SIEMENS

SIMATIC

CN 4100
CNET - Engineering Tool

Operating Manual

Cybersecurity information 1

Basic information on the CNET tool 2

Useful information 3

Getting started 4

Configuration 5

Engineering 6

Operation 7

Functions 8

Block libraries A

06/2024 (V3.0)
A5E50871394-AE

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens Aktiengesellschaft. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens Aktiengesellschaft
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E50871394-AE
Ⓟ 05/2024 Subject to change

Copyright © Siemens 2024.
All rights reserved

Table of contents

1 Cybersecurity information 13

2 Basic information on the CNET tool 15

3 Useful information 17

3.1 Overview of the documentation 17

3.2 Operation of the user interface of the CNET tool 18

4 Getting started 21

4.1 Getting started in CN 4100 21

4.2 Useful information about setup 21

4.3 Reading out the MAC address / assigning an IP address 22

5 Configuration 25

5.1 Using CN 4100 25

5.2 Useful information on configuration 25

5.3 Creating and configuring hardware 28

5.4 Update firmware and BIOS (of non-redundant CPUs) 29

5.5 Updating firmware and BIOS (of redundant CPUs) 29

6 Engineering 31

6.1 Working with CFC or backups 31

6.2 Useful information on engineering 32

6.3 Creating a program 33

6.4 Creating monitoring data 35

6.5 Using a project backup 36

6.6 Upgrading a project 36

6.7 Upgrade CNET (without project backup) 38

7 Operation 41

7.1 Working with the CPU online 41

7.2 Useful information on operation 41

7.3 Using system diagnostics 43

7.4 Using a watch table 44

7.5 Controlling hardware 45

8 Functions 47

8.1 Requirements 47

8.1.1 Requirements ... 47
8.1.2 Compatibility list ... 48
8.1.3 User and password ... 48

8.2 Step-by-step instructions ... 49
8.2.1 Customizing the window size ... 49
8.2.2 Showing / hiding a window ... 50
8.2.3 Navigating in dialogs ... 50

8.3 Menu ... 51
8.3.1 Menu ... 51
8.3.2 Step-by-step instructions ... 52
8.3.2.1 Creating a backup ... 52
8.3.2.2 Restoring a backup ... 54
8.3.2.3 Emptying a project ... 55
8.3.2.4 Viewing the CNET and plugin version ... 57
8.3.2.5 Displaying documentation ... 57
8.3.2.6 Displaying keyboard shortcuts ... 58
8.3.2.7 Having OSS components displayed ... 59
8.3.2.8 Terminating CNET ... 59
8.3.2.9 Navigating to an instance ... 60
8.3.2.10 Showing defaults ... 61
8.3.2.11 Detaching a section from the parameter area ... 62

8.4 Navigation area ... 64
8.4.1 Navigation area ... 64
8.4.2 Step-by-step instructions ... 65
8.4.2.1 Moving an object in the navigation area ... 65
8.4.3 CN 4100 ... 66
8.4.3.1 CN 4100 ... 66
8.4.3.2 Online mode ... 67
8.4.3.3 Offline mode ... 67
8.4.3.4 Device Cockpit ... 67
8.4.3.5 Diagnostics data ... 68
8.4.3.6 Update device state ... 70
8.4.3.7 Time Server Status ... 72
8.4.3.8 Ethernet Trace ... 73
8.4.3.9 Automatic Status Update ... 74
8.4.3.10 Firmware update ... 75
8.4.3.11 OPC UA Certificate ... 75
8.4.3.12 Step-by-step instructions (CN 4100) ... 77
8.4.3.13 Step-by-step instructions (System & Diagnostic) ... 94
8.4.3.14 Step-by-step instructions (OPC UA certificate) ... 124
8.4.4 CFC ("Chart") ... 134
8.4.4.1 CFC ("Diagram") ... 134
8.4.4.2 Step-by-step instructions ... 134
8.4.5 Folder ... 148
8.4.5.1 Folder ... 148
8.4.5.2 Step-by-step instructions ... 149

8.5 Library area ... 154
8.5.1 Library area ... 154
8.5.2 Step-by-step instructions ... 155
8.5.2.1 Using object from the library area ... 155

8.6 Configuration area ... 156
8.6.1 Configuration area ... 156
8.6.2 Operation mode ... 157
8.6.3 Hardware properties ... 157
8.6.3.1 Hardware properties ... 157
8.6.3.2 Basic ... 158
8.6.3.3 Ethernet settings ... 158
8.6.3.4 Advanced network settings ... 160
8.6.3.5 Protocol ... 161
8.6.3.6 First/Second external interface ... 162
8.6.4 Step-by-step instructions ... 163
8.6.4.1 Expanding categories ... 163

8.7 Work area ... 164
8.7.1 Work area ... 164
8.7.2 Editing history ... 166
8.7.3 Step-by-step instructions ... 166
8.7.3.1 Enlarging / reducing a view ... 166
8.7.3.2 Zooming in and zoom out of an object ... 167
8.7.3.3 Removing an object ... 168
8.7.3.4 Copying a selection ... 169
8.7.3.5 Cutting a selection ... 170
8.7.3.6 Pasting the copied selection ... 171
8.7.3.7 Selecting all objects ... 172
8.7.3.8 Moving an object ... 173
8.7.3.9 Interconnecting an object ... 174
8.7.3.10 Moving an interconnection ... 175
8.7.3.11 Redrawing connections ... 176
8.7.3.12 Changing the editing history ... 177
8.7.3.13 Limiting the work space according to use ... 178
8.7.4 Program blocks ... 179
8.7.4.1 Program blocks ... 179
8.7.4.2 Parameters ... 180
8.7.4.3 Data types ... 180
8.7.4.4 Step-by-step instructions ... 181
8.7.5 CFC variables ("Input / Output connector") ... 185
8.7.5.1 CFC variables ("Input / Output connector") ... 185
8.7.5.2 Step-by-step instructions ... 185
8.7.6 Minimap ... 189
8.7.6.1 Minimap ... 189
8.7.6.2 Step-by-step instructions ... 189

8.8 Parameter area ... 190
8.8.1 Parameter area ... 190
8.8.2 Block and connection properties ("Properties") ... 191
8.8.2.1 Block and connection properties ("Properties") ... 191
8.8.2.2 Properties (block) ... 192
8.8.2.3 Input Output Port / Parameter (block) ... 193
8.8.2.4 Generic Property (CFC variable) ... 194
8.8.2.5 Source (CFC variable) ... 194
8.8.3 Watch table ("Subscription") ... 195
8.8.3.1 Watch table ("Subscription") ... 195
8.8.3.2 Instance ports ... 196

8.8.3.3 Quality code ... 198
8.8.3.4 Online view ... 199
8.8.3.5 Step-by-step instructions ... 200
8.8.4 Event log ("Log") ... 217
8.8.4.1 Event log ("Log") ... 217
8.8.4.2 Step-by-step instructions ... 218

8.9 Instance area ... 219
8.9.1 Instance area ... 219

8.10 Filter area ... 221
8.10.1 Filter area ... 221
8.10.2 Step-by-step instructions ... 221
8.10.2.1 Changing the list ... 221
8.10.2.2 Filter by names ... 222

A Block libraries ... 223

A.1 Parameter tables ... 223

A.2 General quality statuses ... 223

A.3 Basics ... 224
A.3.1 Analog calculating functions ... 224
A.3.2 Analog functions ... 224
A.3.2.1 Analog time-independent calculating functions ... 224
A.3.2.2 Analog time-dependent calculating functions ... 225
A.3.3 Binary calculating function ... 225
A.3.4 Binary functions ... 225
A.3.4.1 Binary time-independent logic functions ... 225
A.3.4.2 Binary time-dependent logic functions ... 225
A.3.5 Monitoring ... 226
A.3.5.1 Signal monitoring analog ... 226
A.3.5.2 Signal monitoring binary ... 226
A.3.6 System functions ... 226
A.3.7 S7PutGet (Black Box) ... 226
A.3.7.1 S7PutGet (Black Box) ... 226
A.3.7.2 Black box | Network configuration ... 228
A.3.7.3 Black box | Programming ... 228
A.3.8 TCP | Set up the data transfer ... 230
A.3.9 TCP | Parameter assignment ... 232
A.3.9.1 TCP | Configuration of the connection block ... 232
A.3.9.2 TCP | Interconnection principle for communication ... 233
A.3.9.3 TCP | Configuration of the communication paths ... 234
A.3.9.4 TCP | Communication sequence ... 235
A.3.9.5 TCP | Configuration with redundancy ... 236
A.3.10 Simulation basics ... 237
A.3.11 Simulation ... 239
A.3.11.1 Substitute values ... 239
A.3.12 OPC UA ... 239
A.3.12.1 OPC UA client ... 239
A.3.12.2 OPC UA server ... 239
A.3.12.3 OPC UA Client, OPC UA Server | Certificates ... 240
A.3.13 Modbus ... 241
A.3.13.1 Modbus ... 241

A.3.13.2 Modbus | application server and CN 4100 Modbus 242
A.3.13.3 Modbus | Register and BYTE swap 242
A.3.13.4 Modbus | baud rate 243
A.3.13.5 Modbus | data bits 243
A.3.13.6 Modbus | stop bits 243
A.3.13.7 Modbus | parity 244
A.3.13.8 Modbus | handshake protocol 244
A.3.13.9 Modbus | mode 245

A.4 ACalc4cnet | Analog time-independent calculating function library 245
A.4.1 ABS (ACalc4cnet) | Description 245
A.4.2 ADD (ACalc4cnet) | Description 246
A.4.3 ASIG (ACalc4cnet) | Description 248
A.4.4 AVER (ACalc4cnet) | Description 249
A.4.5 AXFR (ACalc4cnet) | Description 251
A.4.6 COMP (ACalc4cnet) | Description 252
A.4.7 CONV (ACalc4cnet) | Description 253
A.4.8 DB (ACalc4cnet) | Description 255
A.4.9 DIV (ACalc4cnet) | Description 257
A.4.10 EXP (ACalc4cnet) | Description 258
A.4.11 LN (ACalc4cnet) | Description 260
A.4.12 MAX (ACalc4cnet) | Description 261
A.4.13 MIN (ACalc4cnet) | Description 262
A.4.14 MINMAX (ACalc4cnet) | Description 263
A.4.15 MONIT (ACalc4cnet) | Description 266
A.4.16 MULTIPLY (ACalc4cnet) | Description 269
A.4.17 NROOT (ACalc4cnet) | Description 270
A.4.18 POLY (ACalc4cnet) | Description 271
A.4.19 POWER (ACalc4cnet) | Description 274
A.4.20 SUB (ACalc4cnet) | Description 275
A.4.21 TRIGON (ACalc4cnet) | Description 277

A.5 ACalcTimeR4cnet | Analog time-dependent calculating function library 280
A.5.1 DERIV (ACalcTimeR4cnet) | Description 280
A.5.2 INT (ACalcTimeR4cnet) | Description 283
A.5.3 LEADLAG (ACalcTimeR4cnet) | Description 286
A.5.4 PT1 (ACalcTimeR4cnet) | Description 288
A.5.5 RAXFR (ACalcTimeR4cnet) | Description 290

A.6 BCalc4cnet | Binary time-independent calculating function library 293
A.6.1 AND (BCalc4cnet) | Description 293
A.6.2 BSEL (BCalc4cnet) | Description 294
A.6.3 BSIG (BCalc4cnet) | Description 296
A.6.4 BXFR (BCalc4cnet) | Description 297
A.6.5 COUNTER (BCalc4cnet) | Description 298
A.6.6 DYN_OR (BCalc4cnet) | Description 300
A.6.7 EXOR (BCalc4cnet) | Description 301
A.6.8 JK_FF (BCalc4cnet) | Description 302
A.6.9 NOT (BCalc4cnet) | Description 304
A.6.10 OR (BCalc4cnet) | Description 305
A.6.11 RS_FF (BCalc4cnet) | Description 306

A.7 BCalcTimeR4cnet | Binary time-dependent calculating function library 308
A.7.1 LPULSE (BCalcTimeR4cnet) | Description 308

A.7.2 MINPULS (BCalcTimeR4cnet) | Description 310
A.7.3 PULSE (BCalcTimeR4cnet) | Description 311
A.7.4 T_LON (BCalcTimeR4cnet) | Description 313
A.7.5 T_OFF (BCalcTimeR4cnet) | Description 315
A.7.6 T_ON (BCalcTimeR4cnet) | Description 316

A.8 ASignMon4cnet | Signal monitoring analog library 318
A.8.1 ANA (ASignMon4cnet) | Description 318
A.8.2 ASEL2 (ASignMon4cnet) | Description 319
A.8.3 ASEL3 (ASignMon4cnet) | Description 325
A.8.4 ASMON (ASignMon4cnet) | Description 333
A.8.5 ASMON_R (ASignMon4cnet) | Description 338

A.9 BSignMon4cnet | Signal monitoring binary library 347
A.9.1 BIN (BSignMon4cnet) | Description 347
A.9.2 BSMON (BSignMon4cnet) | Description 348
A.9.3 BSMON_R (BSignMon4cnet) | Description 351

A.10 SysFcts4cnet | System function library 355
A.10.1 ASQMON (SysFcts4cnet) | Description 355
A.10.2 BSQMON (SysFcts4cnet) | Description 357
A.10.3 OP_ANA (SysFcts4cnet) | Description 359
A.10.4 OP_BIN (SysFcts4cnet) | Description 361

A.11 CONV4cnet | Signal converter library 362
A.11.1 CONV_AT (CONV4cnet) | Description 362
A.11.2 CONV_B2F (CONV4cnet) | Description 363
A.11.3 CONV_B2I (CONV4cnet) | Description 364
A.11.4 CONV_F2B (CONV4cnet) | Description 365
A.11.5 CONV_F2I (CONV4cnet) | Description 366
A.11.6 CONV_I2B (CONV4cnet) | Description 366
A.11.7 CONV_I2F (CONV4cnet) | Description 367

A.12 BB4cnet | Devices.BlackBox-Library 368
A.12.1 CN_BB_AI8 (BB4cnet) | Description 368
A.12.2 CN_BB_AO8 (BB4cnet) | Description 370
A.12.3 CN_BB_COMM (BB4cnet) | Description 372
A.12.4 CN_BB_DI8 (BB4cnet) | Description 375
A.12.5 CN_BB_DO8 (BB4cnet) | Description 377
A.12.6 CN_BB_II8 (BB4cnet) | Description 378
A.12.7 CN_BB_IO8 (BB4cnet) | Description 379
A.12.8 CN_BB_STRI (BB4cnet) | Description 380
A.12.9 CN_BB_STRO (BB4cnet) | Description 382
A.12.10 CN_BB_STRI32x8 (BB4cnet) | Description 383
A.12.11 CN_BB_STRO32x8 (BB4cnet) | Description 384

A.13 TCON4cnet | Devices.TCON-Library 385
A.13.1 Blocks 385
A.13.1.1 TCON_COMM (TCON4cnet) | Description 385
A.13.1.2 TCON_AO32 (TCON4cnet) | Description 387
A.13.1.3 TCON_AI32 (TCON4cnet) | Description 389
A.13.1.4 TCON_IO32 (TCON4cnet) | Description 390
A.13.1.5 TCON_II32 (TCON4cnet) | Description 392
A.13.1.6 TCON_DO128 (TCON4cnet) | Description 394
A.13.1.7 TCON_DI128 (TCON4cnet) | Description 395

A.13.1.8 TCON_SO (TCON4cnet) | Description ... 397
A.13.1.9 TCON_SI (TCON4cnet) | Description ... 399
A.13.2 Simulation ... 400
A.13.2.1 TCON_AI32, TCON_II32, TCON_SI (TCON4cnet) | Configure simulation ... 400
A.13.2.2 TCON_DI128 (TCON4cnet) | Configure simulation ... 401
A.13.3 Parameter assignment ... 402
A.13.3.1 TCON_COMM (TCON4cnet) | Configure connection ... 402
A.13.3.2 TCON_COMM (TCON4cnet) | Establish connection ... 402
A.13.3.3 TCON_COMM (TCON4cnet) | Monitor communication ... 403
A.13.3.4 TCON_AI32, TCON_II32, TCON_SI (TCON4cnet) | Configure mode of operation ... 404
A.13.3.5 TCON_DI128 (TCON4cnet) | Configure mode of operation ... 404
A.13.3.6 TCON_AI32, TCON_II32, TCON_SI (TCON4cnet) | Configure substitute values ... 405
A.13.3.7 TCON_DI128 (TCON4cnet) | Configure substitute values ... 406
A.13.3.8 TCON_AO32, TCON_IO32, TCON_DO128, TCON_SO (TCON4cnet) | Configure mode of operation ... 406
A.13.3.9 TCON_AO32, TCON_IO32, TCON_SO (TCON4cnet) | Configure process value ... 408
A.13.4 Error handling ... 408
A.13.4.1 TCON_AI32, TCON_II32, TCON_DI128, TCON_SI (TCON4cnet) | Configure error handling ... 408
A.13.4.2 TCON_AO32, TCON_IO32, TCON_SO (TCON4cnet) | Configure error handling ... 409
A.13.4.3 TCON_DO128 (TCON4cnet) | Configure error handling ... 410

