 4 analog outputs
- Voltage output $0 \ldots+10 \mathrm{~V}$
- 12-bit resolution
- Signal function is parameterisable
- 24 V DC supply voltage
- A reference potential for all outputs
- An LED indicates if an output signal is outside the permissible measuring range


## Overview



Fig. 3-36 Elements and circuit diagram
A Displays for module status
B Terminals
1 ... 8 Connection number

## 3 Product description

I/O compound modules - analog I/O
4 analog outputs $0 \ldots 10 \mathrm{~V}$ ( 12 bits) - EPM-S501

## Status displays



| Messages of the status LEDs RUN and MF |  |  |
| :---: | :---: | :--- |
| RUN | MF | Meaning |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | On | Module reports error <br> Bus communication not possible |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (Cl 249) |

## Terminals

Module terminals, spring terminals B

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Analog output AO1 (+) | $\longrightarrow \xrightarrow{(10 \mathrm{~mm}} \stackrel{0.08 \ldots 1.5 \mathrm{~mm}^{2}}{(\text { AWG } 28 \ldots 16)}$ |
|  | 2 | Analog output AO1 (GND) |  |
|  | 3 | Analog output AO3 (+) |  |
|  | 4 | Analog output AO3 (GND) |  |
|  | 5 | Analog output AO2 (+) |  |
|  | 6 | Analog output AO2 (GND) |  |
|  | 7 | Analog output AO4 (+) |  |
| su10002 | 8 | Analog output AO4 (GND) |  |

## Note!

- When connecting the actuators, make sure that the polarity is correct.
- Outputs that are not used are not connected.
- The module does not provide any auxiliary supply for actuators.

Technical data

| EPM-S501: Rated data |  |
| :---: | :---: |
| Module identifier | 1283 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 80 mA |
| Power loss | 1.2 W |
| Analog outputs |  |
| Number of outputs | 4 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumpt. from load voltage L+ | 35 mA (without load) |
| Voltage outputs |  |
| Output voltage ranges | 0 V ... +10 V |
| Min. load impedance | $5 \mathrm{k} \Omega$ |
| Max. capacitive load | $1 \mu \mathrm{~F}$ |
| Operational error limit | +/-0.3 \% |
| Basic error limit | +/-0.2 \% |
| Short circuit protection | Yes |
| Voltage at the outputs | 15 V |
| Current | 30 mA |
| Interference suppression (cross-talk between the outputs) | $>40 \mathrm{~dB}$ |
| Resolution | 12 bits |
| Basic conversion time | 2 ms all channels |
| Substitute values can be applied | Yes |
| Temperature error (relating to the output range) | + $\square 0.01$ \%/K |
| Linearity distortion (relating to the output range) | $\pm 70.1$ \% |
| Output ripple; bandwidth 0 to 50 kHz (relating to the output range) | $\pm \square 0.05$ \% |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the output range) | $\pm \square 0.05$ \% |
| Dwell time |  |
| for ohmic loads | 1.5 ms |
| for capacitive loads | 2 ms |
| Output data size | 8 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |

## 3 Product description

I/O compound modules - analog I/O
4 analog outputs $0 \ldots 10 \mathrm{~V}$ ( 12 bits) - EPM-S501

| EPM-S501: Rated data |
| :--- |
| Electrical isolation <br> Between the channels and the backplane <br> bus <br> Between the channels and the voltage <br> supply <br> Max. potential difference between inputs <br> and Minternal ( $\mathrm{U}_{\text {iso }}$ ) <br> Yes <br> Insulation checked with |

This module detects up to two analog control signals from the higher-level bus system and transmits them to the process level.

## Features

- 2 analog outputs
- Current output 0/4 ... 20 mA
- 12-bit resolution
- Signal function is parameterisable
- 24 V DC supply voltage
- A reference potential for all outputs
- An LED indicates if an output signal is outside the permissible measuring range


## Overview



Fig. 3-37 Elements and circuit diagram
A Displays for module status
B Terminals
1 ... 8 Connection number

## 3 Product description

I/O compound modules - analog I/O
2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| $\square$ $\qquad$ | 3 | AO1 | Red | On: Channel 1, open circuit, parameterisation error |
| $\square$ | 4 | AO2 | Red | On: Channel 2, open circuit, parameterisation error |
| $\left\lvert\, \begin{array}{lll} \square \\ \square \end{array}\right.$ | 5 |  |  |  |
|  | 6 |  |  |  |
| $\begin{aligned} & 0 \\ & 0 \end{aligned}$ $\qquad$ | 7 |  |  |  |
| $10-\frac{a}{a}$ | 8 | - | - | Not assigned |
|  | 9 |  |  |  |
| SLuoour | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■) 249) |

## Terminals

Module terminals, spring terminals ${ }^{\text {B }}$

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
| $1-\sqrt{\square \Xi^{5}}-5$ | 1 | Analog output AO1 (+) | $0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
|  | 2 | Analog output AO1 (GND) |  |
|  | 3 | Not assigned |  |
|  | 4 |  |  |
|  | 5 | Analog output AO2 (+) | (AWG 28 ... 16) |
|  | 6 | Analog output AO2 (GND) |  |
| - | 7 |  |  |
| stu002 | 8 | Not assig |  |

## 1 Note!

- When connecting the actuators, make sure that the polarity is correct.
- Outputs that are not used are not connected.
- The module does not provide any auxiliary supply for actuators.


## Technical data

| EPM-S502: Rated data |  |
| :---: | :---: |
| Module identifier | $1282_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 80 mA |
| Power loss | 0.8 W |
| Analog outputs |  |
| Number of outputs | 2 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumpt. from load voltage L+ | 15 mA (without load) |
| Current outputs |  |
| Output current ranges | $\begin{aligned} & 0 \mathrm{~mA} \ldots+20 \mathrm{~mA} \\ & +4 \mathrm{~mA} \ldots+20 \mathrm{~mA} \end{aligned}$ |
| Max. load impedance | $350 \Omega$ |
| Max. inductive load | 10 mH |
| Operational error limit | +/- $0.4 \% \ldots+/-0.5$ \% |
| Basic error limit | +/- 0.2 \% ... +/- 0.3 \% |
| Short circuit protection | No |
| Voltage at the outputs | 12 V |
| Interference suppression (cross-talk between the outputs) | $>40 \mathrm{~dB}$ |
| Resolution | 12 bits |
| Basic conversion time | 2 ms all channels |
| Substitute values can be applied | Yes |
| Temperature error (relating to the output range) | $\pm \square 0.01 \% / K$ |
| Linearity distortion (relating to the output range) | $\pm 70.1$ \% |
| Output ripple; bandwidth 0 to 50 kHz (relating to the output range) | $\pm 70.05 \%$ |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the output range) | $\pm 70.05 \%$ |
| Dwell time |  |
| for ohmic loads | 0.25 ms |
| for inductive load | 1.5 ms |
| Output data size | 4 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |

## 3 Product description

I/O compound modules - analog I/O
2 analog outputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S502

| Electrical isolation |  |
| :---: | :---: |
| Between the channels and the backplane bus | Yes |
| Between the channels and the voltage supply | Yes |
| Max. potential difference between inputs and Minternal ( $\mathrm{U}_{\text {iso }}$ ) | DC $75 \mathrm{~V} / \mathrm{AC} 60 \mathrm{~V}$ |
| Insulation checked with | DC 500 V |

3.7.11 4 analog outputs $0 / 4$... 20 mA (12 bits) - EPM-S503

This module detects up to four analog control signals from the higher-level bus system and transmits them to the process level.

## Features

- 4 analog outputs
- Current output 0/4 ... 20 mA
- 12-bit resolution
- Signal function is parameterisable
- 24 V DC supply voltage
- A reference potential for all outputs
- An LED indicates if an output signal is outside the permissible measuring range


## Overview



Fig. 3-38 Elements and circuit diagram
A Displays for module status
B Terminals
1 ... 8 Connection number

## 3 Product description

I/O compound modules - analog I/O
4 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S503

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| - | 3 | AO1 | Red | On: Channel 1, open circuit, parameterisation error |
| - | 4 | AO2 | Red | On: Channel 2, open circuit, parameterisation error |
| - | 5 | AO3 | Red | On: Channel 3, open circuit, parameterisation error |
|  | 6 | AO4 | Red | On: Channel 4, open circuit, parameterisation error |
| $\square$ | 7 |  |  |  |
| $10-\square$ | 8 |  |  |  |
|  | 9 | - | - | Not assigned |
| SLu000 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■249) |

## Terminals

Module terminals, spring terminals ${ }^{\text {B }}$

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
| $1-\longdiv { \square _ { 5 } } - 5$ | 1 | Analog output AO1 (+) | $\xrightarrow{(10 \mathrm{~mm}} \stackrel{0.08 \ldots 1.5 \mathrm{~mm}^{2}}{(\text { AWG } 28 \ldots 16)}$ |
|  | 2 | Analog output AO1 (GND) |  |
|  | 3 | Analog output AO3 (+) |  |
|  | 4 | Analog output AO3 (GND) |  |
|  | 5 | Analog output AO2 (+) |  |
|  | 6 | Analog output AO2 (GND) |  |
| $0^{4}$ | 7 | Analog output AO4 (+) |  |
| SL10002 | 8 | Analog output AO4 (GND) |  |

## 1 Note!

- When connecting the actuators, make sure that the polarity is correct.
- Outputs that are not used are not connected.
- The module does not provide any auxiliary supply for actuators.


## Technical data

| EPM-S503: Rated data |  |
| :---: | :---: |
| Module identifier | 1284 ${ }_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 80 mA |
| Power loss | 0.8 W |
| Analog outputs |  |
| Number of outputs | 4 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumpt. from load voltage L+ | 15 mA (without load) |
| Current outputs |  |
| Output current ranges | $\begin{aligned} & 0 \mathrm{~mA} \ldots+20 \mathrm{~mA} \\ & +4 \mathrm{~mA} \ldots+20 \mathrm{~mA} \end{aligned}$ |
| Max. load impedance | $350 \Omega$ |
| Max. inductive load | 10 mH |
| Operational error limit | +/- $0.4 \% \ldots+/-0.5$ \% |
| Basic error limit | +/- 0.2 \% ... +/- 0.3 \% |
| Short circuit protection | No |
| Voltage at the outputs | 12 V |
| Interference suppression (cross-talk between the outputs) | $>40 \mathrm{~dB}$ |
| Resolution | 12 bits |
| Basic conversion time | 2 ms all channels |
| Substitute values can be applied | Yes |
| Temperature error (relating to the output range) | $\pm \square 0.01 \% / K$ |
| Linearity distortion (relating to the output range) | $\pm \square 0.1$ \% |
| Output ripple; bandwidth 0 to 50 kHz (relating to the output range) | $\pm 70.05 \%$ |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the output range) | $\pm 70.05 \%$ |
| Dwell time |  |
| for ohmic loads | 0.25 ms |
| for inductive load | 1.5 ms |
| Output data size | 8 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |

## 3 Product description

I/O compound modules - analog I/O
4 analog outputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S503

| EPM-S503: Rated data |
| :--- |
| Electrical isolation <br> Between the channels and the backplane <br> bus <br> Between the channels and the voltage <br> supply <br> Max. potential difference between inputs <br> and Minternal ( $\mathrm{U}_{\text {iso }}$ ) <br> Yes <br> Insulation checked with |

## $3.8 \quad$ I/O compound modules - temperature measurement

### 3.8.1 Representation of analog values

Analog values can only be processed in a binary form. For this, the analog module converts each process signal into a digital form and passes it on as a word.

| Resolution | Analog value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | HIGH byte (byte 0) |  |  |  |  |  |  |  | LOW byte (byte 1) |  |  |  |  |  |  |  |
| Bit number | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Valency | Sign bit | $2^{14}$ | $2^{13}$ | $2^{12}$ | $2^{11}$ | $2^{10}$ | $2^{9}$ | $2^{8}$ | $2^{7}$ | $2^{6}$ | $2^{5}$ | $2^{4}$ | $2^{3}$ | $2^{2}$ | $2^{1}$ | $2^{0}$ |
| 12 bits + sign bit | Sign bit | Measured value |  |  |  |  |  |  |  |  |  |  |  | 0 | 0 | 0 |
| 15 bits + sign bit | Sign bit | Measured value |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Resolution: In the case of a resolution of 12 bits plus sign bit, the lower-order digits that are not used ( 3 bits) are written to with " 0 ".

Sign bit: Bit $15=" 0 " \rightarrow$ positive value; bit $15=" 1 " \rightarrow$ negative value.
Response in the event of an error: If a measured value exceeds the overrange or falls below the lower range, the following value is output:
Measured value > overrange $\rightarrow 32767$ ( $7 \mathrm{FFF}_{\mathrm{h}}$ )
Measured value < lower range $\rightarrow-32768\left(8000_{h}\right)$
If a parameterisation error occurs, the measured value $32767\left(7 \mathrm{FFF}_{h}\right)$ is output.

## 3 Product description

I/O compound modules - temperature measurement
Four (two) analog inputs for resistance tests - EPM-S404

### 3.8.2

Four (two) analog inputs for resistance tests - EPM-S404
This module detects up to four analog control signals from the process level and transmits them to the higher-level bus system.

## Features

- 4 analog inputs (for 2-wire technology) or 2 analog inputs (for 3- and 4-wire technology)
- For resistance-type sensors 0 ... 3000 and resistance temperature sensors Pt100, Pt1000, Ni100 or Ni1000
- 16-bit resolution
- Signal function is parameterisable
- An LED indicates if an input voltage is outside the permitted measuring range


## STOP Stop!

Overvoltage at the inputs
The electronics of the electronic module are not protected against too high input signals.

## Possible consequences:

- The module is destroyed

Protective measures:

- Make sure that the signals and encoders connected match the measuring range parameterised.


## Overview



Fig. 3-39 Elements and circuit diagram
$\begin{array}{ll}\text { © } & \text { Displays for module status } \\ \text { B } & \text { Terminals } \\ 1 \ldots 8 & \text { Connection number }\end{array}$

I/O compound modules - temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

## Status displays

Module status LEDs A

| View | Pos. | Designation | Colour | Explanation |
| :---: | :---: | :---: | :---: | :---: |
|  | 1 | RUN | Green | On: Module is ready for operation (see following table) |
|  | 2 | MF | Red | On: Module error (see table below) |
|  | 3 | Al1 | Red | On: Channel 1, signal outside the measuring range, parameterisation error, open circuit |
|  | 4 | Al2 | Red | On: Channel 2, signal outside the measuring range, parameterisation error, open circuit |
|  | 5 | Al3 | Red | On: Channel 3 , signal outside the measuring range, parameterisation error, open circuit |
|  | 6 | Al4 | Red | On: Channel 4, signal outside the measuring range, parameterisation error, open circuit |
|  | 7 | - | - | Not assigned |
|  | 8 |  |  |  |
|  | 9 |  |  |  |
| SL10001 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (Cl 249) |

## Terminals

Module terminals, spring terminals B

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Analog input AI1 (+) | $\xrightarrow[(10 \mathrm{~mm}]{(\mathrm{NWG} 28 \ldots 16)}$ |
|  | 2 | Analog input AI1 (GND) |  |
|  | 3 | Analog input AI3 (+) |  |
|  | 4 | Analog input AI3 (GND) |  |
|  | 5 | Analog input Al2 (+) |  |
|  | 6 | Analog input AI2 (GND) |  |
|  | 7 | Analog input Al4 (+) |  |
| SL10002 | 8 | Analog input AI4 (GND) |  |

## Note!

- Use parameter setting to deactivate unused inputs.
- If thermal detectors are connected in a 3 or 4 conductor setup, channels 2 and/or 4 must be deactivated.
- The module does not provide any auxiliary supply for sensors.


## 3 Product description

I/O compound modules - temperature measurement
Four (two) analog inputs for resistance tests - EPM-S404

Technical data

| EPM-S404: Rated data |  |
| :---: | :---: |
| Module identifier | 1030 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 75 mA |
| Power loss | 1 W |
| Analog inputs |  |
| Number of inputs | 4 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumption from load voltage L+ | 30 mA (without load) |
| Resistor inputs |  |
| Resistance ranges | $\begin{aligned} & 0 \ldots 60 \Omega \\ & 0 \ldots 600 \Omega \\ & 0 \ldots 3000 \Omega \end{aligned}$ |
| Operational error limit | $\pm \square 0.4$ \% |
| Basic error limit | $\pm \square 0.2$ \% |
| Resistance thermometer inputs |  |
| Resistance thermometer ranges | Pt100 <br> Pt1000 <br> Ni100 <br> Ni1000 |
| Operational error limit | $\pm \square 0.4$ \% |
| Basic error limit | $\pm \square 0.2$ \% |
| Measuring principle | Sigma-delta |
| Resolution | 16 bits |
| Basic conversion time | 4.2 ... $324.1 \mathrm{~ms}(50 \mathrm{~Hz})$ per channel 3.8 ... $270.5 \mathrm{~ms}(60 \mathrm{~Hz})$ per channel |
| Interference voltage suppression for a frequency of | > 80 dB at $50 \mathrm{~Hz}(\mathrm{UCM}<60 \mathrm{~V})$ |
| Destruction limit (input voltage) | 9 V |
| Temperature error (relating to input range) | $\pm \square 0.005$ \%/K |
| Linearity distortion (relating to input range) | $\pm \square 0.005$ \% |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the input range) | $\pm 0.05$ \% |
| Input data size | 8 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |


| EPM-S404: Rated data |
| :--- |
| Electrical isolation <br> Between the channels and the backplane <br> bus <br> Between the channels and the voltage <br> supply <br> Max. potential difference between inputs <br> $\left(U_{\text {cm }}\right)$ <br> Max. potential difference between inputs <br> and $M_{\text {internal }}\left(U_{\text {iso }}\right)$ <br> DC 6 V <br> Insulation checked with |

## 3 Product description

I/O compound modules - temperature measurement
Two analog inputs for thermocouple measurement - EPM-S405

### 3.8.3 Two analog inputs for thermocouple measurement - EPM-S405

This module detects up to two analog control signals from the process level and transmits them to the higher-level bus system.

## Features

- 2 analog inputs
- For thermocouple type B, C, E, J, K, L, N, R, S or T
- 16-bit resolution
- Internal temperature compensation
- Signal function is parameterisable
- An LED indicates if an input voltage is outside the permitted measuring range


## sTOP Stop!

Overvoltage at the inputs
The electronics of the electronic module are not protected against too high input signals.

## Possible consequences:

- The module is destroyed


## Protective measures:

- Make sure that the signals and encoders connected match the measuring range parameterised.


## Mounting instructions

Variations in temperature within the module may affect the measuring accuracy. Therefore please observe the following recommendations:

- Do not place the module...
- directly next to the bus coupler module
- directly next to a power supply module
- in a position as the last module within an I/O system
- If possible, the ambient temperature should be constant. After a change in the ambient temperature, the module requires approx. 30 minutes until obtaining its ensured measuring accuracy.

I/O compound modules - temperature measurement

## Overview



Fig. 3-40 Elements and circuit diagram
A Displays for module status
B Terminals
1 ... 8 Connection number

## Status displays

Module status LEDs A

| View | Pos. | Designation | Colour | Explanation |
| :---: | :---: | :---: | :---: | :---: |
|  | 1 | RUN | Green | On: Module is ready for operation (see following table) |
|  | 2 | MF | Red | On: Module error (see table below) |
| $1-$ | 3 | Al1 | Red | On: Channel 1, signal outside the measuring range, parameterisation error, open circuit |
| $\left[\begin{array}{l\|l} 0 \\ 0 \\ 0 & - \\ \hline \end{array}\right.$ | 4 | Al2 | Red | On: Channel 2, signal outside the measuring range, parameterisation error, open circuit |
| $\left\|\begin{array}{l} 0 \\ 0 \end{array}\right\|$ | 5 | - | - | Not assigned |
|  | 6 |  |  |  |
| 10- | 7 |  |  |  |
|  | 8 |  |  |  |
|  | 9 |  |  |  |
| sulooor | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (C) 249) |

## 3 Product description

I/O compound modules - temperature measurement Two analog inputs for thermocouple measurement - EPM-S405

## Terminals

Module terminals, spring terminals $B$

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Analog input AI1 (+) |  |
|  | 2 | Analog input AI1 (GND) |  |
|  | 3 | Not assigned |  |
|  | 4 |  |  |
|  | 5 | Analog input Al2 (+) |  |
|  | 6 | Analog input Al2 (GND) |  |
| $4 \bigcirc 8$ | 7 |  |  |
| SL10002 | 8 | Not assigned |  |

## Note!

- Use parameter setting to deactivate unused inputs.
- The module does not provide any auxiliary supply for sensors.


## Technical data

| EPM-S405: Rated data |  |
| :---: | :---: |
| Module identifier | 1027 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 75 mA |
| Power loss | 1 W |
| Analog inputs |  |
| Number of inputs | 2 |
| Cable length |  |
| shielded | 200 m |
| Load voltage |  |
| Nominal value | DC 24 V |
| Current consumption from load voltage L+ | 30 mA (without load) |
| Voltage inputs |  |
| Input voltage ranges | -80 mV ... +80 mV |
| Destruction limit (input voltage) | 30 V |
| Operational error limit | With interference frequency suppression: $\mp \square 0.1$ \% Without interference frequency suppression: $\ddagger \square 0.3$ \% |
| Basic error limit | With interference frequency suppression: $\ddagger \square 0.05$ \% Without interference frequency suppression: $\pm \square 0.25$ \% |

## I/O compound modules - temperature measurement

 Two analog inputs for thermocouple measurement - EPM-S405| EPM-S405: Rated data |  |
| :---: | :---: |
| Thermocouple inputs |  |
| Thermocouple ranges | Types B, C, E, J, K, L, N, R, S, T |
| Operational error limit | With interference frequency suppression: <br> Types E, L, T, J, K, N: $\square \downarrow 1.5 \mathrm{~K}$ <br> Types B, C, R, S: $\pm 74.0 \mathrm{~K}$ <br> Without interference frequency suppression: <br> Types E, L, T, J, K, N: $\pm \square 2.5 \mathrm{~K}$ <br> Types B, C, R, S: $\pm \square 8.0 \mathrm{~K}$ |
| Basic error limit | With interference frequency suppression: <br> Types E, L, T, J, K, N: $\ddagger 1.0 \mathrm{~K}$ <br> Types B, C, R, S: $\pm \square 3.0 \mathrm{~K}$ <br> Without interference frequency suppression: <br> Types E, L, T, J, K, N: $\ddagger \square 2.0 \mathrm{~K}$ <br> Types B, C, R, S: $\pm 77.0 \mathrm{~K}$ |
| Measuring principle | Sigma-delta |
| Resolution | 16 bits |
| Basic conversion time | 4.2 ... $324.1 \mathrm{~ms}(50 \mathrm{~Hz})$ per channel 3.8 ... $270.5 \mathrm{~ms}(60 \mathrm{~Hz})$ per channel |
| Interference voltage suppression for a frequency of | > 90 dB at $50 \mathrm{~Hz}(\mathrm{UCM}<10 \mathrm{~V})$ |
| Temperature error (relating to input range) | $\pm \square 0.001 \% / K$ |
| Linearity distortion (relating to input range) | + 70.005 \% |
| Repeat accuracy (in steady-state vibration at $25^{\circ} \mathrm{C}$, relating to the input range) | + $0.05 \%$ |
| Temperature error of the internal compensation | $\pm 70.2 \%$ |
| Temperature compensation |  |
| parameterisable | Yes |
| external | Yes |
| internal | Yes |
| Input data size | 4 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | Red LEDs per channel |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Between the channels and the voltage supply | Yes |
| Max. potential difference between inputs ( $\mathrm{U}_{\mathrm{cm}}$ ) | DC $140 \mathrm{~V} / \mathrm{AC} 60 \mathrm{~V}$ |
| Max. potential difference between inputs and $M_{\text {internal }}\left(U_{\text {iso }}\right)$ | DC $75 \mathrm{~V} / \mathrm{AC} 60 \mathrm{~V}$ |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound modules - counter
One counter 32 bits, 24 V DC - EPM-S600

## $3.9 \quad$ I/O compound modules - counter

### 3.9.1 One counter 32 bits, 24 V DC - EPM-S600

This module measures the pulses of one connected encoder and processes them according to the mode selected.

## Features

- 1 counter 32 bits (AB), invertible, DC 24 V
- Counting frequency max. 400 kHz
- Latch value, comparison value, set value, input filter
- Hardware gate, digital output
- Alarm and diagnostic function


## Overview



Fig. 3-41 Elements and circuit diagram

| A | Displays for module status |
| :--- | :--- |
| B | Terminals |
| $1 \ldots 8$ | Connection number |

Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
|  | 3 | A | Green | On: Digital input 1 " ${ }^{\text {"//"pulse" triggered }}$ |
|  | 4 | B |  | On: Digital input 5 " ${ }^{\text {" } / / " d i r e c t i o n " ~ t r i g g e r e d ~}$ |
|  | 5 | Latch |  | On: Digital input 4 "Latch" triggered |
|  | 6 | Gate |  | On: Digital input 8 "Hardware gate" triggered |
|  | 7 | Reset |  | On: Digital input 7 "Reset" triggered |
|  | 8 | OUT |  | On: Digital output "OUT" triggered |
|  | 9 |  |  |  |
|  | 10 | - |  | Not assigned |

Messages of the status LEDS RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | On | Module reports error <br> Bus communication not possible |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■249) |

Terminals


## $3 \quad$ Product description

I/O compound modules - counter
One counter 32 bits, 24 V DC - EPM-S600

## Technical data

| EPM-S600: rated data |  |
| :---: | :---: |
| Module identifier | $2241_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 75 mA |
| Power loss | 1 W |
| Digital inputs |  |
| Number of inputs | 5 |
| Cable length |  |
| shielded | 100 m |
| Load voltage |  |
| Nominal value | DC 20.4 ... 28.8 V |
| Current consumption from load voltage L+ | 20 mA (without load) |
| Input voltage |  |
| Nominal value | DC 20.4 ... 28.8 V |
| for signal "0" | DC $0 . . .5 \mathrm{~V}$ |
| for signal "1" | DC $15 . . .28 .8 \mathrm{~V}$ |
| Input current |  |
| for signal "1" | 3 mA |
| 2-wire BERO |  |
| Connection possible | Yes |
| Max. permissible closed-circuit current | 0.5 mA |
| Input delay |  |
| from "0" to "1" | $0.8 \mu s$ |
| from " 1 " to "0" | 0.8 ¢s |

Number of inputs which can be used simultaneously

| horizontal structure | 5 |
| :---: | :---: |
| vertical structure | 5 |
| Input characteristic | IEC 61131, type 1 |
| Input data size | 12 bytes |
| Digital outputs |  |
| Number of outputs | 1 |
| Cable length |  |
| shielded | 100 m |
| unshielded | 100 m |
| Load voltage |  |
| Nominal value | DC 20.4 ... 28.8 V |
| Output delay |  |
| from "0" to " 1 " | $30 \mu \mathrm{~s}$ |
| from " 1 " to "0" | $30 \mu \mathrm{~s}$ |
| Lamp load | 10 W |
| Parallel switching of outputs |  |
| for redundant control | Not possible |
| for power increase | Not possible |
| Control of a digital input | Possible |

## I/O compound modules - counter

| EPM-S600: rated data |  |
| :---: | :---: |
| Switching frequencies |  |
| for ohmic loads, max. | 10 kHz |
| for inductive loads, max. | 0.5 Hz |
| for lamp loads, max. | 10 kHz |
| Limitation (internal) of the inductive breaking voltage | $\mathrm{L}+(-52 \mathrm{~V})$ |
| Short circuit protection |  |
| Type | Electronically |
| Operating threshold | 1 A |
| Output data size | 10 bytes |
| Counter |  |
| Number of counters | 1 |
| Counter width | 32 bits |
| Frequency |  |
| Input frequency, max | 100 kHz |
| Counting frequency, max | 400 kHz |
| Operating mode |  |
| Incremental encoder | Possible |
| Pulse/direction | Possible |
| Pulse | Not possible |
| Frequency measurement | Not possible |
| Period duration measurement | Not possible |
| Connection |  |
| Gate connection | Possible |
| Latch connection | Possible |
| Reset connection | Possible |
| Counter output | Possible |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes, parameterisable |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## Parameterisable functions

## Counting functions

Counting continuously
Counting once
Counting periodically

| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input " $\mathrm{A} / \mathrm{pulse}$ " and direction at " $\mathrm{B} /$ direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate), hardware gate (HW gate), and software gate (SW gate) is made. <br> - The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). <br> - The SW gate is controlled via your user program (status word in the output area). <br> - The HW gate is controlled via the digital gate input. <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

## Counting continuously

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31-1}\right)$ |

Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts from the loading value in the counting range.
- If the counter reaches the upper counting limit when counting forwards, and if a further count pulse in positive direction occurs, the counter skips to the lower counting limit and continues to count from there.
- If the counter reaches the lower counting limit when counting backwards, and if a further negative count pulse occurs, the counter skips to the upper counting limit and continues to count from there.
- If maximum values are exceeded or minimum values are not reached, the status bits STS_OFLW or STS_UFLW are set. These bits remain set until they are reset with RES_SET in the control word. If enabled, additionally a process alarm is triggered.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-42 Example of "Counting continuously"

## 3 Product description

I/O compound modules - counter
One counter 32 bits, 24 V DC - EPM-S600

## Counting once

A) No main counting direction:

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31}-1\right)$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards once from the loading value or backwards in the preset counting range.
- If the counting limits are not reached or exceeded, the counter skips to the other counting limit, respectively, the internal gate is closed automatically and the status bits STS_OFLW or STS_UFLW are set. If enabled, a process alarm is effected.
- For stopping the counting process the internal gate has to be closed.
- For starting the counting process again, the internal gate has to be opened.
- If the gate control is interrupted, the counting process continues at the current counter content.
- If the gate control is cancelled, the counter starts from the loading value.


Fig. 3-43 Example of "Counting once", no main counting direction and with interrupting gate control


Fig. 3-44 Example of "Counting once", no main counting direction and with cancelling gate control
B) Main counting direction forwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31}-1\right)$ |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |

Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards from the loading value.
- If the counter reaches the final value - 1 in the positive direction, it skips to the loading value at the next count pulse and the internal gate is closed automatically.
- For starting the counting process again, the internal gate has to be opened.
- The counter counts from the loading value.
- You can also count above the lower counting limit.


Fig. 3-45 Example of "Counting once" with main counting direction forwards

I/O compound modules - counter
C) Main counting direction backwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31-1)}\right.$ |
| Upper counting limit | $+2147483646\left(2^{31-1)}\right.$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts backwards from the loading value.
- If the counter reaches the final value +1 in the negative direction, it skips to the loading value at the next count pulse and the internal gate is closed automatically.
- For starting the counting process again, the internal gate has to be opened again.
- The counter starts from the loading value.
- You can also count above the upper counting limit.


Fig. 3-46 Example of "Counting once" with main counting direction backwards

## Counting periodically

A) No main counting direction:

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31-1)}\right.$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards from the loading value or backwards in the counting range.
- If there is an overflow or underflow at the respective counting limit, the counter skips to the loading value and continues to count from there.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-47 Example of "Counting periodically" and no main counting direction

I/O compound modules - counter
B) Main counting direction forwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31}-1\right)$ |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |

Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards from the loading value.
- If the counter reaches the final value - 1 in the positive direction, it skips to the loading value at the next positive count pulse and continues to count from there.
- You can also count above the lower counting limit.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-48 Example of "Counting periodically" and main counting direction forwards
C) Main counting direction backwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31-1)}\right.$ |
| Upper counting limit | $+2147483646\left(2^{31-1)}\right.$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts backwards from the loading value.
- If the counter reaches the final value +1 in the negative direction, it skips to the loading value at the next negative count pulse and continues to count from there.
- You can also count above the upper counting limit.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-49 Example of "Counting periodically" and main counting direction backwards

## 3 Product description

I/O compound modules - counter
One counter 32 bits, 24 V DC - EPM-S600

## Gate function

The counter is controlled via the "internal gate" (l-gate) which is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate).

- The SW gate is opened and closed via the user program (control word).
- With an edge change 0-1 at control word bit SW_GATE_SET, the SW gate opens (counter starts).
- With an edge change 0-1 at control word bit SW_GATE_RESET, the SW gate closes (counter stops).
- The HW gate is opened and closed via the digital gate input.

In the parameter setting the HW gate can be deactivated, so that it is only possible to activate the counter via the SW gate.

The following states have an impact on the l-gate:

| SW gate | HW gate | Has an impact on the I-gate |
| :---: | :---: | :---: |
| 0 | Edge change 0-1 | 0 |
| 1 | Edge change 0-1 | 1 |
| Edge change 0-1 | 1 | 1 |
| Edge change $0-1$ | 0 | 0 |
| Edge change $0-1$ | deactivated | 1 |

Via the parameter setting you define whether the gate is to cancel or interrupt the counting process.

- In the case of a cancelling gate function, the counting process starts after a restart from the loading value.
- In the case of an interrupting gate function, the counting process is continued at the current count value after a restart.
Action and response of the gate control:

| Function | SW gate | HW gate | Counter response |
| :--- | :---: | :---: | :---: |
| Gate control via SW gate, <br> cancelling | Edge change 0-1 | deactivated | Restart with loading value |
| Gate control via SW gate, <br> interrupting | Edge change 0-1 | Deactivated | Continuation |
| Gate control via SW/HW <br> gate, cancelling | Edge change 0-1 | 1 | Continuation |
| Gate control via SW/HW <br> gate, interrupting | Edge change 0-1 | Edge change 0-1 | Restart with loading value |

In particular in the case of the gate control via SW/HW gate in the "Counting once" operating mode: If the internal gate has been closed automatically, it can only be opened when the following conditions are met:

| SW gate | HW gate | Response of I-gate |
| :---: | :---: | :---: |
| 1 | Edge change 0-1 | 1 |
| Edge change 0-1 (after edge change <br> $0-1$ at HW gate) | 1 | 1 |

## Comparator

The comparison value is specified via the output area. If a comparison condition is met, bit STS_DO is set in the status word.

## Note!

Please note that bit STS_DO can only be triggered if bit STS_CTRL_DO is set in the status word.

By parameterisation you can define the response of the counter output:

- Output never switches
- Output switches if count value $\geq$ comparison value

As long as the count value is greater than or equals the comparison value, the output remains set

- Output switches if count value $\leq$ comparison value

As long as the count value is smaller than or equals the comparison value, the output remains set.

- Output switches at comparison value

Pulse at comparison value: If the counter reaches the comparison value, the output is set for the parameterised pulse duration. If the pulse duration is $=0$, the output is set until the comparison condition is no longer met. If you have set a main counting direction, the output is only switched from the main counting direction when the comparison value is reached.
Pulse duration: The pulse duration indicates for how long the output is to be set. It can be preselected in steps of 2.048 ms between 0 and 522.24 ms . The pulse duration starts with the setting of the respective digital output. The inaccuracy of the pulse duration is smaller than 2.048 ms . The pulse duration is not re-triggered if the comparison value has been exited and reached again during one pulse output.

## Note!

Together with bit STS_DO, bit STS_CMP is set in the status word. In contrast to bit STS_DO, however, it remains set until it is reset with RES_SET in the control word.

## 3 Product description

I/O compound modules - counter
One counter 32 bits, 24 V DC - EPM-S600

## Hysteresis

The hysteresis for instance serves to avoid frequent switching operations of the output and 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, overflow/underflow, and on the comparison value.
An active hysteresis remains active after the change. The new hysteresis range becomes active at the next hysteresis event.
In the following illustrations the response of the output at hysteresis 0 and hysteresis 3 for the corresponding conditions is represented:

## A) Function mode in the case of count value $\geq$ comparison value

When the comparison condition is met, hysteresis becomes active. When hysteresis is active, the comparison result remains unchanged until the count value exits the hysteresis range set. After the hysteresis range is exited, hysteresis is only activated again when the comparison conditions are met.

(1) Count value $\geq$ comparison value $\rightarrow$ Output is set and hysteresis is activated
(2) Exiting of the hysteresis range $\rightarrow$ Output is reset
(3) Count value $\geq$ comparison value $\rightarrow$ Output is set and hysteresis is activated
(4) Exiting of the hysteresis range, output remains set, since count value $\geq$ comparison value
(5) Count value < comparison value and hysteresis active $\rightarrow$ Output is reset
(6) Count value $\geq$ comparison value $\rightarrow$ Output is not set since hysteresis is activated
(7) Exiting of the hysteresis range, output is set, since count value $\geq$ comparison value
B) Function mode in the case of comparison value with zero pulse duration

When the comparison condition is met, hysteresis becomes active. When hysteresis is active, the comparison result remains unchanged until the count value exits the hysteresis range set. After the hysteresis range is exited, hysteresis is only activated again when the comparison conditions are met.

(1) Count value $=$ comparison value $\rightarrow$ Output is set and hysteresis is activated
(2) Exiting of the hysteresis range $\rightarrow$ Output is reset and count value < comparison value
(3) Count value $=$ comparison value $\rightarrow$ Output is set and hysteresis is activated
(4) Output is reset due to exiting of the hysteresis range and count value > comparison value
(5) Count value $=$ comparison value $\rightarrow$ Output is set and hysteresis is activated
(6) Count value $=$ comparison value and hysteresis active $\rightarrow$ Output remains set
(7) Exiting of the hysteresis range and count value >comparison value $\rightarrow$ Output is reset

## C) Function mode for comparison value with non-zero pulse duration

When the comparison condition is met, hysteresis becomes active and a pulse of the parameterised duration is released. As long as the count value is within the hysteresis range, no further pulse is released. With activation of the hysteresis, the counting direction is stored within the module. If the count value exits the hysteresis range in opposition to the counting direction stored, a pulse of the parameterised duration is released. When the hysteresis range is exited without a change in direction, no pulse is released.


[^3]
### 3.9.2

Two counters 32 bits, 24 V DC - EPM-S601
This module measures the pulses of up to two connected encoders and processes them according to the mode selected.

## Features

- 2 counters 32 bits (AB), invertible, DC 24 V
- Counting frequency max. 400 kHz
- Comparison value, set value, input filter
- Alarm and diagnostic function


## Overview



Fig. 3-50 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number

## Status displays



## 3 Product description

I/O compound modules - counter
Two counters 32 bits, 24 V DC - EPM-S601

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error ( $\mathbb{C}$ 249) |

## Terminals

Module terminals, spring terminals $B$

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
|  | 1 | Digital input, counter 1, "A"/"pulse" <br> Pulse input for counting signal or track A of an encoder for single, double, or quadruple evaluation |  |
|  | 2 | DC 24 V for encoder supply |  |
|  | 3 | GND |  |
|  | 4 | Digital input, counter 2, <br> "A"/"pulse" <br> Pulse input for counting signal or track A of an encoder for single, double, or quadruple evaluation | $0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
|  | 5 | Digital input, counter 1, "B"/"direction" <br> Direction signal or track B of an encoder (can be inverted via parameterisation) | (AWG 28 ... 16) |
|  | 6 | DC 24 V for encoder |  |
|  | 7 | GND |  |
| SL10002 | 8 | Digital input, counter 2, "B"/"direction" <br> Direction signal or track B of an encoder (can be inverted via parameterisation) |  |

Technical data

| EPM-S601: rated data |  |
| :---: | :---: |
| Module identifier | 2243 dec |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 75 mA |
| Power loss | 0.9 W |
| Digital inputs |  |
| Number of inputs | 4 |
| Cable length |  |
| shielded | 100 m |
| Load voltage |  |
| Nominal value | DC 20.4 ... 28.8 V |
| Current consumption from load voltage L+ | 15 mA (without load) |
| Input voltage |  |
| Nominal value | DC 20.4 ... 28.8 V |
| for signal "0" | DC $0 . . .5 \mathrm{~V}$ |
| for signal "1" | DC $15 \ldots 28.8 \mathrm{~V}$ |
| Input current |  |
| for signal "1" | 3 mA |
| 2-wire BERO |  |
| Connection possible | Yes |
| Max. permissible closed-circuit current | 0.5 mA |
| Input delay |  |
| from "0" to "1" | $0.8 \mu \mathrm{~s}$ |
| from " 1 " to "0" | $0.8 \mu \mathrm{~s}$ |
| Number of inputs which can be used simultaneously |  |
| horizontal structure | 4 |
| vertical structure | 4 |
| Input characteristic | IEC 61131, type 1 |
| Input data size | 12 bytes |
| Digital outputs |  |
| Output data size | 12 bytes |

## $3 \quad$ Product description

I/O compound modules - counter
Two counters 32 bits, 24 V DC - EPM-S601

EPM-S601: rated data

| Counter |  |
| :---: | :---: |
| Number of counters | 2 |
| Counter width | 32 bits |
| Frequency |  |
| Input frequency, max | 100 kHz |
| Counting frequency, max | 400 kHz |
| Operating mode |  |
| Incremental encoder | Possible |
| Pulse/direction | Possible |
| Pulse | Not possible |
| Frequency measurement | Not possible |
| Period duration measurement | Not possible |
| Connection |  |
| Gate connection | Not possible |
| Latch connection | Not possible |
| Reset connection | Not possible |
| Counter output | Not possible |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes, parameterisable |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## Parameterisable functions

Counting functions Description

Counting continuously
Counting once
Counting periodically

The counter counts from the loading value to the counting limit, then skips to the opposite counting limit and continues to count from there.

The counter counts once/periodically from the loading value in the specified counting range.

| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input "A/pulse" and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

## Counting continuously

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31-1}\right)$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts from the loading value in the counting range.
- If the counter reaches the upper counting limit when counting forwards, and if a further count pulse in positive direction occurs, the counter skips to the lower counting limit and continues to count from there.
- If the counter reaches the lower counting limit when counting backwards, and if a further negative count pulse occurs, the counter skips to the upper counting limit and continues to count from there.
- If maximum values are exceeded or minimum values are not reached, the status bits STS_OFLW or STS_UFLW are set. These bits remain set until they are reset with RES_SET in the control word. If enabled, additionally a process alarm is triggered.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-51 Example of "Counting continuously"

## Counting once

A) No main counting direction:

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31}-1\right)$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards once from the loading value or backwards in the preset counting range.
- If the counting limits are not reached or exceeded, the counter skips to the other counting limit, respectively, the internal gate is closed automatically and the status bits STS_OFLW or STS_UFLW are set. If enabled, a process alarm is effected.
- For stopping the counting process the internal gate has to be closed.
- For starting the counting process again, the internal gate has to be opened.
- If the gate control is interrupted, the counting process continues at the current counter content.
- If the gate control is cancelled, the counter starts from the loading value.


Fig. 3-52 Example of "Counting once", no main counting direction and with interrupting gate control


Fig. 3-53 Example of "Counting once", no main counting direction and with cancelling gate control
B) Main counting direction forwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31}-1\right)$ |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |

Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards from the loading value.
- If the counter reaches the final value - 1 in the positive direction, it skips to the loading value at the next count pulse and the internal gate is closed automatically.
- For starting the counting process again, the internal gate has to be opened.
- The counter counts from the loading value.
- You can also count above the lower counting limit.


Fig. 3-54 Example of "Counting once" with main counting direction forwards
C) Main counting direction backwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31}-1\right)$ |
| Upper counting limit | $+2147483646\left(2^{31-1)}\right.$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts backwards from the loading value.
- If the counter reaches the final value +1 in the negative direction, it skips to the loading value at the next count pulse and the internal gate is closed automatically.
- For starting the counting process again, the internal gate has to be opened again.
- The counter starts from the loading value.
- You can also count above the upper counting limit.


Fig. 3-55 Example of "Counting once" with main counting direction backwards

## Counting periodically

A) No main counting direction:

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31-1)}\right.$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards from the loading value or backwards in the counting range.
- If there is an overflow or underflow at the respective counting limit, the counter skips to the loading value and continues to count from there.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-56 Example of "Counting periodically" and no main counting direction
B) Main counting direction forwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31}-1\right)$ |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |

Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards from the loading value.
- If the counter reaches the final value - 1 in the positive direction, it skips to the loading value at the next positive count pulse and continues to count from there.
- You can also count above the lower counting limit.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-57 Example of "Counting periodically" and main counting direction forwards
C) Main counting direction backwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31-1)}\right.$ |
| Upper counting limit | $+2147483646\left(2^{31-1)}\right.$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts backwards from the loading value.
- If the counter reaches the final value +1 in the negative direction, it skips to the loading value at the next negative count pulse and continues to count from there.
- You can also count above the upper counting limit.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-58 Example of "Counting periodically" and main counting direction backwards

## Gate function

The counter is controlled via the "internal gate" (l-gate). For this counter, the l-gate is conform to the software gate (SW gate).
The SW gate is opened and closed via the user program (control word).

- With an edge change 0-1 at the control word bit SW_GATE_SET, the SW gate opens (counter starts).
- With an edge change 0-1 at the control word bit SW_GATE_RESET, the SW gate closes (counter stops).
The following states have an impact on the l-gate:

| SW gate | Response of I-gate |
| :---: | :---: |
| 0 | 0 |
| 1 | 1 |
| Edge change 0-1 | 1 |

Via the parameter setting you define whether the gate is to cancel or interrupt the counting process.

- In the case of a cancelling gate function, the counting process starts after a restart from the loading value.
- In the case of an interrupting gate function, the counting process is continued at the current count value after a restart.
Characteristic for the gate control via SW/HW gate in the "Counting once" operating mode: If the I-gate was closed automatically, it can only be opened by means of an edge change 0-1 at SW_GATE_SET.


## Comparator

The comparison value is specified via the output area. The comparison bit can be found in the status word under STS_COMP.

## Note!

Please note that bit STS_COMP can only be triggered if bit STS_CTRL_COMP is set in the status word.

Via the parameter setting you can define the response of the comparison bit:

- Comparison bit never switches
- Comparison bit is set if count value $\geq$ comparison value

As long as the count value is greater than or equals the comparison value, the comparison bit remains set

- Comparison bit is set if count value $\leq$ comparison value

As long as the count value is smaller than or equals the comparison value, the comparison bit remains set.

- Comparison bit is set if count value = comparison value

If the count value = comparison value, the comparison bit is set. The bit remains set until the comparison condition is no longer met. If you have set a main counting direction, the comparison bit is only set when the comparison value from the main counting direction is reached.

## Note!

Together with bit STS_COMP, bit STS_CMP is set in the status word. In contrast to bit STS_COMP, however, it remains set until it is reset with RES_SET in the control word.

## Hysteresis

The hysteresis for instance serves to avoid frequent switching operations of the output and 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, overflow/underflow, and on the comparison value.
An active hysteresis remains active after the change. The new hysteresis range becomes active at the next hysteresis event.
In the following illustrations the response of the output at hysteresis 0 and hysteresis 3 for the corresponding conditions is represented:

## A) Function mode in the case of count value $\geq$ comparison value

When the comparison condition is met, hysteresis becomes active. When hysteresis is active, the comparison result remains unchanged until the count value exits the hysteresis range set. After the hysteresis range is exited, hysteresis is only activated again when the comparison conditions are met.

(1) Count value $\geq$ comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(2) Exiting of the hysteresis range $\rightarrow$ Comparison bit is reset
(3) Count value $\geq$ comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(4) Exiting of the hysteresis range, comparison bit remains set, since count value $\geq$ comparison value
(5) Count value < comparison value and hysteresis active $\rightarrow$ Comparison bit is reset
(6) Count value $\geq$ comparison value $\rightarrow$ Comparison bit is not set since hysteresis is activated
(7) Exiting of the hysteresis range, comparison bit is set, since count value $\geq$ comparison value

## 3 Product description

I/O compound modules - counter

## B) Function mode for count value = comparison value

When the comparison condition is met, hysteresis becomes active. When hysteresis is active, the comparison result remains unchanged until the count value exits the hysteresis range set. After the hysteresis range is exited, hysteresis is only activated again when the comparison conditions are met.

(1) Count value = comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(2) Exiting of the hysteresis range $\rightarrow$ Comparison bit is reset and count value < comparison value

Count value = comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(4) Comparison bit is reset due to exiting of the hysteresis range and count value > comparison value
(5) Count value $=$ comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(6) Count value $=$ comparison value and hysteresis active $\rightarrow$ Comparison bit remains set
(7) Exiting of the hysteresis range and count value > comparison value $\rightarrow$ Comparison bit is reset

One counter 32 bits, 5 V DC - EPM-S602
This module measures the pulses of one connected encoder and processes them according to the mode selected.

## Features

- 1 counter 32 bits (AB), invertible, DC 5 V (differential signal)
- Counting frequency max. 2 MHz
- Comparison value, set value, input filter
- Alarm and diagnostic function


## Overview



Fig. 3-59 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number

## Status displays



## 3 Product description

I/O compound modules - counter
One counter 32 bits, 5 V DC - EPM-S602

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error ( $\mathbb{C}$ 249) |

## Terminals



## Technical data

| EPM-S602: rated data | $2242_{\text {dec }}$ |
| :--- | :--- |
| Module identifier |  |
| Current consumption/power loss 70 mA <br> Current consumption from backplane bus 0.85 W <br> Power loss 0 <br> Digital inputs 100 m <br> Number of inputs <br> Cable length <br> shielded  <br> Load voltage <br> Nominal value <br> Current consumption from load voltage <br> L+ 20 mA (without load) |  |

## I/O compound modules - counter

 One counter 32 bits, 5 V DC - EPM-S602| EPM-S602: rated data |  |
| :---: | :---: |
| Input voltage |  |
| for signal "0" | Differential signal RS422 |
| for signal "1" | Differential signal RS422 |
| Input resistance | $120 \Omega$ |
| Input delay |  |
| from "0" to "1" | $0.8 \mu \mathrm{~s}$ |
| from " 1 " to "0" | $0.8 \mu \mathrm{~s}$ |
| Input data size | 8 bytes |
| Digital outputs |  |
| Output data size | 10 bytes |
| Counter |  |
| Number of counters | 1 |
| Counter width | 32 bits |
| Frequency |  |
| Input frequency, max | 500 kHz |
| Counting frequency, max | 2 MHz |
| Operating mode |  |
| Incremental encoder | Possible |
| Pulse/direction | Possible |
| Pulse | Not possible |
| Frequency measurement | Not possible |
| Period duration measurement | Not possible |
| Connection |  |
| Gate connection | Not possible |
| Latch connection | Not possible |
| Reset connection | Possible |
| Counter output | Not possible |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes, parameterisable |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## Parameterisable functions

| Counting functions | Description |
| :---: | :---: |
| Counting continuously | 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 input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input "A/pulse" and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

## Counting continuously

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31-1}\right)$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts from the loading value in the counting range.
- If the counter reaches the upper counting limit when counting forwards, and if a further count pulse in positive direction occurs, the counter skips to the lower counting limit and continues to count from there.
- If the counter reaches the lower counting limit when counting backwards, and if a further negative count pulse occurs, the counter skips to the upper counting limit and continues to count from there.
- If maximum values are exceeded or minimum values are not reached, the status bits STS_OFLW or STS_UFLW are set. These bits remain set until they are reset with RES_SET in the control word. If enabled, additionally a process alarm is triggered.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-60 Example of "Counting continuously"

## 3 Product description

I/O compound modules - counter
One counter 32 bits, 5 V DC - EPM-S602

## Counting once

A) No main counting direction:

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31}-1\right)$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards once from the loading value or backwards in the preset counting range.
- If the counting limits are not reached or exceeded, the counter skips to the other counting limit, respectively, the internal gate is closed automatically and the status bits STS_OFLW or STS_UFLW are set. If enabled, a process alarm is effected.
- For stopping the counting process the internal gate has to be closed.
- For starting the counting process again, the internal gate has to be opened.
- If the gate control is interrupted, the counting process continues at the current counter content.
- If the gate control is cancelled, the counter starts from the loading value.


Fig. 3-61 Example of "Counting once", no main counting direction and with interrupting gate control


Fig. 3-62 Example of "Counting once", no main counting direction and with cancelling gate control
B) Main counting direction forwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31}-1\right)$ |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |

Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards from the loading value.
- If the counter reaches the final value - 1 in the positive direction, it skips to the loading value at the next count pulse and the internal gate is closed automatically.
- For starting the counting process again, the internal gate has to be opened.
- The counter counts from the loading value.
- You can also count above the lower counting limit.


Fig. 3-63 Example of "Counting once" with main counting direction forwards
C) Main counting direction backwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31-1)}\right.$ |
| Upper counting limit | $+2147483646\left(2^{31-1)}\right.$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts backwards from the loading value.
- If the counter reaches the final value +1 in the negative direction, it skips to the loading value at the next count pulse and the internal gate is closed automatically.
- For starting the counting process again, the internal gate has to be opened again.
- The counter starts from the loading value.
- You can also count above the upper counting limit.


Fig. 3-64 Example of "Counting once" with main counting direction backwards

## Counting periodically

A) No main counting direction:

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31}-1\right)$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards from the loading value or backwards in the counting range.
- If there is an overflow or underflow at the respective counting limit, the counter skips to the loading value and continues to count from there.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-65 Example of "Counting periodically" and no main counting direction

I/O compound modules - counter
B) Main counting direction forwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31}-1\right)$ |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |

Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts forwards from the loading value.
- If the counter reaches the final value - 1 in the positive direction, it skips to the loading value at the next positive count pulse and continues to count from there.
- You can also count above the lower counting limit.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-66 Example of "Counting periodically" and main counting direction forwards
C) Main counting direction backwards:

| Limits | Counting range |
| :--- | :--- |
| Final value | $-2147483646\left(-2^{31}+1\right)$ to $+2147483646\left(2^{31-1)}\right.$ |
| Upper counting limit | $+2147483646\left(2^{31-1)}\right.$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter counts backwards from the loading value.
- If the counter reaches the final value +1 in the negative direction, it skips to the loading value at the next negative count pulse and continues to count from there.
- You can also count above the upper counting limit.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-67 Example of "Counting periodically" and main counting direction backwards

## Gate function

The counter is controlled via the "internal gate" (l-gate). For this counter, the l-gate is conform to the software gate (SW gate).
The SW gate is opened and closed via the user program (control word).

- With an edge change 0-1 at the control word bit SW_GATE_SET, the SW gate opens (counter starts).
- With an edge change 0-1 at the control word bit SW_GATE_RESET, the SW gate closes (counter stops).
The following states have an impact on the l-gate:

| SW gate | Response of I-gate |
| :---: | :---: |
| 0 | 0 |
| 1 | 1 |
| Edge change 0-1 | 1 |

Via the parameter setting you define whether the gate is to cancel or interrupt the counting process.

- In the case of a cancelling gate function, the counting process starts after a restart from the loading value.
- In the case of an interrupting gate function, the counting process is continued at the current count value after a restart.
Characteristic for the gate control via SW/HW gate in the "Counting once" operating mode: If the I-gate was closed automatically, it can only be opened by means of an edge change 0-1 at SW_GATE_SET.


## Comparator

The comparison value is specified via the output area. The comparison bit can be found in the status word under STS_COMP.

## Note!

Please note that bit STS_COMP can only be triggered if bit STS_CTRL_COMP is set in the status word.

Via the parameter setting you can define the response of the comparison bit:

- Comparison bit never switches
- Comparison bit is set if count value $\geq$ comparison value

As long as the count value is greater than or equals the comparison value, the comparison bit remains set

- Comparison bit is set if count value $\leq$ comparison value

As long as the count value is smaller than or equals the comparison value, the comparison bit remains set.

- Comparison bit is set if count value = comparison value

If the count value = comparison value, the comparison bit is set. The bit remains set until the comparison condition is no longer met. If you have set a main counting direction, the comparison bit is only set when the comparison value from the main counting direction is reached.

Note!
Together with bit STS_COMP, bit STS_CMP is set in the status word. In contrast to bit STS_COMP, however, it remains set until it is reset with RES_SET in the control word.

## 3 Product description

I/O compound modules - counter
One counter 32 bits, 5 V DC - EPM-S602

## Hysteresis

The hysteresis for instance serves to avoid frequent switching operations of the output and 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, overflow/underflow, and on the comparison value.
An active hysteresis remains active after the change. The new hysteresis range becomes active at the next hysteresis event.
In the following illustrations the response of the output at hysteresis 0 and hysteresis 3 for the corresponding conditions is represented:

## A) Function mode in the case of count value $\geq$ comparison value

When the comparison condition is met, hysteresis becomes active. When hysteresis is active, the comparison result remains unchanged until the count value exits the hysteresis range set. After the hysteresis range is exited, hysteresis is only activated again when the comparison conditions are met.

(1) Count value $\geq$ comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(2) Exiting of the hysteresis range $\rightarrow$ Comparison bit is reset
(3) Count value $\geq$ comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(4) Exiting of the hysteresis range, comparison bit remains set, since count value $\geq$ comparison value
(5) Count value < comparison value and hysteresis active $\rightarrow$ Comparison bit is reset
(6) Count value $\geq$ comparison value $\rightarrow$ Comparison bit is not set since hysteresis is activated
(7) Exiting of the hysteresis range, comparison bit is set, since count value $\geq$ comparison value

## B) Function mode for count value = comparison value

When the comparison condition is met, hysteresis becomes active. When hysteresis is active, the comparison result remains unchanged until the count value exits the hysteresis range set. After the hysteresis range is exited, hysteresis is only activated again when the comparison conditions are met.

(1) Count value $=$ comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(2) Exiting of the hysteresis range $\rightarrow$ Comparison bit is reset and count value < comparison value

Count value = comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(4) Comparison bit is reset due to exiting of the hysteresis range and count value > comparison value
(5) Count value = comparison value $\rightarrow$ Comparison bit is set and hysteresis is activated
(6) Count value $=$ comparison value and hysteresis active $\rightarrow$ Comparison bit remains set
(7) Exiting of the hysteresis range and count value > comparison value $\rightarrow$ Comparison bit is reset

## 3 Product description

I/O compound modules - counter
Two counters 32 bits, 24 V DC - EPM-S603

### 3.9.4

Two counters 32 bits, 24 V DC - EPM-S603
This module measures the pulses of up to two connected encoders and processes them according to the mode selected.

## Features

- 2 counters 32 bits, DC 24 V
- Counting frequency max. 400 kHz
- Input filter
- Diagnostic function


## Overview



Fig. 3-68 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
|  | 3 | A1 | Green | On: Digital input 1, counter 1, "A1"/"pulse" triggered |
|  | 4 | B1 |  | On: Digital input 5, counter 1, "B1"/"direction" triggered |
|  | 5 | A2 |  | On: Digital input 4, counter 2, "A2"/"pulse" triggered |
|  | 6 | B2 |  | On: Digital input 8, counter 2, "B2"/"direction" triggered |
|  | 7 | - | - | Not assigned |
|  | 8 |  |  |  |
|  | 9 |  |  |  |
|  | 10 |  |  |  |


| Messages of the status LEDs RUN and MF |  |  |
| :---: | :---: | :--- |
| RUN | MF | Meaning |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | On | Module reports error <br> Bus communication not possible |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error ( $\mathbf{\square}$ 249) |
| BLINKING (2 Hz) | BLINKING (2 Hz) | Module reports parameterisation error <br> Bus communication is OK |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
| SLIOOO2 | 1 | Digital input, counter 1, "A"/"pulse" <br> Pulse input for counting signal or track A of an encoder for single, double, or quadruple evaluation |  |
|  | 2 | DC 24 V for encoder supply |  |
|  | 3 | GND |  |
|  | 4 | Digital input, counter 2, "A"/"pulse" <br> Pulse input for counting signal or track A of an encoder for single, double, or quadruple evaluation |  |
|  | 5 | Digital input, counter 1, "B"/"direction" Direction signal or track B of an encoder (can be inverted via parameterisation) |  |
|  | 6 | DC 24 V for encoder |  |
|  | 7 | GND |  |
|  | 8 | Digital input, counter 2, "B"/"direction" <br> Direction signal or track B of an encoder (can be inverted via parameterisation) |  |

## $3 \quad$ Product description

I/O compound modules - counter
Two counters 32 bits, 24 V DC - EPM-S603

Technical data

| EPM-S603: rated data |  |
| :---: | :---: |
| Module identifier | 2244 ${ }_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 75 mA |
| Power loss | 0.9 W |
| Digital inputs |  |
| Number of inputs | 4 |
| Cable length |  |
| shielded | 100 m |
| Load voltage |  |
| Nominal value | DC 20.4 ... 28.8V |
| Current consumption from load voltage L+ | 15 mA (without load) |
| Input voltage |  |
| Nominal value | DC 20.4 ... 28.8 V |
| for signal "0" | DC $0 . . .5 \mathrm{~V}$ |
| for signal "1" | DC $15 . .28 .8 \mathrm{~V}$ |
| Input current |  |
| for signal "1" | 3 mA |
| 2-wire BERO |  |
| Connection possible | Yes |
| Max. permissible closed-circuit current | 0.5 mA |
| Input delay |  |
| from "0" to "1" | $0.8 \mu \mathrm{~s}$ |
| from " 1 " to " 0 " | $0.8 \mu \mathrm{~s}$ |
| Number of inputs which can be used simultaneously |  |
| horizontal structure | 4 |
| vertical structure | 4 |
| Input characteristic | IEC 61131, type 1 |
| Input data size | 12 bytes |
| Digital outputs |  |
| Output data size | 4 bytes |

EPM-S603: rated data

| Counter |  |
| :---: | :---: |
| Number of counters | 2 |
| Counter width | 32 bits |
| Frequency |  |
| Input frequency, max | 100 kHz |
| Counting frequency, max | 400 kHz |
| Operating mode |  |
| Incremental encoder | Possible |
| Pulse / direction | Possible |
| Pulse | Not possible |
| Frequency measurement | Not possible |
| Period duration measurement | Not possible |
| Connection |  |
| Gate connection | Not possible |
| Latch connection | Not possible |
| Reset connection | Not possible |
| Counter output | Not possible |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound modules - counter
Two counters 32 bits, 24 V DC - EPM-S603

## Parameterisable functions

| Counting functions | Description <br> Counting <br> continuously |
| :--- | :--- |
| The counter counts from 0 to the counting limit, then skips to the opposite counting limit <br> and continues to count from there. |  |
| Signal evaluation <br> Single rotary <br> transducer | Description |
| Double rotary <br> transducer | Connection to input "A/pulse" and "B/direction" |
| Quadruple rotary <br> transducer | Pulse at input "A/pulse" and direction at "B/direction" |
| Direction | Description |
| 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). |  |
| Gate function |  |

## Counting continuously

| Limits | Counting range |
| :--- | :--- |
| Lower counting limit | $-2147483648\left(-2^{31}\right)$ |
| Upper counting limit | $+2147483647\left(2^{31-1}\right)$ |

## Function:

- For starting the counting process the internal gate has to be opened.
- The counter always counts from 0 .
- If the counter reaches the upper counting limit when counting forwards, and if a further count pulse in positive direction occurs, the counter skips to the lower counting limit and continues to count from there.
- If the counter reaches the lower counting limit when counting backwards, and if a further negative count pulse occurs, the counter skips to the upper counting limit and continues to count from there.
- If values are exceeded or not reached, the status bits STS_OFLW or STS_UFLW are set. These bits remain set until they are reset with RES_SET in the control word.
- For stopping the counting process the internal gate has to be closed.


Fig. 3-69 Example of "Counting continuously"

## Gate function

The counter is controlled via the "internal gate" (l-gate). For this counter, the l-gate is conform to the software gate (SW gate).
The SW gate is opened and closed via the user program (control word).

- With an edge change 0-1 at the control word bit SW_GATE_SET, the SW gate opens (counter starts from count value).
- With an edge change 0-1 at the control word bit SW_GATE_RESET, the SW gate closes (counter stops).
The following states have an impact on the l-gate:

| SW gate | Response of l-gate |
| :---: | :---: |
| 0 | 0 |
| 1 | 1 |
| Edge change $0-1$ | 1 |

## $3.10 \quad$ I/O compound modules - encoder evaluation

### 3.10.1 SSI - EPM-S604

The module is an SSI interface for direct connection to an SSI encoder.
During the parameterisation you can adapt the module to the corresponding SSI encoder.

## Features

- 1xSSI for absolute value encoders with 8 ... 32 bits
- Connection via differential signal (RS422)
- Clock output for master operating mode
- Clock input for monitoring operation
- Integrated converter for Gray/Dual
- Cyclic encoder value measurement
- Scaling of the encoder value, i. e. subsequent bits are removed


## Overview



Fig. 3-70
Elements and circuit diagram

| A | Displays for module status |
| :--- | :--- |
| B | Terminals |
| $1 \ldots 8$ | Connection number |

## 3 Product description

I/O compound modules - encoder evaluation

## Status displays

## Module status LEDs A

| View | Pos. | Designation | Colour | Explanation |
| :---: | :---: | :---: | :---: | :---: |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| - | 3 | CO | Green | Clock OUT activity |
| - | 4 | DI |  | Data IN activity |
|  | 5 | Cl |  | Clock IN activity |
|  | 6 | - | - | Not assigned |
|  | 7 |  |  |  |
|  | 8 |  |  |  |
|  | 9 |  |  |  |
|  | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■ 249) |

## Terminals

Module terminals, spring terminals $B$

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
| SLIOOO2 | 1 | Digital output "Clock OUT+" Differential output for Clock OUT |  |
|  | 2 | DC 24 V for encoder |  |
|  | 3 | GND |  |
|  | 4 | Digital input "Clock IN+" Differential input for Clock IN |  |
|  | 5 | Digital output "Clock OUT-" Differential output for Clock OUT | $\left(1 / / / \frac{0.08 \ldots 1.5 \mathrm{~mm}^{2}}{(\text { AWG } 28 \ldots 16)}\right.$ |
|  | 6 | Digital input "Data IN+" Differential input for Data IN |  |
|  | 7 | Digital input "Data IN-" Differential input for Data IN |  |
|  | 8 | Digital input "Clock IN-" Differential input for Clock IN |  |

## Technical data

| EPM-S604: rated data |  |
| :---: | :---: |
| Module identifier | 2497 ${ }_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 70 mA |
| Power loss | 1 W |
| SSI |  |
| Number of terminals for SSI encoders | 1 |
| Physics | RS422 |
| Encoder |  |
| Supply voltage | DC 24 V |
| Bit length | 8 ... 32 bits |
| Cycle | 125 kHz ... 2 MHz |
| Operating modes | Master mode, monitoring operation |
| Coding | Binary, gray |
| Scaling | Yes, parameterisable |
| Output data size | 6 bytes |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | Yes, parameterisable |
| Process alarm | No |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Yes, parameterisable |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound modules - encoder evaluation

## Operating modes

In the "Master mode"the module is directly connected to an SSI encoder and also supplies it. For reading out the encoder, the module provides a cycle to the encoder and presents the received data current in the process image.


Fig. 3-71 "Master mode" block diagram
In "Monitoring operation"the monitoring module is operated on an SSI encoder passively to the master module. During this operating mode the SSI encoder has to be supplied via the master module. During operation, the module monitors the signals of the SSI telegram and provides the data flow in the process image. In this case again, the module has to be adapted to the corresponding encoder via parameterisation. The "Baud rate" parameter is irrelevant.


Fig. 3-72 "Monitoring operation" block diagram

I/O compound module - pulse width modulation 2 digital outputs with PWM functionality - EPM-S620

### 3.11 I/O compound module - pulse width modulation

### 3.11.1 2 digital outputs with PWM functionality - EPM-S620

This module has two output channels with PWM functionality (PulseWidthModulation).
By specifying time parameters, you can forward a pulse train to the desired output with the mark-to-space ratio you want.

## Features

- 2 PWM outputs, can be switched between "push/pull" and "highside"
- 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.
- 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.
- Variable period and scanning ratio
- LEDs show the switching states of the digital outputs


## Overview



Fig. 3-73 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number
(0) ... (8) Bit number in bit presentation

## 3 Product description

I/O compound module - pulse width modulation
2 digital outputs with PWM functionality - EPM-S620

## Status displays

Module status LEDs ©


Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■24)On |

## Terminals

| Module terminals, spring terminals B |  |  |  |
| :---: | :---: | :---: | :---: |
| View | Designation | Explanation | Terminal data |
|  | 1 | Digital output DO1 |  |
|  | 2 | 24 V DC |  |
|  | 3 | GND |  |
|  | 4 | Not assigned |  |
|  | 5 | Digital output DO2 |  |
|  | 6 | 24 V DC |  |
|  | 7 | GND |  |
|  | 8 | Not assigned |  |

## Technical data

| EPM-S620: Rated data | $2305_{\mathrm{dec}}$ |
| :--- | :--- |
| Module identifier |  |
| Current consumption/power loss | 0.95 mA |
| Current consumption from backplane bus |  |
| Power loss | 2 |
| Digital outputs |  |
| Number of outputs | 1000 m |
| Cable length | 600 m |
| shielded |  |
| unshielded |  |

## I/O compound module - pulse width modulation 2 digital outputs with PWM functionality - EPM-S620

| EPM-S620: Rated data |  |
| :---: | :---: |
| Load voltage |  |
| Nominal value | DC 24 (DC 20.4 ... 28.8 V ) |
| Current consumption from load voltage L+ | 15 mA (without load) |
| Total current |  |
| for each group, horizontal structure, $40^{\circ} \mathrm{C}$ | 1 A |
| for each group, horizontal structure, $60^{\circ} \mathrm{C}$ | 1 A |
| for each group, vertical structure | 1 A |
| Output current |  |
| for "1" signal, nominal value | 0.5 A |
| Output delay |  |
| from "0" to "1" | Max. 100 ns |
| from " 1 " to "0" | Max. 100 ns |
| Lamp load | 10 W |
| Parallel switching of outputs |  |
| for redundant control | Not possible |
| for power increase | Not possible |
| Control of a digital input | Possible |
| Switching frequencies |  |
| for ohmic loads | Max. 40 Hz |
| for inductive loads | Max. 40 Hz |
| for lamp loads | Max. 40 Hz |
| Limitation (internal) of the inductive breaking voltage | L+ (-52 V) |
| Short circuit protection of the output | Electronically; only highside |
| Operating threshold of the protection system | 2.5 A |
| Input data size | 4 bytes |
| Output data size | 12 bytes |
| Status, alarm, diagnostics |  |
| Status display | Green LEDs per channel |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | Possible |
| Module status | Green LED |
| Module error display | Red LED |
| Channel error display | None |
| Electrical isolation |  |
| Between the channels and the backplane bus | Yes |
| Insulation checked with | DC 500 V |

## 3 Product description

I/O compound module - pulse width modulation
2 digital outputs with PWM functionality - EPM-S620

## Functional principle

Both electronic module outputs are supported by the PWM function (pulse width modulation). Specifying the time parameters produces a pulse train with the desired pulse/pause ratio.


Fig. 3-74 Pulse/pause ratio for output signal
PWM x Signal level of digital output, PWM 1 or PWM 2
(A) Period

B Pulse duration
C Break duration
The pulse pause is obtained by setting parameters for period and pulse duration. The settings determine the pulse/pause ratio.

### 3.12 I/O compound modules - communication

### 3.12.1 RS232 interface - EPM-S640

## Note!

This module can only be actuated on a bus coupler module with HW version 1D or higher.

This module features an RS232 interface for access to field devices with an RS232 interface. For access to devices with an RS232 interface, the protocol of the field device must be implemented in the control in each case.
The following handling blocks are provided:

| IPC / control | Programming via | Bus system | Handling block |
| :--- | :--- | :--- | :--- |
| Lenze Industrial PC <br> $\bullet$ EL 1800-9800 | PLC Designer V2 | - CAN | Data reception: |
| - CS 5800-9800 <br> - CPC 2800 |  | - EtherCAT | L_IO1000_EPMS640/650_RS232/485Write |
| L-force Controller <br> $3200 ~ C ~$ | PLC Designer V3 |  |  |

Further information on the subject ...

- "Serial process interfacing with I/O compound modules", see appendix.
- "Parameterisation", see in the corresponding chapter for the fieldbuses.
- "Handling blocks", see documentation for the PLC Designer.


## Features

- Serial RS232 interface (isolated towards the backplane bus)
- Logic states as voltage level
- Transmission speed 150 bps ... 115.2 kbps
- Data transfer up to a distance of 15 m
- Hardware handshake (RTS/CTS)
- Protocols
- ASCII
- STX/ETX
- 3964(R)
- Up to 250 frames (1024 bytes of receive or transmit buffer)
- Character delay time can be parameterised in ms grid
- Parameter setting via 17 bytes of parameter data
- Modem Signals Management DTR-DSR-DCD


## 3 Product description

I/O compound modules - communication
RS232 interface - EPM-S640

## Overview



Fig. 3-75 Elements and circuit diagram
(A) Displays for module status

B Terminals
1... 8 Connection number

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| $1-$ | 3 | TxD | Green | On: Transmit data |
|  | 4 | RxD | Green | On: Receive data |
| $\left.\left\lvert\, \begin{array}{l} 0 \\ \square \\ \square \\ 0 \end{array}\right.\right]$ | 5 | IF | Red | Blinking: Line interruption, overflow, parity error, or character frame error |
|  | 6 |  |  |  |
| $\begin{array}{\|l\|l\|l\|l\|} \hline \\ 0 \end{array}$ | 7 |  |  |  |
| $10-\square$ | 8 | - | - | Not assigned |
|  | 9 |  |  |  |
| SL10001 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (C) 249) |

## Terminals

Module terminals, spring terminals B

| View | Designation | Explanation | Terminal data |
| :--- | :--- | :--- | :--- |

Note!
RI (Ring indicator) - modem ring is not used!

## Wiring



Fig. 3-76 Without hardware handshake


Fig. 3-77 With hardware handshake

## 3 Product description

I/O compound modules - communication
RS232 interface - EPM-S640

Technical data

| Rated data EPM-S640 |  |
| :--- | :--- |
| Module identifier | $3585_{\text {dec }}$ |
| Current consumption/power loss |  |
| Current consumption from backplane bus | 100 mA |
| Current consumption from load voltage L+ | 10 mA (without load) |
| Power loss | 1 W |
| Status, alarm, diagnostics | Yes |
| Status display | Yes, parameterisable |
| Alarms | No |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Possible |
| Diagnostic information can be read out | Green LED |
| Module status | Red LED |
| Module error display | Red LED |
| Channel error display | $20 / 60$ bytes (selectable) |
| Communication | RS232 |
| Interface | To the backplane bus |
| Electrical isolation | Yes |
| PtP communication | 150 bps ... 115.2 kbps |
| Transmission speed | 15 m |
| Cable length, max. | $8 / 20 / 60$ bytes (selectable) |
| Input data for bus coupler module | $20 / 60$ bytes (selectable) |
| EPM-S110 | 80 bytes |
| EPM-S120 EPM-S140 <br> EPM-S130, EPM-S150, EPM-S160  |  |


| Rated data EPM-S640 |  |
| :---: | :---: |
| Point-to-point protocols |  |
| ASCII |  |
| Telegram length, max. | 1024 bytes |
| Character delay time | 0 ... 65535 in ms steps ( $0=3$-fold character time) |
| Flow control | None, hardware, XON/XOFF |
| Number of frames to be buffered, max. | 250 |
| Recognition of the end of a frame | after the character delay time has elapsed |
| STX/ETX |  |
| Telegram length, max. | 1024 bytes |
| Character delay time TMO | 0 ... 65535 in ms steps ( $0=3$-fold character time) |
| Flow control | None, hardware, XON/XOFF |
| Number of frames to be buffered, max. | 250 |
| Recognition of the end of a frame | by parameterised final character |
| Number of initial characters | 0 ... 2 (characters can be parameterised) |
| Number of final characters | 0 ... 2 (characters can be parameterised) |
| 3964, 3964R |  |
| Telegram length, max. | 1024 bytes |
| Block check times | Only 3964R |
| Priority | LOW/HIGH |
| Character delay time | 0 ... 255 in 20 ms steps ( $0=3$-fold character time) |
| Character delay time acknowledgement time | 0 ... 255 in 20 ms steps ( $0=3$-fold character time) |
| Number of establishment tests | 0 ... 255 |
| Number of transmission tests | 1 ... 255 |

## 3 Product description

I/O compound modules - communication
RS485 interface - EPM-S650

### 3.12.2

## RS485 interface - EPM-S650

This module can be used to communicate with field devices via an RS422 or RS485 interface. For access to devices with an RS422/RS485 interface, the protocol of the field device must be implemented in the control system in each case.
The following handling blocks are provided:

| IPC / control | Programming via | Bus system | Handling block |
| :--- | :--- | :--- | :--- |
| Lenze Industrial PC | PLC Designer V2 | - CAN | Data reception: |
| $\bullet$ EL 1800-9800 |  | - EtherCAT | L_IO1000_EPMS640/650_RS232/485Write |
| - CS 5800-9800 |  | - PROFIBUS | Data transmission: |
| - CPC 2800 |  |  | L_IO1000_EPMS640/650_RS232/485Read |
| L-force Controller <br> 3200 C | PLC Designer V3 |  |  |

Further information on the subject ...

- "Serial process interfacing with I/O compound modules", see appendix.
- "Parameterisation", see in the corresponding chapter for the fieldbuses.
- "Handling blocks", see documentation for the PLC Designer.


## Features

- Serial RS422/RS485 interface (isolated from the backplane bus)
- Transmission speed 150 bps ... 115.2 kbps
- Protocols
- ASCII
- STX/ETX
- 3964(R)
- Up to 250 frames (1024 bytes of receive or transmit buffer)
- Character delay time can be parameterised in ms grid
- Parameter setting via 19 bytes of parameter data


## 1 Note!

More information can be found in the "Communication" chapter.