A.14 OPCUAC4cnet | Protocols.OPC UA.Client-Library ... 410
A.14.1 Blocks ... 410
A.14.1.1 CN_OPCUAC_AI (OPCUAC4cnet) | Description ... 410
A.14.1.2 CN_OPCUAC_AO (OPUAC4cnet) | Description ... 412
A.14.1.3 CN_OPCUAC_DI (OPCUAC4cnet) | Description ... 413
A.14.1.4 CN_OPCUAC_DO (OPCUAC4cnet) | Description ... 414
A.14.1.5 CN_OPCUAC_II (OPCUAC4cnet) | Description ... 415
A.14.1.6 CN_OPCUAC_IO (OPCUAC4cnet) | Description ... 416
A.14.1.7 CN_OPCUAC_SI (OPCUAC4cnet) | Description ... 417
A.14.1.8 CN_OPCUAC_SO (OPCUAC4cnet) | Description ... 418
A.14.1.9 CN_OPCUAC_COMM ... 419
A.14.2 Configuration ... 422
A.14.2.1 CN_OPCUAC_AI, CN_OPCUAC_DI, CN_OPCUAC_SI, CN_OPCUAC_II (OPCUAC4cnet) | Configure mode of operation ... 422
A.14.2.2 CN_OPCUAC_AO, CN_OPCUAC_IO (OPCUAC4cnet) | Configure mode of operation ... 423
A.14.2.3 CN_OPCUAC_DO (OPCUAC4cnet) | Configure mode of operation ... 424
A.14.2.4 CN_OPCUAC_SO (OPCUAC4cnet) | Configure mode of operation ... 425
A.14.2.5 CN_OPCUAC_COMM (OPCUAC4cnet) | Configure connection ... 426
A.14.2.6 CN_OPCUAC_COMM (OPCUAC4cnet) | Establish connection ... 428
A.14.2.7 CN_OPCUAC_COMM (OPCUAC4cnet) | Configure time stamp ... 429
A.14.2.8 CN_OPCUAC_COMM (OPCUAC4cnet) | Configure communication ... 429
A.14.3 Communication ... 430
A.14.3.1 CN_OPCUAC_AO, CN_OPCUAC_DO, CN_OPCUAC_IO, CN_OPCUAC_SO (OPCUAC4cnet) | Monitor communication ... 430
A.14.3.2 CN_OPCUAC_COMM (OPCUAC4cnet) | Monitor communication ... 431

A.15 OPCUAS4cnet | Protocols.OPC UA.Server-Library ... 432
A.15.1 Blocks ... 432
A.15.1.1 CN_OPCUAS_AI (OPCUAS4cnet) | Description ... 432
A.15.1.2 CN_OPCUAS_AO (OPUAS4cnet) | Description ... 432
A.15.1.3 CN_OPCUAS_DI (OPCUAS4cnet) | Description ... 433
A.15.1.4 CN_OPCUAS_DO (OPCUAS4cnet) | Description ... 433

A.15.1.5 CN_OPCUAS_II (OPCUAS4cnet) | Description ... 433
A.15.1.6 CN_OPCUAS_IO (OPCUAS4cnet) | Description ... 434
A.15.1.7 CN_OPCUAS_SI (OPCUAS4cnet) | Description ... 434
A.15.1.8 CN_OPCUAS_SO (OPCUAS4cnet) | Description ... 434
A.15.1.9 CN_OPCUAS_COMM (OPCUAS4cnet) | Description ... 435
A.15.2 Connections ... 435
A.15.2.1 CN_OPCUAS_COMM (OPCUAS4cnet) | Input parameters ... 435
A.15.2.2 CN_OPCUAS_COMM (OPCUAS4cnet) | Output parameters ... 436
A.15.2.3 CN_OPCUAS_AI (OPCUAS4cnet) | Input parameters ... 436
A.15.2.4 CN_OPCUAS_AI (OPCUAS4cnet) | Output parameters ... 437
A.15.2.5 CN_OPCUAS_AO (OPCUAS4cnet) | Input parameters ... 437
A.15.2.6 CN_OPCUAS_DI (OPCUAS4cnet) | Input parameters ... 437
A.15.2.7 CN_OPCUAS_DI (OPCUAS4cnet) | Output parameters ... 437
A.15.2.8 CN_OPCUAS_DO (OPCUAS4cnet) | Input parameters ... 438
A.15.2.9 CN_OPCUAS_II (OPCUAS4cnet) | Input parameters ... 438
A.15.2.10 CN_OPCUAS_II (OPCUAS4cnet) | Output parameters ... 438
A.15.2.11 CN_OPCUAS_IO (OPCUAS4cnet) | Input parameters ... 438
A.15.2.12 CN_OPCUAS_SI (OPCUAS4cnet) | Input parameters ... 439
A.15.2.13 CN_OPCUAS_SI (OPCUAS4cnet) | Output parameters ... 439
A.15.2.14 CN_OPCUAS_SO (OPCUAS4cnet) | Input parameters ... 439
A.15.3 Parameter assignment ... 439
A.15.3.1 CN_OPCUAS_AI, CN_OPCUAS_DI, CN_OPCUAS_SI, CN_OPCUAS_II (OPCUAS4cnet) | Configure mode of operation ... 439
A.15.3.2 CN_OPCUAS_AO, CN_OPCUAS_DO, CN_OPCUAS_IO, CN_OPCUAS_SO (OPCUAS4cnet) | Configure mode of operation ... 440
A.15.3.3 CN_OPCUAS_COMM (OPCUAS4cnet) | Configure connection ... 440
A.15.3.4 CN_OPCUAS_COMM (OPCUAS4cnet) | Configure communication ... 442
A.15.4 Communication ... 443
A.15.4.1 CN_OPCUAS_COMM (OPCUAS4cnet) | Monitor communication ... 443

A.16 MODB4cnet | Protocols.Modbus-Library ... 444
A.16.1 Blocks ... 444
A.16.1.1 MODBUS_AI01 (MODB4cnet) | Description ... 444
A.16.1.2 MODBUS_AI08 (MODB4cnet) | Description ... 446
A.16.1.3 MODBUS_AI16 (MODB4cnet) | Description ... 448
A.16.1.4 MODBUS_AO8 (MODB4cnet) | Description ... 450
A.16.1.5 MODBUS_COMM (MODB4cnet) | Description ... 452
A.16.1.6 MODBUS_DI01 (MODB4cnet) | Description ... 454
A.16.1.7 MODBUS_DI08 (MODB4cnet) | Description ... 455
A.16.1.8 MODBUS_DI16 (MODB4cnet) | Description ... 457
A.16.1.9 MODBUS_DO8 (MODB4cnet) | Description ... 458
A.16.2 Parameter assignment ... 460
A.16.2.1 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Configure connection ... 460
A.16.2.2 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Configure communication ... 460
A.16.2.3 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Configure time stamp ... 463
A.16.2.4 MODBUS_AO8 (MODB4cnet) | Parameterize a connection ... 464
A.16.2.5 MODBUS_AO8 (MODB4cnet) | Parameterize communication ... 464
A.16.2.6 MODBUS_COMM (MODB4cnet) | Parameterize a connection ... 468
A.16.2.7 MODBUS_COMM (MODB4cnet) | Parameterize communication ... 471
A.16.2.8 MODBUS_DI01, MODBUS_DI08, MODBUS_DI16 (MODB4cnet) | Parameterize a connection ... 473

A.16.2.9 MODBUS_DI01, MODBUS_DI08, MODBUS_DI16 (MODB4cnet) | Parameterize communication 473
A.16.2.10 MODBUS_DO8 (MODB4cnet) | Parameterize a connection 474
A.16.2.11 MODBUS_DO8 (MODB4cnet) | Parameterize communication 475
A.16.3 Communication 476
A.16.3.1 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Monitor communication 476
A.16.3.2 MODBUS_AO8, MODBUS_DO8 (MODB4cnet) | Monitor communication 477
A.16.3.3 MODBUS_COMM (MODB4cnet) | Monitor communication 478
A.16.3.4 MODBUS_DI01, MODBUS_DI08, MODBUS_DI16 (MODB4cnet) | Monitor communication 479
A.16.4 Alarming, error handling 480
A.16.4.1 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Parameterize troubleshooting 480
A.16.4.2 MODBUS_COMM (MODB4cnet) | Parameterize troubleshooting 481
A.16.4.3 MODBUS_DI01, MODBUS_DI08, MODBUS_DI16 (MODB4cnet) | Parameterize troubleshooting 481

A.17 SYS4cnet | System-Library 482
A.17.1 Blocks 482
A.17.1.1 BW_RTC_DIAG (SYS4cnet) | Description 482
A.17.2 Configuration 485
A.17.2.1 BW_RTC_DIAG (SYS4cnet) | Configure mode of operation 485

Cybersecurity information

1

Siemens provides products and solutions with industrial cybersecurity functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial cybersecurity concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial cybersecurity measures that may be implemented, please visit
https://www.siemens.com/cybersecurity-industry (https://www.siemens.com/cybersecurity-industry).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Cybersecurity RSS Feed under
https://new.siemens.com/cert (https://www.siemens.com/cert).

Basic information on the CNET tool

2

The CNET tool is a software that provides all the functions required for using the CN 4100 communication system.

CNET download (https://support.industry.siemens.com/cs/ww/en/view/109814147)

CN 4100 Product Information (https://support.industry.siemens.com/cs/ww/en/view/109814144)

Functions

The CNET tool is installed as a software program on a PC/IPC and functions autonomously. All the required services and components for the configuration, up to the operation of CN 4100, are contained in the CNET tool.

The CNET tool offers the following functions:

- Configuration of CN 4100
- Engineering for CN 4100
- Communication with CN 4100
- Diagnostics of CN 4100
- Import or export of engineering data

The representation of the blocks in the CNET tool is similar to that of the LAD programming language.

Setup

A further hardware tool is required for setting up CN 4100.

Useful information about setup (Page 21)

Project upgrade

This project can be upgraded for further use of a project from CNET in a new main version of CNET (e.g. from V1.0 to V2.0).

Useful information on engineering (Page 32)

Program blocks

The CNET tool uses an own set of blocks which are already implemented as a block library in CNET.

Block libraries (Page 223)

Program examples

The following program examples are available as FAQs:

- For CN 4100 generally:
  - Application example with different software programs (https://support.industry.siemens.com/cs/ww/en/view/109801222)

Useful information 3

3.1 Overview of the documentation

Purpose of the documentation

This documentation supports you in using the CNET tool. The documentation encompasses the following topics:

- Commissioning of the hardware
- Configuration of the hardware
- Configuration und Interconnection of the blocks
- Operator control and monitoring
- System diagnostics
- Import or export of engineering data
- Upgrading a project backup or project

Required knowledge

General knowledge in the field of automation technology is required to understand the documentation.

The documentation divides the descriptions on a topic from the step-by-step instructions (procedure) on a topic. For this reason, always first read the description of a chapter (see "Useful information...") before continuing with the step-by-step instructions of the chapter.

Requirements for the use of CNET

The CNET tool is designed for usage in combination with CN 4100. In addition, the following requirements apply for using the software and hardware.

Requirements (Page 47)

Related documentation

More information on using CN 4100 is available in the following documents:

- "CN 4100 Communication System" Equipment Manual (https://support.industry.siemens.com/cs/ww/en/view/109954697)

Conventions

The following terms are used synonymously in this documentation:

Term	Synonyms	Notes
Engineering system	• ES	In as far as not specified otherwise, the term applies for a PC/IPC on which the CNET tool is installed and a project is created.
CN 4100	• SIMATIC CN 4100 • System • Hardware • CPU	In as far as not specified otherwise, the term applies for the hardware used in the CNET tool.
initialize		In as far as not specified otherwise, the term applies for the compiling and loading of the configuration data.
CPU module	• CPU • Control system • PLC	In as far as not specified otherwise, the term applies for a programmable logic controller.
Communications module	• Expansion module • Module • Electronic module • CM • Communications processor • CP	In as far as not specified otherwise, the term applies for an object for communication via a serial or other interface.

Also observe the notes identified as follows:

Note

A note contains important information on the product described in the documentation, on the handling of the product or on that part of the documentation to which special attention is to be drawn.

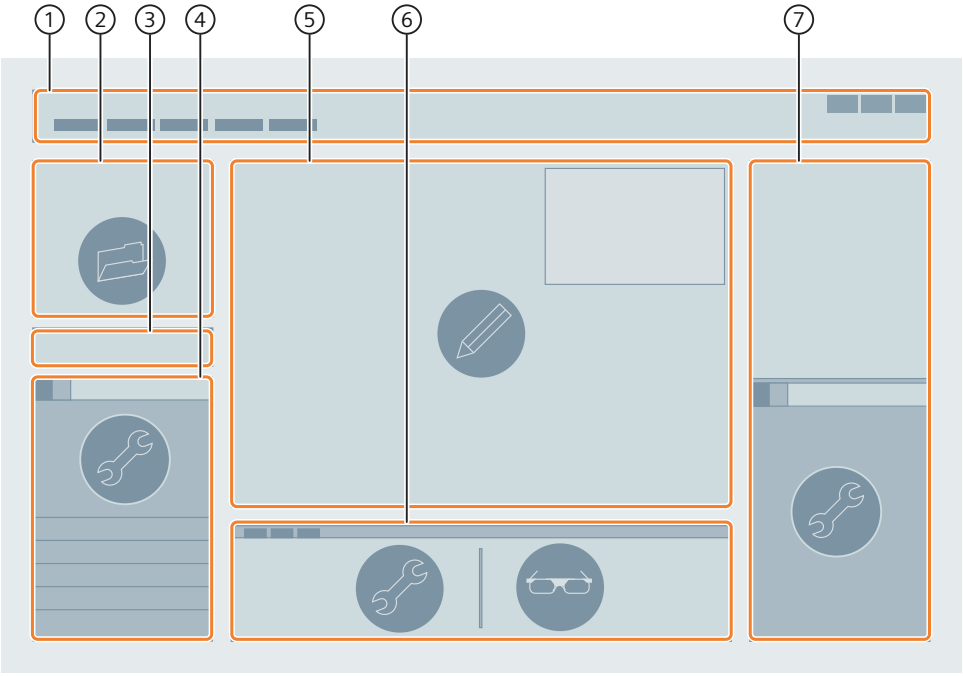
3.2 Operation of the user interface of the CNET tool

In the following reference is made to:

- Structure of the user interface
- Window

Structure of the user interface

The user interface contains the following elements:



①	Menu	The menu provides access to central functions of CNET. Menu (Page 51)
②	Navigation area	The navigation area displays the project navigation. Navigation area (Page 64)
③	Library area	The library area displays the block libraries. Library area (Page 154)
④	Configuration area	The configuration area shows the properties of the selected hardware. Configuration area (Page 156)
⑤	Work area	The work area displays an opened CFC. Work area (Page 164)
⑥	Parameter area	This area displays either the block and connection properties or the watch table. Parameter area (Page 190)
⑦	Instance area	The instance area shows a listing of all created program blocks of a CFC. In addition, the properties of the currently opened CFC are displayed in the lower area. Instance area (Page 219)

Windows/areas

Note

Hidden areas

All areas are shown by default but can be hidden. The watch table can only be seen after switching over to it.

You adapt the windows of the user interface as follows:

Showing / hiding a window (Page 50)

Customizing the window size (Page 49)

Getting started

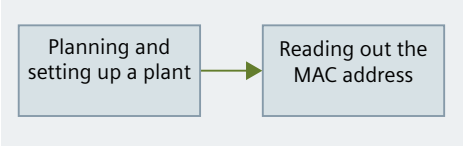
4

4.1 Getting started in CN 4100

After you have set up your engineering system successfully, you can start setting up CN 4100.

You need the following information to set up CN 4100:

Useful information about setup (Page 21)



Planning and setting up a plant

Plan your plant by using a different software (PCS 7) to CNET and include the CN 4100 system in your planning.

You can find more information in the document Plant and User Documenation ("PUD") of your software.

PUD in SIOS (https://support.industry.siemens.com/cs/de/en/view/109748882)

Reading out the MAC address / assigning an IP address

You need the MAC address of the CN 4100 for communication with CNET.

Reading out the MAC address / assigning an IP address (Page 22)

4.2 Useful information about setup

Plant topology

Planning the topology of a plant is an important step in obtaining an overview of the components to be used and in structuring the required interconnections.

Note

The CNET tool does not provide any possibility to represent the topology of a plant. You can only create the CN 4100 system in the CNET tool.

Take the function and possibilities of CN 4100 into consideration at the topology of your plant.

You can find more information on the hardware in the document:

- System Manual "CN 4100 communication system"

MAC address of the CN 4100

Note

Switching on the CN 4100 for the first time

The entire boot process when switching on the CN 4100 for the first time takes several minutes.

In the factory setting, the configuration data of the CN 4100 only contains the MAC address for communication. Use a hardware tool that allows scanning of CN 4100 via the DCP procedure.

Recommended tools:

- PRONETA tool (https://support.industry.siemens.com/cs/de/en/view/109807501)
- SIMATIC Manager (e.g. including PCS 7) (https://support.industry.siemens.com/cs/de/en/view/109805072)
- SINEC PNI (https://support.industry.siemens.com/cs/de/en/view/109804190)

After an IP address has been assigned, the CNET tool can communicate with CN 4100.

Note

IP address cannot be changed after assignment

Once an IP address has been assigned in the CN 4100 during the initial setup, it cannot be changed afterwards. To be able to change the IP address, the CN 4100 CPU must first be reset to factory settings.

4.3 Reading out the MAC address / assigning an IP address

Requirement

- The Engineering System is connected to the CN 4100.
- A DCP-compatible tool is installed on the Engineering System and opened.
- The CN 4100 is switched on.

Procedure

1. Navigate in the DCP-compatible tool to the network properties / MAC address of the connected hardware.
2. Read the MAC address of the CN 4100.

3. Note down or copy the MAC address (for example by using Copy&Paste).
4. Assign an IP address in the DCP-compatible tool of the CN 4100 and note it down.

Result

The MAC address is known to the user. The CN 4100 can be configured via CNET.

Configuration

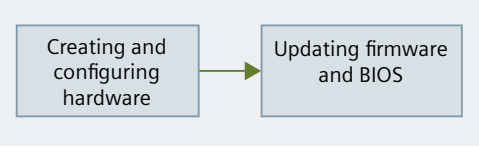
5

5.1 Using CN 4100

After you have successfully installed the CNET tool and read out the MAC address of CN 4100, the configuration of CN 4100 can be started.

You need the following information to configure the hardware in CNET:

Useful information about the configuration (Page 25)



Creating and configuring hardware

You first have to configure the hardware in the CNET tool in order to interact with CN 4100.

Creating and configuring hardware (Page 28)

Updating firmware and BIOS

After you have configured your CN 4100 system, you should check the firmware version of the CN 4100 CPU. On the basis of the version of the CNET tool, a firmware update of the hardware can be advisable.

Update the firmware so that the most up-to-date functions can be used.

Update firmware and BIOS (of non-redundant CPUs) (Page 29)

Updating firmware and BIOS (of redundant CPUs) (Page 29)

5.2 Useful information on configuration

Configuration in CNET

The CNET tool can only communicate with the hardware (CN 4100) if it has been created and configured. Creation of the hardware takes place via the navigation area in CNET.

Navigation area (Page 64)

Consider the entire structure of the respective system when creating the hardware.

CN 4100 (Page 66)

Hardware properties (Page 157)

When configuring CN 4100, the Ethernet settings and possibly protocols to be used for CN 4100 also have to be considered.

Ethernet settings (Page 158)

Protocol (Page 161)

The configuration data for the hardware can be viewed after creation via the configuration area.

Configuration area (Page 156)

Note

Changing the device password

Every newly created CN 4100 already uses a device password. You should change this password immediately in the project after creation.

User and password (Page 48)

Changing the device password (Page 94)

Operation mode

As soon as the configuration data has been loaded into the hardware, it is in the operation mode.

Operation mode (Page 157)

The CN 4100 CPU can be monitored in operation mode and can be filled with project data.

CFC (Page 134)

Note

Accessibility of the hardware can be displayed

The display of accessibility can be activated for the individual hardware.

Automatic Status Update (Page 74)

Firmware updates

Firmware updates provide both bug fixes as well as new functions for the hardware (CN 4100). Accordingly it is recommended to keep the hardware up-to-date.

Firmware updates can also contain BIOS updates.

Note

Firmware update for CNET

The latest, tested firmware update matching the CNET version is supplied with the CNET installation pack. After installation, this is located in the system folder of CNET.

You can find information on compatibility between the CN 4100 firmware versions and the CNET version in the compatibility list.

Compatibility list (Page 48)

Note

Note signing

The firmware updates are provided as signed packages. Make sure a package is signed(* **) before you install it.

*The signing of the firmware updates is automatically checked by CNET when a package is opened. Unsigned packages are rejected.

**You can check the integrity of a CNET installer using the checksum.

You can obtain information about the firmware of a CPU in the "Firmware update" section. You may also update the firmware of the CPU in this section.

Firmware update (Page 75)

Note

Power failures must be strictly avoided during the firmware update

A power failure during the firmware update leads to hardware errors:

- The hardware cannot be reset to a previous state.
- The hardware would need to be sent in for repair.

Ensure that the hardware remains continuously powered during a firmware update.

Note

Firmware update of non-redundant CPUs only offline

On firmware update of a non-redundant CPU, "online" update is not possible. The hardware is not available during the firmware updates.

5.3 Creating and configuring hardware

Requirement

- The Engineering System is connected to the CN 4100.
- The CN 4100 is switched on.
  Info page of the CN 4100 communication system (https://support.industry.siemens.com/cs/ww/en/view/109954697)
- Navigation area and configuration area are visible.
  Showing / hiding a window (Page 50)
- The settings for the CN 4100 are known.
  Hardware properties (Page 157)

Procedure

1. Create the hardware.
   Creating the CN 4100 (Page 77)
2. Change the device password.
   Changing the device password (Page 94)
3. If you want to change the configuration data after they have been created: Carry out the changes for the respective module.
   Processing the CN 4100 (Page 80)
4. Initialize the hardware.
   Initializing the configuration data completely (Page 84)

   Note

   If the hardware has already been initialized and you only want to carry out changes to the hardware: Load the changes into the hardware.

   Initializing the configuration data partially (Page 86)

5. Optional: Use automatic update to display the accessibility of the hardware.
   - Activating automatic update of the device status (Page 116)
   - Deactivating automatic update of the device status (Page 117)

Result

The hardware has been created, configured and initialized.

5.4 Update firmware and BIOS (of non-redundant CPUs)

Requirement

- Navigation area and work area are visible.
- An IP address has been assigned to the hardware.
- The Engineering System is connected to the hardware.
- The hardware is continuously powered and turned on.
- The project is in offline mode.
  Offline mode (Page 67)
- The compatibility list is known and will be taken into account when updating.
  Compatibility list (Page 48)

Procedure

1. Check whether the firmware is up to date.
   Checking the firmware (Page 97)
2. Use the supplied firmware update from the system folder of CNET.
   – Example path: `C:\Program Files\CNet\app\data\firmware`
3. Load the firmware update on the hardware.
   Loading a firmware update (Page 99)

Result

New firmware is installed on the hardware.

5.5 Updating firmware and BIOS (of redundant CPUs)

Requirement

- Navigation area and work area are visible.
- An IP address has been assigned to the hardware.
- The Engineering System is connected to the hardware.
- The hardware is continuously powered and turned on.
- The compatibility list is known and will be taken into account when updating.
  Compatibility list (Page 48)

Procedure

1. Check whether the firmware is up to date.
   Checking the firmware (Page 97)
2. Use the supplied firmware update from the system folder of CNET.
   – Example path: `C:\Program Files\CNet\app\data\firmware`
3. Load the firmware update to "CPU 1".
   Loading a firmware update (Page 99)
4. Let CPU 1 start up until redundant mode is achieved.
5. Proceed for CPU 2 as described in step 3 and 4 for CPU 1.

Result

New firmware is installed on the hardware.

Engineering

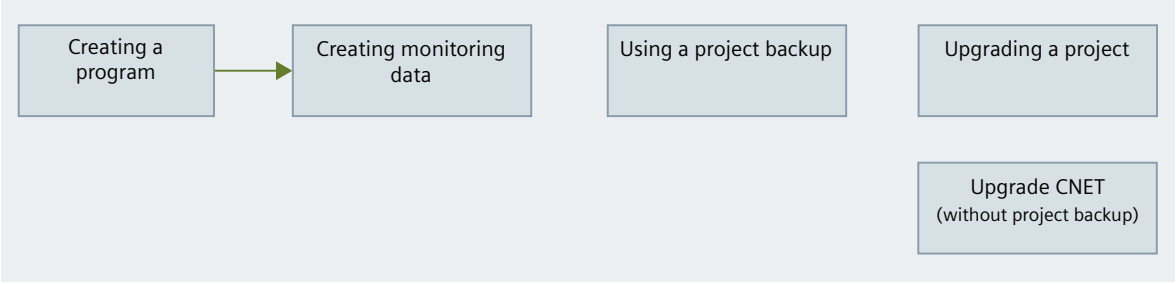
6

6.1 Working with CFC or backups

After you have successfully configured and initialized the hardware, you can start creating programs.

You need the following information for engineering in the CNET tool.

Useful information on engineering (Page 32)



Creating a program

Create one or more programs for execution in CNET.

Creating a program (Page 33)

Creating monitoring data

Create a watch table and specify the parameters for monitoring.

Creating monitoring data (Page 35)

Using a project backup

Use the backup functions of the CNET tool to back up or restore a project.

Using a project backup (Page 36)

Upgrading a project

Use the project upgrade option to continue using an old project in a newer version of CNET.

Upgrading a project (Page 36)

Upgrade CNET (without project backup)

Delete old project data if you don't wish to use them in a later version of CNET.

Upgrade CNET (without project backup) (Page 38)

6.2 Useful information on engineering

Programs in CNET

In the CNET tool programs can be created across one or more CFCs.

CFC ("Diagram") (Page 134)

It is also possible to import/export CFCs.

Note

Scope of the export file of a CFC

The export file of a CFC can contain the CFC and its content and, in addition, the higher-level folder. If the CFC is located inside a folder, the folder is always included in the export/import.

When exporting/importing, the metadata of the objects is also retained.

Note

Export of CFCs from older CNET versions is partially incompatible

Contents of an export file created with an older CNET version may be incompatible with a newer version.

As of CNET V3.0:

- The MODBUS_COMM block was modified.

Procedure:

1. Before importing the contents into a newer CNET version, note the version (name and date) of an export file.
2. Edit the export file using an editor if necessary. Adjust the parameter names and values of affected modules.

CFC variables are available to transfer values between different CFCs.

CFC variables ("Input / Output connector") (Page 185)

The program blocks of the CN 4100 have varying complexities and were categorized according to topics. Each block library corresponds to a different topic.

Program blocks (Page 179)

Block libraries (Page 223)

To specify values, these have to be stored at the input parameters of the program blocks. The parameter values can also be exported/imported.

Parameters (Page 180)

The CN 4100 CPU does not know global data blocks or PLC variables.

The data types of CN 4100 / CNET differ from those of STEP 7.

Data types (Page 180)

The editing history of the work area can be changed.

Editing history (Page 166)

Monitoring values

Watch tables have to be used to monitor project data (engineering) in the CNET tool.

Watch table ("Subscription") (Page 195)

The CNET tool allows several watch tables to be created. However, only one watch table can be used at any one time. Watch tables are hidden by default when the CNET tool is opened.

Upgrading a project backup or project

Only one project can be created at any one time in the CNET tool. The project status – including the configuration data – is saved automatically in the process. To use a project version on a different PC or restore a previous version of a project, CNET offers the project backup option.

When upgrading to a newer major version of CNET (e.g. V1.0 to V2.0), you cannot simply continue to use the current project data in the CNET storage path. To transfer a project to a newer CNET version, you need to use the project backup functions. The procedure differs from the simple project backup.

Note

"IP Gateway" changed

As of CNET V2.0, "IP Gateway" must be taken into consideration on both sides for redundant CPUs.

Ethernet settings (Page 158)

Note

Protocol blocks changed

For the version of CNET relevant to the protocols, the communication data of previously created blocks is reset in some cases.

- As of CNET V2.5:
  OPC UA Client: The password and user must be re-entered on the CN_OPCUAC_COMM block.
- As of CNET V3.0:
  Modbus:
  - The display for the value of the station type has been changed. However, the old values are displayed after the 'project upgrade' (or also when exporting/importing) from a CNET version <v3.0.
  - The station type must be selected again at the MODBUS_COMM block.

6.3 Creating a program

Requirement

- The hardware is configured.
- Navigation area and work area are visible.

Procedure

1. Create a CFC.
   Creating a CFC (Page 134)
   - Alternatively:
     Importing a CFC (Page 142)
     Exporting a CFC (Page 141)
     Renaming/editing a CFC (Page 137)
     Removing a CFC (Page 139)
2. Open the CFC.
   Opening a CFC (Page 136)
3. Create several blocks.
   Creating blocks (Page 181)
   - Alternatively:
     Using object from the library area (Page 155)
     Copying a selection (Page 169)
     Cutting a selection (Page 170)
     Pasting the copied selection (Page 171)
     Selecting all objects (Page 172)
     Changing the editing history (Page 177)
4. Optional: Display the minimap ("Diagram map") and modify the partial image.
   - Showing / hiding a window (Page 50)
   - Moving a partial image (Page 189)
   - Limiting the work space according to use (Page 178)
   - Showing defaults (Page 61)
   - Detaching a section (Page 62)
5. Optional: Modify the view.
   - You can modify the view in the work area to obtain a better overview.
     Enlarging / reducing a view (Page 166)
   - You can modify the window size.
     Customizing the window size (Page 49)
6. If required, change the values at the block.
   Configuring a block (Page 183)
   - Alternatively:
     Importing parameters (Page 146)
     Exporting parameters (Page 144)
7. Interconnect the blocks.
   Interconnecting an object (Page 174)

8. Optional: Define the position or order of the objects.
   - Moving an object (Page 173)
   - Removing an object (Page 168)
   - Moving an interconnection (Page 175)
   - Redrawing connections (Page 176)
9. Optional: Create CFC variables between two CFCs.

   Note

   Two CFC variables are always required for a connection: From the output interface to input interface.

   - Creating and interconnecting an output interface (Page 185)
   - Creating an input interface (Page 186)
   - Displaying properties of a CFC variable (Page 188)

Result

The program has been created.

6.4 Creating monitoring data

Requirement

- The hardware is configured.
- Navigation area and work area are visible.
- At least one program has been created.

Procedure

1. Create a watch table.
   Creating a watch table (Page 200)
2. Alternatively: Open the watch table created beforehand.
   Opening a watch table (Page 201)
3. Create new entries in the watch table on the basis of block parameters.
   Switching to watch table (Page 205)
   Creating entries for monitoring (Page 205)

   Note

   Alternatively: You can remove unwanted entries from the watch table.

   Removing entries from monitoring (Page 208)

4. Optional: Navigate to the instance in the work area via the watch table entry.
   Navigating from the watch table to the instance (Page 212)
5. Optional: Remove an unwanted watch table.
   Switching to watch table (Page 205)
   Removing a watch table (Page 204)

Result

A watch table with monitoring data has been created.

6.5 Using a project backup

Requirement

- A project with content has been created that is to be reused.
- Navigation area and work area are visible.
- The version of CNET is the same when creating and installing the backup.
  - Alternative: Upgrading a project (Page 36)

Procedure

1. Create a backup of the project.
   Creating a backup (Page 52)
2. Optional: Restore an old project version.
   Restoring a backup (Page 54)

Result

You have used the project backup of CNET.

6.6 Upgrading a project

Requirement

- A project with content has been created that is to be reused.
- Navigation area and work area are visible.
- The version of CNET is different when creating and uploading the backup to be used.

- Before upgrading to CNET version V2.0:
  - The CN 4100 CPUs have firmware version V1.12.0.
- Before upgrading to CNET version V3.0:
  - CNET version is V2.5.
  - For the CN 4100 CPUs, an update from firmware version V2.9.2 or V2.20.3 to V3.00.00 was performed.
  - The CNET project or the CPUs remain in offline mode.

Procedure

1. Optional: Create a backup of the project of the version of CNET you are currently using.
   Creating a backup (Page 52)
2. Option a or b: You want to (a) use the current backup with a newer version of CNET. Or (b), you want to continue using a backup of an older CNET version in the current CNET.
   - a) Close and uninstall CNET.
   - b) Close CNET.
3. Delete the following files in the CNET storage location:
   - Delete the contents of "data base" under: `${user.home}/AppData/cnet/data/db/`
   - Delete the contents of "diagram" under: `${user.home}/AppData/cnet/data/diagram`

   Note

   Possibly different paths

   Depending on the configuration of CNET in the operating system, the storage paths may differ from those mentioned above.
   1. Navigate to the system folder of CNET and open "app" > file "`application.properties`".
   2. Check the storage paths ("`spring.datasource.url`" and "`cm.configurator.diag.path`").
4. Option a or b: You want to (a) use a new CNET version or you want to (b) continue using the current CNET version.
   - a) Install a newer CNET version and open it.
   - b) Open CNET.
5. Restore an old project version.
   Restoring a backup (Page 54)
6. Adjust the names of the CPUs if necessary. (The name of "CPU 1" must end with "-A", "CPU 2" with "-B").
   Processing the CN 4100 (Page 80)

7. Activate the selected protocols again in the settings of the respective hardware:
   - Select protocols other than the previously selected protocols. Confirm the settings and close the dialog.
   - Then open the dialog again and select the desired protocols.