## Overview



SLIOS650
Fig. 3-78 Elements and circuit diagram
(A) Displays for module status

B Terminals
1... 8 Connection number

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 | RUN | Green | On: Module is ready for operation |
|  | 2 | MF | Red | On: Module error (see table below) |
| $1-5$ | 3 | TxD |  | On: Transmit data |
|  | 4 | RxD | Green | On: Receive data |
| $\left[\begin{array}{l} 0 \\ 0 \\ 0 \\ 0 \end{array}\right.$ | 5 | IF | Red | Blinking: Line interruption, overflow, parity error, or character frame error |
| - | 6 |  |  |  |
|  | 7 |  |  |  |
| $10-\square$ | 8 | - | - | Not assigned |
|  | 9 |  |  |  |
| SL10001 | 10 |  |  |  |

Messages of the status LEDs RUN and MF

| RUN | MF | Meaning |
| :---: | :---: | :--- |
| On | Off | Module status OK <br> Bus communication is OK |
| On | On | Module reports error <br> Bus communication is OK |
| Off | Module reports error <br> Bus communication not possible |  |
| Off | Off | Error in the bus supply voltage |
| Blinking | Blinking | Configuration error (■49) 249) |

## 3 Product description

I/O compound modules - communication RS485 interface - EPM-S650

## Terminals



## Wiring

- Logic states as a voltage difference between 2 twisted cores
- Serial bus connection
- Full duplex (RS422 four-wire operation)
- Half duplex (RS485 two-wire operation)
- Cable length: 250 m for 115.2 kbps ... 1200 m for 19.2 kbps
- Data transfer rate: Max. 115.2 kbps


SLIO650a
Fig. 3-79 RS422 wiring

| EPM-S650 |
| :---: | :---: | :---: |
| TxD-N (A) 5 |
| TxD-P (B) |
| RxD-N (A) |
| 6 |

Fig. 3-80 RS485 wiring
*) A bridge between pin 8 and 4 in the EPM-S650 activates a $120 \Omega$ terminating resistor between RxD-P (pin 2) and RxD-N (pin 6).

## 3 Product description

I/O compound modules - communication
RS485 interface - EPM-S650

## Parameterisable idle level:

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.
The receiver is connected as follows:

| Parameter | Description |  |
| :--- | :--- | :--- |
| None | No pre-assignment of the receive <br> path. This setting is only advisable <br> for special drivers that are provided <br> with a bus capability. |  |
| Signal $\mathrm{R}(\mathrm{A}) 5 \mathrm{~V}$ (open circuit detection) <br> Signal $\mathrm{R}(\mathrm{B}) \mathrm{OV}$ | If this pre-assignment is set, open <br> circuit detection is possible in full <br> duplex operation (RS422). |  |
| Signal $\mathrm{R}(\mathrm{A})$ <br> Signal $\mathrm{R}(\mathrm{B}) 5 \mathrm{~V}$ | This pre-assignment corresponds to <br> the idle state (no transmitter active) <br> in half duplex operation under <br> RS485. Open circuit detection is not <br> possible. |  |

## Technical data

| EPM-S650: Rated data | $2625_{\text {dec }}$ |
| :--- | :--- |
| Module identifier |  |
| Current consumption/power loss | 100 mA |
| Current consumption from backplane bus | 10 mA (without load) |
| Current consumption from load voltage L+ | 10 W |
| Power loss |  |
| Status, alarm, diagnostics | Yes |
| Status display | Yes, parameterisable |
| Alarms | No |
| Process alarm | Yes, parameterisable |
| Diagnostic alarm | Yes, parameterisable |
| Diagnostic function | Possible |
| Diagnostic information can be read out | Green LED |
| Module status | Red LED |
| Module error display | Red LED |
| Channel error display |  |
| Communication | RS422 and RS485 |
| Interface | To the backplane bus |
| Electrical isolation | Yes |
| PtP communication | No |
| 20 mA / TTY | 150 bps ... 115.2 kbps |
| Transmission speed | 1200 m |
| Cable length, max. |  |


| EPM-S650: Rated data |  |
| :---: | :---: |
| Point-to-point protocols |  |
| ASCII |  |
| Telegram length, max. | 1024 bytes |
| Character delay time | 0 ... 65535 in ms steps ( $0=3$-fold character time) |
| Flow control | None, hardware, XON/XOFF |
| Number of frames to be buffered, max. | 250 |
| Recognition of the end of a frame | after the character delay time has elapsed |
| STX/ETX |  |
| Telegram length, max. | 1024 bytes |
| Character delay time TMO | 0 ... 65535 in ms steps ( $0=3$-fold character time) |
| Flow control | None, hardware, XON/XOFF |
| Number of frames to be buffered, max. | 250 |
| Recognition of the end of a frame | by parameterised final character |
| Number of initial characters | 0 ... 2 (characters can be parameterised) |
| Number of final characters | 0 ... 2 (characters can be parameterised) |
| 3964, 3964R |  |
| Telegram length, max. | 1024 bytes |
| Block check times | Only 3964R |
| Priority | LOW/HIGH |
| Character delay time | 0 ... 255 in 20 ms steps ( $0=3$-fold character time) |
| Character delay time acknowledgement time | 0 ... 255 in 20 ms steps ( $0=3$-fold character time) |
| Number of establishment tests | 0 ... 255 |
| Number of transmission attempts | 1 ... 255 |

## 3 Product description

Power supply modules
I/O supply - EPM-S701

### 3.13

Power supply modules

### 3.13.1 I/O supply - EPM-S701

If the bus coupler main supply does not have sufficient power to feed the I/O level, this module can be used.

## Features

- Incoming supply for I/O supply
- Cable protection by monitored internal fuse


## Overview



Fig. 3-81 Elements and circuit diagram
(A) Displays for module status

B Terminals
1 ... 8 Connection number

## Status displays



## Terminals

Module terminals, spring terminals C

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
| $1-\square_{5}^{5}-5$ | 1 | Not assigned | $0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
|  | 2 | I/O supply +24 V DC |  |
|  | 3 | I/O supply 0 V |  |
|  | 4 | Not assigned |  |
| $3^{7}$ | 5 | Not assigned | $10 \mathrm{~mm} \text { (AWG } 28 \ldots \text { 16) }$ |
|  | 6 | I/O supply +24 V DC |  |
| 1 | 7 | I/O supply 0 V |  |
| SL10002 | 8 | Not assigned |  |

1 Note!

- Terminals 2 and 6 as well as 3 and 7 are bridged internally. Please note that the max. permissible bridge current is 5 A .
- Both the I/O supply and the electronic supply are protected against overload internally by a fuse. When the fuses have been tripped, the main supply of the bus coupler (EPM-S700) must be replaced ( $\subseteq$ 726).

Technical data

| EPM-S701: Rated data |  |
| :---: | :---: |
| Module identifier | - |
| Electrical data |  |
| Input (supply) |  |
| Rated voltage | DC 24 V |
| Voltage range | DC 20.4 ... 28.8 V |
| Output |  |
| I/O supply | DC 24 V , max. 7 A (if UL conformity is required, max. 10 A ) |
| Status, alarm, diagnostics |  |
| Status display | Yes |
| Alarms | No |
| Process alarm | No |
| Diagnostic alarm | No |
| Diagnostic function | No |
| Diagnostic information can be read out | No |
| Module status display | Green LED |
| Module error display | Red LED |

## 3 Product description

Power supply modules
I/O supply and electronic supply - EPM-S702

### 3.13.2 I/O supply and electronic supply - EPM-S702

If the bus coupler main supply does not have sufficient power to feed the I/O level and/or the electronic components, this module can be used.

## Features

- Incoming supply for I/O and electronic supply
- Cable protection by monitored internal fuses


## Overview



Fig. 3-82 Elements and circuit diagram
(A) Displays for module status

B Terminals
(c) Power supply of the I/O level

D Electronic supply
1... 8 Connection number

## Status displays

| Module status LEDs A |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| View | Pos. | Designation | Colour | Explanation |
|  | 1 |  | Green | On: 24 V for I/O supply has been applied |
|  | 2 |  | Red | On: Fuse for I/O supply is defective |
| - | 3 |  | Green | On: 24 V for electronic supply is applied |
| 0 | 4 |  | Red | On: Fuse for electronic supply defective |
| $\square$ | 5 |  |  |  |
| $\begin{aligned} & \square \\ & \square \end{aligned}$ | 6 |  |  |  |
| $\square$ | 7 |  |  |  |
| 10-ロ | 8 | - |  | Not assign |
|  | 9 |  |  |  |
| SL10001 | 10 |  |  |  |

## Terminals

Module terminals, spring terminals ©

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
| $1-\square_{5}^{\square \square_{5}}-5$ | 1 | Not assigned | $\square 0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
|  | 2 | I/O supply +24 V DC |  |
|  | 3 | I/O supply 0 V |  |
| ${ }^{\square}{ }^{\square} \square_{6}^{6}$ | 4 | Electronic supply +24 V DC |  |
|  | 5 | Not assigned |  |
|  | 6 | I/O supply +24 V DC |  |
|  | 7 | I/O supply 0 V |  |
| SL10002 | 8 | Electronic supply 0 V |  |

## 1 Note!

- Terminals 2 and 6 as well as 3 and 7 are bridged internally. Please note that the max. permissible bridge current is 5 A .
- Both the I/O supply and the electronic supply are protected against overload internally by a fuse. When the fuses have been tripped, the main supply of the bus coupler (EPM-S700) must be replaced ( $\amalg 726$ ).


## 3 Product description

Power supply modules
I/O supply and electronic supply - EPM-S702

Technical data

3.14
3.14.1 8 terminals 24 V - EPM-S910

The terminals of this module provide the 24 Volts of the I/O supply. The backplane bus is looped through.

## Features

- 8 terminals 24 V (I/O supply)


## Overview

EPM-S910

Fig. 3-83 Elements and circuit diagram

| A | Terminals |
| :--- | :--- |
| $1 . . .8$ | Connection number |

## Terminals

Module terminals, spring terminals ©

| View | Designation | Explanation | Terminal data |
| :---: | :---: | :---: | :---: |
| $\square \square$ | 1 | I/O supply +24 V DC |  |
|  | 2 |  |  |
|  | 3 |  |  |
|  | 4 |  | $0.08 \ldots 1.5 \mathrm{~mm}^{2}$ |
| $3^{7}$ | 5 |  | (AWG 28 ... 16) |
|  | 6 |  |  |
| $4-40^{8}$ | 7 |  |  |
| SL10002 | 8 |  |  |

## Technical data

| EPM-S910: Rated data |  |
| :--- | :--- |
| Module identifier | - |
| Terminal parameters | DC 30 V |
| Terminal voltage, max. | 10 A |
| Terminal current, max. |  |

## 3 Product description

Power distributor modules
8 terminals 0 V - EPM-S911

### 3.14.2

8 terminals 0 V - EPM-S911
The terminals of this module provide the GND of the I/O supply. The backplane bus is looped through.

## Features

- 8 terminals mass GND (I/O supply)


## Overview



Fig. 3-84 Elements and circuit diagram
A $\quad$ Terminals
1... 8 Connection number

## Terminals



## Technical data

| EPM-S911: Rated data |  |
| :--- | :--- |
| Module identifier | - |
| Terminal parameters |  |
| Terminal voltage, max. | 0 V |
| Terminal current, max. | 10 A |

### 3.14.3 4/4 terminals 24 V/0 V - EPM-S912

The terminals of this module provide the 24 Volts and the GND of the I/O supply. The backplane bus is looped through.

## Features

- 4 terminals 24 V (I/O supply)
- 4 terminals mass GND (I/O supply)


## Overview

EPM-S912

Fig. 3-85 Elements and circuit diagram
(A) Terminals
1... 8 Connection number

## Terminals



## Technical data

| EPM-S912: Rated data |  |
| :--- | :--- |
| Module identifier | - |
| Terminal parameters | DC 30 V |
| Terminal voltage, max. | 10 A |
| Terminal current, max. |  |

## 4 Technical data

## Note!

The technical data of the I/O system modules can be found in the individual module descriptions which are provided in the "Product description" chapter.

## General data

## Conformity and approval

Conformity

| CE | 2006/95/EG | Low-Voltage Directive |
| :--- | :--- | :--- |
| Approval |  |  |
| UL | UL 508 | File No. E343358 |
| Other | - | Products lead-free in accordance with EC Directive <br> $2002 / 95 / E C$ |
| RoHS |  |  |

Protection of persons and device protection

| Degree of protection |  | IP20 |
| :--- | :--- | :--- |
| Electrical isolation <br> To the fieldbus |  | Electrically isolated |
| To the process level |  | Electrically isolated |
| Insulation resistance | IEC 61131-2 |  |
| Insulation voltage <br> against reference earth <br> of the inputs/outputs |  | AC / DC 50V, for test voltage AC 500V |
| Protective measures |  | Against short circuit |


| EMC |  |  |  |
| :---: | :---: | :---: | :---: |
| Noise emission | EN 61000-6-4 | Class A (industrial premises) |  |
| Noise immunity Zone B | EN 61000-6-2 | Industrial premises |  |
|  |  | EN 61000-4-2 | ESD; severity: 3, i.e. 8 kV in the case of air discharge, 4 kV in the case of contact discharge |
|  |  | EN 61000-4-3 | $\begin{aligned} & \text { RF interference (housing) } \\ & 80 \mathrm{MHz} \ldots 1000 \mathrm{MHz}, 10 \mathrm{~V} / \mathrm{m} 80 \% \mathrm{AM}(1 \mathrm{kHz}) \end{aligned}$ |
|  |  | EN 61000-4-4 | Burst, severity: 3 |
|  |  | EN 61000-4-5 | Surge, severity 3* |
|  |  | EN 61000-4-6 | ```RF conducted 150 kHz ... 80 MHz, 10 V/m 80% AM (1 kHz)``` |

[^4]
## Operating conditions

| Ambient conditions |  |  |
| :---: | :---: | :---: |
| Climatic |  |  |
| Storage | EN 60068-2-14 | $-25 \ldots+70^{\circ} \mathrm{C}$ |
| Operation |  |  |
| Horizontal installation | EN 61131-2 | $0 \ldots+60^{\circ} \mathrm{C}$ |
| Vertical installation | EN 61131-2 | $0 \ldots+60^{\circ} \mathrm{C}$ |
| Air humidity | EN 60068-2-30 | RH1 (without condensation, relative humidity 10 ... 95 \%) |
| Pollution | EN 61131-2 | Degree of pollution: 2 |
| Mechanical |  |  |
| Vibration | EN 60068-2-6 | 1 G |
| Shock | EN 60068-2-27 | 15 G |
| Mounting conditions |  |  |
| Mounting place |  | In the control cabinet |
| Mounting position |  | Horizontal and vertical |

## 5 Mechanical installation <br> Important notes

## 5 Mechanical installation

### 5.1 Important notes

- The mounting location must always ensure the operating conditions mentioned in the technical data. Take additional measures if necessary.
- Always ensure permanent mechanical connections.
- The fixing rail and the mounting plate in the control cabinet must be electrically conductive and free of lacquer.
- Plug and unplug the modules only if the supply voltage has been switched off to prevent the modules of the I/O system from being damaged by short circuit.
- Always arrange the modules from the left to the right and start with the bus coupler.
- The modules must always be plugged directly next to each other. Free slots are not permissible since this would interrupt the backplane bus.
- Always use the contact cover - which is included in the scope of supply of the bus coupler module - to protect the last module's contacts to the sides. Otherwise, the modules of the I/O system may become damaged by short circuit or static discharge.
5.2 Dimensions


Fig. 5-1
Dimensions and mounting clearances
© Mounting clearance without shield bus B Mounting clearance with shield bus
All dimensions in millimetres.

## 5 Mechanical installation

Mounting
Standard mounting

### 5.3 Mounting

### 5.3.1 Standard mounting



Fig. 5-2 Mounting modules

| EPM-S1xx | Bus coupler module |
| :--- | :--- |
| EPM-S2xx ... EPM-S6xx | I/O compound module |
| EPM - S7xx | Power supply module |
| EPM-S9xx | Power distributor module |
| A | Contact cover (included in the scope of supply of the bus coupler module) |

### 5.3.2 Block mounting



Fig. 5-3 Mounting modules in blocks
EPM-S1xx Bus coupler module
EPM-S2xx ... EPM-S6xx 1/O compound module
EPM-S7xx Power supply module
EPM-S9xx Power distributor module
(A)

Contact cover (included in the scope of supply of the bus coupler module)

## 5 Mechanical installation

Mounting
Mounting the busbar for the shield connection

### 5.3.3 Mounting the busbar for the shield connection



Fig. 5-4 Mounting the busbar for the shield connection

| EPM-Sxxx | Bus coupler module, I/O compound module, power supply module |
| :--- | :--- |
| A | Busbar support for EPM-S900 shield connection (accessories) |
| B | Busbar $10 \times 3 \mathrm{~mm}$ (available in selected stores) |
| C | Shield connection terminal (available in selected stores) |

Insert the EPM-S900 © busbar support into the base module as shown in the illustration. Mount more supports for assistance in case of longer busbars.
5.3.4 Dismounting the bus coupler module, the I/O compound module, and the power supply module


Fig. 5-5
Dismounting modules

## 6 Electrical installation

### 6.1 Wiring according to EMC

| General notes | The electromagnetic compatibility of the system depends on the type of installation and care <br> taken. Especially consider the following: <br> - Assembly <br> - Shielding <br> - Earthing |
| :--- | :--- |
| - For installations differing from the one described, the evaluation of the conformity with the |  |
| EMC Directive requires a check of the system regarding the EMC limit values. This for instance |  |
| applies to: |  |
| - Use of unshielded cables |  |
| - The compliance with the EMC Directive is in the responsibility of the user. |  |
| - If you observe the following measures, you can assume that no EMC problems will occur |  |
| during operation and that compliance with the EMC Directive and the EMC law is achieved. |  |
| - If devices which do not comply with the CE requirement concerning noise immunity (EN |  |
| 6100042) are operated close to the system, these devices may be electromagnetically |  |
| affected by the system. |  |

## 6.2

## Supply voltage connection

## 1 Note!

The supply inputs of the bus coupler modules (main supply) and the power supply modules are provided with internal fuses which protect them against overvoltages.
The fuse is located in the main supply of the bus coupler module and in the electronic module of the power supply module. If this fuse has tripped, the main supply or the electronic module, respectively, needs to be replaced (■ 726).


SLIOO15
Fig. 6-1 Supply via bus coupler module (main supply)
*) If no UL conformity is required, the maximum permissible load for the I/O supply is 10 A .
A I/O supply
The I/O supply must be secured externally using a fuse that corresponds to the maximum current: fast fuse, or circuit breaker with a $Z$ characteristic
B Electronic supply
We recommend securing the electronic supply externally according to the maximum current: fast fuse, or circuit breaker with a Z characteristic


Fig. 6-2 Supply via Bus coupler module (main supply) and power supply modules
${ }^{*}$ ) If no UL conformity is required, the maximum permissible load for the I/O supply is 10 A .
(A) I/O supply

The I/O supply must be secured externally using a fuse that corresponds to the maximum current: fast fuse, or circuit breaker with a Z characteristic
B Electronic supply
We recommend securing the electronic supply externally according to the maximum current: fast fuse, or circuit breaker with a Z characteristic

### 6.3 Wiring of the control connections

## 1 Note!

Information on wiring the connections of I/O compound modules can be found in the individual descriptions of the modules that are provided in the "Product description" chapter.
(15) Information on mounting the busbar for the shield connection can be found in the "Mechanical installation" chapter ( $\square$ 244).


Fig. 6-3
Attaching the cable shield to the shield bus

| EPM-Sxxx | Bus coupler module, I/O compound module, power supply module |
| :--- | :--- |
| A Busbar support for EPM-S900 shield connection (accessories) <br> B Busbar $10 \times 3 \mathrm{~mm}$ (available in selected stores) <br> C Shield connection terminal (available in selected stores) |  |

## 7 Troubleshooting and fault elimination

### 7.1 Troubleshooting via RUN- and MF-LED

On the front, each module is equipped with the RUN and MF LEDs. You can use these LEDs to determine errors in your system or defective modules.

| Response | Cause | Remedy |
| :--- | :--- | :--- |
| After switch-on, the RUN-LED stays <br> off on each module, and the MF-LED <br> is lit sporadically. | The maximum current for the <br> electronic supply is exceeded. | Place an EPM-S702 power supply <br> module in the position in which the <br> total current for the electronic <br> supply exceeds the maximum <br> current |
| After switch-on, the MF-LED is <br> blinking on one module or on <br> several modules. The RUN-LED <br> remains switched off. | In this position a module is plugged <br> in, which does not comply with the <br> module currently configured. | Coordinate the configuration and <br> the hardware setup. |
| After switch-on, all RUN-LEDs up to <br> the defective module are blinking. <br> At all following modules, the MF <br> LED is lit and the RUN-LED is off. | The module on the right to the <br> blinking modules is defective. | Replace the defective module. |

1 Note!
Further fault messages of the I/O system modules can be found in the individual module descriptions which are provided in the "Product description" chapter.

## 8 CANopen communication

## 8.1 <br> About CANopen

The I/O system supports the communication module CANopen.
The CANopen protocol is a standardised layer-7 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.

### 8.1.1 Structure of the CAN data telegram



Fig. 8-1 Basic structure of the CAN telegram

## 1

## Note!

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.

### 8.1.2 Identifier

The principle of CAN communication is based on a message-oriented data exchange between a transmitter and many receivers. Therefore, all nodes can transmit and receive more or less at the same time.
The so-called identifier in the CAN telegram, also called COB-ID (Communication Object Identifier)), controls 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(■32).
- Network management and sync telegram only require the basic identifier.
- The identifiers can also be set individually(■256).


### 8.1.3 Saving settings

The settings are permanently stored via $11010_{\mathrm{h}}$ (communication protocol DS301/DS401).

## 8.2 <br> Network management (NMT)

Via the network management, the master can change a communication status for the whole CAN network.

## Communication phases

| Status | Explanation |
| :--- | :--- |
| "Initialisation" | Initialisation starts when the I/O system is switched on. In this phase, the I/O <br> system does not take part in the bus data transfer. <br> Furthermore it is in each NMT status possible to restart the entire initialisation <br> or parts of it by transferring different telegrams (see "Status transitions"). All <br> parameters already set are overwritten with their standard values. <br> After initialisation has been completed, the I/O system is automatically set to <br> the status "Pre-operational". |
| "Pre-Operational" <br> (before ready for operation) | The I/O system can receive parameter data. <br> The process data are ignored. |
| "Operational" |  |
| (Ready for operation) | The I/O system can receive parameter and process data. |
| "Stopped" | Parameter and process data cannot be received. Network management <br> telegrams can be received. <br> The module outputs switch to the configured status (see chapter "Monitoring"). |

## Telegram structure

|  | Identifier <br> Value = 0 <br> 11 bit |  | User data <br> Only contains command <br> 2 byte |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |

Fig. 8-2 Telegram for changing the communication phase
The telegram used for network management contains an identifier and the command which is part of the user data and consists of command byte and node address.
Telegrams with the identifier 0 and two bytes user data are used to change between the communication phases.
Only the network master (e.g. controller) can change a communication status for the whole network.

## Note!

Only a change to "Operational" status enables communication via the 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



Fig. 8-3 Network management status transitions

| Status transition | Command (hex) | Network status after change | Effects on process and parameter data |
| :---: | :---: | :---: | :---: |
| (1) | - | Initialisation | Initialisation starts automatically when the mains is switched on. <br> During initialisation, the I/O system does not take part in the data transfer. <br> After initialisation has been completed, the device sends a boot-up message with an identifier to the master. The device is automatically set to the Pre-operational status. |
| (2) | - | Pre-Operational | In this phase, the master determines the I/O system communication. |

From that moment on, the master changes a status for the whole network. A target address, which is part of the command, selects the slave(s).

| (3), (6) | 01 xx | Operational | Network management telegrams, sync, emergency, process data (PDO) and parameter data (SDO) are active. Optional: <br> When the status is changed, event and time-controlled process data (PDO) will be sent once. |
| :---: | :---: | :---: | :---: |
| (4), (7) | 80 xx | Pre-Operational | Network management telegrams, sync, emergency and parameter data (SDO) are active (like "Enter pre-operational state") |
| (5), (8) | 02 xx | Stopped | Parameter and process data cannot be received. Network management telegrams can be received. |
| (9) | 81 xx | Initialisation | The parameters saved last via $1 \mathbf{1 0 1 0}{ }_{h}$ are loaded for all indexes. If no values have been saved yet, the Lenze setting will be loaded. |
| (10) |  |  |  |
| (11) |  |  |  |
| (12) | 82 xx |  | The parameters saved last via $I_{1010 \mathrm{~h}}$ are loaded for all communication parameters (index $0-1$ FFFh). If no values have been saved yet, the Lenze setting will be loaded. |
| (13) |  |  |  |
| (14) |  |  |  |

[^5]
### 8.3 Transmitting process data

Process data are used for control-specific purposes, such as setpoint and actual values, for example.

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


### 8.3.1 Process data telegram

Structure of the process data telegram:

| 11 bits | 8 bytes of user data |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |  |

## Identifier

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

## (1) Note!

In the Lenze setting, the identifiers are defined according to CANopen (assignment in accordance with DS301). Via $12101_{h}$ the identifier calculation according to the system bus (assignment according to the Lenze system bus) can be specified.

## User data

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

## i Note!

Lenze controllers 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 $12100_{h}$ :
0 : PDO length of eight bytes (Lenze setting)
1: PDO length according to process image

### 8.3.2 Identifier 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

|  |  |  | Basic identifier |  |
| :---: | :---: | :---: | :---: | :---: |
|  |  |  | dec | hex |
| PDOs | Process data object 1 |  |  |  |
|  |  | PDO1-Rx | 512 | 200 |
|  |  | PDO1-Tx | 384 | 180 |
|  | Process data object 2 |  |  |  |
|  |  | PDO2-Rx | 768 | 300 |
|  |  | PDO2-Tx | 640 | 280 |
|  | Process data object 3 |  |  |  |
|  |  | PDO3-Rx | 1024 | 400 |
|  |  | PDO3-Tx | 896 | 380 |
|  | Process data object 4 |  |  |  |
|  |  | PDO4-Rx | 1280 | 500 |
|  |  | PDO4-Tx | 1152 | 480 |
|  | Process data object 5 |  |  |  |
|  |  | PDO5-Rx | 1920 | 780 |
|  |  | PDO5-Tx | 1664 | 680 |
|  | Process data object 6 |  |  |  |
|  |  | PDO6-Rx | 576 | 240 |
|  |  | PDO6-Tx | 448 | 1C0 |
|  | Process data object 7 |  |  |  |
|  |  | PDO7-Rx | 832 | 340 |
|  |  | PDO7-Tx | 704 | 2C0 |
|  | Process data object 8 |  |  |  |
|  |  | PDO8-Rx | 1088 | 440 |
|  |  | PDO8-Tx | 960 | 3 CO |
|  | Process data object 9 |  |  |  |
|  |  | PDO9-Rx | 1344 | 540 |
|  |  | PDO9-Tx | 1216 | 4C0 |
|  | Process data object 10 |  |  |  |
|  |  | PDO10-Rx | 1984 | 7C0 |
|  |  | PDO10-Tx | 1728 | 6C0 |

### 8.3.3 Assigning individual parameters

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 $11400_{h}$, subindex $1 . . .11409_{h}$, subindex 1.

## Process data objects for output data

Individual identifier for output data can be set via the indices $11800_{h}$, subindex $1 . . .1809_{h}$, subindex 1.


## Note!

- Set the value which makes the required identifier ( $\mathrm{x}=$ corresponding process data object) in index $1140 x_{h}$, subindex 1 or $1180 x_{h}$, subindex 1.
Make a reset node so that the changes are accepted,


### 8.3.4 Process data transmission mode

## Transmission mode for process input data

The transmission mode is configured via the index $11400_{h}$, subindex 2 (PDO1-Rx) ... $11409_{h}$, subindex 2 (PDO10-Rx):

- Sync-controlled reception
- n -sync-controlled reception
- First, a certain number ( $n$ ) of sync telegrams must be transmitted (I140x ${ }_{h}$, 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 the index $11800_{h}$, subindex 2 (PDO1-Tx) ... $11809_{h}$, subindex 2 (PDO10-Tx):

- Sync-controlled transmission
- n -sync-controlled transmission
- First, a certain number ( n ) of sync telegrams must be transmitted (I180 $\mathrm{x}_{\mathrm{h}}$, subindex $2=2 \ldots 240$ ). 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.


Fig. 8-4 Synchronisation of cyclical process data with the help of a sync telegram (asynchronous data not considered)
(1) Sync telegram

## Transmission sequence

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.

### 8.3.5 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 controller without a master control. Configuration with L-force Engineer | - Direct data exchange between the IO system and the control (IPC). Configuration via L-force PLC Designer |
| - Static PDO mapping (default setting) | - Free mapping by means of a 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:


| 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 | M3M4 | 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, RS485 | RPDO3 | - | - | - | - | - | - | - | - |
|  |  | TPDO3 | M12 | M12 | M12 | M12 | - | - | - | - |
| PDO4 | DI/DO, DI/DO time stamp, DO PWM, AI/AO, counter, SSI, RS232, RS485 | RPDO4 | - | - | - | - | - | - | - | - |
|  |  | TPDO4 | M10 | - | - | - | - | - | - | - |
| ... | ... | ... |  |  |  |  |  |  |  |  |
| PDO10 | DI/DO, DI/DO time stamp, DO PWM, AI/AO, counter, SSI, RS232, 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/RS485 interface (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 a byte in a PDO does not have enough free bits, the I/O compound module is mapped in the next byte.

- Analog I/O:

In the case of analog values the data lengths of which exceed one 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.
8.3.6 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



Fig. 8-5 Mapping contents

| Word | EPM-S640, EPM-S650 |
| :---: | :---: |
| 1 | Rx byte $0 \ldots 1$ |
| 2 | Rx byte $2 \ldots 3$ |
| 3 | Rx byte $4 \ldots 5$ |
| 4 | Rx byte $6 \ldots 7$ |
| 5 | - |
| 6 | - |
| 7 | - |
| 8 | - |
| 9 | Tx byte $0 \ldots 1$ |
| 10 | Tx byte $2 \ldots 3$ |
| 11 | Tx byte $4 \ldots 5$ |
| 12 | Tx byte $6 \ldots 7$ |
| 13 | - |
| 14 | - |
| 15 | - |
| 16 | - |

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

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

## Default mapping



Fig. 8-6 Mapping contents

| Word | EPM-S207 | EPM-S310 |
| :---: | :---: | :---: |
| 1 | - | Ticker value DO1 |
| 2 | - | Running number <br> Channel status DO1 |
| 3 | - | Ticker value DO2 |
| 4 | - | Running number <br> Channel status DO2 |
| 5 | - | - |
| 6 | - | - |
| 7 | - | - |
| 8 | Ticker value DI1 | - |
| 9 | Running number | Status of FIFO memory |
| 10 | Channel status DI1 | Ticker value DI2 |
| 11 | Running number | - |
| 12 | Channel status DI2 | - |
| 13 | - | - |
| 14 | - | - |
| 15 | - | - |

Ticker value EPM-S207 and EPM-S310

| Bit | Designation | Function |
| :--- | :--- | :--- |
| $\mathbf{0 \ldots 1 5}$ | Ticker value | After mains connection, a timer ( $\mu$ s ticker) is <br> started, which after $65535 \mu$ starts with 0 again. |


| Running number, channel status EPM-S207 |  |  |
| :---: | :---: | :---: |
| Bit | Designation | Function |
| 0... 7 | Running number (RN) | Counter which counts from 0 ... 127 and then goes back to 0 . |
| 8 | Channel status DI1 | $\begin{aligned} & \text { 0: FALSE } \\ & \text { 1: TRUE } \end{aligned}$ |
| 9 | Channel status DI2 | $\begin{aligned} & \text { 0: FALSE } \\ & \text { 1: TRUE } \end{aligned}$ |
| $10 . . .15$ | - | Reserved (fix 0) |
| Running number, channel status EPM-S310 |  |  |
| Bit | Designation | Function |
| 0... 7 | Running number (RN) | Counter which counts from 0 ... 127 and then goes back to 0 . |
| 8 ... 11 | - | Reserved (fix 0) |
| 12 | Enable DO1 | 0: inhibit <br> 1: enable |
| 13 | Enable DO2 |  |
| 14 | Channel status DO1 | $\begin{aligned} & \text { 0: FALSE } \\ & \text { 1: TRUE } \end{aligned}$ |
| 15 | Channel status DO2 | $\begin{aligned} & \text { 0: FALSE } \\ & \text { 1: TRUE } \end{aligned}$ |


| Status of FIFO memory for EPM-S310 |  |  |
| :---: | :---: | :---: |
| Bit | Designation | Function |
| 0 ... 5 | RN-LAST | Running number of the time stamp entry last detected as valid 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 to be 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 | Status information of FIFO memory <br> - $00_{h}$ or $80_{h}$ : Everything is OK; you will receive this message directly after the acceptance into the FIFO memory of the module. <br> - $01_{h}$ or $81_{h}$ : No follow-up entry available; the running number does not correspond to the expected running number. Check the running number in the output area. <br> - $02_{h}$ or $82_{h}$ : No new entries available in the FIFO memory. <br> - 03h or 83 h : FIFO memory full. No new entries possible. <br> 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. <br> Note: If less than possible time stamp entries are written, bit 6 must be set in the running number (RN) 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 $80_{h}$ is returned in an OR-ed manner. |
| 24 ... 31 | - | Current number of time stamp entries in FIFO memory. |

## 8 CANopen communication

## Transmitting process data

PDO mapping for I/O compound modules with PWM function
8.3.8 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



Fig. 8-7 Mapping contents


EPM-S620 control word
\(\left.$$
\begin{array}{l|l|l}\hline \text { Bit } & \text { Designation } & \text { Function } \\
\hline 0 \ldots 1 & - & \begin{array}{l}\text { Reserved }\end{array} \\
\hline 2 & \text { PWM response } & \begin{array}{l}\text { 0: push/pull output } \\
\text { Push/pull mode should be used if you need } \\
\text { defined high/low levels for a rapid change. This } \\
\text { is used with a low load especially if "highside" } \\
\text { mode cannot move the output to low fast } \\
\text { enough during a low status. With push/pull, } \\
\text { the output is switched to ground with low } \\
\text { active and to voltage with high active. } \\
1: \text { highside output }\end{array}
$$ <br>
In highside mode, the output switched to low <br>
remains in a state of uncertainty between <br>
ground and voltage. The load has to "pull" itself <br>
to ground. In highside mode, the switch is only <br>

made to high level active.\end{array}\right\}\)|  |  |
| :--- | :--- |
| $3 \ldots 7$ | Output status |
| 8 | Start PWM output |
| 9 | Stop PWM output |
| $10 \ldots 15$ | - |

## 8 <br> CANopen communication

## Transmitting process data

PDO mapping for counters and encoder evaluation
8.3.9 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.


## Default mapping



Fig. 8-8 Mapping contents
$\begin{array}{c|c|c|c|c|c|c}\hline \text { Word } & \text { EPM-S600 } & \text { EPM-S601 } & \text { EPM-S602 } & \text { EPM-S603 } & \text { EPM-S604 } \\ \hline 1 & \text { Comparison value } & \begin{array}{c}\text { Comparison value } \\ \text { C1 }\end{array} & \text { Comparison value } & \text { Control word C1 } & \\$\cline { 1 - 1 } 2 \& Set value \& $\begin{array}{c}\text { Comparison value } \\ \text { C2 }\end{array} & \text { Set value } & & \\$\cline { 1 - 1 } 3 \& Control word C2\end{array}$)$

Comparison value: Here you can select a value which, by comparison with the current counter content, can impact the counter output or trigger a process alarm. The response of the output or 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 ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu \mathrm{s}$ value together with the count value in the input area.

## 8 CANopen communication

Transmitting process data
PDO mapping for counters and encoder evaluation

| EPM-S600 | status word |  |
| :--- | :--- | :--- |
| Bit | Designation | 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 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_UP | Status set for counter direction backwards |
| 8 | STS_CMP* | Comparator status is set if the comparison condition is met. <br> If the comparison is deactivated (counter mode byte $1=000_{b}$ ) , the bit has <br> no function. |
| 9 | STS_OFLW* | Status set if final value was reached |
| 10 | STS_UFLW* | Status set in the case of overflow |
| 11 | STS_ZP* | Status set in the case of underflow |
| 12 | STS_LTCH | Status set in the case of zero crossing |
| 13 | - | Status of latch input |
| 14 | Reserved |  |
| 15 | The bits are set until reset with RES_SET (bit 6 control word) |  |
|  |  |  |


| EPM-S600 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | Function |
| 0 | CTRL_SYNC_SET | Activates the reset mode |
| 1 | CTRL_DO_SET | Enables the digital output |
| 2 | SW_GATE_SET | Sets the software gate |
| 3 | - | Reserved |
| 4 | - | Reserved |
| 5 | COUNTERVAL_SET | Sets the counter temporarily to the value in the set value |
| 6 | CTS_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | CTRL_DO_RESET | Reserved |
| 8 | SW_GATE_RESET | Reactivates the reset mode |
| 9 | - | Resets the software gate |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |


| EPM-S601 | status word |  |
| :--- | :--- | :--- |
| Bit | Designation | 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 for counter direction backwards |
| 8 | STS_CMP* | Status set for counter direction forwards |
| 9 | STS_END* | Comparator status is set if the comparison condition is met. <br> If the comparison is deactivated (counter mode byte $1=000_{b}$ b <br> no function. |
| 10 | STS_OFLW | Status set if final value was reached |
| 11 | STS_UFLW* | Status set in the case of overflow |
| 12 | STS_ZP* | Status set in the case of underflow |
| 13 | - | Status set in the case of zero crossing |
| 14 | Reserved |  |
| 15 | Reserved |  |
| $*$ | The bits are set until reset with RES_SET (bit 6 control word) |  |


| EPM-S601 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | 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 | Sets the counter temporarily to the value in the set value |
| 6 | RES_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | - | Reserved |
| 8 | CTRL_COMP_RESET | Reserved |
| 9 | SWhibits comparison bit |  |
| 10 | - | Resets the software gate |
| 11 |  | Reserved |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

## 8 CANopen communication

Transmitting process data
PDO mapping for counters and encoder evaluation


| EPM-S602 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | Function |
| 0 | CTRL_SYNC_SET | Activates the reset mode |
| 1 | CTRL_COMP_SET | Enables the comparison bit |
| 2 | SW_GATE_SET | Sets the software gate |
| 3 | - | Reserved |
| 4 | - | Reserved |
| 5 | COUNTERVAL_SET | Sets the counter temporarily to the value in the set value |
| 6 | CES_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | CTRL_SYNC_RESET | Reserved |
| 8 | SW_GATE_RESET | Reactivates the reset mode |
| 9 | - | Resers the software gate |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |


| EPM-S603 status word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | 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 for counter direction backwards |
| 8 | STS_C_UP | Status set for counter direction forwards |
| 9 | - | Reserved |
| 10 | - | Reserved |
| 11 | STS_OFLW* | Status set in the case of overflow |
| 12 | STS_UFLW* | Status set in the case of underflow |
| 13 | STS_ZP* | Status set in the case of zero crossing |
| 14 | - | Reserved |
| 15 | - | Reserved |

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

| EPM-S603 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | Function |
| 0 | - | Reserved |
| 1 | - | Reserved |
| 2 | SW_GATE_SET | Sets the software gate |
| 3 | - | Reserved |
| 4 | - | Reserved |
| 5 | - | Reserved |
| 6 | RES_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | - | Reserved |
| 8 | - | Reserved |
| 9 | - | Reserved |
| 10 | SW_GATE_RESET | Resets the software gate |
| 11 | - | Reserved |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

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 controllers the basic identifiers for the process data object 1 must be adapted.

1. Set RPDO1 via index $1400_{h}$, subindex 1 to 770 .
2. Set TPDO1 via index $1800_{h}$, subindex 1 to 769 .

The input data are accepted on sync telegram transmission.


Data transfer between I/O system and controller
PDO-Rx The I/O system receives the status information from the controller PDO-Tx The I/O system transmits the status information to the controller (A) Controller with node address $1(C 0350=1)$
(1) $768_{\mathrm{d}}$ (basic identifier) +1 (node address) $=769_{\mathrm{d}}$ (identifier)
(2) $769_{\mathrm{d}}$ (basic identifier) +1 (node address) $=770_{\mathrm{d}}$ (identifier)

B CAN bus coupler module with node address 2
(3) $767_{\mathrm{d}}$ (basic identifier) +2 (node address) $=769_{\mathrm{d}}$ (identifier)
(4) $\quad 768_{d}$ (basic identifier) +2 (node address) $=770_{d}$ (identifier)

### 8.3.11 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 $0 \ldots 11$ | Bit $12 \ldots 29$ | 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 $31=0$
$0 \times 40000202$
The PDO is activated with a new identifier. The changes will be accepted in the Pre-Operational or Operational status.

| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selectio |  |  |  |
| $11400_{h}$ |  |  |  |  |  | - 256 |
| 1 | COB-ID used by RxPDO 1 | $\begin{gathered} 513+ \\ \text { NID } \end{gathered}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 . . .24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | Reserved |  |  |
|  |  |  | $255$ | Process data update on every occurrence of an event | Every received value is accepted |  |
| ... | ... |  |  |  |  |  |
| $\mathrm{I} 1409_{\mathrm{h}}$ |  |  |  |  |  | -1256 |
| 1 | COB-ID used by RxPDO 10 | $\begin{array}{r} 1984 \\ + \text { NID } \end{array}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 \ldots 24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | 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 $0 \ldots 11$ | Bit $12 \ldots 29$ | 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 | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Sele |  |  |  |
| $\begin{array}{r} 1800_{\mathrm{h}} \\ 1 \end{array}$ |  |  |  |  |  | -1 256 |
|  | COB-ID used by TxPDO 1 | $\begin{gathered} 384+ \\ \text { NID } \end{gathered}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... Process data transfer after sync no. 240 | The output data are accepted after transmission of the set number (1 ... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180x $x_{h}$, 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 I180x ${ }_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms \} 65535 | Blocking time |  |
| 5 | Event time | 100 | 0 | \{1 ms \} 65535 | Cycle time |  |
| ... | ... |  |  |  |  |  |


| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Sele |  |  |  |
| $\begin{aligned} & \frac{\mathrm{I} 1809 \mathrm{~h}}{1} \\ & 1 \end{aligned}$ |  |  |  |  |  | ¢ 256 |
|  | COB-ID used by TxPDO 10 | $\begin{array}{r} 1728 \\ + \text { NID } \end{array}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... <br> Process data transfer after sync no. $240$ | The output data are accepted after transmission of the set number (1... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180xh , subindex 5 |  |
|  |  |  |  | Event-controlled process data transfer |  |  |
|  |  |  | 255 | Event-controlled process data transfer with cyclic overlay | Only if a cycle time is set in I180 $x_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms 665535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms \} 65535 | Cycle time |  |

### 8.4 Transmitting parameter data

Parameter data are the so-called indices.
Parameters are usually set only once during commissioning.
Parameter data are transmitted as so-called SDOs (Service Data Objects) via the system bus and acknowledged by the receiver, i.e. the transmitter gets a feedback if the transmission was successful.

### 8.4.1 Telegram structure

Structure of the telegram for parameter data:

| 11 bits | 8 bytes of user data |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | Instruction <br> code | Index <br> Low byte | High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 | Low |
| :---: |

- The subchapters below explain the individual telegram components in detail.
- The chapter 8.4.2 contains an example of how to write a parameter. $\qquad$
- The chapter 8.4.3 contains an example of how to read a parameter.280)


## Identifier

| IdentifierInstruction <br> code | Index <br> Low byte | High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |

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


## Instruction code

| Identifier | Instruction <br> code | Index |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Low byte | High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |

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


Instruction code for parameters with 4 bytes of data length:

| Command | $\mathbf{4}$ bytes of data |  | Information |
| :--- | :---: | :---: | :--- | :--- |
|  | hex | $\mathbf{d e c}$ |  |
| 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 |

## Instruction "Error Response"

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 |
| :---: | :---: | :---: | :---: | :--- |
|  | 3 |  | Access denied |
|  | 5 | 6 | Wrong subindex |

## Parameter addressing (Index/subindex)

| Identifier | Instruction <br> code | Index <br> Low byte High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |

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 $\mathrm{I} 2400_{\mathrm{h}}$ (monitoring time for PDO1) is to be addressed:

| 11 bits | 8 bytes of user data |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | Instruction code | Index |  |  |  |  |  |  |
|  |  | Low byte | High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
|  |  | $00_{\text {h }}$ | 24h | 1 |  |  |  |  |

## Parameter data (data 1 ... data 4)

| Identifier | Instruction <br> code | Index <br> Low byte |  | High byte | Subindex | Data 1 | Data 2 |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | Data 3 $\quad$ Data 4

Up to 4 bytes (data 1 ... data 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 $2400_{\mathrm{h}}$ (monitoring time).

```
Data 1...4}=1\times1000=1000=000003 E8h
```

| 11 bits | 8 bytes of user data |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | Instruction code | Index |  |  |  |  |  |  |
|  |  | Low byte | High byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
|  |  |  |  |  | $E 8{ }_{h}$ | $03_{h}$ | $00{ }_{h}$ | $00_{\text {h }}$ |
|  |  |  |  |  | (LSB) |  |  | (MSB) |

### 8.4.2 Writing a parameter (example)

## Task

An I/O system 1000 has been assigned node address 2 . The function of the first channel (voltage signal $0 \ldots+10 \mathrm{~V}, 0 \ldots 27648 \mathrm{dec}$ ) 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 |  |  | Information |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | $\begin{aligned} & =\text { Basic identifier + node address } \\ & =1536+2=1538=602_{\mathrm{h}} \end{aligned}$ |  |  | - Basic identifier for parameter channel 1 (output) = 1536 <br> - Node address of the I/O system = 2 |  |  |  |  |
| Instruction code | $=23 \mathrm{~h}$ |  |  | - Write Request command (transmitting parameters to the I/O system) |  |  |  |  |
| Index | $=13100{ }_{\text {h }}$ |  |  | - First channel of the analog module |  |  |  |  |
| Subindex | = 1 |  |  | - First analog module |  |  |  |  |
| Data 1 <br> Data 2 ... 4 | $\begin{aligned} & =0 \mathrm{~F}_{\mathrm{h}} \\ & =00_{\mathrm{h}} \end{aligned}$ |  |  | - 0 ... 10 V, 0 ... $07648_{\text {dec }}$ <br> - No function |  |  |  |  |
| 11 bits |  |  |  | Eight bytes of user data |  |  |  |  |
| Identifier | Instructio n code | LOW byte HIGH byte |  | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 602h | $23_{h}$ | $00_{h}$ | $31_{h}$ | $0 F_{h}$ | $00{ }_{h}$ | 00 h | 00 h | $00{ }_{h}$ |
|  |  |  |  | (LSB) |  |  |  | (MSB) |
| Read Request <br> Read Response | $\begin{aligned} & \text { Identifier }=153 \\ & \text { Identifier }=141 \end{aligned}$ |  |  |  |  |  |  |  |

Fig. 8-10 Writing a parameter

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

|  | Formula |  |  | Information |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | $\begin{aligned} & =\text { Basic identifier + node address } \\ & =1408+2=1410 \end{aligned}$ |  |  | - Basic identifier for parameter channel 1 (input) $=1408$ <br> - Node address of the I/O system = 2 |  |  |  |  |
| Instruction code | $=60 \mathrm{~h}$ |  |  | - 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 bits | Eight bytes of user data |  |  |  |  |  |  |  |
| Identifier | Instructio n code | LOW byte | ex HIGH byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 1410 | 60 h | $01_{h}$ | $30_{h}$ | 0 | 0 | 0 | 0 | 3 |

### 8.4.3 Reading a parameter (example)

## Task

An I/O system 1000 has been assigned node address 2 . 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



Fig. 8-11 Reading a parameter

## Telegram from the $\mathrm{I} / \mathrm{O}$ system (value of the requested parameter)

|  | Formula |  |  | Information |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Identifier | $\begin{aligned} & =\text { Basic identifier }+ \text { node address } \\ & =1408+2=1410 \end{aligned}$ |  |  | - Basic identifier for parameter channel 1 (input) $=1408$ <br> - Node address of the I/O system = 2 |  |  |  |  |
| Instruction code | $=43 \mathrm{~h}$ |  |  | - 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 <br> Data 2 <br> Data 3 <br> Data 4 <br> Data $1 . . .4$ | $\begin{aligned} & =00_{h} \\ & =00_{h} \\ & =05_{\mathrm{h}} \\ & =3 B_{h} \\ & =\mathbf{0 0} 0005 \mathbf{3 B _ { h }} \end{aligned}$ |  |  | - Assumption: The first channel of the first analog module outputs a voltage signal $0 \ldots+10 \mathrm{~V}$ with a resolution of $0 \ldots 07648 \mathrm{dec}$. |  |  |  |  |
| 11 bits | Eight bytes of user data |  |  |  |  |  |  |  |
| Identifier | Instructio n code | LOW byte | ex HIGH byte | Subindex | Data 1 | Data 2 | Data 3 | Data 4 |
| 1410 | $43_{h}$ | 00 h | $31_{\text {h }}$ | 0 | $00{ }_{h}$ | $00{ }_{\text {h }}$ | 05 h | 3Bh |

### 8.5 Response of the station after switch-on

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


## 8.6

## 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 (■31).


## 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 ( -132 ).
- The set node address can be read in index $1100 B_{h}$.


## 8 CANopen communication

General function of the parameter setting
Parameterising digital I/Os

### 8.7 General function of the parameter setting

### 8.7.1 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 index I6002h.
- Digital outputs are parameterised via index $16202_{h}$.

The subindex depends on the slot.

| Index |
| :---: |
| 16002 n |
| I6202n |

Module

幺
Module 64
Subindex 64


Fig. 8-12 Display of the digital I/O parameter data

| Byte | Assignment | Bit0 | 0 | Signal is transmitted in original <br> form | $00_{\mathrm{h}}$ |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 0 | Polarity of the transmitted <br> signals | Bit $1 . . .7$ | Signal is transmitted in inverse <br> form <br> Reserve |  |  |

## 1 Note!

Store changed parameters in the EEPROM via index $1010{ }_{h}$. The settings are maintained after the supply voltage is switched off.

### 8.7.2 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 $0 \times 3100$... $0 \times 3129$.
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 $\mathbf{I} / \mathbf{0})$ | \#2 (analog I/O) | \#3 (counter) | ... |
| :--- | :---: | :---: | :---: | :---: |
| Signal functions | $0 \times 3100 / 1$ | $0 \times 3100 / 2$ | $0 \times 3100 / 3$ |  |
|  | $:$ | $:$ | $:$ | $\ldots$ |

### 8.8 Setting the parameters of analog I/O

8.8.1 2 analog inputs $0 \ldots 10 \mathrm{~V}$ (12 bits) - EPM-S400

Parameter data

| Index/sub <br> index | Name | Description/value | Lenze |  |
| :--- | :--- | :--- | :--- | :--- |
| $0 \times 3100 / \mathrm{x}$ | Function channel 1 | $16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}}$ <br> $32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}}$ <br> $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated | $10_{\mathrm{h}}$ |  |
| $0 \times 3101 / \mathrm{x}$ | Function channel 2 | 0 | $10_{\mathrm{h}}$ |  |
| $0 \times 3102 / \mathrm{x}$ | Reserved | $\ldots$ | $\ldots$ |  |
| $\ldots$ | 0 | Reserved |  |  |
| $0 \times 311 \mathrm{D} / \mathrm{x}$ | R |  |  |  |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \text { V }}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

### 8.8.2

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

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Function channel 1 | $\begin{aligned} & 16\left(10_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots . .16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $10_{\text {h }}$ |
| 0x3101/x | Function channel 2 |  | $10_{\text {h }}$ |
| 0x3102/x | Function channel 3 |  | $10_{\text {h }}$ |
| 0x3103/x | Function channel 4 |  | $10^{\text {h }}$ |
| 0x3104/x | Reserved | 0 |  |
| ... | ... | ... |  |
| 0x311D/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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 . . .10 \mathrm{~V} \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 27648 \\ & \mathrm{D}=27648 * \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | $13824$ | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \text { V }}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

### 8.8.3 2 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S402

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| $0 \times 3100 / \mathrm{x}$ <br> $0 \times 3101 / \mathrm{x}$ | Function channel 1 <br> Function channel 2 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0 . . .{16384_{\text {dec }}}$ $49\left(31_{h}\right): 0 \ldots 20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{h}$ <br> $31_{\text {h }}$ |
| 0x3102/x | Reserved | 0 |  |
| ... | ... | ... |  |
| 0x311D/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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{h}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648 *(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{h}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

### 8.8.4

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

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0 . . .{16384_{\mathrm{dec}}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ 255 ( $\mathrm{FF}_{\mathrm{h}}$ ): channel deactivated | $31_{\text {h }}$ |
| 0x3101/x | Function channel 2 |  | $31_{\text {h }}$ |
| 0x3102/x | Function channel 3 |  | $31_{\text {h }}$ |
| 0x3103/x | Function channel 4 |  | $31_{\text {h }}$ |
| 0x3104/x | Reserved | 0 |  |
| ... | ... | ... |  |
| 0x311D/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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | $8192$ | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 16 / 27648+4 \\ & D=27648^{*}(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{\mathrm{h}}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 |  | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

8.8.5

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

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Diagnostics | Bits 0 ... 5: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 0x3101/x | Reserved | 0 |  |
| 0x3102/x | Limit value monitoring | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 1: Channel 2 ( $0=$ inhibited; $1=$ enabled $)$ <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |
| 0x3103/x | Interference frequency suppression | Bit 1, 0: Channel 1 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bit 3, 2: Channel 2 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 0x3104/x | Function channel 1 | $\begin{aligned} & 12\left(12_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots 27648_{\mathrm{dec}} \\ & 34\left(22_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-16384 \ldots 16384_{\mathrm{dec}} \\ & 16(10 \mathrm{~h}): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\text {dec }} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots . . .16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
| 0x3105/x | Reserved | 0 |  |
| $\begin{aligned} & 0 x 3106 / x \\ & 0 \times 3107 / x \end{aligned}$ | 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. <br> 7FFF ${ }_{h}$ : Limit value alarm deactivated | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| $\begin{aligned} & 0 \times 3108 / x \\ & 0 \times 3109 / x \end{aligned}$ | 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. $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |
| 0x310A/x | Function channel 2 | $\begin{aligned} & 12\left(12_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots . .27648_{\mathrm{dec}} \\ & 34\left(22_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-16384 \ldots . .16384_{\text {dec }} \\ & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots . .27648_{\text {dec }} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots . .16384_{\text {dec }} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
| 0x310B/x | Reserved | 0 |  |
| $\begin{aligned} & 0 \times 310 C / x \\ & 0 \times 310 D / x \end{aligned}$ | 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. <br> 7FFF ${ }_{h}$ : Limit value alarm deactivated | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| $\begin{aligned} & 0 \times 310 E / x \\ & 0 \times 310 F / x \end{aligned}$ | 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(12_{\mathrm{h}}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 |  | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -13824 | CA00 |  |  |
|  | -10 | -27648 | 9400 |  |  |
|  | -11.76 | -32512 | 8100 | Underflow |  |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(22_{\mathrm{h}}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 | Nominal range |  |
|  | 5 | 8192 | 2000 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -8192 | E000 |  |  |
|  | -10 | -16384 | C000 |  |  |
|  | -12.5 | -20480 | B000 | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -1.76 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 | Nominal range |  |
|  | 5 | 8192 | 2000 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -2 | -3277 | F333 | Underflow |  |

### 8.8.6

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

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Diagnostics | Bits 0 ... 5: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 0x3101/x | Reserved | 0 |  |
| 0x3102/x | Limit value monitoring | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated $)$ <br> Bit 1: Channel 2 ( $0=$ deactivated; $1=$ activated) <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |
| 0x3103/x | Interference frequency suppression | $\begin{aligned} & \text { Bit 1, } 0: \text { Channel } 1 \\ & \text { 00: Deactivated } \\ & 01: 60 \mathrm{~Hz} \\ & \text { 10: } 50 \mathrm{~Hz} \\ & \text { Bit 3, } 2: \text { Channel } 2 \\ & 00: \text { Deactivated } \\ & 01: 60 \mathrm{~Hz} \\ & \text { 10: } 50 \mathrm{~Hz} \\ & \text { Bits } 7 \ldots 4 \text { Reserved } \end{aligned}$ | $00_{\text {h }}$ |
| 0x3104/x | Function channel 1 | $48\left(30_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0 . . .{16384_{\text {dec }}}$ 49 (31h): 0 ... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ 65 ( $41_{\mathrm{h}}$ ): 0 ... $20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $41_{\text {h }}$ |
| 0x3105/x | Reserved | 0 |  |
| $\begin{aligned} & 0 \times 3106 / x \\ & 0 \times 3107 / x \end{aligned}$ | 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. <br> 7FFF $h$ : Limit value alarm deactivated | $7 \mathrm{FFF}_{\text {h }}$ |
| $\begin{aligned} & 0 \times 3108 / x \\ & 0 \times 3109 / x \end{aligned}$ | 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. <br> $8000_{\mathrm{h}}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |
| 0x310A/x | Function channel 2 | $48\left(30_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots$ 27648 $_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $41_{\text {h }}$ |
| 0x310B/x | Reserved | 0 |  |
| $\begin{aligned} & 0 \times 310 C / x \\ & 0 \times 310 D / x \end{aligned}$ | 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. <br> 7FFF $h$ : Limit value alarm deactivated | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| $\begin{aligned} & 0 \times 310 E / x \\ & 0 \times 310 F / x \end{aligned}$ | 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 10 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -3.52 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 10 | 8192 | 2000 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -4 | -3277 | F333 | Underflow |  |
| $\underset{\left(30_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648 *(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 12 | 13824 | 3600 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 1,19 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{\mathrm{h}}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0.8 | -3277 | F333 | Underflow |  |

8.8.7 2 analog outputs $0 \ldots 10 \mathrm{~V}$ (12 bits) - EPM-S500

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Reserved | 0 |  |
| 0x3101/x | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; 1 = activated) <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | 00 h |
| 0x3102/x | Function channel 1 | $\begin{aligned} & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots . .27648_{\mathrm{dec}} \\ & 32\left(20_{\mathrm{h}}\right): 0 . .10 \mathrm{~V} / 0 \ldots . .16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $10_{\text {h }}$ |
| 0x3103/x | Function channel 2 |  | $10_{\text {h }}$ |
| 0x3104/x | Reserved |  |  |
| ... | ... | $\ldots$ |  |
| 0x311D/x | Reserved | 0 |  |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | $13824$ | $3600$ |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \mathrm{~V}}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

### 8.8.8 <br> 4 analog outputs 0 ... 10 V (12 bits) - EPM-S501

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Reserved | 0 |  |
| 0x3101/x | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; 1 = activated ) <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 0x3102/x | Function channel 1 | $\begin{aligned} & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}} \\ & 32(20 \mathrm{~h}): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right) \text { : channel deactivated } \end{aligned}$ | $10_{\text {h }}$ |
| 0x3103/x | Function channel 2 |  | $10_{\text {h }}$ |
| 0x3104/x | Function channel 3 |  | $10_{\text {h }}$ |
| 0x3105/x | Function channel 4 |  | $10_{\text {h }}$ |
| 0x3106/x | Reserved | 0 |  |
| ... | ... | ... |  |
| 0x311D/x | Reserved | 0 |  |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

### 8.8.9 <br> 2 analog outputs 0/4 ... 20 mA (12 bits) - EPM-S502

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Reserved | 0 |  |
| 0x3101/x | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 0x3102/x | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0 . . .{27648_{d e c}}$ <br> $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0$... ${16384_{\text {dec }}}$ <br> $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ <br> $65\left(41_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ <br> $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |
| 0x3103/x | Function channel 2 |  | $31_{\text {h }}$ |
| 0x3104/x | Reserved |  |  |
| ... | ... | ... |  |
| 0x311D/x | Reserved | 0 |  |

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.) | $\begin{aligned} & \text { Current (I) } \\ & {[\mathrm{mA}]} \end{aligned}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{\mathrm{h}}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} \mathrm{I} / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 12 | 13824 | 3600 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\underset{\left(40_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

### 8.8.10

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

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Reserved | 0 |  |
| 0x3101/x | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated $)$ <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 0x3102/x | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... 27648 ${ }_{\text {dec }}$ $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0$... ${16384_{\text {dec }}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 1638_{\text {dec }}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |
| 0x3103/x | Function channel 2 |  | $31_{\text {h }}$ |
| 0x3104/x | Function channel 3 |  | $31_{\text {h }}$ |
| 0x3105/x | Function channel 4 |  | $31_{\text {h }}$ |
| 0x3106/x | Reserved | 0 |  |
| ... | ... | ... |  |
| 0x311D/x | Reserved | 0 |  |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} \mathrm{I} / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 16 / 27648+4 \\ & D=27648^{*}(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{\mathrm{h}}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

8 CANopen communication
Setting the parameters of analog I/O
Error behaviour
8.8.11 Error behaviour

| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selectio |  |  |  |
| $16443_{h}$ | Error mode analog output |  | 0 | \{1\} 255 | Configures analog output monitoring | [1348 |
|  |  |  |  | All analog outputs retain the last value output |  |  |
|  |  |  | 255 | Response from 16444h | In I6444h the response can be configured individually for each analog output |  |
| 1 | Channel 1 | 0 |  |  |  |  |
| 2 | Channel 2 | 0 |  |  |  |  |
| ... | ... | ... |  |  |  |  |
| 36 | Channel 36 | 0 |  |  |  |  |
| $16444_{h}$ | Error value analog output |  | -32768 | \{1\} 32767 | Configures the individual analog output responses | [1348 |
| 1 | Channel 1 | 0 |  |  | The analog outputs provide the |  |
| 2 | Channel 2 | 0 |  |  |  |  |
| ... | ... | ... |  |  |  |  |
| 36 | Channel 36 | 0 |  |  |  |  |

### 8.9 Parameterising the temperature measurement

8.9.1 Four (two) analog inputs for resistance tests - EPM-S404

Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 0 ... 5: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 0x3101/x | Wire-breakage detection | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{h}$ |
| 0x3102/x | Limit value monitoring | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 3: Channel 4 <br> Bit 4 ... 7: Reserved | $00_{\text {h }}$ |
| 0x3103/x | Reserved |  |  |
| 0x3104/x | Temperature system | Bit 0, 1: $00_{b}={ }^{\circ} \mathrm{C} ; 01_{\mathrm{b}}={ }^{\circ} \mathrm{F} ; 10_{\mathrm{b}}=\mathrm{K}$ Bits 2 ... 7: Reserved | $00_{h}$ |
| 0x3105/x | Interference frequency suppression | Bit 0, 1: $01_{\mathrm{b}}=60 \mathrm{~Hz} ; 10_{\mathrm{b}}=50 \mathrm{~Hz}$ Bits 2 ... 7: Reserved | $02_{\text {h }}$ |

## 8 CANopen communication

Parameterising the temperature measurement
Four (two) analog inputs for resistance tests - EPM-S404

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| Channel 1 |  |  |  |
| 0x3106/x | Function channel 1 | ```Thermal detector: \(80\left(50_{h}\right)\) : PT100 2-wire conductor \(-200^{\circ} \mathrm{C} \ldots+850^{\circ} \mathrm{C} /-2000 \ldots\) \(+8500_{\text {dec }}\) \(81\left(51_{h}\right)\) : PT1000 2 -wire conductor \(-200^{\circ} \mathrm{C} \ldots+850^{\circ} \mathrm{C} /-2000 \ldots\) \(+8500{ }_{\text {dec }}\) \(82\left(52_{h}\right)\) : Ni100 2-wire conductor \(-60^{\circ} \mathrm{C}\)... \(+250^{\circ} \mathrm{C} /-600 \ldots\) +2500 dec \(83\left(53_{h}\right)\) : Ni1000 2 -wire conductor \(-60^{\circ} \mathrm{C} \ldots+250^{\circ} \mathrm{C} /-600 \ldots\) \(+2500_{\text {dec }}\) \(88\left(58_{h}\right)\) : PT100 3-wire conductor \(-200^{\circ} \mathrm{C} . . .+850^{\circ} \mathrm{C} /-2000 \ldots\) \(+8500_{\text {dec }}\) 89 (59h): PT1000 3-wire conductor \(-200^{\circ} \mathrm{C} . . .+850^{\circ} \mathrm{C} /-2000 \ldots\) \(+8500_{\text {dec }}\) \(90\left(5 \mathrm{~A}_{\mathrm{h}}\right)\) : Ni100 3-wire conductor \(-60^{\circ} \mathrm{C}\)... \(+250^{\circ} \mathrm{C} /-600 \ldots\) \(+2500_{\text {dec }}\) \(91\left(5 \mathrm{~B}_{\mathrm{h}}\right)\) : Ni1000 3-wire conductor \(-60^{\circ} \mathrm{C} \ldots+250^{\circ} \mathrm{C} /-600 \ldots\) \(+2500_{\text {dec }}\) \(96\left(60_{\mathrm{h}}\right)\) : PT100 4-wire conductor \(-200^{\circ} \mathrm{C} . . .+850^{\circ} \mathrm{C} /-2000 \ldots\) \(+8500_{\text {dec }}\) \(97\left(61_{h}\right)\) : PT1000 4-wire conductor \(-200^{\circ} \mathrm{C} \ldots+850^{\circ} \mathrm{C} /-2000 \ldots\) \(+8500_{\text {dec }}\) \(98\left(62_{h}\right)\) : Ni100 4-wire conductor \(-60^{\circ} \mathrm{C} . . .+250^{\circ} \mathrm{C} /-600 \ldots\) \(+2500_{\text {dec }}\) \(99\left(63_{h}\right)\) : Ni1000 4-wire conductor \(-60^{\circ} \mathrm{C} \ldots+250^{\circ} \mathrm{C} /-600 \ldots\) +2500 dec Resistor: \(112\left(70_{h}\right)\) : R 60- \(\Omega 2\)-wire conductor 0.00 ... \(+60.00 / 0\)... \(+32767_{\text {dec }}\) 113 ( \(71_{\mathrm{h}}\) ): R 600- \(\Omega 2\)-wire conductor \(0.00 \ldots+600.00 / 0\)... \(+32767_{\text {dec }}\) 114 ( \(72_{h}\) ): R 3000- \(\Omega 2\)-wire conductor \(0.00 \ldots+3000.00 / 0\)... \(+32767_{\text {dec }}\) 115 ( \(73_{h}\) ): R 6000- \(\Omega 2\)-wire conductor \(0.00 \ldots+6000.00 / 0\)... \(+32767_{\text {dec }}\) \(128(80 \mathrm{~h}):\) R 60- \(\Omega 4\)-wire conductor \(0.00 \ldots+60.00 / 0\)... \(+32767_{\mathrm{dec}}\) 129 ( \(81_{\mathrm{h}}\) ): R 600- \(\Omega 4\)-wire conductor \(0.00 \ldots+600.00 / 0\)... \(+32767_{\text {dec }}\) 130 ( 82 h ): R 3000- \(\Omega 4\)-wire conductor \(0.00 \ldots+3000.00 / 0 \ldots\) \(+32767_{\mathrm{dec}}\) \(144\left(90_{h}\right)\) : R 60- \(\Omega 2\)-wire conductor \(0.00 \ldots+60.00 / 0 \ldots+6000_{\text {dec }}\) 145 ( \(91_{h}\) ): R 600- \(\Omega 2\)-wire conductor \(0.00 \ldots+600.00 / 0\)... \(+6000_{\text {dec }}\) \(146\left(92_{h}\right):\) R 3000- 22 -wire conductor \(0.00 \ldots+3000.00 / 0\)... \(+30000_{\text {dec }}\) \(160\left(\mathrm{~A} 0_{h}\right):\) R 60- \(\Omega 2\)-wire conductor \(0.00 \ldots+60.00 / 0 \ldots+6000_{\text {dec }}\) \(161\left(A 1_{h}\right): R 600-\Omega 2\)-wire conductor \(0.00 \ldots+600.00 / 0 \ldots\) \(+6000_{\text {dec }}\) \(162\left(A 2_{h}\right):\) R 3000- \(\Omega 2\)-wire conductor \(0.00 \ldots+3000.00 / 0 \ldots\) \(+30000_{\text {dec }}\) \(255\left(\mathrm{FF}_{\mathrm{h}}\right)\) : channel deactivated``` | $50_{\text {h }}$ |

Parameterising the temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3107/x | Conversion time channel 1 | You can set the transformer speed as a function of the interference frequency suppression (see $0 \times 3105 / \mathrm{x}$ ) for each channel. <br> 0 ( $00_{\mathrm{h}}$ ): At 50 Hz : $324.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $270.5 \mathrm{~ms} /$ channel 16 bits <br> 1 ( $01_{h}$ ): At $50 \mathrm{~Hz}: 164.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $137.2 \mathrm{~ms} /$ channel 16 bits <br> 2 ( $02_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 84.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 70.5 \mathrm{~ms} / \mathrm{channel} 16$ bits 3 ( $03_{\mathrm{h}}$ ): at $50 \mathrm{~Hz}: 44.1 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 37.2 \mathrm{~ms} /$ channel 16 bits 4 (04h): At $50 \mathrm{~Hz}: 24.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 20.5 \mathrm{~ms} /$ channel 16 bits 5 (05h): At $50 \mathrm{~Hz}: 14.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 12.2 \mathrm{~ms} /$ channel 16 bits 6 ( 06 h): At $50 \mathrm{~Hz}: 9.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 8.0 \mathrm{~ms} / \mathrm{channel} 16$ bits 7 ( $07_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 6.6 \mathrm{~ms} /$ channel 15 bits; at $60 \mathrm{~Hz}: 5.9 \mathrm{~ms} /$ channel 15 bits 8 (08h): At $50 \mathrm{~Hz}: 4.2 \mathrm{~ms} /$ channel 13 bits; at $60 \mathrm{~Hz}: 3.8 \mathrm{~ms} /$ channel 13 bits | $00_{\text {h }}$ |
| 0x3108/x | Upper limit value <br> (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 7FFFh for the upper or 8000h 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. | $7 \mathrm{~F}_{\mathrm{h}}$ |
| 0x3109/x | Upper limit value channel 1 (LOW byte) |  | $\mathrm{FF}_{\mathrm{h}}$ |
| 0x310A/x | Lower limit value (HIGH byte) channel 1 |  | $80_{\text {h }}$ |
| 0x310B/x | Lower limit value channel 1 (LOW byte) |  | $00_{\text {h }}$ |
| Channel 2 |  |  |  |
| 0x310C/x | Function channel 2 | See channel 1 | $50_{\text {h }}$ |
| 0x310D/x | Conversion time channel 2 | See channel 1 | $00_{\text {h }}$ |
| 0x310E/x | Upper limit value channel 2 (HIGH byte) | See channel 1 | $7 F_{h}$ |
| 0x310F/x | Upper limit value (LOW byte) channel 2 |  | $\mathrm{FF}_{\mathrm{h}}$ |
| 0x3110/x | Lower limit value channel 2 (HIGH byte) |  | $80_{\text {h }}$ |
| 0x3111/x | Lower limit value (LOW byte) channel 2 |  | $00_{\text {h }}$ |
| Channel 3 (for two-wire conductor connections only) |  |  |  |
| 0x3112/x | Function channel 3 | See channel 1 | $50_{\text {h }}$ |
| 0x3113/x | Conversion time channel 3 | See channel 1 | $00_{\text {h }}$ |
| 0x3114/x | Upper limit value channel 3 (HIGH byte) | See channel 1 | $7 \mathrm{~F}_{\mathrm{h}}$ |
| 0x3115/x | Upper limit value channel 3 (LOW byte) |  | $\mathrm{FF}_{\mathrm{h}}$ |
| 0x3116/x | Lower limit value channel 3 (HIGH byte) |  | $80_{\text {h }}$ |
| 0x3117/x | Lower limit value channel 3 (LOW byte) |  | $00_{\text {h }}$ |

## 8 CANopen communication

## Parameterising the temperature measurement

Four (two) analog inputs for resistance tests - EPM-S404

| Index/sub <br> index | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- |
| Channel 4 (for two-wire conductor connections only) |  |  |  |
| $0 x 3118 / x$ | Function channel 4 | See channel 1 | $50_{h}$ |
| $0 \times 3119 / x$ | Conversion time <br> channel 4 | See channel 1 | $00_{\mathrm{h}}$ |
| $0 \times 311 \mathrm{~A} / \mathrm{x}$ | Upper limit value <br> channel 4 (HIGH byte) |  | $7 \mathrm{~F}_{\mathrm{h}}$ |
| $0 \times 311 \mathrm{~B} / \mathrm{x}$ | Upper limit value <br> channel 4 (LOW byte) | See channel 1 | $\mathrm{FF}_{\mathrm{h}}$ |
| $0 \times 311 \mathrm{C} / \mathrm{x}$ | Lower limit value <br> channel 4 (HIGH byte) |  | $80_{\mathrm{h}}$ |
| $0 \times 311 \mathrm{D} / \mathrm{x}$ | Lower limit value <br> channel 4 (LOW byte) |  | $00_{\mathrm{h}}$ |

(i) Note!

- Use parameter setting to deactivate unused inputs.
- If thermal detectors are connected in a 3 or 4 conductor setup, channels 2 and/or 4 must be deactivated.
- The module does not provide any auxiliary supply for sensors.