   Processing the CN 4100 (Page 80)
8. If there is a redundant CPU in the project:
   - Check the Ethernet settings of the "CPU 1" for the use of the entry "IP Gateway". Expanding categories (Page 163)
   - If the "IP Gateway" entry of "CPU 1" is filled:
     Enter the gateway at the "CPU 2" as well.
     Processing the CN 4100 (Page 80)
9. If Modbus or OPC UA blocks are to be used in the project:
   - After upgrading from CNET V1.0 to a higher version:
     Exchange the OPC UA blocks.
     Exporting a CFC (Page 141)
     Importing a CFC (Page 142)
   - After upgrading from CNET V2.0 to a higher version:
     Enter the password and the user of the OPC UA connection again on the CN_OPCUAC_COMM block.
   - After upgrading from CNET V2.5 to a higher version:
     Re-select the station type to be used at the MODBUS_COMM block.
10. Check the firmware of the CPUs.

    Note

    CNET version must match the firmware of the CPU.

    If required, update the firmware of the CPUs. You can find more information under: Using CN 4100 (Page 25)
11. If upgraded to CNET V3.0 or higher:
    Initialize the hardware.
    Initializing configuration data completely (Page 84)

Result

You have upgraded an older project to a newer version of CNET.

6.7 Upgrade CNET (without project backup)

Requirement

- An older CNET version is installed on the PC.
- CNET is closed.

- Project data of the older CNET version is still in the storage location of CNET.
- A newer CNET version is to be installed, but the old project data not backed up.

Procedure

1. Uninstall CNET via the Control Panel of the operating system.
2. Delete the following files in the CNET storage location:
   – Delete the contents of "data base" under: `${user.home}/AppData/cnet/data/db/`
   – Delete the contents of "diagram" under: `${user.home}/AppData/cnet/data/diagram`

   Note

   Possibly different paths

   Depending on the configuration of CNET in the operating system, the storage paths may differ from those mentioned above.

   1. Navigate to the system folder of CNET and open "app" > file "`application.properties`".
   2. Check the storage paths ("`spring.datasource.url`" and "`cm.configurator.diag.path`").
3. Install the newer CNET version.
4. Open CNET and check the firmware of the CPUs.

   Note

   CNET version must match the firmware of the CPU.

   If required, update the firmware of the CPUs. You can find more information under: Using CN 4100 (Page 25)

Result

You have installed a newer version of CNET on the PC.

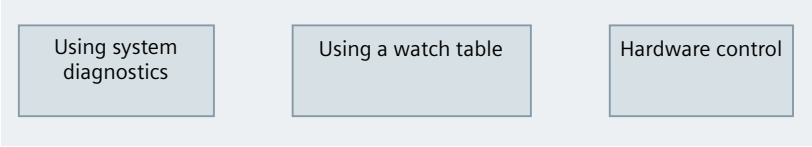
Operation 7

7.1 Working with the CPU online

After you have completed the steps for the configuration and the engineering of your project, you can operate the hardware via CNET.

You need the following information to configure the hardware via CNET:

Useful information about the operation (Page 41)



Using system diagnostics

Use the diagnostics function of CNET to have the status of a selected hardware displayed.

Using system diagnostics (Page 43)

Using a watch table

Observe selected parameters of a created program.

Using a watch table (Page 44)

Hardware control

Use the function for rebooting or resetting the hardware.

Hardware control (Page 45)

7.2 Useful information on operation

System diagnostics

As soon as the hardware has been initialized, it can be diagnosed via the Device Cockpit. The CNET project must therefore not specifically go into Online mode.

Device Cockpit (Page 67)

If the redundant hardware is to be created redundantly, the diagnostics data is subdivided by CPU and also informs about the MASTER/STANDBY status.

Note

Data of selected hardware

If you select another hardware in the navigation area while a Device Cockpit is open, the data in the Device Cockpit is not updated.

Note that the data in the Device Cockpit always corresponds to the previously selected hardware.

In addition to the diagnostics data, CN 4100 also outputs information on the general device status.

Update device state (Page 70)

Events related to CNET can be read out via the event log.

Event log ("Log") (Page 217)

CNET also offers error logging of the Ethernet interfaces.

Ethernet Trace (Page 73)

Watch tables

As soon as you have created a watch table with entries you can use them in the Online mode.

Online mode (Page 67)

Watch table (Page 195)

Online view (Page 199)

Note

Considered project data

Note that only the previously loaded project data can be considered for the monitoring. You create monitoring in offline mode.

Offline mode (Page 67)

Note

Non-transferred data is displayed

Data that has not yet been transferred to the hardware is indicated by a blue accent in the navigation area and in the work area.

Note

Program data changes can be transferred

For transfer of the program data to the hardware, there is an option to load the changes ("Delta activation"). This enables only the changes (to parameters) to be considered for the transfer without having to stop the program. If parameters relevant for communication have been changed, the connection of the relevant communication blocks must be reinitialized.

Reboot or reset

From the navigation area, you can reboot or reset the hardware involved.

During a reboot, the CN 4100 is shut down and restarted.

Note

Duration of the reboot

The reboot of CN 4100 can take some minutes.

During a reset, the configuration data in the CN 4100 is deleted and the configuration in CNET is reset to the factory settings.

Hardware properties (Page 157)

7.3 Using system diagnostics

Requirement

- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.

Procedure

1. Check the device status.
   Display device status (Page 111)
2. If necessary, update the device status.
   Updating device status (Page 113)
3. Display the diagnostics data.
   Displaying diagnostics data (Page 106)
4. If required, update the diagnostics data of the hardware.
   Updating diagnostics data (Page 107)
5. Optional: Export a log file of the hardware.
   Exporting a log file (Page 108)
6. Optional: View the CNET event log.
   Showing the event log (Page 218)
7. Optional: Use logging of the Ethernet connections.
   - Ethernet | Show IP (Page 118)
   - Ethernet | Start logging (Page 120)
   - Ethernet | Stop logging (Page 122)

Result

The current diagnostics data of the selected hardware are displayed.

7.4 Using a watch table

Requirement

- The hardware is configured.
- Navigation area and work area are visible.
- At least one program has been created.
- A watch table with entries is created.
- The watch table is displayed.

Procedure

1. Determine the status of the monitoring data.
   - a) Enter the values for monitoring freely.
     Enabling entries for monitoring (Page 210)
   - b) Cancel the monitoring of the values.
     Canceling monitoring (Page 211)
2. Optional: For a watch table entry, find the instance in the work area.
   Navigating from the watch table to the instance (Page 212)
3. Transfer the project data to the hardware.
   Downloading the project and placing it online (Page 89)
4. Optional: If you want to modify the monitoring:
   - Terminate the online mode.
     Setting a project offline (Page 92)
   - Carry out steps 1 to 3 again.
5. Optional: If you want to force the value at a parameter, you can use the force function.
   - a) Activate forcing for one parameter.
     Activating forcing of the watch table (Page 213)
   - b) Deactivate forcing for the parameters of the selected hardware.
     Deactivating forcing of the watch table (Page 214)
     Deactivating forcing for all parameters (Page 83)
6. Use the online view of the watch table.
   - a) Display current status values.
     Activating online view (Page 215)
   - b) Cancel the updated display of the status values.
     Deactivating online view (Page 216)

Result

- a) Monitoring is activated. The current values are read out.
- b) Monitoring is deactivated. Old values are displayed.

7.5 Controlling hardware

Requirement

- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.

Procedure

- a) Restart the hardware.
  Restarting hardware (Page 101)
- b) Resets the configuration data of the hardware to factory settings.
  Resetting hardware (Page 104)

Result

You have changed the operating state of the hardware.

Functions

8

8.1 Requirements

8.1.1 Requirements

The following requirements have to be fulfilled in order to use the CNET tool on a PC/IPC.

Requirements for PC hardware

- 4-core CPU with >=2.0 GHz, >=2 GB RAM, onboard RJ45, >=10 GB of free hard disk space (SSD recommended)

Note

The hardware requirements for an Engineering System of PCS 7 V9.1. apply.

Requirements for PC software

- Operating system (64-bit):
  - Windows 10
  - Windows Server 2016
  - Windows Server 2019

Note

The CNET tool is written in the JAVA programming language. The JAVA installation is included in the installation of CNET.

Range of validity of the software

The CNET tool is compatible with all components of the CN 4100 communication system.

8.1.2 Compatibility list

To use the CNET tool, compatibility with the firmware version of CN 4100 must be met.

Note

Determining the firmware version for upgrading a CNET project

An upgraded CNET project (of a CNET version) is not compatible with every firmware version of CN 4100.

First, perform a firmware update with the highest firmware version of CN 4100 that is compatible with your current CNET version. After that, you can go through the steps for upgrading your CNET project.

Compatibility with the firmware version

The respective CNET version is compatible with the following firmware version of CN 4100:

CNET version	Firmware version of CN 4100
V1.0	<= V1.12.0
V2.0.5	V2.0.10
V2.5	>=V2.0.10
V3.0	>=V3.00.00

8.1.3 User and password

To protect the hardware and CNET from access, the security-relevant components in CNET are protected by a password prompt.

Device password

- The device password is used for all password prompts.
- The device password can be changed.
- Default device password: "configCN4100"

User

- There is only one user in CNET.
- Default user name: "cnconf"

8.2 Step-by-step instructions

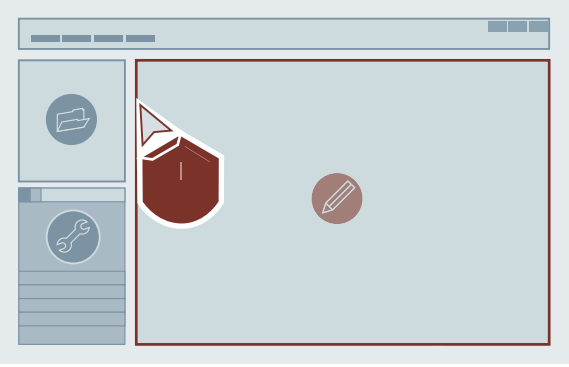
8.2.1 Customizing the window size

Requirement

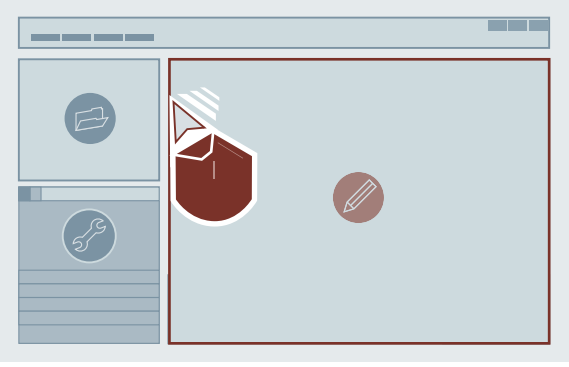
- The window of the respective area is displayed.

Procedure

1. Select the frame of an area.



2. Move the frame to the desired position.



Result

The window size is changed.

8.2.2 Showing / hiding a window

Requirement

For showing:

- You have just opened the CNET tool.
- Or: The respective window for an area to be displayed has been hidden.

For hiding:

- The relevant window is shown.

Note

Function for showing/hiding watch tables

The function for showing/hiding watch tables ("Subscriptions") is only available if a watch table has been created beforehand.

Procedure

Select the desired entry in the "Views" menu.



Result

Depending on the previous display, the window of the selected entry is displayed or hidden.

8.2.3 Navigating in dialogs

Requirement

- The window of the respective area is displayed.

Procedure

Select the respective button.

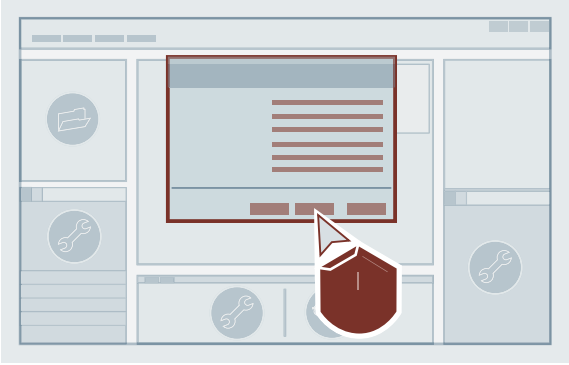
Note

"Previous": Displays the previous page of the dialog.

"Next": Displays the next page of the dialog.

"Cancel": Closes the dialog.

"Finish": Confirms the selection and terminates the dialog.



Result

You have navigated to the desired page of the dialog.

8.3 Menu

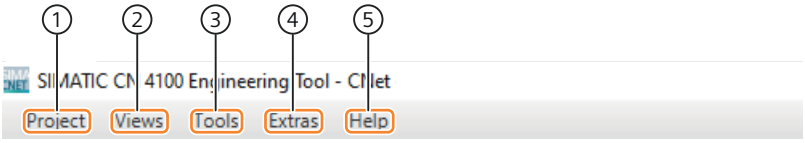
8.3.1 Menu

Definition

The menu is an area of the CNET tool where the user can access project functions and can have components displayed.

Structure

The menu is structured as follows:



①	Project	Opens the selection for the project security or for terminating CNET. Creating a backup (Page 52) Restoring a backup (Page 54) Existing CNET (Page 59) Opens the selection for resetting the project to be able to create a new project. Emptying a project (Page 55)
②	Views	Opens the selection for the following windows of CNET. which can be shown or hidden Minimap (Page 189) ("Diagram map") Watch table (Page 195) ("Subscriptions") Instance area (Page 219) ("Instances") Configuration area (Page 156) ("Hierarchy") Showing or hiding a window (Page 50)
③	Tools	Provides additional functions. You can call a block via its instance. Navigate to instance (Page 60)
④	Extras	Opens additional settings/defaults of CNET. Showing defaults (Page 61) Detaching a section (Page 62)
⑤	Help	Opens the selection for information via the CNET tool. Viewing the CNET and plugin version (Page 57) Having OSS components displayed (Page 59) Displaying documentation (Page 57) Displaying keyboard shortcuts (Page 58)

8.3.2 Step-by-step instructions

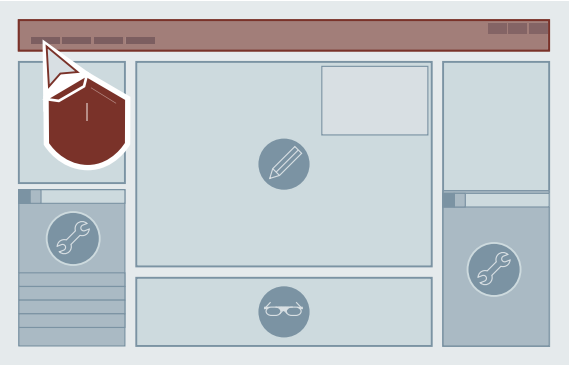
8.3.2.1 Creating a backup

Requirement

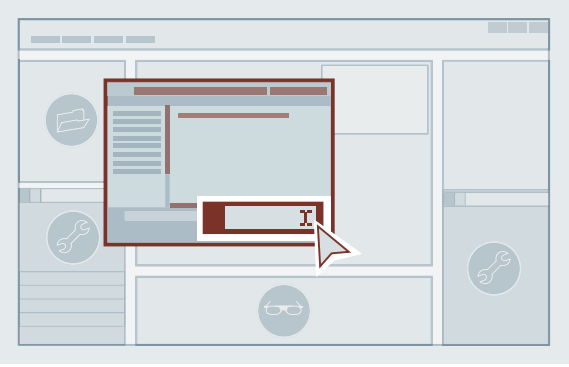
- You have created hardware in the CNET tool and want to back up the project.

Procedure

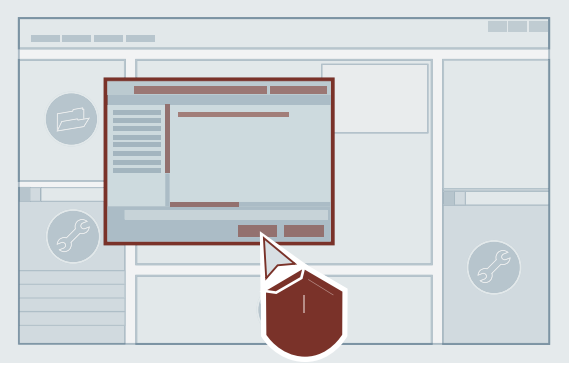
1. Select the "Project" > "Backup..." menu.



2. Select a file name and storage location.



3. Confirm the entry.



Result

A backup of the project is stored at the desired location.

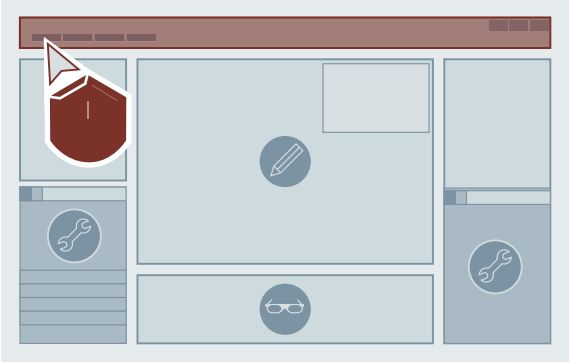
8.3.2.2 Restoring a backup

Requirement

- You have a backup of a CNET project that you wish to restore in CNET.