Parameterising the temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

Measuring range

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 2-wire: PT100 } \\ \left(50_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| 2-wire: PT1000 (51h) | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 2-wire: NI100 } \\ \left(52_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 2-wire: NI1000 (53h) | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | -1050 dec | Underflow |
| $\begin{gathered} \text { 3-wire: PT100 } \\ \left(58_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | -2430 dec | Underflow |
| 3-wire: PT1000 (59h) | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | -2430 dec | Underflow |
| $\begin{gathered} \text { 3-wire: NI100 } \\ \left(5 A_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | +2950 ${ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 3-wire: NI1000 (5Bh) | $+295{ }^{\circ} \mathrm{C}$ | +2950 ${ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 4-wire: PT100 } \\ \left(60_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| 4-wire: PT1000 (61h) | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430{ }_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 4-wire: NI100 } \\ \left(62_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| $\begin{aligned} & \text { 4-wire: NI1000 } \\ & \left(63_{h}\right) \end{aligned}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(70_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 600 \Omega \\ \left(71_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | 0 ... 32767 ${ }_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(72_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 3000 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |

## 8 CANopen communication

Parameterising the temperature measurement
Four (two) analog inputs for resistance tests - EPM-S404

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(78_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .60 \Omega$ | $0 \ldots 32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(79_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | 0... 32767 ${ }_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(7 A_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | 0 ... $3000 \Omega$ | $0 \ldots 32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(80_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 \ldots 32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots . .600 \Omega \\ \left(81_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 \ldots 32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(82_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | 0 ... 32767 | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(90_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .60 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots . .600 \Omega \\ \left(91_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 \ldots 6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(92_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | 0 ... $3000 \Omega$ | $0 \ldots 30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(98_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .60 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(99_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(9 A_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 \ldots 30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(\mathrm{~A} 0_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 \ldots 6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(\mathrm{~A} 1_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{~A} 2_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 \ldots 30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |

Parameterising the temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(D 0_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . . .60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 600 \Omega \\ \left(\mathrm{D} 1_{\mathrm{h}}\right) \end{gathered}$ | $705.5 \Omega$ | 32511 ${ }_{\text {dec }}$ | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{D} 2_{h}\right) \end{gathered}$ | 3528 ת | $32511_{\text {dec }}$ | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(D 8_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . .60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(D 9_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(\text { DA }_{h}\right) \end{gathered}$ | 3528 ת | 32511 ${ }_{\text {dec }}$ | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(E O_{h}\right) \end{gathered}$ | $70.55 \Omega$ | 32511 ${ }_{\text {dec }}$ | Overflow |
|  | $0 . .60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots . .600 \Omega \\ \left(E 1_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(E 2_{h}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |

### 8.9.2 <br> Two analog inputs for thermocouple measurement - EPM-S405

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 0 ... 5: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{h}$ |
| 0x3101/x | Wire-breakage detection | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 0x3102/x | Limit value monitoring | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 0x3103/x | Reserved | 0 |  |
| 0x3104/x | Temperature system | Bit 0, 1: $00_{b}={ }^{\circ} \mathrm{C} ; 10_{b}={ }^{\circ} \mathrm{F} ; 11_{\mathrm{b}}=\mathrm{K}$ Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 0x3105/x | Interference frequency suppression | Bit 0, 1: $01_{b}=60 \mathrm{~Hz} ; 10_{b}=50 \mathrm{~Hz}$ Bits 2 ... 7: Reserved | 02h |
| Channel 1 |  |  |  |
| 0x3106/x | Function channel 1 | External temperature compensation: <br> $176\left(60_{\mathrm{h}}\right)$ : type J, $-210.0 \ldots+1200.0^{\circ} \mathrm{C} /-2100 \ldots+12000_{\text {dec }}$ <br> $177\left(61_{\mathrm{h}}\right)$ : type K, $-270.0 \ldots+1372.0^{\circ} \mathrm{C} /-2700 \ldots+13720_{\mathrm{dec}}$ <br> $178\left(62_{h}\right)$ : type $\mathrm{N}-270.0 \ldots+1300.0^{\circ} \mathrm{C} /-2700 \ldots+13000_{\text {dec }}$ <br> $179\left(63_{h}\right)$ : type R, $-50.0 \ldots+1769.0^{\circ} \mathrm{C} /-500 \ldots+17690_{\mathrm{dec}}$ <br> $180\left(64_{h}\right)$ : type S, $-50.0 \ldots+1769.0^{\circ} \mathrm{C} /-500 \ldots+17690_{\text {dec }}$ <br> $181\left(65_{h}\right)$ : type T, $-270.0 \ldots+400.0^{\circ} \mathrm{C} /-2700 \ldots+4000_{\text {dec }}$ <br> 182 ( $66_{\mathrm{h}}$ ): type B, $0.0 \ldots+1820.0^{\circ} \mathrm{C} / 0 \ldots+18200_{\text {dec }}$ <br> $183\left(67_{\mathrm{h}}\right)$ : type C, $0.0 \ldots+2315.0^{\circ} \mathrm{C} / 0 \ldots+23150_{\mathrm{dec}}$ <br> $184\left(68_{h}\right)$ : type E, $-270.0 \ldots+1000.0^{\circ} \mathrm{C} /-2700 \ldots+10000_{\text {dec }}$ <br> $185\left(69_{h}\right)$ : type L, $-200.0 \ldots+900.0^{\circ} \mathrm{C} /-2000 \ldots+9000_{\text {dec }}$ <br> Internal temperature compensation: <br> $192\left(\mathrm{CO}_{\mathrm{h}}\right)$ : type J, $-210.0 \ldots+1200.0^{\circ} \mathrm{C} /-2100 \ldots+12000_{\text {dec }}$ <br> $193\left(\mathrm{C} 1_{\mathrm{h}}\right)$ : type K, $-270.0 \ldots+1372.0^{\circ} \mathrm{C} /-2700 \ldots+13720_{\text {dec }}$ <br> $194\left(\mathrm{C} 2_{\mathrm{h}}\right)$ : type $\mathrm{N}-270.0 \ldots+1300.0^{\circ} \mathrm{C} /-2700 \ldots+13000_{\text {dec }}$ <br> $195\left(\mathrm{C}_{\mathrm{h}}\right)$ : type R, $-50.0 \ldots+1769.0^{\circ} \mathrm{C} /-500 \ldots+17690_{\mathrm{dec}}$ <br> $196\left(\mathrm{C} 4_{\mathrm{h}}\right)$ : type S, $-50.0 \ldots+1769.0^{\circ} \mathrm{C} /-500 \ldots+17690_{\text {dec }}$ <br> $197\left(\mathrm{C} 5_{\mathrm{h}}\right)$ : type T, $-270.0 \ldots+400.0^{\circ} \mathrm{C} /-2700 \ldots+4000_{\mathrm{dec}}$ <br> 198 (C6h): type B, $0.0 \ldots+1820.0^{\circ} \mathrm{C} / 0 \ldots+18200_{\text {dec }}$ <br> $199\left(C 7_{\mathrm{h}}\right)$ : type C, $0.0 \ldots+2315.0^{\circ} \mathrm{C} / 0 \ldots+23150_{\mathrm{dec}}$ <br> $200(\mathrm{C} 8 \mathrm{~h})$ : type $\mathrm{E},-270.0 \ldots+1000.0^{\circ} \mathrm{C} /-2700 \ldots+10000_{\mathrm{dec}}$ <br> $201\left(C 9_{h}\right)$ : type L, $-200.0 \ldots+900.0^{\circ} \mathrm{C} /-2000 \ldots+9000_{\text {dec }}$ <br> $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $C 1_{\text {h }}$ |

Parameterising the temperature measurement Two analog inputs for thermocouple measurement - EPM-S405

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3107/x | Conversion time channel 1 | You can set the transformer speed as a function of the interference frequency suppression (see $0 \times 3105 / \mathrm{x}$ ) for each channel. <br> 0 ( $00_{\mathrm{h}}$ ): At 50 Hz : $324.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $270.5 \mathrm{~ms} /$ channel 16 bits <br> 1 ( $01_{\mathrm{h}}$ ): at $50 \mathrm{~Hz}: 164.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $137.2 \mathrm{~ms} /$ channel 16 bits <br> 2 ( $02_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 84.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $70.5 \mathrm{~ms} /$ channel 16 bits <br> 3 ( $03_{\mathrm{h}}$ ): at $50 \mathrm{~Hz}: 44.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $37.2 \mathrm{~ms} /$ channel 16 bits <br> 4 ( 04 h): At $50 \mathrm{~Hz}: 24.2 \mathrm{~ms} / c h a n n e l ~ 16 ~ b i t s ; ~ a t ~ 60 ~ H z: ~$ <br> $20.5 \mathrm{~ms} /$ channel 16 bits <br> 5 ( $05_{\mathrm{h}}$ ): at $50 \mathrm{~Hz}: 14.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $12.2 \mathrm{~ms} /$ channel 16 bits <br> $6\left(06_{h}\right)$ : at 50 Hz : $9.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $8.0 \mathrm{~ms} /$ channel 16 bits <br> $7\left(07_{\mathrm{h}}\right)$ : at 50 Hz : $6.6 \mathrm{~ms} /$ channel 15 bits; at 60 Hz : <br> $5.9 \mathrm{~ms} /$ channel 15 bits <br> $8\left(08_{h}\right)$ : at $50 \mathrm{~Hz}: 4.2 \mathrm{~ms} /$ channel 13 bits; at 60 Hz : <br> $3.8 \mathrm{~ms} /$ channel 13 bits | $02_{\text {h }}$ |
| 0x3108/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 7FFFh for the upper or 8000h for the lower limit value, the corresponding limit value is deactivated. <br> If your measured value is outside a limit value and you have activated the limit value monitoring, a process alarm is triggered. | $7 \mathrm{~F}_{\mathrm{h}}$ |
| 0x3109/x | Upper limit value channel 1 (LOW byte) |  | $\mathrm{FF}_{\mathrm{h}}$ |
| 0x310A/x | Lower limit value (HIGH byte) channel 1 |  | $80_{\text {h }}$ |
| 0x310B/x | Lower limit value channel 1 (LOW byte) |  | $00_{\text {h }}$ |
| Channel 2 |  |  |  |
| 0x310C/x | Function channel 2 | See channel 1 | $\mathrm{Cl}_{\mathrm{h}}$ |
| 0x310D/x | Conversion time channel 2 | See channel 1 | $02_{\text {h }}$ |
| 0x310E/x | Upper limit value channel 2 (HIGH byte) |  | $7 \mathrm{~F}_{\mathrm{h}}$ |
| 0x310F/x | Upper limit value (LOW byte) channel 2 |  | $\mathrm{FF}_{\mathrm{h}}$ |
| 0x3110/x | Lower limit value channel 2 (HIGH byte) |  | $80_{\text {h }}$ |
| 0x3111/x | Lower limit value (LOW byte) channel 2 |  | 00 h |
| 0x3112/x | Reserved | 0 |  |
| ... | ... | ... |  |
| 0x311D/x | Reserved | 0 |  |

## Measuring range

| Measuring range (Fct. no.) | Measured value |  |  | Range |
| :---: | :---: | :---: | :---: | :---: |
|  | [ ${ }^{\text {C }}$ ] | [ $\left.{ }^{\circ} \mathrm{F}\right]$ | [K] |  |
| $\begin{aligned} & \text { Type J: } \\ & \quad-210 \ldots+1200{ }^{\circ} \mathrm{C} \\ & -346 \ldots 2192{ }^{\circ} \mathrm{F} \\ & 63.2 \ldots 1473.2 \mathrm{~K} \\ & \left(\mathrm{BO} 0_{\mathrm{h}}: \text { ext. comp. } 0^{\circ} \mathrm{C}\right) \\ & \left(\mathrm{CO}_{\mathrm{h}} \text { : int. comp. } 0^{\circ} \mathrm{C}\right) \end{aligned}$ | +14500 | 26420 | 17232 | Overflow |
|  | $-2100 \ldots+12000$ | -3460 ... +21920 | 632 ... 14732 | Nominal range |
|  | - | - | - | Underflow |
| $\begin{aligned} & \text { Type K: } \\ & \quad-210 \ldots+13722^{\circ} \mathrm{C} \\ & -454 \ldots . .2501 .6^{\circ} \mathrm{F} \\ & 0 \ldots 1645.2 \mathrm{~K} \\ & \left(\mathrm{~B} 1_{\mathrm{h}}: \text { ext. comp. } 0^{\circ} \mathrm{C}\right) \\ & \left(\mathrm{C} 1_{\mathrm{h}} \text { : int. comp. } 0^{\circ} \mathrm{C}\right) \end{aligned}$ | +16220 | 29516 | 18952 | Overflow |
|  | $-2700 \ldots+13720$ | -4540 ... 25016 | 0 ... 16452 | Nominal range |
|  | - | - | - | Underflow |
| $\begin{aligned} & \text { Type } \mathrm{N}: \\ & \quad-270 \ldots+1300^{\circ} \mathrm{C} \\ & -454 \ldots 2372{ }^{\circ} \mathrm{F} \\ & 0 \ldots 1573.2 \mathrm{~K} \\ & \left(\mathrm{~B} 2_{\mathrm{h}}: \text { ext. comp. } 0^{\circ} \mathrm{C}\right. \text { ) } \\ & \left(\mathrm{C} 2_{\mathrm{h}} \text { : int. comp. } 0^{\circ} \mathrm{C}\right. \text { ) } \end{aligned}$ | +15500 | 28220 | 18232 | Overflow |
|  | $-2700 \ldots+13000$ | -4540 ... 23720 | 0 ... 15732 | Nominal range |
|  | - | - | - | Underflow |
| $\begin{aligned} & \text { Type R: } \\ & \quad-50 \ldots+1769^{\circ} \mathrm{C} \\ & -58 \ldots . .3216 .2^{\circ} \mathrm{F} \\ & 223.2 \ldots 2042.2^{\mathrm{K}} \\ & \left(\mathrm{~B} 3_{\mathrm{h}} \text { : ext. comp. } 0^{\circ} \mathrm{C}\right) \\ & \left(\mathrm{C} 3_{\mathrm{h}} \text { : int. comp. } 0^{\circ} \mathrm{C}\right) \end{aligned}$ | +20190 | 32766 | 22922 | Overflow |
|  | $-500 \ldots+17690$ | -580 ... 32162 | 2232 ... 20422 | Nominal range |
|  | -1700 | -2740 | 1032 | Underflow |
| $\begin{aligned} & \text { Type S: } \\ & \quad-50 \ldots+1769^{\circ} \mathrm{C} \\ & -58 \ldots 3216.2^{\circ} \mathrm{F} \\ & 223.2 \ldots 2042.2 \mathrm{~K} \\ & \left(\mathrm{~B} 4_{\mathrm{h}}: \text { ext. comp. } 0^{\circ} \mathrm{C}\right. \text { ) } \\ & \left(\mathrm{C} 4_{\mathrm{h}} \text { : int. comp. } 0^{\circ} \mathrm{C}\right. \text { ) } \end{aligned}$ | +20190 | 32766 | 22922 | Overflow |
|  | $-500 \ldots+17690$ | -580 ... 32162 | 2232 ... 20422 | Nominal range |
|  | -1700 | -2740 | 1032 | Underflow |
| $\begin{aligned} & \text { Type T: } \\ & \quad-270 \ldots+440{ }^{\circ} \mathrm{C} \\ & -454 \ldots 752^{\circ} \mathrm{F} \\ & 3.2 \ldots 673.2 \mathrm{~K} \\ & \left(\mathrm{~B} 2_{\mathrm{h}}: \text { ext. comp. } 0^{\circ} \mathrm{C}\right. \text { ) } \\ & \left(\mathrm{C} 2_{\mathrm{h}} \text { : int. comp. } 0^{\circ} \mathrm{C}\right) \end{aligned}$ | +5400 | 10040 | 8132 | Overflow |
|  | $-2700 \ldots+4000$ | -4540 ... 7520 | $32 . . .6732$ | Nominal range |
|  | - | - | - | Underflow |
| $\begin{aligned} & \text { Type B: } \\ & 0 \ldots+1820{ }^{\circ} \mathrm{C} \\ & 32 \ldots 2786.5^{\circ} \mathrm{F} \\ & 273.2 \ldots 2093.2 \mathrm{~K} \\ & \left(\mathrm{~B} 6_{\mathrm{h}}: \text { ext. comp. } 0^{\circ} \mathrm{C}\right. \text { ) } \\ & \left(\mathrm{C} 6_{\mathrm{h}}: \text { int. comp. } 0^{\circ} \mathrm{C}\right. \text { ) } \end{aligned}$ | +20700 | 32766 | 23432 | Overflow |
|  | 0 ... +18200 | 320 ... 27865 | 2732 ... 20932 | Nominal range |
|  | -1200 | -1840 | 1532 | Underflow |
| ```Type C: 0 ... +2315 % C 32 .. 2786.5  273.2 ... 2093.2 K (B7h (C7h: int. comp. }0\mp@subsup{}{}{\circ}\textrm{C}\mathrm{ )``` | +25000 | 32766 | 23432 | Overflow |
|  | $0 \ldots+23150$ | 320 ... 27865 | 2732 ... 20932 | Nominal range |
|  | -1200 | -1840 | 1532 | Underflow |
| $\begin{aligned} & \text { Type E: } \\ & \quad-270 \ldots+1000{ }^{\circ} \mathrm{C} \\ & -454 \ldots 1832{ }^{\circ} \mathrm{F} \\ & 0 \ldots .1273 .2 \mathrm{~K} \\ & \left(\mathrm{~B} 8_{\mathrm{h}}: \text { ext. comp. } 0^{\circ} \mathrm{C}\right. \text { ) } \\ & \left(\mathrm{C}_{\mathrm{h}} \text { : int. comp. } 0^{\circ} \mathrm{C}\right. \text { ) } \end{aligned}$ | +12000 | 21920 | 14732 | Overflow |
|  | $-2700 \ldots+10000$ | -4540 ... 18320 | 0 ... 12732 | Nominal range |
|  | - | - | - | Underflow |


| Measuring range | Measured value |  |  | Range |
| :---: | :---: | :---: | :---: | :---: |
| (Fct. no.) | [ ${ }^{\circ} \mathrm{C}$ ] | [ $\left.{ }^{\circ} \mathrm{F}\right]$ | [K] |  |
| Type L: $-200 \ldots+900^{\circ} \mathrm{C}$ | +11500 | 21020 | 14232 | Overflow |
| $\begin{aligned} & -328 \ldots 1652^{\circ} \mathrm{F} \\ & 73.2 \ldots 1173.2 \mathrm{~K} \end{aligned}$ | $-2000 \ldots+9000$ | -3280 ... 16520 | 732 ... 11732 | Nominal range |
| (B9 ${ }^{\mathrm{h}}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C9h: int. comp. $0^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |

### 8.9.3 Error behaviour

| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selectio |  |  |  |
| $16443_{h}$ | Error mode analog output |  | 0 | \{1\} 255 | Configures analog output monitoring | -1 348 |
|  |  |  |  | All analog outputs retain the last value output |  |  |
|  |  |  | 255 | Response from 16444h | In I6444h the response can be configured individually for each analog output |  |
| 1 | Channel 1 | 0 |  |  |  |  |
| 2 | Channel 2 | 0 |  |  |  |  |
| ... | ... | ... |  |  |  |  |
| 36 | Channel 36 | 0 |  |  |  |  |
| I6444h | Error value analog output |  | -32768 | \{1\} 32767 | Configures the individual analog output responses | [1348 |
| 1 | Channel 1 | 0 |  |  | The analog outputs provide the |  |
| 2 | Channel 2 | 0 |  |  |  |  |
| ... | ... | ... |  |  |  |  |
| 36 | Channel 36 | 0 |  |  |  |  |

### 8.10 Parameterising the counter

### 8.10.1 Commissioning examples

The indexes for the signal setting in the following examples refer to the EPM-S600. In the case of different counters the function selection can be placed on a different index ( $\mathbb{C}$ Parameter data of the counter).

## Example 1: Counting upwards

| Step | Setting | Comment |
| :---: | :--- | :--- |
| $\mathbf{1}$ | Set signal evaluation: <br> Index $0 \times 310 \mathrm{~A} / \mathrm{x}$, bit $2 \ldots 0=100_{\mathrm{b}}$ (direction) | A signal evaluation has to be specified, otherwise <br> the counting process is not started. |
| $\mathbf{2}$ | Accept parameter setting: <br> $0 \times 31 \mathrm{FF}=255_{\text {dec }}$ | No storage into the EEPROM of the bus coupler. |
| 3 | Enable of the software gates via control word <br> ■ $266:$ 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: <br> Index $0 \times 310 \mathrm{~A} / \mathrm{x}$, bit $2 \ldots 0=100_{\mathrm{b}}$ (direction) | A signal evaluation has to be specified, otherwise <br> the counting process is not started. |
| 2 | Accept parameter setting: <br> $0 \times 31 F F=255_{\text {dec }}$ | No storage into the EEPROM of the bus coupler. |
| 3 | Transfer set value to the counter via PDO: <br> e.g. $2147483583_{\text {dec }} \rightarrow$ Tx-PDO $=\square \square \square \square\|7 F\| F F\|F F\| B F ~$ |  |
| 4 | Activation of the set value via control word <br> Ca $266:$ Bit 5 (COUNTERval_SET) $=$ HIGH | Transfer control word via PDO to the counter. |

## 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 $0 \times 310 \mathrm{~A} / \mathrm{x}$, bit $2 \ldots 0=100_{\mathrm{b}}$ (direction) | A signal evaluation has to be specified, otherwise the counting process is not started. |
| 2 | Set comparator: Index $0 \times 3109 / x$, bit $2 \ldots 0=010_{b}$ (count value $\leq$ comparison value) |  |
| 3 | Accept parameter setting: $0 \times 31 \mathrm{FF}=255_{\mathrm{dec}}$ | No storage into the EEPROM of the bus coupler. |
| 4 | Transfer comparison value to the counter via PDO: e.g. $255_{\text {dec }} \rightarrow$ Tx-PDO $=F F\|00\| 00\|00\| 00$ |  |
| 5 | Enable digital output via control word $■$ 266: Bit 1 (CTRL_DO_SET) $=\mathrm{HIGH}$ | Transfer control word to the counter via PDO. |

### 8.10.2 One counter 32 bits, 24 V DC - EPM-S600

The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> range. |
| Counting periodically |  | | Signal evaluation |
| :--- |
| Single rotary <br> transducer |
| Double rotary <br> transducer |
| Quadruple rotary <br> transducer |
| Connection to input "A/pulse" and "B/direction" |


| Additional functions | Description |
| :--- | :--- |
| Main counting | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> Forwards: Limitation of the counting range upwards. The counter counts from the <br> parameterised loading value in the positive direction to the parameterised final value -1 <br> and then skips to the loading value again with the encoder pulse that is following. <br> Backwards: Limitation of the counting range downwards. The counter counts from the <br> parameterised loading value in the negative direction to the parameterised final value +1 <br> 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 <br> this counter a differentiation between the internal gate (l-gate), hardware gate (HW <br> gate), and software gate (SW gate) is made. <br> - The l-gate is the AND logic operation of the software gate (SW gate) and the hardware <br> gate (HW gate). <br> - The SW gate is controlled via your user program (status word in the output area). <br> - The HW gate is controlled via the digital gate input. |
| The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process <br> continues from the loading value again. <br> Interrupting gate function:After closing the gate and opening it again, the counting <br> process continues with the last current counter content. |  |
| If a positive edge occurs at the latch input, the current count value is stored in the latch <br> register. The latch register is accessed via the input area. After a STOP-RUN transition, <br> latch is always 0. |  |
| You can specify a comparison value which, depending on the count value, activates the <br> digital output or triggers a process alarm. |  |
| By specification of a hysteresis you can for instance prevent the output from being <br> permanently switched if the value of an encoder signal fluctuates around the comparison <br> value. |  |
| Hysteresis <br> The activation of a process alarm can be parameterised. A process alarm can be triggered <br> in the case of the following events: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |  |
| 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. |  |

Further information can be found in the chapter "Product description".

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 5 ... 0: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | 00 h |
| 0x3101/x | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02 h |
| 0x3102/x | Input frequency track B |  | $02_{h}$ |
| 0x3103/x | Input frequency latch |  | 02 h |
| 0x3104/x | Input frequency gate |  | $02{ }_{\text {h }}$ |
| 0x3105/x | Input frequency reset |  | $00_{h}$ |
| 0x3106/x | Reserved |  |  |
| 0x3107/x | Alarm response | Setting activates process alarm Bit 0: Proc. alarm HW gate open <br> Bit 1: Proc. alarm HW gate closed <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bit 6: Proc. alarm latch value <br> Bit 7: Reserved | $80_{\text {h }}$ |
| 0x3108/x | Counter function | ```Bits 5 ... 0: 000000 000001 b = counting once, main counting direction forwards 000010}b= counting once, main counting direction backwards 000100 b = counting once, no main counting direction 001000 forwards 010000}b= counting periodically, main counting directio backwards 100000b = counting periodically, no main counting direction Bits 7 ... 6: Reserved``` | $40_{\text {h }}$ |
| 0x3109/x | Comparator | ```Bit 2 ... 0: output switches (... if condition is met) \(000_{b}=\) never \(001_{b}=\) count value \(\geq\) comparison value \(010_{b}=\) count value \(\leq\) comparison value \(100_{b}=\) count value \(=\) comparison value Bit 3: Invert counting direction track B \(0=\) no (do not invert) 1 = yes (invert) Bits 6 ... 4: Reset \(000_{b}=\) deactivated \(001_{b}=\) HIGH level \(011_{b}=\) rising edge \(101_{b}=\) rising edge, once Bit 7: Reserved``` | $00_{\text {h }}$ |


| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x310A/x | Signal evaluation | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at $A$ and $B$ ) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bit 6 ... 3: Hardware gate (HW gate) <br> $000_{b}=$ deactivated (counter starts by setting SW gate) <br> $001_{b}=$ activated (HIGH level at gate activates the HW gate. <br> Counter starts if HW and SW gate are set.) <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 310 B / x \\ & \ldots \\ & 0 \times 310 E / x \end{aligned}$ | Final value | Counting method: <br> 0x310B: byte 3 (high byte) <br> 0x310C: byte 2 <br> 0x310D: byte 1 <br> $0 \times 310 \mathrm{E}$ : byte 0 (low byte) | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 310 F / x \\ & \ldots \\ & 0 \times 3112 / x \end{aligned}$ | Loading value | Counting method: <br> 0x310B: byte 3 (high byte) <br> 0x3110: byte 2 <br> 0x3111: byte 1 <br> $0 \times 3112$ : byte 0 (low byte) | $00_{\text {h }}$ |
| 0x3113/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. <br> The hysteresis has an effect on the zero crossing, comparison, overflow/underflow. | $00_{\text {h }}$ |
| 0x3114/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. | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 3115 / x \\ & \ldots \\ & 0 \times 3129 / x \end{aligned}$ | Reserved |  |  |

### 8.10.3 Two counters 32 bits, 24 V DC - EPM-S601

The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> Counting periodically <br> range. |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input " $\mathrm{A} / \mathrm{pulse}$ " and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

Further information can be found in the chapter "Product description".

## Parameterising the counter

Two counters 32 bits, 24 V DC - EPM-S601

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; 1 = enabled) <br> Bit 7: Reserved <br> Other values are not permissible! | 00 h |
| 0x3101/x | Input frequency counter 1, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02 h |
| 0x3102/x | Input frequency counter 1, track B |  | $02_{h}$ |
| 0x3103/x | Input frequency counter 2, track A |  | 02 h |
| 0x3104/x | Input frequency counter 2, track B |  | $02_{h}$ |
| 0x3105/x | Alarm response counter 1 | Setting activates process alarm Bits 1 ... 0: reserved <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bits 7 ... 6: Reserved | 00 h |
| 0x3106/x | Counter function counter 1 | ```Bits 5 ... 0: 000000 000001 b = once: forwards 000010b = once: backwards 000100 b = once: no main counting direction 001000b 0100000b = periodically: backwards 100000}b=\mathrm{ periodically: no main counting direction Bits 7 ... 6: Reserved``` | $00_{\text {h }}$ |
| 0x3107/x | Comparator counter 1 | Bits $2 \ldots 0$ : Comparison bit is set (... if condition is met) <br> $000_{b}=$ never <br> $001_{b}=$ count value $\geq$ comparison value <br> $010_{b}=$ count value $\leq$ comparison value <br> $100_{b}=$ count value $=$ comparison value <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 0x3108/x | Signal evaluation counter 1 | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at A and B) <br> $011_{b}=$ rotary transducer quadruple (at A and B) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: Reserved <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{\text {h }}$ |


| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| $\begin{aligned} & 0 \times 3109 / x \\ & \ldots \\ & 0 \times 310 C / x \end{aligned}$ | Set value counter 1 | Counting method: <br> 0x3109: byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> $0 \times 310 \mathrm{C}$ : byte 0 (low byte) | $00_{\text {h }}$ |
| $\begin{aligned} & 0 x 310 D / x \\ & \ldots \\ & 0 \times 3110 / x \end{aligned}$ | Final value counter 1 | Counting method: <br> 0x310D: byte 3 (high byte) <br> 0x310E: byte 2 <br> 0x310F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 3111 / x \\ & \ldots \\ & 0 \times 3114 / x \end{aligned}$ | Loading value counter 1 | Counting method: <br> 0x3111: byte 3 (high byte) <br> 0x3112: byte 2 <br> 0x3113: byte 1 <br> $0 \times 3114$ : byte 0 (low byte) | $00_{h}$ |
| 0x3115/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. <br> The hysteresis has an effect on the zero crossing, comparison, overflow/underflow. | $00_{\text {h }}$ |
| 0x3116/x | Reserved |  |  |
| 0x3117/x | Alarm response counter 2 | See counter 1 | $00_{h}$ |
| 0x3118/x | Counter function counter 2 | See counter 1 | $00_{\text {h }}$ |
| 0x3119/x | Comparator counter 2 | See counter 1 | $00_{\text {h }}$ |
| 0x311A/x | Signal evaluation counter 2 | See counter 1 | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 311 B / x \\ & \ldots \\ & 0 \times 311 E / x \end{aligned}$ | Set value counter 2 | See counter 1 | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 311 F / x \\ & \ldots \\ & 0 \times 3122 / x \end{aligned}$ | Final value counter 2 | See counter 1 | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 3123 / x \\ & \ldots \\ & 0 \times 3126 / x \end{aligned}$ | Loading value counter 2 | See counter 1 | $00_{\text {h }}$ |
| 0x3127/x | Hysteresis counter 2 | See counter 1 | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 3128 / x \\ & 0 \times 3129 / x \end{aligned}$ | Reserved |  |  |

The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> Counting periodically <br> range. |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input " $\mathrm{A} / \mathrm{pulse}$ " and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

Further information can be found in the chapter "Product description".

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; 1 = enabled) <br> Bit 7: Reserved <br> Other values are not permissible! | 00 h |
| 0x3101/x | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02h |
| 0x3102/x | Input frequency track B |  | $02_{h}$ |
| 0x3103/x | Input frequency reset |  | $02^{\text {h }}$ |
| 0x3104/x | Reserved |  |  |
| 0x3105/x | Alarm response | Setting activates process alarm <br> Bits 1 ... 0: reserved <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bits 7 ... 6: Reserved | 00 h |
| 0x3106/x | Counter function | ```Bits 5 ... 0: 000000 000001 b = once: forwards 000010b = once: backwards 000100 b = once: no main counting direction 001000 b = periodically: forwards 0100000b = periodically: backwards 100000b}=\mathrm{ periodically: no main counting direction Bits 7 ... 6: Reserved``` | 00 h |
| 0x3107/x | Comparator | ```Bits 2 ... 0: Comparison bit is set (... if condition is met) 000b = never 001b}=\mathrm{ count value }\geq\mathrm{ comparison value 010b}=\mathrm{ count value }\leq\mathrm{ comparison value 100}b=\mathrm{ count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bits 6 ... 4: Reset 000 001b}=\mathrm{ HIGH level 011b}= = rising edg 101b}= rising edge, onc Bit 7: Reserved``` | $00_{\text {h }}$ |

## Parameterising the counter

One counter 32 bits, 5 V DC - EPM-S602

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3108/x | Signal evaluation | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at $A$ and $B$ ) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: Reserved <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 3109 / x \\ & \ldots \\ & 0 \times 310 C / x \end{aligned}$ | Final value | Counting method: <br> 0x310D: byte 3 (high byte) <br> 0x310E: byte 2 <br> 0x311F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 310 D / x \\ & \ldots \\ & 0 \times 3110 / x \end{aligned}$ | Loading value | Counting method: 0x3109: byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> $0 \times 310 \mathrm{C}$ : byte 0 (low byte) | $00_{\text {h }}$ |
| 0x3111/x | Hysteresis |  | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 3112 / x \\ & \ldots \\ & 0 \times 3129 / x \end{aligned}$ | Reserved |  | $00_{\text {h }}$ |

### 8.10.5 Two counters 32 bits, 24 V DC - EPM-S603

The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from 0 to the counting limit, then skips to the opposite counting limit <br> and continues to count from there. |
|  | Description |
| Signal evaluation <br> Single rotary <br> transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary <br> transducer | Pulse at input "A/pulse" and direction at "B/direction" |
| Quadruple rotary <br> transducer |  |
| Direction |  |

## Additional functions Description

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

Further information can be found in the chapter "Product description".

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Input frequency counter 0, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02 h |
| 0x3101/x | Input frequency counter 0, track B |  | $02^{\text {h }}$ |
| 0x3102/x | Input frequency counter 1, track A |  | $02_{\text {h }}$ |
| 0x3103/x | Input frequency counter 1, track B |  | 02h |
| 0x3104/x | Counting direction counter 0, track B | Bits 2 ... 0 : Reserved <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | 00 h |
| 0x3105/x | Signal evaluation counter 0 | ```Bits 2 ... 0: Signal evaluation 000 the counter are ignored) 001b}=\mathrm{ rotary transducer single (at A and B) 010 011b}= rotary transducer quadruple (at A and B) 100b}=\mathrm{ direction (pulse at A and direction at B) Bits 7 ... 3: Reserved``` | 00 h |
| 0x3106/x | Counting direction counter 1, track B | Bits 2 ... 0: Reserved <br> Bit 3: Invert counting direction track B <br> 0 = no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | 00 h |
| 0x3107/x | Signal evaluation counter 1 | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at $A$ and $B$ ) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 7 ... 3: Reserved | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 3108 / x \\ & 0 \times 3129 / x \end{aligned}$ | Reserved |  |  |

### 8.11 Parameterising the encoder evaluation

### 8.11.1 <br> SSI - EPM-S604

Further information can be found in the chapter "Product description".
Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 5 ... 0: reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 0x3101/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. <br> HIGH LOW <br> $00_{h} 30_{h}: 1 \mu \mathrm{~s}$ <br> $00_{h} 60_{h}: 2 \mu \mathrm{~s}$ <br> $00_{h} \mathrm{CO}_{\mathrm{h}}: 4 \mu \mathrm{~s}$ <br> $01_{h} 80_{h}: 8 \mu \mathrm{~s}$ <br> $03_{h} 00_{h}: 16 \mu \mathrm{~s}$ <br> $06_{h} 00_{h}: 32 \mu s$ <br> $09_{\mathrm{h}} 00_{\mathrm{h}}: 48 \mu \mathrm{~s}$ <br> $0 C_{h} 00_{h}: 64 \mu s$ | OCh |
| 0x3102/x | Dead time LOW byte |  | $00_{\text {h }}$ |
| 0x3103/x | Baud rate HIGH byte | In the "Monitoring operation" mode, the baud rate is irrelevant. Enter the baud rate here. This corresponds to the clock frequency via which the connected encoder communicates. More information on this can be found in the data sheet for your encoder. <br> HIGH LOW <br> $00_{h} 18_{\mathrm{h}}$ : 2 MHz <br> $00_{h} 20_{h}: 1.5 \mathrm{MHz}$ <br> $00_{\mathrm{h}} 30_{\mathrm{h}}: 1 \mathrm{MHz}$ <br> $00_{\mathrm{h}} 60_{\mathrm{h}}: 500 \mathrm{kHz}$ <br> $00_{h} \mathrm{CO}_{\mathrm{h}}: 250 \mathrm{kHz}$ <br> $01_{h} 80_{h}: 125 \mathrm{kHz}$ | $01_{\text {h }}$ |
| 0x3104/x | Baud rate LOW byte |  | $80_{\text {h }}$ |
| 0x3105/x | Reserved |  |  |
| 0x3106/x | Scaling | Depending on the encoder, further bits are transmitted in addition to the encoder value. Scaling serves to determine how many bits postpositioned 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: $00_{h} \ldots 0 F_{h}=0$ bits ... 15 bits | $00_{\text {h }}$ |


| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3107/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. $\begin{aligned} & 7\left(07_{h}\right)=" 8 \text { bits" } \\ & 8\left(08_{h}\right)=" 9 \text { bits" } \\ & 9\left(09_{h}\right)=" 10 \text { bits" } \\ & 10\left(0 A_{h}\right)=" 11 \text { bits" } \\ & 11\left(0 B_{h}\right)=" 12 \text { bits" } \\ & 12\left(0 C_{h}\right)=" 13 \text { bits" } \\ & 13\left(0 D_{h}\right)=" 14 \text { bits" } \\ & \left.140 E_{h}\right)=" 15 \text { bits" } \\ & 15\left(0 F_{h}\right)=" 16 \text { bits" } \\ & 16\left(10_{h}\right)=" 17 \text { bits" } \\ & 17\left(11_{h}\right)=" 18 \text { bits" } \\ & 18\left(12_{h}\right)=" 19 \text { bits" } \\ & 19\left(13_{h}\right)=" 20 \text { bits" } \\ & 20\left(14_{h}\right)=" 21 \text { bits" } \\ & 21\left(15_{h}\right)=" 22 \text { bits" } \\ & 22\left(16_{h}\right)=" 23 \text { bits" } \\ & 23\left(17_{h}\right)=" 24 \text { bits" } \\ & 24\left(18_{h}\right)=" 25 \text { bits" } \\ & 25\left(19_{h}\right)=" 26 \text { bits" } \\ & 26\left(1 A_{h}\right)=" 27 \text { bits" } \\ & 27\left(1 B_{h}\right)=" 28 \text { bits" } \\ & 28\left(1 C_{h}\right)=" 29 \text { bits" } \\ & 29\left(1 D_{h}\right)=" 30 \text { bits" } \\ & 30\left(1 E_{h}\right)=" 31 \text { bits" } \\ & 31\left(1 F_{h}\right)=" 32 \text { bits" } \end{aligned}$ | $18_{\text {h }}$ |


| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3108/x |  | Bit 1 ... 0: ready for operation <br> 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. <br> In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder. <br> $01_{b}=$ monitoring operation <br> $10_{b}=$ master mode <br> Bit 2: shifting direction <br> 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". <br> $0=$ LSB first (LSB is transmitted first) <br> $1=$ MSB first (MSB is transmitted first) <br> Bit 3: edge clock signal <br> 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. <br> 0 = falling edge <br> 1 = rising edge <br> Bit 4: coding <br> 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. <br> $0=$ standard code <br> 1 = Gray code <br> Bits 7 ... 5: reserved | $1 E_{h}$ |
| $\begin{aligned} & 0 \times 3109 / x \\ & \ldots \\ & 0 \times 310 B / x \end{aligned}$ | Reserved |  |  |
| 0x310C/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. <br> $0\left(00_{h}\right)=$ inhibited <br> $1\left(01_{h}\right)=$ enabled | 00 h |
| $\begin{aligned} & 0 \times 310 D / x \\ & \ldots \\ & 0 \times 3129 / x \end{aligned}$ | Reserved |  |  |

### 8.12 Time stamp parameterising

8.12.1 2 digital inputs with time stamp function - EPM-S207

Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/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. | $\begin{aligned} & 14_{h} \text { or } 3 C_{h} \\ & \text { (fix) } \end{aligned}$ |
| 0x3101/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. | $00_{\text {h }}$ (fix) |
| 0x3102/x | Input delay DI1 | $\begin{aligned} & 00_{\mathrm{h}}=1 \mu \mathrm{~s} \\ & 02_{\mathrm{h}}=3 \mu \mathrm{~s} \\ & 04_{\mathrm{h}}=10 \mu \mathrm{~s} \end{aligned}$ | 02 h |
| 0x3103/x | Input delay DI2 | $\begin{aligned} & 07_{\mathrm{h}}=86 \mu \mathrm{~S} \\ & 09_{\mathrm{h}}=342 \mu \mathrm{~s} \\ & 0 \mathrm{C}_{\mathrm{h}}=273 \mu \mathrm{~s} \end{aligned}$ <br> Other values are not permissible. | 02h |
| 0x3104/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 | $00_{\text {h }}$ |
| 0x3105/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 | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 3106 / x \\ & \ldots \\ & 0 \times 3129 / x \end{aligned}$ | Reserved |  |  |

### 8.12.2 2 digital outputs with time stamp function - EPM-S310

## Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/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. | $\begin{aligned} & 14_{h} \text { or } 3 C_{h} \\ & (\mathrm{fix}) \end{aligned}$ |
| 0x3101/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. | $00_{\text {h }}$ (fix) |
| $\begin{aligned} & 0 \times 3102 / x \\ & \ldots \\ & 0 \times 3129 / x \end{aligned}$ | Reserved |  |  |

### 8.13 Parameterising technology modules

8.13.1 2 digital outputs with PWM functionality - EPM-S620

Parameter data

| Index/sub index | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 0x3100/x | PWM 1: <br> 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. <br> Values below $25 \mu$ s are ignored. If the pulse duration is higher or equal to the period, the DO output is set permanently. <br> Value range: 1200 ... 8388607 ( $25 \mu \mathrm{~s}$... approx. 175 ms ) | $1 \mathrm{~F}^{\text {a }} \mathrm{h}$ |
| 0x3101/x | PWM 1: <br> Period byte 2 |  |  |
| 0x3102/x | PWM 1: <br> Period byte 1 |  |  |
| 0x3103/x | PWM 1: <br> Period byte 0 |  |  |
| 0x3104/x | PWM 2: Period byte 3 |  | 1 F 40 h |
| 0x3105/x | PWM 2: <br> Period byte 2 |  |  |
| 0x3106/x | PWM 2: <br> Period byte 1 |  |  |
| 0x3107/x | PWM 2: <br> Period byte 0 |  |  |
| $\begin{aligned} & 0 \times 3108 / x \\ & \ldots \\ & 0 \times 3129 / x \end{aligned}$ | Reserved |  |  |
| 0x5620/x | PWM pulse duration | Selection of pulse duration <br> Permissible values. 48 ... 8388607 (corresponds to 1 ... $175000 \mu \mathrm{~s}$ |  |

## 1 Note!

The pulse/pause ratio is determined by specifying the period ( $0 \times 3100$... $0 \times 3107$ ) and pulse duration ( $0 \times 5620$ ).

### 8.13.2

RS232 interface - EPM-S640

## Parameter data

| Parameter data - ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x3100/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. |  |
| 0x3101/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. |  |
| 0x3102/x | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 0x3103/x | Baud rate | $00_{h}$ : 9600 Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud 06h: 2400 Baud $07_{h}$ : 4800 Baud 08h: 7200 Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud OCh: 38400 Baud 0D $\mathrm{h}: 57600$ Baud 0E $\mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{\text {h }}$ |
| 0x3104/x | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{h}: \text { ASCII }$ | $01_{\text {h }}$ |
| 0x3105/x | Data format | Bit $1 / 0$ number of data bits $00_{b}: 5$ <br> $01_{b}: 6$ <br> $10_{b}: 7$ <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{\mathrm{b}}$ : none <br> $01_{b}$ : odd <br> $11_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None <br> $01_{b}$ : Hardware <br> $10_{b}$ : XON/XOFF | $13_{h}$ |
| 0x3106/x | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is | $00_{\text {h }}$ |
| 0x3107/x | ZNA (LOW byte) | executed. $0 \ldots 65535[\mathrm{~ms}]\left(0000_{\mathrm{h}} . . . \text { FFFF }_{\mathrm{h}}\right)$ | $00_{\text {h }}$ |


| Parameter data-ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x3108/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).$0 \ldots 65535 \text { [ms] }\left(0000_{h} . \therefore \text { FFFF }_{h}\right)$ | $00_{\text {h }}$ |
| 0x3109/x | Character delay time (LOW byte) |  | 0Ah |
| 0x310A/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. $1 \ldots 255\left(01_{h} \ldots F_{h}\right)$ | $01_{\text {h }}$ |
| $\begin{aligned} & 0 \times 310 B / x \\ & \ldots \\ & 0 \times 3110 / x \end{aligned}$ | Reserved |  |  |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ <br> subindex | Name | Description/value | Lenze |
| 0x3100/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. |  |
| 0x3101/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. |  |
| 0x3102/x | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 0x3103/x | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud 06h: 2400 Baud $07_{h}$ : 4800 Baud 08h: 7200 Baud $09 \mathrm{~h}: 9600$ Baud $0 A_{h}$ : 14400 Baud 0Bh: 19200 Baud OCh: 38400 Baud 0Dh: 57600 Baud $0 \mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{h}: 109700$ Baud | $00_{\text {h }}$ |
| 0x3104/x | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{\mathrm{h}}: \text { STX/ETX }$ | 02h |
| 0x3105/x | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{b}: 6$ <br> $10_{b}: 7$ <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control <br> $00_{b}$ : None <br> $01_{b}$ : Hardware <br> $10_{\mathrm{b}}$ : XON/XOFF | $13_{h}$ |
| 0x3106/x | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is | $00_{\text {h }}$ |
| 0x3107/x | ZNA (LOW byte) | executed. $0 . . .65535[\mathrm{~ms}]\left(0000_{\mathrm{h}} . \therefore \text { FFFF }_{\mathrm{h}}\right)$ | $00_{\text {h }}$ |
| 0x3108/x | TMO (HIGH byte) | TMO serves to define the maximally permissible | OAh |
| 0x3109/x | TMO (LOW byte) | $0 \ldots 65535[\mathrm{~ms}]\left(0000_{\mathrm{h}} . . . \text { FFFF }_{\mathrm{h}}\right)$ |  |
| 0x310A/x | No. of start identifiers | $00_{h}$ : 1 start identifier (2. start identifier is ignored) $01_{h}$ : 2 start identifiers | $01_{\text {h }}$ |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x310B/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. $1 \ldots 255\left(01_{h} \ldots F_{h}\right)$ | 02h |
| 0x310C/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. | $00_{\text {h }}$ |
| 0x310D/x | No. of end identifier | $00_{h}$ : 1 end identifier (2. end identifier ( $0 \times 310 \mathrm{D} / \mathrm{x}$ ) is ignored) <br> $01_{h}$ : 2 end identifiers | $01_{\text {h }}$ |
| 0x310E/x | End identifier 1 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 \ldots 255\left(01_{h} \ldots F_{h}\right)$ | $03_{h}$ |
| 0x310F/x | End identifier 2 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 \ldots 255\left(01_{h} \ldots F_{h}\right)$ | $00_{\text {h }}$ |
| 0x3110/x | Reserved |  |  |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x3100/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. |  |
| 0x3101/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. |  |
| 0x3102/x | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled) <br> Bit 7: Reserved Other values are not permissible! | $00_{\text {h }}$ |
| 0x3103/x | Baud rate | $00_{h}$ : 9600 Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud $05_{h}$ : 1800 Baud 06h: 2400 Baud $07_{h}$ : 4800 Baud 08h: 7200 Baud $09_{h}$ : 9600 Baud $0 A_{h}: 14400$ Baud $0 B_{h}$ : 19200 Baud 0Ch: 38400 Baud 0Dh: 57600 Baud 0E ${ }_{h}$ : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{h}: 109700$ Baud | $00_{\text {h }}$ |
| 0x3104/x | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{h}: 3964 \\ & 03_{h}: 3964 R \end{aligned}$ | $03_{\text {h }}$ |
| 0x3105/x | Data format | Bit $1 / 0$ number of data bits $00_{b}: 5$ <br> $01_{b}: 6$ <br> $10_{b}$ : 7 <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{b}: 1 \\ & 10_{b}: 1.5 \end{aligned}$ $11_{b}: 2$ <br> Bit 7/6 flow control $00_{\mathrm{b}}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | $13_{h}$ |
| 0x3106/x | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \ldots 65535 \text { [20-ms steps] }\left(00_{h} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $00_{\text {h }}$ |
| 0x3107/x | Character delay time | 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). <br> 0 ... 65535 [20-ms steps] $\left(00_{h} . . . \mathrm{FF}_{\mathrm{h}}\right)$ | OAh |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x3108/x | Acknowledgement time | The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $0 A_{h}$ |
| 0x3109/x | Block wait time | Maximum time between the recognition of a requested frame (DLE) and the response. $0 \ldots 255[\mathrm{~ms}]\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $0 A_{h}$ |
| 0x310A/x | STX repetitions | Maximum number of times the EPM-S640 tries to establish a connection. <br> 0 ... 255 [ms] ( $00_{h} . . . \mathrm{FF}_{\mathrm{h}}$ ) | $05_{\text {h }}$ |
| 0x310B/x | DBL | If the block wait time has been exceeded, the number of STX repetitions for the requested protocol can be specified here. $0 \ldots 255[\mathrm{~ms}]\left(00_{h} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | 06h |
| 0x310C/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 the protocols 3964(R), both partners must have different priorities. <br> $00_{\mathrm{h}}$ : LOW <br> $01_{h}$ : HIGH | $00_{\text {h }}$ |
| $\begin{aligned} & 0 \times 310 D / x \\ & \ldots \\ & 0 \times 3110 / x \end{aligned}$ | Reserved |  |  |

### 8.13.3 RS485 interface - EPM-S650

## Parameter data

| Parameter data - ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x3100/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. |  |
| 0x3101/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. |  |
| 0x3102/x | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 0x3103/x | Baud rate | $00_{h}$ : 9600 Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud 06h: 2400 Baud $07_{h}$ : 4800 Baud 08h: 7200 Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud OCh: 38400 Baud 0D $\mathrm{h}: 57600$ Baud 0E $\mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{\text {h }}$ |
| 0x3104/x | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{h}: \text { ASCII }$ | $01_{\text {h }}$ |
| 0x3105/x | Data format | Bit $1 / 0$ number of data bits $00_{b}: 5$ <br> $01_{b}: 6$ <br> $10_{b}: 7$ <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{\mathrm{b}}$ : none <br> $01_{b}$ : odd <br> $11_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None <br> $01_{b}$ : Hardware <br> $10_{b}$ : XON/XOFF | $13_{h}$ |
| 0x3106/x | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is | $00_{\text {h }}$ |
| 0x3107/x | ZNA (LOW byte) | executed. $0 \ldots 65535[\mathrm{~ms}]\left(0000_{\mathrm{h}} . . . \text { FFFF }_{\mathrm{h}}\right)$ | $00_{\text {h }}$ |



| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x3100/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. |  |
| 0x3101/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. |  |
| 0x3102/x | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| 0x3103/x | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04 h : 1200 Baud $05_{h}$ : 1800 Baud $06_{h}$ : 2400 Baud $07_{\mathrm{h}}$ : 4800 Baud $08_{\mathrm{h}}$ : 7200 Baud 09 h : 9600 Baud $0 A_{h}$ : 14400 Baud OBh: 19200 Baud $0 C_{h}: 38400$ Baud $0 \mathrm{D}_{\mathrm{h}}: 57600$ Baud 0Eh: 76800 Baud $0 F_{h}$ : 115200 Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{h}$ |
| 0x3104/x | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{h}: \operatorname{STX} / E T X$ | $02_{\text {h }}$ |
| 0x3105/x | Data format | Bit 1/0 number of data bits $00_{b}$ : 5 <br> $01_{b}$ : 6 <br> 10 : 7 <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{\mathrm{b}}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{\mathrm{b}}$ : XON/XOFF | $13_{\text {h }}$ |
| 0x3106/x | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is | $00_{h}$ |
| 0x3107/x | ZNA (LOW byte) | executed. $0 \ldots 65535 \text { [ms] }\left(0000_{h} . \therefore \text { FFFF }_{h}\right)$ | $00_{\text {h }}$ |
| 0x3108/x | TMO (HIGH byte) | TMO serves to define the maximally permissible | OAh |
| 0x3109/x | TMO (LOW byte) |  |  |
| 0x310A/x | No. of start identifiers | $00_{\mathrm{h}}: 1$ start identifier (2. start identifier is ignored) $01_{h}: 2$ start identifiers | $01_{\text {h }}$ |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x310B/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. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | $02_{\text {h }}$ |
| 0x310C/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. | $00_{\text {h }}$ |
| 0x310D/x | No. of end identifier | $00_{h}$ : 1 end identifier ( 2 . end identifier ( $0 \times 310 \mathrm{D} / \mathrm{x}$ ) is ignored) <br> $01_{h}$ : 2 end identifiers | $01_{\text {h }}$ |
| 0x310E/x | End identifier 1 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} \ldots F_{h}\right)$ | $03_{\text {h }}$ |
| 0x310F/x | End identifier 2 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 \ldots 255\left(01_{h} \ldots F F_{h}\right)$ | $00_{\text {h }}$ |
| 0x3110/x | Reserved |  |  |
| 0x3111/x | Operating mode | Operating mode of the interface $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $00_{h}$ |
| 0x3112/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 ( $\square$ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal R(A) 5 V (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. <br> 02 h : Signal $R(A) 0 \mathrm{~V}$; signal $R(B) 5 \mathrm{~V}$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $13_{h}$ |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x3100/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. |  |
| 0x3101/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. |  |
| 0x3102/x | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled) <br> Bit 7: Reserved Other values are not permissible! | $00_{\text {h }}$ |
| 0x3103/x | Baud rate | $00_{h}$ : 9600 Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud $05_{h}$ : 1800 Baud 06h: 2400 Baud $07_{h}$ : 4800 Baud 08h: 7200 Baud $09_{h}$ : 9600 Baud $0 A_{h}: 14400$ Baud $0 B_{h}$ : 19200 Baud 0Ch: 38400 Baud 0Dh: 57600 Baud 0E ${ }_{h}$ : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{h}: 109700$ Baud | $00_{\text {h }}$ |
| 0x3104/x | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{h}: 3964 \\ & 03_{h}: 3964 R \end{aligned}$ | $03_{\text {h }}$ |
| 0x3105/x | Data format | Bit $1 / 0$ number of data bits $00_{b}: 5$ <br> $01_{b}: 6$ <br> $10_{b}$ : 7 <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{b}: 1 \\ & 10_{b}: 1.5 \end{aligned}$ $11_{b}: 2$ <br> Bit 7/6 flow control $00_{\mathrm{b}}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | $13_{h}$ |
| 0x3106/x | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \ldots 65535 \text { [20-ms steps] }\left(00_{h} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $00_{\text {h }}$ |
| 0x3107/x | Character delay time | 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). <br> 0 ... 65535 [20-ms steps] $\left(00_{h} . . . \mathrm{FF}_{\mathrm{h}}\right)$ | OAh |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Index/ subindex | Name | Description/value | Lenze |
| 0x3108/x | Acknowledgement time | The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} \ldots F_{h}\right)$ | $0 A_{h}$ |
| 0x3109/x | Block wait time | Maximum time between the recognition of a requested frame (DLE) and the response. $0 \ldots 255 \text { [ms] }\left(00_{h} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $0 A_{h}$ |
| 0x310A/x | STX repetitions | Maximum number of times the EPM-S640 tries to establish a connection. $0 \ldots 255[\mathrm{~ms}]\left(00_{\mathrm{h}} \ldots . . \mathrm{FF}_{\mathrm{h}}\right)$ | 05h |
| 0x310B/x | DBL | If the block wait time has been exceeded, the number of STX repetitions for the requested protocol can be specified here. $0 \ldots 255 \text { [ms] }\left(00_{h} \ldots . . F_{h}\right)$ | $06_{\text {h }}$ |
| 0x310C/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 the protocols 3964(R), both partners must have different priorities. <br> $00_{\mathrm{h}}$ : LOW <br> $01_{h}$ : HIGH | $00_{h}$ |
| $\begin{aligned} & 0 \times 310 D / x \\ & \ldots \\ & 0 \times 3110 / x \end{aligned}$ | Reserved |  |  |
| 0x3111/x | Operating mode | Operating mode of the interface <br> $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $00_{h}$ |
| 0x3112/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 ( 1 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal $R(A) 5 V$ (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal $R(A) 0$ V; signal $R(B) 5 V$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $13_{\text {h }}$ |

### 8.14 "Store" function

All parameter changes and the structure of the station are permanently stored via index I1010 hex (Store parameter).

## 1 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 (both red LEDs are blinking).
The stored structure of the station is deleted via index $11011_{\text {hex }}$ (Restore parameter).

| Index | Name | Possible settings |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |
| $\mathrm{I} 1010_{\mathrm{h}}$ | 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). | ¢1340 |
| 0 |  |  | Number of subindexes assigned | Read only <br> Number of the subindexes used by the index $1010^{\text {h }}$ |  |
| 1 | Save |  | $0=$ no function <br> $1702257011_{d}=$ save parameters | The numerical value is ASCII-coded and complies with: $65766173_{h}=" E " \text { "V" "A" "S" }$ |  |

### 8.15 Loading the default setting

All parameter changes are reset to the default setting via index $11011_{\text {hex }}$.
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 |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |
| $\mathrm{I} 1011_{\mathrm{h}}$ | 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). | c1340 |
| 0 |  |  | Number of subindexes assigned | Read only <br> Number of the subindexes used by the index 1011 $_{h}$ |  |
| 1 | Load |  | $\begin{aligned} & 0=\text { no function } \\ & 1684107116_{d}=\text { load Lenze setting } \end{aligned}$ | The numerical value is ASCII-coded and complies with: 6461 6F 6Ch = "D" "A" "O" "L" After executing "Load", acceptance is effected by <br> - Voltage off/on, or <br> - Transmission of a reset node telegram (00 82 xx , mit $\mathrm{xx}=$ node address) |  |

## $8.16 \quad$ Node Guarding



Fig. 8-13 Node Guarding Protocol

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

## Description

The Node Guarding Protocol monitors the connection between master and slave.
Via index $1100 C_{h}$, Guard time, a time [ms] can be set and in index $1100 D_{h}$, 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 11029 h. The outputs switch to defined states ( $\mathbb{\square} 356$ ).
- A change to the Operational status triggers a reset.


## Status telegram

|  | 1 byte of user data |  |
| :---: | :---: | :---: |
| Identifier | Device status (bits 0 ... 6) | Toggle bit |
| 1792d $\left(700_{h}\right)$ |  |  |
| Identifier: |  |  |
|  | Formula | Information |
| Identifier | $\begin{aligned} & =\text { Basic identifier }+ \text { node address } \\ & =1792_{d}+x x \end{aligned}$ | The basic identifier for Node Guarding is firmly adjusted to $1792_{d}\left(700_{h}\right)$ $\mathrm{xx}=$ Node address of the I/O system |

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

| Command <br> (hex) | Device status |
| :---: | :--- |
| 04 | Stopped |
| 05 | Operational |
| 7 F | Pre-Operational |

Indices for setting

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| ${ }_{*} 100 C_{h}$ | Guard time | 0 | 0 | \{1 ms \} | 65535 | Node Guarding Monitoring time $0=$ monitoring not active | ¢ 342 |
| ${ }_{*} 100 \mathrm{D}_{\mathrm{h}}$ | Life time factor | 0 | 0 | \{1\} | 255 | Node Guarding <br> Response time computation factor <br> $0=$ monitoring not active <br> The response time is computed as: <br> Monitoring period x factor | ¢ 342 |
| $1100 \mathrm{E}_{\mathrm{h}}$ | Node Guarding identifier |  |  |  |  | Display only <br> Identifier = Basic identifier + node <br> address <br> (basic identifier cannot be modified) | [1342 |

## 1 Note!

The Lenze PLC's 9300 servo PLC and Drive PLC in connection with the function library LenzeCanDSxDrv.lib support the "Node Guarding" function.

### 8.17

Heartbeat


## 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 $11017_{h}$.

- Producer heartbeat is automatically started if a time $>0$ is entered in the index 1017 h 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 change into the Operational status triggers a reset.


## Status telegram

|  | 1 byte of user data |  |
| :---: | :---: | :---: |
| Identifier | Device status (bits 0 ... 6) | Bit 7 |
| 1792d ${ }_{\text {d }}\left(700_{h}\right.$ ) |  | Reserved |
| Identifier: |  |  |
|  | Formula | Information |
| Identifier | $\begin{aligned} & =\text { Basic identifier }+ \text { node address } \\ & =1792_{d}+x x \end{aligned}$ | The basic identifiers for heartbeat is firmly adjusted to $1792_{d}\left(700_{h}\right)$ <br> xx = Node address of the I/O system |

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

| Command <br> (hex) | Status |
| :---: | :--- |
| 00 | Boot-up |
| 05 | Operational |
| 04 | Stopped |
| 7 F | Pre-Operational |

## 1 Note!

The Lenze 9300 servo PLC and Drive PLC in connection with the function library LenzeCanDSxDrv.lib support the "heartbeat" function.

## 8 CANopen communication

## Monitoring

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

### 8.18 Monitoring

8.18.1 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 $12400_{h}$.

If no PDO is received within the time set in $12400_{h}$, the outputs will switch to their defined error status (see following sections).

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze |  |  |  |  |  |
| $12400_{h}$ | Timer value |  | 0 | \{1 ms \} | 65535 | Monitoring time for process data input objects | [1346 |
| 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 |  |  |  |  |  |

### 8.18.2 Digital output monitoring

Via the index I6206h you can configure the reactions of the digital outputs which are to take place when no telegrams, Node Guarding Events or Heartbeat, have been received in the adjusted monitoring time.

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Sele |  |  |  |  |
| $16206_{h}$ | Error mode digital output |  | 0 | \{1\} | 255 | Configures digital output monitoring | ■ 347 |
|  |  |  |  | All digital outputs retain the last status output. |  |  |  |
|  |  |  | 255 | Response from 16207h |  | In 16207h, 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 $16207_{h}$ the response can be configured individually for each digital output.

| Index | Name | Possible settings |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |
| $16207_{h}$ | Error value digital output | 0 | 0 价 255 | Configures the individual digital output responses | - 347 |
|  |  |  | 8 bits of information |  |  |
|  |  |  | Bit Output switches to LOW value 0 |  |  |
|  |  |  | Bit Output retains last status output value 1 |  |  |
| 1 | Byte 1 | 0 |  |  |  |
| 2 | Byte 2 | 0 |  |  |  |
| ... | ... | ... |  |  |  |
| 64 | Byte 80 | 0 |  |  |  |

### 8.18.3 Monitoring of the analog outputs

Via the index 16443 hou can configure the reactions of the analog outputs which are to take place when no telegrams, Node Guarding Events or Heartbeat, have been received in the adjusted monitoring time.

- Monitoring is started on receipt of the next PDO telegram after the settings.
- If a telegram is not transmitted within the adjusted time, the module switches to the "Pre-Operational" state. No further process data are transmitted.
- A change into the "Operational" state triggers a reset.


Individual response setting
Via index $16444_{h}$ the response can be configured individually for each analog output.

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| $16444_{h}$ | Error value analog output |  | -32768 | \{1\} | 32767 | Configures the individual analog output responses The analog outputs provide the set value | [1] 348 |
| 1 | Channel 1 | 0 |  |  |  |  |  |
| 2 | Channel 2 | 0 |  |  |  |  |  |
| ... | ... | ... |  |  |  |  |  |
| 36 | Channel 36 | 0 |  |  |  |  |  |

### 8.19 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 indexes can be used for purposes of diagnostics. They show operating states. Settings cannot be made.

| Index | Information displayed | Description |
| :---: | :---: | :---: |
| 11014 h | Emergency telegram | [1349 |
| 11027 h | Reading out the module IDs | [1355 |
| $16000_{\text {h }}$ | Status of digital inputs | -1. 356 |
| $16200_{\text {h }}$ | Status of digital outputs | ¢ 356 |
| $16401_{\text {h }}$ | Status of analog inputs | [1356 |
| $16411_{\text {h }}$ | Status of analog outputs | [1356 |
| $15400_{h} \ldots 15403_{h}$ | Counter status | ¢ 357 |
| 11003 h | Current errors |  |

### 8.19.1 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.
$\left.\begin{array}{|l|l|l|l|l|l|}\hline \text { Index } & \text { Name } & \begin{array}{l}\text { Possible settings } \\ \text { Lenze }\end{array} & \text { Selection }\end{array}\right)$

## 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 $1001{ }_{\text {h }}$ | Error information |  |  |  |  |
| Error code | Error code | The error code $81_{\text {h }}$ (= device error) is displayed in index I1001h (error register). | 1 | 2 | 3 | 4 | 5 |

## 8 CANopen communication

## Diagnostics

Emergency telegram

## Error codes

| Error code | Meaning | Error information |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 1 | 2 | 3 | 4 | 5 |
| 0x0000 | Reset Emergency | 0x00 | 0x00 | $0 \times 00$ | $0 \times 00$ | $0 \times 00$ |
| 0x8100 | Heartbeat consumer | Node ID | LOW byte timer value | HIGH byte timer value | $0 \times 00$ | $0 \times 00$ |
| $0 \times 8130$ | Node guarding error | LOW byte guard time | HIGH byte guard time | LifeTime | $0 \times 00$ | $0 \times 00$ |
| 0x8157 | Module removed from slot [ n ]; no communication | $0 \times 05$ | 0x[n] | $0 \times 00$ | $0 \times 00$ | 0x00 |
| $0 \times 8210$ | PDO not executed due to length error | PDO number | Wrong length | PDO length | $0 \times 00$ | $0 \times 00$ |
| 0x8220 | PDO length exceeded | PDO number | 0x1000 | PDO number | $0 \times 00$ | $0 \times 00$ |
| 0x1000 | Module configuration was changed | $0 \times 01$ | 0x00 | $0 \times 00$ | $0 \times 00$ | 0x00 |
| 0x1000 | Error during initialisation of the backplane modules | $0 \times 02$ | Module number | LOW byte Error register | HIGH byte Error register | 0x00 |
| 0x1000 | Diagnostic alarm | $0 \times 40+$ module number | Diagnostic byte 1 | Diagnostic byte 2 | Diagnostic byte 3 | Diagnostic byte 4 |
| 0x0000 | Process alarm | 0×80 + module number | Diagnostic byte 1 | Diagnostic byte 2 | Diagnostic byte 3 | Diagnostic byte 4 |
| 0x1000 | Backplane bus: Initialisation error | EO | Module number | LOW byte Error bitfield | HIGH byte Error bitfield | $0 \times 00$ |
| 0x1000 | Backplane bus: Initialisation error pre-operational $\rightarrow$ operational | EO | 0x00 | $0 \times 00$ | $0 \times 00$ | $0 \times 00$ |
| 0x1000 | Backplane bus: Bus error | E1 | $0 \times 00$ | $0 \times 00$ | $0 \times 00$ | 0x00 |
| 0x1001 | Lenze PDO control, monitoring time exceeded | FF | $0 \times 10$ | PDO number | Monitoring time that has been set in [ms] |  |
| 0x2000 | Description of the process data width not permissible for modules with a time stamp functionality or a serial interface (index $3100 / \mathrm{x}, 3101 / \mathrm{x}$ ) | E2 | 0x00 | $0 \times 00$ | $0 \times 00$ | 0x00 |

## Process alarm

| EPM-S404 - process alarm |  |
| :---: | :--- |
| Diagnostic byte | Bit $7 \ldots 0$ |
| 1 | Bit 0: Limit value exceeded channel 1 <br> Bit 1: Limit value exceeded channel 2 <br> Bit 7 $\ldots$ 2: 0 (fixed) |
| 2 | Bit $0:$ Limit value not reached, channel 1 <br> Bit 1: Limit value not reached, channel 2 <br> Bit 7 ... 2:0 (fixed) |
| $3 / 4$ | Ticker value at the time of the alarm |


| EPM-S405 - process alarm |  |
| :---: | :---: |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0: Limit exceedance channel 1 <br> Bit 1: Limit exceedance channel 2 <br> Bit 2: Limit exceedance channel 3 <br> Bit 3: Limit exceedance channel 4 <br> Bit 7 ... 4: 0 (fix) |
| 2 | Bit 0: Limit value underflow channel 1 <br> Bit 1: Limit value underflow channel 2 <br> Bit 2: Limit value underflow channel 3 <br> 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: 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 | $\mu s$-ticker value at the time of the alarm <br> The I/O compound module features an integrated 32-bit timer ( $\mu$ s-ticker) which is started at switch-on and starts at 0 again after $23^{2}-1 \mu \mathrm{~s}$. These two bytes represent the lower 2 bytes of the $\mu \mathrm{s}$-ticker ( $0 . . .2^{16}$ - 1 ) |
| EPM-S600 - process alarm |  |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7... 5: 0 (fixed) |
| 2 | State of the inputs at the time of the alarm <br> Bit 0: A/pulse <br> Bit 1: $\mathrm{B} /$ direction <br> Bit 2: Latch <br> Bit 3: Hardware gate <br> Bit 4: Reset <br> Bit 7... 5: 0 (fixed) |
| 3/4 | Ticker value at the time of the alarm |

EPM-S601 and EPM-S602 - process alarm

| Diagnostic byte | Bit 7 ... 0 |
| :---: | :---: |
| 8 | Bit 0: 0 <br> Bit 1: 0 <br> Bit 2: Counter 0, overflow/underflow/final value <br> Bit 3: Counter 0, comparison value reached <br> Bit 4: 0 <br> Bit 5: 0 <br> Bit 6: Counter 1, overflow/underflow/final value <br> Bit 7: Counter 1, comparison value reached |
| 9 | State of the inputs at the time of the alarm <br> Bit 0: Counter 0, A/pulse <br> Bit 1: Counter 0, B/direction <br> Bit 2: Counter 1, A/pulse <br> Bit 3: Counter 1, B/direction <br> Bit 7 ... 4: 0 (fixed) |
| $10 . . .11$ | 16 bit $\mu s$ value at the time of the alarm |

## Diagnostic alarm

| EPM-S600-diagnostic alarm |  |
| :---: | :---: |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 2 | Bit 3 ... 0: Module class, $\mathbf{1 0 0 0}_{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| EPM-S601-diagnostic alarm |  |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 2 | Bit 3 ... 0: Module class, $1000_{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |


| EPM-S603 - diagnostic alarm |  |
| :---: | :---: |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | 0 (fixed) |
| 2 | Bit 3 ... 0: Module class, 1000 : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | 0 (fixed) |
| EPM-S604-diagnostic alarm |  |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0 : Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of missing external supply voltage <br> Bit 6 ... 5: 0 (fixed) <br> Bit 7: Parameterisation error |
| 2 | Bit 3 ... 0: Module class, 1000 : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 5 ... 0:0 (fixed) <br> Bit 6: Process alarm lost <br> Bit 7: 0 (fixed) |
| EPM-S640 - diagnostic alarm |  |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0 : set in the case of module fault <br> Bit 1: set in the case of internal error <br> Bit 2: set in the case of external error (cable break only for RS422) <br> Bit 3: 0 (fixed) <br> Bit 4: set in the case of missing external supply voltage <br> Bit 5, 6: 0 (fixed) <br> Bit 7: set in the case of parameterisation error |
| 2 | Bits 3 ... 0 : module class, $1100_{b}$ : communication module Bit 4: set if channel information available <br> Bits 7... 5: 0 (fixed) |
| 3 | Bits 3... 0: 0 (fixed) <br> Bit 4: set in the case of internal communication error <br> Bits 7 ... 5: 0 (fixed) |
| 4 | Bits 6 ... 0: channel type, $60_{\mathrm{h}}$ : communication processor Bit 7: 0 (fixed) |


| Diagnostic byte | Bit 7 ... 0 |
| :---: | :---: |
| 1 | Bit 0 : set in the case of module fault <br> Bit 1: set in the case of internal error <br> Bit 2: set in the case of external error (cable break only for RS422) <br> Bit 3: 0 (fixed) <br> Bit 4: set in the case of missing external supply voltage <br> Bit 5, 6: 0 (fixed) <br> Bit 7: set in the case of parameterisation error |
| 2 | Bits 3 ... 0: module class, $1100_{\mathrm{b}}$ : communication module Bit 4: set if channel information available <br> Bits 7... 5: 0 (fixed) |
| 3 | Bits 3... 0:0 (fixed) <br> Bit 4: set in the case of internal communication error <br> Bits 7 ... 5: 0 (fixed) |
| 4 | Bits 6 ... 0: channel type, $60_{h}$ : communication processor Bit 7: 0 (fixed) |

### 8.19.2 Reading out the module identifiers

The number of $I / O$ compound modules connected and the module types used can be read out via index $11027_{h}$. Each module type can be identified unambiguously via a hex value.

| Index | Subindex | Read out... | Module type | Module identifier |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 11027 h | 0 | ... the number of plugged modules (0 ... 64) | - | $23_{\text {dec }}$ | $017{ }_{\text {h }}$ |
|  | $1 . . .64$ | ... the module type in slots $1 . . .64$ | EPM-S200 | $1_{\text {dec }}$ | 001 h |
|  |  |  | EPM-S201 | 3 dec | 003 h |
|  |  |  | EPM-S202 | $5{ }_{\text {dec }}$ | 005h |
|  |  |  | EPM-S203 | 8 dec | 008 h |
|  |  |  | EPM-S204 | 2 dec | 002h |
|  |  |  | EPM-S205 | $4_{\text {dec }}$ | 004h |
|  |  |  | EPM-S206 | $7{ }_{\text {dec }}$ | 007h |
|  |  |  | EPM-S207 | $3841_{\text {dec }}$ | $\mathrm{FO1}_{\mathrm{h}}$ |
|  |  |  | EPM-S300 | 257 dec | $101_{\text {h }}$ |
|  |  |  | EPM-S301 | $260{ }_{\text {dec }}$ | 104h |
|  |  |  | EPM-S302 | $262_{\text {dec }}$ | $106_{\text {h }}$ |
|  |  |  | EPM-S303 | 259 dec | 103 h |
|  |  |  | EPM-S304 | $261{ }_{\text {dec }}$ | 105 h |
|  |  |  | EPM-S305 | 263 dec | 107 h |
|  |  |  | EPM-S306 | 258 dec | 102h |
|  |  |  | EPM-S308 | $265_{\text {dec }}$ | 109h |
|  |  |  | EPM-S309 | $264{ }_{\text {dec }}$ | 108 h |
|  |  |  | EPM-S310 | 3905dec | F41 ${ }_{\text {h }}$ |
|  |  |  | EPM-S400 | $1025_{\text {dec }}$ | $401{ }_{\text {h }}$ |
|  |  |  | EPM-S401 | $10288_{\text {dec }}$ | 404h |
|  |  |  | EPM-S402 | $1026_{\text {dec }}$ | 402 h |
|  |  |  | EPM-S403 | 1029 dec | $405_{\text {h }}$ |
|  |  |  | EPM-S404 | 1030 dec | 406 h |
|  |  |  | EPM-S405 | 1027 dec | 403 h |
|  |  |  | EPM-S406 | $1036{ }_{\text {dec }}$ | $40 C_{h}$ |
|  |  |  | EPM-S408 | $1035_{\text {dec }}$ | $40 \mathrm{~B}_{\mathrm{h}}$ |
|  |  |  | EPM-S500 | $1281_{\text {dec }}$ | 501 h |
|  |  |  | EPM-S501 | $12833_{\text {dec }}$ | 503 h |
|  |  |  | EPM-S502 | $1282_{\text {dec }}$ | 502h |
|  |  |  | EPM-S503 | $1284_{\text {dec }}$ | 504h |
|  |  |  | EPM-S600 | $2241_{\text {dec }}$ | $8 \mathrm{C} 1_{\text {h }}$ |
|  |  |  | EPM-S601 | $2243{ }_{\text {dec }}$ | $8 \mathrm{C} 3_{\mathrm{h}}$ |
|  |  |  | EPM-S602 | $2242_{\text {dec }}$ | $8 \mathrm{C} 2_{\text {h }}$ |
|  |  |  | EPM-S603 | $2244_{\text {dec }}$ | $8 \mathrm{C4}$ h |
|  |  |  | EPM-S604 | 2497 dec | $9 \mathrm{C} 1_{\text {h }}$ |
|  |  |  | EPM-S620 | $2305_{\text {dec }}$ | 901 h |
|  |  |  | EPM-S640 | 3585dec | $E 01{ }_{\text {h }}$ |
|  |  |  | EPM-S650 | 2625 dec | $\mathrm{A} 41_{\text {h }}$ |

### 8.19.3 Status of the digital inputs

Via the index $16000_{h}$ the status of the digital inputs can be displayed.

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze |  |  |  |  |  |
| $16000_{h}$ | Digital input |  | 0 | \{1\} | 255 | Read only Digital input status | c. 356 |
| 1 | Byte 1 |  | 0 \{1\} 255 |  |  |  |  |
| 2 | Byte 2 |  |  |  |  |  |  |
| ... | ... |  |  |  |  |  |  |
| 80 | Byte 80 |  |  |  |  |  |  |

### 8.19.4 Status of the digital outputs

Via the index $16200_{\mathrm{h}}$ the status of the digital outputs can be displayed:

| Index | Name | Possible settings |  |  |  | Important |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |  |
| ${ }_{*}^{*} 200{ }_{\text {h }}$ | Digital output |  | 0 | \{1\} | 255 |  | Digital output status The outputs can be set | [1356 |
| 1 | Byte 1 |  |  |  |  |  | manually (forcing): |  |
| 2 | Byte 2 |  |  |  |  |  | $12360_{h}$ |  |
| ... | ... |  |  |  |  |  |  |  |
| 80 | Byte 80 |  |  |  |  |  |  |  |

### 8.19.5 Status of the analog inputs

Via the index $16401_{h}$ the status of the analog inputs can be displayed.

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| 16401 h | Analog input |  | -32768 | \{1\} | 32767 | Display only Analog input status | ¢ 356 |
| 1 | Channel 1 |  |  |  |  |  |  |
| 2 | Channel 2 |  |  |  |  |  |  |
| $\cdots$ | ... |  |  |  |  |  |  |
| 36 | Channel 36 |  |  |  |  |  |  |

### 8.19.6 Status of the analog outputs

Via the index $16411_{h}$ the status of the analog outputs can be displayed:

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| ${ }_{*} 6411_{\text {h }}$ | Analog output |  | -32768 | \{1\} | 32767 | - Analog output status <br> - The outputs can be set | ¢ 356 |
| 1 | Channel 1 |  |  |  |  | manually (forcing): |  |
| 2 | Channel 2 |  |  |  |  | - Depends on CAN status and 12360h |  |
| ... | ... |  |  |  |  |  |  |
| 36 | Channel 36 |  |  |  |  |  |  |

### 8.19.7 Status of the counters

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


### 8.19.8 Status of digital inputs with time stamp function

Index $15430_{\mathrm{h}}$ can be used to display the status of the digital inputs with time stamp function.

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| 15430 h | Dl time stamp state |  | 0000h | \{1\} | FFFFh | Read only <br> Status of time stamp <br> 15 entries per module <br> Bit 0: channel status DI1 (0: <br> FALSE; 1: TRUE) <br> Bit 1: channel status DI2 (0: <br> FALSE; 1: TRUE) <br> Bits 2 ... 7: reserved <br> Bits 8 ... 15: counter which counts from 0 ... 127 and then goes back to 0. <br> Bits 16 ... 32: ticker value | ¢. 260 |
| 1 | DWORD 1 |  |  |  |  |  |  |
| 2 | DWORD 2 |  |  |  |  |  |  |
| $\cdots$ 80 | DWORD 64 |  |  |  |  |  |  |
| 80 | DWORD 64 |  |  |  |  |  |  |

## Diagnostics

Status of digital outputs with time stamp function

### 8.19.9 Status of 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 |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| 15440 ${ }_{\text {h }}$ | DO time stamp state |  | 0000 h | \{1\} | FFFFh | Read only <br> Status of FIFO memory <br> 1 entry per module <br> Bits 0 ... 5: running number of time stamp entry last written to the FIFO memory. <br> Bits 6 ... 7: reserved <br> Bits 8 ... 13: running number of time stamp entry to be processed next. <br> Bits 14 ... 15: reserved <br> Bits 16 ... 23: <br> - 00h or 80 h : everything ok <br> - 01h or 81h: no following entry <br> - 02 h or 82 h : no new entries <br> - 03 h or 83 h : FIFO 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. <br> Bits 24 ... 31:number of time stamp entries in FIFO memory. | [1260 |
| 1 | DWORD 1 |  |  |  |  |  |  |
| 2 | DWORD 2 |  |  |  |  |  |  |
| .. <br> 80 | DWORD 64 |  |  |  |  |  |  |

You can specify values using the following index.

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| 15640 ${ }_{\text {h }}$ | DO time stamp control |  | 0000h | \{1\} | FFFFh | 15 entries per module <br> Bits 0 ... 5: reserved <br> Bit 6: channel status D01 (0: | [1260 |
| 1 | DWORD 1 |  |  |  |  |  |  |
| 2 | DWORD 2 |  |  |  |  | Bit 7: channel status D02 (0: <br> FALSE; 1: TRUE) |  |
| $\cdots$ | ... |  |  |  |  | Bits 8 ... 15: counter which counts from 0 ... 127 and then goes back |  |
| 80 | DWORD 64 |  |  |  |  | to 0. <br> Bits 16 ... 32: ticker value |  |

### 8.19.10

## Status of 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 |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| 15420h | PWM state |  | 0000h | \{1\} |  | Read only <br> Status of PWM <br> 2 entries per module <br> Bit 0: reserved <br> Bit 1: PWM status (0: PWM output stopped; 1: PWM output active) <br> Bit 2: output status (0: push/pull output; 1: highside output) <br> Bits 3 ... 15: reserved | ¢ 264 |
| 1 | WORD 1 |  |  |  |  |  |  |
| 2 $\ldots$ | WORD 2 |  |  |  |  |  |  |
| 80 | WORD 64 |  |  |  |  |  |  |
| 15620 ${ }_{\text {h }}$ | PWM pulse duration |  | 0000h | \{1\} | FFFFh | Status of PWM <br> 2 entries per module | ¢1264 |
| 1 | DWORD 1 |  |  |  |  | Specification of pulse duration in [ $\mu \mathrm{s}$ ]. |  |
| 2 | DWORD 2 |  |  |  |  |  |  |
| ... | ... |  |  |  |  |  |  |
| 80 | DWORD 64 |  |  |  |  |  |  |
| $\mathrm{I}^{5621}{ }_{\text {h }}$ | PMW control |  | 0000 h | \{1\} | FFFFh | Control word PWM 2 entries per module Bits 0 ... 1: reserved | [1356 |
| 1 2 | WORD 1 |  |  |  |  | Bit 2: <br> - 0: push/pull output The output signal is switched to HIGH level active and LOW |  |
| 2 | WORD 2 |  |  |  |  | - 1: highside output The output signal is only switched to HIGH level active. |  |
| ... | ... |  |  |  |  | Bits 3 ... 7: reserved Bit 8: 0-1 edge: PWM output starts |  |
| 80 | WORD 64 |  |  |  |  | Bit 9: 1-0 edge: PWM output stops Bits 10 ... 15: reserved |  |

### 8.20 Index table

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



| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |
| ${ }_{*}^{11006}$ h | Sync interval ( $\mu \mathrm{s}$ ) | 0 | $0 \quad\{1 \mu s\}$ | 4294967295 | - The I/O system acts as sync consumer: <br> - Set bit $30=0$ under $11005_{h}$ <br> - After the time set under I1006h, I/O system switches to the communication status set under I1029h <br> - A reset will be carried out with the next sync telegram <br> - With $11006_{h}=0$, the monitoring is deactivated |  |
| 11008 h | DIS: Device name |  |  |  | Display only Device name |  |
| 11009 ${ }_{\text {h }}$ | DIS: Hardware version |  |  |  | Display only Hardware version |  |
| $1100 A_{h}$ | DIS: Software version |  |  |  | Display only Software version |  |
| $1100 B_{h}$ | Node ID |  | 1 \{1\} | 127 | Read only CANopen node address |  |
| $I 100 C_{h}$ | Guard time | 0 | $0 \quad\{1 \mathrm{~ms}\}$ | 65535 | Node Guarding Monitoring time $0=$ monitoring not active | [1] 342 |
| $1100 D_{h}$ | Life time factor | 0 | 0 \{1\} | 255 | Node Guarding <br> Response time computation factor <br> $0=$ monitoring not active <br> The response time is computed as: <br> Monitoring period x factor | [1] 342 |
| $1100 \mathrm{E}_{\mathrm{h}}$ | Node Guarding identifier |  |  |  | Display only <br> Identifier = Basic identifier + node address <br> (basic identifier cannot be modified) | [1] 342 |
| $\mathrm{I} 1010_{\mathrm{h}}$ | 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). | [1] 340 |
| 0 |  |  | Number of subindexes assigned |  | Read only <br> Number of the subindexes used by the index $1010_{h}$ |  |
| 1 | Save |  | $\begin{aligned} & 0=\text { no function } \\ & 1702257011_{d}=\text { save parameters } \end{aligned}$ |  | The numerical value is ASCII-coded and complies with: $65766173_{h}=" E$ " "V" "A" "S" |  |



| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selectio |  |  |  |
| $\mathrm{I} 1400_{\mathrm{h}}$ |  |  |  |  |  | ¢ 256 |
| 1 | COB-ID used by RxPDO 1 | $\begin{gathered} 513+ \\ \text { NID } \end{gathered}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 \ldots .24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | Reserved |  |  |
|  |  |  | 255 | Process data update on every occurrence of an event | Every received value is accepted |  |
| $\mathrm{I} 1401_{\mathrm{h}}$ |  |  |  |  |  | 1-1 256 |
| 1 | COB-ID used by RxPDO 2 | $\begin{gathered} 768+ \\ \text { NID } \end{gathered}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 \ldots 24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | Reserved |  |  |
|  |  |  | 255 | Process data update on every occurrence of an event | Every received value is accepted |  |
| $11402_{h}$ |  |  |  |  |  | ¢1256 |
| 1 | COB-ID used by RxPDO 3 | $\begin{array}{r} 1024 \\ + \text { NID } \end{array}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 \ldots 24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | Reserved |  |  |
|  |  |  | 255 | Process data update on every occurrence of an event | Every received value is accepted |  |
| $\mathrm{I} 1403_{\mathrm{h}}$ |  |  |  |  |  | ¢ 256 |
| 1 | COB-ID used by RxPDO 4 | $\begin{array}{r} 1280 \\ + \text { NID } \end{array}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 \ldots 24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | Reserved |  |  |
|  |  |  | 255 | Process data update on every occurrence of an event | Every received value is accepted |  |


| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selectio |  |  |  |
| $11404_{h}$ |  |  |  |  |  | -1256 |
| 1 | COB-ID used by RxPDO 5 | $\begin{array}{r} 1920 \\ + \text { NID } \end{array}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 . . .24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | Reserved |  |  |
|  |  |  | 255 | Process data update on every occurrence of an event | Every received value is accepted |  |
| $\mathrm{I} 1405_{\mathrm{h}}$ |  |  |  |  |  | ¢ 256 |
| 1 | COB-ID used by RxPDO 6 | $\begin{gathered} 576+ \\ \text { NID } \end{gathered}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 \ldots 24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | Reserved |  |  |
|  |  |  | $255$ | Process data update on every occurrence of an event | Every received value is accepted |  |
| $\mathrm{I} 1406_{\mathrm{h}}$ |  |  |  |  |  | ¢1256 |
| 1 | COB-ID used by RxPDO 7 | $\begin{gathered} 832+ \\ \text { NID } \end{gathered}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 . . .24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | Reserved |  |  |
|  |  |  | $255$ | Process data update on every occurrence of an event | Every received value is accepted |  |
| $\mathrm{I} 1407_{\mathrm{h}}$ |  |  |  |  |  | ¢ 256 |
| 1 | COB-ID used by RxPDO 8 | $\begin{array}{r} 1088 \\ + \text { NID } \end{array}$ | 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 |  |
|  |  |  | $\begin{aligned} & 0 \ldots 24 \\ & 0 \end{aligned}$ | Process data update on every sync telegram transmission | The input data are accepted on sync telegram transmission. |  |
|  |  |  | $\begin{aligned} & 241 \ldots \\ & 254 \end{aligned}$ | Reserved |  |  |
|  |  |  | 255 | Process data update on every occurrence of an event | Every received value is accepted |  |



## 8 CANopen communication

## Index table




## 8 CANopen communication

## Index table




| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| $\mathrm{I} 1609_{\mathrm{h}}$ |  |  |  |  |  | Mapping parameters for receive PDOs |  |
| 0 | Number of mapped RxPDO10 |  | 0 | \{1\} | 255 | 8 bit value |  |
| 1 | 1st mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 2 | 2nd mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 3 | 3rd mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 4 | 4th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 5 | 5th mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 6 | 6th mapped object |  | 00000000 ${ }_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 7 | 7th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 8 | 8th mapped object |  | 00000000 ${ }_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| $\mathrm{I} 1800_{\mathrm{h}}$ |  |  |  |  |  |  | ㄴ. 256 |
| 1 | COB-ID used by TxPDO 1 | $\begin{gathered} 384+ \\ \text { NID } \end{gathered}$ | 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 Fun | deactiva |  | The output data are accepted on sync telegram transmission. |  |
|  |  |  | $1 \ldots$ Proc <br> 240 $\ldots$ <br>  Proc <br>  240 | ata trans <br> ata trans | ync no. 1 ync no. | The output data are accepted after transmission of the set number (1... 240) of sync telegrams. |  |
|  |  |  | 254 Tim | trolled p | transfer | Only if a cycle time is set in I180x ${ }_{h}$, subindex 5 |  |
|  |  |  | 255 Eve | trolled p | a transfer |  |  |
|  |  |  | 255 Eve | trolled c overlay | ta transfer | Only if a cycle time is set in I180xh, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms \} | 65535 | Blocking time |  |
| 5 | Event time | 100 | 0 | \{1 ms \} | 65535 | Cycle time |  |


| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Sele |  |  |  |
| $\mathrm{I} 801_{\mathrm{h}}$ |  |  |  |  |  | [1256 |
| 1 | COB-ID used by TxPDO 2 | $\begin{gathered} 640+ \\ \text { NID } \end{gathered}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... <br> Process data transfer after sync no. $240$ | The output data are accepted after transmission of the set number (1 ... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180x ${ }_{h}$, 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 I180x ${ }_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms\} 65535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms\} 65535 | Cycle time |  |
| I1802h |  |  |  |  |  | ¢ 256 |
| 1 | COB-ID used by TxPDO 3 | $\begin{gathered} 896+ \\ \text { NID } \end{gathered}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... <br> Process data transfer after sync no. $240$ | The output data are accepted after transmission of the set number (1 ... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180xh, 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 I180x ${ }_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms\} 65535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms \} 65535 | Cycle time |  |


| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Sele |  |  |  |
| $\begin{aligned} & \mathrm{I} 1803_{\mathrm{h}} \\ & \frac{1}{5} \end{aligned}$ |  |  |  |  |  | ¢ 256 |
|  | COB-ID used by TxPDO 4 | $\begin{array}{r} 1152 \\ + \text { NID } \end{array}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... <br> Process data transfer after sync no. $240$ | The output data are accepted after transmission of the set number (1 ... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180x ${ }_{h}$, 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 I180xh, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms \} 65535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms \} 65535 | Cycle time |  |
| $\mathrm{I} 1804_{h}$ |  |  |  |  |  | ■ 256 |
| 1 | COB-ID used by TxPDO 5 | $\begin{array}{r} 1664 \\ + \text { NID } \end{array}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... <br> Process data transfer after sync no. $240$ | The output data are accepted after transmission of the set number (1 ... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180xh, 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 I180x ${ }_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms 665535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms 665535 | Cycle time |  |


| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze |  |  |  |  |
| $\mathrm{I} 1805_{\mathrm{h}}$ |  |  |  |  |  | ¢ 256 |
| 1 | COB-ID used by TxPDO 6 | $\begin{gathered} 448+ \\ \text { NID } \end{gathered}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... <br> Process data transfer after sync no. $240$ | The output data are accepted after transmission of the set number (1... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180xh, 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 I180x ${ }_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms\} 65535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms\} 65535 | Cycle time |  |
| $\mathrm{I} 1806_{h}$ |  |  |  |  |  | [1256 |
| 1 | COB-ID used by TxPDO 7 | $\begin{aligned} & 704+ \\ & \text { NID } \end{aligned}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... <br> Process data transfer after sync no. $240$ | The output data are accepted after transmission of the set number (1 ... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180xh, 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 I180x ${ }_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms \} 65535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms \} 65535 | Cycle time |  |


| Index | Name | Possible settings |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Sele |  |  |  |
| $\frac{1}{1807_{\mathrm{h}}}$ |  |  |  |  |  | ■ 256 |
|  | COB-ID used by TxPDO 8 | $\begin{gathered} 960+ \\ \text { NID } \end{gathered}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... <br> Process data transfer after sync no. $240$ | The output data are accepted after transmission of the set number (1 ... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180x ${ }_{h}$, 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 I180x ${ }_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms\} 65535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms \} 65535 | Cycle time |  |
| $\mathrm{l} 1808_{\mathrm{h}}$ |  |  |  |  |  | $\square 256$ |
| 1 | COB-ID used by TxPDO 9 | $\begin{aligned} & 1216 \\ & + \text { NID } \end{aligned}$ | 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. |  |
|  |  |  | $\begin{aligned} & 1 \ldots \\ & 240 \end{aligned}$ | Process data transfer after sync no. 1 ... <br> Process data transfer after sync no. $240$ | The output data are accepted after transmission of the set number (1... 240) of sync telegrams. |  |
|  |  |  | 254 | Time-controlled process data transfer | Only if a cycle time is set in I180xh, 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 I180x ${ }_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms \} 65535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms\} 65535 | Cycle time |  |


| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| $\frac{11809_{h}}{1}$ |  |  |  |  |  |  | -1 256 |
|  | COB-ID used by TxPDO 10 | $\begin{array}{r} 1728 \\ + \text { NID } \end{array}$ | 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 |  |
|  |  |  | $0 \quad$ Func | deactiva |  | The output data are accepted on sync telegram transmission. |  |
|  |  |  | $1 \ldots$ Proc <br> 240 $\ldots$ <br>  Proc <br>  240 | ata trans <br> ata trans | ync no. 1 <br> ync no. | The output data are accepted after transmission of the set number (1 ... 240) of sync telegrams. |  |
|  |  |  | $254 \text { Tim }$ | rolled p | transfer | Only if a cycle time is set in I180x $x_{h}$, subindex 5 |  |
|  |  |  | 255 Eve | trolled p | ta transfer |  |  |
|  |  |  | $255 \quad$Eve <br> wit | rolled p overlay | transfer | Only if a cycle time is set in I180x $x_{h}$, subindex 5 |  |
| 3 | Inhibit time | 0 | 0 | \{1 ms \} | 65535 | Blocking time |  |
| 5 | Event time | 0 | 0 | \{1 ms \} | 65535 | Cycle time |  |
| $11 \mathrm{AOO} 0_{\mathrm{h}}$ |  |  |  |  |  | Mapping parameters for transmit PDOs |  |
| 0 | Number of mapped TxPDO1 |  | 0 | \{1\} | 255 | 8 bit value |  |
| 1 | 1st mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 2 | 2nd mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 3 | 3rd mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 4 | 4th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 5 | 5th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 6 | 6th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 7 | 7th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 8 | 8th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |

## 8 CANopen communication

## Index table

| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| $\mathrm{I} 1 \mathrm{~A} 01_{\mathrm{h}}$ |  |  |  |  |  | Mapping parameters for transmit PDOs |  |
| 0 | Number of mapped TxPDO2 |  | 0 | \{1\} | 255 | 8 bit value |  |
| 1 | 1st mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 2 | 2nd mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 3 | 3rd mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 4 | 4th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 5 | 5th mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 6 | 6th mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 7 | 7th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 8 | 8th mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| $\mathrm{I} 1 \mathrm{~A} 02_{\mathrm{h}}$ |  |  |  |  |  | Mapping parameters for transmit PDOs |  |
| 0 | Number of mapped TxPDO3 |  | 0 | \{1\} | 255 | 8 bit value |  |
| 1 | 1st mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 2 | 2nd mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 3 | 3rd mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 4 | 4th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 5 | 5th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 6 | 6th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 7 | 7th mapped object |  | 00000000h | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |
| 8 | 8th mapped object |  | $00000000_{\text {h }}$ | \{1\} | FFFFFFFF $_{\text {h }}$ | 32 bit value |  |



## 8 CANopen communication

## Index table







| Index | Name | Possible settings |  |  |  | Important |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |  |  |
| $15620{ }_{\text {h }}$ | PWM pulse duration |  | 0000 h | \{1\} | FFFFh | Status of PWM <br> 2 entries per module Specification of pulse duration in [ $\mu \mathrm{s}$ ]. | $\square 264$ |
| 1 | DWORD 1 |  |  |  |  |  |  |
| 2 | DWORD 2 |  |  |  |  |  |  |
| ... | ... |  |  |  |  |  |  |
| 80 | DWORD 64 |  |  |  |  |  |  |
| I5621 ${ }_{\text {h }}$ | PMW control |  | $0000{ }_{\text {h }}$ | \{1\} | FFFFh | Control word PWM <br> 2 entries per module <br> Bits 0 ... 1: reserved | [1356 |
| 1 | WORD 1 |  |  |  |  | Bit 2: <br> - 0: push/pull output The output signal is switched to HIGH level active and LOW |  |
| 2 | WORD 2 |  |  |  |  | level active. <br> - 1: highside output The output signal is only switched to HIGH level active. |  |
| ... | ... |  |  |  |  | Bits 3 ... 7: reserved Bit 8: 0-1 edge: PWM output starts |  |
| 80 | WORD 64 |  |  |  |  | Bit 9: 1-0 edge: PWM output stops <br> Bits 10 ... 15: reserved |  |
| 15640 ${ }_{\text {h }}$ | DO time stamp control |  | $0000_{\text {h }}$ | \{1\} | FFFFh | 15 entries per module <br> Bits 0 ... 5: reserved <br> Bit 6: channel status D01 (0: | 1-1 260 |
| 1 | DWORD 1 |  |  |  |  | FALSE; 1: TRUE) |  |
| 2 | DWORD 2 |  |  |  |  | Bit 7: channel status D02 (0: <br> FALSE; 1: TRUE) |  |
| ... | ... |  |  |  |  | from 0 ... 127 and then goes back |  |
| 80 | DWORD 64 |  |  |  |  | Bits 16 ... 32: ticker value |  |
| $16000_{h}$ | Digital input |  | 0 | \{1\} | 255 | Read only | [1356 |
| 1 | Byte 1 |  |  |  |  | Digital input status |  |
| 2 | Byte 2 |  |  |  |  |  |  |
| $\cdots$ | ... |  |  |  |  |  |  |
| 80 | Byte 80 |  |  |  |  |  |  |
| $16002_{h}$ | Change polarity digital input |  | 0 | \{1\} | 255 | Inverts digital input signals | ¢ 284 |
| 1 | Byte 1 | 0 |  |  |  |  |  |
| 2 | Byte 2 | 0 |  |  |  |  |  |
| $\cdots$ | ... | ... |  |  |  |  |  |
| 80 | Byte 80 | 0 |  |  |  |  |  |
| ${ }_{*}^{16200}{ }_{\text {h }}$ | Digital output |  | 0 | \{1\} | 255 | - Digital output status <br> - The outputs can be set | ¢ 356 |
| 1 | Byte 1 |  |  |  |  | manually (forcing): |  |
| 2 | Byte 2 |  |  |  |  | $12360_{h}$ |  |
| ... | ... |  |  |  |  |  |  |
| 80 | Byte 80 |  |  |  |  |  |  |

## 8 CANopen communication

## Index table




## $9 \quad$ PROFIBUS communication <br> Via Profibus-DP

## 9 PROFIBUS communication

## $9.1 \quad$ Via 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

Profibus-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.

## Note!

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

### 9.2 System configuration

### 9.2.1 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.

### 9.2.2 Mono-master system



Fig. 9-1 PROFIBUS-DP mono-master system
In the case of mono-master systems, only one master on the bus is active during operation. The slaves are coupled to the master via the transmission medium in a decentralised manner. This system configuration achieves the shortest bus cycle time.

### 9.2.3 Multi-master system



Fig. 9-2 PROFIBUS-DP multi-master system
A Subsystem consisting of master 1 and slaves 1 ... 3 with cyclic data transfer.
B Subsystem consisting of master 3 and slaves $4 \ldots 6$ with cyclic data transfer.
(C) For configuration and diagnostics, master 2 can communicate with slave 1 ... 6 . The data transfer is acyclic.

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

### 9.3 Communication

### 9.3.1 Bus access

The transmission protocol offers two bus access procedures.

## Master $\leftrightarrow$ 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 $\leftrightarrow$ 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.

### 9.3.2 Cyclic data transfer

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


Fig. 9-3 DP cycle and cycle of backplane bus
A Backplane bus with transmit and receive buffer
B Input / output modules PO: process image of outputs Pl : process image of inputs
(1) PROFIBUS cycle
(2) Backplane bus cycle

## Backplane bus cycle

During a backplane bus cycle

- the input data (PI) on the inputs is collected and transmitted to the transmit buffer (buffer send),
- the output data (PO) of the receive buffer (buffer receive) is written to the outputs.


## PROFIBUS cycle

During a PROFIBUS cycle, the master successively addresses all its assigned slaves with a DataExchange. During a DataExchange, the memory areas assigned to the PROFIBUS are transmitted.

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


### 9.3.3 Acyclic data transfer

The PROFIBUS-DP-V1 service can be used as an optional extension to enable an acyclic parameter data transfer. PROFIBUS-DP-V0 and PROFIBUS-DP-V1 may be operated simultaneously in one network.

The integration of the acyclic service in a fixed bus cycle depends on the correct configuration of DPM 1:

- If configured, a time slot is reserved.
- If not configured, the acyclic service is added when a DP-V1 slave is accessed acyclically with a DPM 2.
- The acyclic service always has lower priority.


## Parameter data transfer between DPM 1 and slaves



Fig. 9-4 Acyclic data transfer
$\longleftrightarrow$ Cyclic process data transfer between DPM 1 and slave $1 \ldots 3$
$4-\rightarrow$ Acyclic parameter data transfer between DPM 1 and slave 3

1. DPM 1 has the send authorisation (token) and communicates in a fixed sequence with slave 1 , then with slave 2 etc. up to the last slave of the current list via the MSO channel by means of request and response.
2. DPM 1 transfers the token to DPM 2.
3. During the remaining cycle time (time slot), DPM 2 establishes an acyclical connection to one of the slaves in order to transmit 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 acyclical data record transfer.

5. Once all of the data records have been transferred, DPM 2 establishes the connection within one time slot.

## 1 Note!

DPM 1 can perform the acyclical parameter data exchange via the MS1 channel.

## 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 <br> receipt. To prevent alarm messages from being overwritten, the slave can only <br> 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 <br> received an alarm message. |
| Status | The slave transmits a status message to the master. The master does not <br> 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 <br> 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 <br> 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.

### 9.3.4 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.


## 9.4 <br> 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: LENZOC3A.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(1)39).
- At the master, the address is set during the configuration.
- Set the baud rate
- The baud rate is set in the configuration tool.
- The baud rate must correspond to the length of the bus cable.
- Parameterise slaves


## 1 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.

## $9 \quad$ PROFIBUS communication

Setting the parameters of analog I/O
2 analog inputs 0 ... 10 V ( 12 bits) - EPM-S400

### 9.5 Setting the parameters of analog I/O

9.5.1 2 analog inputs $0 \ldots 10 \mathrm{~V}$ (12 bits) - EPM-S400

## Parameter data

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

| Data set <br> No. |  | Nyte |  | Description/value |
| :--- | :--- | :--- | :--- | :--- |$\quad$ Lenze | ( |
| :--- |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | $13824$ | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \text { V }}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 71h : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 02h) |
| 6 | Number of channels of a module (here $02_{\text {h }}$ :) |
| 7 | Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $10 . .15$ | 0 (fixed) |

### 9.5.2

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 <br> No. |  | Byte | Name | Description/value |
| :--- | :--- | :--- | :--- | :--- |
| 128 | 0 | Function channel 1 |  | Lenze |
| 129 | 0 | Function channel 2 | $16\left(10_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}}$ | $10_{\mathrm{h}}$ |
| 130 | 0 | Function channel 3 | $32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}}$ | $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated |
| 131 | 0 | Function channel 4 |  | $10_{\mathrm{h}}$ |

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] | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 |  |  |
|  | 5 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{h}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here $044_{\mathrm{h}}$ :) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $12 . .15$ | 0 (fixed) |

## $9 \quad$ PROFIBUS communication

Setting the parameters of analog $1 / 0$
2 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S402
9.5.3 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 |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| 128 | 0 | Function channel 1 | $\begin{aligned} & 48\left(30_{\mathrm{h}}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots{27648_{\mathrm{dec}}}^{64\left(40_{\mathrm{h}}\right): 4 \ldots 20 \mathrm{~mA} / 0 . . .16384_{\mathrm{dec}}} \end{aligned}$ | $31_{\text {h }}$ |
| 129 | 0 | Function channel 2 | $\begin{aligned} & 49\left(31_{h}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}} \\ & 65\left(41_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $31_{\text {h }}$ |

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.) | $\begin{aligned} & \text { Current (I) } \\ & {[\mathrm{mA}]} \end{aligned}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D * 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{\mathrm{h}}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} \mathrm{I} / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\underset{\left(40_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{l}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{h}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 02h) |
| 6 | Number of channels of a module (here $02_{\text {h }}$ :) |
| 7 | Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $10 . .15$ | 0 (fixed) |

### 9.5.4 4 analog inputs $0 / 4$... 20 mA ( $\mathbf{1 2}$ bits) - EPM-S403

## Parameter data

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

| Data set <br> No. |  | Byte | Name | Description/value |
| :--- | :--- | :--- | :--- | :--- |
| 128 | 0 | Function channel 1 | $48\left(30_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}}$ | Lenze |
| 129 | 0 | Function channel 2 | $64\left(40_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ | $31_{\mathrm{h}}$ |
| 130 | 0 | Function channel 3 | $49\left(31_{h}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}}$ | $65\left(41_{h}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ |
| 131 | 0 | Function channel 4 | $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated | $31_{\mathrm{h}}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{\mathrm{h}}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648 *(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{h}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{\mathrm{h}}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here $088_{h}$ ) |
| 6 | Number of channels of a module (here 04h:) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $12 . .15$ | 0 (fixed) |

## 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 $)$ <br> Bit 1: Channel 2 ( $0=$ deactivated; 1 = activated) <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |
| 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 Bits $7 \ldots 4:$ Reserved | $00_{\text {h }}$ |
| 128 | 0 | Function channel 1 | $\begin{aligned} & 12\left(12_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots . .27648_{\mathrm{dec}} \\ & 34\left(222_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-16384 \ldots . .16384_{\mathrm{dec}} \\ & 16(10 \mathrm{~h}): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}} \\ & 32\left(20_{\mathrm{h}}\right): 0 . .10 \mathrm{~V} / 0 \ldots . .16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
|  | 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. <br> 7FFF : Limit value alarm deactivated | $7 \mathrm{FFF}_{\text {h }}$ |
|  | 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. $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |
| 129 | 0 | Function channel 2 | $\begin{aligned} & 12\left(12_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots . .27648_{\mathrm{dec}} \\ & 34\left(22_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-16384 \ldots . .16384_{\mathrm{dec}} \\ & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
|  | 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. <br> 7FFF h : Limit value alarm deactivated | 7FFF ${ }_{\text {h }}$ |
|  | 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. $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(12_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648^{*} U / 10 \end{aligned}$ |
|  | 10 |  | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -13824 | CA00 |  |  |
|  | -10 | -27648 | 9400 |  |  |
|  | -11.76 | -32512 | 8100 | Underflow |  |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(22_{\mathrm{h}}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 | Nominal range |  |
|  | 5 | 8192 | 2000 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -8192 | E000 |  |  |
|  | -10 | -16384 | C000 |  |  |
|  | -12.5 | -20480 | B000 | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -1.76 | -4864 | ED00 | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \text { V }}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{h}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here $044_{\mathrm{h}}$ :) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $12 . .15$ | 0 (fixed) |

### 9.5.6

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 | Bits 0 ... 5: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
|  |  | Reserved | 0 |  |
|  |  | Limit value monitoring | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated $)$ <br> Bit 1: Channel 2 ( $0=$ deactivated; $1=$ activated) <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |
| 1 |  | Interference frequency suppression | Bit 1, 0: Channel 1 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bit 3, 2: Channel 2 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 128 |  | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $41_{\text {h }}$ |
|  |  | 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. <br> 7FFF ${ }_{h}$ : Limit value alarm deactivated | $7 \mathrm{FFF}_{\mathrm{h}}$ |
|  |  | 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. $8000_{h}$ : Limit value alarm deactivated | $8000_{h}$ |
| 129 |  | Function channel 2 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $41_{\text {h }}$ |
|  |  | 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. <br> $7 F F F_{h}$ : Limit value alarm deactivated | 7FFF ${ }_{\text {h }}$ |
|  |  | 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. $8000_{h}$ : Limit value alarm deactivated | $8000_{h}$ |

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


## 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available <br> Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{\mathrm{h}}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here 04 h :) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $12 . .15$ | 0 (fixed) |

## $9 \quad$ PROFIBUS communication

Setting the parameters of analog I/O
2 analog outputs 0 ... 10 V (12 bits) - EPM-S500
9.5.7 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 <br> No. |  | Byte | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0 | 1 | Short-circuit detection | Bit 0: Channel $1(0$ = deactivated; 1 = activated) <br> Bit 1: Channel 2 <br> Bits $2 \ldots 7:$ Reserved | $00_{h}$ |  |
| 128 | 0 | Function channel 1 | $16\left(10_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}}$ <br> $32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}}$ <br> $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated | $10_{\mathrm{h}}$ |  |
| 129 | 0 | Function channel 2 |  |  |  |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & \mathrm{D}=27648^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 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 anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $73_{h}$ : analog output Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 02h) |
| 6 | Number of channels of a module (here $02_{\text {h }}$ :) |
| 7 | Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .$. 1:0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4: 0 (fixed) |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 2... 1: 0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4: 0 (fixed) |
| $10 . . .15$ | 0 (fixed) |

## 9 <br> PROFIBUS communication

Setting the parameters of analog I/O
4 analog outputs 0 ... 10 V (12 bits) - EPM-S501
9.5.8

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 $)$ <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 128 | 0 | Function channel 1 | $\begin{aligned} & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots{27648_{\mathrm{dec}}}^{32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $10_{\text {h }}$ |
| 129 | 0 | Function channel 2 |  | $10_{\text {h }}$ |
| 130 | 0 | Function channel 3 |  | $10_{\text {h }}$ |
| 131 | 0 | Function channel 4 |  | $10_{\text {h }}$ |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 27648 \\ & \mathrm{D}=27648 * \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 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 anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available <br> Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $73_{\mathrm{h}}$ : analog output Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here $044_{\mathrm{h}}$ :) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 2 ... 1:0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4:0 (fixed) |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 2 ... 1: 0 (fixed) <br> Bit 3: Short circuit after M <br> Bit 7... 4: 0 (fixed) |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit 2 ... 1: 0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4: 0 (fixed) |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 2 ... 1:0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4: 0 (fixed) |
| $12 . .15$ | 0 (fixed) |

## $9 \quad$ PROFIBUS communication

Setting the parameters of analog $1 / 0$
2 analog outputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S502

### 9.5.9 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 | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated $)$ <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 128 | 0 | Function channel 1 |  | $31_{\text {h }}$ |
| 129 | 0 | Function channel 2 | $49\left(31_{\mathrm{h}}\right): 0$... $20 \mathrm{~mA} / 0 . .{27648_{\text {dec }}}^{\text {2 }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0$... 16384 ${ }_{\mathrm{dec}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | $\begin{gathered} 10 \\ 0 \end{gathered}$ | $\begin{gathered} 13824 \\ 0 \end{gathered}$ | $\begin{aligned} & 3600 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | $\begin{gathered} 10 \\ 0 \end{gathered}$ | $\begin{gathered} 8192 \\ 0 \end{gathered}$ | $\begin{aligned} & 2000 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 16 / 27648+4 \\ & D=27648 *(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | $\begin{gathered} 12 \\ 4 \end{gathered}$ | 13824 0 | $\begin{aligned} & 3600 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{h}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | $\begin{gathered} 12 \\ 4 \end{gathered}$ | $\begin{gathered} 8192 \\ 0 \end{gathered}$ | $\begin{aligned} & 2000 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | 0 | -4096 | F000 | 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)

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $73_{h}$ : analog output Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 02h) |
| 6 | Number of channels of a module (here $02_{h}$ :) |
| 7 | Bit 0: Channel error, channel 1 <br> Bit 1: Channel error, channel 2 <br> Bit 7 ... 2: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 2 ... 1:0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .$. 1: 0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| $10 . . .15$ | 0 (fixed) |

## 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 $)$ <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 128 | 0 | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $49\left(31_{h}\right): 0 . . .20 \mathrm{~mA} / 0 \ldots 27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |
| 129 | 0 | Function channel 2 |  | $31_{\text {h }}$ |
| 130 | 0 | Function channel 3 |  | $31_{\text {h }}$ |
| 131 | 0 | Function channel 4 |  | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | $0000$ |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} \mathrm{I} / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | $8192$ | $2000$ | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\underset{\left(30_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 16 / 27648+4 \\ & D=27648^{*}(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 12 | 13824 | 3600 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{h}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 |  |  |  |
|  | 0 | -4096 | F000 | 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)

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $73_{h}$ : analog output Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here 04 h :) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .1$ 1: 0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .1$ 1: 0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .1$ 1: 0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . . .1: 0$ (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| $12 . .15$ | 0 (fixed) |

### 9.6 Parameterising the temperature measurement

### 9.6.1 $\quad$ Four (two) analog inputs for resistance tests - EPM-S404

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. Bits 0 ... 5: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
|  | 1 | Wire-breakage detection | Bit 0: Channel 1 ( $0=$ inhibited; $1=$ enabled ) Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
|  | 2 | Limit value monitoring | Bit 0: Channel 1 ( $0=$ inhibited; $1=$ enabled) : <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
|  | 3 | Reserved |  |  |
| 1 | 0 | Temperature system | Bit 0, 1: $00_{b}={ }^{\circ} \mathrm{C} ; 01_{\mathrm{b}}={ }^{\circ} \mathrm{F} ; 10_{\mathrm{b}}=\mathrm{K}$ Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
|  | 1 | Interference frequency suppression | Bit 0, 1: $01_{b}=60 \mathrm{~Hz} ; 10_{b}=50 \mathrm{~Hz}$ Bits 2 ... 7: Reserved | 02h |
| Channel 1 |  |  |  |  |
| 128 | 0 | Function channel 1 | $80\left(50_{h}\right) . . .162\left(\mathrm{~A}_{\mathrm{h}}\right)$ : see "measuring range" $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $50_{\text {h }}$ |
|  | 1 | Conversion time channel 1 | You can set the transformer speed as a function of the interference frequency suppression (see $0 \times 3105 / x$ ) for each channel. <br> $0\left(00_{\mathrm{h}}\right)$ : At $50 \mathrm{~Hz}: 324.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $270.5 \mathrm{~ms} / \mathrm{channel} 16$ bits <br> 1 ( $01_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 164.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 137.2 \mathrm{~ms} / c h a n n e l ~ 16$ bits 2 (02h): At $50 \mathrm{~Hz}: 84.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 70.5 \mathrm{~ms} /$ channel 16 bits $3\left(03_{h}\right)$ : at $50 \mathrm{~Hz}: 44.1 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 37.2 \mathrm{~ms} /$ channel 16 bits 4 ( 04 h): At $50 \mathrm{~Hz}: 24.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 20.5 \mathrm{~ms} /$ channel 16 bits 5 (05h): At $50 \mathrm{~Hz}: 14.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 12.2 \mathrm{~ms} /$ channel 16 bits 6 ( 06 h ): At $50 \mathrm{~Hz}: 9.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 8.0 \mathrm{~ms} / \mathrm{channel} 16$ bits $7(07 \mathrm{~h})$ : At 50 Hz : $6.6 \mathrm{~ms} /$ channel 15 bits; at $60 \mathrm{~Hz}: 5.9 \mathrm{~ms} / c h a n n e l ~ 15$ bits 8 ( 08 h ): At $50 \mathrm{~Hz}: 4.2 \mathrm{~ms} /$ channel 13 bits; at $60 \mathrm{~Hz}: 3.8 \mathrm{~ms} / c h a n n e l ~ 13$ bits | $00_{\text {h }}$ |
|  | 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 $7 \mathrm{FFF}_{\mathrm{h}}$ for the | $7^{7 F F F_{h}}$ |
|  | 4,5 | Lower limit value channel 1 | upper or $8000_{h}$ 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. | $8000_{\text {h }}$ |

Parameterising the temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| Channel 2 |  |  |  |  |
| 129 | 0 | Function channel 2 | See channel 1 | $50_{\text {h }}$ |
|  | 1 | Conversion time channel 2 | See channel 1 | 00 h |
|  | 2, 3 | Upper limit value channel 2 | See channel 1 | $7 \mathrm{FFF}_{\mathrm{h}}$ |
|  | 4, 5 | Lower limit value channel 2 (HIGH byte) |  | $8000{ }_{\text {h }}$ |
| Channel 3 (for two-wire conductor connections only) |  |  |  |  |
| 130 | 0 | Function channel 3 | See channel 1 | $50_{\text {h }}$ |
|  | 1 | Conversion time channel 3 | See channel 1 | 00 h |
|  | 2, 3 | Upper limit value channel 3 | See channel 1 | 7FFF ${ }_{\text {h }}$ |
|  | 4, 5 | Lower limit value channel 3 |  | $8000{ }_{\text {h }}$ |
| Channel 4 (for two-wire conductor connections only) |  |  |  |  |
| 131 | 0 | Function channel 4 | See channel 1 | $50_{\text {h }}$ |
|  | 1 | Conversion time channel 4 | See channel 1 | 00 h |
|  | 2, 3 | Upper limit value channel 4 | See channel 1 | 7FFF ${ }_{\text {h }}$ |
|  | 4, 5 | Lower limit value channel 4 |  | $8000{ }_{\text {h }}$ |

## Note!

- Use parameter setting to deactivate unused inputs.
- If thermal detectors are connected in a 3 or 4 conductor setup, channels 2 and/or 4 must be deactivated.
- The module does not provide any auxiliary supply for sensors.


## Measuring range

| $\begin{array}{c}\text { Measuring range } \\ \text { (Fct. no.) }\end{array}$ | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
|  | $+1000^{\circ} \mathrm{C}$ |  |  |$)$

## $9 \quad$ PROFIBUS communication

Parameterising the temperature measurement
Four (two) analog inputs for resistance tests - EPM-S404

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 3-wire: PT100 } \\ \left(58_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| 3-wire: PT1000 (59h) | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 3-wire: NI100 } \\ \left(5 A_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 3-wire: NI1000 (5Bh) | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| $\begin{aligned} & \text { 4-wire: PT100 } \\ & \left(60_{h}\right) \end{aligned}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| 4-wire: PT1000 (61h) | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 4-wire: NI100 } \\ \left(62_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | +2950 ${ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 4-wire: NI1000 (63h) | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(70_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . .60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots . .600 \Omega \\ \left(71_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .600 \Omega$ | $0 \ldots 32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(72_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | 0 ... 32767 dec | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(78_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 600 \Omega \\ \left(79_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(7 A_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(80_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |

Parameterising the temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(81_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(82_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | 0 ... 32767 | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(90_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .60 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots . .600 \Omega \\ \left(91_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(92_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(98_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 \ldots 6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(99_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(9 A_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 3000 \Omega$ | $0 . . .30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(\mathrm{AO}_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 \ldots 6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(\mathrm{~A} 1_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{~A} 2_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | 0...30000 ${ }_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(D 0_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 600 \Omega \\ \left(\mathrm{D} 1_{\mathrm{h}}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{D} 2_{h}\right) \end{gathered}$ | 3528 ת | $32511_{\text {dec }}$ | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(D 8_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |


| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { 3-wire: } 0 \ldots . .600 \Omega \\ & (\mathrm{D} 9 \mathrm{~h}) \end{aligned}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{DA}_{h}\right) \end{gathered}$ | 3528 ת | $32511_{\text {dec }}$ | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(E O_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | -.. 27648 | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(E 1_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(E 2_{h}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |

## Diagnostics and alarm

| Trigger | Process alarm | Diagnostic alarm | Parameterisable |
| :--- | :---: | :---: | :---: |
| Configuration/parameterisation error | - | 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 |
| :---: | :--- |
| 8 | Bit 0: Limit value exceeded channel 1 <br> Bit 1: Limit value exceeded channel 2 <br> Bit 2: Limit value exceeded channel 3 <br> Bit 3: Limit value exceeded channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 9 | Bit 0: Limit value not reached channel 1 <br> Bit 1: Limit value not reached channel 2 <br> Bit 2: Limit value not reached, channel 3 <br> Bit 3: Limit value not reached, channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| $10 \ldots 11$ | Ticker value at the time of the alarm <br> After mains connection, a timer ( $\mu \mathrm{s}$ ticker) is started which after $65535 \mu \mathrm{~s}$ starts with 0 again. |

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00 h ). 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 processingincoming 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 $_{\text {incoming }}$ 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 are lost. During this time (first diagnostic alarm 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 incoming/outgoing.
Example:


## 9 <br> PROFIBUS communication

Parameterising the temperature measurement
Four (two) analog inputs for resistance tests - EPM-S404

## Diagnostic alarm processing:

Using 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 ca no longer be clearly assigned to the last diagnostic alarm.
Data set 1 is structured as follows:

| Data set 1, diagnostics ${ }_{\text {incoming }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $1000_{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here 08 h ) |
| 6 | Number of channels of a module (here $01_{h}$ ) |
| 7 | Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed) |
| 8 | Diagnostic alarm due to process alarm lost to ... <br> Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7... 5: 0 (fixed) |
| $9 . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message outgoing is effected $^{\text {d }}$ |

### 9.6.2 Two 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. Bits 0 ... 5: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
|  | 1 | Wire-breakage detection | Bit 0: Channel 1 ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
|  | 2 | Limit value monitoring | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
|  | 3 | Reserved | 0 |  |
| 1 | 0 | Temperature system | Bit 0, 1: $00_{b}={ }^{\circ} \mathrm{C} ; 10_{\mathrm{b}}={ }^{\circ} \mathrm{F} ; 11_{\mathrm{b}}=\mathrm{K}$ Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
|  | 1 | Interference frequency suppression | Bit 0, 1: $01_{b}=60 \mathrm{~Hz} ; 10_{b}=50 \mathrm{~Hz}$ Bits 2 ... 7: Reserved | $02_{\text {h }}$ |
| Channel 1 |  |  |  |  |
| 128 | 0 | Function channel 1 | ```\(176\left(60_{h}\right)\)... 201 (C9h): see "measuring range" External temperature compensation: 176 (60h) ... 185 (69h) Internal temperature compensation: \(192\left(\mathrm{CO}_{\mathrm{h}}\right)\) : ... 201 (C9h) 255 ( \(\mathrm{FF}_{\mathrm{h}}\) ): channel deactivated``` | $\mathrm{C} 1_{\text {h }}$ |
|  | 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. <br> $0\left(00_{h}\right)$ : At $50 \mathrm{~Hz}: 324.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $270.5 \mathrm{~ms} / \mathrm{channel} 16$ bits <br> $1\left(01_{h}\right)$ : at 50 Hz : $164.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $137.2 \mathrm{~ms} /$ channel 16 bits <br> 2 (02h): At $50 \mathrm{~Hz}: 84.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $70.5 \mathrm{~ms} / \mathrm{channel} 16$ bits <br> $3\left(03_{h}\right)$ : at $50 \mathrm{~Hz}: 44.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $37.2 \mathrm{~ms} /$ channel 16 bits <br> 4 ( 04 h): At $50 \mathrm{~Hz}: 24.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $20.5 \mathrm{~ms} / \mathrm{channel} 16$ bits <br> 5 ( 05 h ): at $50 \mathrm{~Hz}: 14.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $12.2 \mathrm{~ms} /$ channel 16 bits <br> $6\left(06_{h}\right)$ : at 50 Hz : $9.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $8.0 \mathrm{~ms} / \mathrm{channel} 16$ bits <br> $7\left(07_{h}\right)$ : at 50 Hz : $6.6 \mathrm{~ms} /$ channel 15 bits; at 60 Hz : <br> $5.9 \mathrm{~ms} /$ channel 15 bits <br> $8(08 \mathrm{~h})$ : at $50 \mathrm{~Hz}: 4.2 \mathrm{~ms} /$ channel 13 bits; at 60 Hz : <br> $3.8 \mathrm{~ms} / \mathrm{channel} 13$ bits | 02h |
|  | 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 $7 \mathrm{FFF}_{\mathrm{h}}$ for the | $7 \mathrm{FFF}_{\mathrm{h}}$ |
|  | 4, 5 | Lower limit value channel 1 | upper or $8000_{h}$ for the lower limit value, the corresponding limit value is deactivated. <br> If the measured value is outside a limit value and the limit value monitoring is activated, a process alarm is triggered. | 8000 h |

PROFIBUS communication
Parameterising the temperature measurement
Two analog inputs for thermocouple measurement - EPM-S405

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
|  | Byte |  |  |  |
| Channel 2 |  |  |  |  |
| 129 | 0 | Function channel 2 | See channel 1 | $\mathrm{C} 1_{\text {h }}$ |
|  | 1 | Conversion time channel 2 | See channel 1 | 02 h |
|  | 2, 3 | Upper limit value channel 2 | See channel 1 | 7FFF ${ }_{\text {h }}$ |
|  | 3, 4 | Lower limit value channel 2 |  | $8000{ }_{\text {h }}$ |

## Measuring range



| Measuring range | Measured value |  |  | Range |
| :---: | :---: | :---: | :---: | :---: |
| (Fct. no.) | [ ${ }^{\circ} \mathrm{C}$ ] | [ $\left.{ }^{\circ} \mathrm{F}\right]$ | [K] |  |
| Type C: $0 \ldots+2315^{\circ} \mathrm{C}$ | +25000 | 32766 | 23432 | Overflow |
| $\begin{gathered} 32 \ldots .2786 .5^{\circ} \mathrm{F} \\ 273.2 \ldots 2093.2 \mathrm{~K} \end{gathered}$ | $0 \ldots+23150$ | 320 ... 27865 | 2732 ... 20932 | Nominal range |
| (B7 ${ }_{h}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> $\left(C 7_{h}\right.$ : int. comp. $0{ }^{\circ} \mathrm{C}$ ) | -1200 | -1840 | 1532 | Underflow |
| Type E: $-270 \ldots+1000^{\circ} \mathrm{C}$ | +12000 | 21920 | 14732 | Overflow |
| $\begin{gathered} -454 \ldots 1832^{\circ} \mathrm{F} \\ 0 \ldots 1273.2 \mathrm{~K} \end{gathered}$ | $-2700 \ldots+10000$ | -4540 ... 18320 | 0 ... 12732 | Nominal range |
| (B8 ${ }_{h}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C8 ${ }_{h}$ : int. comp. $0{ }^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |
| Type L: $-200 \ldots+900^{\circ} \mathrm{C}$ | +11500 | 21020 | 14232 | Overflow |
| $\begin{gathered} -328 \ldots 1652^{\circ} \mathrm{F} \\ 73.2 \ldots 1173.2 \mathrm{~K} \end{gathered}$ | -2000 ... +9000 | -3280 ... 16520 | 732 ... 11732 | Nominal range |
| (B9 ${ }_{h}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C9 ${ }_{h}$ : int. comp. $0{ }^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |

Diagnostics and alarm

| Trigger | Process alarm | Diagnostic alarm | Parameterisable |
| :--- | :---: | :---: | :---: |
| Configuration/parameterisation error | - | 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 ... $\mathbf{0}$ |
| :---: | :--- |
| 8 | Bit 0: Limit value exceeded channel 1 <br> Bit 1: Limit value exceeded channel 2 <br> Bit 7 ... 2: 0 (fixed) |
| 9 | Bit 0: Limit value not reached, channel 1 <br> Bit 1: Limit value not reached, channel 2 <br> Bit 7 ... 2: 0 (fixed) |
| $10 \ldots 11$ | Ticker value at the time of the alarm <br> After mains connection, a timer ( $\mu \mathrm{s}$ ticker) is started which after $65535 \mu$ starts with 0 again. |

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00 h ). 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 $O B 40$ is interrupted and branched to the diagnostic alarm processingincoming 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 ${ }_{\text {incoming }}$ 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 are lost. During this time (first diagnostic alarm 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 incoming/outgoing.
Example:


## Diagnostic alarm processing:

Using 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 ca no longer be clearly assigned to the last diagnostic alarm.
Data set 1 is structured as follows:

| Data set 1, diagnostics incoming $^{\text {a }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $\mathbf{1 0 0 0}_{b}$ : function module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here $08{ }_{\text {h }}$ ) |
| 6 | Number of channels of a module (here $01{ }_{h}$ ) |
| 7 | Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed) |
| 8 | Diagnostic alarm due to process alarm lost to ... <br> Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7 ... 5: 0 (fixed) |
| $9 . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message outgoing is effected $^{\text {d }}$ |

## $9 \quad$ PROFIBUS communication

Parameterising the counter
One counter 32 bits, 24 V DC - EPM-S600

### 9.7 Parameterising the counter

### 9.7.1 One 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 |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> range. |
| Counting periodically |  | | Signal evaluation |
| :--- |
| Single rotary <br> transducer |
| Double rotary <br> transducer |
| Connection to input "A/pulse" and "B/direction" |
| Quadruple rotary |
| transducer |


| Additional functions | Description |
| :---: | :---: |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate), hardware gate (HW gate), and software gate (SW gate) is made. <br> - The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). <br> - The SW gate is controlled via your user program (status word in the output area). <br> - The HW gate is controlled via the digital gate input. <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

## 9 <br> PROFIBUS communication

Parameterising the counter
One counter 32 bits, 24 V DC - EPM-S600

## Read data: 12 bytes

| Input area |  |  |
| :--- | :--- | :--- |
| Addr. | Access | Assignment |
| +0 | Double word | Count value |
| +4 | Double word | Latch value |
| +8 | Word | Status word (see the following table) |
| +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 ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu$ s value together with the count value in the input area.

| EPM-S600 status word |  |  |
| :---: | :---: | :---: |
| Bit | Designation | 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 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 for counter direction backwards |
| 8 | STS_C_UP | Status set for counter direction forwards |
| 9 | STS_CMP* | Comparator status is set if the comparison condition is met. If the comparison is deactivated (counter mode byte $1=000_{b}$ ), the bit has no function. |
| 10 | STS_END* | Status set if final value was reached |
| 11 | STS_OFLW* | Status set in the case of overflow |
| 12 | STS_UFLW* | Status set in the case of underflow |
| 13 | STS_ZP* | Status set in the case of zero crossing |
| 14 | STS_LTCH | Status of latch input |
| 15 | - | Reserved |

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


## Write data: 10 bytes

Output area

| Addr. | Access | Assignment |
| :--- | :--- | :--- |
| +0 | Double word | Comparison value |
| +4 | Double word | Set value |
| +8 | Word | Control word (see the following table) |

Comparison value: Here you can select a value which, by comparison with the current counter content, can impact the counter output or trigger a process alarm. The response of the output or 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.

| EPM-S600 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | Function |
| 0 | CTRL_SYNC_SET | Activates the reset mode |
| 1 | CTRL_DO_SET | Enables the digital output |
| 2 | SW_GATE_SET | Sets the software gate |
| 3 | - | Reserved |
| 4 | - | Reserved |
| 5 | COUNTERVAL_SET | Sets the counter temporarily to the value in the set value |
| 6 | RES_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | CTRL_SYNC_RESET | Reserved |
| 8 | CTRL_DO_RESET | Inhibits the digital output |
| 9 |  |  |
| 10 |  | RW_GATE_RESET |

Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $00_{\text {h }}$ | 0 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bits 5 ... 0 : reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled) <br> Bit 7: Reserved | $00_{\text {h }}$ |
| $01_{\text {h }}$ | 0 | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02 h |
|  | 1 | Input frequency track B |  | $02_{\text {h }}$ |
|  | 2 | Input frequency latch |  | 02 h |
|  | 3 | Input frequency gate |  | $02_{\text {h }}$ |
|  | 4 | Input frequency reset |  | $00_{\text {h }}$ |
|  | 5 | Reserved |  |  |
| $80_{\text {h }}$ | 0 | Alarm response | Setting activates process alarm <br> Bit 0: Proc. alarm HW gate open <br> Bit 1: Proc. alarm HW gate closed <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bit 6: Proc. alarm latch value <br> Bit 7: Reserved | $80_{\text {h }}$ |


| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
|  | 1 | Counter function | Bit 5 ... 0 : <br> $000000_{b}=$ counting continuously <br> $000001_{b}=$ counting once, main counting direction <br> forwards <br> $000010_{b}=$ counting once, main counting direction backwards <br> $000100_{b}=$ counting once, no main counting direction <br> $001000_{b}=$ counting periodically, main counting <br> direction forwards <br> $010000_{b}=$ counting periodically, main counting direction backwards <br> $100000_{b}=$ counting periodically, no main counting direction <br> Bits 7 ... 6: Reserved | $40_{\text {h }}$ |
|  | 2 | Comparator | ```Bit 2 ... 0: output switches (... if condition is met) \(000_{b}=\) never \(001_{b}=\) count value \(\geq\) comparison value \(010_{b}=\) count value \(\leq\) comparison value \(100_{b}=\) count value \(=\) comparison value Bit 3: Invert counting direction track B \(0=\) no (do not invert) 1 = yes (invert) Bits 6 ... 4: Reset \(000_{b}=\) deactivated \(001_{b}=\) HIGH level \(011_{b}=\) rising edge \(101_{b}=\) rising edge, once Bit 7: Reserved``` | 00 h |
|  | 3 | Signal evaluation | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bit 6 ... 3: Hardware gate (HW gate) <br> $000_{b}=$ deactivated (counter starts by setting SW gate) <br> $001_{b}=$ activated (HIGH level at gate activates the HW <br> gate. Counter starts if HW and SW gate are set.) <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | 00 h |
| $81_{\text {h }}$ | 0 | Final value | Counting method: <br> 0x310B: byte 3 (high byte) <br> 0x3110: byte 2 <br> 0x3111: byte 1 <br> $0 \times 3112$ : byte 0 (low byte) | 00 h |
|  | 1 | Loading value | Counting method: 0x310B: byte 3 (high byte) <br> 0x310C: byte 2 <br> 0x310D: byte 1 <br> $0 \times 310 \mathrm{E}$ : byte 0 (low byte) | $00_{\text {h }}$ |


| Data set |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- |
| No. | Byte | Name | Description/value | Lenze |
| 2 | Hysteresis | The hysteresis for instance serves to avoid frequent <br> switching operations of the output and/or triggering of <br> the alarm when the count value is within the range of the <br> comparison value. For the hysteresis you can select a <br> range between 0 and 255. With the settings 0 and 1 the <br> hysteresis is switched off. <br> The hysteresis has an effect on the zero crossing, <br> comparison, overflow/underflow. | $00_{\mathrm{h}}$ |  |
| 3 | Pulse | The pulse duration indicates for how long the output is to <br> be set if the parameterised comparison criterion is <br> reached or exceeded. The pulse duration can be specified <br> in steps of 2.048 ms between 0 and 522.24 ms. If the pulse <br> duration is =0, the output is set until the comparison <br> condition is no longer met. | $00_{\mathrm{h}}$ |  |

## 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 ... $\mathbf{0}$ |
| :---: | :--- |
| Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7 .. 5: 0 (fixed) |  |
| 9 | State of the inputs at the time of the alarm <br> Bit 0: A/pulse <br> Bit 1: B/direction <br> Bit 2: Latch <br> Bit 3: Hardware gate <br> Bit 4: Reset <br> Bit 7 ... 5: 0 (fixed) |
| $10 \ldots 11$ | 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 ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu \mathrm{s}$ value together with the count value in the input area.

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00h). 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 $O B 40$ is interrupted and branched to the diagnostic alarm processingincoming 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 are lost. During this time (first diagnostic alarm 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 incoming/outgoing.
Example:


## Diagnostic alarm processing:

Using 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 ca no longer be clearly assigned to the last diagnostic alarm.
Data set 1 is structured as follows:

| Data set 1, diagnostics ${ }_{\text {incoming }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $\mathbf{1 0 0 0}_{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here $01_{h}$ ) |
| 7 | Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed) |
| 8 | Diagnostic alarm due to process alarm lost to ... <br> Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7... 5: 0 (fixed) |
| $9 . . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message outgoing $^{\text {is effected }}$ |

### 9.7.2

Two 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 |
| :--- | :--- |
| Counting | The counter counts from the loading value to the counting limit, then skips to the |
| continuously | opposite counting limit and continues to count from there. |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input " $\mathrm{A} / \mathrm{pulse}$ " and direction at " $\mathrm{B} /$ direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

## 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 (see following table) |
| +10 | Word | Counter 2: status word (see following table) |

Count value: Current counter content

| EPM-S601 | status word |  |
| :--- | :--- | :--- |
| Bit | Designation | 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_UP | Status set for counter direction backwards |
| 8 | STS_CMP* | Status set for counter direction forwards |
| 9 | STS_OFLW* | Comparator status is set if the comparison condition is met. <br> If the comparison is deactivated (counter mode byte $1=000_{\text {b }}$ ), the bit has <br> no function. |
| 10 | STS_UFLW* | Status set if final value was reached |
| 11 | STS_ZP* | Status set in the case of overflow |
| 12 | - | Status set in the case of underflow |
| 13 | Status set in the case of zero crossing |  |
| 14 | Reserved |  |
| 15 | Reserved |  |

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


## 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 (see following table) |
| +10 | Word | Counter 2: Control word (see following table) |

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 set $80_{h}$ for counter 0 and $82_{h}$ for counter 1 .

## Parameterising the counter

Two counters 32 bits, 24 V DC - EPM-S601

| EPM-S601 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | 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 | Sets the counter temporarily to the value in the set value |
| 6 | RES_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | - | Reserved |
| 8 | CTRL_COMP_RESET | Reserved |
| 9 | SW_GATE_RESET | Resets the software gate |
| 10 | - | Reserved |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| 00 h | 0 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bits 5 ... 0: reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | 00 h |
| $01_{\text {h }}$ | 0 | Input frequency counter 1, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02 h |
|  | 1 | Input frequency counter 1, track B |  | $02_{\text {h }}$ |
|  | 2 | Input frequency counter 2, track A |  | 02h |
|  | 3 | Input frequency counter 2, track B |  | 02 h |
| $80_{\text {h }}$ | 0 | Alarm response counter 1 | Setting activates process alarm <br> Bits 1 ... 0: reserved <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bits 7 ... 6: Reserved | $00_{\text {h }}$ |
|  | 1 | Counter function counter 1 | ```Bit 5 ... 0: 000000 000001 b = once: forwards 000010b = once: backwards 000100b = once: no main counting direction 001000 b = periodically: forwards 0100000b = periodically: backwards 100000b}=\mathrm{ = periodically: no main counting direction Bits 7 ... 6: Reserved``` | 00 h |


| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
|  | 2 | Comparator counter 1 | Bits 2 ... 0 : Comparison bit is set (... if condition is met) $000_{b}=$ never <br> $001_{b}=$ count value $\geq$ comparison value <br> $010_{b}=$ count value $\leq$ comparison value <br> $100_{b}=$ count value $=$ comparison value <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{h}$ |
|  | 3 | Signal evaluation counter 1 | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter <br> details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at A and B) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: Reserved <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{\text {h }}$ |
| $81_{\text {h }}$ | 0... 3 | Set value counter 1 | Counting method: <br> $0 \times 3111$ : byte 3 (high byte) <br> 0x3112: byte 2 <br> 0x3113: byte 1 <br> $0 \times 3114$ : byte 0 (low byte) | $00_{\text {h }}$ |
|  | 4... 7 | Final value counter 1 | Counting method: 0x310D: byte 3 (high byte) <br> $0 \times 310$ E: byte 2 <br> $0 \times 310$ F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
|  | 8... 11 | Loading value counter 1 | Counting method: <br> $0 \times 3109$ : byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> $0 \times 310 \mathrm{C}$ : byte 0 (low byte) | $00_{\text {h }}$ |
|  | 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. <br> The hysteresis has an effect on the zero crossing, comparison, overflow/underflow. | $00_{\text {h }}$ |
|  | 13 | Reserved |  |  |
| $82_{\text {h }}$ | 0 | Alarm response counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 1 | Counter function counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 2 | Comparator counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 3 | Signal evaluation counter 2 | See counter 1 | $00_{\text {h }}$ |
| 83 h | 0... 3 | Set value counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 4... 7 | Final value counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 8... 11 | Loading value counter 2 | See counter 1 | $00_{\text {h }}$ |


| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
|  | 12 | Hysteresis counter 2 | See counter 1 | $00_{\text {h }}$ |

## Process alarm

A process alarm causes a call of the $O B 40$. Within the $O B 40$ 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 $7 . .0$ |
| :---: | :---: |
| 8 | Bit 0: 0 <br> Bit 1: 0 <br> Bit 2: Counter 0, overflow/underflow/final value <br> Bit 3: Counter 0, comparison value reached <br> Bit 4: 0 <br> Bit 5: 0 <br> Bit 6: Counter 1, overflow/underflow/final value <br> Bit 7: Counter 1, comparison value reached |
| 9 | State of the inputs at the time of the alarm <br> Bit 0: Counter 0, A/pulse <br> Bit 1: Counter 0, $\mathrm{B} /$ direction <br> Bit 2: Counter 1, A/pulse <br> Bit 3: Counter 1, B/direction <br> Bit 7 ... 4: 0 (fixed) |
| $10 . . .11$ | 16 bit $\mu \mathrm{s}$ value at the time of the alarm |

Ticker value: After mains connection a timer ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu \mathrm{s}$ value together with the count value in the input area.

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00h). 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 $O B 40$ is interrupted and branched to the diagnostic alarm processingincoming 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 ${ }_{\text {incoming }}$ 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 incoming and diagnostic alarm outgoing are not buffered and are lost. During this time (first diagnostic alarm 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 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, diagnostics incoming $^{\text {a }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $\mathbf{1 0 0 0}_{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here 02 h ) |
| 7 | Bit 0: Error in channel group 0 (counter 0) <br> Bit 1: Error in channel group 1 (counter 1) <br> Bit 7... 2: 0 (fixed) |
| 8 | Channel group 0: Diagnostic alarm Idue to lost process alarm to ... <br> Bit $1 . . .0: 0$ (fixed) <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 7... 4: 0 (fixed) |
| 9 | Channel group 1: Diagnostic alarm Idue to lost process alarm to ... <br> Bit $1 . . .0: 0$ (fixed) <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 7... 4: 0 (fixed) |
| $10 . . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message outgoing is effected $^{\text {d }}$ |

### 9.7.3 One 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 |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting |
| Counting periodically | range. |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input " $\mathrm{A} / \mathrm{pulse}$ " and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

## $9 \quad$ PROFIBUS communication

## Parameterising the counter

One counter 32 bits, 5 V DC - EPM-S602

## Read data: $\mathbf{8}$ bytes

| Input area |  |  |
| :--- | :--- | :--- |
| Addr. | Access | Assignment |
| +0 | Double word | Count value |
| +4 | Word | Status word (see the following table) |
| +6 | Word | Ticker value |

## Count value: Current counter content

Ticker value: After mains connection a timer ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu s$ value together with the count value in the input area.

| EPM-S602 status word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | 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 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 for counter direction backwards |
| 8 | STS_C_UP | Status set for counter direction forwards |
| 9 | STS_CMP* | Comparator status is set if the comparison condition is met. <br> If the comparison is deactivated (counter mode byte $\left.1=000_{b}\right)$, the bit has <br> no function. |
| 10 | STS_END* | Status set if final value was reached |
| 11 | STS_OFLW* | Status set in the case of overflow |
| 12 | STS_UFLW* | Status set in the case of underflow |
| 13 | STS_ZP* | Status set in the case of zero crossing |
| 14 | - | Reserved |
| 15 | - | Reserved |
|  |  |  |

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


## Write data: $\mathbf{1 0}$ bytes

| Output area |  |  |
| :--- | :--- | :--- |
| Addr. | Access | Assignment |
| +0 | Double word | Comparison value |
| +4 | Double word | Set value |
| +8 | Word | Control word (see the following table) |

Comparison value: Here you can select a value which, by comparison with the current counter content, can impact the counter output or trigger a process alarm. The response of the output or 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.

| EPM-S602 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | Function |
| 0 | CTRL_SYNC_SET | Activates the reset mode |
| 1 | CTRL_COMP_SET | Enables the comparison bit |
| 2 | SW_GATE_SET | Sets the software gate |
| 3 | - | Reserved |
| 4 | - | Reserved |
| 5 | COUNTERVAL_SET | Sets the counter temporarily to the value in the set value |
| 6 | RES_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | - | Reserved |
| 8 | CTRL_SYNC_RESET | Deactivates the reset mode |
| 9 | CTRL_COMP_RESET | Inhibits the comparison bit |
| 10 | SW_GATE_RESET | Resets the software gate |
| 11 | - | Reserved |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $00_{h}$ | 0 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bits 5 ... 0: reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| $01_{\text {h }}$ | 0 | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $2\left(02_{h}\right): 100 \mathrm{kHz}$ <br> 3 (03h): 60 kHz <br> 4 (04h): 30 kHz <br> 6 (06h): 10 kHz <br> $7\left(07_{h}\right): 5 \mathrm{kHz}$ <br> 8 (08h): 2 kHz <br> 9 (09h): 1 kHz <br> Other values are not permissible! | $02_{\text {h }}$ |
|  | 1 | Input frequency track B |  | $02_{\text {h }}$ |
|  | 2 | Input frequency reset |  | 02h |
|  | 3 | Reserved |  |  |
| $80_{\text {h }}$ | 0 | Alarm response | Setting activates process alarm Bits 1 ... 0: reserved <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bits 7 ... 6: Reserved | $00_{\text {h }}$ |
|  | 1 | Counter function | Bit 5 ... 0 : <br> $000000_{b}=$ counting continuously <br> $000001_{b}=$ once: forwards <br> $000010_{b}=$ once: backwards <br> $000100_{b}=$ once: no main counting direction <br> $001000_{b}=$ periodically: forwards <br> $010000_{b}=$ periodically: backwards <br> $100000_{b}=$ periodically: no main counting direction <br> Bits 7 ... 6: Reserved | $00_{\text {h }}$ |


| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
|  | 2 | Comparator | Bits 2 ... 0 : Comparison bit is set (... if condition is met) <br> $000_{b}=$ never <br> $001_{b}=$ count value $\geq$ comparison value <br> $010_{b}=$ count value $\leq$ comparison value <br> $100_{b}=$ count value $=$ comparison value <br> Bit 3: Invert counting direction track B <br> 0 = no (do not invert) <br> 1 = yes (invert) <br> Bits 6 ... 4: Reset <br> $000_{b}=$ deactivated <br> $001_{b}=$ HIGH level <br> $011_{b}=$ rising edge <br> $101_{b}=$ rising edge, once <br> Bit 7: Reserved | $00_{\text {h }}$ |
|  | 3 | Signal evaluation | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter <br> details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: Reserved <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{\text {h }}$ |
| $81_{\text {h }}$ | 4... 7 | Final value | Counting method: <br> 0x310D: byte 3 (high byte) <br> $0 \times 310 \mathrm{E}$ : byte 2 <br> 0x311F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
|  | 8... 11 | Loading value | Counting method: <br> 0x3109: byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> 0x310C: byte 0 (low byte) | $00_{h}$ |
|  | 12 | Hysteresis |  | $00_{h}$ |

## 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 $7 . .0$ |
| :---: | :---: |
| 8 | Bit 0: 0 <br> Bit 1: 0 <br> Bit 2: Counter 0, overflow/underflow/final value <br> Bit 3: Counter 0, comparison value reached <br> Bit 4: 0 <br> Bit 5: 0 <br> Bit 6: Counter 1, overflow/underflow/final value <br> Bit 7: Counter 1, comparison value reached |
| 9 | State of the inputs at the time of the alarm <br> Bit 0: Counter 0, A/pulse <br> Bit 1: Counter $0, \mathrm{~B} /$ direction <br> Bit 2: Counter 1, $\mathrm{A} /$ pulse <br> Bit 3: Counter 1, B/direction <br> Bit 7... 4: 0 (fixed) |
| $10 . .11$ | 16 bit $\mu s$ value at the time of the alarm |

Ticker value: After mains connection a timer ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu$ s value together with the count value in the input area.

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00 h ). 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 $O B 40$ is interrupted and branched to the diagnostic alarm processing incoming 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 ${ }_{\text {incoming }}$ 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 incoming and diagnostic alarm outgoing are not buffered and are lost. During this time (first diagnostic alarmincoming 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 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, diagnostics incoming $^{\text {a }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $\mathbf{1 0 0 0}_{b}$ : function module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0: 0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here 08 h ) |
| 6 | Number of channels of a module (here 02 h ) |
| 7 | Bit 0: Error in channel group 0 (counter 0) <br> Bit 1: Error in channel group 1 (counter 1) <br> Bit 7 ... 2: 0 (fixed) |
| 8 | Channel group 0: Diagnostic alarm Idue to lost process alarm to ... <br> Bit 1... 0:0 (fixed) <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 7... 4: 0 (fixed) |
| 9 | Channel group 1: Diagnostic alarm Idue to lost process alarm to ... <br> Bit 1... 0:0 (fixed) <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 7... 4: 0 (fixed) |
| $10 . . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message ${ }_{\text {outgoing }}$ is effected |

## 9 PROFIBUS communication

## Parameterising the counter

Two counters 32 bits, 24 V DC - EPM-S603

### 9.7.4 Two 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 |
| :--- | :--- |
| Counting <br> continuously | The counter counts from 0 to the counting limit, then skips to the opposite counting limit <br> and continues to count from there. |
| Signal evaluation <br> Single rotary <br> transducer | Description |
| Double rotary <br> transducer | Connection to input "A/pulse" and "B/direction" |
| Quadruple rotary <br> transducer | Pulse at input "A/pulse" and direction at "B/direction" |
| Direction | Description <br> The gate function serves to start, stop, and interrupt a counting function. In the case of <br> this counter the internal gate (I-gate) is conform to the software gate (SW gate) which <br> you control via your user program (status word in the output area). |
| Additional functions |  |
| Gate function |  |

Read data: 12 bytes

| Addr. | Access | Assignment |
| :---: | :---: | :---: |
| +0 | Double word | Counter 1: count value |
| +4 | Double word | Counter 2: count value |
| +8 | Word | Counter 1: status word (see following table) |
| +10 | Word | Counter 2: status word (see following table) |
| EPM-S603 status word |  |  |
| Bit | Designation | 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 for counter direction backwards |
| 8 | STS_C_UP | Status set for counter direction forwards |
| 9 | - | Reserved |
| 10 | - | Reserved |
| 11 | STS_OFLW* | Status set in the case of overflow |
| 12 | STS_UFLW* | Status set in the case of underflow |
| 13 | STS_ZP* | Status set in the case of zero crossing |
| 14 | - | Reserved |
| 15 | - | Reserved |

[^6]
## Write data: 4 bytes

Output area in the process image

| Addr. | Access | Assignment |
| :--- | :--- | :--- |
| +0 | Word | Counter 1: Control word (see following table) |
| +2 | Word | Counter 2: Control word (see following table) |


| EPM-S603 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | Function |
| 0 | - | Reserved |
| 1 | - | Reserved |
| 2 | SW_GATE_SET | Sets the software gate |
| 3 | - | Reserved |
| 4 | - | Reserved |
| 5 | - | Reserved |
| 6 | - | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | - | Reserved |
| 8 | SW_GATE_RESET | Reserved |
| 9 | - | Reserved |
| 10 |  | Resets the software gate |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $01_{\text {h }}$ | 0 | Input frequency counter 1, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | $02_{\text {h }}$ |
|  | 1 | Input frequency counter 1, track B |  | 02h |
|  | 2 | Input frequency counter 2, track A |  | 02 h |
|  | 3 | Input frequency counter 2, track B |  | $02_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Counting direction counter 1, track B | Bits 2 ... 0: Reserved <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{h}$ |
|  | 1 | Signal evaluation counter 1 | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 7 ... 3: Reserved | $00_{\text {h }}$ |
| $82^{\text {h }}$ | 0 | Counting direction counter 2, track B | Bits 2 ... 0: Reserved <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
|  | 1 | Signal evaluation counter 2 | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter <br> details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 7 ... 3: Reserved | $00_{\text {h }}$ |

## Diagnostic data

Using the SFB 52 you can access the diagnostic data of the module anytime. 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, $1000_{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here $00_{h}$ ) |
| 6 | Number of channels of a module (here 02 h ) |
| $7 . .15$ | 0 (fixed) |

### 9.8 Parameterising the encoder evaluation

### 9.8.1 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 <br> 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 ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the encoder value the time value of the timer is stored as 16 -bit $\mu$ s value together with the encoder value in the input area.

Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| 00 h | 0 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bits 5 ... 0 : reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Dead 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. <br> HIGH LOW <br> $00_{h} 30_{h}: 1 \mu \mathrm{~s}$ <br> $00_{h} 60_{h}: 2 \mu \mathrm{~s}$ <br> $00_{h} \mathrm{CO}_{\mathrm{h}}: 4 \mu \mathrm{~s}$ <br> $01_{h} 80_{h}: 8 \mu s$ <br> $03_{\mathrm{h}} 00_{\mathrm{h}}: 16 \mu \mathrm{~s}$ <br> $06_{h} 00_{h}: 32 \mu \mathrm{~s}$ <br> $09_{\mathrm{h}} 00_{\mathrm{h}}: 48 \mu \mathrm{~s}$ <br> $0 C_{h} 00_{h}: 64 \mu s$ | 0 COO h |



| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
|  | 5 |  | Bit 1 ... 0: Ready for operation <br> 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. <br> In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder. <br> $01_{b}=$ monitoring operation <br> $10_{b}=$ master mode <br> Bit 2: Shifting direction <br> 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". <br> $0=$ LSB first (LSB is transmitted first) <br> $1=$ MSB first (MSB is transmitted first) <br> Bit 3: edge clock signal <br> Here you can specify the edge type of the clock signal, in the case of which the encoder supplies data. <br> Information on this can be found in the data sheet for your encoder. Usually the SSI encoders respond to rising edges. <br> 0 = falling edge <br> 1 = rising edge <br> Bit 4: Coding <br> 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. <br> $0=$ standard code <br> 1 = Gray code <br> Bits 7 ... 5: reserved | $1 E_{h}$ |
|  | 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. <br> $0\left(00_{h}\right)=$ inhibited <br> $1\left(01_{h}\right)=$ enabled | 00 h |

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00 h ). 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 ${ }_{\text {incoming }}$ 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 $_{\text {incoming }}$ 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 are lost. During this time (first diagnostic alarm 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 incoming/outgoing.
Example:


## Diagnostic alarm processing:

Using 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 ca no longer be clearly assigned to the last diagnostic alarm.
Data set 1 is structured as follows:

| Data set 1, diagnostics ${ }_{\text {incoming }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of missing external supply voltage <br> Bit 6... 5: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $1000_{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 08 h ) |
| 6 | Number of channels of a module (here $01{ }_{h}$ ) |
| 7 | Bit 0: Error in channel group 0 |
| 8... 15 | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message ${ }_{\text {outgoing }}$ is effected |

### 9.9 Time stamp parameterising

9.9.1 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 the edge change is saved here. Parameters can be set for the following variants by incorporating the GSD file LE010C3A.gsd:
20 bytes, 5 time stamp entries:

| Addr. | $\mathbf{+ 0}$ | $\mathbf{+ 1}$ | $\mathbf{+ 2}$ | +3 |
| :--- | :--- | :--- | :--- | :--- |
| +0 | PAE | RN | 16 -bit $\mu$ s value |  |
| +4 | PAE | RN-1 | 16 -bit $\mu$ s value |  |
| +8 | PAE | RN-2 | 16 -bit $\mu$ s value |  |
| +12 | PAE | RN-3 | 16 -bit $\mu$ s value |  |
| +16 | PAE | RN-4 | 16 -bit $\mu$ s value |  |

60 bytes, 15 time stamp entries:

| Addr. | $\mathbf{+ 0}$ | $\mathbf{+ 1}$ | $\mathbf{+ 2}$ | $+\mathbf{+ 3}$ |
| :--- | :--- | :--- | :--- | :--- |
| +0 | PAE | RN | 16-bit $\mu$ s value |  |
| +4 | PAE | RN-1 | 16 -bit $\mu$ s value |  |
| +8 | PAE | RN-2 | 16 -bit $\mu$ s value |  |
| +12 | PAE | RN-3 | 16 -bit $\mu$ s value |  |
| $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ |  |
| +56 | PAE | RN-14 | 16 -bit $\mu$ s value |  |

Running number: the "running number" (RN) is a consecutive number between 0 ... 127, which always starts afresh from 0 . The "running number" reflects the time sequence of the edges
Ticker value: After mains connection, a timer ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change in the encoder value the time value of the timer is stored as a 16 -bit $\mu$ s value together with the encoder value in the input area.