Procedure

1. Select the "Project" > "Restore..." menu.



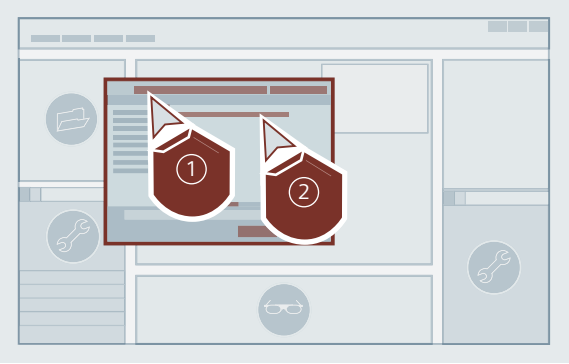
2. Confirm overwriting of the current project data.

Note

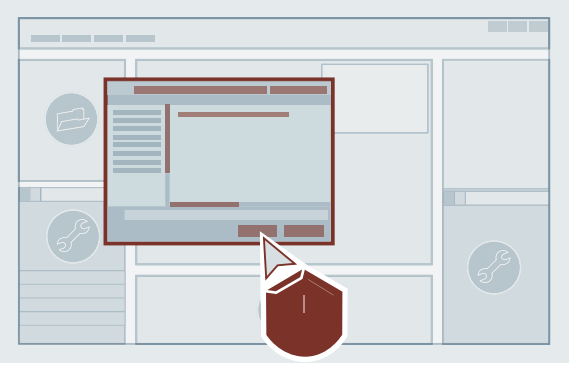
If you do not want any project data (for example hardware) to be stored yet in the CNET tool, continue with step 3.



3. Select the corresponding JAR file at the storage location of the backed up project.



4. Confirm your selection.



Result

The backup of the desired project has been restored in the CNET tool.

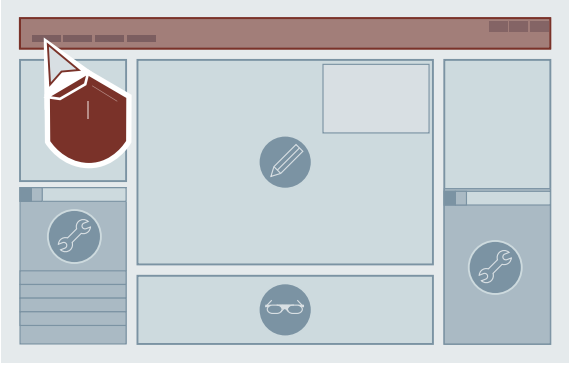
8.3.2.3 Emptying a project

Requirement

- The project environment in CNET is already filled with data.

Procedure

1. Select the "Project" > "Reset..." menu.



2. Select the removal of all project data with (a) or without (b) project backup:
   - a) Select "Backup".
   - b) Select "Continue with reset".



3. Further steps according to selection a or b:
   - a) Proceed as with step 2 to 3 of the following chapter:
     Creating a backup (Page 52)
   - b) Confirm.



Result

The project environment of CNET is empty.

8.3.2.4 Viewing the CNET and plugin version

Requirement

- The CNET tool is opened.

Procedure

Select the "Help" > "About CN 4100 Engineering Tool" menu.



Result

The dialog with the information on the version of the CNET is displayed.

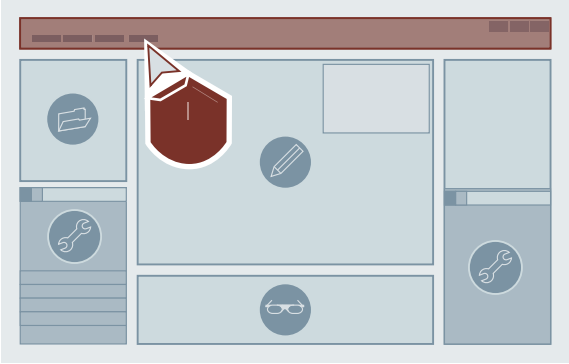
8.3.2.5 Displaying documentation

Requirement

- The CNET tool is opened.

Procedure

Select the "Help" > "Documentation" menu.



Result

The CNET documentation is displayed.

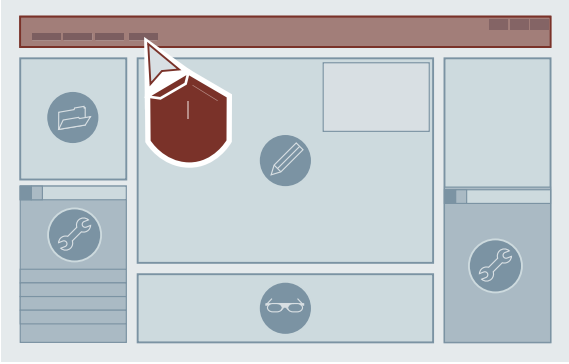
8.3.2.6 Displaying keyboard shortcuts

Requirement

- The CNET tool is opened.

Procedure

Select the "Help" > "Shortcuts" menu.



Result

The keyboard shortcuts for CNET operation are displayed.

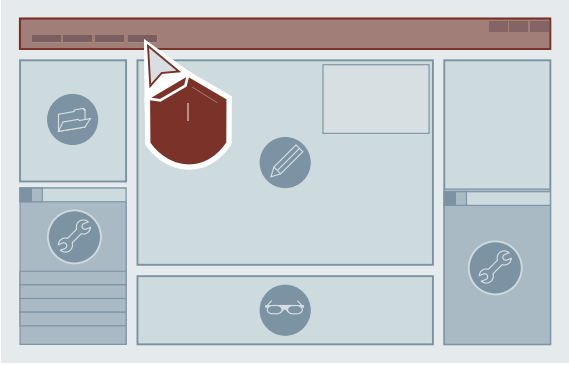
8.3.2.7 Having OSS components displayed

Requirement

- The CNET tool is opened.

Procedure

Select the "Help" > "OSS components" menu.



Result

The dialog with the information on the OSS components of CNET is displayed.

8.3.2.8 Terminating CNET

Requirement

- The CNET tool is opened.

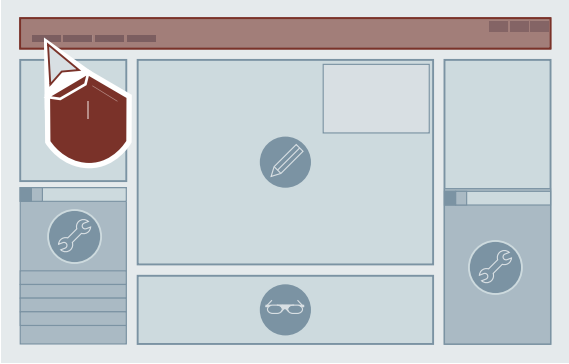
Note

Saving the project

Changes are stored automatically in CNET. The changes in the project are retained in CNET even after the CNET tool has been terminated.

Procedure

Select the "Project" > "Quit" menu.



Result

The CNET tool is closed.

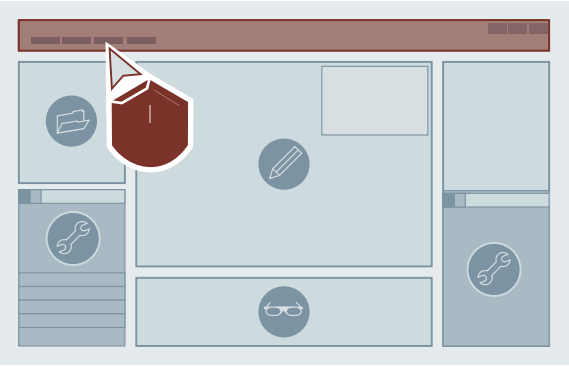
8.3.2.9 Navigating to an instance

Requirement

- The CNET tool is opened.
- One or more blocks have been created.

Procedure

1. Select the "Tools" > "Navigate to instance" menu.



2. Select the block (the desired instance).

Note

You can select the block via its path or by an entry in the input field. The input field allows you to search for any data (ID, name, display name) that is included in the current selection list.



3. Optional: Close the dialog with "OK".

Result

The selected block is displayed in the work area.

8.3.2.10 Showing defaults

Requirement

- The CNET tool is opened.

Procedure

1. Select the "Extras" > "Preferences" menu.



2. Optional: Select the desired settings and confirm them.

Note

The dialog is subdivided into different areas. It consists of:

- Settings for confirming deletion:
  Select the option per displayed topic.
- Settings for the display:
  Select the representation mode of CNET.



Result

The defaults are displayed. The new settings are applied by CNET after confirmation.

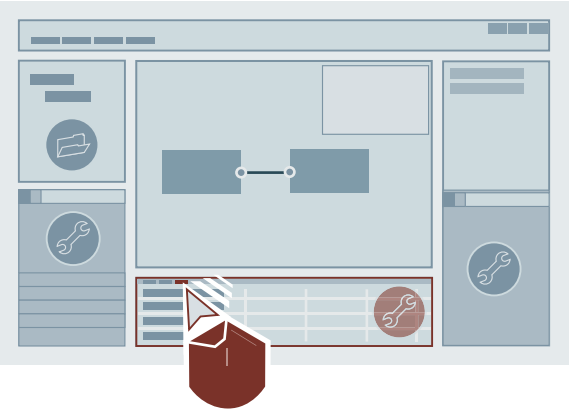
8.3.2.11 Detaching a section from the parameter area

Requirement

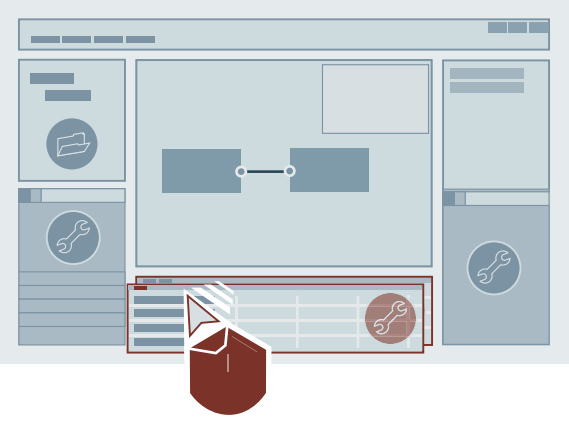
- The CNET tool is opened.

Procedure

1. Open the CNET defaults.
   Disable the "Docks always on top" option.
   Showing defaults (Page 61)
2. Select a section of the parameter area and drag it to the side.



3. Drop the desired section at the desired position.



Result

The selected section of the parameter area has been detached and is shown as a separate window.

8.4 Navigation area

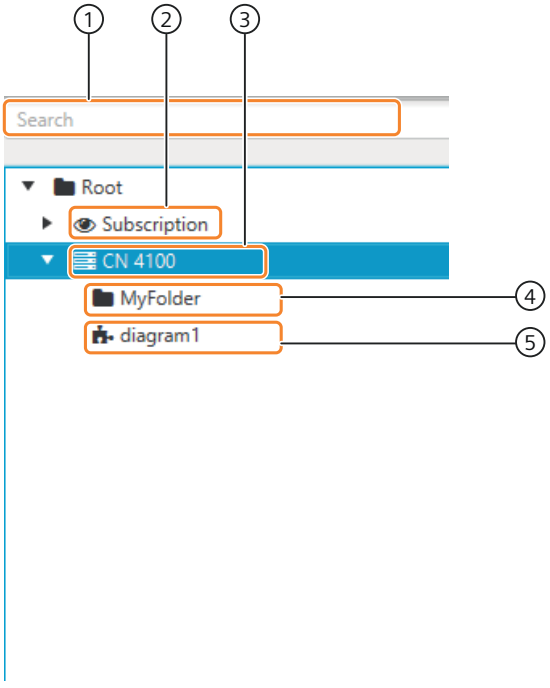
8.4.1 Navigation area

Definition

The navigation area is an area of the CNET tool in which the user can manage and operate the hardware of their project. In addition, CFCs can also be created via this area.

Structure

The navigation area has the following structure:



①	Search	Offers the possibility to filter the project navigation by the name of the hardware. Filter area (Page 221)
②	Subscription	Offers the possibility to create, open or edit watch tables. Watch table (Page 195) ("Subscription")

③	Hardware "CN 4100"	CN 4100 (Page 66) Configuration area (Page 156) Hardware properties (Page 157) Online mode (Page 67) ("Activate engineering") Offline mode (Page 67) ("Deactivate engineering") Device Cockpit (Page 67) Update device state (Page 70) Time Server Status (Page 72) ("Time") Ethernet Trace (Page 73) Firmware update (Page 75) OPC UA Certificate (Page 75) Automatic Status Update (Page 74)
④	Created folder "MyFolder"	Folder (Page 148)
⑤	CFC "diagram1"	CFC (Page 134) ("Diagram")

8.4.2 Step-by-step instructions

8.4.2.1 Moving an object in the navigation area

Requirement

- The navigation area is visible.

Procedure

1. Select an object (e.g. a folder) in the navigation area.



2. Move the object to the new position.



Result

The position of the selected object in the navigation area is changed.

8.4.3 CN 4100

8.4.3.1 CN 4100

Definition

The CN 4100 is the hardware that can be created via the navigation area.

Structure

A CN 4100 communication system is referenced as a single hardware object in the project. The core of the hardware object is the CN 4100 CPU that can be created redundantly.

A hardware object consists of:

- One or two CPUs
- Optionally one or two communications modules.

8.4.3.2 Online mode

Definition

The online mode is the state of the project in CNET in which the project data are synchronized with the hardware and the programs are executed. Reset and Reboot of the hardware are not available in online mode.

8.4.3.3 Offline mode

Definition

The offline mode is the state of the project in CNET in which the project data are not synchronized with the hardware. The offline mode is used to configure the hardware and to create the project data.

8.4.3.4 Device Cockpit

Definition

For the Device Cockpit, this is an area of the CNET tool that provides the user with an overview of the status of a hardware.

Note

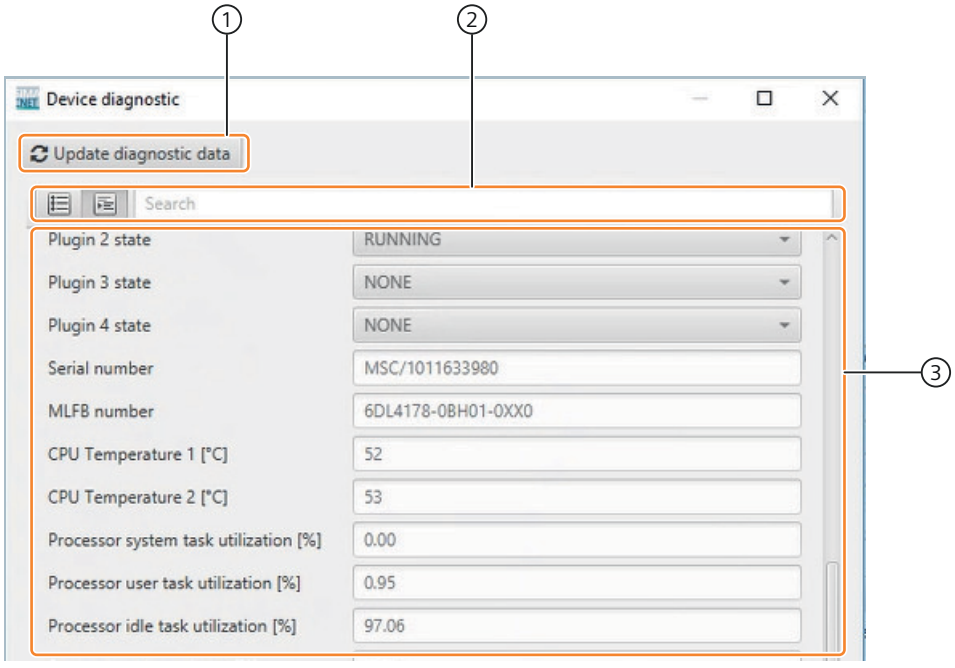
Save log data

Log data of the hardware can also be read out and saved as a compressed file on the PC as part of the diagnostics.

Save log file (Page 108)

Structure

The Device Cockpit is structured as follows:



①	Update diagnostic data	Button. Enables you to refresh the displayed diagnostics data. Updating diagnostics data (Page 107)
②	Search	Is used as entry field for the search for a hardware property. Filter area (Page 221)
③	Diagnostics data	Shows the status of a hardware. Is only shown for an initialized and connected hardware. Diagnostics data (Page 68)

8.4.3.5 Diagnostics data

Definition

The diagnostics data are an area of the Device Cockpit and provides an overview of the status of a CPU to the user.