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $02{ }_{\text {h }}$ | 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. | $\begin{aligned} & 14_{h} \text { or } 3 C_{h} \\ & \text { (fix) } \end{aligned}$ |
|  | 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. | $00{ }_{\text {h }}$ (fix) |
| $01_{\text {h }}$ | 0 | Input delay DI 1 | $\begin{aligned} & 00_{\mathrm{h}}=1 \mu \mathrm{~s} \\ & 02_{\mathrm{h}}=3 \mu \mathrm{~s} \\ & 04_{\mathrm{h}}=10 \mu \mathrm{~s} \\ & 07_{\mathrm{h}}=86 \mu \mathrm{~s} \\ & 09_{\mathrm{h}}=342 \mu \mathrm{~s} \\ & 0 \mathrm{C}_{\mathrm{h}}=273 \mu \mathrm{~s} \\ & \text { Other values are not permissible. } \end{aligned}$ | 02 h |
|  | 1 | Input delay DI 2 |  | $02^{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Edge 0-1 an 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 | $00_{\text {h }}$ |
|  | 1 | Edge 1-0 at DI x | Time stamp entry on falling edge Bit 0: DI 1 (0: inhibit, 1 = enable) <br> Bit 1: DI 2 (0: inhibit, 1 = enable) <br> Bits 7 ... 2: Reserved | 00 h |

## 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 | Bits 3 ... 0: module class, $1111_{b}$ : digital module Bit 4: channel information available Bits 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bits 6 ... 0: channel type, $70_{h}$ : digital module Bit 7: more channel types available (0: yes; 1: no) |
| 5 | Number of diagnostic bits output by the module per channel (here $00{ }_{\text {h }}$ ) |
| 6 | Number of channels of a module (here 02 h ) |
| $7 . .15$ | 0 (fixed) |

### 9.9.2

## 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.

## Read data: 4 bytes

| Input area |  |  |
| :--- | :--- | :--- |
| Addr. | Access | Assignment |
| +0 | Byte | Bits $5 \ldots 0:$ running number (RN = Running Number) of the last FIFO entry <br> Bit 6: 1 (fixed) <br> Bit 7: 0 (fixed) |
| $+\mathbf{+ 1}$ | Byte | Bits $5 \ldots 0:$ running number of the next FIFO entry <br> Bit 6:1 (fixed) <br> Bit 7:1 (fixed) |
| $+\mathbf{+ 2}$ | Byte | Status |
| +3 | Byte | Number of time stamp entries in FIFO memory. |

Running number: 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 $00_{h} / 80_{h}$ : everything OK
Code $01_{h} / 81_{h}$ : no following time stamp entry
Code $02_{h} / 82_{h}$ : no new time stamp entries.
Code $03_{h} / 83_{h}$ : FIFO memory is full. No new time stamp entries can be accepted.
If bit 6 of the last processed running number (RN) was set, the code is returned at $80_{h}$ 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 | Status of outputs (PAA) |
| +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 GSD file LEO10C3A.gsd.gsd:
20 bytes, 5 time stamp entries:

| Addr. | $\boldsymbol{+ 0}$ | $\mathbf{+ 1}$ | $\mathbf{+ 2}$ |
| :--- | :--- | :--- | :--- | :--- |
| +0 | PAA | RN | +3 |
| +4 | PAA | RN-1 | -bit $\mu$ s value |
| +8 | PAA | RN-2 | 16 -bit $\mu$ s value |
| +12 | PAA | RN-3 | 16 -bit $\mu$ s value |
| +16 | PAA | RN-4 | 16 -bit $\mu$ s value |

60 bytes, 15 time stamp entries:

| Addr. | $\mathbf{+ 0}$ | $\mathbf{+ 1}$ | +2 |
| :--- | :--- | :--- | :--- | :--- |
| +0 | PAA | RN | +3 |
| +4 | PAA | RN-1 | -bit $\mu$ s value |
| +8 | PAA | RN-2 | 16 -bit $\mu$ s value |
| +12 | PAA | RN-3 | 16 -bit $\mu$ s value |
| $\ldots$ | $\ldots$ | $\ldots$ | 16 -bit $\mu$ s value |
| +56 | PAA | RN-14 | $\ldots$ |

Running number: the "running number" (RN) 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. This should be incremented with every time stamp entry.

## 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 $\mu \mathrm{s}$ at which the status of the outputs is to be accepted (value range: 0 ... 65535).

## Parameter data

| Data set <br> No. |  | Byte | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $02_{h}$ | 0 | Length - process <br> image input data | Length of input data (backplane bus communication); the <br> values are specified by the system. Other values are not <br> permissible. | $14_{h}$ or $3 C_{h}$ <br> (fix) |  |
| $\mathbf{1}$ | Length - process <br> image output data | Length of output data (backplane bus communication); <br> the values are specified by the system. Other values are <br> not permissible. | $00_{h}$ (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 | Bits 3 ... 0: module class, $1111_{\mathrm{b}}$ : digital module Bit 4: channel information available Bits 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bits 6 ... 0: channel type, $70_{h}$ : digital module Bit 7: more channel types available (0: yes; 1: no) |
| 5 | Number of diagnostic bits output by the module per channel (here $00{ }_{\text {h }}$ ) |
| 6 | Number of channels of a module (here $02{ }_{\text {h }}$ ) |
| $7 . .15$ | 0 (fixed) |

### 9.10 Parameterising technology modules

### 9.10.1 2 digital outputs with PWM functionality - EPM-S620

The following functions can be parameterised:
By integration of the GSE file VIO10C19.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 |
| +2 | PWMSTS_II | 2 | PWM 2: status |
| Status PVMx |  |  |  |
| Bit | Name | Function |  |
| 0 | - | Reserved |  |
| 1 | STS_PVM | Status PWM <br> 0: PWM output stopped <br> 1: PWM output active |  |
| 2 | STS_OUTBV | Output status <br> 0: push/pull output <br> 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 here by stating the duration of the HIGH level for the corresponding PWM channel. The pulse duration should be chosen as a factor for the 20.83 ns basis.
Value range: 48 ... 8388607 ( $1 \mu \mathrm{~s}$... approx. 175 ms )
PWMPD_I, PWMPD_II control word:here you can specify the PWM output response for the corresponding channel and start or stop PWM output.

| Control word PWMPDx |  | Function |
| :--- | :--- | :--- |
| Bit | Name | - |
| 0 | CTRL_OUTBV | Reserved <br> PWM output response <br> 0: push/pull output <br> Push/pull mode should be used if you need defined high/low levels for <br> a rapid change. This is used with a low load especially if"highside" <br> mode cannot move the output to low fast enough during a low status. <br> With push/pull, the output is switched to ground with low active and <br> to voltage with high active. |
| 1 | 1: highside output <br> In highside mode, the output switched to low remains in a state of <br> uncertainty between ground and voltage. The load has to "pull" itself <br> to ground. In highside mode, the switch is only made to high level <br> active. |  |
| $3 \ldots 7$ | - | Reserved |
| 8 | CTRL_STRT | Edge 0-1 starts PWM output on channel $x$ |

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $80_{\text {h }}$ | 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. <br> Values below $25 \mu$ s are ignored. If the pulse duration is higher or equal to the period, the DO output is set permanently. <br> Value range: 1200 ... 8388607 ( $25 \mu \mathrm{~s} . .$. approx. 175 ms ) | $1 \mathrm{~F} 40_{\text {h }}$ |
| $81_{\text {h }}$ | 0 | PWM 2: Period |  | 1 F 40 h |

## Diagnostic data

Since this module does not support alarms, the diagnostic data provides information about this module.

| Name | Byte | Function | Default |
| :---: | :---: | :---: | :---: |
| ERR_A | 1 | Reserved | $00_{\text {h }}$ |
| MODTYP | 1 | Module information <br> Byte 0 : <br> Bit 3 ... 0: module class (1111b: digital module) Bit 4: channel information available Bits 7 ... 4: 0 reserved | 15 h |
| ERR_C | 1 | Reserved | $00_{\text {h }}$ |
| ERR_D | 1 | Reserved | $00_{\text {h }}$ |
| CHTYP | 1 | Channel type <br> Byte 0: <br> Bits 6 ... 0: channel type ( 72 h : digital output) <br> Bit 7: reserved | 72 h |
| NUMBIT | 1 | Number of diagnostic bits per channel Byte 0: here $00_{h}$ | $00_{\text {h }}$ |
| NUMCH | 1 | Number of channels in module Byte 0 : here $02_{h}$ | $02_{\text {h }}$ |
| CHERR | 1 | Reserved | $00_{h}$ |
| CHOERR ... CH7ERR | 6 | Reserved | $00_{h}$ |
| DIAG_US | 4 | Value of $\mu$ s ticker when diagnostics occur Bytes 0 ... 3 | 0 |

Information on the transmission principles can be found in the appendix (■ 727).

## Parameter data

| Parameter data - ASCII protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
| 02h | 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. |  |
| $00_{h}$ | 0 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04 h : 1200 Baud $05 \mathrm{~h}: 1800$ Baud 06h: 2400 Baud $07_{\mathrm{h}}$ : 4800 Baud $08_{\mathrm{h}}$ : 7200 Baud 09 h: 9600 Baud $0 A_{h}$ : 14400 Baud $0 \mathrm{~B}_{\mathrm{h}}$ : 19200 Baud $0 C_{h}: 38400$ Baud 0Dh: 57600 Baud OE ${ }^{\text {h }}$ : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{h}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{\mathrm{h}} \text { : ASCII }$ | $01_{\text {h }}$ |
|  | 2 | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{b}$ : 6 <br> $10_{b}: 7$ <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits <br> $01_{b}$ : 1 <br> $10_{b}: 1.5$ <br> $11_{b}$ : 2 <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | 13 h |


| Parameter data - ASCII protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
|  | 3 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 . . .65535 \text { [ms] }\left(0000_{h} . . . \text { FFFF }_{h}\right)$ | $0^{0000}{ }_{\text {h }}$ |
|  | 4 | Character delay time | 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). <br> 0 ... 65535 [ms] $\left(0000_{h} . .\right.$. FFFF $\left._{h}\right)$ | $000 A_{h}$ |
|  | 5 | 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. <br> 1 ... $255\left(01_{h} . . . F_{h}\right)$ | $01_{\text {h }}$ |


| Parameter data STX/ETX protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set No. | Byte | Name | Description/value | Lenze |
| 02 h | 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. |  |
| $00_{h}$ | 0 | Diagnostics | Bits 5 ... 0 : Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| $80_{h}$ | 0 | Baud rate | $00_{h}$ : 9600 Baud $01_{h}$ : 150 Baud $02 \mathrm{~h}: 300$ Baud $03 \mathrm{~h}: 600$ Baud $04_{h}$ : 1200 Baud $05 \mathrm{~h}: 1800$ Baud $06{ }_{h}$ : 2400 Baud $07_{h}$ : 4800 Baud $08_{h}$ : 7200 Baud $09 \mathrm{~h}: 9600$ Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud $0 D_{h}: 57600$ Baud OE $\mathrm{E}_{\mathrm{h}}: 76800$ Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{\text {h }}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{h}: \text { STX/ETX }$ | 02h |



| Parameter data 3964(R) protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
| 02h | 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. |  |
|  |  | Length - process image output data | Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible. |  |
| 00 h | 0 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| $80_{h}$ | 0 | Baud rate | $00_{h}$ : 9600 Baud 01h: 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud $06_{h}$ : 2400 Baud $07_{h}$ : 4800 Baud $08_{h}$ : 7200 Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}: 19200$ Baud $0 C_{h}: 38400$ Baud 0D $\mathrm{D}_{\mathrm{h}}: 57600$ Baud $0 E_{h}: 76800$ Baud $0 F_{h}: 115200$ Baud $10_{h}$ : 109700 Baud | $00_{h}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{\mathrm{h}}: 3964 \\ & 03_{\mathrm{h}}: 3964 \mathrm{R} \end{aligned}$ | $03_{\text {h }}$ |
|  | 2 | Data format | Bit 1/0 number of data bits $00_{b}$ : 5 <br> $01_{b}$ : 6 <br> 10 : 7 <br> $11_{b}: 8$ <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{b}: 1 \\ & 10_{b}: 1.5 \\ & 11_{b}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | $13_{\text {h }}$ |
|  | 3 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} . \therefore . F_{h}\right)$ | $00_{h}$ |
|  | 4 | Character delay time | 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). <br> 0 ... 65535 [ $20-\mathrm{ms}$ steps] ( $00_{\mathrm{h}} . . . \mathrm{FF}_{\mathrm{h}}$ ) | OAh |


| Parameter data 3964(R) protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
|  | 5 | Acknowledgement time | The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. $0 . . .65535 \text { [20-ms steps] }\left(00_{h} \ldots \text { FF }_{h}\right)$ | $0 A_{h}$ |
|  | 6 | Block wait time | Maximum time between the recognition of a requested frame (DLE) and the response. $0 \ldots 255 \text { [ms] }\left(00_{h} \ldots \operatorname{FF}_{h}\right)$ | $0 A_{h}$ |
|  | 7 | STX repetitions | Maximum number of times the module attempts to establish a connection. $0 \ldots 255[\mathrm{~ms}]\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | 05h |
|  | 8 | DBL | If the block wait time has been exceeded, the number of STX repetitions for the requested protocol can be specified here. $0 \ldots 255[\mathrm{~ms}]\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | 06h |
|  | 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 the protocols 3964(R), both partners must have different priorities. <br> $00_{\mathrm{h}}$ : LOW <br> $01_{h}$ : HIGH | $00_{h}$ |

## 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 set 01h |  |  |  |
| :---: | :---: | :---: | :---: |
| Name | Byte | Function | Default |
| ERR_A | 1 | ERR_A-diagnostics <br> Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error (cable break) <br> Bit 3: Reserved <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 5, 6: Reserved <br> Bit 7: Set in the case of parameterisation error | $00_{\text {h }}$ |
| MODTYP | 1 | Module information <br> Byte 0: <br> Bit 3 ... 0: Module class (0111b: Gateway module) <br> Bit 4: channel information available <br> Bits 7 ... 4: 0 reserved | 17 h |
| ERR_C | 1 | ERR_A-diagnostics Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| ERR_D | 1 | ERR_D diagnostics <br> Bit 3 ... 0: Reserved <br> Bit 4: Set in the case of internal communication error <br> Bit 7 ... 5: Reserved | $00_{\text {h }}$ |
| CHTYP | 1 | Channel type <br> Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| NUMBIT | 1 | Number of diagnostic bits of the module per channel (here 08 h ) | 08 h |
| NUMCH | 1 | Number of channels in module Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| CHERR | 1 | Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| CHOERR ... CH7ERR | 8 | Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| DIAG_US | 4 | Value of $\mu$ s ticker when diagnostics occur Bytes 0 ... 3 | $00_{\text {h }}$ |

Information on the transmission principles can be found in the appendix (ㅁ. 727).

## Parameter data

| Parameter data - ASCII protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
| $02_{\text {h }}$ | 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. |  |
| $00_{h}$ | 0 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{h}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02h: 300 Baud $03_{\mathrm{h}}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud 06h: 2400 Baud $07_{h}$ : 4800 Baud $08_{h}$ : 7200 Baud $09_{h}$ : 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud OCh: 38400 Baud $0 \mathrm{D}_{\mathrm{h}}: 57600$ Baud 0E ${ }_{h}$ : 76800 Baud $0 F_{h}: 115200$ Baud $10_{h}$ : 109700 Baud | $00_{h}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{h}: \text { ASCII }$ | $01_{\text {h }}$ |
|  | 2 | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{\mathrm{b}}$ : 6 <br> $10_{b}$ : 7 <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{\mathrm{b}}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{\mathrm{b}}$ : Hardware $10_{b}$ : XON/XOFF | $13_{\text {h }}$ |


| Parameter data - ASCII protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
|  | 3 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 . . .65535 \text { [ms] }\left(0000_{h} . . . \text { FFFF }_{h}\right)$ | $0^{0000}{ }_{\text {h }}$ |
|  | 4 | Character delay time | 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). <br> 0 ... 65535 [ms] $\left(0000_{h} . .\right.$. FFFF $\left._{h}\right)$ | $000 A_{h}$ |
|  | 5 | 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. <br> 1 ... $255\left(01_{h} . . . F_{h}\right)$ | $01_{\text {h }}$ |
|  | 6 | Operating mode | Operating mode of the interface <br> $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $01_{h}$ |
|  | 7 | 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 ( $\square$ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{\mathrm{h}}$ : Signal R(A) 5 V (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal $R(A) 0$ V; signal $R(B) 5 V$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $01_{\text {h }}$ |


| Parameter data STX/ETX protocol <br> Data set <br> No. | Byte | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $02_{\mathrm{h}}$ | 0 | Length - process image <br> input data | Length of input data (backplane bus <br> communication); the values are specified by the <br> system. Other values are not permissible. |  |
| $\mathbf{1}$ | Length - process image <br> output data | Length of output data (backplane bus <br> communication); the values are specified by the <br> system. Other values are not permissible. |  |  |
| $00_{\mathrm{h}}$ | 0 | Diagnostics | Bits $5 \ldots .0$ Reserved <br> Bit 6: Diagnostic alarm $(0=$ inhibited; $1=$ <br> enabled) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\mathrm{h}}$ |


| Parameter data STX/ETX protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set No. | Byte | Name | Description/value | Lenze |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02 h : 300 Baud $03_{h}$ : 600 Baud 04 h : 1200 Baud 05h: 1800 Baud 06h: 2400 Baud 07 h : 4800 Baud 08h: 7200 Baud 09 h : 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud ODh: 57600 Baud 0Eh: 76800 Baud $0 F_{h}: 115200$ Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{\text {h }}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{h}: S T X / E T X$ | 02h |
|  | 2 | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{b}: 6$ <br> $10_{b}: 7$ <br> $11_{b}$ : 8 <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $01_{b}: 1$ <br> $10_{b}: 1.5$ <br> $11_{b}: 2$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | $13_{\text {h }}$ |
|  | 3 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 . . .65535 \text { [ms] }\left(0000_{\mathrm{h}} . \therefore \text { FFFF }_{\mathrm{h}}\right)$ | $0000{ }_{h}$ |
|  | 4 | TMO | TMO serves to define the maximally permissible interval between two frames. <br> $0 . . .65535$ [ms] $\left(0000_{h} . .\right.$. FFFF $\left._{h}\right)$ | 0Ah |
|  | 5 | No. of start identifiers | $00_{h}: 1$ start identifier (2. start identifier is ignored) <br> $01_{h}$ : 2 start identifiers | $01_{\text {h }}$ |
|  | 6 | Start identifier 1 | ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | 02h |
|  | 7 | Start identifier 2 | ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. | $00_{\text {h }}$ |
|  | 8 | No. of end identifier | $00_{h}$ : 1 end identifier (2. end identifier ( $0 \times 310 \mathrm{D} / \mathrm{x}$ ) is ignored) <br> $01_{h}$ : 2 end identifiers | $01_{h}$ |


| Parameter data STX/ETX protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
|  | 9 | End identifier 1 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} \ldots F_{h}\right)$ | $03_{h}$ |
|  | 10 | End identifier 2 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} \ldots F F_{h}\right)$ | $00_{h}$ |
|  | 11 | Operating mode | Operating mode of the interface $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. $01_{h}$ : 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. <br> Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $01_{\text {h }}$ |
|  | 12 | 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 (■ 228). $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal $R(A) 5 V$ (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. $02_{\mathrm{h}}$ : Signal $R(A) 0 \mathrm{~V}$; signal $R(B) 5 \mathrm{~V}$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $01_{\text {h }}$ |

Parameter data 3964(R) protocol

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| 02 h | 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. |  |
|  |  | Length - process image output data | Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible. |  |
| 00 h | 0 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |


| Parameter data 3964(R) protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{h}: 9600$ Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud 06h: 2400 Baud $07_{h}$ : 4800 Baud 08h: 7200 Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud 0Dh: 57600 Baud $0 \mathrm{E}_{\mathrm{h}}: 76800$ Baud $0 F_{h}: 115200$ Baud 10h: 109700 Baud | $00_{h}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{\mathrm{h}}: 3964 \\ & 03_{\mathrm{h}}: 3964 \mathrm{R} \end{aligned}$ | $03_{h}$ |
|  | 2 | Data format | Bit $1 / 0$ number of data bits $00_{b}: 5$ <br> $01_{b}$ : 6 <br> $10_{\mathrm{b}}: 7$ <br> $11_{\mathrm{b}}$ : 8 <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \end{aligned}$ $11_{\mathrm{b}}: 2$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{\mathrm{b}}$ : XON/XOFF | 13 h |
|  | 3 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \ldots 65535 \text { [20-ms steps] }\left(00_{h} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $00_{h}$ |
|  | 4 | Character delay time | 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). <br> 0 ... 65535 [ $20-\mathrm{ms}$ steps] ( $00_{\mathrm{h}} . . . \mathrm{FF}_{\mathrm{h}}$ ) | OAh |
|  | 5 | Acknowledgement time | The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} \ldots \text { FF }_{h}\right)$ | $0 A_{h}$ |
|  | 6 | Block wait time | Maximum time between the recognition of a requested frame (DLE) and the response. $0 \ldots 255 \text { [ms] }\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $0 A_{h}$ |
|  | 7 | STX repetitions | Maximum number of times the module attempts to establish a connection. $0 . . .255[\mathrm{~ms}]\left(00_{h} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | 05h |


| Parameter data 3964(R) protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
|  | 8 | DBL | If the block wait time has been exceeded, the number of STX repetitions for the requested protocol can be specified here. $0 \ldots 255 \text { [ms] (00 } \ldots \text {... FF } h)$ | $06_{\text {h }}$ |
|  | 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 the protocols $3964(\mathrm{R})$, both partners must have different priorities. $00_{\mathrm{h}} \text { : LOW }$ <br> $01_{h}$ : HIGH | $00_{h}$ |
|  | 10 | Operating mode | Operating mode of the interface <br> $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $0^{1}$ |
|  | 11 | 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 (■228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal R(A) 5 V (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal R(A) 0 V; signal R(B) 5 V. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $01_{h}$ |

## 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.

| Name | Byte | Function | Default |
| :---: | :---: | :---: | :---: |
| ERR_A | 1 | ERR_A-diagnostics <br> Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error (cable break) <br> Bit 3: Reserved <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 5, 6: Reserved <br> Bit 7: Set in the case of parameterisation error | $00_{\text {h }}$ |
| MODTYP | 1 | Module information <br> Byte 0: <br> Bit 3 ... 0: Module class (0111b: Gateway module) <br> Bit 4: channel information available <br> Bits 7 ... 4: 0 reserved | 17 h |
| ERR_C | 1 | ERR_A-diagnostics Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| ERR_D | 1 | ERR_D diagnostics <br> Bit 3 ... 0: Reserved <br> Bit 4: Set in the case of internal communication error <br> Bit 7 ... 5: Reserved | $00_{\text {h }}$ |
| CHTYP | 1 | Channel type Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| NUMBIT | 1 | Number of diagnostic bits of the module per channel (here 08 h) | 08 h |
| NUMCH | 1 | Number of channels in module Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| CHERR | 1 | Bit 0: set in the event of an error of channel group 1 Bits 7 ... 10 (fixed) | $00_{\text {h }}$ |
| CHOERR | 8 | Channel-specific error: channel x : <br> Bits 3 ... 0:0 (fixed) <br> Bit 4 : set in the case of open circuit (only possible for RS422) <br> Bits 7... 5: 0 (fixed) | $00_{\text {h }}$ |
| CH1ERR ... CH7ERR | 8 | Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| DIAG_US | 4 | Value of $\mu$ s ticker when diagnostics occur Bytes 0 ... 3 | $00_{\text {h }}$ |

## 10 EtherCAT communication

## 10.1

### 10.1.1

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 | $\begin{aligned} & 48 \ldots 1498 \\ & \text { bytes } \end{aligned}$ | 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 Ether-CAT frame (type)
- Datagrams

FCS

- Checksum of the EtherCAT frame


### 10.1.2 EtherCAT datagrams

EtherCAT datagrams have the following structure:

| Header | Data | WKC |
| :---: | :---: | :---: |
| 10 bits | Max. 1486 bytes | 2 bytes |

### 10.1.3 EtherCAT state machine

Each fieldbus node is led through a state machine by the master. The state changes of the bus are shown in the following illustration.


## Status

Description
Init

- No communication on the "Application layer"
- The master has access to the "DL information register".

Pre-operation • Mailbox communication on the "Application layer"
al

- No process data communication

Safe-operatio

- Mailbox communication on the "Application layer"
nal
- Process data communication (Only the inputs are evaluated; the outputs are in the "Safe" status.)

Operational

- Inputs and outputs are evaluated.

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 so-called "datagrams" via the process data channel.

- The I/O channels are controlled by means of the process data.
- The transmission of 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, the data for example are stored directly in the I/O area.
- Process data are not saved in the I/O system.
- Process data for instance are input and output data of the I/O system.


## (i) Note!

With regard to their direction, process data telegrams between the master and the EtherCAT bus coupler modules that are part of the bus are differentiated into:

- process data telegrams from EtherCAT bus coupler module (Tx data)
- process data telegrams to EtherCAT bus coupler module (Rx data)


## 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

### 10.2.1

## Access to 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 0x6000 + EtherCAT slot no. Subindexes give you access to the corresponding input data. The relevant module description contains the subindex assignment.

| Index | Subindex | Name | Type | Attr. | Default | Meaning |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 0x6000 <br> $\ldots$ | $0 \times 00$ | Input data | Unsigned8 | ro |  | Number of input data <br> subindexes for the <br> corresponding EtherCAT <br> Slot no. |

## Output data

During read-only access to the output area of an I/O compound module, addressing takes place via index 0x7000 + EtherCAT slot no. Subindexes 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 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 0x7000 <br> $\ldots$ | $0 \times 00$ | Output data | Unsigned8 | ro |  | Number of output data <br> subindexes for the <br> 0x703F |
|  |  |  |  |  | sorresponding EtherCAT <br> slot no. |  |
|  | $0 \times 01$ <br> $0 \times 02$ <br> $\ldots$ |  |  | ro |  |  |
| ro |  |  |  |  |  |  |

## 10.3

 Transmitting parameter dataNote!
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 tasks from a slave if the slave is at least in the "Pre-operational" state.

## Acyclic data transmission

Parameters ...

- are values which are stored under an index in the Lenze I/O system.
- for instance are used if the system is set once or if material is changed within a machine.
- are transmitted with a low priority.


### 10.4 General function of the parameter setting

Parameterise I/O compound modules using SDO transfer.
Addressing takes place via index 0x3100 + EtherCAT slot no. Subindexes give you access to the corresponding parameters. The relevant module description contains the subindex assignment.

| Index | Subindex | Name | Type | Attr. | Meaning |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $0 \times 3100$ <br> $\ldots$ <br> $0 \times 313 F$ | $0 \times 00$ | Parameter | Unsigned8 | ro | Number of input data <br> subindexes for the <br> corresponding EtherCAT slot <br> no. |

If the module is parameterisable, the following applies:
Index 0x3100: access to EtherCAT slot no. 1
Index 0x3101: access to EtherCAT slot no. 2

Index 0x313F: access to EtherCAT slot no. 64
The following example shows access to the parameters of the module in slot 4 via index $0 \times 3103$.

| Phy. slot | $\mathbf{1}$ | $\mathbf{2}$ | $\mathbf{3}$ | $\mathbf{4}$ |
| :--- | :---: | :---: | :---: | :---: |
| Module | DI | DI | DO | Al |
| Index | $0 \times 3100^{*}$ | $03101^{*}$ | $0 \times 3102^{*}$ | $0 \times 3103$ |
| EtherCAT slot no. | 1 | - | 2 | 3 |

* This entry is not executed because the module is not parameterisable.


### 10.5 Setting the parameters of analog I/O

10.5.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400

Parameter data

| Subindex | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- |
| 01 | Function channel 1 | $16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}}$ | $10_{\mathrm{h}}$ |
| 02 | Function channel 2 | $32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}}$ | $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 27648 \\ & \mathrm{D}=27648 * \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \text { V }}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

## Parameter data

| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 01 | Function channel 1 | $16\left(10_{\mathrm{h}}\right): 0$... $10 \mathrm{~V} / 0$... $27648_{\mathrm{dec}}$ $32\left(20_{h}\right): 0$... $10 \mathrm{~V} / 0 \ldots 16384_{\text {dec }}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $10_{h}$ |
| 02 | Function channel 2 |  | $10_{\text {h }}$ |
| 03 | Function channel 3 |  | $10^{\text {h }}$ |
| 04 | Function channel 4 |  | $10_{\text {h }}$ |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 |  |  |
|  | 5 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

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

## Parameter data

| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 01 | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... ${27648_{\text {dec }}}$ $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\text {dec }}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |
| 02 | Function channel 2 |  | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | $\begin{gathered} 10 \\ 0 \end{gathered}$ | $\begin{gathered} 13824 \\ 0 \end{gathered}$ | $\begin{aligned} & 3600 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} \mathrm{I} / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | $\begin{gathered} 10 \\ 0 \end{gathered}$ | $\begin{gathered} 8192 \\ 0 \end{gathered}$ | $\begin{aligned} & 2000 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\underset{\left(30_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 16 / 27648+4 \\ & D=27648 *(1-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | $\begin{gathered} 12 \\ 4 \end{gathered}$ | 13824 <br> 0 | 3600 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\underset{\left(40_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | $\begin{gathered} 12 \\ 4 \end{gathered}$ | $\begin{gathered} 8192 \\ 0 \end{gathered}$ | 2000 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

### 10.5.4

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

## Parameter data

| Subindex | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- |
| 01 | Function channel 1 | $48\left(30_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{d e c}$ | $31_{h}$ |
| 02 | Function channel 2 | $64\left(40_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ | $31_{\mathrm{h}}$ |
| 03 | Function channel 3 | $49\left(31_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}}$ | $35\left(41_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ |
| 04 | Function channel 4 | $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated | $31_{\mathrm{h}}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{\mathrm{h}}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 16 / 27648+4 \\ & D=27648 *(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\underset{\left(40_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

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

| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 01 | Diagnostics | Bits 0 ... 5: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 02 | Reserved | 0 |  |
| 03 | Limit value monitoring | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated $)$ <br> Bit 1: Channel 2 ( $0=$ deactivated; 1 = activated) <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |
| 04 | Interference frequency suppression | $\begin{aligned} & \text { Bit 1, } 0: \text { Channel } 1 \\ & \text { 00: Deactivated } \\ & \text { 01: } 60 \mathrm{~Hz} \\ & \text { 10: } 50 \mathrm{~Hz} \\ & \text { Bit 3, } 2: \text { Channel } 2 \\ & \text { 00: Deactivated } \\ & \text { 01: } 60 \mathrm{~Hz} \\ & \text { 10: } 50 \mathrm{~Hz} \\ & \text { Bits } 7 \ldots 4 \text { Reserved } \end{aligned}$ | $00_{\text {h }}$ |
| 05 | Function channel 1 | $\begin{aligned} & 12\left(12_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots . .27648_{\text {dec }} \\ & 34\left(22_{\mathrm{h}}\right):-10 \ldots . .10 \mathrm{~V} /-16384 \ldots . .16384_{\text {dec }} \\ & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\text {dec }} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots . .16384_{\text {dec }} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
| 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. <br> 7FFF $_{h}$ : Limit value alarm deactivated | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |
| 09 | Function channel 2 | $\begin{aligned} & 12\left(12_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots . .27648_{\mathrm{dec}} \\ & 34\left(22_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-16384 \ldots . .16384_{\mathrm{dec}} \\ & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots . .16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
| OA | Reserved | 0 |  |
| OB | 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. <br> 7FFF ${ }_{h}$ : Limit value alarm deactivated | 7FFF ${ }_{\text {h }}$ |
| OC | 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(12_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & \mathrm{D}=27648 * U / 10 \end{aligned}$ |
|  | 10 |  | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -13824 | CA00 |  |  |
|  | -10 | -27648 | 9400 |  |  |
|  | -11.76 | -32512 | 8100 | Underflow |  |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(22_{\mathrm{h}}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 | Nominal range |  |
|  | 5 | 8192 | 2000 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -8192 | E000 |  |  |
|  | -10 | -16384 | C000 |  |  |
|  | -12.5 | -20480 | B000 | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -1.76 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 | Nominal range |  |
|  | 5 | 8192 | 2000 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -2 | -3277 | F333 | Underflow |  |

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

| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 01 | Diagnostics | Bits 0 ... 5: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; 1 = enabled) <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 02 | Reserved | 0 |  |
| 03 | Limit value monitoring | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated $)$ <br> Bit 1: Channel 2 ( $0=$ deactivated; 1 = activated) <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |
| 04 | Interference frequency suppression | $\begin{gathered} \text { Bit 1, 0: Channel } 1 \\ \text { 00: Deactivated } \\ 01: 60 \mathrm{~Hz} \\ \text { 10: } 50 \mathrm{~Hz} \\ \text { Bit 3, 2: Channel } 2 \\ 00: \text { Deactivated } \\ 01: 60 \mathrm{~Hz} \\ \text { 10: } 50 \mathrm{~Hz} \\ \text { Bits } 7 \ldots 4: \text { Reserved } \end{gathered}$ | $00_{\text {h }}$ |
| 05 | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... ${27648_{\text {dec }}}$ $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0$... 16384 ${ }_{\mathrm{dec}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0$... 16384dec $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $41_{\text {h }}$ |
| 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. <br> 7FFF h : Limit value alarm deactivated | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000{ }_{\text {h }}$ |
| 09 | Function channel 2 | $\begin{aligned} & 48\left(30_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}} \\ & 64\left(40_{\mathrm{h}}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}} \\ & 49\left(31_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}} \\ & 65\left(41_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $41_{\text {h }}$ |
| OA | Reserved | 0 |  |
| OB | 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. <br> 7FFF ${ }_{h}$ : Limit value alarm deactivated | 7FFF ${ }_{\text {h }}$ |
| OC | 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000{ }_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | $\begin{gathered} 20 \\ 10 \\ 0 \end{gathered}$ | $\begin{gathered} 27648 \\ 13824 \\ 0 \end{gathered}$ | $\begin{aligned} & 6 C 00 \\ & 3600 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | -3.52 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | $\begin{gathered} 20 \\ 10 \\ 0 \end{gathered}$ | $\begin{gathered} 16384 \\ 8192 \\ 0 \end{gathered}$ | $\begin{aligned} & 4000 \\ & 2000 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | -4 | -3277 | F333 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | $\begin{gathered} 20 \\ 12 \\ 4 \end{gathered}$ | $\begin{gathered} 27648 \\ 13824 \\ 0 \end{gathered}$ | $\begin{aligned} & 6 C 00 \\ & 3600 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | 1,19 | -4864 | ED00 | Underflow |  |
| $\underset{\left(40_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | $\begin{gathered} 20 \\ 12 \\ 4 \end{gathered}$ | $\begin{gathered} 16384 \\ 8192 \\ 0 \end{gathered}$ | $\begin{aligned} & 4000 \\ & 2000 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | 0.8 | -3277 | F333 | Underflow |  |

10.5.7 2 analog outputs 0 ... 10 V ( 12 bits) - EPM-S500

## Parameter data

| Subindex | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- |
| 01 | Reserved | 0 | Bit 0: Channel $1(0$ = deactivated; 1 = activated $)$ <br> Bit 1: Channel 2 <br> Bits $2 \ldots 7:$ Reserved |
| 02 | Short-circuit <br> detection | Function channel 1 | $16\left(10_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{d e c}$ <br> $32\left(20_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\text {dec }}$ <br> $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated |
| 03 | Function channel 2 | $00_{\mathrm{h}}$ |  |
| 04 |  |  |  |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \mathrm{~V}}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | $8192$ | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

## Parameter data

| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 01 | Reserved | 0 |  |
| 02 | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 03 | Function channel 1 | $\begin{aligned} & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\text {dec }} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\text {dec }} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $10_{\text {h }}$ |
| 04 | Function channel 2 |  | $10_{\text {h }}$ |
| 05 | Function channel 3 |  | $10_{\text {h }}$ |
| 06 | Function channel 4 |  | $10_{\text {h }}$ |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

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

## Parameter data

| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 01 | Reserved | 0 |  |
| 02 | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; 1 = activated ) <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 03 | Function channel 1 | $\begin{aligned} & 48\left(30_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}} \\ & 64\left(40_{\mathrm{h}}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}} \\ & 49\left(31_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}} \\ & 65\left(41_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $31_{\text {h }}$ |
| 04 | Function channel 2 |  | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} \mathrm{I} / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | $8192$ | $2000$ | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 12 | 13824 | 3600 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{\mathrm{h}}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

## Parameter data

| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 01 | Reserved | 0 |  |
| 02 | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; 1 = activated) <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 03 | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0$... ${16384_{\text {dec }}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 . . .27648_{\mathrm{dec}}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |
| 04 | Function channel 2 |  | $31_{\text {h }}$ |
| 05 | Function channel 3 |  | $31_{\text {h }}$ |
| 06 | Function channel 4 |  | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\underset{\left(30_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{\mathrm{h}}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

### 10.6 Parameterising the temperature measurement

10.6.1 Four (two) analog inputs for resistance tests - EPM-S404

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. <br> Bits 0 ... 5: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 02 | Wire-breakage detection | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 03 | Limit value monitoring | Bit 0: Channel 1 ( $0=$ inhibited; $1=$ enabled) <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{h}$ |
| 04 | Reserved |  |  |
| 05 | Temperature system | Bit 0, 1: $00_{b}={ }^{\circ} \mathrm{C} ; 01_{\mathrm{b}}={ }^{\circ} \mathrm{F} ; 10_{\mathrm{b}}=\mathrm{K}$ Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 06 | Interference frequency suppression | Bit 0, 1: $01_{b}=60 \mathrm{~Hz} ; 10_{b}=50 \mathrm{~Hz}$ Bits 2 ... 7: Reserved | $02_{\text {h }}$ |


| Subindex <br> Channel 1 | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 07 | Function channel 1 |  | $50_{\text {h }}$ |


| 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 $0 \times 3105 / \mathrm{x}$ ) for each channel. <br> $0\left(00_{\mathrm{h}}\right)$ : At $50 \mathrm{~Hz}: 324.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $270.5 \mathrm{~ms} /$ channel 16 bits <br> 1 ( $01_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 164.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $137.2 \mathrm{~ms} /$ channel 16 bits <br> 2 (02h): At $50 \mathrm{~Hz}: 84.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 70.5 \mathrm{~ms} /$ channel 16 bits $3\left(03_{h}\right)$ : at $50 \mathrm{~Hz}: 44.1 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 37.2 \mathrm{~ms} /$ channel 16 bits 4 ( 04 h): At $50 \mathrm{~Hz}: 24.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 20.5 \mathrm{~ms} /$ channel 16 bits 5 ( 05 h ): At $50 \mathrm{~Hz}: 14.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 12.2 \mathrm{~ms} /$ channel 16 bits 6 ( $06_{h}$ ): At $50 \mathrm{~Hz}: 9.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 8.0 \mathrm{~ms} /$ channel 16 bits 7 ( $07_{\mathrm{h}}$ ): At 50 Hz : $6.6 \mathrm{~ms} /$ channel 15 bits; at $60 \mathrm{~Hz}: 5.9 \mathrm{~ms} /$ channel 15 bits 8 ( 08 h ): At $50 \mathrm{~Hz}: 4.2 \mathrm{~ms} /$ channel 13 bits; at $60 \mathrm{~Hz}: 3.8 \mathrm{~ms} /$ channel 13 bits | $00_{\text {h }}$ |
| 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 7FFFh for the upper or 8000 h 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. | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| OA | Lower limit value channel 1 |  | $8000_{\text {h }}$ |
| Channel 2 |  |  |  |
| OB | Function channel 2 | See channel 1 | $50_{\text {h }}$ |
| OC | Conversion time channel 2 | See channel 1 | $00_{\text {h }}$ |
| OD | Upper limit value channel 2 | See channel 1 | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| OE | Lower limit value channel 2 | See Channel 1 | $8000_{\text {h }}$ |
| Channel 3 (for two-wire conductor connections only) |  |  |  |
| OF | Function channel 3 | See channel 1 | $50_{\text {h }}$ |
| 10 | Conversion time channel 3 | See channel 1 | $00_{\text {h }}$ |
| 11 | Upper limit value channel 3 | See channel 1 | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| 12 | Lower limit value channel 3 |  | $8000_{h}$ |


| Subindex <br> Channel 4 | Name | Description/value |  |
| :--- | :--- | :--- | :--- |
| 13 | Function channel 4 | See channel 1 | Lenze |
| 14 | Conversion time <br> channel 4 | See channel 1 | $50_{\mathrm{h}}$ |
| 15 | Upper limit value <br> channel 4 | See channel 1 | $00_{\mathrm{h}}$ |
| 16 | Lower limit value <br> channel 4 |  | $7 \mathrm{FFF}_{\mathrm{h}}$ |

## 1 Note!

- Use parameter setting to deactivate unused inputs.
- If thermal detectors are connected in a 3 or 4 conductor setup, channels 2 and/or 4 must be deactivated.
- The module does not provide any auxiliary supply for sensors.

Measuring range

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 2-wire: PT100 } \\ \left(50_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| 2-wire: PT1000 (51h) | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 2-wire: NI100 } \\ \left(52_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 2-wire: NI1000 ( $53_{h}$ ) | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 3-wire: PT100 } \\ \left(58_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{aligned} & \text { 3-wire: PT1000 } \\ & \left(59_{h}\right) \end{aligned}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| 3-wire: NI100 (5Ah) | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 3-wire: NI1000 (5Bh) | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | -1050 dec | Underflow |
| $\begin{aligned} & \text { 4-wire: PT100 } \\ & \left(60_{h}\right) \end{aligned}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | -2430 dec | Underflow |


| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 4-wire: PT1000 } \\ \left(61_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | +10000 dec | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 4-wire: NI100 } \\ \left(62_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 4-wire: NI1000 $\left(63_{h}\right)$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | -1050 dec | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(70_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | 0 ... 32767 ${ }_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 600 \Omega \\ \left(71_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(72_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(78_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(79_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(7 A_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . .3000 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(80_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{aligned} & \text { 4-wire: } 0 \ldots . . .600 \Omega \\ & \left(81_{h}\right) \end{aligned}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | 0 ... 32767 dec | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(82_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . .3000 \Omega$ | 0 ... 32767 | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(90_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 \ldots 6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 600 \Omega \\ \left(91_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(92_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | 0 ... 30000 dec | Nominal range |
|  | - | - | Underflow |

Parameterising the temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(98_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 600 \Omega \\ \left(99_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(9 A_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(A 0_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | , | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(\mathrm{~A} 1_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{~A} 2_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . .3000 \Omega$ | $0 \ldots 30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(\mathrm{DO}_{\mathrm{h}}\right) \end{gathered}$ | $70.55 \Omega$ | 32511 ${ }_{\text {dec }}$ | Overflow |
|  | $0 . .60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots . .600 \Omega \\ \left(D 1_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{D} 2_{\mathrm{h}}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . .3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(D 8_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 600 \Omega \\ \left(D 9_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{DA}_{h}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . .3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(E O_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(E 1_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(E 2_{h}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |

10.6.2 Two 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. <br> Bits 0 ... 5: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved | 00 h |
| 02 | Wire-breakage detection | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00{ }_{h}$ |
| 03 | Limit value monitoring | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | 00 h |
| 04 | Reserved | 0 |  |
| 05 | Temperature system | Bit 0, 1: $00_{b}={ }^{\circ} \mathrm{C} ; 10_{\mathrm{b}}={ }^{\circ} \mathrm{F} ; 11_{\mathrm{b}}=\mathrm{K}$ Bits 2 ... 7: Reserved | 00 h |
| 06 | Interference frequency suppression | Bit 0, 1: $01_{\mathrm{b}}=60 \mathrm{~Hz} ; 10_{\mathrm{b}}=50 \mathrm{~Hz}$ Bits 2 ... 7: Reserved | 02 h |
| Channel 1 |  |  |  |
| 07 | Function channel 1 | External temperature compensation: <br> $176\left(60_{\mathrm{h}}\right)$ : type J, $-210.0 \ldots+1200.0^{\circ} \mathrm{C} /-2100 \ldots+12000_{\text {dec }}$ <br> $177\left(61_{\mathrm{h}}\right)$ : type K, $-270.0 \ldots+1372.0^{\circ} \mathrm{C} /-2700 \ldots+13720_{\text {dec }}$ <br> $178\left(62_{h}\right)$ : type $\mathrm{N}-270.0 \ldots+1300.0^{\circ} \mathrm{C} /-2700 \ldots+13000_{\text {dec }}$ <br> $179\left(63_{h}\right)$ : type R, $-50.0 \ldots+1769.0^{\circ} \mathrm{C} /-500 \ldots+17690_{\text {dec }}$ <br> $180\left(64_{h}\right)$ : type S, $-50.0 \ldots+1769.0^{\circ} \mathrm{C} /-500 \ldots+17690_{\text {dec }}$ <br> $181\left(65_{h}\right)$ : type T, $-270.0 \ldots+400.0^{\circ} \mathrm{C} /-2700 \ldots+4000_{\text {dec }}$ <br> $182\left(66_{h}\right)$ : type $B, 0.0 \ldots+1820.0^{\circ} \mathrm{C} / 0 \ldots+18200_{\text {dec }}$ <br> $183\left(67_{\mathrm{h}}\right)$ : type C, $0.0 \ldots+2315.0^{\circ} \mathrm{C} / 0 \ldots+23150_{\mathrm{dec}}$ <br> $184(68 \mathrm{~h})$ : type $\mathrm{E},-270.0 \ldots+1000.0^{\circ} \mathrm{C} /-2700 \ldots+10000_{\mathrm{dec}}$ <br> $185\left(69_{h}\right)$ : type L, $-200.0 \ldots+900.0^{\circ} \mathrm{C} /-2000 \ldots+9000_{\text {dec }}$ <br> Internal temperature compensation: <br> $192\left(\mathrm{CO}_{\mathrm{h}}\right)$ : type J, $-210.0 \ldots+1200.0^{\circ} \mathrm{C} /-2100 \ldots+12000_{\text {dec }}$ <br> $193\left(\mathrm{C} 1_{\mathrm{h}}\right)$ : type K, $-270.0 \ldots+1372.0^{\circ} \mathrm{C} /-2700 \ldots+13720_{\mathrm{dec}}$ <br> $194\left(\mathrm{C} 2_{\mathrm{h}}\right)$ : type $\mathrm{N}-270.0 \ldots+1300.0^{\circ} \mathrm{C} /-2700 \ldots+13000_{\text {dec }}$ <br> $195\left(\mathrm{CB}_{\mathrm{h}}\right)$ : type R, $-50.0 \ldots+1769.0^{\circ} \mathrm{C} /-500 \ldots+17690_{\mathrm{dec}}$ <br> $196\left(\mathrm{C} 4_{\mathrm{h}}\right)$ : type S, $-50.0 \ldots+1769.0^{\circ} \mathrm{C} /-500 \ldots+17690_{\mathrm{dec}}$ <br> $197\left(\mathrm{C} 5_{h}\right)$ : type T, $-270.0 \ldots+400.0^{\circ} \mathrm{C} /-2700 \ldots+4000_{\text {dec }}$ <br> 198 (C6h): type B, $0.0 \ldots+1820.0^{\circ} \mathrm{C} / 0 \ldots+18200_{\text {dec }}$ <br> $199\left(C 7_{h}\right)$ : type C, $0.0 \ldots+2315.0^{\circ} \mathrm{C} / 0 \ldots+23150_{\text {dec }}$ <br> $200(\mathrm{C} 8 \mathrm{~h})$ : type $\mathrm{E},-270.0 \ldots+1000.0^{\circ} \mathrm{C} /-2700 \ldots+10000_{\mathrm{dec}}$ <br> $201\left(C 9_{h}\right)$ : type L, $-200.0 \ldots+900.0^{\circ} \mathrm{C} /-2000 \ldots+9000_{\text {dec }}$ <br> $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $\mathrm{C} 1_{\mathrm{h}}$ |


| 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 $0 \times 3105 / \mathrm{x}$ ) for each channel. <br> 0 ( $00_{\mathrm{h}}$ ): At 50 Hz : $324.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $270.5 \mathrm{~ms} /$ channel 16 bits <br> 1 ( $01_{\mathrm{h}}$ ): at 50 Hz : $164.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $137.2 \mathrm{~ms} /$ channel 16 bits <br> 2 ( $02_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 84.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $70.5 \mathrm{~ms} /$ channel 16 bits <br> 3 ( $03_{\mathrm{h}}$ ): at $50 \mathrm{~Hz}: 44.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $37.2 \mathrm{~ms} /$ channel 16 bits <br> 4 ( $04_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 24.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $20.5 \mathrm{~ms} /$ channel 16 bits <br> $5\left(05_{h}\right)$ : at $50 \mathrm{~Hz}: 14.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $12.2 \mathrm{~ms} /$ channel 16 bits <br> $6\left(06_{h}\right)$ : at $50 \mathrm{~Hz}: 9.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $8.0 \mathrm{~ms} /$ channel 16 bits <br> $7\left(07_{\mathrm{h}}\right)$ : at 50 Hz : $6.6 \mathrm{~ms} /$ channel 15 bits; at 60 Hz : <br> $5.9 \mathrm{~ms} /$ channel 15 bits <br> $8\left(08_{\mathrm{h}}\right)$ : at $50 \mathrm{~Hz}: 4.2 \mathrm{~ms} /$ channel 13 bits; at 60 Hz : <br> $3.8 \mathrm{~ms} /$ channel 13 bits | $02_{\text {h }}$ |
| 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 7FFFh for the upper or 8000 h for the lower limit | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| OA | Lower limit value channel 1 | 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. | $8000_{h}$ |
| Channel 2 |  |  |  |
| OB | Function channel 2 | See channel 1 | $\mathrm{C} 1_{\text {h }}$ |
| OC | Conversion time channel 2 | See channel 1 | 02h |
| OD | Upper limit value channel 2 | See channel 1 | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| OE | Lower limit value channel 2 |  | $8000_{\text {h }}$ |

Measuring range

| Measuring range (Fct. no.) | Measured value |  |  | Range |
| :---: | :---: | :---: | :---: | :---: |
|  | [ ${ }^{\text {C }}$ ] | [ ${ }^{\mathrm{F}}$ ] | [K] |  |
| Type J: $-210 \ldots+1200{ }^{\circ} \mathrm{C}$ | +14500 | 26420 | 17232 | Overflow |
| $\begin{gathered} -346 \ldots 2192^{\circ} \mathrm{F} \\ 63.2 \ldots 1473.2^{\mathrm{K}} \end{gathered}$ | -2100 ... +12000 | -3460 ... +21920 | 632 ... 14732 | Nominal range |
| ( $\mathrm{BO}_{\mathrm{h}}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> $\left(\mathrm{CO}_{\mathrm{h}}\right.$ : int. comp. $0^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |
| Type K: $-210 \ldots+1372{ }^{\circ} \mathrm{C}$ | +16220 | 29516 | 18952 | Overflow |
| $\begin{gathered} -454 \ldots 2501.6^{\circ} \mathrm{F} \\ 0 \ldots 1645.2 \mathrm{~K} \end{gathered}$ | $-2700 \ldots+13720$ | -4540 ... 25016 | 0 ... 16452 | Nominal range |
| ( $\mathrm{B} 1_{\mathrm{h}}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> $\left(\mathrm{C1}_{\mathrm{h}}\right.$ : int. comp. $0^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |
| $\begin{aligned} & \text { Type } \mathrm{N}: \\ & \quad-270 \ldots+1300^{\circ} \mathrm{C} \end{aligned}$ | +15500 | 28220 | 18232 | Overflow |
| $\begin{gathered} -454 \ldots 2372^{\circ} \mathrm{F} \\ 0 \ldots 1573.2 \mathrm{~K} \end{gathered}$ | -2700 ... +13000 | -4540 ... 23720 | 0 ... 15732 | Nominal range |
| (B2h: ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C2h: int. comp. $0^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |


| Measuring range | Measured value |  |  | Range |
| :---: | :---: | :---: | :---: | :---: |
| (Fct. no.) | [ ${ }^{\circ} \mathrm{C}$ ] | [ $\left.{ }^{\circ} \mathrm{F}\right]$ | [K] |  |
| Type R: $-50 \ldots+1769^{\circ} \mathrm{C}$ | +20190 | 32766 | 22922 | Overflow |
| $\begin{gathered} -58 \ldots 3216.2^{\circ} \mathrm{F} \\ 223.2 \ldots 2042.2 \mathrm{~K} \end{gathered}$ | $-500 \ldots+17690$ | -580 ... 32162 | 2232 ... 20422 | Nominal range |
| ( $\mathrm{B}_{\mathrm{h}}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> $\left(\mathrm{C} 3_{h}\right.$ : int. comp. $0^{\circ} \mathrm{C}$ ) | -1700 | -2740 | 1032 | Underflow |
| Type S: $-50 \ldots+1769^{\circ} \mathrm{C}$ | +20190 | 32766 | 22922 | Overflow |
| $\begin{gathered} -58 \ldots 3216.2^{\circ} \mathrm{F} \\ 223.2 \ldots 2042.2 \mathrm{~K} \end{gathered}$ | $-500 \ldots+17690$ | -580 ... 32162 | 2232 ... 20422 | Nominal range |
| (B4h: ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C4h : int. comp. $0^{\circ} \mathrm{C}$ ) | -1700 | -2740 | 1032 | Underflow |
| $\begin{aligned} & \text { Type T: } \\ & -270 \ldots+440^{\circ} \mathrm{C} \end{aligned}$ | +5400 | 10040 | 8132 | Overflow |
| $\begin{array}{r} -454 \ldots . .752^{\circ} \mathrm{F} \\ 3.2 \ldots 673.2 \mathrm{~K} \end{array}$ | $-2700 \ldots+4000$ | -4540 ... 7520 | 32 ... 6732 | Nominal range |
| (B2h: ext. comp. $0^{\circ} \mathrm{C}$ ) <br> ( $\mathrm{C} 2_{\mathrm{h}}$ : int. comp. $0^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |
| Type B: $0 \ldots+1820^{\circ} \mathrm{C}$ | +20700 | 32766 | 23432 | Overflow |
| $\begin{gathered} 32 \ldots 2786.5^{\circ} \mathrm{F} \\ 273.2 \ldots 2093.2^{\mathrm{K}} \end{gathered}$ | $0 \ldots+18200$ | 320 ... 27865 | 2732 ... 20932 | Nominal range |
| (B6h: ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C6 ${ }_{h}$ : int. comp. $0{ }^{\circ} \mathrm{C}$ ) | -1200 | -1840 | 1532 | Underflow |
| Type C: $0 \ldots+2315^{\circ} \mathrm{C}$ | +25000 | 32766 | 23432 | Overflow |
| $\begin{gathered} 32 \ldots .2786 .5^{\circ} \mathrm{F} \\ 273.2 \ldots 2093.2 \mathrm{~K} \end{gathered}$ | $0 \ldots+23150$ | 320 ... 27865 | 2732 ... 20932 | Nominal range |
| (B7h: ext. comp. $0^{\circ} \mathrm{C}$ ) <br> $\left(C 7_{h}\right.$ : int. comp. $0{ }^{\circ} \mathrm{C}$ ) | $-1200$ | -1840 | 1532 | Underflow |
| Type E: $-270 \ldots+1000^{\circ} \mathrm{C}$ | +12000 | 21920 | 14732 | Overflow |
| $\begin{gathered} -454 \ldots 1832^{\circ} \mathrm{F} \\ 0 \ldots 1273.2 \mathrm{~K} \end{gathered}$ | $-2700 \ldots+10000$ | -4540 ... 18320 | 0 ... 12732 | Nominal range |
| (B8 ${ }_{h}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> ( $\mathrm{C} 8_{\mathrm{h}}$ : int. comp. $0{ }^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |
| $\begin{aligned} & \text { Type L: } \\ & -200 \ldots+900^{\circ} \mathrm{C} \end{aligned}$ | +11500 | 21020 | 14232 | Overflow |
| $\begin{gathered} -328 \ldots 1652^{\circ} \mathrm{F} \\ 73.2 \ldots 1173.2 \mathrm{~K} \end{gathered}$ | $-2000 \ldots+9000$ | -3280 ... 16520 | 732 ... 11732 | Nominal range |
| (B9 ${ }^{\text {h }}$ ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C9h: int. comp. $0^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |

### 10.7 Parameterising the counter

10.7.1 One counter 32 bits, 24 V DC - EPM-S600

The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> range. |
| Counting periodically |  | | Signal evaluation |
| :--- |
| Single rotary <br> transducer |
| Double rotary <br> transducer |
| Quadruple rotary <br> transducer |
| Connection to input "A/pulse" and "B/direction" |


| Additional functions | Description |
| :---: | :---: |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate), hardware gate (HW gate), and software gate (SW gate) is made. <br> - The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). <br> - The SW gate is controlled via your user program (status word in the output area). <br> - The HW gate is controlled via the digital gate input. <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

Further information can be found in the chapter "Product description".

## 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. <br> Bits 5 ... 0: reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 02 | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02 h |
| 03 | Input frequency track B |  | 02 h |
| 04 | Input frequency latch |  | $02_{\text {h }}$ |
| 05 | Input frequency gate |  | $02_{\text {h }}$ |
| 06 | Input frequency reset |  | $00_{\text {h }}$ |
| 07 | Reserved |  |  |
| 08 | Alarm response | Setting activates process alarm Bit 0: proc. alarm HW gate open <br> Bit 1: proc. alarm HW gate closed <br> Bit 2: proc. alarm overflow <br> Bit 3: proc. alarm underflow <br> Bit 4: proc. alarm comparison value <br> Bit 5: proc. alarm final value <br> Bit 6: proc. alarm latch value <br> Bit 7: Reserved | $80_{\text {h }}$ |
| 09 | Counter function | ```Bits 5... 0: 000000 000001 b = counting once, main counting direction forwards 000010}b= counting once, main counting direction backwards 000100 b = counting once, no main counting direction 001000b}=\mathrm{ counting periodically, main counting direction forwards 010000}b= counting periodically, main counting directio backwards 100000 Bits 7 ... 6: reserved``` | $40_{\text {h }}$ |
| OA | Comparator | ```Bit 2 ... 0: output switches (... if condition is met) \(000_{b}=\) never \(001_{b}=\) count value \(\geq\) comparison value \(010_{b}=\) count value \(\leq\) comparison value \(100_{b}=\) count value \(=\) comparison value Bit 3: invert counting direction track \(B\) \(0=\) no (do not invert) 1 = yes (invert) Bits 6 ... 4: reset \(000_{b}=\) deactivated \(001_{b}=\) HIGH level \(011_{b}=\) rising edge \(101_{b}=\) rising edge, once Bit 7: Reserved``` | $00_{h}$ |


| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| OB | Signal evaluation | Bits 2 ... 0: signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at $A$ and $B$ ) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: hardware gate (HW gate) <br> $000_{b}=$ deactivated (counter starts by setting SW gate) <br> $001_{b}=$ activated (HIGH level at gate activates the HW gate. <br> Counter starts if HW and SW gate are set.) <br> Bit 7: gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{\text {h }}$ |
| OC | Final value | Counting method: <br> 0x310B: byte 3 (high byte) <br> 0x3110: byte 2 <br> 0x3111: byte 1 <br> $0 \times 3112$ : byte 0 (low byte) | $00_{\text {h }}$ |
| OD | Loading value | Counting method: <br> 0x310B: byte 3 (high byte) <br> 0x310C: byte 2 <br> 0x310D: byte 1 <br> $0 \times 310 \mathrm{E}$ : byte 0 (low byte) | $00_{\text {h }}$ |
| OE | 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. <br> The hysteresis has an effect on the zero crossing, comparison, overflow/underflow. | $00_{\text {h }}$ |
| OF | 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. | $00_{h}$ |

### 10.7.2

Two counters 32 bits, 24 V DC - EPM-S601
The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting |
| Counting periodically | range. |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input "A/pulse" and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

Further information can be found in the chapter "Product description".

## 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. <br> Bits 5 ... 0: reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 02 | Input frequency counter 1, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02h |
| 03 | Input frequency counter 1, track B |  | 02h |
| 04 | Input frequency counter 2, track A |  | $02_{\text {h }}$ |
| 05 | Input frequency counter 2, track B |  | 02h |
| 06 | Alarm response counter 1 | Setting activates process alarm <br> Bits 1 ... 0: reserved <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bits 7 ... 6: Reserved | $00_{\text {h }}$ |
| 07 | Counter function counter 1 | ```Bit 5 ... 0: 000000 000001 b = once: forwards 000010b = once: backwards 000100b = once: no main counting direction 0010000b = periodically: forwards 010000}b=\mathrm{ periodically: backwards 100000b}=\mathrm{ periodically: no main counting direction Bits 7 ... 6: Reserved``` | $00_{\text {h }}$ |
| 08 | Comparator counter 1 | Bits 2 ... 0: Comparison bit is set (... if condition is met) <br> $000_{b}=$ never <br> $001_{b}=$ count value $\geq$ comparison value <br> $010_{b}=$ count value $\leq$ comparison value <br> $100_{b}=$ count value $=$ comparison value <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 09 | Signal evaluation counter 1 | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at $A$ and $B$ ) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: Reserved <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{\text {h }}$ |
| OA | Set value counter 1 | Counting method: <br> $0 \times 3111$ : byte 3 (high byte) <br> 0x3112: byte 2 <br> 0x3113: byte 1 <br> $0 \times 3114$ : byte 0 (low byte) | $00_{h}$ |


| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| OB | Final value counter 1 | Counting method: <br> $0 \times 310 \mathrm{D}$ : byte 3 (high byte) <br> 0x310E: byte 2 <br> 0x310F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
| OC | Loading value counter 1 | Counting method: <br> 0x3109: byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> $0 \times 310 \mathrm{C}$ : byte 0 (low byte) | $00_{\text {h }}$ |
| OD | 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. <br> The hysteresis has an effect on the zero crossing, comparison, overflow/underflow. | $00_{\text {h }}$ |
| OE | Reserved |  |  |
| OF | Alarm response counter 2 | See counter 1 | $00_{\text {h }}$ |
| 10 | Counter function counter 2 | See counter 1 | $00_{\text {h }}$ |
| 11 | Comparator counter 2 | See counter 1 | $00_{\text {h }}$ |
| 12 | Signal evaluation counter 2 | See counter 1 | $00_{\text {h }}$ |
| 13 | Set value counter 2 | See counter 1 | $00_{\text {h }}$ |
| 14 | Final value counter 2 | See counter 1 | $00_{\text {h }}$ |
| 15 | Loading value counter 2 | See counter 1 | $00_{\text {h }}$ |
| 16 | Hysteresis counter 2 | See counter 1 | $00_{\text {h }}$ |

One counter 32 bits, 5 V DC - EPM-S602
The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> range. |
| Counting periodically |  |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input "A/pulse" and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

Further information can be found in the chapter "Product description".

## 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. <br> Bits 5 ... 0: reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 02 | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02h |
| 03 | Input frequency track B |  | $02_{\text {h }}$ |
| 04 | Input frequency reset |  | 02h |
| 05 | Reserved |  |  |
| 06 | Alarm response | Setting activates process alarm <br> Bits 1 ... 0: reserved <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bits 7 ... 6: Reserved | $00_{h}$ |
| 07 | Counter function | Bit 5 ... 0: <br> $000000_{b}=$ counting continuously <br> $000001_{b}=$ once: forwards <br> $000010_{b}=$ once: backwards <br> $000100_{b}=$ once: no main counting direction <br> $001000_{b}=$ periodically: forwards <br> $010000_{b}=$ periodically: backwards <br> $100000_{b}=$ periodically: no main counting direction <br> Bits 7 ... 6: Reserved | $00_{h}$ |
| 08 | Comparator | Bits 2 ... 0: Comparison bit is set (... if condition is met) $000_{b}=\text { never }$ <br> $001_{b}=$ count value $\geq$ comparison value <br> $010_{b}=$ count value $\leq$ comparison value <br> $100_{b}=$ count value $=$ comparison value <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 6 ... 4: Reset <br> $000_{b}=$ deactivated <br> $001_{b}=$ HIGH level <br> $011_{b}=$ rising edge <br> $101_{b}=$ rising edge, once <br> Bit 7: Reserved | $00_{\text {h }}$ |


| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 09 | Signal evaluation | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: Reserved <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{\text {h }}$ |
| OA | Final value | Counting method: <br> 0x310D: byte 3 (high byte) <br> 0x310E: byte 2 <br> 0x311F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
| OB | Loading value | Counting method: <br> 0x3109: byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> $0 \times 310 \mathrm{C}$ : byte 0 (low byte) | $00_{h}$ |
| OC | Hysteresis |  | $00_{h}$ |

10.7.4 Two counters 32 bits, 24 V DC - EPM-S603

The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from 0 to the counting limit, then skips to the opposite counting limit <br> and continues to count from there. |
|  | Description |
| Signal evaluation <br> Single rotary <br> transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary <br> transducer |  |
| Quadruple rotary <br> transducer | Pulse at input "A/pulse" and direction at "B/direction" |
| Direction |  |


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

Further information can be found in the chapter "Product description".

## Parameter data

| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 01 | Input frequency counter 0, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | $02_{\text {h }}$ |
| 02 | Input frequency counter 0, track B |  | 02h |
| 03 | Input frequency counter 1, track A |  | 02h |
| 04 | Input frequency counter 1, track B |  | 02h |
| 05 | Counting direction counter 0, track B | Bits 2 ... 0: Reserved <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 06 | Signal evaluation counter 0 | ```Bits 2 ... 0: Signal evaluation 000 the counter are ignored) 001b}= rotary transducer single (at A and B) (' 010b = rotary transducer double (at A and B) 011 100}b=\mathrm{ direction (pulse at A and direction at B) Bits 7... 3: Reserved``` | $00_{\text {h }}$ |
| 07 | Counting direction counter 1, track B | Bits 2 ... 0: Reserved <br> Bit 3: Invert counting direction track B <br> 0 = no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 08 | Signal evaluation counter 1 | ```Bits 2 ... 0: Signal evaluation 000 the counter are ignored) 001b}= rotary transducer single (at A and B 010b = rotary transducer double (at A and B) 011 100b}=\mathrm{ direction (pulse at A and direction at B) Bits 7... 3: Reserved``` | $00_{\text {h }}$ |

### 10.8 Parameterising the encoder evaluation

10.8.1 SSI - EPM-S604

Further information can be found in the chapter "Product description".
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. <br> Bits 5 ... 0: reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 02 | Dead 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. $\begin{aligned} & \text { HIGH LOW } \\ & 00_{\mathrm{h}} 30_{\mathrm{h}}: 1 \mu \mathrm{~s} \\ & 00_{\mathrm{h}} 60_{\mathrm{h}}: 2 \mu \mathrm{~s} \\ & 00_{\mathrm{h}} C 0_{\mathrm{h}}: 4 \mu \mathrm{~s} \\ & 01_{\mathrm{h}} 80_{\mathrm{h}}: 8 \mu \mathrm{~s} \\ & 03_{\mathrm{h}} 00_{\mathrm{h}}: 16 \mu \mathrm{~s} \\ & 06_{\mathrm{h}} 00_{\mathrm{h}}: 32 \mu \mathrm{~s} \\ & 09_{\mathrm{h}} 00_{\mathrm{h}}: 48 \mu \mathrm{~s} \\ & 0 C_{h} 00_{\mathrm{h}}: 64 \mu \mathrm{~s} \end{aligned}$ | 0 COO h |
| 03 | Baud rate | In the "Monitoring operation" mode, the baud rate is irrelevant. Enter the baud rate here. This corresponds to the clock frequency via which the connected encoder communicates. More information on this can be found in the data sheet for your encoder. <br> HIGH LOW <br> $00_{h} 18_{h}: 2 \mathrm{MHz}$ <br> $00_{\mathrm{h}} 20_{\mathrm{h}}: 1.5 \mathrm{MHz}$ <br> $00_{\mathrm{h}} 30_{\mathrm{h}}: 1 \mathrm{MHz}$ <br> $00_{h} 60_{\mathrm{h}}: 500 \mathrm{kHz}$ <br> $00_{h} \mathrm{CO}_{\mathrm{h}}: 250 \mathrm{kHz}$ <br> $01_{\mathrm{h}} 80_{\mathrm{h}}: 125 \mathrm{kHz}$ | 0180 h |
| 04 | Reserved |  |  |
| 05 | Scaling | Depending on the encoder, further bits are transmitted in addition to the encoder value. Scaling serves to determine how many bits postpositioned 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: $00_{h} \ldots 0 F_{h}=0$ bits ... 15 bits | $00_{h}$ |


| 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. <br> $7\left(07_{h}\right)=$ " 8 bits" <br> $8(08$ h $)=" 9$ bits" <br> $9\left(09_{h}\right)=" 10$ bits" <br> $10\left(0 A_{h}\right)=$ "11 bits" <br> $11\left(0 B_{h}\right)=" 12$ bits" <br> $12\left(0 C_{h}\right)=" 13$ bits" <br> $13\left(0 D_{h}\right)=" 14$ bits" <br> $140 \mathrm{E}_{\mathrm{h}}$ ) = "15 bits" <br> $15\left(0 \mathrm{~F}_{\mathrm{h}}\right)=$ " 16 bits" <br> $16\left(10_{h}\right)=" 17$ bits" <br> $17\left(11_{h}\right)=" 18$ bits" <br> $18\left(12_{h}\right)=" 19$ bits" <br> $19\left(13_{h}\right)=$ "20 bits" <br> $20(14 \mathrm{~h})=$ "21 bits" <br> $21(15 h)=" 22$ bits" <br> $22\left(16_{h}\right)=$ " 23 bits" <br> $23\left(17_{h}\right)=$ " 24 bits" <br> $24\left(18_{h}\right)=$ "25 bits" <br> $25\left(19_{h}\right)=$ " 26 bits" <br> $26\left(1 A_{h}\right)=" 27$ bits" <br> $27\left(1 B_{h}\right)=" 28$ bits" <br> $28\left(1 C_{h}\right)=$ "29 bits" <br> $29\left(1 D_{h}\right)=" 30$ bits" <br> $30\left(1 \mathrm{E}_{\mathrm{h}}\right)=$ " 31 bits" <br> $31\left(1 F_{h}\right)=$ " 32 bits" | 18 h |


| Subindex | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 07 |  | Bit 1 ... 0: Ready for operation <br> 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. <br> In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder. <br> $01_{b}=$ monitoring operation <br> $10_{b}=$ master mode <br> Bit 2: Shifting direction <br> 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". <br> $0=$ LSB first (LSB is transmitted first) <br> $1=$ MSB first (MSB is transmitted first) <br> Bit 3: edge clock signal <br> 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. <br> 0 = falling edge <br> 1 = rising edge <br> Bit 4: Coding <br> 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. <br> $0=$ standard code <br> 1 = Gray code <br> Bits 7 ... 5: reserved | $1 E_{h}$ |
| 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. $\begin{aligned} & 0\left(00_{h}\right)=\text { inhibited } \\ & 1\left(01_{h}\right)=\text { enabled } \end{aligned}$ | $00_{\text {h }}$ |

### 10.9 Time stamp parameterising

10.9.1 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. | $\begin{aligned} & 14_{h} \text { or } 3 C_{h} \\ & \text { (fix) } \end{aligned}$ |
| 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. | $00_{\text {h }}$ (fix) |
| 03 | Input delay DI 0 | $\begin{aligned} & 00_{\mathrm{h}}=1 \mu \mathrm{~s} \\ & 02_{\mathrm{h}}=3 \mu \mathrm{~s} \\ & 04_{\mathrm{h}}=10 \mu \mathrm{~s} \\ & 07_{\mathrm{h}}=86 \mu \mathrm{~s} \\ & 09_{\mathrm{h}}=342 \mu \mathrm{~s} \\ & 0 C_{\mathrm{h}}=273 \mu \mathrm{~s} \\ & \text { Other values are not permissible. } \end{aligned}$ | 02h |
| 04 | Input delay DI 1 |  | $02_{\text {h }}$ |
| 05 | Edge 0-1 an 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 | $00_{h}$ |
| 06 | Edge 1-0 at DI x | Time stamp entry on falling edge Bit 0: DI 0 (0: inhibit, 1 = enable) <br> Bit 1: DI 1 (0: inhibit, 1 = enable) <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |

10.9.2 2 digital outputs with time stamp function - EPM-S310

Parameter data

| Subindex | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- |
| 01 | Length - process <br> image input data | Length of input data (backplane bus communication); the values <br> are specified by the system. Other values are not permissible. | $14_{h}$ or $3 C_{h}$ <br> (fix) |
| 02 | Length - process <br> image output data | Length of output data (backplane bus communication); the <br> values are specified by the system. Other values are not <br> permissible. | $00_{h}$ (fix) |

### 10.10 Parameterising technology modules

10.10.1 2 digital outputs with PWM functionality - EPM-S620

Parameter data

| Subindex | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- |
| 01 | PWM 0: period | Set parameters here for the total time for pulse duration and <br> pulse pause. The time should be selected as a factor for the <br> 20.83 ns basis. | $1 F^{\text {1F40 }} \mathrm{h}$ |
| 02 | PWM 1: period | Values below $25 \mu$ s are ignored. If the pulse duration is higher or <br> equal to the period, the DO output is set permanently. <br> Value range: $1200 \ldots 8388607(25 \mu \mathrm{~s} \ldots$ approx. 175 ms$)$ | 1F40 $_{\mathrm{h}}$ |

### 10.10.2

RS232 interface - EPM-S640
Parameter data

| Parameter data - ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| 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. |  |
| 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. |  |
| 03 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 04 | Baud rate | $00_{h}$ : 9600 Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud $05 h$ : 1800 Baud 06h: 2400 Baud $07{ }_{h}: 4800$ Baud $08_{h}: 7200$ Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud 0B ${ }_{h}$ : 19200 Baud 0Ch: 38400 Baud 0D $\mathrm{h}: 57600$ Baud $0 \mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 F_{h}: 115200$ Baud $10_{h}$ : 109700 Baud | $00_{h}$ |
| 05 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{h}: \text { ASCII }$ | $01_{\text {h }}$ |
| 06 | Data format | ```Bit 1/0 number of data bits \(00_{b}: 5\) \(01_{b}\) : 6 10 : 7 \(11_{b}: 8\) Bit 3/2 parity \(00_{b}\) : none \(01_{b}\) : odd \(10_{b}\) : even \(11_{b}\) : even Bit 5/4 number of stop bits \(01_{b}\) : 1 \(10_{\mathrm{b}}: 1.5\) \(11_{b}\) : 2 Bit 7/6 flow control \(00_{b}\) : None \(01_{b}\) : Hardware \(10_{b}\) : XON/XOFF``` | $13_{h}$ |
| 07 | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is | $00_{h}$ |
| 08 | ZNA (LOW byte) | executed. $0 . . .65535[\mathrm{~ms}]\left(0000_{h} . . . \text { FFFF }_{\mathrm{h}}\right)$ | $00_{h}$ |


| Parameter data-ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Subindex | Name | Description/value | Lenze |
| 09 | 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). $0 . . .65535[\mathrm{~ms}]\left(0000_{\mathrm{h}} . \therefore \mathrm{FFFF}_{\mathrm{h}}\right)$ | 250 ms |
| OA | Character delay time (LOW 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). $0 \ldots 65535[\mathrm{~ms}]\left(0000_{\mathrm{h}} . \therefore \text { FFFF }_{\mathrm{h}}\right)$ |  |
| OB | 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. $1 \ldots 255\left(01_{h} \ldots F_{h}\right)$ | $01_{\text {h }}$ |
| OC ... 11 | Reserved |  |  |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| 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. |  |
| 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. |  |
| 03 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 04 | Baud rate | $00_{h}$ : 9600 Baud $01_{h}: 150$ Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud $06{ }_{h}$ : 2400 Baud 07h: 4800 Baud $08_{h}$ : 7200 Baud $09_{h}$ : 9600 Baud $0 A_{h}: 14400$ Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud $0 \mathrm{D}_{\mathrm{h}}: 57600$ Baud 0E $\mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 F_{h}: 115200$ Baud $10_{h}$ : 109700 Baud | $00_{h}$ |
| 05 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{h}: \text { STX/ETX }$ | $02_{\text {h }}$ |
| 06 | Data format | Bit 1/0 number of data bits $00_{b}$ : 5 <br> $01_{b}$ : 6 <br> $10_{b}: 7$ <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits <br> $01_{b}: 1$ <br> $10_{b}: 1.5$ <br> $11_{b}$ : 2 <br> Bit 7/6 flow control <br> $00_{b}$ : None <br> $01_{b}$ : Hardware <br> 10b: XON/XOFF | $13_{h}$ |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Subindex | Name | Description/value | Lenze |
| 07 | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is executed.$0 \text {... } 65535 \text { [ms] }\left(0000_{h} . \therefore \text { FFFF }_{h}\right)$ | $0000_{\text {h }}$ |
| 08 | ZNA (LOW byte) |  | 0000h |
| 09 | TMO (HIGH byte) | TMO serves to define the maximally permissible interval between two frames. <br> 0 ... 65535 [ms] $\left(0000_{h} . .\right.$. FFFF $\left._{h}\right)$ | OAh |
| OA | TMO (LOW byte) |  | OAh |
| OB | No. of start identifiers | $00_{\mathrm{h}}$ : 1 start identifier (2. start identifier is ignored) $01_{h}$ : 2 start identifiers | $01_{\text {h }}$ |
| OC | Start identifier 1 | ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | 02h |
| OD | Start identifier 2 | ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. | $00_{\text {h }}$ |
| OE | No. of end identifier | $00_{h}$ : 1 end identifier (2. end identifier ( $0 \times 310 \mathrm{D} / \mathrm{x}$ ) is ignored) $01_{h}$ : 2 end identifiers | $01_{\text {h }}$ |
| OF | End identifier 1 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | $03_{h}$ |
| 10 | End identifier 2 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. <br> 1 ... $255\left(01_{h} . . . F_{h}\right)$ | $00_{\text {h }}$ |
| 11 | Reserved |  |  |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| 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. |  |
| 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. |  |
| 03 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| 04 | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02 h : 300 Baud $03_{\mathrm{h}}$ : 600 Baud 04h: 1200 Baud 05 h : 1800 Baud $06_{h}$ : 2400 Baud $07 \mathrm{~h}: 4800$ Baud $08_{\mathrm{h}}$ : 7200 Baud $09^{\mathrm{h}}$ : 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud 0Dh: 57600 Baud $0 \mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 F_{h}: 115200$ Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{h}$ |
| 05 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{h}: 3964 \\ & 03_{\mathrm{h}}: 3964 \mathrm{R} \end{aligned}$ | $03_{h}$ |
| 06 | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{b}$ : 6 <br> 10 : 7 <br> $11_{b}$ : 8 <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{\mathrm{b}}$ : Hardware $10_{b}$ : XON/XOFF | $13^{\text {h }}$ |
| 07 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} . \therefore \text { FF }_{h}\right)$ | $00_{h}$ |
| 08 | Character delay time | 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). <br> 0 ... 65535 [20-ms steps] $\left(00_{h} . . . \mathrm{FF}_{\mathrm{h}}\right)$ | OAh |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Subindex | Name | Description/value | Lenze |
| 09 | Acknowledgement time | The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} . \therefore F_{h}\right)$ | $0 A_{h}$ |
| 0A | Block wait time | Maximum time between the recognition of a requested frame (DLE) and the response. $0 \ldots 255[\mathrm{~ms}]\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $0 A_{h}$ |
| OB | STX repetitions | Maximum number of times the module attempts to establish a connection. $0 \ldots 255[\mathrm{~ms}]\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $05_{h}$ |
| OC | DBL | If the block wait time has been exceeded, the number of STX repetitions for the requested protocol can be specified here. $0 \ldots 255[\mathrm{~ms}]\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | 06h |
| OD | 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 the protocols 3964(R), both partners must have different priorities. <br> $00_{h}$ : LOW <br> $01_{h}$ : HIGH | $00_{\text {h }}$ |
| OE ... 11 | Reserved |  |  |

### 10.10.3 RS485 interface - EPM-S650

## Parameter data

| Parameter data-ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| 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. |  |
| 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. |  |
| 03 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 04 | Baud rate | $00_{h}$ : 9600 Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud $04_{h}$ : 1200 Baud $05 h$ : 1800 Baud $06_{h}: 2400$ Baud $07_{h}: 4800$ Baud $08_{h}: 7200$ Baud $09 \mathrm{~h}: 9600$ Baud $0 A_{h}$ : 14400 Baud 0B ${ }_{h}$ : 19200 Baud $0^{0} C_{h}: 38400$ Baud 0D $\mathrm{h}: 57600$ Baud $0 \mathrm{E}_{\mathrm{h}}: 76800$ Baud $0 F_{h}$ : 115200 Baud $10_{h}: 109700$ Baud | $00_{\text {h }}$ |
| 05 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{h}: \text { ASCII }$ | $01_{\text {h }}$ |
| 06 | Data format | ```Bit 1/0 number of data bits \(00_{b}: 5\) \(01_{\mathrm{b}}\) : 6 10 : 7 \(11_{b}: 8\) Bit 3/2 parity \(00_{b}\) : none \(01_{b}\) : odd \(10_{b}\) : even \(11_{b}\) : even Bit 5/4 number of stop bits \(01_{b}\) : 1 \(10_{\mathrm{b}}: 1.5\) \(11_{b}\) : 2 Bit 7/6 flow control \(00_{b}\) : None \(01_{b}\) : Hardware \(10_{\mathrm{b}}\) : XON/XOFF``` | $13_{h}$ |
| 07 | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is | $00_{\text {h }}$ |
| 08 | ZNA (LOW byte) | executed. $0 . . .65535 \text { [ms] }\left(0000_{\mathrm{h}} . . . \text { FFFF }_{\mathrm{h}}\right)$ | $00_{\text {h }}$ |


| Parameter data - ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Subindex | Name | Description/value | Lenze |
| 09 | 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). $0 \ldots 65535[\mathrm{~ms}]\left(0000_{\mathrm{h}} . \therefore \mathrm{FFFF}_{\mathrm{h}}\right)$ | 250 ms |
| OA | Character delay time (LOW 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). $0 \ldots 65535[\mathrm{~ms}]\left(0000_{\mathrm{h}} . \therefore \mathrm{FFFF}_{\mathrm{h}}\right)$ |  |
| OB | 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. $1 \ldots 255\left(01_{h} \ldots F_{h}\right)$ | $01_{\text {h }}$ |
| OC ... 11 | Reserved |  |  |
| 12 | Operating mode | Operating mode of the interface $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $00_{\text {h }}$ |
| 13 | 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 ( $\mathbb{C}$ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal R(A) 5 V (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal $R(A) 0 \mathrm{~V}$; signal $R(B) 5 \mathrm{~V}$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $00_{\text {h }}$ |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| 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. |  |
| 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. |  |
| 03 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 04 | Baud rate | $00_{h}$ : 9600 Baud $01_{h}: 150$ Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud $06{ }_{h}$ : 2400 Baud 07h: 4800 Baud $08_{h}$ : 7200 Baud $09_{h}$ : 9600 Baud $0 A_{h}: 14400$ Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud $0 \mathrm{D}_{\mathrm{h}}: 57600$ Baud 0E $\mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 F_{h}: 115200$ Baud $10_{h}$ : 109700 Baud | $00_{h}$ |
| 05 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{h}: \text { STX/ETX }$ | $02_{\text {h }}$ |
| 06 | Data format | Bit 1/0 number of data bits $00_{b}$ : 5 <br> $01_{b}$ : 6 <br> $10_{b}: 7$ <br> $11_{b}: 8$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits <br> $01_{b}: 1$ <br> $10_{b}: 1.5$ <br> $11_{b}$ : 2 <br> Bit 7/6 flow control <br> $00_{b}$ : None <br> $01_{b}$ : Hardware <br> 10b: XON/XOFF | $13_{h}$ |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Subindex | Name | Description/value | Lenze |
| 07 | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is executed. <br> 0 ... 65535 [ms] $\left(0000_{h} . .\right.$. FFFF $\left._{h}\right)$ | $0000_{\text {h }}$ |
| 08 | ZNA (LOW byte) |  | $0000{ }_{\text {h }}$ |
| 09 | TMO (HIGH byte) | TMO serves to define the maximally permissible interval between two frames. <br> 0 ... 65535 [ms] $\left(0000_{h} . .\right.$. FFFF $\left._{h}\right)$ | OAh |
| OA | TMO (LOW byte) |  | OAh |
| OB | No. of start identifiers | $00_{h}$ : 1 start identifier (2. start identifier is ignored) $01_{h}: 2$ start identifiers | $01_{\text {h }}$ |
| OC | Start identifier 1 | ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. $1 . . .255\left(01_{h} \ldots F_{h}\right)$ | 02h |
| OD | Start identifier 2 | ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. | $00_{\text {h }}$ |
| OE | No. of end identifier | $00_{h}$ : 1 end identifier (2. end identifier ( $0 \times 310 \mathrm{D} / \mathrm{x}$ ) is ignored) <br> $01_{h}$ : 2 end identifiers | $01_{\text {h }}$ |
| OF | End identifier 1 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | $03_{\text {h }}$ |
| 10 | End identifier 2 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} \ldots F F_{h}\right)$ | $00_{\text {h }}$ |
| 11 | Reserved |  |  |
| 12 | Operating mode | Operating mode of the interface $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $00_{h}$ |
| 13 | 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 ( $\amalg$ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal $R(A) 5 V$ (open circuit monitoring); signal $R(B) 0$ V. In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal $R(A) 0 \mathrm{~V}$; signal $R(B) 5 \mathrm{~V}$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $00_{\text {h }}$ |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| 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. |  |
| 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. |  |
| 03 | Diagnostics | Bits 5 ... 0 : Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| 04 | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02 h : 300 Baud $03_{\mathrm{h}}$ : 600 Baud 04h: 1200 Baud 05 h : 1800 Baud 06h: 2400 Baud $07_{\mathrm{h}}$ : 4800 Baud 08 h : 7200 Baud $09_{\mathrm{h}}$ : 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud OD h : 57600 Baud OE h : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{h}$ |
| 05 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{\mathrm{h}}: 3964 \\ & 03_{\mathrm{h}}: 3964 \mathrm{R} \end{aligned}$ | $03_{h}$ |
| 06 | Data format | Bit 1/0 number of data bits $00_{b}$ : 5 <br> $01_{b}: 6$ <br> 10 b: 7 <br> 11 b : 8 <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | $13_{h}$ |
| 07 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. <br> 0 ... 65535 [ $20-\mathrm{ms}$ steps] $\left(00_{\mathrm{h}} . . . \mathrm{FF}_{\mathrm{h}}\right.$ ) | $00_{\text {h }}$ |
| 08 | Character delay time | 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). <br> 0 ... 65535 [20-ms steps] ( $00_{h} . . . \mathrm{FF}_{\mathrm{h}}$ ) | OAh |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Subindex | Name | Description/value | Lenze |
| 09 | Acknowledgement time | The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} . \therefore F F_{h}\right)$ | $0 \mathrm{~A}_{\mathrm{h}}$ |
| OA | Block wait time | Maximum time between the recognition of a requested frame (DLE) and the response. $0 \ldots 255 \text { [ms] }\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $0 A_{h}$ |
| OB | STX repetitions | Maximum number of times the module attempts to establish a connection. <br> $0 . . .255$ [ms] $\left(00_{h} . . . F_{h}\right)$ | $05_{\text {h }}$ |
| OC | DBL | If the block wait time has been exceeded, the number of STX repetitions for the requested protocol can be specified here. $0 \ldots 255 \text { [ms] }\left(00_{h} \ldots F_{h}\right)$ | 06h |
| OD | 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 the protocols $3964(R)$, both partners must have different priorities. <br> $00_{h}$ : LOW <br> $01_{h}$ : HIGH | $00_{h}$ |
| OE ... 11 | Reserved |  |  |
| 12 | Operating mode | Operating mode of the interface <br> $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $00_{h}$ |
| 13 | 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 ( $\mathbb{\square}$ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{\mathrm{h}}$ : Signal R(A) 5 V (open circuit monitoring); signal R(B) 0 V . In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{\text {h }}$ Signal $R(A) 0$ V; signal $R(B) 5 V$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $00_{h}$ |

### 10.11 Monitoring

In case of error, the digital and analog outputs are switched to the state FALSE or 0 V . Exception: digital outputs with time stamp functionality retain the last set value.