Structure

The diagnostics data have the following structure:

①	System state	System status 0: NONE 1: SHUTDOWN 2: STARTING 3: STOPPED 4: STARTED 5: STARTED_SYNC 6: STANDBY 7: MASTER_STANDALONE 8: MASTER_SYNC 9: MASTER_REDUNDANT
②	Internal automation cycle slice	Part of the internal automation cycle
③	Activation command	Activation command 0: No command 1: Switch program 2: Commit activation 3: Cancel activation 4: Stop program
④	Requested RTC system state	Requested system status of the Runtime Container 0: NONE 1: SHUTDOWN 2: STARTING 3: STOPPED 4: STARTED 5: STARTED_SYNC 6: STANDBY 7: MASTER_STANDALONE 8: MASTER_SYNC 9: MASTER_REDUNDANT
⑤	Status plant bus communication	Status of the plant network 0: OK >0: Failure counter
⑥	Number processing modules	Number of instances processed
⑦	Cycle processing time	Processing time of the instances [ms]
⑧	RT communication status	Status of the redundancy connections 0: OK 1: OK_LINK1_FAILED 2: OK_LINK2_FAILED 3: ERROR
⑨	Pending activation	Activation process 0: false 1: true

⑩	Cumulative cycle shift	Cumulative cycle switchover
⑪	Plugin [0..4] state	Status of the plugin 0: NONE 1: STARTING 2: RUNNING 3: SHUTDOWN_REQUIRED

Additional information on the hardware is:

- Serial number (in ASCII format)
- Order number (in ASCII format)
- MAC address
- Temperature of the CPU
- Memory information
- Utilization of network interfaces

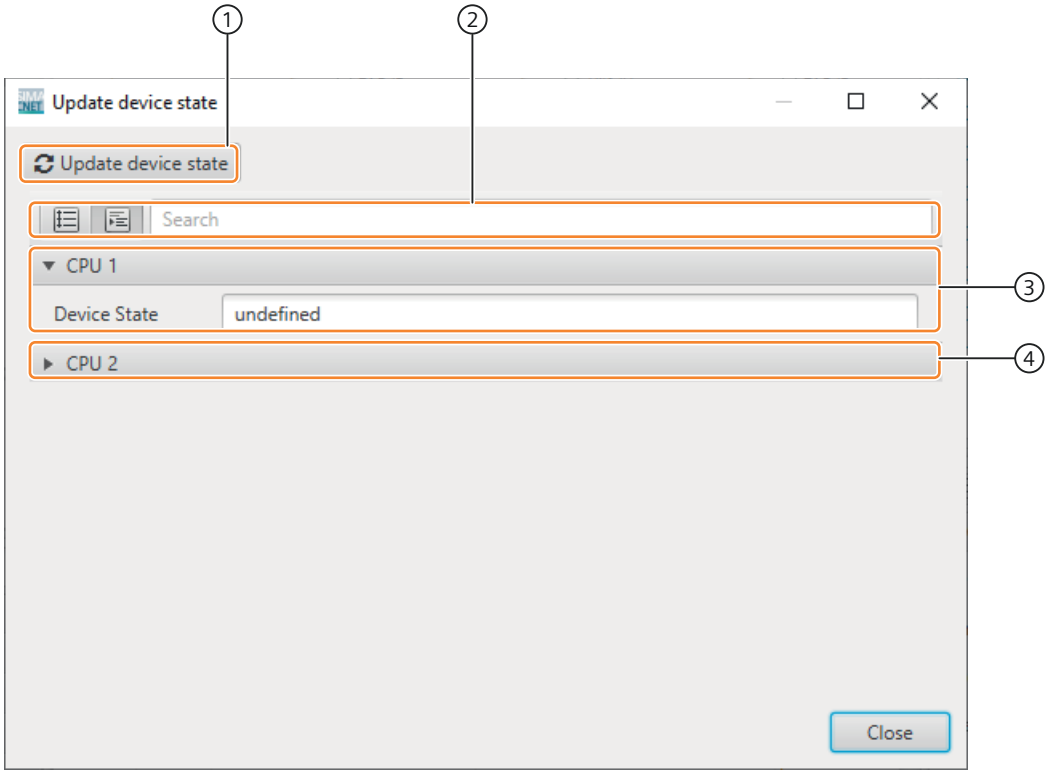
8.4.3.6 Update device state

Definition

For the Update device state, this is an area of the CNET tool which shows the user the device status of the CPUs of the CN 4100.

Structure

Update device state is structured as follows:



①	Update device state	Button. Enables you to refresh the displayed data. Updating device status (Page 113)
②	Search	Used to adjust the display. Filter area (Page 221)
③	Device status of CPU 1	Shows the device status of the respective CPU. • "undefined" = CN 4100 cannot be communicated with yet. Possible causes: – No connection established / CN 4100 is not connected – The assignment of an IP and the initialization of the device are missing. • "Upload of system configuration is possible" = Communicating with CN 4100. CN 4100 can be configured. • "configured" = CN 4100 is configured and ready for engineering. Display device status (Page 111)
④	Device status of CPU 2	

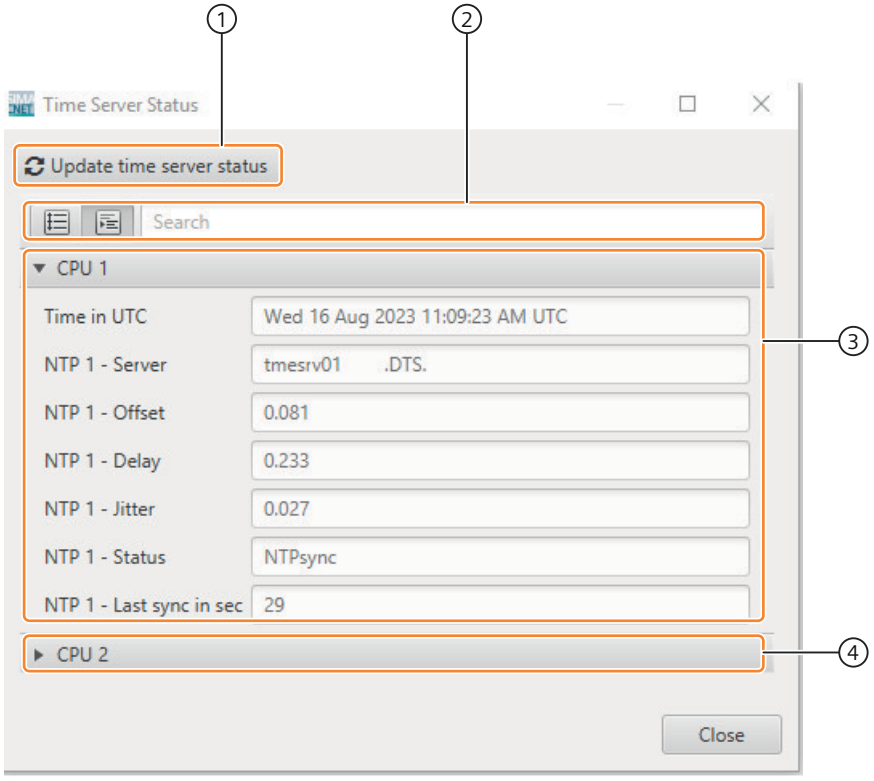
8.4.3.7 Time Server Status

Definition

The Time Server Status is an area of the CNET tool that shows the user the status of the time servers of the CN 4100.

Structure

Time Server Status is structured as follows:



①	Update time server status	Button. Enables you to refresh the displayed data. Updating the status of the time server (Page 115)
②	Search	Used to adjust the display. Filter area (Page 221)

③	CPU 1 Status of NTP server [1..2]	Shows the status of the NTP server [1..2] of the relevant CPU. The time is specified in UTC. Only information (NTP server, CPU) that has also been configured beforehand is displayed. • Time in UTC = Shows the time transmitted from the server (in UTC). • NTP... -Server = Displays the name of the host or the IP of the server. • NTP... -Offset = Displays the time offset [ms] from the server. • NTP... -Delay = Displays the time delay [ms] during which the telegrams are retrieved. • NTP... -Jitter = Displays the time jitter [ms] of the clock. • NTP... -Status = Displays the status of the last 8 telegrams. • NTP... -Last sync in sec = Displays the period [s] of the last synchronization with the server. Displaying the status of the time server (Page 113)
④	CPU 2 Status of NTP server [1..2]	

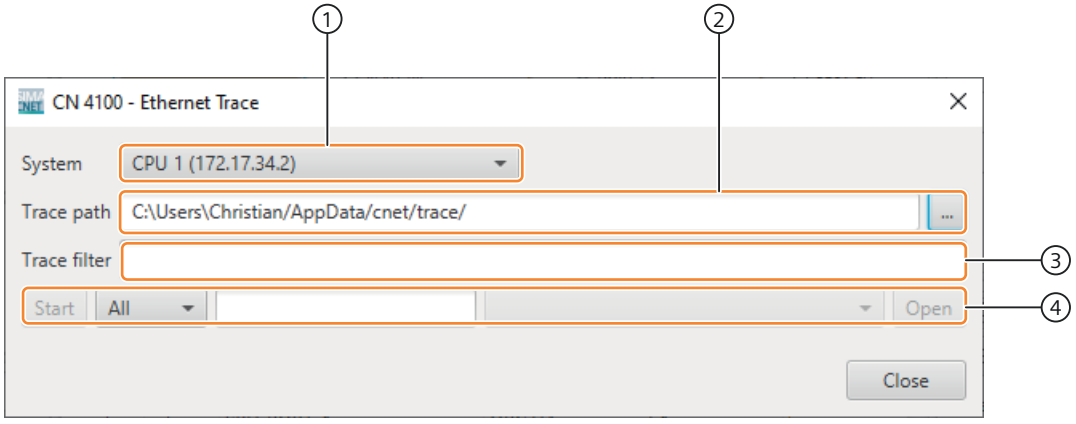
8.4.3.8 Ethernet Trace

Definition

Ethernet Trace is an area of the CNET tool that serves to log errors on the respective Ethernet connection.

Structure

Ethernet Trace is structured as follows:



①	System	Button. Allows you to select the CPU. Its IP address is displayed.
②	Trace path	Shows the storage location for logging in the OS used. The storage location can be changed using the button.
③	Trace filter	Allows you to make specific settings relating to recording or sorting of the information to be logged.
④	Start, All, Open	• "Start" button: Starts logging. • First drop-down list ("All"): Allows you to select a specific connections or all connections to be logged. The following input field is used to name the logging. • Second drop-down list: Used to select a created log file. • "Open" button: Opens the selected log file.

8.4.3.9 Automatic Status Update

Definition

The Automatic Status Update is the state of the hardware in CNET in which the display of the device status is automatically updated. The accessibility of the hardware and the CFCs is checked in the process. The status of the accessibility is displayed in the navigation area at the hardware object and the lower-level objects.

Display at the object

The object in the navigation area is shown in color:

- red = hardware not accessible
- yellow =
  - Data was not transferred or not transferred completely to the hardware
  - Hardware partly not accessible (with redundant CPU)
- green = hardware accessible

Application

Activating automatic update of the device status (Page 116)

Deactivating automatic update of the device status (Page 117)

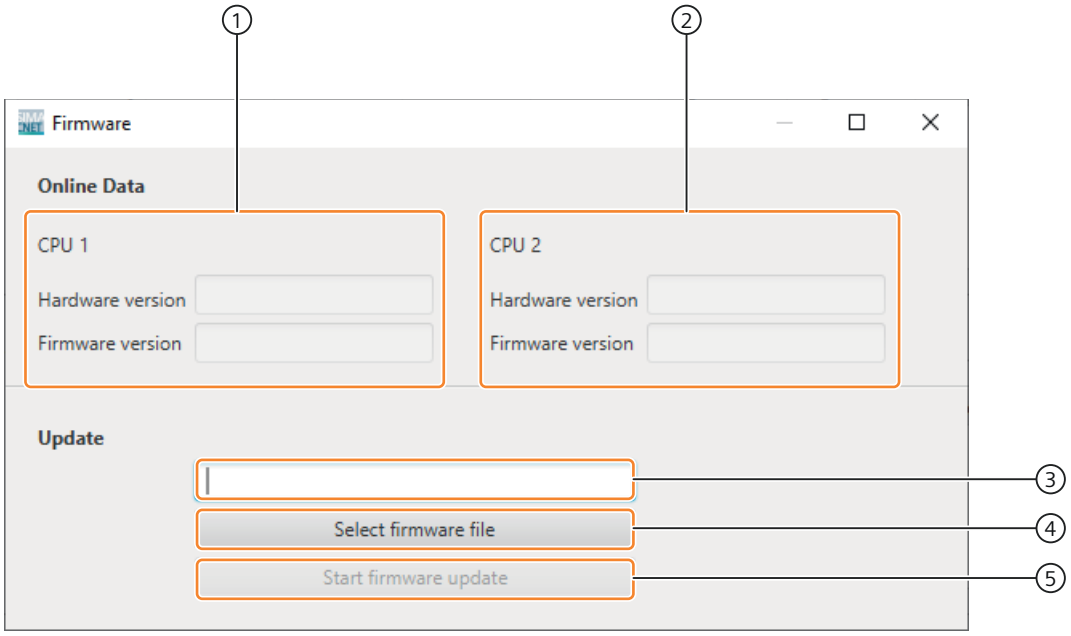
8.4.3.10 Firmware update

Definition

For the Firmware update, this is an area of the CNET tool that informs the user about the firmware and hardware version of the CPU and provides functions for updating.

Structure

Firmware update is structured as follows:



①	Online Data CPU 1	Displays the information for CPU 1.
②	Online Data CPU 2	Displays the information for CPU 2. The information is only displayed if there is redundancy.
③	Text box	Shows the file name of the selected firmware update.
④	Select firmware file	Button. Opens the storage location of a firmware update on the PC.
⑤	Start firmware update	Button. Opens the dialog for selecting the CPU and starts the firmware update.

8.4.3.11 OPC UA Certificate

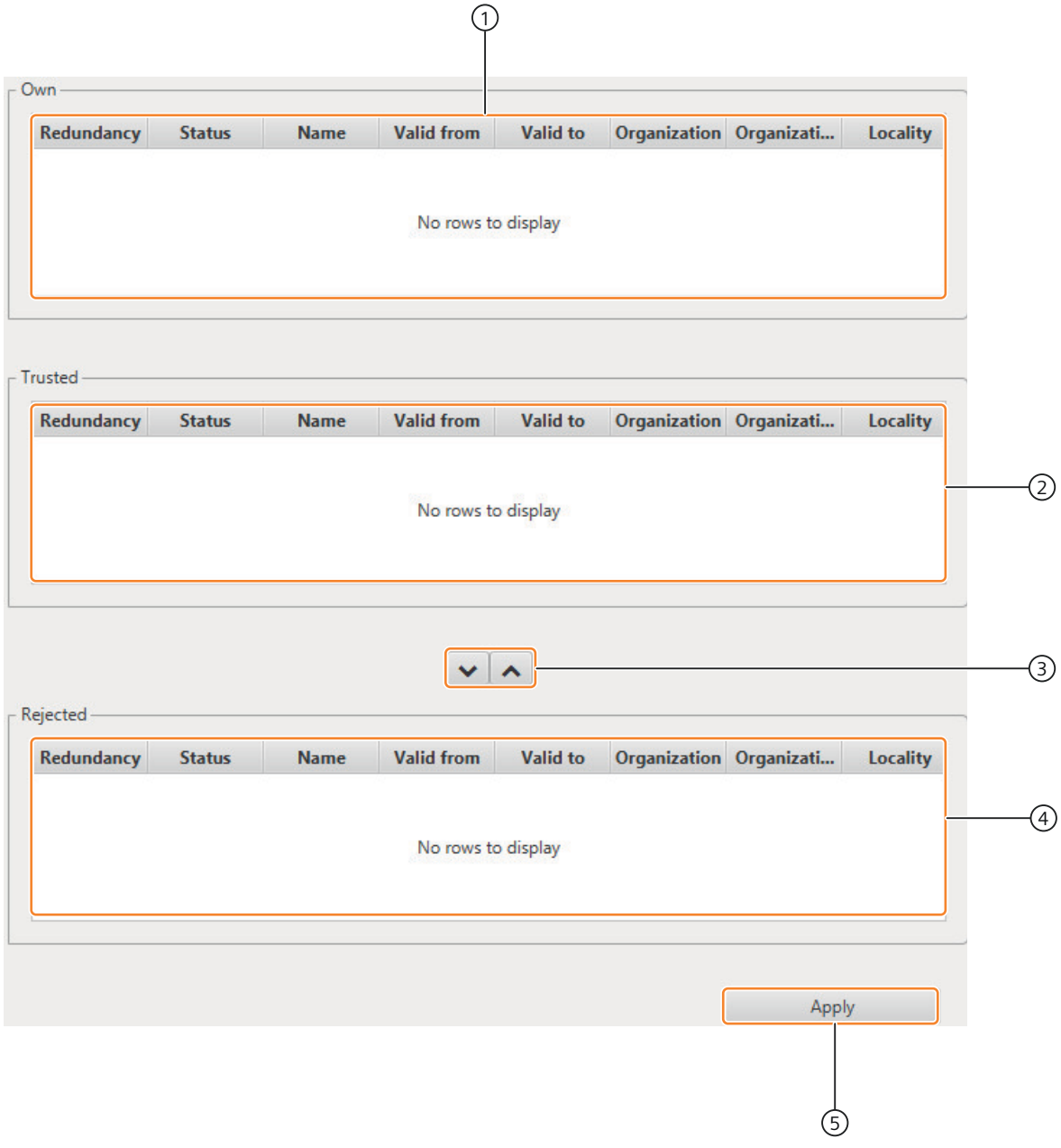
Definition

The OPC UA Certificate is an area of the CNET tool which provides an overview of the utilizable certificates for OPC UA to the user. OPC UA Certificate is displayed in the work area.