### 10.12 Diagnostics

### 10.12.1 SDO error codes

Via the SDO service and the object directory, you can 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 | Error |
| :--- | :--- |
| $0 \times 05030000$ | Toggle bit not alternated |
| $0 \times 05040000$ | SDO protocol timed out |
| $0 \times 05040001$ | Client/server command specifier not valid or unknown |
| $0 \times 05040002$ | Invalid block size (block mode only) |
| $0 \times 05040003$ | Invalid sequence number (block mode only) |
| $0 \times 05040004$ | CRC error (block mode only) |
| $0 \times 05040005$ | Out of memory |
| $0 \times 06010000$ | Unsupported access to an object |
| $0 \times 06010001$ | Attempt to read a write only object |
| $0 \times 06010002$ | Attempt to write a read only object |
| $0 \times 06020000$ | Object does not exist in the object dictionary |
| $0 \times 06040041$ | Object cannot be mapped to the PDO |
| $0 \times 06040042$ | The number and length of the objects to be mapped would exceed PDO length |
| $0 \times 06040043$ | General parameter incompatibility reason |
| $0 \times 06040047$ | General internal incompatibility in the device |
| $0 \times 06060000$ | Access failed due to an hardware error |
| $0 \times 06070010$ | Data type does not match, length of service parameter does not match |
| $0 \times 06070012$ | Data type does not match, length of service parameter too high |
| $0 \times 06070013$ | Data type does not match, length of service parameter too low |
| $0 \times 06090011$ | Sub-index does not exist |
| $0 \times 06090030$ | Value range of parameter exceeded (only for write access) |
| $0 \times 06090031$ | Value of parameter written too high |
| $0 \times 06090032$ | Value of parameter written too low |
| $0 \times 06090036$ | Maximum value is less than minimum value |
| $0 \times 08000000$ | General error |
| $0 \times 08000020$ | Data cannot be transferred or stored to the application |
| $0 \times 08000021$ | Data cannot be transferred or stored to the application because of local control |
| $0 \times 08000022$ | Data cannot be transferred or stored to the application because of the present device state |
| $0 \times 08000023$ | Object directory dynamic generation fails or no object directory is present (e.g. object <br> directory is generated from file and generation fails because of an file error) |

### 10.12.2

## 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 |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| 0xF100 | $0 \times 00$ | Interrupt status | Unsigned8 | ro | 2 |  |
|  | 0x01 | Hardware interrupt <br> counter | Unsigned32 | ro | $0 \times 00000000$ | Counter for process <br> alarm |
|  | $0 \times 02$ | Diagnostic interrupt <br> counter | Unsigned32 | ro | $0 \times 00000000$ | Counter for <br> 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 $0 \times 06$ 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 $0 \times 06$ using the index assigned.
The following index assignment applies:

- Writing to subindex $0 \times 06$ of index $0 \times 5000$ : resets counter for process alarms
- Writing to subindex $0 \times 06$ of index $0 \times 5002$ : resets counter for diagnostic alarms


## Process alarm data

If the alarm status indicates a process alarm, you have access to the current process alarm data via index $0 \times 5000$.

| Index | Subindex | Name | Type | Attr. | Default | Meaning |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 0x5000 | 0x00 | Hardware interrupt data | Unsigned8 | ro | 0x00 | Current process alarm data |
|  | $0 \times 01$ | Slot number | Unsigned8 | ro | $0 \times 00$ | EtherCAT slot no. of module on which the alarm has occurred |
|  | 0x02 | Diagnostic byte 1 | Unsigned8 | ro | $0 \times 00$ | Process alarm data (see tables below) |
|  | $0 \times 03$ | Diagnostic byte 2 | Unsigned8 | ro | $0 \times 00$ |  |
|  | $0 \times 04$ | Diagnostic byte 3 | Unsigned8 | ro | $0 \times 00$ |  |
|  | $0 \times 05$ | Diagnostic byte 4 | Unsigned8 | ro | 0x00 |  |
|  | 0x06 | Acknowledge | Unsigned8 | rw | $0 \times 00$ | 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: Limit value exceeded channel 1 <br> Bit 1: Limit value exceeded channel 2 <br> 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 |
| EPM-S405-process alarm |  |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0: Limit exceedance channel 1 <br> Bit 1: Limit exceedance channel 2 <br> Bit 2: Limit exceedance channel 3 <br> Bit 3: Limit exceedance channel 4 Bit 7 ... 4: 0 (fix) |
| 2 | Bit 0: Limit value underflow channel 1 <br> Bit 1: Limit value underflow channel 2 <br> Bit 2: Limit value underflow channel 3 <br> Bit 3: Limit value underflow channel 4 <br> 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: Limit value exceeded channel 1 <br> Bit 1: Limit value exceeded channel 2 Bit 7 ... 2: 0 (fixed) |
| 2 | Bit 0: Limit value not reached, channel 1 <br> Bit 1: Limit value not reached, channel 2 <br> Bit 7 ... 2: 0 (fixed) |
| 3/4 | $\mu s$-ticker value at the time of the alarm <br> The I/O compound module features an integrated 32-bit timer ( $\mu$ s-ticker) which is started at switch-on and starts at 0 again after $23^{2}-1 \mu \mathrm{~s}$. These two bytes represent the lower 2 bytes of the $\mu \mathrm{s}$-ticker ( $0 . .2^{16}-1$ ) |
| EPM-S600 - process alarm |  |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7... 5: 0 (fixed) |
| 2 | State of the inputs at the time of the alarm <br> Bit 0: A/pulse <br> Bit 1: $B /$ direction <br> Bit 2: Latch <br> Bit 3: Hardware gate <br> Bit 4: Reset <br> Bit 7... 5: 0 (fixed) |
| 3/4 | Ticker value at the time of the alarm |
| EPM-S601 and EPM-S602-process alarm |  |
| Diagnostic byte | Bit 7 ... 0 |
| 8 | Bit 0: 0 <br> Bit 1: 0 <br> Bit 2: Counter 0, overflow/underflow/final value <br> Bit 3: Counter 0, comparison value reached <br> Bit 4: 0 <br> Bit 5: 0 <br> Bit 6: Counter 1, overflow/underflow/final value <br> Bit 7: Counter 1, comparison value reached |
| 9 | State of the inputs at the time of the alarm <br> Bit 0: Counter 0, A/pulse <br> Bit 1: Counter 0, $\mathrm{B} /$ direction <br> Bit 2: Counter 1, $A$ /pulse <br> Bit 3: Counter 1, B/direction <br> Bit 7 ... 4: 0 (fixed) |
| $10 . . .11$ | 16 bit $\mu$ s value at the time of the alarm |

## Diagnostic data (bytes 1 ... 4)

If the alarm status indicates a diagnostic alarm, index $0 \times 5002$ 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 |
|  | $0 \times 01$ | 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) |
|  | $0 \times 03$ | Diagnostic byte 2 | Unsigned8 | ro | 0x00 |  |
|  | 0x04 | Diagnostic byte 3 | Unsigned8 | ro | 0x00 |  |
|  | $0 \times 05$ | 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 |


| Diagnostic byte | Bits 7 ... 0 |
| :---: | :---: |
| 1 | Bit 0: set if module fault <br> Bit 1: set if internal error <br> Bit 2: set if external error <br> Bit 3: set if channel error available <br> Bit 4: set if there is no external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: set if parameterisation error |
| 2 | Bits 3 ... 0: module class, $0101_{\mathrm{b}}$ : analog module Bit 4: set if channel information available Bits 7 ... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bits 3... 0:0 (fixed) <br> Bit 4: set if internal communication error <br> Bit 5: reserved <br> Bit 6: set if process alarm lost <br> Bit 7: 0 (fixed) |


| EPM-S600-diagnostic alarm |  |
| :---: | :---: |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0 : Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6... 4:0 (fixed) <br> Bit 7: Parameterisation error |
| 2 | Bit 3 ... 0: Module class, $1000_{b}$ : function module Bit 4: Channel information available <br> Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| EPM-S601-diagnostic alarm |  |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | Bit 0 : Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6... 4:0 (fixed) <br> Bit 7: Parameterisation error |
| 2 | Bit 3 ... 0: Module class, $10000_{b}$ : function module Bit 4: Channel information available <br> Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |

## EPM-S602 diagnostic alarm

| Diagnostic byte | Bits 7 ... 0 |
| :---: | :---: |
| 1 | Bit 0 : set if module fault <br> Bit 1: set if internal error <br> Bit 2: set if external error <br> Bit 3: set if channel error exists <br> Bit 4: set if there is no external voltage supply <br> Bits 7.... 5: 0 (fixed) |
| 2 | Bits 3 ... 0: module class, $1000^{b}$ : function module Bit 4: set if channel information available Bits 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bits 5 ... 0: 0 (fixed) <br> Bit 6: set if process alarm lost <br> Bit 7: 0 (fixed) |

## EPM-S603 - diagnostic alarm

| Diagnostic byte | Bit $\mathbf{7} \ldots \mathbf{0}$ |
| :---: | :--- |
| 1 | 0 (fixed) |
| 2 | Bit $3 \ldots 0:$ Module class, $1000_{\mathrm{b}}$ : function module <br> Bit 4: Channel information available <br> Bit 7 ... 5:0 (fixed) |
| 3 | 0 (fixed) |
| 4 | 0 (fixed) |


| Diagnostic byte | Bit 7 ... 0 |
| :---: | :---: |
| 1 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of missing external supply voltage <br> Bit 6... 5: 0 (fixed) <br> Bit 7: Parameterisation error |
| 2 | Bit 3 ... 0: Module class, $1000_{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 5 ... 0:0 (fixed) 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 0 | Diagnostic alarm data (see module description) |
|  | 0x03 |  | Unsigned8 | ro | 0 |  |
|  | 0x04 |  | Unsigned8 | ro | 0 |  |
|  | 0x05 |  | Unsigned8 | ro | 0 |  |
|  | 0x06 |  | Unsigned8 | ro | 0 |  |
|  | $0 \times 07$ |  | 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 |  |
|  | 0×10 |  | Unsigned8 | ro | 0 |  |
|  | $0 \times 11$ |  | Unsigned8 | ro | 0 |  |
|  | 0x12 |  | Unsigned32 | ro | 0 |  |


| Diagnostic byte | Bits 7 ... 0 |
| :---: | :---: |
| 1 | Bit 0 : set if module fault <br> Bit 1: set if internal error <br> Bit 2: set if external error <br> Bit 3: set if channel error available <br> Bit 4: set if there is no external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: set if parameterisation error |
| 2 | Bits 3 ... 0: module class, $0101_{b}$ : analog module Bit 4: set if channel information available Bits 7... 5:0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bits 3... 0:0 (fixed) <br> Bit 4: set if internal communication error <br> Bit 5: reserved <br> Bit 6: set if process alarm lost <br> Bit 7: 0 (fixed) |
| 5 | Bits 6 ... 0: channel type, $71_{\text {h }}$ : analog input Bit 7: 0 (fixed) |
| 6 | Number of diagnostic bits output by the module per channel (here 08h) |
| 7 | Number of channels of a module (here 04h:) |
| 8 | Bit 0: channel error, channel group 0 Bit 1: channel error, group 1 Bits 7... 2: 0 (fixed) |
| 9 | Channel-specific error: channel 0: <br> Bit 0: set if project planning/parameterising error <br> Bits 3 ... 1: 0 (fixed) <br> Bit 4: set if open circuit <br> Bit 5: set if process alarm lost <br> Bit 6: set if measuring range not reached <br> Bit 7: set if measuring range exceeded |
| 10 | Channel-specific error: channel 1: <br> Bit 0: set if project planning/parameterising error <br> Bits 3 ... 1: 0 (fixed) <br> Bit 4: set if open circuit <br> Bit 5: set if process alarm lost <br> Bit 6: set if measuring range not reached <br> Bit 7: set if measuring range exceeded |
| $11 . . .16$ | 0 (fixed) |


| Diagnostic byte | Bit 7 ... 0 |
| :---: | :---: |
| 1 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 2 | Bit 3 ... 0: Module class, $1000^{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 5 | Bits 6 ... 0: channel type, 76h: counter module Bit 7: further channel types available, 0: no, 1: yes |
| 6 | Number of diagnostic bits output by the module per channel (here 08 h ) |
| 7 | Number of channels of a module (here $01_{h}$ ) |
| 8 | Bit 0: error in channel group 0 (counter) Bits 7 ... 1: 0 (fixed) |
| 9 | Diagnostic alarm due to process alarm lost to ... <br> Bit 0: hardware gate open <br> Bit 1: hardware gate closed <br> Bit 2: overflow/underflow/final value <br> Bit 3: comparison value reached <br> Bit 4: latch value <br> Bits 7 ... 5: 0 (fixed) |
| $10 . . .16$ | 0 (fixed) |


| Diagnostic byte | Bit 7 ... 0 |
| :---: | :---: |
| 1 | Bit 0 : Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6... 4:0 (fixed) <br> Bit 7: Parameterisation error |
| 2 | Bit 3 ... 0: Module class, $1000_{\mathrm{b}}$ : function module <br> Bit 4: Channel information available <br> Bit 7 ... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 5 | Bits 6 ... 0: channel type, $76_{h}$ : counter module Bit 7: further channel types available, $0:$ no, 1 : yes |
| 6 | Number of diagnostic bits output by the module per channel (here 08h) |
| 7 | Number of channels of a module (here $02_{\text {h }}$ ) |
| 8 | Bit 0: error in channel group 0 (counter 0) Bit 1: error in channel group 1 (counter 1) Bits 7... 2: 0 (fixed) |
| 9 | Channel group 0: diagnostic alarm due to lost process alarm to ... <br> Bits 1 ... 0:0 (fixed) <br> Bit 2: overflow/underflow/final value <br> Bit 3: comparison value reached <br> Bits 7... 4:0 (fixed) |
| 10 | Channel group 1: diagnostic alarm due to lost process alarm to ... <br> Bits 1... 0:0 (fixed) <br> Bit 2: overflow/underflow/final value <br> Bit 3: comparison value reached <br> Bits 7 ... 4:0 (fixed) |
| $11 . .16$ | 0 (fixed) |


| Diagnostic byte | Bits 7 ... 0 |
| :---: | :---: |
| 1 | Bit 0 : set if module fault <br> Bit 1: set if internal error <br> Bit 2: set if external error <br> Bit 3: set if channel error exists <br> Bit 4: set if there is no external voltage supply <br> Bits 7 .... 5: 0 (fixed) |
| 2 | Bits 3 ... 0: module class, $1000^{b}$ : function module Bit 4: set if channel information available Bits 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bits 5 ... 0: 0 (fixed) <br> Bit 6: set if process alarm lost <br> Bit 7: 0 (fixed) |
| 5 | Bits 6 ... 0 : channel type, $76_{h}$ : counter module Bit 7: further channel types available, $0:$ no, 1 : yes |
| 6 | Number of diagnostic bits output by the module per channel (here $08_{h}$ ) |
| 7 | Number of channels of a module (here 02 h ) |
| 8 | Bit 0: error in channel group 0 (counter 0) Bit 1: error in channel group 1 (counter 1) Bits 7 ... 2: 0 (fixed) |
| 9 | Channel group 0: diagnostic alarm due to lost process alarm to ... <br> Bits 1... 0:0 (fixed) <br> Bit 2: overflow/underflow/final value <br> Bit 3: comparison value reached <br> Bits 7 ... 4:0 (fixed) |
| 10 | Channel group 1: diagnostic alarm due to lost process alarm to ... <br> Bits 1... 0:0 (fixed) <br> Bit 2: overflow/underflow/final value <br> Bit 3: comparison value reached <br> Bits 7... 4:0 (fixed) |
| $11 . . .16$ | 0 (fixed) |
| EPM-S603 - diagnostic alarm |  |
| Diagnostic byte | Bit 7 ... 0 |
| 1 | 0 (fixed) |
| 2 | Bit 3 ... 0: Module class, 1000 : function module <br> Bit 4: Channel information available <br> Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | 0 (fixed) |
| 5 | Bits 6 ... 0: channel type, 76h: counter module Bit 7: further channel types available, $0:$ no, 1 : yes |
| 6 | Number of diagnostic bits output by the module per channel (here $00_{h}$ ) |
| 7 | Number of channels of a module (here $022_{\text {h }}$ ) |
| $8 . .16$ | 0 (fixed) |


| Diagnostic byte | Bit 7 ... 0 |
| :---: | :---: |
| 1 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of missing external supply voltage <br> Bit 6... 5: 0 (fixed) <br> Bit 7: Parameterisation error |
| 2 | Bit 3 ... 0: Module class, $1000^{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 5 | Bits 6 ... 0: channel type, 76h: counter module Bits 7: 0 (fixed) |
| 6 | Number of diagnostic bits output by the module per channel (here 08 h ) |
| 7 | Number of channels of a module (here $01_{h}$ ) |
| 8 | Bit 0: error in channel group 0 |
| $9 . .16$ | 0 (fixed) |

### 10.12.3 Standard objects

The following indexes can be used for diagnostic purposes. They show operating states. Settings cannot be made.

| Index | Designation | Possible settings |  | IMPORTANT |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |
| $0 \times 1000{ }_{\text {h }}$ | Device type |  |  | Device type of the module Read only |  |
| $0 \times 1003 \mathrm{~h}$ | Last error |  |  | Current error, the fault memory is cleared by a reset or power cycle. <br> Read only |  |
| 0 | Last error |  |  |  |  |
| 1 | Error code |  |  | Bus status |  |
| 2 | Module number |  |  | Module slot |  |
| 3 | Error description |  |  | Error code |  |
| $0 \times 1008$ h | Device name |  |  | Device name of the module Read only |  |
| 0x1009h | Hardware Version |  |  | Hardware version of the module Read only |  |
| $0 \times 100 A_{h}$ | Software version |  |  | Software version of the module Read only |  |
| $0 \times 100 B_{h}$ | System version |  |  | Delivery status, depending on FPGA versions of bus coupler and modules, at least system version 2 Read only |  |
| 0x1018h | 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 |  |  |  |  |
| $\begin{aligned} & 0 \times 1600_{h} \\ & \ldots \\ & 0 \times 163 F \end{aligned}$ | 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 |  |  |  |  |
| ... | ... |  |  |  |  |



| Index | Designation | Possible settings |  | IMPORTANT |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |
| 0x1C32h | 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 |  |  |  |  |
| $0 \times 1 \mathrm{C} 33_{\text {h }}$ | 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 |  |  |  |  |
| 0x3000h | Parameter EtherCAT coupler |  |  |  |  |
| 0 | Coupler parameter |  |  |  |  |
| 1 | Auto-acknowl edge |  |  | States how the EtherCAT coupler is to respond to alarms. <br> - 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. <br> - 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 | Designation | Possible settings |  | IMPORTANT |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |
| $\begin{aligned} & 0 \times 3100_{h} \\ & \ldots \\ & 0 \times 313 F_{h} \end{aligned}$ | Parameter EtherCAT coupler |  |  | This object provides access to the parameters of an I/O compound module. The EtherCAT slot is addressed via the index. Subindexes provide access to the corresponding parameter. The respective module description contains the subindex assignment. <br> 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 |  |  |  |  |
| ... | ... |  |  |  |  |
| 0x4000 ${ }_{\text {h }}$ | Clear IO counter |  |  | Writing any value to the corresponding subindex clears |  |
| 0 | Clear master counter |  |  | the counter |  |
| 1 | Clear module counter |  |  |  |  |
| 0x4001h |  |  |  |  |  |
| 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 |  |  |  |  |
| 0x4002h |  |  |  |  |  |
| 0 | Module MDL counter |  |  |  |  |
| 1 | Slot 1 |  |  |  |  |
| 2 | Slot 2 |  |  |  |  |
| ... | ... |  |  |  |  |
| 40 | Slot 64 |  |  |  |  |

$\left.\begin{array}{r|l|l|l|l|l|l}\hline \text { Index } & \text { Designation } & \text { Possible settings } \\ \text { Lenze } & \text { Selection }\end{array}\right)$

| Index | Designation | Possible settings |  | IMPORTANT |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | Lenze | Selection |  |  |
| $0 \times 5000{ }_{h}$ |  |  |  | If object 0xF100 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 <br> 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 $0 \times 5000$. |  |

### 10.12.4 Storage of the PDOs

| Index | Description |
| :--- | :--- |
| $0 \times 6000$ | Input PDO (1. slot) |
| $0 \times 6001$ | Input PDO (2. slot) |
| $\ldots$ | ... |
| $0 x 601 \mathrm{~F}$ | Input PDO (32. slot) |
|  |  |
| $0 x 7000$ | Output PDO (1. slot) |
| $0 \times 7001$ | Output PDO (1. slot) |
| $\ldots$ | $\ldots$ |
| $0 x 701 \mathrm{~F}$ | Output PDO (32. slot) |

### 10.12.5 Emergency telegram

Emergency messages are triggered by device-internal mechanisms and are reported to the master via the EtherCAT mailbox service.

If a state change cannot be carried out for an EtherCAT gateway, this is reported by a corresponding emergency message.
An emergency message is structured as follows:


EtherCAT-specific error codes
$\left.\begin{array}{l|l|l|l}\hline & \text { Description } & \text { Current State } & \text { Resulting State } \\ \hline \text { Code } & \text { No error } & \text { Any } & \text { Current state } \\ \hline 0 \times 0000 & \text { Unspecified error } & \text { Any } & \text { Any + E } \\ \hline 0 \times 0001 & \text { No Memory } & \text { Any } & \text { Any + E } \\ \hline 0 \times 0011 & \begin{array}{l}\text { Invalid requested state } \\ \text { change }\end{array} & \begin{array}{l}\mathrm{I} \rightarrow \mathrm{S}, \mathrm{I} \rightarrow \mathrm{O}, \\ \mathrm{P} \rightarrow \mathrm{O}, \mathrm{O} \rightarrow \mathrm{B},\end{array} \\ \mathrm{S} \rightarrow \mathrm{B}, \mathrm{P} \rightarrow \mathrm{B}\end{array}\right)$

| Code | Description | Current State | Resulting State |
| :---: | :---: | :---: | :---: |
| 0x0024 | Invalid Input Mapping | $P \rightarrow S$ | 0x0019 |
| 0x0025 | Invalid Output Mapping | $P \rightarrow S$ | P + E |
| 0x0026 | Inconsistent Settings | $P \rightarrow S$ | $P+E$ |
| 0x0027 | Free-run not supported | $P \rightarrow S$ | $P+E$ |
| 0x0028 | Synchronization not supported | $P \rightarrow S$ | P + E |
| 0x0029 | Free-run needs 3 Buffer Mode | $P \rightarrow S$ | P + E |
| 0x002A | Background Watchdog | S, O | P + E |
| 0x002B | No Valid Inputs and Outputs | $\mathrm{O}, \mathrm{S} \rightarrow \mathrm{O}$ | S+E |
| 0x002C | Fatal Sync Error | 0 | S + E |
| 0x002D | No Sync Error | $\mathrm{S} \rightarrow \mathrm{O}$ | S + E |
| 0x0030 | Invalid DC SYNC Configuration | $\mathrm{O}, \mathrm{S} \rightarrow \mathrm{O}, \mathrm{P} \rightarrow \mathrm{S}$ | $P+E, S+E$ |
| $0 \times 0031$ | Invalid DC Latch Configuration | $\mathrm{O}, \mathrm{S} \rightarrow \mathrm{O}, \mathrm{P} \rightarrow \mathrm{S}$ | $P+E, S+E$ |
| 0x0032 | PLL Error | $\mathrm{O}, \mathrm{s} \rightarrow \mathrm{O}$ | S + E |
| 0x0033 | DC Sync IO Error | $0, \mathrm{~s} \rightarrow 0$ | S + E |
| 0x0034 | DC Sync Timeout Error | $\mathrm{O}, \mathrm{S} \rightarrow \mathrm{O}$ | S +E |
| 0x0035 | DC Invalid Sync Cycle Time | $P \rightarrow S$ | P + E |
| $0 \times 0036$ | DC SyncO Cycle Time | $P \rightarrow S$ | $P+E$ |
| 0x0037 | DC Sync1 Cycle Time | $P \rightarrow S$ | P + E |
| 0x0041 | MBX_AOE | B, P, S, O | Current state + E |
| 0x0042 | MBX_EOE | B, P, S, O | Current state +E |
| $0 \times 0043$ | 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 |  |  |

Manufacturer-specific error codes

| Code | Description | Current State | Resulting State |
| :--- | :--- | :--- | :--- |
| $0 \times 8000$ | no module recognized or <br> present | I | $\mathrm{I}+\mathrm{E}$ |
| $0 \times 8001$ | Module at system bus <br> needs update | $\mathrm{P}>\mathrm{S}$ | $\mathrm{P}+\mathrm{E}$ |
| $0 \times 8002$ | Init error | $\mathrm{P}>\mathrm{S}$ | $\mathrm{PO}+\mathrm{E}$ |
| $0 \times 8003$ | unexpected restart <br> (Watchdog) | $\mathrm{P}>\mathrm{S}$ | $\mathrm{PO}+\mathrm{E}$ |

## SDO 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) |

## 11 DeviceNet communication

## 11.1 <br> 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 500 m at a rate of $125 \mathrm{kbit} / \mathrm{s}, 250 \mathrm{~m}$ at a rate of $250 \mathrm{kbit} / \mathrm{s}$ or 100 m at a rate of $500 \mathrm{kbit} / \mathrm{s}$.
The length of the tap lines can be up to 6 m 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 method

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.

## 11.2

## Access to the I/O system 1000

The following illustration shows the access under DeviceNet to the areas "I/O", "parameters" and "diagnostics".


## 1 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.

## 11 DeviceNet communication

Access to the I/O system 1000
Access to I/O area

### 11.2.1

## Access to 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 distributor 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.

## Configure DeviceNet scanner (master)

- Switch off the voltage supply of the DeviceNet bus coupler and set the baud rate and the DeviceNet address (■54).
- Start your configuration tool for the DeviceNet scanner.
- Set the "POLL IO" connection type in the DeviceNet scanner.
- Enter the number of input and output data: Number of input data: Produced connection size Number of output data: Consumed connection size
- Enter a basic address for the input and output data (mapping).
- Activate the DeviceNet bus coupler EPM-S150 in the scan list.
- Start the DeviceNet scanner.

After the DeviceNet scanner has been configured, the input and output modules can be triggered under the configured addresses.

### 11.2.2 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 parameter setting

- 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 |  |
| $2 x$ | 2 bytes | Hexadecimal | 2xhhhh |  |
| $4 x$ | 4 bytes | Hexadecimal | 4xhhhhhhhh |  |
| + | 1 byte | decimal positive | +ddd |  |
| $2+$ | 2 bytes | decimal positive | 2+ddddd |  |
| $4+$ | 4 bytes | decimal positive | 4+ddddddddd |  |
| - | 2 byte | decimal negative | -ddd |  |
| $2-$ | 4 bytes | decimal negative | 2-ddddd |  |
| $4-$ | decimal negative | 4-dddddddddd |  |  |
| b |  | Binary | xbbbbbbb |  |

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 $2 \times 7$ FFF $2 \times 8000 \times C 1 \times 022 \times 7$ FFF $2 \times 8000$

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 (■54).
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.

## 11 DeviceNet communication

Access to the I/O system 1000
Access to diagnostic data

### 11.2.3 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 ( 64 h)

| No. | Name | Information | Format | Example |
| :---: | :--- | :--- | :--- | :--- |
| $\mathbf{1}$ | 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, <br> String | 00000205 |
| 5 | FpgaVersion | FPGA version | Unsigned16, <br> 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 <br> number) |

## I/O compound module

## Class code:

Slot 1: $101\left(65_{h}\right)$
Slot 2: $102\left(66_{h}\right)$
...
Slot 64: 164 (A4h)

| 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 | $\begin{aligned} & \text { x00 x15 x00 x00 } \\ & \text { x73 x08 x04 x00 } \\ & \text { x00 x00 x00 x00 } \\ & \text { x00 x00 x00 x00 } \\ & 4 \times 000020 E B \end{aligned}$ |

### 11.3 Setting the parameters of analog I/O

11.3.1 2 analog inputs 0 ... 10 V (12 bits) - EPM-S400

Parameter data

| Pos. in <br> string of <br> digits | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- |
| 1 | Function channel 1 | $16\left(10_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{d e c}$ <br> $32\left(20_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\text {dec }}$ <br> $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated | $10_{\mathrm{h}}$ |
| 2 | Function channel 2 | $10_{h}$ |  |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 27648 \\ & \mathrm{D}=27648 * \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \text { V }}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | $8192$ | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

## 11 DeviceNet communication

Setting the parameters of analog I/O
4 analog inputs 0 ... 10 V ( 12 bits) - EPM-S401
11.3.2

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

## Parameter data

| Pos. in <br> string of <br> digits | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- |
| 1 | Function channel 1 |  |  |
| 2 | Function channel 2 | $16\left(10_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\text {dec }}$ |  |
| $32\left(20_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}}$ | $10_{h}$ |  |  |
| 3 | Function channel 3 | $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated | $10_{h}$ |
| 4 | Function channel 4 |  | $10_{h}$ |

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.) | $\begin{aligned} & \text { Voltage (U) } \\ & {[\mathrm{V}]} \end{aligned}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | $13824$ | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

Setting the parameters of analog I/O 2 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ ( 12 bits) - EPM-S402
11.3.3

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

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... ${27648_{\text {dec }}}$ $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0 . . .{16384_{\mathrm{dec}}}$ 49 (31h): 0 ... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |
| 2 | Function channel 2 |  | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{\mathrm{h}}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648 *(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{h}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

## 11 DeviceNet communication

Setting the parameters of analog $1 / 0$
4 analog inputs 0/4 ... 20 mA (12 bits) - EPM-S403
11.3.4

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

## Parameter data

| Pos. in <br> string of <br> digits | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- |
| 1 | Function channel 1 | $48\left(30_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{d e c}$ |  |
| 2 | Function channel 2 | $64\left(40_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{d e c}$ | $31_{h}$ |
| 3 | Function channel 3 | $49\left(31_{h}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{d e c}$ |  |
| $65\left(41_{h}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{d e c}$ | $31_{h}$ |  |  |
| 4 | Function channel 4 | $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated | $31_{\mathrm{h}}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | $\begin{gathered} 10 \\ 0 \end{gathered}$ | $\begin{gathered} 13824 \\ 0 \end{gathered}$ | $\begin{aligned} & 3600 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{\mathrm{h}}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | $\begin{gathered} 10 \\ 0 \end{gathered}$ | $\begin{gathered} 8192 \\ 0 \end{gathered}$ | $\begin{aligned} & 2000 \\ & 0000 \end{aligned}$ | Nominal range |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D * 16 / 27648+4 \\ & D=27648^{*}(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | $\begin{gathered} 12 \\ 4 \end{gathered}$ | 13824 <br> 0 | $\begin{aligned} & 3600 \\ & 0000 \end{aligned}$ |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\underset{\left(40_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 4 | $\begin{gathered} 8192 \\ 0 \end{gathered}$ | $\begin{aligned} & 2000 \\ & 0000 \end{aligned}$ |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Diagnostics | Bits 0 ... 5: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 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 | $00_{\text {h }}$ |
| 4 | Interference frequency suppression | Bit 1, 0: Channel 1 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bit 3, 2: Channel 2 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 5 | Function channel 1 | $\begin{aligned} & 12\left(12_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots 24^{2} . .28_{\text {dec }} \\ & 34\left(22_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-16384 . .16384_{\text {dec }} \\ & 16(10 \mathrm{~h}): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\text {dec }} \\ & 32\left(20_{\mathrm{h}}\right): 0 . .10 \mathrm{~V} / 0 \ldots 16384_{\text {dec }} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
| 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. <br> 7FFF h : Limit value alarm deactivated | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |
| 9 | Function channel 2 | $\begin{aligned} & 12\left(12_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots 27648_{\text {dec }} \\ & 34\left(22_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-16384 . .16384_{\text {dec }} \\ & 16\left(1 \mathrm{~h}_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\text {dec }} \\ & 32\left(20_{\mathrm{h}}\right): 0 . .10 \mathrm{~V} / 0 \ldots 16384_{\text {dec }} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
| 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. <br> 7FFF ${ }_{h}$ : Limit value alarm deactivated | 7FFF ${ }_{\text {h }}$ |
| 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000_{h}$ |

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

11 DeviceNet communication
Setting the parameters of analog $1 / \mathrm{O}$
2 analog inputs - $10 \ldots+10 \mathrm{~V}$ ( 16 bits) - EPM-S406

| Measuring range (Fct. no.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(12_{\mathrm{h}}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -13824 | CA00 |  |  |
|  | -10 | -27648 | 9400 |  |  |
|  | -11.76 | -32512 | 8100 | Underflow |  |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(22_{\mathrm{h}}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 | Nominal range |  |
|  | 5 | 8192 | 2000 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -8192 | E000 |  |  |
|  | -10 | -16384 | C000 |  |  |
|  | -12.5 | -20480 | B000 | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648^{*} U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -1.76 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 | Nominal range |  |
|  | 5 | $8192$ | $2000$ |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -2 | -3277 | F333 | Underflow |  |

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Diagnostics | Bits 0 ... 5: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{h}$ |
| 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 | $00_{h}$ |
| 4 | Interference frequency suppression | Bit 1, 0: Channel 1 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bit 3, 2: Channel 2 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 5 | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... ${27648_{\text {dec }}}$ $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0$... 16384 ${ }_{\text {dec }}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $41_{\text {h }}$ |
| 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. <br> $7 \mathrm{FFF}_{\mathrm{h}}$ : Limit value alarm deactivated | $7 \mathrm{FFF}_{\text {h }}$ |
| 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000{ }_{\text {h }}$ |
| 9 | Function channel 2 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0$... ${16384_{\text {dec }}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ $65\left(41_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $41_{\text {h }}$ |
| 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. <br> 7FFF ${ }_{h}$ : Limit value alarm deactivated | 7FFF ${ }_{\text {h }}$ |
| 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. <br> $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |

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

11 DeviceNet communication
Setting the parameters of analog $1 / \mathrm{O}$
2 analog inputs $0 / 4 \ldots 20 \mathrm{~mA}$ (16 bits) - EPM-S408

| Measuring range (Fct. no.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 10 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -3.52 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{\mathrm{h}}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 10 | 8192 | 2000 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -4 | -3277 | F333 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{h}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 12 | 13824 | 3600 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 1,19 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{\mathrm{h}}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0.8 | -3277 | F333 | Underflow |  |

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Reserved | 0 |  |
| 2 | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; 1 = activated) <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 3 | Function channel 1 | $\begin{aligned} & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots . .27648_{\mathrm{dec}} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $10_{\text {h }}$ |
| 4 | Function channel 2 |  | $10_{\text {h }}$ |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

## 11 DeviceNet communication

Setting the parameters of analog I/O
4 analog outputs 0 ... 10 V (12 bits) - EPM-S501
11.3.8

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

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Reserved | 0 |  |
| 2 | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 3 | Function channel 1 | $\begin{aligned} & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots{27648_{\mathrm{dec}}}^{32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $10_{\text {h }}$ |
| 4 | Function channel 2 |  | $10_{\text {h }}$ |
| 5 | Function channel 3 |  | $10_{\text {h }}$ |
| 6 | Function channel 4 |  | $10_{\text {h }}$ |

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

| Output area <br> (Fct. no.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 27648 \\ & \mathrm{D}=27648^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \text { V }}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & D=16384^{*} U / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |

Setting the parameters of analog I/O 2 analog outputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S502
11.3.9

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

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Reserved | 0 |  |
| 2 | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated $)$ <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 3 | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... ${27648_{\text {dec }}}$ $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |
| 4 | Function channel 2 |  | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{h}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | $13824$ | $3600$ | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 20 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{I} / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | $8192$ | $2000$ | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 12 | 13824 | 3600 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{\mathrm{h}}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 |  | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

## 11 DeviceNet communication

Setting the parameters of analog $1 / \mathrm{O}$
4 analog outputs $0 / 4 \ldots 20 \mathrm{~mA}$ (12 bits) - EPM-S503

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

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Reserved | 0 |  |
| 2 | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 3 | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... ${27648_{\text {dec }}}$ <br> $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ <br> $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ <br> $65\left(41_{\mathrm{h}}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ <br> $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |
| 4 | Function channel 2 |  | $31_{\text {h }}$ |
| 5 | Function channel 3 |  | $31_{\text {h }}$ |
| 6 | Function channel 4 |  | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{h}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 | Nominalrange |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} \mathrm{I} / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | $8192$ | $2000$ | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D * 16 / 27648+4 \\ & D=27648 *(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 12 | 13824 | 3600 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{h}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 16 / 16384+4 \\ & D=16384^{*}(I-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | Underflow |  |

### 11.4 Parameterising the temperature measurement

11.4.1 Four (two) analog inputs for resistance tests - EPM-S404

Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 0 ... 5: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 2 | Wire-breakage detection | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 3 | Limit value monitoring | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 4 | Reserved |  |  |
| 5 | Temperature system | Bit $0,1: 00_{b}={ }^{\circ} \mathrm{C} ; 01_{\mathrm{b}}={ }^{\circ} \mathrm{F} ; 10_{\mathrm{b}}=\mathrm{K}$ Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 6 | Interference frequency suppression | Bit 0, 1: $01_{b}=60 \mathrm{~Hz} ; 10_{\mathrm{b}}=50 \mathrm{~Hz}$ Bits 2 ... 7: Reserved | $02_{\text {h }}$ |


| Pos. in <br> string of <br> digits | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Channel 1 |  |  |  |
| 7 | Function channel 1 | Thermal |  |

7


| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 8 | Conversion time channel 1 | You can set the transformer speed as a function of the interference frequency suppression (see $0 \times 3105 / \mathrm{x}$ ) for each channel. <br> 0 ( $00_{\mathrm{h}}$ ): At 50 Hz : $324.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $270.5 \mathrm{~ms} /$ channel 16 bits <br> 1 ( $01_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 164.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $137.2 \mathrm{~ms} /$ channel 16 bits <br> 2 ( $02_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 84.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 70.5 \mathrm{~ms} /$ channel 16 bits 3 ( $03_{\mathrm{h}}$ ): at $50 \mathrm{~Hz}: 44.1 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 37.2 \mathrm{~ms} /$ channel 16 bits 4 (04h): At $50 \mathrm{~Hz}: 24.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 20.5 \mathrm{~ms} /$ channel 16 bits 5 ( $05_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 14.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 12.2 \mathrm{~ms} /$ channel 16 bits 6 (06h): At $50 \mathrm{~Hz}: 9.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 8.0 \mathrm{~ms} /$ channel 16 bits 7 ( $077_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 6.6 \mathrm{~ms} /$ channel 15 bits; at $60 \mathrm{~Hz}: 5.9 \mathrm{~ms} /$ channel 15 bits 8 (08) : At $50 \mathrm{~Hz}: 4.2 \mathrm{~ms} /$ channel 13 bits; at $60 \mathrm{~Hz}: 3.8 \mathrm{~ms} /$ channel 13 bits | $00_{\text {h }}$ |
| 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 7FFFh for the upper or 8000h for the lower limit value, the corresponding limit value is deactivated. <br> If your measured value is outside a limit value and you have activated the limit value monitoring, a process alarm is triggered. | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| 10 | Lower limit value channel 1 |  | $8000_{\text {h }}$ |
| Channel 2 |  |  |  |
| 11 | Function channel 2 | See channel 1 | $50_{\text {h }}$ |
| 12 | Conversion time channel 2 | See channel 1 | $00_{\text {h }}$ |
| 13 | Upper limit value channel 2 | See channel 1 | 7FFF ${ }_{\text {h }}$ |
| 14 | Lower limit value channel 2 |  | $8000_{\text {h }}$ |
| Channel 3 (for two-wire conductor connections only) |  |  |  |
| 15 | Function channel 3 | See channel 1 | $50_{\text {h }}$ |
| 16 | Conversion time channel 3 | See channel 1 | $00_{\text {h }}$ |
| 17 | Upper limit value channel 3 | See channel 1 | 7FFF ${ }_{\text {h }}$ |
| 18 | Lower limit value channel 3 |  | $8000{ }_{\text {h }}$ |

11 DeviceNet communication
Parameterising the temperature measurement
Four (two) analog inputs for resistance tests - EPM-S404

| Pos. in <br> string of <br> digits <br> Channel 4 (for two-wire conductor connections only) | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- |
| 19 | Function channel 4 | See channel 1 |  |
| 20 | Conversion time <br> channel 4 | See channel 1 | $50_{h}$ |
| 21 | Upper limit value <br> channel 4 | See channel 1 | $00_{h}$ |
| 22 | Lower limit value <br> channel 4 |  | $7 \mathrm{FFF}_{\mathrm{h}}$ |

## 1 Note!

- Use parameter setting to deactivate unused inputs.
- If thermal detectors are connected in a 3 or 4 conductor setup, channels 2 and/or 4 must be deactivated.
- The module does not provide any auxiliary supply for sensors.

Measuring range

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 2-wire: PT100 } \\ \left(50_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | -2430 dec | Underflow |
| $\begin{aligned} & \text { 2-wire: PT1000 } \\ & \left(51_{h}\right) \end{aligned}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | -2430 dec | Underflow |
| $\begin{gathered} \text { 2-wire: NI100 } \\ \left(52_{h}\right) \end{gathered}$ | $+295^{\circ} \mathrm{C}$ | +2950 dec | Overflow |
|  | $-60 . .+250^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| $\begin{aligned} & \text { 2-wire: NI1000 } \\ & \left(53_{h}\right) \end{aligned}$ | $+295{ }^{\circ} \mathrm{C}$ | +2950 dec | Overflow |
|  | $-60 \ldots+250^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| $\begin{aligned} & \text { 3-wire: PT100 } \\ & \left(58_{h}\right) \end{aligned}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430{ }_{\text {dec }}$ | Underflow |
| $\begin{aligned} & \text { 3-wire: PT1000 } \\ & \left(59_{\mathrm{h}}\right) \end{aligned}$ | $+1000^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 . . .+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | -2430 dec | Underflow |
| 3-wire: NII00 (5Ah) | $+295^{\circ} \mathrm{C}$ | +2950 dec | Overflow |
|  | $-60 \ldots+250^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 3-wire: NI1000 (5Bh) | $+295^{\circ} \mathrm{C}$ | +2950 dec | Overflow |
|  | $-60 \ldots+250^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | -1050 dec | Underflow |
| $\begin{aligned} & \text { 4-wire: PT100 } \\ & \left(60_{h}\right) \end{aligned}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | -2430 dec | Underflow |

Parameterising the temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{aligned} & \text { 4-wire: PT1000 } \\ & \left(61_{h}\right) \end{aligned}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 . . .+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 4-wire: NI100 } \\ \left(62_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 4-wire: NI1000 (63h) | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | -1050 dec | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(70_{h}\right) \end{gathered}$ | - |  | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots . .600 \Omega \\ \left(71_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 \ldots 32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(72_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(78_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(79_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(7 A_{h}\right) \end{gathered}$ | - |  | Overflow |
|  | $0 . .3000 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(80_{h}\right) \end{gathered}$ | - |  | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(81_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(82_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | 0 ... 32767 | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(90_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots . .600 \Omega \\ \left(91_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .6000{ }_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(92_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |

11 DeviceNet communication
Parameterising the temperature measurement
Four (two) analog inputs for resistance tests - EPM-S404

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(98_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .60 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 600 \Omega \\ \left(99_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(9 A_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(A 0_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 \ldots 6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(\mathrm{A1}_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{~A} 2_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(D 0_{h}\right) \end{gathered}$ | 70.55 ת | $32511_{\text {dec }}$ | Overflow |
|  | $0 . .60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots . .600 \Omega \\ \left(D 1_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{D} 2_{\mathrm{h}}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(D 8_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . .60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(D 9_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{DA}_{h}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(E O_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(E 1_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(E 2_{h}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |

### 11.4.2 Two analog inputs for thermocouple measurement - EPM-S405

## Parameter data

$\left.\begin{array}{l|l|l|l|l}\begin{array}{l}\text { Pos. in } \\ \text { string of } \\ \text { digits }\end{array} & \text { Name } & \text { Description/value } & \text { Lenze } \\ \hline 01 & \text { Diagnostics } & \begin{array}{l}\text { A diagnostic alarm occurs if the same event triggers a further } \\ \text { process alarm during a process alarm processing. } \\ \text { Bits 0 } \ldots 5: \text { Reserved } \\ \text { Bit 6: diagnostic alarm }(0=\text { inhibited; } 1=\text { enabled })\end{array} & 00_{\mathrm{h}} \\ & & \text { Bit 7: Reserved }\end{array}\right]$

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 08 | Conversion time channel 1 | You can set the transformer speed as a function of the interference frequency suppression (see $0 \times 3105 / x$ ) for each channel. <br> $0\left(00_{\mathrm{h}}\right)$ : At $50 \mathrm{~Hz}: 324.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $270.5 \mathrm{~ms} / \mathrm{channel} 16$ bits <br> $1\left(01_{h}\right)$ : at 50 Hz : $164.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $137.2 \mathrm{~ms} /$ channel 16 bits <br> 2 (02h): At $50 \mathrm{~Hz}: 84.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $70.5 \mathrm{~ms} /$ channel 16 bits <br> $3\left(03_{h}\right)$ : at $50 \mathrm{~Hz}: 44.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $37.2 \mathrm{~ms} /$ channel 16 bits <br> 4 ( $04_{h}$ ): At $50 \mathrm{~Hz}: 24.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $20.5 \mathrm{~ms} /$ channel 16 bits <br> 5 ( 05 h ): at $50 \mathrm{~Hz}: 14.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $12.2 \mathrm{~ms} /$ channel 16 bits <br> 6 ( 06 h): at 50 Hz : $9.2 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : <br> $8.0 \mathrm{~ms} / \mathrm{channel} 16$ bits <br> $7\left(07_{h}\right)$ : at 50 Hz : $6.6 \mathrm{~ms} /$ channel 15 bits; at 60 Hz : <br> $5.9 \mathrm{~ms} /$ channel 15 bits <br> $8\left(08_{h}\right)$ : at $50 \mathrm{~Hz}: 4.2 \mathrm{~ms} / c h a n n e l ~ 13$ bits; at 60 Hz : <br> $3.8 \mathrm{~ms} / \mathrm{ch}$ annel 13 bits | 02 h |
| 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 7FFFh for the upper or 8000h for the lower limit | $7 \mathrm{FFF}_{\mathrm{h}}$ |
| OA | Lower limit value channel 1 | 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. | $8000{ }_{\text {h }}$ |
| Channel 2 |  |  |  |
| OB | Function channel 2 | See channel 1 | $\mathrm{C1} \mathrm{~h}$ |
| OC | Conversion time channel 2 | See channel 1 | 02h |
| OD | Upper limit value channel 2 | See channel 1 | 7FFF ${ }_{\text {h }}$ |
| OE | Lower limit value channel 2 |  | $8000{ }_{\text {h }}$ |

## Measuring range

| Measuring range (Fct. no.) | Measured value |  |  | Range |
| :---: | :---: | :---: | :---: | :---: |
|  | [ ${ }^{\text {C }}$ ] | [ $\left.{ }^{\circ} \mathrm{F}\right]$ | [K] |  |
| Type J: $-210 \ldots+1200^{\circ} \mathrm{C}$ | +14500 | 26420 | 17232 | Overflow |
| $\begin{gathered} -346 \ldots 2192^{\circ} \mathrm{F} \\ 63.2 \ldots 1473.2^{\mathrm{K}} \end{gathered}$ | -2100 ... +12000 | -3460 ... +21920 | 632 ... 14732 | Nominal range |
| ( $\mathrm{BO}_{\mathrm{h}}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> $\left(\mathrm{CO}_{\mathrm{h}}\right.$ : int. comp. $0^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |
| $\begin{aligned} & \text { Type K: } \\ & \quad-210 \ldots+1372{ }^{\circ} \mathrm{C} \end{aligned}$ | +16220 | 29516 | 18952 | Overflow |
| $\begin{gathered} -454 \ldots 2501.6^{\circ} \mathrm{F} \\ 0 \ldots 1645.2 \mathrm{~K} \end{gathered}$ | -2700 ... +13720 | -4540 ... 25016 | 0 ... 16452 | Nominal range |
| ( $\mathrm{B} 1_{\mathrm{h}}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> ( $\mathrm{C1}_{\mathrm{h}}$ : int. comp. $0^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |



## 11 DeviceNet communication

## Parameterising the counter

One counter 32 bits, 24 V DC - EPM-S600

### 11.5 Parameterising the counter

11.5.1 One counter 32 bits, 24 V DC - EPM-S600

The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> range. |
| Counting periodically |  | | Signal evaluation |
| :--- |
| Single rotary <br> transducer |
| Double rotary <br> transducer |
| Quadruple rotary <br> transducer |
| Connection to input "A/pulse" and "B/direction" |


| Additional functions | Description |
| :---: | :---: |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate), hardware gate (HW gate), and software gate (SW gate) is made. <br> - The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). <br> - The SW gate is controlled via your user program (status word in the output area). <br> - The HW gate is controlled via the digital gate input. <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

Further information can be found in the chapter "Product description".

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 5 ... 0: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
| 2 | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | $02_{\text {h }}$ |
| 3 | Input frequency track B |  | 02h |
| 4 | Input frequency latch |  | 02h |
| 5 | Input frequency gate |  | $02_{\text {h }}$ |
| 6 | Input frequency reset |  | $00_{\text {h }}$ |
| 7 | Reserved |  |  |
| 8 | Alarm response | Setting activates process alarm <br> Bit 0: Proc. alarm HW gate open <br> Bit 1: Proc. alarm HW gate closed <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bit 6: Proc. alarm latch value <br> Bit 7: Reserved | $80_{\text {h }}$ |
| 9 | Counter function | Bit 5 ... 0 : <br> $000000_{b}=$ counting continuously <br> $000001_{b}=$ counting once, main counting direction forwards <br> $000010_{b}=$ counting once, main counting direction backwards <br> $000100_{b}=$ counting once, no main counting direction <br> $001000_{b}=$ counting periodically, main counting direction forwards <br> $010000_{b}=$ counting periodically, main counting direction backwards <br> $100000_{b}=$ counting periodically, no main counting direction <br> Bits 7 ... 6: Reserved | $40_{\text {h }}$ |
| 10 | Comparator | Bit 2 ... 0 : output switches (... if condition is met) $000_{b}=$ never <br> $001_{b}=$ count value $\geq$ comparison value <br> $010_{b}=$ count value $\leq$ comparison value <br> $100_{b}=$ count value $=$ comparison value <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 6 ... 4: Reset <br> $000_{b}=$ deactivated <br> $001_{b}=$ HIGH level <br> $011_{b}=$ rising edge <br> $101_{b}=$ rising edge, once <br> Bit 7: Reserved | 00 h |


| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 11 | Signal evaluation | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at $A$ and $B$ ) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bit 6 ... 3: Hardware gate (HW gate) <br> $000_{b}=$ deactivated (counter starts by setting SW gate) <br> $001_{b}=$ activated (HIGH level at gate activates the HW gate. <br> Counter starts if HW and SW gate are set.) <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{\text {h }}$ |
| 12 | Final value | Counting method: <br> 0x310B: byte 3 (high byte) <br> 0x3110: byte 2 <br> 0x3111: byte 1 <br> $0 \times 3112$ : byte 0 (low byte) | $00_{\text {h }}$ |
| 13 | Loading value | Counting method: <br> 0x310B: byte 3 (high byte) <br> 0x310C: byte 2 <br> 0x310D: byte 1 <br> $0 \times 310 \mathrm{E}$ : byte 0 (low byte) | $00_{\text {h }}$ |
| 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. <br> The hysteresis has an effect on the zero crossing, comparison, overflow/underflow. | $00_{\text {h }}$ |
| 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. | $00_{h}$ |

## 11 DeviceNet communication

## Parameterising the counter

Two counters 32 bits, 24 V DC - EPM-S601

### 11.5.2

Two counters 32 bits, 24 V DC - EPM-S601
The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| continuously | The counter counts once/periodically from the loading value in the specified counting <br> Counting periodically |
| range. |  |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input " $\mathrm{A} / \mathrm{pulse}$ " and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

Further information can be found in the chapter "Product description".

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 5 ... 0: reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| 2 | Input frequency counter 1, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | $02_{\text {h }}$ |
| 3 | Input frequency counter 1, track B |  | $02_{\text {h }}$ |
| 4 | Input frequency counter 2, track A |  | $02_{\text {h }}$ |
| 5 | Input frequency counter 2, track B |  | 02h |
| 6 | Alarm response counter 1 | Setting activates process alarm Bits 1 ... 0: reserved <br> Bit 2: proc. alarm overflow <br> Bit 3: proc. alarm underflow <br> Bit 4: proc. alarm comparison value <br> Bit 5: proc. alarm final value <br> Bits 7 ... 6: reserved | $00_{\text {h }}$ |
| 7 | Counter function counter 1 | ```Bits 5 ... 0: 000000 000001 b = once: forwards 000010b = once: backwards 000100b = once: no main counting direction 001000b 0100000b = periodically: backwards 100000}b= periodically: no main counting directio Bits 7 ... 6: reserved``` | $00_{\text {h }}$ |
| 8 | Comparator counter 1 | Bits $2 \ldots 0$ : comparison bit is set (... if condition is met) <br> $000_{b}=$ never <br> $001_{b}=$ count value $\geq$ comparison value <br> $010_{b}=$ count value $\leq$ comparison value <br> $100_{b}=$ count value $=$ comparison value <br> Bit 3: invert counting direction track $B$ <br> 0 = no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: reserved | $00_{\text {h }}$ |
| 9 | Signal evaluation counter 1 | Bits 2 ... 0: signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at A and B) <br> $011_{b}=$ rotary transducer quadruple (at A and B) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: reserved <br> Bit 7: gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{h}$ |


| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 10 | Set value counter 1 | Counting method: <br> 0x3111: byte 3 (high byte) <br> 0x3112: byte 2 <br> 0x3113: byte 1 <br> $0 \times 3114$ : byte 0 (low byte) | $00_{\text {h }}$ |
| 11 | Final value counter 1 | Counting method: <br> 0x310D: byte 3 (high byte) <br> $0 \times 310 \mathrm{E}$ : byte 2 <br> 0x310F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
| 12 | Loading value counter 1 | Counting method: <br> 0x3109: byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> 0x310C: byte 0 (low byte) | $00_{\text {h }}$ |
| 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. <br> The hysteresis has an effect on the zero crossing, comparison, overflow/underflow. | $00_{\text {h }}$ |
| 14 | Reserved |  |  |
| 15 | Alarm response counter 2 | See counter 1 | $00_{\text {h }}$ |
| 16 | Counter function counter 2 | See counter 1 | $00_{\text {h }}$ |
| 17 | Comparator counter 2 | See counter 1 | $00_{\text {h }}$ |
| 18 | Signal evaluation counter 2 | See counter 1 | $00_{\text {h }}$ |
| 19 | Set value counter 2 | See counter 1 | $00_{\text {h }}$ |
| 20 | Final value counter 2 | See counter 1 | $00_{\text {h }}$ |
| 21 | Loading value counter 2 | See counter 1 | $00_{\text {h }}$ |
| 22 | Hysteresis counter 2 | See counter 1 | $00_{\text {h }}$ |

### 11.5.3 One counter 32 bits, 5 V DC - EPM-S602

The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting |
| Counting periodically | range. |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input " $\mathrm{A} / \mathrm{pulse}$ " and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input "A/pulse" and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

Further information can be found in the chapter "Product description".

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 5 ... 0: reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; 1 = enabled) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 2 | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02h |
| 3 | Input frequency track B |  | 02h |
| 4 | Input frequency reset |  | 02h |
| 5 | Reserved |  |  |
| 6 | Alarm response | Setting activates process alarm <br> Bits 1 ... 0: reserved <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bits 7 ... 6: Reserved | $00_{\text {h }}$ |
| 7 | Counter function | ```Bit 5 ... 0: 000000 000001 b = once: forwards 000010b = once: backwards 000100 b = once: no main counting direction 001000 b = periodically: forwards 010000 b = periodically: backwards 100000}b=\mathrm{ periodically: no main counting direction Bits 7 ... 6: Reserved``` | $00_{\text {h }}$ |
| 8 | Comparator | ```Bits 2 ... 0: Comparison bit is set (... if condition is met) 000b}=\mathrm{ = never 001b}=\mathrm{ count value }\geq\mathrm{ comparison value 010b}=\mathrm{ count value }\leq\mathrm{ comparison value 100}b=\mathrm{ count value = comparison value Bit 3: Invert counting direction track B 0 = no (do not invert) 1 = yes (invert) Bits 6 ... 4: Reset 000 001b}= = HIGH level 011b}= rising edg 101b = rising edge, once Bit 7: Reserved``` | $00_{\text {h }}$ |


| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 9 | Signal evaluation | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at $A$ and $B$ ) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: Reserved <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | $00_{h}$ |
| 10 | Final value | Counting method: <br> 0x310D: byte 3 (high byte) <br> $0 \times 310 \mathrm{E}$ : byte 2 <br> 0x311F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
| 11 | Loading value | Counting method: <br> 0x3109: byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> $0 \times 310 \mathrm{C}$ : byte 0 (low byte) | $00_{h}$ |
| 12 | Hysteresis |  | $00_{\text {h }}$ |

## 11 DeviceNet communication <br> Parameterising the counter <br> Two counters 32 bits, 24 V DC - EPM-S603

### 11.5.4 Two counters 32 bits, 24 V DC - EPM-S603

The following functions can be parameterised:

| Counting functions | Description |
| :--- | :--- |
| Counting <br> continuously | The counter counts from 0 to the counting limit, then skips to the opposite counting limit <br> and continues to count from there. |
| Signal evaluation <br> Single rotary <br> transducer | Description |
| Double rotary <br> transducer | Connection to input "A/pulse" and "B/direction" |
| Quadruple rotary <br> transducer | Pulse at input "A/pulse" and direction at "B/direction" |
| Direction | Description |
| Additional functions |  |
| The gate function serves to start, stop, and interrupt a counting function. In the case of |  |
| this counter the internal gate (l-gate) is conform to the software gate (SW gate) which |  |

Further information can be found in the chapter "Product description".

## Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Input frequency counter 0, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02h |
| 2 | Input frequency counter 0, track B |  | 02h |
| 3 | Input frequency counter 1, track A |  | $02_{\text {h }}$ |
| 4 | Input frequency counter 1, track B |  | 02h |
| 5 | Counting direction counter 0, track B | Bits 2 ... 0: Reserved <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 6 | Signal evaluation counter 0 | ```Bits 2 ... 0: Signal evaluation 000 the counter are ignored) 001 = rotary transducer single (at A and B) 010}=\mathrm{ = rotary transducer double (at A and B) 011}\mp@subsup{b}{b}{}=\mathrm{ rotary transducer quadruple (at A and B) 100b}=\mathrm{ direction (pulse at A and direction at B) Bits 7 ... 3: Reserved``` | $00_{\text {h }}$ |
| 7 | Counting direction counter 1, track B | Bits 2 ... 0: Reserved <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 8 | Signal evaluation counter 1 | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at $A$ and $B$ ) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 7 ... 3: Reserved | $00_{\text {h }}$ |

## 11.6

Parameterising the encoder evaluation
11.6.1

SSI - EPM-S604

Further information can be found in the chapter "Product description".
Parameter data

| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 1 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. <br> Bits 5 ... 0: reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| 2 | Dead 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. $\begin{aligned} & \text { HIGH LOW } \\ & 00_{h} 30_{h}: 1 \mu \mathrm{~s} \\ & 00_{h} 60_{h}: 2 \mu \mathrm{~s} \\ & 00_{h} C 0_{h}: 4 \mu \mathrm{~s} \\ & 01_{h} 80_{h}: 8 \mu \mathrm{~s} \\ & 03_{h} 00_{\mathrm{h}}: 16 \mu \mathrm{~s} \\ & 06_{\mathrm{h}} 00_{\mathrm{h}}: 32 \mu \mathrm{~s} \\ & 09_{\mathrm{h}} 00_{\mathrm{h}}: 48 \mu \mathrm{~s} \\ & 0 C_{h} 00_{\mathrm{h}}: 64 \mu \mathrm{~s} \end{aligned}$ | 0 COO h |
| 3 | Baud rate | In the "Monitoring operation" mode, the baud rate is irrelevant. Enter the baud rate here. This corresponds to the clock frequency via which the connected encoder communicates. More information on this can be found in the data sheet for your encoder. <br> HIGH LOW <br> $00_{h} 18_{h}: 2 \mathrm{MHz}$ <br> $00_{\mathrm{h}} 20_{\mathrm{h}}: 1.5 \mathrm{MHz}$ <br> $00_{h} 30_{h}: 1 \mathrm{MHz}$ <br> $00_{h} 60_{h}: 500 \mathrm{kHz}$ <br> $00_{h} \mathrm{CO}_{\mathrm{h}}: 250 \mathrm{kHz}$ <br> $01_{\mathrm{h}} 80_{\mathrm{h}}: 125 \mathrm{kHz}$ | 0180 $h$ |
| 4 | Reserved |  |  |
| 5 | Scaling | Depending on the encoder, further bits are transmitted in addition to the encoder value. Scaling serves to determine how many bits postpositioned 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. <br> Value range: $00_{h} \ldots 0 F_{h}=0$ bits ... 15 bits | 00 h |


| Pos. in string of digits | 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. <br> $7\left(07_{h}\right)=" 8$ bits" <br> $8\left(08_{h}\right)=$ " 9 bits" <br> $9\left(09_{h}\right)=" 10$ bits" <br> $10\left(0 A_{h}\right)=" 11$ bits" <br> $11\left(0 B_{h}\right)=" 12$ bits" <br> $12\left(0 C_{h}\right)=$ " 13 bits" <br> $13\left(0 D_{h}\right)=" 14$ bits" <br> $\left.140 \mathrm{E}_{\mathrm{h}}\right)=$ " 15 bits" <br> $15\left(0 F_{\mathrm{h}}\right)=$ " 16 bits" <br> $16\left(10_{h}\right)=" 17$ bits" <br> $17\left(11_{h}\right)=$ "18 bits" <br> $18(12 \mathrm{~h})=$ " 19 bits" <br> $19\left(13_{h}\right)=$ " 20 bits" <br> $20(14 \mathrm{~h})=$ " 21 bits" <br> $21(15 \mathrm{~h})=$ " 22 bits" <br> $22\left(16_{h}\right)=$ " 23 bits" <br> $23(17 \mathrm{~h})=$ " 24 bits" <br> $24(18$ h) = " 25 bits" <br> $25(19 \mathrm{~h})=$ " 26 bits" <br> $26\left(1 \mathrm{~A}_{\mathrm{h}}\right)=$ " 27 bits" <br> $27\left(1 \mathrm{~B}_{\mathrm{h}}\right)=$ " 28 bits" <br> $28\left(1 C_{h}\right)=$ "29 bits" <br> $29\left(1 D_{h}\right)=$ " 30 bits" <br> $30\left(1 \mathrm{E}_{\mathrm{h}}\right)=$ " 31 bits" <br> $31\left(1 \mathrm{~F}_{\mathrm{h}}\right)=$ " 32 bits" | 18 h |


| Pos. in string of digits | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: |
| 7 |  | Bit 1 ... 0: ready for operation <br> 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. <br> In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder. <br> $01_{b}=$ monitoring operation <br> $10_{b}=$ master mode <br> Bit 2: shifting direction <br> 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". <br> $0=$ LSB first (LSB is transmitted first) <br> $1=$ MSB first (MSB is transmitted first) <br> Bit 3: edge clock signal <br> 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. <br> 0 = falling edge <br> 1 = rising edge <br> Bit 4: coding <br> 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. <br> $0=$ standard code <br> 1 = Gray code <br> Bits 7 ... 5: reserved | $1 E_{h}$ |
| 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. <br> $0\left(00_{h}\right)=$ inhibited <br> $1\left(01_{h}\right)=$ enabled | $00_{\text {h }}$ |

## 11.7 <br> Time stamp parameterising

11.7.1 2 digital inputs with time stamp function - EPM-S207

Parameter data

| Pos. in string of digits | 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. | $\begin{aligned} & 14_{h} \text { or } 3 C_{h} \\ & \text { (fix) } \end{aligned}$ |
| 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. | 00h (fix) |
| 3 | Input delay DI1 | $\begin{aligned} & 00_{\mathrm{h}}=1 \mu \mathrm{~s} \\ & 02_{\mathrm{h}}=3 \mu \mathrm{~s} \\ & 04_{\mathrm{h}}=10 \mu \mathrm{~s} \\ & 07_{\mathrm{h}}=86 \mu \mathrm{~s} \\ & 09_{\mathrm{h}}=342 \mu \mathrm{~s} \\ & 0 C_{\mathrm{h}}=273 \mu \mathrm{~s} \\ & \text { Other values are not permissible. } \end{aligned}$ | 02h |
| 4 | Input delay DI2 |  | 02h |
| 5 | Edge 0-1 an DI x | Time stamp entry on rising edge Bit 0: DI1 (0: inhibit, 1 = enable) <br> Bit 1: DI2 (0: inhibit, 1 = enable) <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |
| 6 | Edge 1-0 at DI x | Time stamp entry on falling edge Bit 0: DI1 (0: inhibit, 1 = enable) <br> Bit 1: DI2 (0: inhibit, 1 = enable) Bits 7 ... 2: Reserved | $00_{\text {h }}$ |

11.7 .2

2 digital outputs with time stamp function - EPM-S310

## Parameter data

| Pos. in <br> string of <br> digits | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- |
| 1 | Length - process <br> image input data | Length of input data (backplane bus communication); the values <br> are specified by the system. Other values are not permissible. | $14_{h}$ or $3 C_{h}$ <br> (fix) |
| 2 | Length - process <br> image output data | Length of output data (backplane bus communication); the <br> values are specified by the system. Other values are not <br> permissible. | $00_{h}$ (fix) |

## 11 DeviceNet communication

Parameterising technology modules
2 digital outputs with PWM functionality - EPM-S620

### 11.8 Parameterising technology modules

11.8.1 2 digital outputs with PWM functionality - EPM-S620

Parameter data

| Pos. in string of digits | 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. | $1 \mathrm{~F} 40_{\text {h }}$ |
| 2 | PWM 2: Period | Values below $25 \mu$ s are ignored. If the pulse duration is higher or equal to the period, the DO output is set permanently. Value range: 1200 ... 8388607 ( $25 \mu \mathrm{~s}$... approx. 175 ms ) | 1 F 40 h |

## 11.8 .2

 RS232 interface - EPM-S640Parameter data

| Parameter data - ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | 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 | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| 4 | Baud rate | $00_{h}$ : 9600Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud $06_{h}: 2400$ Baud $07_{h}: 4800$ Baud 08h: 7200 Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud 0B ${ }_{h}$ : 19200 Baud $0^{0} C_{h}: 38400$ Baud 0D $\mathrm{h}: 57600$ Baud $0 \mathrm{E}_{\mathrm{h}}: 76800$ Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10 \mathrm{~h}: 109700$ Baud | $00_{\text {h }}$ |
| 5 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{h}: \text { ASCII }$ | $01_{\text {h }}$ |
| 6 | Data format | Bit 1/0 number of data bits $\begin{aligned} & 00_{b}: 5 \\ & 01_{b}: 6 \\ & 10_{b}: 7 \\ & 11_{b}: 8 \end{aligned}$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control <br> $00_{b}$ : None <br> $01_{b}$ : Hardware <br> $10_{b}$ : XON/XOFF | $13_{\text {h }}$ |
| 7 | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is | $00_{\text {h }}$ |
| 8 | ZNA (LOW byte) | executed. $0 \ldots 65535 \text { [ms] }\left(0000_{\mathrm{h}} \ldots . . \text { FFFF }_{\mathrm{h}}\right)$ | $00_{\text {h }}$ |

11 DeviceNet communication
Parameterising technology modules
RS232 interface - EPM-S640

| Parameter data-ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | Name | Description/value | Lenze |
| 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).$0 \ldots 65535 \text { [ms] }\left(0000_{h} . \therefore \text { FFFF }_{h}\right)$ | OAh |
| 10 | Character delay time (LOW byte) |  | OAh |
| 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. $1 \ldots 255\left(01_{h} \ldots F_{h}\right)$ | $01_{\text {h }}$ |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | 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 | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 4 | Baud rate | $00_{h}$ : 9600Baud $01_{h}$ : 150 Baud 02h: 300 Baud 03h: 600 Baud 04h: 1200 Baud 05h: 1800 Baud 06h: 2400 Baud $07_{h}$ : 4800 Baud 08h: 7200 Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud OCh: 38400 Baud 0D h : 57600 Baud $0 \mathrm{E}_{\mathrm{h}}: 76800$ Baud $0 \mathrm{~F}_{\mathrm{h}}: 115200$ Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{\text {h }}$ |
| 5 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{h}: \text { STX/ETX }$ | 02 h |
| 6 | Data format | Bit 1/0 number of data bits $\begin{aligned} & 00_{b}: 5 \\ & 01_{b}: 6 \\ & 10_{b}: 7 \\ & 11_{b}: 8 \end{aligned}$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | $13_{h}$ |
| 7 | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \text {... } 65535 \text { [ms] }\left(0000_{h} . . . \text { FFFF }_{h}\right)$ | 00 h |
| 8 | ZNA (LOW byte) | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \ldots 65535 \text { [ms] }\left(0000_{h} . . . \text { FFFF }_{h}\right)$ | 00 h |
| 9 | TMO | TMO serves to define the maximally permissible interval between two frames. $0 \ldots 65535 \text { [ms] }\left(0000_{\mathrm{h}} \ldots . . \text { FFFF }_{\mathrm{h}}\right)$ | OAh |

## 11 DeviceNet communication

Parameterising technology modules
RS232 interface - EPM-S640

| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | Name | Description/value | Lenze |
| 10 | No. of start identifiers | $00_{\mathrm{h}}: 1$ start identifier (2. start identifier is ignored) $01_{h}$ : 2 start identifiers | $01_{\text {h }}$ |
| 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. $1 . . .255\left(01_{h} \ldots F_{h}\right)$ | 02 h |
| 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. | $00_{h}$ |
| 13 | No. of end identifier | $00_{h}$ : 1 end identifier (2. end identifier ( $0 \times 310 \mathrm{D} / \mathrm{x}$ ) is ignored) <br> $01_{h}$ : 2 end identifiers | $01_{h}$ |
| 14 | End identifier 1 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | $03_{h}$ |
| 15 | End identifier 2 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} \ldots F F_{h}\right)$ | $00_{h}$ |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | 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 | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 4 | Baud rate | 00h: 9600Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud 06h: 2400 Baud $07{ }_{h}: 4800$ Baud $08_{h}: 7200$ Baud 09h: 9600 Baud $0 A_{h}: 14400$ Baud 0B ${ }_{h}$ : 19200 Baud 0C ${ }_{h}$ : 38400 Baud 0Dh: 57600 Baud $0 \mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 F_{h}: 115200$ Baud $10_{h}$ : 109700 Baud | 00 h |
| 5 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{h}: 3964 \\ & 03_{h}: 3964 R \end{aligned}$ | $03_{\text {h }}$ |
| 6 | Data format | Bit 1/0 number of data bits $\begin{aligned} & 00_{b}: 5 \\ & 01_{b}: 6 \\ & 10_{b}: 7 \\ & 11_{b}: 8 \end{aligned}$ <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control <br> $00_{b}$ : None <br> $01_{b}$ : Hardware <br> $10_{b}$ : XON/XOFF | 13 h |
| 7 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} \ldots F_{h}\right)$ | 00 h |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | Name | Description/value | Lenze |
| 8 | Character delay time | 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). <br> 0 ... 65535 [ $20-\mathrm{ms}$ steps] ( $00_{\mathrm{h}} . \therefore \mathrm{FF}_{\mathrm{h}}$ ) | OAh |
| 9 | Acknowledgement time | The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} . \therefore F F_{h}\right)$ | $0 \mathrm{~A}_{\mathrm{h}}$ |
| 10 | Block wait time | Maximum time between the recognition of a requested frame (DLE) and the response. $\left.0 \ldots 255 \text { [ms] ( } 00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | $0 \mathrm{~A}_{\mathrm{h}}$ |
| 11 | STX repetitions | Maximum number of times the EPM-S640 tries to establish a connection. <br> $0 . . .255$ [ms] $\left(00_{h} \ldots F_{h}\right)$ | 05h |
| 12 | DBL | If the block wait time has been exceeded, the number of STX repetitions for the requested protocol can be specified here. $0 \ldots 255[\mathrm{~ms}]\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | 06h |
| 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 the protocols 3964(R), both partners must have different priorities. <br> $00_{h}$ : LOW <br> $01_{h}$ : HIGH | $00_{\text {h }}$ |

## 11.8 .3

 RS485 interface - EPM-S650
## Parameter data

| Parameter data-ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | 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 | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| 4 | Baud rate | $00_{h}$ : 9600Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud $04_{h}$ : 1200 Baud $05 \mathrm{~h}: 1800$ Baud 06h: 2400 Baud $07{ }_{h}: 4800$ Baud 08h: 7200 Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud 0B $h$ : 19200 Baud $0 C_{h}: 38400$ Baud 0D $\mathrm{h}: 57600$ Baud $0 \mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{h}: 109700$ Baud | $00_{\text {h }}$ |
| 5 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{h} \text { : ASCII }$ | $01_{\text {h }}$ |
| 6 | Data format | Bit 1/0 number of data bits $\begin{aligned} & 00_{b}: 5 \\ & 01_{b}: 6 \\ & 10_{b}: 7 \\ & 11_{b}: 8 \end{aligned}$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control <br> $00_{b}$ : None <br> $01_{b}$ : Hardware <br> $10_{b}$ : XON/XOFF | $13_{h}$ |
| 7 | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is | $00_{h}$ |
| 8 | ZNA (LOW byte) | executed. $0 \ldots 65535[\mathrm{~ms}]\left(0000_{\mathrm{h}} . . . \text { FFFF }_{\mathrm{h}}\right)$ | $00_{\text {h }}$ |


| Parameter data-ASCII protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | Name | Description/value | Lenze |
| 9 <br>  <br> 10 | 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).$0 \text {... } 65535 \text { [ms] }\left(0000_{h} . . . \text { FFFF }_{h}\right)$ | OAh |
| 10 | Character delay time (LOW byte) |  | OAh |
| 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. $1 \ldots 255\left(01_{h} \ldots F_{h}\right)$ | $01_{\text {h }}$ |
| 12 | Operating mode | Operating mode of the interface $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. <br> Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $00_{\text {h }}$ |
| 13 | 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 ( $\mathbb{C}$ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal R(A) 5 V (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal $R(A) 0 V$; signal $R(B) 5 \mathrm{~V}$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $13_{h}$ |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | 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 | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 4 | Baud rate | $00_{h}$ : 9600Baud $01_{h}$ : 150 Baud 02h: 300 Baud 03h: 600 Baud 04h: 1200 Baud 05h: 1800 Baud 06h: 2400 Baud $07_{h}$ : 4800 Baud 08h: 7200 Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud OCh: 38400 Baud 0D h : 57600 Baud $0 \mathrm{E}_{\mathrm{h}}: 76800$ Baud $0 \mathrm{~F}_{\mathrm{h}}: 115200$ Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{\text {h }}$ |
| 5 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{h}: \text { STX/ETX }$ | 02 h |
| 6 | Data format | Bit 1/0 number of data bits $\begin{aligned} & 00_{b}: 5 \\ & 01_{b}: 6 \\ & 10_{b}: 7 \\ & 11_{b}: 8 \end{aligned}$ <br> Bit 3/2 parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | $13_{h}$ |
| 7 | ZNA (HIGH byte) | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \text {... } 65535 \text { [ms] }\left(0000_{h} . . . \text { FFFF }_{h}\right)$ | 00 h |
| 8 | ZNA (LOW byte) | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \ldots 65535 \text { [ms] }\left(0000_{h} . . . \text { FFFF }_{h}\right)$ | 00 h |
| 9 | TMO | TMO serves to define the maximally permissible interval between two frames. $0 \ldots 65535 \text { [ms] }\left(0000_{\mathrm{h}} \ldots . . \text { FFFF }_{\mathrm{h}}\right)$ | OAh |


| Parameter data STX/ETX protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | Name | Description/value | Lenze |
| 10 | No. of start identifiers | $00_{\mathrm{h}}$ : 1 start identifier (2. start identifier is ignored) $01_{\mathrm{h}}: 2$ start identifiers | $01_{\text {h }}$ |
| 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. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | $02_{\text {h }}$ |
| 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. | $00_{\text {h }}$ |
| 13 | No. of end identifier | $00_{h}$ : 1 end identifier (2. end identifier ( $0 \times 310 \mathrm{D} / \mathrm{x}$ ) is ignored) $01_{h}$ : 2 end identifiers | $01_{h}$ |
| 14 | End identifier 1 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | $03_{\text {h }}$ |
| 15 | End identifier 2 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | $00_{h}$ |
| 16 | Operating mode | Operating mode of the interface <br> $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{\mathrm{h}}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $00_{h}$ |
| 17 | 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 (■ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{\text {h }}$ Signal R(A) 5 V (open circuit monitoring); signal R(B) 0 V . In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal $R(A) 0 V$; signal $R(B) 5 V$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $13_{h}$ |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | 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 | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| 4 | Baud rate | 00h: 9600Baud $01_{h}$ : 150 Baud 02h: 300 Baud $03_{h}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud 06h: 2400 Baud $07{ }_{h}: 4800$ Baud $08_{h}: 7200$ Baud 09h: 9600 Baud $0 A_{h}: 14400$ Baud 0B ${ }_{h}$ : 19200 Baud 0C ${ }_{h}$ : 38400 Baud 0Dh: 57600 Baud $0 \mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 F_{h}: 115200$ Baud $10_{h}$ : 109700 Baud | 00 h |
| 5 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{h}: 3964 \\ & 03_{h}: 3964 R \end{aligned}$ | $03_{\text {h }}$ |
| 6 | Data format | Bit 1/0 number of data bits $\begin{aligned} & 00_{b}: 5 \\ & 01_{b}: 6 \\ & 10_{b}: 7 \\ & 11_{b}: 8 \end{aligned}$ <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{\mathrm{b}}: 1 \\ & 10_{\mathrm{b}}: 1.5 \\ & 11_{\mathrm{b}}: 2 \end{aligned}$ <br> Bit 7/6 flow control <br> $00_{b}$ : None <br> $01_{b}$ : Hardware <br> $10_{b}$ : XON/XOFF | 13 h |
| 7 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} \ldots F_{h}\right)$ | 00 h |


| Parameter data 3964(R) protocol |  |  |  |
| :---: | :---: | :---: | :---: |
| Pos. in string of digits | Name | Description/value | Lenze |
| 8 | Character delay time | 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). $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{\mathrm{h}} . . . \mathrm{FF}_{\mathrm{h}}\right)$ | OAh |
| 9 | Acknowledgement time | The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} \ldots F_{h}\right)$ | $0 A_{h}$ |
| 10 | Block wait time | Maximum time between the recognition of a requested frame (DLE) and the response. $0 \ldots 255 \text { [ms] }\left(00_{h} \ldots F_{h}\right)$ | $0 A_{h}$ |
| 11 | STX repetitions | Maximum number of times the EPM-S640 tries to establish a connection. $0 \ldots 255 \text { [ms] }\left(00_{h} \ldots F_{h}\right)$ | 05 h |
| 12 | DBL | If the block wait time has been exceeded, the number of STX repetitions for the requested protocol can be specified here. $0 \ldots 255[\mathrm{~ms}]\left(00_{\mathrm{h}} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | 06h |
| 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 the protocols 3964(R), both partners must have different priorities. <br> $00_{h}$ : LOW <br> $01_{h}$ : HIGH | $00_{\text {h }}$ |
| 16 | Operating mode | Operating mode of the interface $00_{\mathrm{h}}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $00_{\text {h }}$ |
| 17 | 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 ( $\mathbb{C}$ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal $R(A) 5 V$ (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal $R(A) 0$ V; signal $R(B) 5 \mathrm{~V}$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $13_{h}$ |

## 12 PROFINET communication

## 12.1 <br> 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:

- PROFINETIO
- 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 ( $A R=$ 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 100Mbit/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.

## 12.2

## 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 range
- 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) |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0009 \mathrm{~h} \\ +0 \end{gathered}$ | ... | $\begin{array}{r} \text { API } \\ +24 \end{array}$ | $\begin{aligned} & \text { Slot } \\ & +28 \end{aligned}$ | Subslot +30 | $\begin{gathered} \text { Index } \\ +34 \end{gathered}$ | Length $+36$ | $\ldots+64$ |
| Response with data (ReadResponse) |  |  |  |  |  |  |  |
| $\begin{gathered} 0009_{h} \\ +0 \end{gathered}$ | ... | $\begin{array}{r} \text { API } \\ +24 \end{array}$ | $\begin{aligned} & \text { Slot } \\ & +28 \end{aligned}$ | Subslot <br> +30 | $\begin{gathered} \text { Index } \\ +34 \end{gathered}$ | Length +36 | $\ldots+64$ |


| Request frame (WriteRequest) |
| :--- |
| $8008_{\mathrm{h}}$ <br> +0 F |

### 12.2.1

## Access to 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 $=8028 \mathrm{~h}$ : Reading of the input data (slot $1 \ldots$... 64)
- Index $=8029$ h: Reading of the output data (slot $1 . . .64$ )


### 12.2.2 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 | $007 \mathrm{D}_{\mathrm{h}}$ |
| All parameters of the PROFINET coupler | 0 | $007 \mathrm{E}_{\mathrm{h}}$ |
| All module parameters incl. header (4 bytes) | $1 \ldots 64$ | $007 \mathrm{D}_{\mathrm{h}}$ |
| Data record DS $00_{h}$ of the module parameters | $1 \ldots 64$ | $007 \mathrm{E}_{\mathrm{h}}$ |
| Data record DS $01_{h}$ of the module parameters | $1 \ldots 64$ | $007 \mathrm{~F}_{\mathrm{h}}$ |
| Data record DS $80_{h} \ldots 90_{h}$ of the module parameters | $1 \ldots 64$ | $0080_{h} \ldots 0090_{h}$ |

### 12.2.3

## 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 $=0000_{h}: 4$ Byte: | Byte 0: Diagnostic byte, byte $1 \ldots 3: 0$ (fix) |
| :--- | :--- |
| Index $=0001_{h}: 20$ Byte: | Byte 0: Diagnostic byte, byte $1 . .19: 0$ (fix) |


| Structure of diagnostic data of PROFINET bus coupler | Lenze |  |
| :---: | :--- | :---: |
| Byte | Bit $7 \ldots .0$ | $00_{\mathrm{h}}$ |
| 0 | Diagnostic byte <br> Bit 0: Error at backplane bus <br> Bit 1: Parameters have been rejected by the addressed module or coupler (error in data <br> consistency) <br> Bit 2: General bus coupler parameter error (data could not be saved) <br> Bit 3: Version error at the backplane bus (at least one module at the backplane bus is not <br> supported) <br> Bit 5, 4: 0 (fix) <br> Bit 6: Port error with activated port monitoring <br> Bit 7: Configuration error backplane bus (actual configuration unequal to setpoint <br> configuration) |  |
| $2 \ldots 3$ (19) | $00_{\mathrm{h}}$ (fix) | $00_{\mathrm{h}}$ |

## 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 $=0000_{\mathrm{h}}$ : Data record DS $00_{\mathrm{h}}$ of the diagnostic data
- Index $=0001_{\mathrm{h}}$ : Data record DS $01_{\mathrm{h}}$ of the diagnostic data


## 1 Note!

Information on how to assign the ranges can be found in the descriptions of the corresponding I/O compound module.

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Module fault, i.e. an error has been detected <br> Bit 1: Internal error in the module <br> Bit 2: External error - module cannot be addressed anymore <br> Bit 3: Channel error in module <br> Bit 4: External supply voltage is missing <br> Bit 5, 6: Reserved <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class <br> 1111: Digital module <br> 0101: Analog module <br> 1000: Counter module, SSI module <br> 0111: Time stamp module, gateway module <br> Bit 4: Channel information available <br> 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 <br> 70h: Module with digital inputs <br> 71h: Module with analog inputs <br> 72h: Module with digital outputs <br> 73h: Module with analog outputs <br> 74h: Module with analog inputs/outputs <br> 76h: 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 $\mu \mathrm{s}$ ticker when diagnostic data is generated <br> (The I/O compound module features a timer that is started with mains ON and restarts with 0 after 232-1 $\mu \mathrm{s}$.) |

### 12.3 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 |  |  |
| :---: | :---: | :---: |
| Byte | Bit 7 ... 0 | Lenze |
| 0 | Bit 0: Process alarm <br> 0 = inhibit <br> 1 = enable <br> Bit 1: Diagnostic alarm <br> 0 = inhibit <br> 1 = enable <br> Bit 2: \#Diagnostic alarm type <br> $0=$ manufacturer-specific data <br> 1 = channel-specific data <br> Bit 3 ... 6: Reserved <br> Bit 7: Data format <br> 0 = Data format Motorola <br> 1 = Data format Intel | $00_{\text {h }}$ |
| $2 . . .6$ | $00_{\text {h }}$ (fix) | 00 h |

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.
Manufacturer-specific data: You always obtain the data record DS 01h of the diagnostic data of a module.

Channel-specific data: You always obtain the data record DS 00h 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.

## 12.4

## 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 | 2 | $I \& M 0: 0020_{h}$ <br> $I \& M 1: 0021_{h}$ <br> $I \& M 2: 0022_{h}$ <br> $I \& M 3: 0023_{h}$ |
| - BlockType |  | $I \& M 0: 0038_{h}$ <br> $I \& M 1: 0038_{h}$ <br> $I \& M 2: 0012_{h}$ <br> $I \& M 3: 0038_{h}$ |
| - BlockLength | 2 | $01_{h}$ |
| - BlockVersionHigh | 1 | $00_{h}$ |
| - BlockVersionLow | 4 | $I \& M 0 /$ Index AFF2 $: 54_{h}$ <br> I\&M1 / Index AFF1 $: 54_{h}$ <br> I\&M2 / Index AFF2 <br> (see the following table) |

## I\&M data for PROFINET-IO

| Identification data | Access | Lenze | Description |
| :---: | :---: | :---: | :---: |
| Identification data 0: (Index AFFOh) |  |  |  |
| VendorIDHigh | Read (1 byte) | 02h | Name of the |
| VendorIDLow | Read (1 byte) | 2Bh | manufact |
| 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 <br> - SWRevisionP̈refix <br> - IM_SWRevision_Functional_Enhancement <br> - IM_SWRevision_Bug_Fix <br> - IM_SWRevision_Internal_Change | Read (1 byte) (1 byte) (1 byte) (1 byte) | $\begin{aligned} & \text { V, R, P, U, T } \\ & \text { ooh ... FFh } \\ & \text { OOh ... FFh } \\ & \text { 00h ... FFh } \end{aligned}$ | Firmware version |
| IM_REVISION_COUNTER | Read (2 bytes) | 0000h | for internal use |
| IM_PROFILE_ID | Read (2 bytes) | 0000h | for internal use |
| IM_PROFILE_SPECIFIC_TYPE | Read (2 bytes) | 0005h | for internal use |
| IM_VERSION <br> - IM_Version_Major <br> - IM_Version_Minor | Read (1 byte) (1 byte) | 0101h | Version of the I\&M data <br> (e.g. 0101h $=$ <br> version 1.1) |
| IM_SUPPORTED | Read (2 bytes) | 000Eh | I\&M1 ... I\&M3 are available |
| Maintenance data 1: (Index AFF1h) |  |  |  |
| 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 AFF2h) |  |  |  |
| IM_DATE | Read/write (16 bytes) | YYYY-MMDD HH:MM | Selection of an date |
| Maintenance data 3: (Index AFF3h) |  |  |  |
| IM_DESCRIPTOR | Read/write (54 bytes) | - | Selection of a comment |

## 12.5

## 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:
$0000_{h} \ldots$. $^{7 F F F_{h}}$ : Manufacturer-specific index numbers
$8000_{h} \ldots$ F7FF $_{h}$ : 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 |  |
| 0000 h | Read DS $00{ }_{\text {h }}$ diagnostic data |
| $0001_{\text {h }}$ | Read DS 01 ${ }_{\text {h }}$ diagnostic data |
| $007 \mathrm{D}_{\mathrm{h}}$ | Read all parameter data |
| 007Eh | Read DS $00{ }_{\text {h }}$ of the parameter data |
| $007 \mathrm{~F}_{\mathrm{h}}$ (only I/O compound modules) | Read DS $01_{h}$ of the parameter data |
| $0080 \mathrm{~h} .0090_{\mathrm{h}}$ (only I/O compound modules) | Read DS 80 h . DS 90 h of the parameter data |
| $\begin{aligned} & 8000_{h} / 8001_{h} / 800 A_{h} / 800 B_{h} / 800 C_{h} / \\ & 8010_{h} / 8011_{h} / 8012_{h} / 8013_{h} / 801 E_{h} / \\ & 802 A_{h} / 802 B_{h} / 802 C_{h} / 802 \mathrm{D}_{h} / 802 \mathrm{~F}_{\mathrm{h}} / \\ & 8030_{h} / 8031_{h} / 8050_{h} / 8051_{h} / 8052_{h} / \\ & 8053_{h} / 8054_{h} / 8060_{h} / 8061_{h} / 8062_{h} / \\ & 8070_{h} / 8080_{h} / 8090_{h} \end{aligned}$ | See PROFINET specification |
| 8028 h (only I/O compound modules) | Read input data of a sub-slot |
| 8029 h (only I/O compound modules) | Read output data of a sub-slot |
| $\mathrm{AFFO}_{\mathrm{h}}$ | Read I\&M 0 (serial no., name, SW/HW version) |
| AFF1 ${ }_{\text {h }}$ (only PROFINET bus coupler9) | Read I\&M 1 (designation and mounting place) |
| AFF2h (only PROFINET bus coupler9) | Read I\&M 2 (date of installation) |
| AFF3 ${ }_{h}$ (only PROFINET bus coupler9) | Read I\&M 3 (comment) |
|  | See PROFINET specification |
| Writable index numbers |  |
| 007D h | Write all parameter data |
| $007 \mathrm{E}_{\mathrm{h}}$ | Write DS $00{ }_{\mathrm{h}}$ of the parameter data |
| $007 \mathrm{~F}_{\mathrm{h}}$ (only I/O compound modules) | Write DS $01_{h}$ of the parameter data |
| $0080 \mathrm{~h} .0090_{\mathrm{h}}$ (only I/O compound modules) | Write DS 80 h . DS $90_{\mathrm{h}}$ of the parameter data |
| AFF1 ${ }_{\text {h }}$ (only PROFINET bus coupler9) | Write I\&M 1 (designation and mounting place) |
| AFF2h (only PROFINET bus coupler9) | Write I\&M 2 (date of installation) |
| AFF3h (only PROFINET bus coupler9) | Write I\&M 3 (comment) |

## 12.6

Setting the parameters of analog $1 / 0$
12.6.1 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 <br> No. |  | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- |
| 128 | 0 | Function channel 1 | $16\left(10_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}}$ <br> $32\left(20_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}}$ <br> $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated | $10_{\mathrm{h}}$ |
| 129 | 0 | Function channel 2 | $10_{h}$ |  |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 27648 \\ & \mathrm{D}=27648^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | $13824$ | 3600 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \mathrm{~V}}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{h}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 02h) |
| 6 | Number of channels of a module (here $02_{\text {h }}$ :) |
| 7 | Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $10 . .15$ | 0 (fixed) |

## Parameter data

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

| Data set <br> No. |  | Byte | Name | Description/value |
| :--- | :--- | :--- | :--- | :--- |
| 128 | 0 | Function channel 1 |  | Lenze |
| 129 | 0 | Function channel 2 | $16\left(10_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}}$ | $10_{\mathrm{h}}$ |
| 130 | 0 | Function channel 3 | $32\left(20_{h}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\mathrm{dec}}$ | $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & \mathrm{D}=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\underset{\left(20_{h}\right)}{0 \ldots 10 \mathrm{~V}}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{h}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here 04 h :) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $12 . .15$ | 0 (fixed) |

### 12.6.3

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 |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| 128 | 0 | Function channel 1 | $\begin{aligned} & 48\left(30_{\mathrm{h}}\right): 4 \ldots 20 \mathrm{~mA} / 0 . . .27648_{\mathrm{dec}} \\ & 64\left(40_{\mathrm{h}}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 1^{2} . . .2384_{\mathrm{dec}} \end{aligned}$ | $31_{\text {h }}$ |
| 129 | 0 | Function channel 2 | 49 (31h): 0 ... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots{16384_{\text {dec }}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D * 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{\mathrm{h}}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 16 / 27648+4 \\ & D=27648 *(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{h}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 12 | 8192 | 2000 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{h}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 02h) |
| 6 | Number of channels of a module (here $02_{\text {h }}$ :) |
| 7 | Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $10 . .15$ | 0 (fixed) |

## Parameter data

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

| Data set <br> No. |  | Byte | Name | Description/value |
| :--- | :--- | :--- | :--- | :--- |
| 128 | 0 | Function channel 1 | $48\left(30_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}}$ | Lenze |
| 129 | 0 | Function channel 2 | $64\left(40_{h}\right): 4 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ | $31_{\mathrm{h}}$ |
| 130 | 0 | Function channel 3 | $49\left(31_{h}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}}$ | $65\left(41_{\mathrm{h}}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ |
| 131 | 0 | Function channel 4 | $255\left(\mathrm{FF}_{\mathrm{h}}\right):$ channel deactivated | $31_{\mathrm{h}}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{h}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{\mathrm{h}}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} / / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\underset{\left(40_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 12 | 8192 | 2000 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{\mathrm{b}}$ : Analog module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{h}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here 04h:) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $12 . .15$ | 0 (fixed) |

## 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 $)$ <br> Bit 1: Channel 2 ( $0=$ deactivated; $1=$ activated) <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |
| 1 | 0 | Interference frequency suppression | Bit 1, 0: Channel 1 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bit 3, 2: Channel 2 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 128 | 0 | Function channel 1 | $\begin{aligned} & 12\left(12_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots . .27648_{\mathrm{dec}} \\ & 34\left(22_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-16384 \ldots 16384_{\mathrm{dec}} \\ & 16\left(1 \mathrm{~h}_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\text {dec }} \\ & 32\left(20_{\mathrm{h}}\right): 0 . .10 \mathrm{~V} / 0 . . .16384_{\mathrm{dec}} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
|  | 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. <br> $7 \mathrm{FFF}_{\mathrm{h}}$ : Limit value alarm deactivated | 7FFF ${ }_{\text {h }}$ |
|  | 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. $8000_{h}$ : Limit value alarm deactivated | $8000{ }_{\text {h }}$ |
| 129 | 0 | Function channel 2 | $\begin{aligned} & 12\left(12_{h}\right):-10 \ldots 10 \mathrm{~V} /-27648 \ldots . .27648_{\text {dec }} \\ & 34\left(22_{\mathrm{h}}\right):-10 \ldots 10 \mathrm{~V} /-16384 \ldots 16384_{\text {dec }} \\ & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\text {dec }} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 . . .16384_{\text {dec }} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $20_{\text {h }}$ |
|  | 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. <br> 7FFF $\mathrm{F}_{\mathrm{h}}$ : Limit value alarm deactivated | 7FFF ${ }_{\text {h }}$ |
|  | 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. $8000_{h}$ : Limit value alarm deactivated | $8000_{\text {h }}$ |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(12_{\mathrm{h}}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 27648 \\ & \mathrm{D}=27648^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 |  | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -13824 | CA00 |  |  |
|  | -10 | -27648 | 9400 |  |  |
|  | -11.76 | -32512 | 8100 | Underflow |  |
| $\begin{gathered} -10 \ldots+10 \mathrm{~V} \\ \left(22_{\mathrm{h}}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 | Nominal range |  |
|  | 5 | 8192 | 2000 |  |  |
|  | 0 | 0 | $0000$ |  |  |
|  | -5 | -8192 | E000 |  |  |
|  | -10 | -16384 | C000 |  |  |
|  | -12.5 | -20480 | B000 | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & D=27648 * U / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -1.76 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{\mathrm{b}}$ : Analog module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{h}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here 04h:) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1:0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $12 . .15$ | 0 (fixed) |

### 12.6.6

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 |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| 0 |  | Diagnostics | Bits 0 ... 5: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
|  |  | Reserved | 0 |  |
|  |  | Limit value monitoring | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated $)$ <br> Bit 1: Channel 2 ( $0=$ deactivated; $1=$ activated) <br> Bits 7 ... 2: Reserved | $00_{\text {h }}$ |
| 1 |  | Interference frequency suppression | Bit 1, 0: Channel 1 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bit 3, 2: Channel 2 <br> 00: Deactivated <br> 01: 60 Hz <br> 10: 50 Hz <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
| 128 |  | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 27648_{\mathrm{dec}}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $41_{\text {h }}$ |
|  |  | 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. <br> 7FFF ${ }_{h}$ : Limit value alarm deactivated | $7 \mathrm{FFF}_{\mathrm{h}}$ |
|  |  | 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. $8000_{h}$ : Limit value alarm deactivated | $8000_{h}$ |
| 129 |  | Function channel 2 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... $27648_{\text {dec }}$ $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 27648_{\text {dec }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $41_{\text {h }}$ |
|  |  | 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. <br> $7 F F F_{h}$ : Limit value alarm deactivated | 7FFF ${ }_{\text {h }}$ |
|  |  | 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. $8000_{h}$ : Limit value alarm deactivated | $8000_{h}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 10 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -3.52 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384 * I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 10 | 8192 | 2000 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | -4 | -3277 | F333 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 16 / 27648+4 \\ & D=27648 *(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 12 | 13824 | 3600 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 1,19 | -4864 | ED00 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{h}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 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

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $71_{\mathrm{h}}$ : Analog input Bit 7: 0 (fix) |
| 5 | Number of diagnostic bits output by the module per channel (here $088_{h}$ ) |
| 6 | Number of channels of a module (here 04h:) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 5 ... 1: 0 (fixed) <br> Bit 6: Measuring range not reached <br> Bit 7: Measuring range exceeded |
| $12 . .15$ | 0 (fixed) |

12.6.7 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 <br> No. |  | Byte | Name | Description/value |
| :--- | :--- | :--- | :--- | :--- |$\quad$ Lenze

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & \mathrm{D}=27648^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{U}=\mathrm{D}^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 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 anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $73_{h}$ : analog output Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 02h) |
| 6 | Number of channels of a module (here $02_{\text {h }}$ :) |
| 7 | Bit 0: Channel error, channel 1 Bit 1: Channel error, channel 2 Bit 7 ... 2: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .$. 1:0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4: 0 (fixed) |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 2... 1: 0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4: 0 (fixed) |
| $10 . . .15$ | 0 (fixed) |

## 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 <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 128 | 0 | Function channel 1 | $\begin{aligned} & 16\left(10_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 27648_{\mathrm{dec}} \\ & 32\left(20_{\mathrm{h}}\right): 0 \ldots 10 \mathrm{~V} / 0 \ldots 16384_{\text {dec }} \\ & 255\left(\mathrm{FF}_{\mathrm{h}}\right): \text { channel deactivated } \end{aligned}$ | $10_{\text {h }}$ |
| 129 | 0 | Function channel 2 |  | $10_{\text {h }}$ |
| 130 | 0 | Function channel 3 |  | $10_{\text {h }}$ |
| 131 | 0 | Function channel 4 |  | $10_{\text {h }}$ |

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.) | $\begin{gathered} \text { Voltage (U) } \\ {[\mathrm{V}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 10 \text { V } \\ \left(10_{h}\right) \end{gathered}$ | 11.76 | 32511 | 7EFF | Overflow | $\begin{aligned} & U=D^{*} 10 / 27648 \\ & \mathrm{D}=27648^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 27648 | 6C00 | Nominal range |  |
|  | 5 | 13824 | 3600 |  |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 V |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 10 \mathrm{~V} \\ \left(20_{h}\right) \end{gathered}$ | 12.5 | 20480 | 5000 | Overflow | $\begin{aligned} & U=D^{*} 10 / 16384 \\ & \mathrm{D}=16384^{*} \mathrm{U} / 10 \end{aligned}$ |
|  | 10 | 16384 | 4000 |  |  |
|  | 5 | 8192 | 2000 | Nominal range |  |
|  | 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 anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $73_{h}$ : analog output Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here 04h:) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit 2... 1:0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4:0 (fixed) |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit 2... 1:0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4:0 (fixed) |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .1$ 1: 0 (fixed) <br> Bit 3: Short circuit after $M$ <br> Bit 7... 4:0 (fixed) |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit 2... 1: 0 (fixed) <br> Bit 3: Short circuit after M <br> Bit 7... 4: 0 (fixed) |
| $12 . .15$ | 0 (fixed) |

### 12.6.9

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 | Short-circuit detection | Bit 0: Channel 1 ( $0=$ deactivated; $1=$ activated $)$ <br> Bit 1: Channel 2 <br> Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
| 128 | 0 | Function channel 1 |  | $31_{\text {h }}$ |
| 129 | 0 | Function channel 2 | $49\left(31_{\mathrm{h}}\right): 0$... $20 \mathrm{~mA} / 0 . .{27648_{\text {dec }}}^{\text {2 }}$ $65\left(41_{h}\right): 0$... $20 \mathrm{~mA} / 0$... 16384 ${ }_{\mathrm{dec}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{\mathrm{h}}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | $0000$ | Nominalrange |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} I / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | $8192$ | $2000$ | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(30_{\mathrm{h}}\right) \end{gathered}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 16 / 27648+4 \\ & D=27648 *(I-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 | Nominal range |  |
|  | 12 | 13824 | 3600 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{h}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | 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)

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{\mathrm{b}}$ : Analog module Bit 4: Channel information available <br> Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $73_{h}$ : analog output Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 02h) |
| 6 | Number of channels of a module (here $02_{h}$ :) |
| 7 | Bit 0: Channel error, channel 1 <br> Bit 1: Channel error, channel 2 <br> Bit 7... 2: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .$. 1:0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .$. 1:0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| $10 . .15$ | 0 (fixed) |

## 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 <br> Bit 1: Channel 2 <br> Bit 2: Channel 3 <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
| 128 | 0 | Function channel 1 | $48\left(30_{h}\right): 4$... $20 \mathrm{~mA} / 0$... ${27648_{\text {dec }}}$ $64\left(40_{\mathrm{h}}\right): 4$... $20 \mathrm{~mA} / 0$... $16384_{\mathrm{dec}}$ $49\left(31_{h}\right): 0$... $20 \mathrm{~mA} / 0 . . .27648_{\text {dec }}$ $65\left(41_{h}\right): 0 \ldots 20 \mathrm{~mA} / 0 \ldots 16384_{\mathrm{dec}}$ $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $31_{\text {h }}$ |
| 129 | 0 | Function channel 2 |  | $31_{\text {h }}$ |
| 130 | 0 | Function channel 3 |  | $31_{\text {h }}$ |
| 131 | 0 | Function channel 4 |  | $31_{h}$ |

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.) | $\begin{gathered} \text { Current (I) } \\ {[\mathrm{mA}]} \end{gathered}$ | Decimal (D) | Hex | Range | Conversion |
| :---: | :---: | :---: | :---: | :---: | :---: |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(31_{h}\right) \end{gathered}$ | 23.52 | 32511 | 7EFF | Overflow | $\begin{aligned} & I=D^{*} 20 / 27648 \\ & D=27648 * I / 20 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 10 | 13824 | 3600 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\begin{gathered} 0 \ldots 20 \mathrm{~mA} \\ \left(41_{h}\right) \end{gathered}$ | 25 | 20480 | 5000 | Overflow | $\begin{aligned} & I=D^{*} 20 / 16384 \\ & D=16384^{*} \mathrm{I} / 20 \end{aligned}$ |
|  | 20 | 16384 | 4000 |  |  |
|  | 10 | 8192 | 2000 | Nominal range |  |
|  | 0 | 0 | 0000 |  |  |
|  | Not possible, is limited to 0 mA |  |  | Underflow |  |
| $\underset{\left(30_{h}\right)}{4 \ldots 20 \mathrm{~mA}}$ | 22.81 | 32511 | 7EFF | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 27648+4 \\ & \mathrm{D}=27648^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 27648 | 6C00 |  |  |
|  | 12 | 13824 | 3600 | Nominal range |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -6912 | E500 | Underflow |  |
| $\begin{gathered} 4 \ldots 20 \mathrm{~mA} \\ \left(40_{\mathrm{h}}\right) \end{gathered}$ | 24 | 20480 | 5000 | Overflow | $\begin{aligned} & \mathrm{I}=\mathrm{D}^{*} 16 / 16384+4 \\ & \mathrm{D}=16384^{*}(\mathrm{I}-4) / 16 \end{aligned}$ |
|  | 20 | 16384 | 4000 | Nominal range |  |
|  | 12 | 8192 | 2000 |  |  |
|  | 4 | 0 | 0000 |  |  |
|  | 0 | -4096 | F000 | 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)

By the use of the SFB 52 you can access the diagnostic data of the module anytime. Data set 1 has the following structure:

| Byte | Bit 7 ... 0 |
| :---: | :---: |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 6: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bits 3 ... 0: Module class, $0101_{b}$ : Analog module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, $73_{h}$ : analog output Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here 04h:) |
| 7 | Bit 0: Channel error channel 1 <br> Bit 1: Channel error channel 2 <br> Bit 2: Channel error channel 3 <br> Bit 3: Channel error channel 4 <br> Bit 7 ... 4: 0 (fixed) |
| 8 | Channel-specific errors: channel 1: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .1$ 1: 0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| 9 | Channel-specific errors: channel 2: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .1$ 1: 0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| 10 | Channel-specific errors: channel 3: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . .1$ 1: 0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| 11 | Channel-specific errors: channel 4: <br> Bit 0: Configuration/parameterisation error <br> Bit $2 . . .1$ : 0 (fixed) <br> Bit 3: Open circuit <br> Bit 7... 4:0 (fixed) |
| $12 . .15$ | 0 (fixed) |

### 12.7 Parameterising the temperature measurement

### 12.7.1 Four (two) analog inputs for resistance tests - EPM-S404

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. Bits 0 ... 5: Reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved | $00_{\text {h }}$ |
|  | 1 | Wire-breakage detection | Bit 0: Channel 1 ( $0=$ inhibited; 1 = enabled) <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
|  | 2 | Limit value monitoring | Bit 0: Channel 1 ( 0 = inhibited; 1 = enabled) <br> Bit 3: Channel 4 <br> Bits 4 ... 7: Reserved | $00_{\text {h }}$ |
|  | 3 | Reserved |  |  |
| 1 | 0 | Temperature system | Bit 0, 1: $00_{b}={ }^{\circ}{ }^{\circ} ; 0_{b}={ }^{\circ} \mathrm{F} ; 10_{\mathrm{b}}=\mathrm{K}$ Bits 2 ... 7: Reserved | $00_{\text {h }}$ |
|  | 1 | Interference frequency suppression | Bit 0, 1: $01_{\mathrm{b}}=60 \mathrm{~Hz} ; 10_{\mathrm{b}}=50 \mathrm{~Hz}$ Bits 2 ... 7: Reserved | 02h |
| Channel 1 |  |  |  |  |
| 128 | 0 | Function channel 1 | $80\left(50_{h}\right)$... $162\left(A 2_{h}\right)$ : see "measuring range" $255\left(\mathrm{FF}_{\mathrm{h}}\right)$ : channel deactivated | $50_{\text {h }}$ |
|  | 1 | Conversion time channel 1 | You can set the transformer speed as a function of the interference frequency suppression (see $0 \times 3105 / x$ ) for each channel. <br> $0\left(00_{h}\right)$ : At $50 \mathrm{~Hz}: 324.1 \mathrm{~ms} /$ channel 16 bits; at 60 Hz : $270.5 \mathrm{~ms} / \mathrm{channel} 16$ bits <br> 1 ( $01_{\mathrm{h}}$ ): At $50 \mathrm{~Hz}: 164.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 137.2 \mathrm{~ms} / c h a n n e l ~ 16$ bits <br> 2 (02h): At $50 \mathrm{~Hz}: 84.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 70.5 \mathrm{~ms} /$ channel 16 bits $3\left(03_{h}\right)$ : at $50 \mathrm{~Hz}: 44.1 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 37.2 \mathrm{~ms} /$ channel 16 bits 4 ( 04 h ): At $50 \mathrm{~Hz}: 24.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 20.5 \mathrm{~ms} /$ channel 16 bits 5 (05h): At $50 \mathrm{~Hz}: 14.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 12.2 \mathrm{~ms} /$ channel 16 bits 6 ( $06_{\text {h }}$ ): At $50 \mathrm{~Hz}: 9.2 \mathrm{~ms} /$ channel 16 bits; at $60 \mathrm{~Hz}: 8.0 \mathrm{~ms} / \mathrm{channel} 16$ bits $7\left(07_{\mathrm{h}}\right)$ : At 50 Hz : $6.6 \mathrm{~ms} /$ channel 15 bits; at $60 \mathrm{~Hz}: 5.9 \mathrm{~ms} / c h a n n e l ~ 15$ bits $8\left(08_{h}\right)$ : At $50 \mathrm{~Hz}: 4.2 \mathrm{~ms} /$ channel 13 bits; at $60 \mathrm{~Hz}: 3.8 \mathrm{~ms} /$ channel 13 bits | $00_{\text {h }}$ |
|  | 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 $7 \mathrm{FFF}_{\mathrm{h}}$ for the | $7 \mathrm{FFF}_{\mathrm{h}}$ |
|  | 4,5 | Lower limit value channel 1 | upper or $8000_{h}$ 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. | $8000_{\text {h }}$ |

Parameterising the temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
|  |  |  |  |  |
| Channel 2 |  |  |  |  |
| 129 | 0 | Function channel 2 | See channel 1 | $50_{\text {h }}$ |
|  | 1 | Conversion time channel 2 | See channel 1 | $00_{\text {h }}$ |
|  | 2, 3 | Upper limit value channel 2 | See channel 1 | 7FFF ${ }_{\text {h }}$ |
|  | 4, 5 | Lower limit value channel 2 (HIGH byte) |  | $8000_{\text {h }}$ |
| Channel 3 (for two-wire conductor connections only) |  |  |  |  |
| 130 | 0 | Function channel 3 | See channel 1 | $50_{\text {h }}$ |
|  | 1 | Conversion time channel 3 | See channel 1 | $00_{\text {h }}$ |
|  | 2, 3 | Upper limit value channel 3 | See channel 1 | $7 \mathrm{FFF}_{\mathrm{h}}$ |
|  | 4, 5 | Lower limit value channel 3 |  | $8000_{\text {h }}$ |
| Channel 4 (for two-wire conductor connections only) |  |  |  |  |
| 131 | 0 | Function channel 4 | See channel 1 | $50_{\text {h }}$ |
|  | 1 | Conversion time channel 4 | See channel 1 | $00_{\text {h }}$ |
|  | 2, 3 | Upper limit value channel 4 | See channel 1 | 7FFF ${ }_{\text {h }}$ |
|  | 4, 5 | Lower limit value channel 4 |  | $8000{ }_{\text {h }}$ |

Measuring range

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 2-wire: PT100 } \\ \left(50_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 2-wire: PT1000 } \\ \left(51_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 2-wire: NI100 } \\ \left(52_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 2-wire: NI1000 (53h) | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | -1050 dec | Underflow |
| $\begin{gathered} \text { 3-wire: PT100 } \\ \left(58_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| 3-wire: PT1000 (59h) | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 3-wire: NI100 } \\ \left(5 A_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |


| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| 3-wire: NI1000 (5Bh) | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 4-wire: PT100 } \\ \left(60_{h}\right) \end{gathered}$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| 4-wire: PT1000$\left(61_{h}\right)$ | $+1000{ }^{\circ} \mathrm{C}$ | $+10000_{\text {dec }}$ | Overflow |
|  | $-200 \ldots+850^{\circ} \mathrm{C}$ | $-2000 \ldots+8500_{\text {dec }}$ | Nominal range |
|  | $-243{ }^{\circ} \mathrm{C}$ | $-2430_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 4-wire: NI100 } \\ \left(62_{h}\right) \end{gathered}$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 \ldots+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| 4-wire: NI1000 $\left(63_{h}\right)$ | $+295{ }^{\circ} \mathrm{C}$ | $+2950{ }_{\text {dec }}$ | Overflow |
|  | $-60 \ldots+250{ }^{\circ} \mathrm{C}$ | $-600 . . .+2500_{\text {dec }}$ | Nominal range |
|  | $-105{ }^{\circ} \mathrm{C}$ | $-1050{ }_{\text {dec }}$ | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(70_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .60 \Omega$ | $0 \ldots 32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 600 \Omega \\ \left(71_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | 0 ... 32767 dec | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(72_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(78_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(79_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | 0 ... 32767dec | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(7 A_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(80_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(81_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .32767_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(82_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | 0 ... $3000 \Omega$ | 0 ... 32767 | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(90_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |

Parameterising the temperature measurement Four (two) analog inputs for resistance tests - EPM-S404

| Measuring range (Fct. no.) | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| $\begin{gathered} \text { 2-wire: } 0 \ldots . .600 \Omega \\ \left(91_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .600 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(92_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 \ldots 30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(98_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(99_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 \ldots 6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(9 A_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 \ldots 30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(\mathrm{AO}_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 \ldots 6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 600 \Omega \\ \left(\mathrm{~A} 1_{h}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 \ldots 6000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{~A} 2_{\mathrm{h}}\right) \end{gathered}$ | - | - | Overflow |
|  | $0 . . .3000 \Omega$ | $0 \ldots 30000_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 60 \Omega \\ \left(\mathrm{DO}_{\mathrm{h}}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots . .600 \Omega \\ \left(D 1_{h}\right) \end{gathered}$ | $705.5 \Omega$ | 32511 dec | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 2-wire: } 0 \ldots 3000 \Omega \\ \left(\mathrm{D} 2_{\mathrm{h}}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | 0 ... $3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 60 \Omega \\ \left(D 8_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . .60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots . .600 \Omega \\ \left(D 9_{h}\right) \end{gathered}$ | $705.5 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 \ldots 600 \Omega$ | $0 \ldots 27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 3-wire: } 0 \ldots 3000 \Omega \\ \left(D A_{h}\right) \end{gathered}$ | $3528 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . .3000 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |
| $\begin{gathered} \text { 4-wire: } 0 \ldots 60 \Omega \\ \left(E O_{h}\right) \end{gathered}$ | $70.55 \Omega$ | $32511_{\text {dec }}$ | Overflow |
|  | $0 . . .60 \Omega$ | $0 . . .27648_{\text {dec }}$ | Nominal range |
|  | - | - | Underflow |


| Measuring range | Measured value | Signal range | Range |
| :---: | :---: | :---: | :---: |
| (Fct. no.) |  |  |  |

## Diagnostics and alarm

| Trigger | Process alarm | Diagnostic alarm | Parameterisable |
| :--- | :---: | :---: | :---: |
| Configuration/parameterisation error | - | 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 | C | X |
| Process alarm lost | - | X | - |

## Process alarm

A process alarm causes a call of the $O B 40$. Within the $O B 40$ 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 |
| :---: | :---: |
| 8 | Bit 0: Limit value exceeded channel 1 <br> Bit 1: Limit value exceeded channel 2 <br> Bit 2: Limit value exceeded channel 3 <br> Bit 3: Limit value exceeded channel 4 <br> Bit 7... 4: 0 (fixed) |
| 9 | Bit 0: Limit value not reached channel 1 <br> Bit 1: Limit value not reached channel 2 <br> Bit 2: Limit value not reached, channel 3 <br> Bit 3: Limit value not reached, channel 4 <br> Bit 7... 4: 0 (fixed) |
| 10 ... 11 | Ticker value at the time of the alarm After mains connection, a timer ( $\mu \mathrm{s}$ ticker) is started which after $65535 \mu \mathrm{~s}$ starts with 0 again. |

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00 h ). 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 processingincoming 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 ${ }_{\text {incoming }}$ 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 incoming and diagnostic alarm ${ }_{\text {outgoing }}$ are not buffered and are lost. During this time (first diagnostic alarm 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 incoming/outgoing.
Example:


## Diagnostic alarm processing:

Using 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 ca no longer be clearly assigned to the last diagnostic alarm.
Data set 1 is structured as follows:

| Data set 1, diagnostics ${ }_{\text {incoming }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $\mathbf{1 0 0 0}_{b}$ : function module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here $08{ }_{\text {h }}$ ) |
| 6 | Number of channels of a module (here $01{ }_{\text {h }}$ ) |
| 7 | Bit 0: Error in channel group 0 (counter) Bit 7... 1: 0 (fixed) |
| 8 | Diagnostic alarm due to process alarm lost to ... <br> Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7... 5: 0 (fixed) |
| $9 . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message outgoing $^{\text {is effected }}$ |

### 12.7.2 Two analog inputs for thermocouple measurement - EPM-S405

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


| Data set <br> No. <br> Channel 2 |  | Byame | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- |
| 129 | 0 | Function channel 2 | See channel 1 | $\mathrm{C1}_{\mathrm{h}}$ |
| 1 | Conversion time <br> channel 2 | See channel 1 | $02_{\mathrm{h}}$ |  |
| 2,3 | Upper limit value <br> channel 2 | See channel 1 | $7 \mathrm{FFF}_{\mathrm{h}}$ |  |
|  | 3,4 | Lower limit value <br> channel 2 |  | $8000_{\mathrm{h}}$ |

Measuring range

| Measuring range (Fct. no.) | Measured value |  |  | Range |
| :---: | :---: | :---: | :---: | :---: |
|  | [ $\left.{ }^{\circ} \mathrm{C}\right]$ | [ ${ }^{\circ} \mathrm{F}$ ] | [K] |  |
| ```Type J: -210 ... +1200 % C -346 ... 2192 % F 63.2 ... 1473.2 K (B0. ext. comp. 0 % C) (COh: int. comp. 0 }\mp@subsup{}{}{\circ}\textrm{C}\mathrm{ )``` | +14500 | 26420 | 17232 | Overflow |
|  | $-2100 \ldots+12000$ | -3460 ... +21920 | 632 ... 14732 | Nominal range |
|  | - | - | - | Underflow |
| ```Type K: -210 ... +1372 ' C -454 ... 2501.6 % F 0 ... 1645.2 K (B1 h: ext. comp. 0 % C) (C1 h: int. comp. 0 }\mp@subsup{}{}{\circ}\textrm{C}\mathrm{ )``` | +16220 | 29516 | 18952 | Overflow |
|  | $-2700 \ldots+13720$ | -4540 ... 25016 | 0 ... 16452 | Nominal range |
|  | - | - | - | Underflow |
| ```Type N: -270 ... +1300 %}\textrm{C -454 ... 2372 % F 0 ... 1573.2 K (B2h: ext. comp. 0 % C) (C2h: int. comp. 0 % C)``` | +15500 | 28220 | 18232 | Overflow |
|  | $-2700 \ldots+13000$ | -4540 ... 23720 | 0 ... 15732 | Nominal range |
|  | - | - | - | Underflow |
| ```Type R: -50 ... +1769 钅 -58 ... 3216.2 ' F 223.2 ... 2042.2 K (B3h. ext. comp.0 % C) (C3h: int. comp. 0 }\mp@subsup{}{}{\circ}\textrm{C}\mathrm{ )``` | +20190 | 32766 | 22922 | Overflow |
|  | $-500 \ldots+17690$ | -580 ... 32162 | 2232 ... 20422 | Nominal range |
|  | -1700 | -2740 | 1032 | Underflow |
| ```Type S: -50 ... +1769 钅 -58 ... 3216.2 ' F 223.2 ... 2042.2 K (B4h: ext. comp. 0 % C) (C4h: int. comp. 0 % C)``` | +20190 | 32766 | 22922 | Overflow |
|  | $-500 \ldots+17690$ | -580 ... 32162 | 2232 ... 20422 | Nominal range |
|  | -1700 | -2740 | 1032 | Underflow |
| Type T: $\begin{gathered} -270 \ldots+440^{\circ} \mathrm{C} \\ -454 \ldots . .752^{\circ} \mathrm{F} \end{gathered}$ $3.2 \text {... } 673.2 \text { K }$ <br> (B2h: ext. comp. $0^{\circ} \mathrm{C}$ ) <br> ( $\mathrm{C} 2_{\mathrm{h}}$ : int. comp. $0^{\circ} \mathrm{C}$ ) | +5400 | 10040 | 8132 | Overflow |
|  | $-2700 \ldots+4000$ | -4540 ... 7520 | 32 ... 6732 | Nominal range |
|  | - | - | - | Underflow |
| Type B: $0 \ldots+1820^{\circ} \mathrm{C}$ $32 . . .2786 .5^{\circ} \mathrm{F}$ <br> 273.2 ... 2093.2 K <br> (B6h: ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C6h : int. comp. $0^{\circ} \mathrm{C}$ ) | +20700 | 32766 | 23432 | Overflow |
|  | 0 ... +18200 | 320 ... 27865 | 2732 ... 20932 | Nominal range |
|  | -1200 | -1840 | 1532 | Underflow |


| Measuring range | Measured value |  |  | Range |
| :---: | :---: | :---: | :---: | :---: |
| (Fct. no.) | [ $\left.{ }^{\circ} \mathrm{C}\right]$ | [ ${ }^{\circ} \mathrm{F}$ ] | [K] |  |
| Type C: $0 \ldots+2315^{\circ} \mathrm{C}$ | +25000 | 32766 | 23432 | Overflow |
| $\begin{gathered} 32 \ldots 2786.5^{\circ} \mathrm{F} \\ 273.2 \ldots 2093.2 \mathrm{~K} \end{gathered}$ | $0 \ldots+23150$ | 320 ... 27865 | 2732 ... 20932 | Nominal range |
| (B7 ${ }_{h}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C7 ${ }_{h}$ : int. comp. $0{ }^{\circ} \mathrm{C}$ ) | -1200 | -1840 | 1532 | Underflow |
| Type E: $-270 \ldots+1000^{\circ} \mathrm{C}$ | +12000 | 21920 | 14732 | Overflow |
| $\begin{gathered} -454 \ldots 1832^{\circ} \mathrm{F} \\ 0 \ldots 1273.2 \mathrm{~K} \end{gathered}$ | -2700 ... +10000 | -4540 ... 18320 | 0 ... 12732 | Nominal range |
| (B8 ${ }_{h}$ : ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C8 ${ }_{h}$ : int. comp. $0{ }^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |
| Type L: $-200 \ldots+900^{\circ} \mathrm{C}$ | +11500 | 21020 | 14232 | Overflow |
| $\begin{gathered} -328 \ldots 1652^{\circ} \mathrm{F} \\ 73.2 \ldots 1173.2 \mathrm{~K} \end{gathered}$ | -2000 ... +9000 | -3280 ... 16520 | 732 ... 11732 | Nominal range |
| (B9h: ext. comp. $0^{\circ} \mathrm{C}$ ) <br> (C9 ${ }_{h}$ : int. comp. $0{ }^{\circ} \mathrm{C}$ ) | - | - | - | Underflow |

Diagnostics and alarm

| Trigger | Process alarm | Diagnostic alarm | Parameterisable |
| :--- | :---: | :---: | :---: |
| Configuration/parameterisation error | - | 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 ... $\mathbf{0}$ |
| :---: | :--- |
| 8 | Bit 0: Limit value exceeded channel 1 <br> Bit 1: Limit value exceeded channel 2 <br> Bit 7 ... 2: 0 (fixed) |
| 9 | Bit 0: Limit value not reached, channel 1 <br> Bit 1: Limit value not reached, channel 2 <br> Bit 7 ... 2: 0 (fixed) |
| $10 \ldots 11$ | Ticker value at the time of the alarm <br> After mains connection, a timer ( $\mu \mathrm{s}$ ticker) is started which after $65535 \mu$ starts with 0 again. |

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00h). 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 incoming 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 ${ }_{\text {incoming }}$ 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 are lost. During this time (first diagnostic alarm 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 incoming/outgoing.
Example:


## Diagnostic alarm processing:

Using 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 ca no longer be clearly assigned to the last diagnostic alarm.
Data set 1 is structured as follows:

| Data set 1, diagnostics incoming $^{\text {a }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $1000_{b}$ : function module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here 08 h ) |
| 6 | Number of channels of a module (here $01{ }_{h}$ ) |
| 7 | Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed) |
| 8 | Diagnostic alarm due to process alarm lost to ... <br> Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7... 5: 0 (fixed) |
| $9 . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message outgoing $^{\text {is effected }}$ |

## 12 <br> PROFINET communication

Parameterising the counter
One counter 32 bits, 24 V DC - EPM-S600

### 12.8 Parameterising the counter

12.8.1 One 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 |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> range. |
| Counting periodically |  | | Signal evaluation |
| :--- |
| Single rotary <br> transducer |
| Double rotary <br> transducer |
| Quadruple rotary <br> transducer |
| Connection to input "A/pulse" and "B/direction" |
| Direction |


| Additional functions | Description |
| :---: | :---: |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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. <br> - The I-gate is the AND logic operation of the software gate (SW gate) and the hardware gate (HW gate). <br> - The SW gate is controlled via your user program (status word in the output area). <br> - The HW gate is controlled via the digital gate input. <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

## Read data: 12 bytes

| Input area |  |  |
| :--- | :--- | :--- |
| Addr. | Access | Assignment |
| +0 | Double word | Count value |
| +4 | Double word | Latch value |
| +8 | Word | Status word (see the following table) |
| +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 ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu \mathrm{s}$ value together with the count value in the input area.

| EPM-S600 status word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | 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 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 for counter direction backwards |
| 8 | STS_C_UP | Status set for counter direction forwards |
| 9 | STS_CMP* | $\left.\begin{array}{l}\text { Comparator status is set if the comparison condition is met. } \\ \text { If the comparison is deactivated (counter mode byte } 1=000 \\ \text { no }\end{array}\right)$, the bit has |
| 10 | STS_END* | Status set if final value was reached |
| 11 | STS_OFLW* | Status set in the case of overflow |
| 12 | STS_UFLW* | Status set in the case of underflow |
| 13 | STS_ZP* | Status set in the case of zero crossing |
| 14 | STS_LTCH | Status of latch input |
| 15 | - | Reserved |

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


## Write data: 10 bytes

Output area

| Addr. | Access | Assignment |
| :--- | :--- | :--- |
| +0 | Double word | Comparison value |
| +4 | Double word | Set value |
| +8 | Word | Control word (see the following table) |

Comparison value: Here you can select a value which, by comparison with the current counter content, can impact the counter output or trigger a process alarm. The response of the output or 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.

| EPM-S600 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | Function |
| 0 | CTRL_SYNC_SET | Activates the reset mode |
| 1 | CTRL_DO_SET | Enables the digital output |
| 2 | SW_GATE_SET | Sets the software gate |
| 3 | - | Reserved |
| 4 | - | Reserved |
| 5 | COUNTERVAL_SET | Sets the counter temporarily to the value in the set value |
| 6 | RES_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | - | Reserved |
| 8 | CTRL_SYNC_RESET | Deactivates the reset mode |
| 9 | CTRL_DO_RESET | Inhibits the digital output |
| 10 | SW_GATE_RESET | Resets the software gate |
| 11 | - | Reserved |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $00_{\text {h }}$ | 0 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bits 5 ... 0 : reserved <br> Bit 6: diagnostic alarm ( $0=$ inhibited; $1=$ enabled) <br> Bit 7: Reserved | $00_{\text {h }}$ |
| $01_{\text {h }}$ | 0 | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | 02 h |
|  | 1 | Input frequency track B |  | $02_{\text {h }}$ |
|  | 2 | Input frequency latch |  | 02 h |
|  | 3 | Input frequency gate |  | $02_{\text {h }}$ |
|  | 4 | Input frequency reset |  | $00_{\text {h }}$ |
|  | 5 | Reserved |  |  |
| $80_{\text {h }}$ | 0 | Alarm response | Setting activates process alarm <br> Bit 0: Proc. alarm HW gate open <br> Bit 1: Proc. alarm HW gate closed <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bit 6: Proc. alarm latch value <br> Bit 7: Reserved | $80_{\text {h }}$ |


| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
|  | 1 | Counter function | Bit 5 ... 0 : <br> $000000_{b}=$ counting continuously <br> $000001_{b}=$ counting once, main counting direction <br> forwards <br> $000010_{b}=$ counting once, main counting direction backwards <br> $000100_{b}=$ counting once, no main counting direction <br> $001000_{b}=$ counting periodically, main counting <br> direction forwards <br> $010000_{b}=$ counting periodically, main counting direction backwards <br> $100000_{b}=$ counting periodically, no main counting direction <br> Bits 7 ... 6: Reserved | $40_{\text {h }}$ |
|  | 2 | Comparator | ```Bit 2 ... 0: output switches (... if condition is met) \(000_{b}=\) never \(001_{b}=\) count value \(\geq\) comparison value \(010_{b}=\) count value \(\leq\) comparison value \(100_{b}=\) count value \(=\) comparison value Bit 3: Invert counting direction track B \(0=\) no (do not invert) 1 = yes (invert) Bits 6 ... 4: Reset \(000_{b}=\) deactivated \(001_{b}=\) HIGH level \(011_{b}=\) rising edge \(101_{b}=\) rising edge, once Bit 7: Reserved``` | 00 h |
|  | 3 | Signal evaluation | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bit 6 ... 3: Hardware gate (HW gate) <br> $000_{b}=$ deactivated (counter starts by setting SW gate) <br> $001_{b}=$ activated (HIGH level at gate activates the HW <br> gate. Counter starts if HW and SW gate are set.) <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | 00 h |
| $81_{\text {h }}$ | 0 | Final value | Counting method: 0x310B: byte 3 (high byte) <br> 0x3110: byte 2 <br> 0x3111: byte 1 <br> $0 \times 3112$ : byte 0 (low byte) | 00 h |
|  | 1 | Loading value | Counting method: <br> 0x310B: byte 3 (high byte) <br> 0x310C: byte 2 <br> 0x310D: byte 1 <br> $0 \times 310 \mathrm{E}$ : byte 0 (low byte) | $00_{\text {h }}$ |


| Data set |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- |
| No. | Byte | Name | Description/value | Lenze |
| 2 | Hysteresis | The hysteresis for instance serves to avoid frequent <br> switching operations of the output and/or triggering of <br> the alarm when the count value is within the range of the <br> comparison value. For the hysteresis you can select a <br> range between 0 and 255. With the settings 0 and 1 the <br> hysteresis is switched off. <br> The hysteresis has an effect on the zero crossing, <br> comparison, overflow/underflow. | $00_{\mathrm{h}}$ |  |
| 3 | Pulse | The pulse duration indicates for how long the output is to <br> be set if the parameterised comparison criterion is <br> reached or exceeded. The pulse duration can be specified <br> in steps of 2.048 ms between 0 and 522.24 ms. If the pulse <br> duration is =0, the output is set until the comparison <br> condition is no longer met. | $00_{\mathrm{h}}$ |  |

## 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 $7 \ldots 0$ |
| :---: | :--- |
| 8 | Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7 ... 5: 0 (fixed) |
| 9 | State of the inputs at the time of the alarm <br> Bit 0: A/pulse <br> Bit 1: B/direction <br> Bit 2: Latch <br> Bit 3: Hardware gate <br> Bit 4: Reset <br> Bit 7 ... 5: 0 (fixed) |
| $10 \ldots 11$ | 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 ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu \mathrm{s}$ value together with the count value in the input area.

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00h). 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 incoming 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 are lost. During this time (first diagnostic alarm 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 incoming/outgoing.
Example:


## Diagnostic alarm processing:

Using 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 ca no longer be clearly assigned to the last diagnostic alarm.
Data set 1 is structured as follows:

| Data set 1, diagnostics incoming $^{\text {a }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $1000_{b}$ : function module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here 08 h ) |
| 6 | Number of channels of a module (here $01{ }_{h}$ ) |
| 7 | Bit 0: Error in channel group 0 (counter) Bit 7 ... 1: 0 (fixed) |
| 8 | Diagnostic alarm due to process alarm lost to ... <br> Bit 0: Hardware gate open <br> Bit 1: Hardware gate closed <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 4: Latch value <br> Bit 7... 5: 0 (fixed) |
| $9 . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message outgoing $^{\text {is effected }}$ |

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 |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> range. |
| Counting periodically | the |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input "A/pulse" and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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. <br> 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). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

## 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 (see following table) |
| +10 | Word | Counter 2: status word (see following table) |

Count value: Current counter content

| EPM-S601 | status word |  |
| :--- | :--- | :--- |
| Bit | Designation | 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_UP | Status set for counter direction backwards |
| 8 | STS_CMP* | Status set for counter direction forwards |
| 9 | STS_OFLW* | Comparator status is set if the comparison condition is met. <br> If the comparison is deactivated (counter mode byte $1=000_{\text {b }}$ ), the bit has <br> no function. |
| 10 | STS_UFLW* | Status set if final value was reached |
| 11 | STS_ZP* | Status set in the case of overflow |
| 12 | - | Status set in the case of underflow |
| 13 | Status set in the case of zero crossing |  |
| 14 | Reserved |  |
| 15 | Reserved |  |

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


## 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 (see following table) |
| +10 | Word | Counter 2: Control word (see following table) |

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 set $80_{h}$ for counter 0 and $82_{h}$ for counter 1 .

| EPM-S601 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | 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 | Sets the counter temporarily to the value in the set value |
| 6 | RES_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | - | Reserved |
| 8 | - | Reserved |
| 9 | CTRL_COMP_RESET | Inhibits comparison bit |
| 10 | SW_GATE_RESET | Resets the software gate |
| 11 | - | Reserved |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| 00 h | 0 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bits 5 ... 0: reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | 00 h |
| $01_{\text {h }}$ | 0 | Input frequency counter 1, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | $02^{\text {h }}$ |
|  | 1 | Input frequency counter 1, track B |  | 02 h |
|  | 2 | Input frequency counter 2, track A |  | $02_{\text {h }}$ |
|  | 3 | Input frequency counter 2, track B |  | $02_{h}$ |
| $80_{\text {h }}$ | 0 | Alarm response counter 1 | Setting activates process alarm <br> Bits 1 ... 0: reserved <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bits 7 ... 6: Reserved | 00 h |
|  | 1 | Counter function counter 1 | ```Bit 5 ... 0: 000000 000001 b = once: forwards 000010b = once: backwards 000100 b = once: no main counting direction 0010000b = periodically: forwards 010000}b= periodically: backward 100000}b= periodically: no main counting directio Bits 7 ... 6: Reserved``` | 00 h |
|  | 2 | Comparator counter 1 | Bits 2 ... 0: Comparison bit is set (... if condition is met) $000_{b}=$ never <br> $001_{b}=$ count value $\geq$ comparison value <br> $010_{b}=$ count value $\leq$ comparison value <br> $100_{b}=$ count value $=$ comparison value <br> Bit 3: Invert counting direction track B <br> 0 = no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | 00 h |
|  | 3 | Signal evaluation counter 1 | ```Bits 2 ... 0: Signal evaluation 000b}=\mathrm{ counter deactivated (the other parameter details for the counter are ignored) 001b}=\mathrm{ rotary transducer single (at A and B) 010b = rotary transducer double (at A and B) 011b}= rotary transducer quadruple (at A and B) 100b}=\mathrm{ direction (pulse at A and direction at B) Bits 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)``` | $00_{\text {h }}$ |


| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $81_{\text {h }}$ | 0... 3 | Set value counter 1 | Counting method: <br> 0x3111: byte 3 (high byte) <br> 0x3112: byte 2 <br> 0x3113: byte 1 <br> $0 \times 3114$ : byte 0 (low byte) | $00_{\text {h }}$ |
|  | 4... 7 | Final value counter 1 | Counting method: <br> 0x310D: byte 3 (high byte) <br> $0 \times 310 \mathrm{E}$ : byte 2 <br> 0x310F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
|  | 8... 11 | Loading value counter 1 | Counting method: <br> 0x3109: byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> $0 \times 310 \mathrm{C}$ : byte 0 (low byte) | $00_{\text {h }}$ |
|  | 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. <br> The hysteresis has an effect on the zero crossing, comparison, overflow/underflow. | $00_{\text {h }}$ |
|  | 13 | Reserved |  |  |
| $82_{\text {h }}$ | 0 | Alarm response counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 1 | Counter function counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 2 | Comparator counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 3 | Signal evaluation counter 2 | See counter 1 | $00_{\text {h }}$ |
| 83 h | 0... 3 | Set value counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 4... 7 | Final value counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 8... 11 | Loading value counter 2 | See counter 1 | $00_{\text {h }}$ |
|  | 12 | Hysteresis counter 2 | See counter 1 | $00_{h}$ |

## 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 |
| :---: | :---: |
| 8 | Bit 0: 0 <br> Bit 1: 0 <br> Bit 2: Counter 0, overflow/underflow/final value <br> Bit 3: Counter 0, comparison value reached <br> Bit 4: 0 <br> Bit 5: 0 <br> Bit 6: Counter 1, overflow/underflow/final value <br> Bit 7: Counter 1, comparison value reached |
| 9 | State of the inputs at the time of the alarm <br> Bit 0: Counter 0, A/pulse <br> Bit 1: Counter $0, \mathrm{~B} /$ direction <br> Bit 2: Counter 1, A/pulse <br> Bit 3: Counter 1, B/direction <br> Bit 7 ... 4:0 (fixed) |
| $10 . . .11$ | 16 bit $\mu s$ value at the time of the alarm |

Ticker value: After mains connection a timer ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu$ s value together with the count value in the input area.

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00h). 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 processingincoming 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 $_{\text {incoming }}$ 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 incoming and diagnostic alarm ${ }_{\text {outgoing }}$ are not buffered and are lost. During this time (first diagnostic alarm 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 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, diagnostics ${ }_{\text {incoming }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $\mathbf{1 0 0 0}_{\mathrm{b}}$ : function module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) <br> Bit 6: Process alarm lost <br> Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here 08h) |
| 6 | Number of channels of a module (here 02 h ) |
| 7 | Bit 0: Error in channel group 0 (counter 0) <br> Bit 1: Error in channel group 1 (counter 1) Bit 7 ... 2: 0 (fixed) |
| 8 | Channel group 0: Diagnostic alarm Idue to lost process alarm to ... <br> Bit 1... 0:0 (fixed) <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 7... 4: 0 (fixed) |
| 9 | Channel group 1: Diagnostic alarm Idue to lost process alarm to ... <br> Bit $1 . .0: 0$ (fixed) <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 7... 4: 0 (fixed) |
| $10 . . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message outgoing is effected $^{\text {d }}$ |

### 12.8.3 One 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 |
| :--- | :--- |
| Counting <br> continuously | The counter counts from the loading value to the counting limit, then skips to the <br> opposite counting limit and continues to count from there. |
| Counting once | The counter counts once/periodically from the loading value in the specified counting <br> range. |
| Counting periodically | the |


| Signal evaluation | Description |
| :---: | :---: |
| Single rotary transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary transducer |  |
| Quadruple rotary transducer |  |
| Direction | Pulse at input "A/pulse" and direction at "B/direction" |
| Additional functions | Description |
| Main counting direction | The main counting direction can be parameterised: <br> None: The whole counting range is available. <br> 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 (l-gate) is conform to the software gate (SW gate) which you control via your user program (status word in the output area). <br> The following response can be parameterised: <br> Cancelling gate function:After closing the gate and opening it again, the counting process continues from the loading value again. <br> 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: <br> - Hardware gate open <br> - Hardware gate closed <br> - Counting limit - overflow <br> - Counting limit - underflow <br> - Comparison value reached <br> - Final value reached <br> - Latch value reached |
| Diagnostic alarm | 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. |

## Read data: 8 bytes

| Input area |  |  |
| :--- | :--- | :--- |
| Addr. | Access | Assignment |
| +0 | Double word | Count value |
| +4 | Word | Status word (see the following table) |
| +6 | Word | Ticker value |

## Count value: Current counter content

Ticker value: After mains connection a timer ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu s$ value together with the count value in the input area.

| EPM-S602 status word |  | Function |
| :--- | :--- | :--- |
| Bit | Designation | Reset was active |
| 0 | STS_SYNC | Is set if the comparison bit is enabled |
| 1 | STS_CTRL_COMP | Software gate status (set if SW gate active) |
| 2 | STS_SW-GATE | Status of reset input |
| 3 | STS_RST | Reserved |
| 4 | - | Status of internal gate (set if internal gate active) |
| 5 | STS_GATE | Status of comparison bit |
| 6 | STS_COMP | Status set for counter direction backwards |
| 7 | STS_CMP* | Status set for counter direction forwards |
| 8 | STS_END* | Comparator status is set if the comparison condition is met. |
| If the comparison is deactivated (counter mode byte $1=000_{b}$ ), the bit has |  |  |
| no function. |  |  |
| 9 | STS_OFLW* | Status set if final value was reached |
| 10 | STS_UFLW* | Status set in the case of overflow |
| 11 | STS_ZP* | Status set in the case of underflow |
| 12 | - | Status set in the case of zero crossing |
| 13 | - | Reserved |
| 14 | Reserved |  |
| 15 |  |  |

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


## Write data: $\mathbf{1 0}$ bytes

Output area

| Addr. | Access | Assignment |
| :--- | :--- | :--- |
| +0 | Double word | Comparison value |
| +4 | Double word | Set value |
| +8 | Word | Control word (see the following table) |

Comparison value: Here you can select a value which, by comparison with the current counter content, can impact the counter output or trigger a process alarm. The response of the output or 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.

| EPM-S602 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | Function |
| 0 | CTRL_SYNC_SET | Activates the reset mode |
| 1 | CTRL_COMP_SET | Enables the comparison bit |
| 2 | SW_GATE_SET | Sets the software gate |
| 3 | - | Reserved |
| 4 | - | Reserved |
| 5 | COUNTERVAL_SET | Sets the counter temporarily to the value in the set value |
| 6 | RES_SET | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | CTRL_SYNC_RESET | Reserved |
| 8 | CTRL_COMP_RESET | Inhibits the comparison bit |
| 9 | SW_GATE_RESET | Resets the software gate |
| 10 |  | Reserved |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $00_{h}$ | 0 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bits 5 ... 0: reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{h}$ |
| $01_{h}$ | 0 | Input frequency track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. <br> 2 (02h): 100 kHz <br> $3\left(03_{h}\right): 60 \mathrm{kHz}$ <br> 4 (04h): 30 kHz <br> 6 (06h): 10 kHz <br> $7\left(07_{h}\right): 5 \mathrm{kHz}$ <br> $8\left(08_{h}\right): 2 \mathrm{kHz}$ <br> $9\left(09_{h}\right): 1 \mathrm{kHz}$ <br> Other values are not permissible! | 02h |
|  | 1 | Input frequency track B |  | 02h |
|  | 2 | Input frequency reset |  | 02h |
|  | 3 | Reserved |  |  |
| $80_{h}$ | 0 | Alarm response | Setting activates process alarm Bits 1 ... 0: reserved <br> Bit 2: Proc. alarm overflow <br> Bit 3: Proc. alarm underflow <br> Bit 4: Proc. alarm comparison value <br> Bit 5: Proc. alarm final value <br> Bits 7 ... 6: Reserved | $00_{h}$ |
|  | 1 | Counter function | ```Bit 5 ... 0: 000000专 = counting continuously 000001 b = once: forwards 000010b = once: backwards 000100 b = once: no main counting direction 001000}b=\mathrm{ periodically: forwards 010000 b = periodically: backwards 100000}b= periodically: no main counting directio Bits 7 ... 6: Reserved``` | $00_{\text {h }}$ |


| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
|  | 2 | Comparator | Bits $2 \ldots 0$ : Comparison bit is set (... if condition is met) $000_{b}=\text { never }$ <br> $001_{b}=$ count value $\geq$ comparison value <br> $010_{b}=$ count value $\leq$ comparison value <br> $100_{b}=$ count value $=$ comparison value <br> Bit 3: Invert counting direction track B <br> 0 = no (do not invert) <br> 1 = yes (invert) <br> Bits 6 ... 4: Reset <br> $000_{b}=$ deactivated <br> $001_{b}=$ HIGH level <br> $011_{b}=$ rising edge <br> $101_{b}=$ rising edge, once <br> Bit 7: Reserved | $00_{\text {h }}$ |
|  | 3 | Signal evaluation | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter <br> details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at $A$ and $B$ ) <br> $010 \mathrm{~b}=$ rotary transducer double (at A and B ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 6 ... 3: Reserved <br> Bit 7: Gate function (internal gate) <br> $0=$ abort (counting process starts again from loading value) <br> 1 = interrupt (counting process is continued with counter content) | 00 h |
| 81 h | 4... 7 | Final value | Counting method: <br> 0x310D: byte 3 (high byte) <br> $0 \times 310$ E: byte 2 <br> 0x311F: byte 1 <br> $0 \times 3110$ : byte 0 (low byte) | $00_{\text {h }}$ |
|  | 8... 11 | Loading value | Counting method: <br> 0x3109: byte 3 (high byte) <br> 0x310A: byte 2 <br> 0x310B: byte 1 <br> $0 \times 310 \mathrm{C}$ : byte 0 (low byte) | 00 h |
|  | 12 | Hysteresis |  | $00_{\text {h }}$ |

## 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 |
| :---: | :---: |
| 8 | Bit 0: 0 <br> Bit 1: 0 <br> Bit 2: Counter 0, overflow/underflow/final value <br> Bit 3: Counter 0, comparison value reached <br> Bit 4: 0 <br> Bit 5: 0 <br> Bit 6: Counter 1, overflow/underflow/final value <br> Bit 7: Counter 1, comparison value reached |
| 9 | State of the inputs at the time of the alarm <br> Bit 0: Counter 0, A/pulse <br> Bit 1: Counter $0, \mathrm{~B} /$ direction <br> Bit 2: Counter 1, A/pulse <br> Bit 3: Counter 1, B/direction <br> Bit 7 ... 4: 0 (fixed) |
| $10 . . .11$ | 16 bit $\mu$ s value at the time of the alarm |

Ticker value: After mains connection a timer ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the count value the time value of the timer is stored as 16 -bit $\mu$ s value together with the count value in the input area.