OPC UA Client, OPC UA Server | Certificates (Page 240)

Structure

The OPC UA Certificate is structured as follows:



①	Own	Shows an overview of imported certificates.
②	Trusted	Displays an overview of the certificates declared as trustworthy. OPC UA | Open overview of the certificates (Page 124)
③	Arrows	Buttons for up and down. Are used to move certificates from one list into the other list.

④	Rejected	Displays an overview of the "rejected" certificates. All the unanswered certificates transferred by the server are included in this list. Removing an OPC UA certificate (Page 128) Displaying an OPC UA certificate (Page 132)
⑤	Apply	Button. Provides the possibility to apply the settings for using a certificate. Integrating an OPC UA certificate (Page 127)

8.4.3.12 Step-by-step instructions (CN 4100)

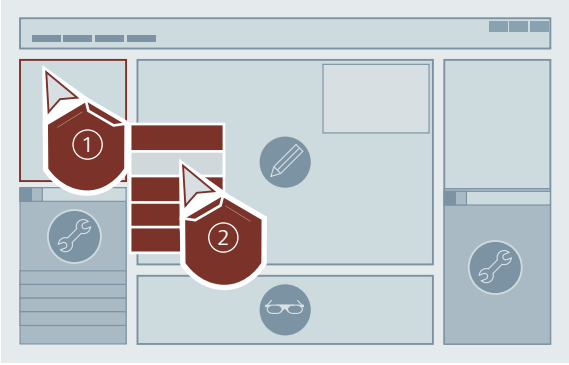
Creating the CN 4100

Requirement

- The navigation area is visible.
- The hardware properties are known.
  Hardware properties (Page 157)

Procedure

1. In the navigation area, select > "CN 4100" >"Add".



2. Select the matching configuration data for the hardware.

Note

Use unique names

In CNET hardware must be differ by name from each other. The names of the CPUs must also end in "-A" or "-B".

Note

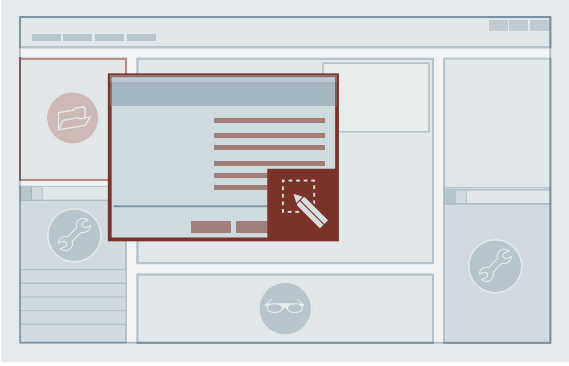
Note red edges

The red edges in the text boxes mark checked entry fields. The verification has the purpose of avoided faulty entries.

Note

Avoid duplicate entries

Incomplete entries or entries used twice (e.g. for the Ethernet settings / IP address) are marked by CNET via the text box.

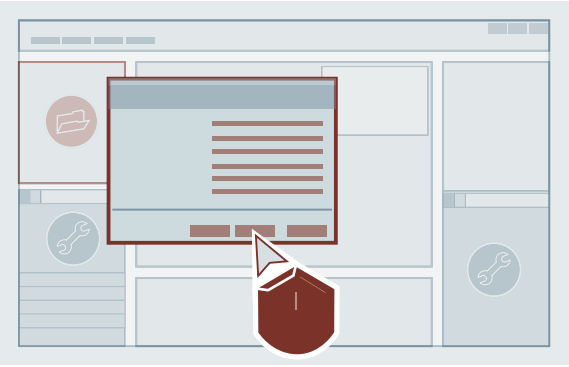


3. Select the Ethernet settings and select "Next".

Note

Specifying and separating subnets correctly

The different Ethernet interfaces of CN 4100 must use different subnets.



4. Perform steps 2 to 3 for all pages of the dialog.

Note

Further network settings

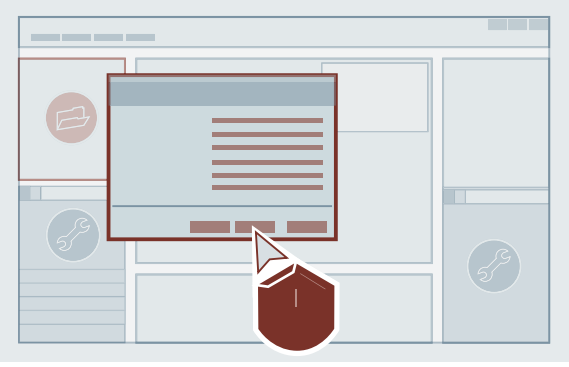
- Provide the IP addresses for the DNS servers and the logging server.
- Depending on the chosen method for time synchronization, additional settings will be displayed.

Note

Runtime settings

- Base Cycle: Displays the cycle time to be used for program processing.
- Peer-to-Peer: You can only make P2P settings after the hardware has already been created in CNET. Use the editing function for this purpose.

5. Select "Finish" on the last page.



Result

The hardware (CN 4100) has been created and configured.

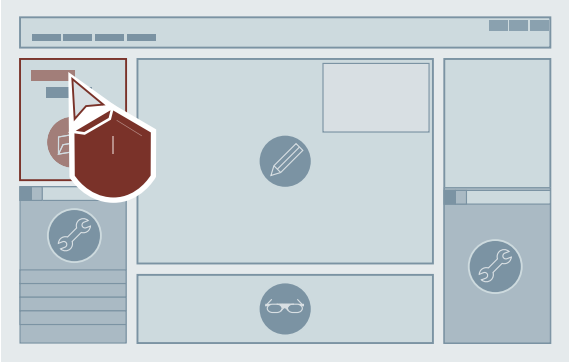
Processing the CN 4100

Requirement

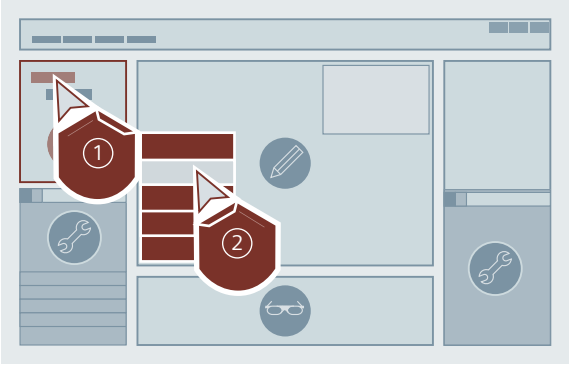
- The navigation area is visible.
- The hardware properties are known.
  Hardware properties (Page 157)
- The hardware has been created.

Procedure

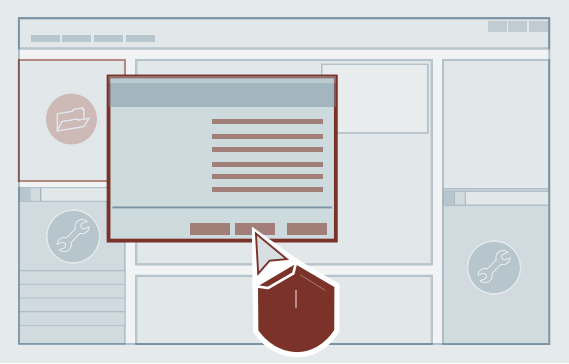
1. Select the hardware in the navigation area.



2. Select "Edit".



3. Navigate to the desired page.

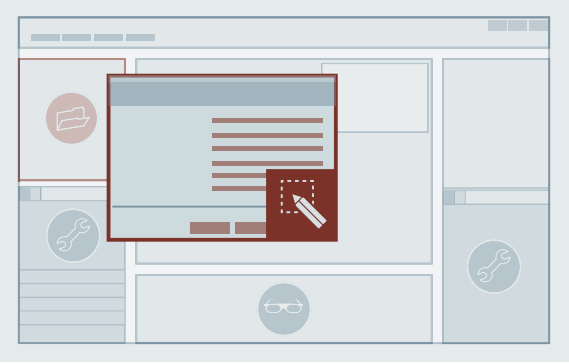


4. Change the value at the selected parameter.

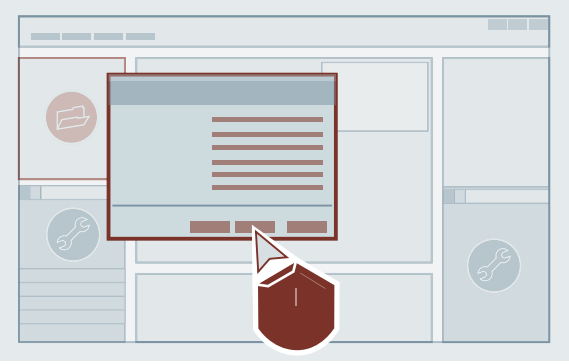
Note

Runtime settings

- Base Cycle: Displays the cycle time to be used for program processing.
- Peer-to-Peer Connections: Displays the selectable communication partners for a P2P connection.
  Change the settings if necessary.



5. Select "Finish" on the last page.



Result

The hardware data have been modified.

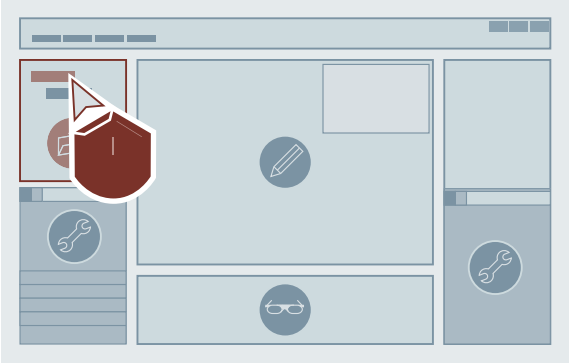
Remove the CN4100

Requirement

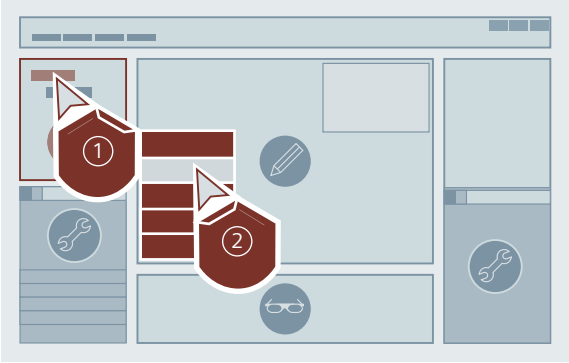
- The navigation area is visible.
- The hardware has been created.

Procedure

1. Select the hardware in the navigation area.



2. Select "Remove".



3. Confirm.



Result

The selected hardware has been removed.

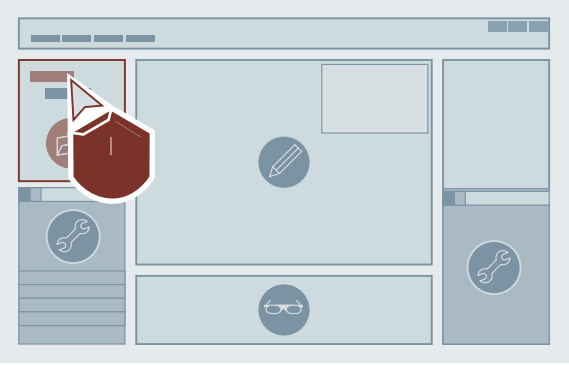
Deactivating forcing for all parameters

Requirement

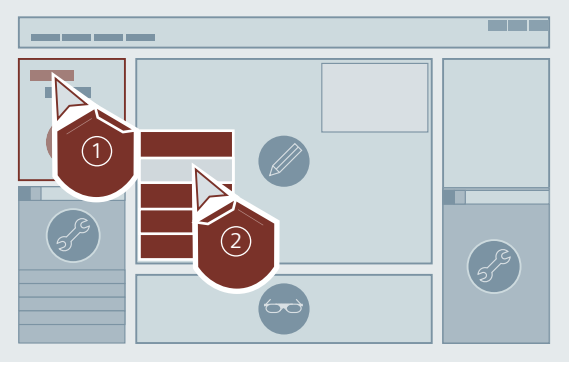
- The navigation area is visible.
- The hardware has been created.

Procedure

1. Select the hardware in the navigation area.



2. Select "CN 4100" > "Unforce".



Result

Forcing is canceled for all values of the selected hardware.

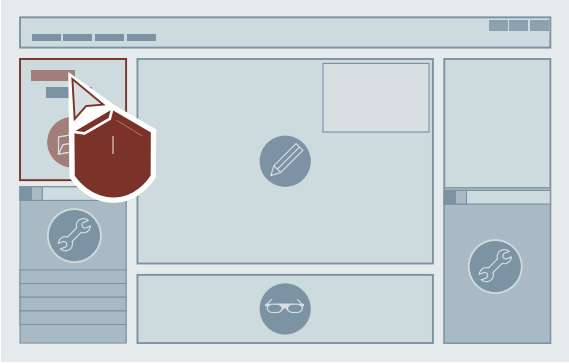
Initializing configuration data completely

Requirement

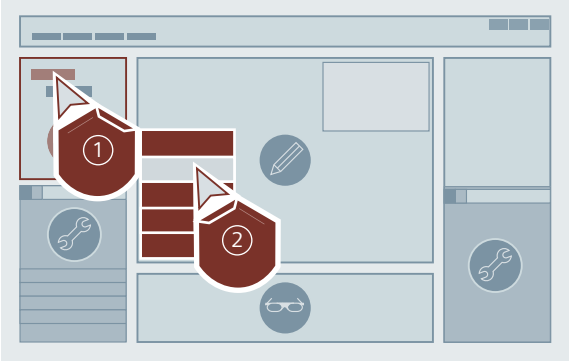
- The navigation area is visible.
- The hardware has been configured.
- The device password is known.
  User and password (Page 48)

Procedure

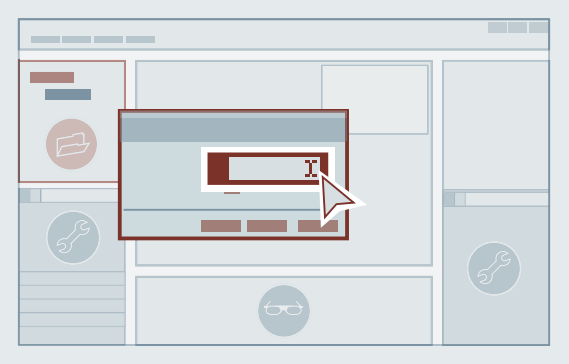
1. Select the hardware in the navigation area.



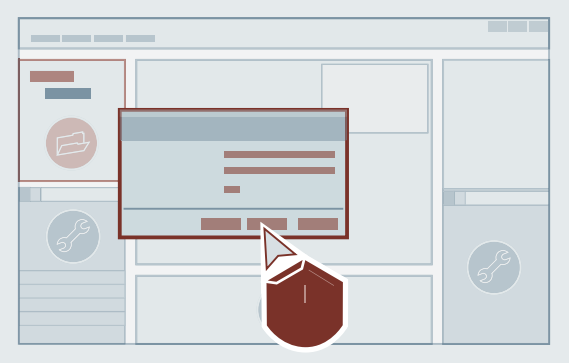
2. Select "CN 4100" > "Configure" > "Download system and runtime configuration".



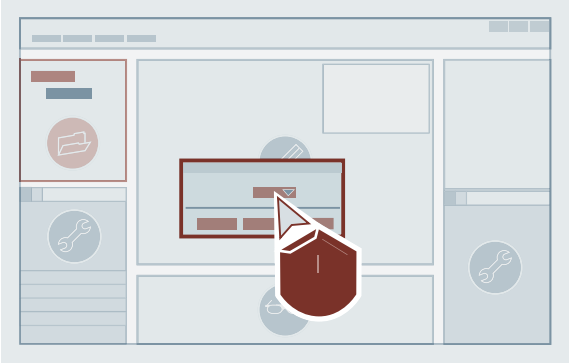
3. Enter the device password.
   – Optional addition: Select "Host verifier".



4. Select "Next".



5. Select the setting for the desired CPU.



6. Select "Finish".



Result

All configuration data of the hardware are compiled and uploaded into the CN 4100.

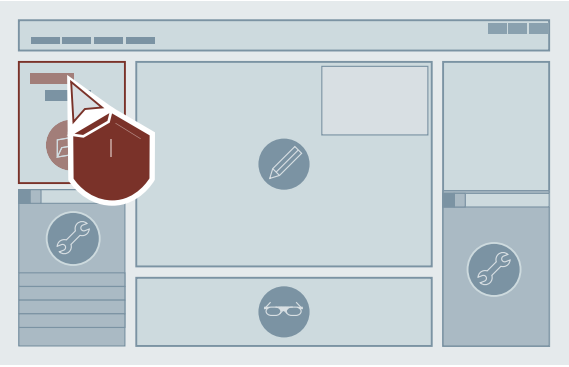
Initializing the configuration data partially

Requirement

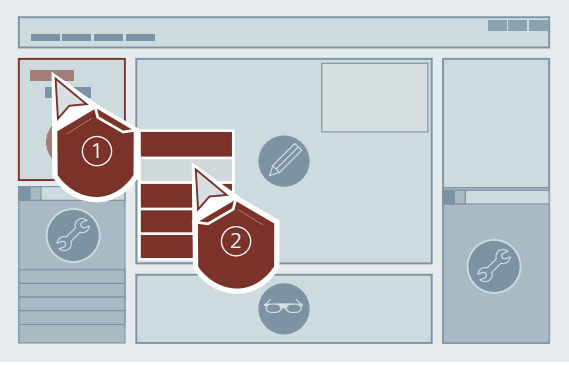
- The navigation area is visible.
- The hardware has been configured and initialized.
- Changes have been made to the configuration data.
- The device password is known.
  User and password (Page 48)

Procedure

1. Select the hardware in the navigation area.



2. Select "CN 4100" > "Configure" > "Download runtime configuration".



3. Proceed as of step 3 in the following section:
   Initializing the configuration data completely (Page 84)

Result

The changes of the configuration data are compiled and loaded into the CN 4100.