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00 h ). 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 incoming 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 ${ }_{\text {incoming }}$ has been triggered is processed, the diagnostic alarm processing is called again as diagnostic alarmoutgoing.
All events of a channel between the diagnostic alarm incoming and diagnostic alarm ${ }_{\text {outgoing }}$ are not buffered and are lost. During this time (first diagnostic alarm 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 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, diagnostics ${ }_{\text {incoming }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set if module fault <br> Bit 1: 0 (fixed) <br> Bit 2: Set in the case of an external error <br> Bit 3: Set in the case of a channel error <br> Bit 6 ... 4: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $\mathbf{1 0 0 0}_{b}$ : function module Bit 4: Channel information available Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here $08{ }_{\text {h }}$ ) |
| 6 | Number of channels of a module (here 02h) |
| 7 | Bit 0: Error in channel group 0 (counter 0) <br> Bit 1: Error in channel group 1 (counter 1) <br> Bit 7... 2: 0 (fixed) |
| 8 | Channel group 0: Diagnostic alarm Idue to lost process alarm to ... <br> Bit $1 . . .0: 0$ (fixed) <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 7... 4: 0 (fixed) |
| 9 | Channel group 1: Diagnostic alarm Idue to lost process alarm to ... <br> Bit $1 . . .0: 0$ (fixed) <br> Bit 2: Overflow/underflow/final value <br> Bit 3: Comparison value reached <br> Bit 7 ... 4: 0 (fixed) |
| $10 . .15$ | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message ${ }_{\text {outgoing }}$ is effected |

### 12.8.4 Two 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 |
| :--- | :--- |
| Counting <br> continuously | The counter counts from 0 to the counting limit, then skips to the opposite counting limit <br> and continues to count from there. |
|  | Description |
| Signal evaluation <br> Single rotary <br> transducer | Connection to input "A/pulse" and "B/direction" |
| Double rotary <br> transducer | Pulse at input "A/pulse" and direction at "B/direction" |
| Quadruple rotary <br> transducer | Description |
| Direction | The gate function serves to start, stop, and interrupt a counting function. In the case of <br> this counter the internal gate (I-gate) is conform to the software gate (SW gate) which <br> you control via your user program (status word in the output area). |
| Additional functions |  |

Read data: 12 bytes

| Addr. | Access | Assignment |
| :---: | :---: | :---: |
| +0 | Double word | Counter 1: count value |
| +4 | Double word | Counter 2: count value |
| +8 | Word | Counter 1: status word (see following table) |
| +10 | Word | Counter 2: status word (see following table) |
| EPM-S603 status word |  |  |
| Bit | Designation | 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 for counter direction backwards |
| 8 | STS_C_UP | Status set for counter direction forwards |
| 9 | - | Reserved |
| 10 | - | Reserved |
| 11 | STS_OFLW* | Status set in the case of overflow |
| 12 | STS_UFLW* | Status set in the case of underflow |
| 13 | STS_ZP* | Status set in the case of zero crossing |
| 14 | - | Reserved |
| 15 | - | Reserved |

[^7]
## Write data: 4 bytes

## Output area in the process image

| Addr. | Access | Assignment |
| :--- | :--- | :--- |
| +0 | Word | Counter 1: Control word (see following table) |
| +2 | Word | Counter 2: Control word (see following table) |


| EPM-S603 control word |  |  |
| :--- | :--- | :--- |
| Bit | Designation | Function |
| 0 | - | Reserved |
| 1 | - | Reserved |
| 2 | SW_GATE_SET | Sets the software gate |
| 3 | - | Reserved |
| 4 | - | Reserved |
| 5 | - | Reserved |
| 6 | - | Resets bits STS_CMP, STS_END, STS_OFLW, STS_UFLW and STS_ZP with a <br> rising edge |
| 7 | - | Reserved |
| 8 | SW_GATE_RESET | Reserved |
| 9 | - | Reserved |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $01_{\text {h }}$ | 0 | Input frequency counter 1, track A | Filters for instance serve to filter signal peaks in the case of an unclean input signal. $\begin{aligned} & 2\left(02_{\mathrm{h}}\right): 100 \mathrm{kHz} \\ & 3\left(03_{\mathrm{h}}\right): 60 \mathrm{kHz} \\ & 4\left(04_{\mathrm{h}}\right): 30 \mathrm{kHz} \\ & 6\left(06_{\mathrm{h}}\right): 10 \mathrm{kHz} \\ & 7\left(07_{\mathrm{h}}\right): 5 \mathrm{kHz} \\ & 8\left(08_{\mathrm{h}}\right): 2 \mathrm{kHz} \\ & 9\left(09_{\mathrm{h}}\right): 1 \mathrm{kHz} \end{aligned}$ <br> Other values are not permissible! | $02_{\text {h }}$ |
|  | 1 | Input frequency counter 1, track B |  | 02h |
|  | 2 | Input frequency counter 2, track A |  | 02 h |
|  | 3 | Input frequency counter 2, track B |  | $02_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Counting direction counter 1, track B | Bits 2 ... 0: Reserved <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{h}$ |
|  | 1 | Signal evaluation counter 1 | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 7 ... 3: Reserved | $00_{\text {h }}$ |
| $82^{\text {h }}$ | 0 | Counting direction counter 2, track B | Bits 2 ... 0: Reserved <br> Bit 3: Invert counting direction track B <br> $0=$ no (do not invert) <br> 1 = yes (invert) <br> Bits 7 ... 4: Reserved | $00_{\text {h }}$ |
|  | 1 | Signal evaluation counter 2 | Bits 2 ... 0: Signal evaluation <br> $000_{b}=$ counter deactivated (the other parameter <br> details for the counter are ignored) <br> $001_{b}=$ rotary transducer single (at A and B) <br> $010_{b}=$ rotary transducer double (at $A$ and $B$ ) <br> $011_{b}=$ rotary transducer quadruple (at $A$ and $B$ ) <br> $100_{b}=$ direction (pulse at $A$ and direction at $B$ ) <br> Bits 7 ... 3: Reserved | $00_{\text {h }}$ |

## Diagnostic data

Using the SFB 52 you can access the diagnostic data of the module anytime. 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 $7 \ldots \mathbf{0}$ |
| :---: | :--- |
| 0 | 0 (fixed) |
| 1 | Bit 3 ... 0: Module class, $1000_{\mathrm{b}}$ : function module <br> Bit 4: Channel information available <br> Bit 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module <br> Bit 7: Further channel types available, 0: no, 1: yes |
| 5 | Number of diagnostic bits output by the module per channel (here $00_{\mathrm{h}}$ ) |
| 6 | Number of channels of a module (here $02_{\mathrm{h}}$ ) |
| $7 \ldots 15$ | 0 (fixed) |

### 12.9 Parameterising the encoder evaluation

### 12.9.1 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 <br> 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 ( $\mu \mathrm{s}$ ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change of the encoder value the time value of the timer is stored as 16 -bit $\mu$ s value together with the encoder value in the input area.

Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| 00 h | 0 | Diagnostics | A diagnostic alarm occurs if the same event triggers a further process alarm during a process alarm processing. Bits 5 ... 0 : reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled ) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Dead 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. <br> HIGH LOW <br> $00_{h} 30_{h}: 1 \mu \mathrm{~s}$ <br> $00_{h} 60_{h}: 2 \mu \mathrm{~s}$ <br> $00_{h} \mathrm{CO}_{\mathrm{h}}: 4 \mu \mathrm{~s}$ <br> $01_{h} 80_{h}: 8 \mu s$ <br> $03_{\mathrm{h}} 00_{\mathrm{h}}: 16 \mu \mathrm{~s}$ <br> $06_{h} 00_{h}: 32 \mu \mathrm{~s}$ <br> $09_{\mathrm{h}} 00_{\mathrm{h}}: 48 \mu \mathrm{~s}$ <br> $0 C_{h} 00_{h}: 64 \mu s$ | 0 COO h |



| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
|  | 5 |  | Bit 1 ... 0: Ready for operation <br> 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. <br> In the "Master mode" operating mode the module provides a cycle to the encoder and receives data by the encoder. <br> $01_{b}=$ monitoring operation <br> $10_{b}=$ master mode <br> Bit 2: Shifting direction <br> 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". <br> $0=$ LSB first (LSB is transmitted first) <br> $1=$ MSB first (MSB is transmitted first) <br> Bit 3: edge clock signal <br> Here you can specify the edge type of the clock signal, in the case of which the encoder supplies data. <br> Information on this can be found in the data sheet for your encoder. Usually the SSI encoders respond to rising edges. <br> 0 = falling edge <br> 1 = rising edge <br> Bit 4: Coding <br> 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. <br> 0 = standard code <br> 1 = Gray code <br> Bits 7 ... 5: reserved | $1 E_{h}$ |
|  | 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. <br> $0\left(00_{h}\right)=$ inhibited <br> $1\left(01_{h}\right)=$ enabled | $00_{\text {h }}$ |

## Diagnostic alarm

You have the possibility of globally activating a diagnostic alarm for the module via parameterisation (data set 00 h ). 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 incoming 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 alarmincoming 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 outgoing.
All events of a channel between the diagnostic alarm incoming and diagnostic alarm outgoing are not buffered and are lost. During this time (first diagnostic alarm 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 incoming/outgoing.
Example:


## Diagnostic alarm processing:

Using 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 ca no longer be clearly assigned to the last diagnostic alarm.
Data set 1 is structured as follows:

| Data set 1, diagnostics ${ }_{\text {incoming }}$ |  |
| :---: | :---: |
| Byte | Bit 7 ... 0 |
| 0 | Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error <br> Bit 3: Set if channel error available <br> Bit 4: Set in the case of missing external supply voltage <br> Bit 6... 5: 0 (fixed) <br> Bit 7: Parameterisation error |
| 1 | Bit 3 ... 0: Module class, $1000_{b}$ : function module Bit 4: Channel information available Bit 7... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | Bit 5 ... 0:0 (fixed) Bit 6: Process alarm lost Bit 7: 0 (fixed) |
| 4 | Bit 6 ... 0: Channel type, 76h: counter module Bit 7: 0 (fixed) |
| 5 | Number of diagnostic bits output by the module per channel (here 08\%) |
| 6 | Number of channels of a module (here $01_{h}$ ) |
| 7 | Bit 0: Error in channel group 0 |
| 8... 15 | 0 (fixed) |
|  | Data set 1, diagnostics ${ }_{\text {outgoing }}$ |
|  | After the error recovery a diagnostic message outgoing is effected $^{\text {d }}$ |

### 12.10 <br> Time stamp parameterising

12.10.1 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 the edge change is saved here. Parameters can be set for the following variants by incorporating the GSD file LE010C3A.gsd:
20 bytes, 5 time stamp entries:

| Addr. | $\mathbf{+ 0}$ | $\mathbf{+ 1}$ | $\mathbf{+ 2}$ |
| :--- | :--- | :--- | :--- | :--- |
| +0 | PAE | RN | +3 |
| +4 | PAE | RN-1 | -bit $\mu$ s value |
| +8 | PAE | RN-2 | 16 -bit $\mu$ s value |
| +12 | PAE | RN-3 | 16 -bit $\mu$ s value |
| +16 | PAE | RN-4 | 16 -bit $\mu$ s value |

60 bytes, 15 time stamp entries:

| Addr. | $\mathbf{+ 0}$ | $\mathbf{+ 1}$ | $\boldsymbol{+ 2}$ |
| :--- | :--- | :--- | :--- | :--- |
| +0 | PAE | RN | +3 |
| +4 | PAE | RN-1 | -bit $\mu$ s value |
| +8 | PAE | RN-2 | 16 -bit $\mu$ s value |
| +12 | PAE | RN-3 | 16 -bit $\mu$ s value |
| $\ldots$ | $\ldots$ | $\ldots$ | 16 -bit $\mu$ s value |
| +56 | PAE | RN-14 | $\ldots$ |

Running number: the "running number" (RN) is a consecutive number between 0 ... 127, which always starts afresh from 0 . The "running number" reflects the time sequence of the edges
Ticker value: After mains connection, a timer ( $\mu$ s ticker) is started, which after $65535 \mu \mathrm{~s}$ starts with 0 again. With every change in the encoder value the time value of the timer is stored as a 16 -bit $\mu \mathrm{s}$ value together with the encoder value in the input area.

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| 02h | 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. | $\begin{aligned} & 14_{\mathrm{h}} \text { or } 3 C_{\mathrm{h}} \\ & (\mathrm{fix}) \end{aligned}$ |
|  | 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. | $00_{h}$ (fix) |
| $01_{\text {h }}$ | 0 | Input delay DI 1 | $\begin{aligned} & 00_{\mathrm{h}}=1 \mu \mathrm{~s} \\ & 02_{\mathrm{h}}=3 \mu \mathrm{~s} \\ & 04_{\mathrm{h}}=10 \mu \mathrm{~s} \end{aligned}$ | $02_{\text {h }}$ |
|  | 1 | Input delay DI 2 | $\begin{aligned} & 07_{\mathrm{h}}=86 \mu \mathrm{~s} \\ & 09_{\mathrm{h}}=342 \mu \mathrm{~s} \\ & 0 \mathrm{C}_{\mathrm{h}}=273 \mu \mathrm{~s} \end{aligned}$ <br> Other values are not permissible. | $02_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Edge 0-1 an 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 | $00_{h}$ |
|  | 1 | Edge 1-0 at 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 | $00_{\text {h }}$ |

## 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 | Bits 3 ... 0: module class, $1111_{b}$ : digital module Bit 4: channel information available Bits 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bits 6 ... 0: channel type, $70_{h}$ : digital module Bit 7: more channel types available (0: yes; 1: no) |
| 5 | Number of diagnostic bits output by the module per channel (here $00_{h}$ ) |
| 6 | Number of channels of a module (here 02 h ) |
| $7 . . .15$ | 0 (fixed) |

### 12.10.2

2 digital outputs with time stamp function - EPM-S310
By integration of the GSE file VIO10C19.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 | Bits 5 ... 0: running number (RN = Running Number) of the last FIFO entry <br> Bit 6: 1 (fixed) <br> Bit 7: 0 (fixed) |
| +1 | Byte | Bits $5 \ldots 0:$ running number of the next FIFO entry <br> Bit 6: 1 (fixed) <br> Bit 7: 1 (fixed) |
| +2 | Byte | Status |
| +3 | Byte | Number of time stamp entries in FIFO memory. |

Running number: 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 $00_{h} / 80_{h}$ : everything OK
Code $01_{h} / 81_{h}$ : no following time stamp entry
Code $02_{h} / 82_{h}$ : no new time stamp entries.
Code $03_{h} / 83_{h}$ : FIFO memory is full. No new time stamp entries can be accepted.
If bit 6 of the last processed running number (RN) was set, the code is returned at $80_{h}$ 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 | Status of outputs (PAA) |
| +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 GSD file LEO10C3A.gsd.gsd:
20 bytes, 5 time stamp entries:

| Addr. | $\boldsymbol{+ 0}$ | $\mathbf{+ 1}$ | $\mathbf{+ 2}$ |
| :--- | :--- | :--- | :--- | :--- |
| +0 | PAA | RN | +3 |
| +4 | PAA | RN-1 | -bit $\mu$ s value |
| +8 | PAA | RN-2 | 16 -bit $\mu$ s value |
| +12 | PAA | RN-3 | 16 -bit $\mu$ s value |
| +16 | PAA | RN-4 | 16 -bit $\mu$ s value |

60 bytes, 15 time stamp entries:

| Addr. | $\boldsymbol{+ 0}$ | $\boldsymbol{+ 1}$ | $\boldsymbol{+ 2}$ |
| :--- | :--- | :--- | :--- | :--- |
| +0 | PAA | RN | 16 -bit $\mu$ s value |
| +4 | PAA | RN-1 | 16 -bit $\mu$ s value |
| +8 | PAA | RN-2 | 16 -bit $\mu$ s value |
| +12 | PAA | RN-3 | 16 -bit $\mu$ s value |
| $\ldots$ | $\ldots$ | $\ldots$ | $\ldots$ |
| +56 | PAA | RN-14 | 16 -bit $\mu$ s value |

Running number: the "running number" (RN) 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. This should be incremented with every time stamp entry.

## 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 $\mu \mathrm{s}$ 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 |  |  |  |
| $02{ }_{h}$ | 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. | $\begin{aligned} & 14_{h} \text { or } 3 C_{h} \\ & \text { (fix) } \end{aligned}$ |
|  | 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. | 00h (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 | Bits 3 ... 0: module class, $1111_{b}$ : digital module Bit 4: channel information available Bits 7 ... 5: 0 (fixed) |
| 2 | 0 (fixed) |
| 3 | 0 (fixed) |
| 4 | Bits $6 \ldots 0$ : channel type, $70_{h}$ : digital module Bit 7: more channel types available (0: yes; 1: no) |
| 5 | Number of diagnostic bits output by the module per channel (here $00{ }_{\text {h }}$ ) |
| 6 | Number of channels of a module (here 02 h ) |
| 7 ... 15 | 0 (fixed) |

### 12.11 Parameterising technology modules

12.11.1 2 digital outputs with PWM functionality - EPM-S620

The following functions can be parameterised:
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 |
| +2 | PWMSTS_II | 2 | PWM 2: status |
| Status PVMx |  |  |  |
| Bit | Name | Function |  |
| 0 | - | Reserved |  |
| 1 | STS_PVM | Status PWM <br> 0: PWM output stopped <br> $1: ~ P W M ~ o u t p u t ~ a c t i v e ~$ |  |
| 2 | STS_OUTBV | Output status <br> 0: push/pull output <br> $1:$ highside output |  |
| $3 \ldots 15$ | - | 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 here by stating the duration of the HIGH level for the corresponding PWM channel. The pulse duration should be chosen as a factor for the 20.83 ns basis.

Value range: 48 ... 8388607 ( $1 \mu \mathrm{~s}$... approx. 175ms)
PWMPD_I, PWMPD_Il control word:here you can specify the PWM output response for the corresponding channel and start or stop PWM output.

| Control word PWMPDx |  |  |
| :---: | :---: | :---: |
| Bit | Name | Function |
| 0 | - | Reserved |
| 1 | CTRL_OUTBV | PWM output response <br> 0: push/pull output <br> 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. <br> 1: highside output <br> 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 | - | Reserved |
| 8 | CTRL_STRT | Edge 0-1 starts PWM output on channel x |
| 9 | CTRL_STP | Edge 0-1 stops PWM output on channel x |
| $10 . . .15$ | - | Reserved |

## Parameter data

| Data set |  | Name | Description/value | Lenze |
| :---: | :---: | :---: | :---: | :---: |
| No. | Byte |  |  |  |
| $80_{\text {h }}$ | 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. <br> Values below $25 \mu$ s are ignored. If the pulse duration is higher or equal to the period, the DO output is set permanently. <br> Value range: 1200 ... 8388607 ( $25 \mu \mathrm{~s} . .$. approx. 175 ms ) | $1 \mathrm{~F} 40_{\text {h }}$ |
| $81_{\text {h }}$ | 0 | PWM 2: Period |  | 1F40h |

## Diagnostic data

Since this module does not support alarms, the diagnostic data provides information about this module.

| Diagnostic data set - data set 01h |  |  |  |
| :---: | :---: | :---: | :---: |
| Name | Byte | Function | Default |
| ERR_A | 1 | Reserved | $00_{\text {h }}$ |
| MODTYP | 1 | Module information <br> Byte 0: <br> Bit 3 ... 0: module class (1111b: digital module) <br> Bit 4: channel information available <br> Bits 7 ... 4: 0 reserved | 15 h |
| ERR_C | 1 | Reserved | $00_{\text {h }}$ |
| ERR_D | 1 | Reserved | $00_{\text {h }}$ |
| CHTYP | 1 | Channel type <br> Byte 0: <br> Bits 6 ... 0: channel type (72h: digital output) <br> Bit 7: reserved | $72_{\text {h }}$ |
| NUMBIT | 1 | Number of diagnostic bits per channel Byte 0: here $00_{h}$ | $00_{\text {h }}$ |
| NUMCH | 1 | Number of channels in module Byte 0: here 02h | 02h |
| CHERR | 1 | Reserved | $00_{\text {h }}$ |
| CH0ERR ... CH7ERR | 6 | Reserved | $00_{\text {h }}$ |
| DIAG_US | 4 | Value of $\mu$ s ticker when diagnostics occur Bytes 0 ... 3 | 0 |

### 12.11.2

RS232 interface - EPM-S640

Information on the transmission principles can be found in the appendix (■1727).

## Parameter data

| Parameter data - ASCII protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
| $02_{\text {h }}$ | 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. |  |
| $00_{h}$ | 0 | Diagnostics | Bits 5 ... 0 : Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02 h : 300 Baud $03_{h}$ : 600 Baud 04 h : 1200 Baud $05 \mathrm{~h}: 1800$ Baud $06_{h}$ : 2400 Baud $07_{\mathrm{h}}$ : 4800 Baud $08_{\mathrm{h}}$ : 7200 Baud $09_{\mathrm{h}}$ : 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud $0 \mathrm{D}_{\mathrm{h}}: 57600$ Baud $0 \mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{\text {h }}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{h}: \text { ASCII }$ | $01_{\text {h }}$ |
|  | 2 | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{b}$ : 6 <br> 10 : 7 <br> $11_{\mathrm{b}}$ : 8 <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{b}: 1 \\ & 10_{b}: 1.5 \\ & 11_{b}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | $13_{\text {h }}$ |



| Parameter data STX/ETX protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set <br> No. | Byte | Name | Description/value | Lenze |
| 02h | 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. |  |
| $00_{h}$ | 0 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; 1 = enabled) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{h}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02h: 300 Baud $03_{\mathrm{h}}$ : 600 Baud 04h: 1200 Baud 05h: 1800 Baud $06 h_{h} 2400$ Baud $07_{h}$ : 4800 Baud 08h: 7200 Baud 09h: 9600 Baud $0 A_{h}$ : 14400 Baud 0Bh: 19200 Baud $0 C_{h}: 38400$ Baud 0Dh: 57600 Baud 0E $\mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 F_{h}: 115200$ Baud 10h: 109700 Baud | $00_{\text {h }}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{h}: \text { STX/ETX }$ | 02h |



| Parameter data 3964(R) protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
| 02h | 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. |  |
|  |  | Length - process image output data | Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible. |  |
| $00_{h}$ | 0 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02 h : 300 Baud $03_{\mathrm{h}}$ : 600 Baud 04 h : 1200 Baud $05 \mathrm{~h}: 1800$ Baud 06h: 2400 Baud $07_{\mathrm{h}}$ : 4800 Baud $08_{\mathrm{h}}$ : 7200 Baud $09_{\mathrm{h}}$ : 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud $0 \mathrm{D}_{\mathrm{h}}: 57600$ Baud 0E h : 76800 Baud $0 F_{h}: 115200$ Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{h}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{\mathrm{h}}: 3964 \\ & 03_{\mathrm{h}}: 3964 \mathrm{R} \end{aligned}$ | $03_{h}$ |
|  | 2 | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{b}: 6$ <br> $10_{b}: 7$ <br> $11_{\mathrm{b}}$ : 8 <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $01_{b}$ : 1 <br> $10_{\mathrm{b}}$ : 1.5 <br> $11_{b}$ : 2 <br> Bit 7/6 flow control <br> $00_{\mathrm{b}}$ : None <br> $01_{\mathrm{b}}$ : Hardware <br> $10_{b}$ : XON/XOFF | $13_{h}$ |
|  | 3 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} . . . \text { FF }_{h}\right)$ | $00_{h}$ |
|  | 4 | Character delay time | 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). <br> 0 ... 65535 [20-ms steps] ( $00_{h} . . . \mathrm{FF}_{\mathrm{h}}$ ) | OAh |


| Parameter data 3964(R) protocol <br> Data set <br> No. | Name |  | Lescription/value |
| :--- | :--- | :--- | :--- | :--- | Lenze

## 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.

| Name | Byte | Function | Default |
| :---: | :---: | :---: | :---: |
| ERR_A | 1 | ERR_A-diagnostics <br> Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error (cable break) <br> Bit 3: Reserved <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 5, 6: Reserved <br> Bit 7: Set in the case of parameterisation error | 00 h |
| MODTYP | 1 | Module information <br> Byte 0: <br> Bit 3 ... 0: Module class (0111b: Gateway module) <br> Bit 4: channel information available <br> Bits 7 ... 4: 0 reserved | $17_{\text {h }}$ |
| ERR_C | 1 | ERR_A-diagnostics Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| ERR_D | 1 | ERR_D diagnostics <br> Bit 3 ... 0: Reserved <br> Bit 4: Set in the case of internal communication error <br> Bit 7 ... 5: Reserved | 00 h |
| CHTYP | 1 | Channel type <br> Bit 7 ... 0: Reserved | 00 h |
| NUMBIT | 1 | Number of diagnostic bits of the module per channel (here 08 h) | 08 h |
| NUMCH | 1 | Number of channels in module Bit 7 ... 0: Reserved | 00 h |
| CHERR | 1 | Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| CH0ERR ... CH7ERR | 8 | Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| DIAG_US | 4 | Value of $\mu$ s ticker when diagnostics occur Bytes 0 ... 3 | 00 h |

### 12.11.3 <br> RS485 interface - EPM-S650

Information on the transmission principles can be found in the appendix (■1727).

## Parameter data

| Parameter data - ASCII protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
| $02_{\text {h }}$ | 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. |  |
| $00_{h}$ | 0 | Diagnostics | Bits 5 ... 0 : Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02 h : 300 Baud $03_{h}$ : 600 Baud 04 h : 1200 Baud $05 \mathrm{~h}: 1800$ Baud $06_{h}$ : 2400 Baud $07_{\mathrm{h}}$ : 4800 Baud $08_{\mathrm{h}}$ : 7200 Baud $09_{\mathrm{h}}$ : 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud $0 \mathrm{D}_{\mathrm{h}}: 57600$ Baud $0 \mathrm{E}_{\mathrm{h}}$ : 76800 Baud $0 \mathrm{~F}_{\mathrm{h}}$ : 115200 Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{\text {h }}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $01_{h}: \text { ASCII }$ | $01_{\text {h }}$ |
|  | 2 | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{b}$ : 6 <br> 10 : 7 <br> $11_{\mathrm{b}}$ : 8 <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $\begin{aligned} & 01_{b}: 1 \\ & 10_{b}: 1.5 \\ & 11_{b}: 2 \end{aligned}$ <br> Bit 7/6 flow control $00_{b}$ : None $01_{b}$ : Hardware $10_{b}$ : XON/XOFF | $13_{\text {h }}$ |


| Parameter data - ASCII protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
|  | 3 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 . . .65535[\mathrm{~ms}]\left(0000_{h} . . . \text { FFFF }_{h}\right)$ | $0^{0000}{ }_{\text {h }}$ |
|  | 4 | Character delay time | 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). <br> $0 . . .65535$ [ms] ( $0000_{h} . .$. FFFF $_{h}$ ) | $000 \mathrm{~A}_{\mathrm{h}}$ |
|  | 5 | 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. <br> $1 . . .255\left(01_{h} \ldots F_{h}\right)$ | $01_{\text {h }}$ |
|  | 6 | Operating mode | Operating mode of the interface <br> $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $0^{\text {h }}$ |
|  | 7 | 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 (■ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal R(A) 5 V (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal R(A) 0 V; signal R(B) 5 V. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $01_{\text {h }}$ |


| Parameter data STX/ETX protocol <br> Data set <br> No. | Byte | Name | Description/value | Lenze |
| :--- | :--- | :--- | :--- | :--- | :--- |
| $02_{\mathrm{h}}$ | 0 | Length - process image <br> input data | Length of input data (backplane bus <br> communication); the values are specified by the <br> system. Other values are not permissible. |  |
| 1 | Length - process image <br> output data | Length of output data (backplane bus <br> communication); the values are specified by the <br> system. Other values are not permissible. |  |  |
| $00_{\mathrm{h}}$ | 0 | Diagnostics | Bits 5 $\ldots .0$ Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ <br> enabled) <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\mathrm{h}}$ |


| Parameter data STX/ETX protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set No. | Byte | Name | Description/value | Lenze |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud $02_{\mathrm{h}}$ : 300 Baud $03_{\mathrm{h}}$ : 600 Baud 04h: 1200 Baud $05 \mathrm{~h}: 1800$ Baud $06_{h}$ : 2400 Baud 07 h : 4800 Baud $08_{h}$ : 7200 Baud 09 h: 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud 0Dh: 57600 Baud OE h : 76800 Baud $0 F_{h}$ : 115200 Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{\text {h }}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $02_{h}: \text { STX/ETX }$ | 02h |
|  | 2 | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{b}: 6$ <br> $10_{\mathrm{b}}$ : 7 <br> $11_{b}: 8$ <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits <br> $01_{b}$ : 1 <br> $10_{\mathrm{b}}: 1.5$ <br> $11_{b}$ : 2 <br> Bit 7/6 flow control <br> $00_{b}$ : None <br> $01_{\mathrm{b}}$ : Hardware <br> $10_{b}$ : XON/XOFF | $13_{h}$ |
|  | 3 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 . . .65535 \text { [ms] }\left(0000_{\mathrm{h}} . \therefore \text { FFFF }_{\mathrm{h}}\right)$ | $0000{ }_{\text {h }}$ |
|  | 4 | TMO | TMO serves to define the maximally permissible interval between two frames. <br> $0 . . .65535$ [ms] $\left(0000_{h} . .\right.$. FFFF $\left._{h}\right)$ | OAh |
|  | 5 | No. of start identifiers | $00_{h}: 1$ start identifier (2. start identifier is ignored) <br> $01_{h}$ : 2 start identifiers | $01_{\text {h }}$ |
|  | 6 | Start identifier 1 | ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. $1 \ldots 255\left(01_{h} \ldots F_{h}\right)$ | 02h |
|  | 7 | Start identifier 2 | ASCII value of the initial character that is sent in advance of a frame and marks the start of a transmission. | $00_{\text {h }}$ |
|  | 8 | No. of end identifier | $00_{h}$ : 1 end identifier (2. end identifier ( $0 \times 310 \mathrm{D} / \mathrm{x}$ ) is ignored) <br> $01_{h}$ : 2 end identifiers | $01_{h}$ |


| Parameter data STX/ETX protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
|  | 9 | End identifier 1 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. $1 . . .255\left(01_{h} . . . F_{h}\right)$ | $03_{h}$ |
|  | 10 | End identifier 2 | ASCII value of the end character that is sent after a frame and marks the end of a transmission. <br> 1 ... $255\left(01_{h} . . . F_{h}\right)$ | $00_{h}$ |
|  | 11 | Operating mode | Operating mode of the interface $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. $01_{h}$ : 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. <br> Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $01_{h}$ |
|  | 12 | 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 ( $\mathbb{\square}$ 228). $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{h}$ : Signal $R(A) 5 V$ (open circuit monitoring); signal $R(B) O V$. In full duplex operation under RS422, open circuit monitoring is possible. $02_{\text {h }}$ Signal $R(A) 0 \mathrm{~V}$; signal $R(B) 5 \mathrm{~V}$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $01_{h}$ |


| Parameter data 3964(R) protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
| 02h | 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. |  |
|  |  | Length - process image output data | Length of output data (backplane bus communication); the values are specified by the system. Other values are not permissible. |  |
| $00_{h}$ | 0 | Diagnostics | Bits 5 ... 0: Reserved <br> Bit 6: Diagnostic alarm ( $0=$ inhibited; $1=$ enabled $)$ <br> Bit 7: Reserved <br> Other values are not permissible! | $00_{\text {h }}$ |
| $80_{\text {h }}$ | 0 | Baud rate | $00_{\mathrm{h}}$ : 9600 Baud $01_{\mathrm{h}}$ : 150 Baud 02 h : 300 Baud $03_{\mathrm{h}}$ : 600 Baud 04 h : 1200 Baud $05 \mathrm{~h}: 1800$ Baud 06h: 2400 Baud $07_{\mathrm{h}}$ : 4800 Baud $08_{\mathrm{h}}$ : 7200 Baud $09_{\mathrm{h}}$ : 9600 Baud $0 A_{h}$ : 14400 Baud $0 B_{h}$ : 19200 Baud $0 C_{h}: 38400$ Baud $0 \mathrm{D}_{\mathrm{h}}: 57600$ Baud 0E h : 76800 Baud $0 F_{h}: 115200$ Baud $10_{\mathrm{h}}$ : 109700 Baud | $00_{h}$ |
|  | 1 | Protocol | The protocol to be used. This setting influences the structure of the parameter data. $\begin{aligned} & 03_{\mathrm{h}}: 3964 \\ & 03_{\mathrm{h}}: 3964 \mathrm{R} \end{aligned}$ | $03_{h}$ |
|  | 2 | Data format | Bit 1/0 number of data bits $00_{b}: 5$ <br> $01_{b}: 6$ <br> $10_{b}: 7$ <br> $11_{\mathrm{b}}$ : 8 <br> Bit $3 / 2$ parity <br> $00_{b}$ : none <br> $01_{b}$ : odd <br> $10_{b}$ : even <br> $11_{b}$ : even <br> Bit 5/4 number of stop bits $01_{b}$ : 1 <br> $10_{\mathrm{b}}$ : 1.5 <br> $11_{b}$ : 2 <br> Bit 7/6 flow control <br> $00_{\mathrm{b}}$ : None <br> $01_{\mathrm{b}}$ : Hardware <br> $10_{b}$ : XON/XOFF | $13_{h}$ |
|  | 3 | ZNA | Time delay after command; waiting time to be complied with until the next transmit request is executed. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} . . . \text { FF }_{h}\right)$ | $00_{h}$ |
|  | 4 | Character delay time | 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). <br> 0 ... 65535 [20-ms steps] ( $00_{h} . . . \mathrm{FF}_{\mathrm{h}}$ ) | OAh |


| Parameter data 3964(R) protocol |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
| Data set |  | Name | Description/value | Lenze |
| No. | Byte |  |  |  |
|  | 5 | Acknowledgement time | The acknowledgement time defines the max. permissible interval until the partner is acknowledged while establishing/terminating a connection. $0 \text {... } 65535 \text { [20-ms steps] }\left(00_{h} . \therefore \text { FF }_{h}\right)$ | $0 A_{h}$ |
|  | 6 | Block wait time | Maximum time between the recognition of a requested frame (DLE) and the response. $\left.0 \ldots 255 \text { [ms] ( } 00_{\mathrm{h}} \ldots . . \mathrm{FF}_{\mathrm{h}}\right)$ | $0 \mathrm{~A}_{\mathrm{h}}$ |
|  | 7 | STX repetitions | Maximum number of times the module attempts to establish a connection. <br> $0 \ldots 255$ [ms] $\left(00_{h} . . . F_{h}\right)$ | 05h |
|  | 8 | DBL | If the block wait time has been exceeded, the number of STX repetitions for the requested protocol can be specified here. $0 \ldots 255 \text { [ms] }\left(00_{h} \ldots \mathrm{FF}_{\mathrm{h}}\right)$ | 06h |
|  | 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 the protocols 3964(R), both partners must have different priorities. <br> $00_{\mathrm{h}}$ : LOW <br> $01_{h}$ : HIGH | $00_{\text {h }}$ |
|  | 10 | Operating mode | Operating mode of the interface $00_{h}$ : Half duplex - two-wire operation (RS485); the data are transferred alternately in both directions between the communication partners. <br> $01_{h}$ : 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. Note: For the "Half duplex" parameterisation under RS485, no software data flow control is possible. | $01_{\text {h }}$ |
|  | 11 | 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 (■ 228). <br> $00_{h}$ : No pre-assignment of the receive path. This setting is only advisable for special drivers that are provided with a bus capability. <br> $01_{\mathrm{h}}$ : Signal R(A) 5 V (open circuit monitoring); signal $R(B) 0 \mathrm{~V}$. In full duplex operation under RS422, open circuit monitoring is possible. <br> $02_{h}$ : Signal $R(A) 0$ V; signal $R(B) 5 \mathrm{~V}$. This pre-assignment corresponds to the idle state (no transmitter active) in half duplex operation under RS485. Open circuit monitoring is not possible. | $01_{\text {h }}$ |

## 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.

| Name | Byte | Function | Default |
| :---: | :---: | :---: | :---: |
| ERR_A | 1 | ERR_A-diagnostics <br> Bit 0: Set in the case of module fault <br> Bit 1: Set in the case of internal error <br> Bit 2: Set in the case of external error (cable break) <br> Bit 3: Reserved <br> Bit 4: Set in the case of a missing external supply voltage <br> Bit 5, 6: Reserved <br> Bit 7: Set in the case of parameterisation error | 00 h |
| MODTYP | 1 | Module information <br> Byte 0: <br> Bit 3 ... 0: Module class (0111b: Gateway module) <br> Bit 4: channel information available <br> Bits 7 ... 4: 0 reserved | 17 h |
| ERR_C | 1 | ERR_A-diagnostics Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| ERR_D | 1 | ERR_D diagnostics <br> Bit 3 ... 0: Reserved <br> Bit 4: Set in the case of internal communication error <br> Bit 7 ... 5: Reserved | 00 h |
| CHTYP | 1 | Channel type <br> Bit 7 ... 0: Reserved | 00 h |
| NUMBIT | 1 | Number of diagnostic bits of the module per channel (here 08 h) | 08 h |
| NUMCH | 1 | Number of channels in module Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| CHERR | 1 | Bit 0: set in the event of an error of channel group 1 Bits 7 ... 10 (fixed) | 00 h |
| CHOERR | 8 | Channel-specific error: channel x : <br> Bits 3 ... 0:0 (fixed) <br> Bit 4 : set in the case of open circuit (only possible for RS422) <br> Bits 7... 5: 0 (fixed) | $00_{\text {h }}$ |
| CH1ERR ... CH7ERR | 8 | Bit 7 ... 0: Reserved | $00_{\text {h }}$ |
| DIAG_US | 4 | Value of $\mu$ s ticker when diagnostics occur Bytes 0 ... 3 | $00_{\text {h }}$ |

## 13 Modbus TCP communication

### 13.1 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.

## Protocols

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) |  |
| $\mathbf{1}$ | Physical layer (bit transmission) |  |

Telegram structure

| Layer 2 | Layer 3 | Layer 4 |  | Layer 7 |
| :---: | :---: | :---: | :---: | :---: |
| MAC/DLL | IP | TCP | API | $\ldots$ |
| 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 Modbus TCP bus coupler, the Modbus TCP protocol is used which will be explained in detail in the following section.


Fig. 13-1 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 ModbusTCP 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 ([] 719).

## 13.2

## 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 range
- 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.


Fig. 13-2 Example for addressing

### 13.2.1

Access to 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 definition

Usually, the access under Modbus takes place via the ranges $0 x, 1 x, 3 x$ and $4 x$.
$0 x$ And $1 x$ serve to access digital ranges and $3 x$ and $4 x$ 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:
$0 x$ : Bit for master output; access via function code $01_{h}, 05_{h}, 0 F_{h}$
$1 x$ : Bit range for master input; access via function code $02_{h}$
$3 x$ : Word range for master input; access via function code $04_{h}, 17_{h}$
$4 x$ : Word range for master output; access via function code $03_{h}, 06_{h}, 10_{h}, 16_{h}, 17_{h}$

| IN | $\begin{aligned} & \text { 1x: 0001h } \\ & \text { 1x: 0002h } \\ & \text { 1x: 0003h } \end{aligned}$ |  |  |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | Byte $0^{1}$ | Byte 1 | Byte ${ }^{\text {Bra }}$ | Byte 31 | Byte 41 | Byte5 | \|Byte <br> 1 | Byte 71 | Byte 81 | Byte, ${ }^{1}$ |  |
|  | 3 x : 0001h |  | 3 x : 0002h |  | 3x: 0003h |  | 3x: 0004h |  | 3x: 0005h |  |  |
| $\cdots$ | Byte 10 | Byte 11 | Byte 12 | \| ${ }^{\text {Byte } 11} 1$ | Byte 14 | Byte 15 | 11916 | Byte 17 | Byte 18 | Byte 19 | $\ldots$ |
|  | 3 x : 0006h |  | 3x: 0007h |  | 3x: 0008h |  | 3x: 0009h |  | 3x: 000Ah |  |  |
| OUT | 0x: 0001h <br> 0x: 0002h <br> 0x: 0003h |  |  |  |  |  |  |  |  |  |  |
|  | Byte 1 | Byte 1 ${ }^{1}$ | By\| ${ }^{\text {Byte }}$ | Byle 31 | Byte 4 | Byte ${ }^{\text {Bra }}$ | Byte 6 | Byte 7 | Byte 81 | Byte9 ${ }^{1}$ |  |
|  | 4x:0001h |  | 4x:0002h |  | 4x:0003h |  | 4x: 0004h |  | 4x: 0005h |  |  |
| $\ldots$ | Byte 10 | Byte 11 | Byte 12 | Byte 13 | Byte 14 | Byte 15 | Byte 16 16 | Byte 17 | Byte 18 | Byte 19 | $\ldots$ |
|  | 4x:0006h |  | 4x:0007h |  | 4x: 0008h |  | 4x:0009h |  | 4x: 000Ah |  |  |

Function codes

### 13.2.2 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 |
| :--- | :--- | :--- |
| $01_{h}$ | Read $n$ bits | Read $n$ bits from the master output range $0 x$ |
| $02_{h}$ | Read $n$ bits | Read $n$ bits from the master input range $1 x$ |
| $03_{h}$ | Read $n$ words | Read $n$ words from the master output range $4 x$ |
| $04_{h}$ | Read $n$ words | Read $n$ words from the master input range $3 x$ |
| $05_{h}$ | Write 1 bit | Write 1 bit in the master output range $0 x$ |
| $06_{h}$ | Write 1 word | Write 1 word in the master output range $4 x$ |
| $0 F_{h}$ | Write $n$ bits | Write $n$ bits in the master output range $0 x$ |
| $10_{h}$ | Write $n$ words | Write $n$ words in the master output range $4 x$ |
| $16_{h}$ | Mask 1 word | Mask 1 word in the master output range $4 x$ |
| $17_{h}$ | Write $n$ words and read m words | Write $n$ words in the master output range $4 x$ and <br> the response contains m read words of the master <br> input range $3 x$ |

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 $80_{h}$. If no error has occurred, the function code is returned.
Coupler response:
Function code OR $80_{h} \rightarrow$ error \& error number
Function code $\rightarrow \mathrm{OK}$
In the event of an error you will receive additionally an error number in another byte. Here, the following error numbers exist:
$01_{h}$ : Function number is not supported
$02_{\mathrm{h}}$ : Faulty addressing
$03_{h}$ : Faulty data
$04_{h}$ : System SLIO bus is not initialised
$07_{h}$ : General error

## Read n bits

Code $01_{h}$ : Read $n$ bits of master output range $0 x$.
Code $02_{h}$ : Read $n$ bits of master input range $1 x$.
Command frame:

| Modbus TCP header |  |  | Slave address | Function <br> code | Address <br> 1. bit | Number <br> Bits |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :---: | :---: | :---: | :---: |
| x | x | $\mathbf{0}$ | $\mathbf{0}$ | 0 | 6 |  |  |  |
| 6 bytes |  |  |  | 1 byte | 1 byte | 1 word | 1 word |  |

Response message:


## Read n words

$03_{h}$ : Read $n$ words of the master output range $4 x$.
$04_{\mathrm{h}}$ :Read n words of the master input range 3 x .
Command frame:

| Modbus TCP header |  |  | Slave address | Function <br> code | Address <br> Word | Number <br> Words |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| x | x | 0 | 0 | 0 | 6 |  | 1 byte | 1 word | 1 word |
| 6 bytes |  |  | 1 byte | 1 by |  |  |  |  |  |

Response message:

| Modbus TCP header |  |  |  |  | Slave address | Function code | Number of read bytes | Data <br> 1. word | Data <br> 2. word | ... |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| X | X | 0 | 0 | 0 |  |  |  |  |  |  |
| 6 bytes |  |  |  |  | 1 byte | 1 byte | 1 word | 1 word |  |  |
|  |  |  |  |  |  |  |  | max. 126 words |  |  |

## Write 1 bit

Code $05_{\mathrm{h}}$ : Write 1 bit in master output range $0 x$.
A status is changed under "Status bit" with the following values:
"Status bit" $=0000_{h} \rightarrow$ bit $=0$
"Status bit" $=$ FFOO $_{h} \rightarrow$ bit $=1$
Command frame:


Response message:

| Modbus TCP header |  |  |  |  |  | Slave address | Function code | Address Bit | Status Bit |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| X | X | 0 | 0 | 0 | 6 |  |  |  |  |  |
| 6 bytes |  |  |  |  |  | 1 byte | 1 byte | 1 word | 1 word |  |

## Write 1 word

Code $06_{h}$ : Write 1 word in master output range $4 x$.
Command frame:

| Modbus TCP header |  |  | Slave address | Function <br> code | Address <br> Word | Value <br> Word |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| x | x | $\mathbf{0}$ | 0 | 0 | 6 |  |  |  |
| 6 bytes |  |  | 1 byte | 1 byte | 1 word | 1 word |  |  |

Response message:


## Write n bits

Code $0 F_{h}$ : Write $n$ bits in master output range $0 x$
Please observe that the number of bits must be additionally given in bits.
Command frame:


Response message:

| Modbus TCP header |  |  |  | Slave <br> address | Function <br> code | Address <br> $1 . b i t$ | Number <br> Bits |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| x | x | 0 | 0 | 0 | 6 |  |  |  |
| 6 bytes |  |  | 1 byte | 1 byte | 1 word | 1 word |  |  |

## Write n words

Code $10_{h}$ : Write $n$ words in master output range.
Command frame:


Response message:

| Modbus TCP header |  |  |  |  |  | Slave address | Function code | Address <br> 1. word | Number Words |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| X | x | 0 | 0 | 0 | 6 |  |  |  |  |
| 6 bytes |  |  |  |  |  | 1 byte | 1 byte | 1 word | 1 word |

## Mask 1 word

Code $16_{h}$ : This function serves to mask a word in the master output area 4 x .
Command frame:


Response message:

| Modbus TCP header |  |  |  | Slave <br> address | Function <br> code | Address <br> Word | AND <br> Mask | OR <br> Mask |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| x | x | 0 | 0 | 0 | 8 |  |  |  |  |
| 6 bytes |  |  | 1 byte | 1 byte | 1 word | 1 word | 1 word |  |  |

## Write n words and read m words

Code $17_{h}$ : This function serves to write $n$ words in the master output range $4 x$ via a request and read n words of the master input range 3 x .
Command frame:

| Modbus TCP header |  |  |  |  | Slave address | Function code | Read address |  | Write address | Write number of bytes | Write data 1. byte | Write data 2. byte | ... |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| x | x | 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |
| 6 bytes |  |  |  |  | 1 byte | 1 byte | 1 word | 1 word | 1 word | 1 byte | 1 word | 1 word |  |
|  |  |  |  |  |  |  |  |  |  |  | max. 122 words |  |  |

Response message:


### 13.2.3 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.

### 13.2.4 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 |
| :---: | :---: |
| $\begin{aligned} & 0 x / 1 x: \\ & 4000_{h} \ldots 403 F_{h} \end{aligned}$ | Bit access to process alarm status: <br> $0 x / 1 x$ : 4000: Process alarm status Modbus TCP slot 0 0x/1x: 4001: Process alarm status Modbus TCP slot 1 ... <br> $0 x / 1 x$ : 403F: Process alarm status Modbus TCP slot 63 |
| $\begin{aligned} & \text { 0x/1x: } \\ & 5000_{h} \ldots 503 F_{h} \end{aligned}$ | Bit access to diagnostic alarm status: <br> 0x/1x: 5000: Diagnostic alarm status Modbus TCP slot 0 0x/1x: 5001: Diagnostic alarm status Modbus TCP slot 1 ... <br> $0 x / 1 x$ : 503F: Diagnostic alarm status Modbus TCP slot 63 |
| $3 \mathrm{x}: 4000_{\mathrm{h}} \ldots 41 \mathrm{FF}_{\mathrm{h}}$ | Word access to process alarm data 3x: 4000h ... 4007h: Modbus TCP slot 0 3x: 4008h ... 400F $h$ : Modbus TCP slot 1 ... 3x: 41F8 ${ }_{h} . .41 \mathrm{FF}_{\mathrm{h}}$ : Modbus TCP slot 63 |
| 3x: 4000 ${ }_{\text {h }} \ldots 4007^{\text {h }}$ | 16 bytes of process alarm data of Modbus TCP slot 0 |
| 3x: 4008h $\ldots 4^{400 F_{h}}$ | 16 bytes of process alarm data of Modbus TCP slot 1 |
| 3x: 4010 $\mathrm{h} . . .4017_{h}$ | 16 bytes of process alarm data of Modbus TCP slot 2 |
| ... | ... |
|  | 16 bytes of process alarm data of Modbus TCP slot 63 |
| $3 \mathrm{x}: 5000_{\mathrm{h}} \ldots{500 \mathrm{~F}_{\mathrm{h}}}$ | 32 bytes of diagnostic alarm data of Modbus TCP slot 0 |
|  | 32 bytes of diagnostic alarm data of Modbus TCP slot 1 |
| ... |  |
| $3 \mathrm{x}: 53 \mathrm{FO} \mathrm{h}_{\mathrm{h}} \ldots 53 \mathrm{FF}_{\mathrm{h}}$ | 32 bytes of diagnostic alarm data of Modbus TCP slot 63 |

### 13.3 Possible access operations to the EPM-S160 Modbus TCP bus coupler module

### 13.3.1 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.

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 |
| :---: | :--- | :--- |
|  | Module on 1. slot | Information <br> Data <br> Parameter |
| 1 | Module on 2. slot | Information <br> Data <br> Parameter |
| $\ldots$. | Last module | Information <br> Data <br> Parameter |
| $n$ |  | Communication settings <br> Security settings <br> 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.

## 14 Maintenance

### 14.1 Regular checks

The system is maintenance-free. Nevertheless, visual inspections must be carried out at regular intervals which must not be too long, depending on the ambient conditions.
Please check the following:

- Does the environment of the system still meet the operating conditions specified in the Technical data?
- Is the heat dissipation impeded by dust or dirt?
- Are the mechanical and electrical connections still okay?


### 14.2 Cleaning

## STOP Stop!

Sensitive surfaces and components
The system can be damaged if it is not appropriately cleaned.
Possible consequences:

- Housings will get scratched or dull if cleaning agents containing alcohol, solvents or abrasives are used.
- Electrical components will be damaged if humidity enters in the housing.

Protective measures:

- Deenergise the complete system before cleaning.
- Wipe the housing using a clean, lint-free, soft cloth. For stubborn dirt, dampen the cloth with water and an ordinary household cleaning agent.


## 14.3

## Repair

### 14.3.1 Replacing the electronic module/main supply

## 1 Note!

If an I/O compound module is defective, just replace its electronic module. Leave the base module and the wiring unchanged.
The input fuse of the power supply module is located in the electronic module. If this fuse has tripped, the electronic module needs to be replaced. The same applies to the main supply which is integrated in bus coupler modules.


Fig. 14-1 Replacing the electronic module
How to proceed:

1. Press the locking button underneath the electronic module (or, in the case of bus coupler modules, the main supply, respectively) and remove the electronic module from the base module at the same time.
2. Slide a new electronic module of the same type on the base module until it locks into place.

## 15 Appendix

### 15.1 Serial process interfacing

With the EPM-S640 and EPM-S650 I/O compound modules a serial process interfacing to different target or source systems can be implemented.

EPM-S640: Serial process interfacing via RS232 interface
EPM-S650: Serial process interfacing via RS422 or RS485 interface

## Serial transmission of a character

The point-to-point coupling between two communication partners is the simplest form of the exchange of information. The EPM-S640 I/O compound module provides the interface between a higher-level system (control) and a serially connected communication partner. In the case of serial data transmission, the individual bits of a byte of a piece of information to be transmitted are transferred successively in a defined sequence.

## Character frame

The bidirectional data exchange makes a distinction between half duplex and full duplex operation. In half duplex operation, data are either sent or received at a certain time. A simultaneous data exchange can only be effected in full duplex operation.
Each character to be transmitted is preceded by a synchronisation pulse as start bit. The end of the character transfer is set by the stop bit. In addition to the start and stop bit, further parameterisable agreements between the communication partners are required for a serial data transmission.

This character frame consists of the following elements:

- Transmission speed (baud rate)
- Character delay time and acknowledgement time
- Parity
- Number of data bits
- Number of stop bits


## Communication

During the transmission, data which are written to the corresponding output area by a higher-level system via the backplane bus are written to the output buffer and are output via the interface from there. If the communication processor receives data via the interface, these data are stored in a ring buffer and are entered in the input area of the higher-level system via the backplane bus.


### 15.1.1

## Protocols

The following protocols are supported:

- ASCII
- STX/ETX
- 3964(R)


## Note!

Information on the parameterisation can be found in the corresponding chapters regarding the fieldbuses.

ASCII protocol: Data communication via ASCII is a simple form of data exchange and can be compared to a multicast/broadcast function.

The logical separation of the telegrams is effected via the character delay time. Within this time, the transmitter must have sent its telegram to the receiver. A telegram is only transferred to the higher-level system if it has been received completely. As long as the "Time after request" has not elapsed, no new transmit request is accepted.
These two time specifications serve to establish a simple serial communication.
Since, apart from the use of the parity bit, no further backup measures are implemented for ASCII transmissions, the data transfer may be very efficient, however, it is not secured. With the parity the inversion of one bit within a character is secured. If several bits of a character are inverted, this error can no longer be detected.
STX/ETX protocol: STX/ETX is a simple protocol with headers and trailers. It is used for transmitting ASCII characters $\left(20_{h} \ldots 7 \mathrm{~F}_{\mathrm{h}}\right)$. This is done without block checks (BCC). If data are to be read in by the peripherals, STX (Start of Text) must be available as initial character, followed by the characters to be transmitted. An ETX (End of Text) must be inserted as the terminating character. The user data, i.e. all characters between STX and ETX, are transmitted to the control after the ETX terminating character has been received. When data are sent from the control to a peripheral device, the user data are transmitted to the EMP-S640 or EPM-S650 I/O compound module and are then transmitted to the communication partner with STX as initial character and ETX as final character.


Fig. 15-1 Telegram structure
Up to two initial characters and final characters can be freely defined. Here, a "time delay after command" can be selected for the transmitter as well.

3964(R) protocol: 3964(R) manages the data transfer of a point-to-point coupling between the EPM-S640 or EPM-S650 I/O compound module and a communication partner.

During the data transfer, control characters are added to the user data. These enable the communication partner to check whether the data has arrived completely and correctly.
The following control characters are evaluated:

- STX (Start of Text)
- DLE (Data Link Escape)
- ETX (End of Text)
- BCC (Block Check Character; only for 3964R)
- NAK (Negative Acknowledge)


## 1 Note!

When a DLE is transmitted as information character, it is distinguished from the DLE control character by being sent twice while establishing/terminating a connection (DLE duplication). The receiver undoes the DLE duplication. With 3964(R), the communication partner must be assigned with a lower priority. When both communication partners place a transmit request at the same time, the partner with lower priority will delay its transmit request.


Fig. 15-2 Sequence
You can transfer a maximum of 250 bytes per frame.
Time-out times: Acknowledgement time is monitored between STX and DLE, as well as between BCC and DLE. Character delay time is monitored during the entire frame reception. After the acknowledgement time has elapsed after STX, STX is retransmitted and after 5 attempts a NAK is sent and the connection establishment is aborted. The same happens when after a STX a NAK or any character are received. After the acknowledgement time has elapsed after the frame (after BCC byte) or when a character unequal to DLE is received, the connection establishment and the frame will be repeated. Here, also 5 attempts are made, afterwards a NAK is sent and the transmission is aborted.

Passive operation: When the driver waits for the connection establishment and receives a character unequal to STS, a NAK is sent. When a NAK character is received, the driver does not send a respond. If the character delay time is exceeded while receiving a character, a NAK is sent and a renewed connection establishment is waited for. When the driver is not ready yet when the STX has received, a NAK is sent.
Block Check Character (BCC byte): For further backup, a block check character is added to the end of the frame of 3964 . The BCC byte is created by a XOR connection via the data of the entire frame including DLE/ETX. When a BCC byte is received that differs from the calculated one, a NAK is sent instead of the DLE.

Initialisation conflict: If two partners simultaneously attempt to establish a connection within the acknowledgement time, the partner with the lower priority sends the DLE and changes to receive mode.

Data Link Escape (DLE character): The DLE character in a frame is duplicated by the driver, i.e., DLE/DLE is sent. During the reception, duplicated DLEs are stored as a single DLE in the buffer. The frame always terminates with the combination DLE/ETX/BCC (only for 3964R).
Control codes:

- $02 \mathrm{~h}=\mathrm{STX}$
- $03 \mathrm{~h}=\mathrm{ETX}$
- 10h = DLE
- $15 \mathrm{~h}=$ NAK


### 15.1.2 I/O range

Depending on the higher-level fieldbus system, the I/O compound module assigns the following number of bytes in the address range for the input and output, respectively.

- PROFIBUS: 8 bytes, 20 bytes or 60 bytes (selectable)
- PROFINET: 20 bytes or 60 bytes (selectable)
- CANopen: 8 bytes
- EtherCAT: 60 bytes
- DeviceNET: 60 bytes
- ModbusTCP: 60 bytes


### 15.1.3 Principle of backplane bus communication

## Transmit data

During transmission, the higher-level system enters the data to be output in the output area and transfers them to the I/O compound module together with the control byte. At every telegram, the I/O compound module responds with an acknowledgement by copying bits 3 ... 0 of byte 0 of the output area to bits 7 ... 4 of byte 0 of the input area, or by returning a corresponding status message via this byte. Depending on the length of the data to be transmitted, the telegram is to be transferred to the I/O compound module in one or several fragments. In the case of the fragmented transmission, each fragment is acknowledged by the I/O compound module.
Transmission without fragmentation:


Example of transmission without fragmentation: IO size $=60$ bytes, length $=40$ bytes


## Transmission with fragmentation:

In the case of the fragmented transmission, the number of user data and already part of the user data are transmitted with the 1. telegram (header). Then the fragment telegrams follow.
Procedure: Write 1. telegram $\rightarrow$ write fragment $\rightarrow$ write last fragment
Calculation of the number of fragments: number of fragments $=($ length +3$) /($ IO_Size 1)


| Write fragment |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Higher-level system |  |  | EPM-S640/EPM-S650 |  | Description |
| Byte | Function |  | Byte | Function |  |
| 0 | Control byte |  |  |  | Bits 3...0: <br> $0_{h} \ldots 7_{h}=$ fragment number <br> $8_{h}=$ idle state, no data available. <br> $\mathrm{B}_{\mathrm{h}}=$ carry out a reset on the EPM-S6x0. <br> Bits 7...4: <br> Reserved for reception. |
| 1 ... n-1 | User data |  |  |  | User data; $n=$ number of the bytes assigned in the address range (IO size) |
|  |  | $\rightarrow$ |  |  |  |
|  |  | $\leftarrow$ | 0 | Acknowledgement / status | Bits 3...0: <br> Reserved for reception. <br> Bits 7...4: <br> $0_{h} \ldots 7_{h}=$ acknowledgement fragment number <br> $8_{h}$ : acknowledgement, idle state <br> $9_{h}$ : fragmented transmission started. <br> $\mathrm{C}_{\mathrm{h}}$ : status, reset was carried out on <br> EPM-S6x0. <br> $D_{h}$ : status, the length specified is invalid. <br> $\mathrm{E}_{\mathrm{h}}$ : status, EPM-S6x0 communication error, partner is not responding. |
| Write last fragment |  |  |  |  |  |
| Higher-level system |  | $\begin{aligned} & \rightarrow \\ & \leftarrow \end{aligned}$ | EPM-S640/EPM-S650 |  | Description |
| Byte | Function |  | Byte | Function |  |
| 0 | Control byte |  |  |  | Bits 3...0: <br> $8_{h}=$ idle state, no data available. <br> $A_{h}=$ transmission of last fragment. <br> $B_{h}=$ carry out a reset on the EPM-S6x0. <br> Bits 7...4: <br> Reserved for reception. |
| 1 ... n-1 | User data |  |  |  | User data; $\mathrm{n}=$ number of the bytes assigned in the address range (IO size) |
|  |  |  | 0 | Acknowledgement / status | Bits 3...0: <br> Reserved for reception. <br> Bits 7...4: <br> $8_{h}$ : acknowledgement, idle state <br> $A_{h}$ : acknowledgement, last fragment received. <br> $C_{h}$ : status, reset was carried out on EPM-S6x0. <br> $D_{h}$ : status, the length specified is invalid. <br> $\mathrm{E}_{\mathrm{h}}$ : status, EPM-S6x0 communication error, partner is not responding. |

Example of transmission with fragmentation: IO size = 16 bytes, length $=50$ bytes

| Header Higher-level system |  |  | EPM-S640/EPM-S650 |  | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  |  |  |  |
| Byte | Value |  | Byte | Value |  |
| 0 | 09h |  |  |  | Command: start transmission with fragmentation. |
| 1 | $00_{\text {h }}$ |  |  |  | Telegram information |
| 2 | 00 h |  |  |  | User data length high byte |
| 3 | 32 h |  |  |  | User data length low byte |
| $4 . .15$ | x |  |  |  | User data byte $0 . . .11$ |
|  |  | $\rightarrow$ |  |  |  |
|  |  | $\leftarrow$ | 0 | $90_{\text {h }}$ | Status: fragmented transmission started. |


| 1. fragment <br> Higher-level system |  |  |  |  | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $\rightarrow$$\leftarrow$ | EPM-S640/EPM-S650 |  |  |
| Byte | Value |  | Byte | Value |  |
| 0 | 00 h |  |  |  | Fragment number |
| $1 . . .15$ | x |  |  |  | User data byte $12 . . .26$ |
|  |  |  |  |  |  |
|  |  |  | 0 | $00{ }_{\text {h }}$ | Acknowledgement $00{ }_{\text {h }}$ |
|  |  |  |  |  |  |
| 2. fragment <br> Higher-level system |  |  | EPM-S640/EPM-S650 |  | Description |
|  |  | $\begin{aligned} & \rightarrow \\ & \leftarrow \end{aligned}$ |  |  |  |
| Byte | Value |  | Byte | Value |  |
| 0 | $01_{\text {h }}$ |  |  |  | Fragment number |
| $1 \text {... } 15$ | x |  |  |  | User data byte 27 ... 41 |
|  |  |  |  |  |  |
|  |  |  | 0 | $10_{\text {h }}$ | Acknowledgement $01_{h}$ Note: At the first fragment, the serial number 01 is transferred. In the acknowledgement, this information is returned in a mirrored manner. |


| Last fragment Higher-level system |  |  | EPM-S640/EPM-S650 |  | Description |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Byte | Value |  | Byte | Value |  |
| 0 | $0 A_{h}$ | $\begin{aligned} & \rightarrow \\ & \leftarrow \end{aligned}$ |  |  | Fragment number |
| 1 ... 8 | x |  |  |  | User data byte 42 ... 49 |
| $11 . .15$ | - |  |  |  | Not used |
|  |  |  |  |  |  |
|  |  |  | 0 | $\mathrm{AO}_{\mathrm{h}}$ | Acknowledgement of last fragment |

## Receive data

During reception, the data received are entered automatically in the input area of the higher-level system by the I/O compound module. Depending on the length of the data received, the telegram is transmitted to the higher-level system in one or several fragments.

## Reception without fragmentation:



Example of reception without fragmentation: IO size $=60$ bytes, length $=40$ bytes


## Received with fragmentation:

(Calculation of the number of frequencies: number of fragments $=$ length $+7 / 10$ _Size -1


Example of reception with fragmentation: 1 O size $=16$ bytes, length $=40$ bytes




## 16 Index

## A

Air humidity, 239
Ambient conditions

- Climatic, 239
- Mechanical, 239

Analog inputs, Status request, 356
Analog outputs, \#Status request, 356
Application as directed, 22
Approval, 238
Attaching the shield, 248

## B

Baud rate, 31, 57
Bus coupler module

- CANopen, 29
- Connecting the supply, 246, 247
- DeviceNet, 54
- Dismounting, 244
- elements, 25
- EtherCAT, 44
- Modbus TCP, 63
- Mounting, 242, 243
- PROFIBUS, 37
- PROFINET, 49
- Replacing a fuse, 726

Busbar, 244, 248
Busbar support, 244, 248

## C

CAN data telegram, structure, 250

## CANopen

- Baud rate, 31
- Bus coupler module, 29
- Communication, 250
- Node address, 31

Cleaning, 725
COB-ID, 250
CoDeSys, 393, 621

## Communication

- CANopen, 250
- DeviceNet, 557
- EtherCAT, 480
- Modbus TCP, 713
- PROFIBUS, 386
- PROFINET, 613

Communication Object Identifier, 250

Communication phases, 252
Conformity, CE, 238
Connecting the supply, 246, 247
Connection, EtherCAT, 47 , 52, 65
Contact cover, 242, 243
Counter, querying the status, 357
Cover, 242, 243

## D

Data transfer, 272
Definition of notes used, 15
Definitions, 16
Degree of protection, 238
Device protection, 238
Device status

- of the heartbeat producer, 345
- of the I/O system IP20, 343
- of the slave, 343


## DeviceNet

- baud rate, 57
- bus coupler module, 54
- communication, 557
- node address, 57

Digital inputs, Status request, 356
Digital inputs with time stamp, Query status, 357

Digital modules

- Digital input, setting parameters, 284
- Digital output, setting parameters, 284
Digital outputs, Status request, 356

Digital outputs with PWM, Status request, 359

Digital outputs with time stamp, Query status, 358

Dimensions, 241
Disposal, 20

## E

Electrical installation, 245

- Mounting steps, 246, 247, 248
- Wiring according to EMC, 245

Electrical isolation, 238
Electronic supply, 246, 247
EMC, 238

- Assembly, 245
- Earthing, 245
- Shielding, 245

Emergency telegram, 349, 554
EPM-S110, 29
EPM-S120, 37
EPM-S130, 44
EPM-S140, 49
EPM-S150, 54
EPM-S160, 63
EPM-S200, 67
EPM-S201, 70
EPM-S202, 73
EPM-S203, 76
EPM-S204, 67
EPM-S205, 70
EPM-S206, 73
EPM-S207, 79, 326, 459, 521, 599
, 692
EPM-S300, 83
EPM-S301, 87
EPM-S302, 91
EPM-S303, 83
EPM-S304, 87

EPM-S305, 91
EPM-S306, 95
EPM-S308, 106
EPM-S309, 98
EPM-S310, 101, 326, 461, 521 , 599, 694

EPM-S400, 111, 286, 394, 486, 563, 626

EPM-S401, 114, 287, 396, 487, 564, 628

EPM-S402, 117, 288, 398, 488 , 565, 630

EPM-S403, 121, 289, 400, 489, 566, 632

EPM-S404, 148, 299, 416, 498 , 575, 648

EPM-S405, 152, 306, 423, 504, 581, 655

EPM-S406, 125, 290, 402, 490, 567, 634

EPM-S408, 128, 292, 405, 492, 569, 637

EPM-S500, 131, 294, 408, 494, 571, 640

EPM-S501, 135, 295, 410, 495, 572, 642

EPM-S502, 139, 296, 412, 496, 573, 644

EPM-S503, 143, 297, 414, 497, 574, 646

EPM-S600, 156, 311, 428, 507, 584, 660

EPM-S601, 173, 315, 436, 511, 588, 668

EPM-S602, 189, 318, 443, 514, 591, 676

EPM-S603, 204, 321, 450, 516, 594, 683

EPM-S604, 211, 323, 454, 518, 596, 687

EPM-S620, 215, 327, 464, 522, 600, 697

EPM-S640, 219, 328, 467, 523 , 601, 700

EPM-S650, 224, 334, 473, 529, 607, 706
EPM-S701, 230
EPM-S702, 232

EPM-S910, 235
EPM-S911, 236
EPM-S912, 237
Error Response, 277

## EtherCAT

- Bus coupler module, 44
- Communication, 480


## F

Fieldbus

- Baud rate, 31
- baud rate, 57
- Node address, 31
- node address, 39, 57

Fuses, 246, 247, 726

## G

General data, 238

## H

Heartbeat, 344
Heartbeat Producer, 344

## I

I/O compound module

- Analog I/O, setting parameters, 286 , 394, 486, 563, 626
- Analog I/Os, parameterising, 285
- Attaching the shield, 248
- counter, parameterising, 310, 428, 507,584, 660
- digital I/Os, parameterising, 284
- Dismounting, 244
- elements, 25
- encoder evaluation, parameterising, 323, 454, 518, 596, 687
- Mounting, 242, 243
- Replacing the electronic module, 726
- technology module communication, parameterising, 327, 464, 522, 600, 697
- temperature measurement, parameterising, 299, 416, 498, 575, 648
- Time stamp, parameterising, 326 , 459,521,599, 692
I/O supply, 246, 247
Identifier, 250

Index, 277
Installation, CE-typical drive system

- Assembly, 245
- Earthing, 245
- Shielding, 245

Installation, electrical, 245

- Mounting steps, 246, 247, 248

Installation, mechanical, 240

- Dimensions, 241
- Mounting steps, 242, 243, 244

Instruction code, 277
Insulation resistance, 238
Insulation voltage, 238

## L

Loading the default setting, 341

## M

Maintenance, 725

- Cleaning, 725
- Regular checks, 725
- Repair, 726

Mechanical installation, 240

- Dimensions, 241
- Mounting steps, 242, 243, 244


## Modbus TCP

- bus coupler module, 63
- communication, 713

Module identifiers, reading out, 355
Monitoring, 346, 535

- Analog outputs, 348

Mounting clearances, 241
Mounting conditions, 239
Mounting place, 239
Mounting position, 239

## Mounting steps

- Block mounting, 243
- Busbar, 244
- Dismounting, 244
- Standard, 242


## N

Network management (NMT), 252
Network, CAN, Communication
phases, 252
Node address, 31, 39, 57

Node Guarding, 342
Node ID, 31, 57
Node-ID, 39
Noise emission, 238
Noise immunity, 238
Notes, definition, 15

## 0

Operating conditions, 239
Operating temperature, 239

## P

Parameter data, 278

- Meaning for digital modules, 284
- storing for digital modules, 284
- telegram structure, 276

Parameter setting

- Analog I/O, 286, 394, 486, 563, 626
- counter, $310,428,507,584,660$
- Digital modules, 284

Display of the parameter data, 284 Meaning of the parameter data, 284

- encoder evaluation, 323, 454, 518, 596, 687
- temperature measurement, 299, 416, 498, 575, 648
- Time stamp, 326, 459, 521, 599, 692


## Parameterisation

- digital I/Os, 284
- technology module communication, 327, 464, 522, 600, 697
PDO mapping, 258
Pollution, 239
Power distributor module, 235, 236, 237
- elements, 26

Power supply module, 230, 232

- Connecting the supply, 247
- Dismounting, 244
- elements, 26
- Mounting, 242, 243
- Replacing a fuse, 726

Process data, transmission mode for, 256
Process data objects, Identifier, 255 , 256
Process data telegram, 254
Product description, 21

- application as directed, 22
- CANopen bus coupler module, 29
- DeviceNet bus coupler module, 54
- EtherCAT bus coupler module, 44
- Modbus TCP bus coupler module, 63
- PROFIBUS bus coupler module, 37
- PROFINET bus coupler module, 49


## PROFIBUS

- Bus coupler module, 37
- communication, 386
- node address, 39

PROFIBUS-DP-V0, 386
PROFIBUS-DP-V1, 386
PROFINET

- bus coupler module, 49
- communication, 613

Protection of persons, 238
Protective measures, 238

## R

Read Request, 277
Read Response, 277
Reading a parameter, 280
Regular checks, 725
Repair, 726
Replacing the electronic module, 726
Replacing the main supply, 726
RJ45 socket, EtherCAT connection, 47 , 52, 65

## $S$

Safety instructions, 18

- application as directed, 22
- definition, 15
- layout, 15

Shield connection, 244, 248
Shield connection terminal, 244, 248

Shielding, EMC, 245
Shock resistance, 239
Storage temperature, 239
Store, 340
Subindex, 277
Sync telegram, for cyclic process data, 257

## T

Technical data, 238

- General data, 238
- Operating conditions, 239

Time monitoring, 346
Transmitting parameter data, 276, 484
Transmitting PDO, 254, 482
Transmitting process data, 254, 482
Transmitting SDO, 276, 484
Troubleshooting and fault elimination, 249

- Monitoring, 346, 535


## U

User data, 254

## V

Validity, documentation, 12
Vibration resistance, 239

## W

Wiring according to EMC, 245
Write Request, 277
Write Response, 277
Writing a parameter, 279

兹 Lenze Automation GmbH Postfach 1013 52，D－31763 Hameln Hans－Lenze－Str．1，D－31855 Aerzen Germany

| Service | Lenze Service GmbH |
| :---: | :---: |
|  | Breslauer Straße 3，D－32699 Extertal |
|  | Germany |
| 㐌 | 0080002446877 （24 h helpline） |
| 回 | ＋49 5154 82－1112 |
|  | service＠lenze．com |


[^0]:    * Calculation example: $5 \times$ EPM-S400 and $5 \times$ EPM - S501 $\rightarrow(5 \times 6$ bytes $)+(5 \times 10$ bytes $)=80$ bytes; i.e. a reserve of 144 bytes

[^1]:    "-": Not relevant; "x Hz": Blinking with x Hz; "[an]": Option; "Puls": Pulsating

[^2]:    A Displays for module status
    B Terminals
    1 ... 8 Connection number

[^3]:    (1) Count value = comparison value $\rightarrow$ Pulse of the parameterised period is output, hysteresis is activated, and the counting direction is stored
    (2) Exiting of the hysteresis range against the counting direction stored $\rightarrow$ Pulse of the parameterised pulse duration is output and hysteresis is deactivated
    (3) Count value = comparison value $\rightarrow$ Pulse of the parameterised pulse duration is output, hysteresis is activated, and the counting direction is stored
    (4) Hysteresis range is exited without change of the counting direction $\rightarrow$ Hysteresis is deactivated
    (5) Count value = comparison value $\rightarrow$ Pulse of the parameterised pulse duration is output, hysteresis is activated, and the counting direction is stored
    (6) Count value = comparison value and hysteresis active $\rightarrow$ No pulse
    (7) Exiting of the hysteresis range against the counting direction stored $\rightarrow$ Pulse of the parameterised pulse duration is output and hysteresis is deactivated

[^4]:    * Due to the high-energy single current pulses, a surge requires a suitable external connection with lightning protection elements like for instance lightning conductors and overvoltage arresters.

[^5]:    $\mathrm{xx}=00_{\mathrm{h}} \quad$ With this assignment, all controllers connected are addressed by the telegram. All controllers can change their status at the same time.
    $x x=$ node ID If a node address is indicated, the status will only be changed for the controller addressed.

[^6]:    * The bits are set until reset with RES_SET (bit 6 control word)

[^7]:    * The bits are set until reset with RES_SET (bit 6 control word)