Export runtime

Note

Function not visible

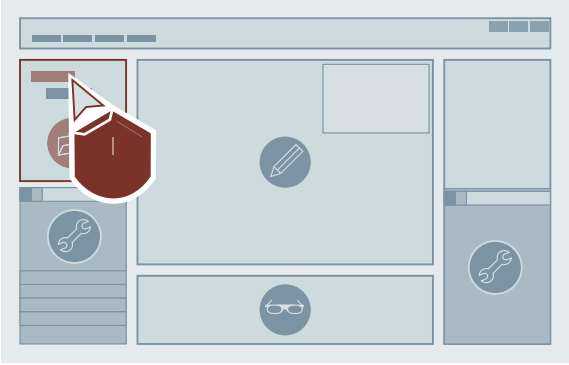
The "Export runtime" function has only been released for Siemens Support.

Requirement

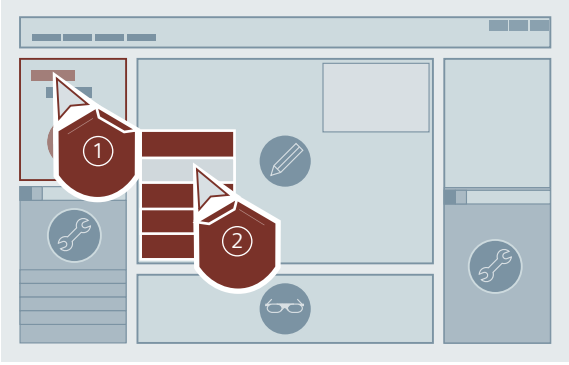
- The navigation area can be seen.
- The hardware has been configured and initialized.

Procedure

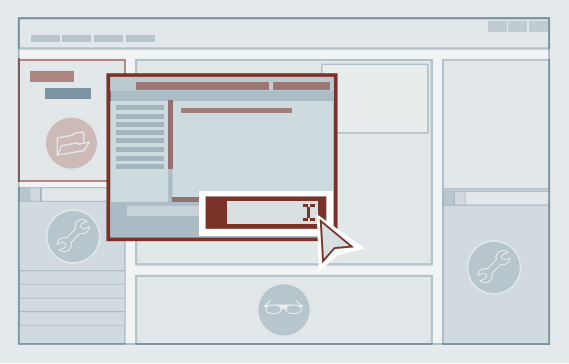
1. Select the hardware in the navigation area.



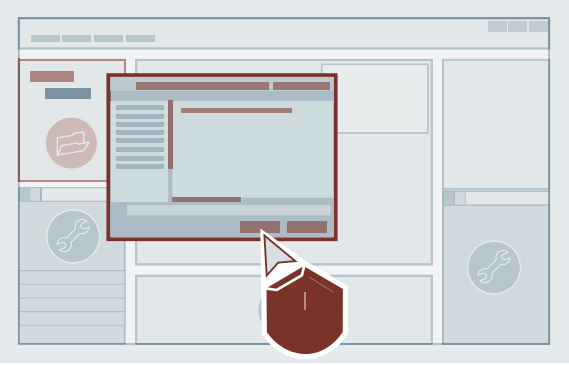
2. Select "CN 4100" > "Configure" > "Export runtime configuration".



3. Select the storage location and enter a name for the export.



4. Confirm.



Result

The changes of the configuration (runtime) are exported on the PC.

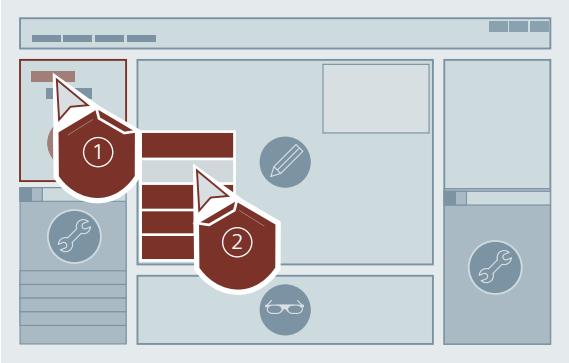
Loading a project and setting it online

Requirement

- The hardware has been created and configured.
- The configuration data are compiled and loaded.

Procedure

1. Select "Activate..." for the desired hardware in the navigation area.



   – Alternative for changed versions of project data that has already been transferred: Select an object for which changed project data is to be transferred.

   Note

   Selection in the work area possible

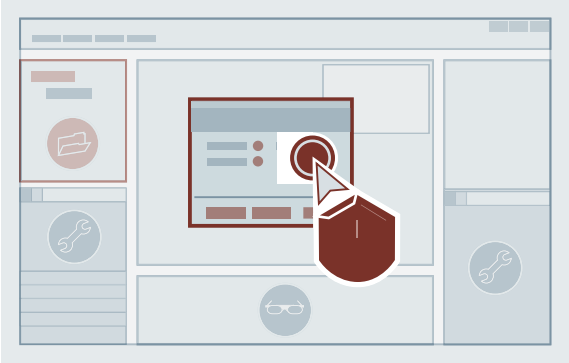
   Project data of selected objects can also be transferred from the work area.

   In the work area, select one or multiple objects and then "Activate selected instances".

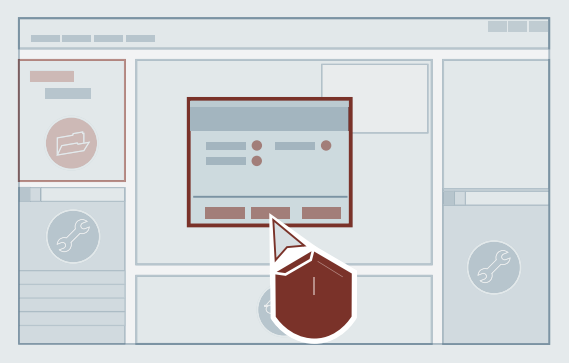
2. Select the settings for transferring the program data to the hardware.

   Note

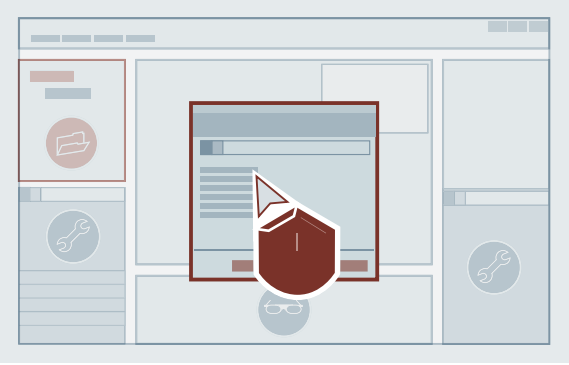
   - "Delta activation" selects only the revision of the program data for transfer.
   - "Execute instance parameter checkers" checks the enable of the resources for the parameters of the blocks. (important for black box blocks.)



3. Confirm the settings.



4. Optional (when changes have been made at the program):
   If necessary, check the changes of the shown instances.



5. Confirm.



Result

The project data are transferred to the hardware and the Engineering is activated. The project is in online mode.

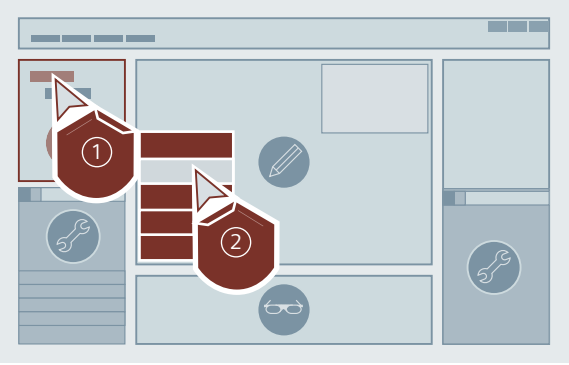
Setting a project offline

Requirement

- The project is in online mode.
  Online mode (Page 67)

Procedure

1. In the navigation area, select the desired hardware > "Deactivate...".



   – Alternative for changed versions of project data that has already been transferred: Select an object for which engineering is to be disabled.

   Note

   Selection in the work area possible

   Online mode of selected objects can also be selected from the work area.

   In the work area, select one or multiple objects and then "Deactivate selected instances".

2. Optional: Select "Disconnect".

   Note

   Only select this option if the program data has to be updated and the hardware is not aware of it. (e.g. after a firmware update without previous setting offline.)



3. Confirm.



Result

The project is in offline mode.

8.4.3.13 Step-by-step instructions (System & Diagnostic)

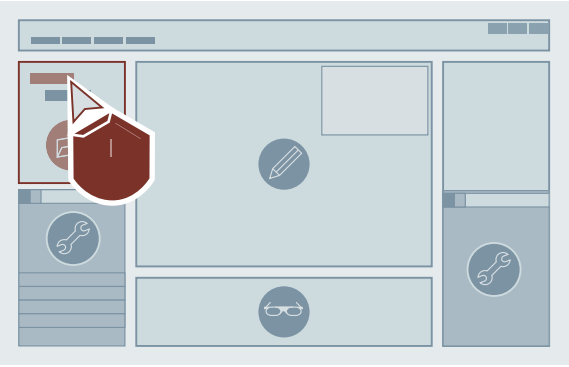
Changing the device password

Requirement

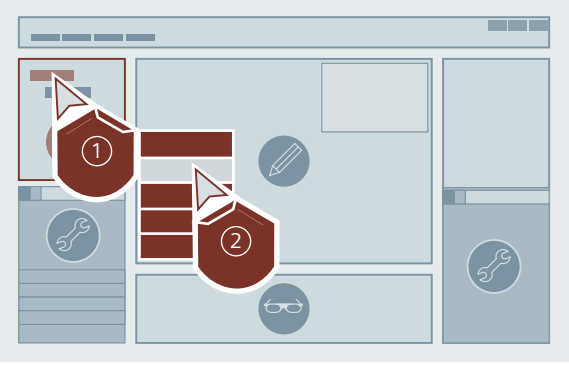
- The navigation area is visible.
- The hardware has been created.
- The device password is known.
  User and password (Page 48)

Procedure

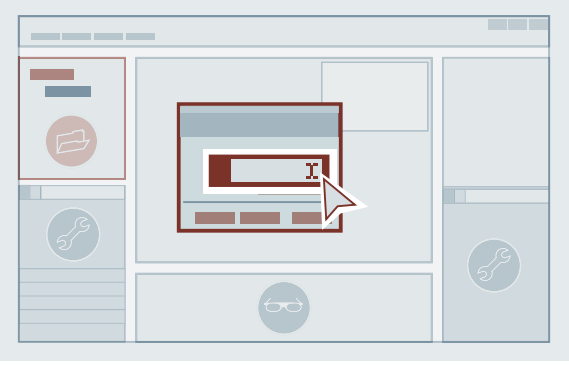
1. Select the hardware in the navigation area.



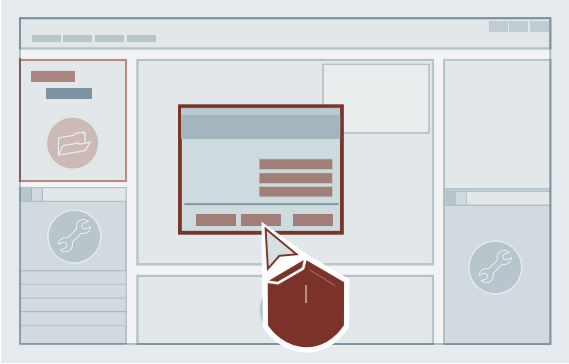
2. Select "CN 4100" > "System" > "Change device password".



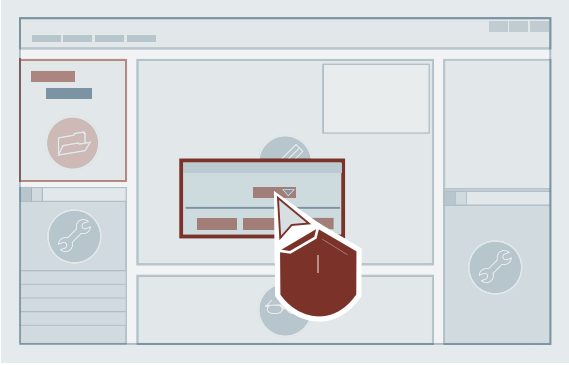
3. Enter the old password and a new one.



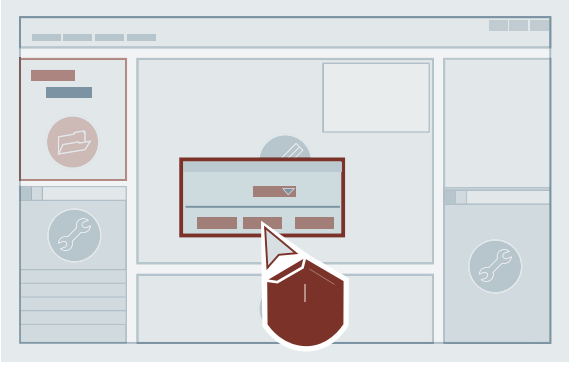
4. Select "Next".



5. Optional for redundancy:
   Select the setting for the desired CPU.



6. Select "Finish".



Result

The password of the selected hardware has been changed.

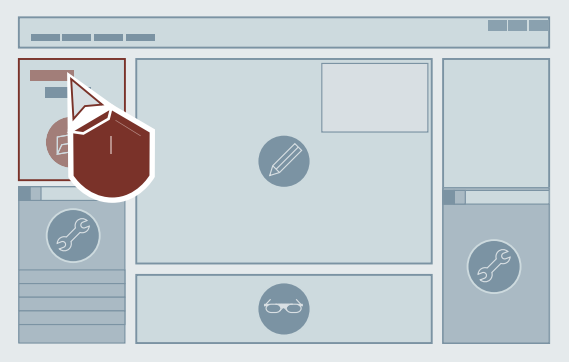
Checking the firmware

Requirement

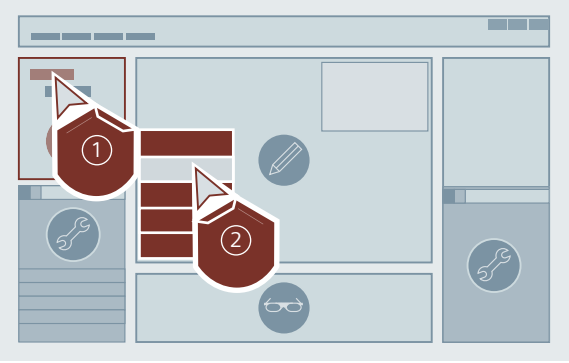
- Navigation area and work area are visible.
- An IP address has been assigned to the hardware.
- The Engineering System is connected to the hardware.
- The device password is known.
  User and password (Page 48)

Procedure

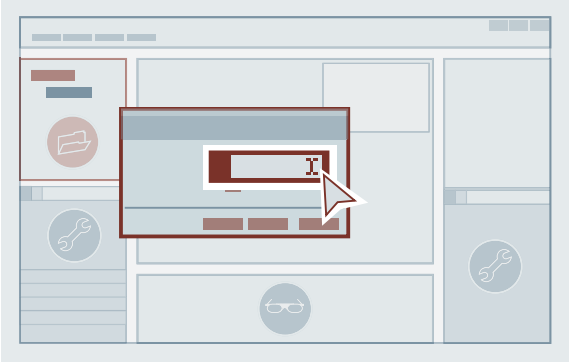
1. Select the hardware.



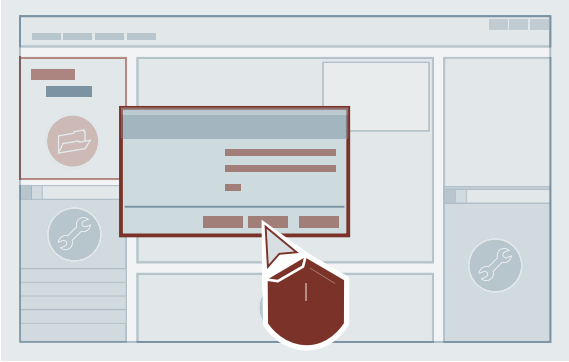
2. Select "CN 4100" > "System" > "Firmware update".



3. Enter the device password.
   – Optional addition: Select "Host verifier".



4. Confirm.



5. Take a look at the data.



Result

You have informed yourself about the firmware status of the hardware.

Loading the firmware update

Requirement

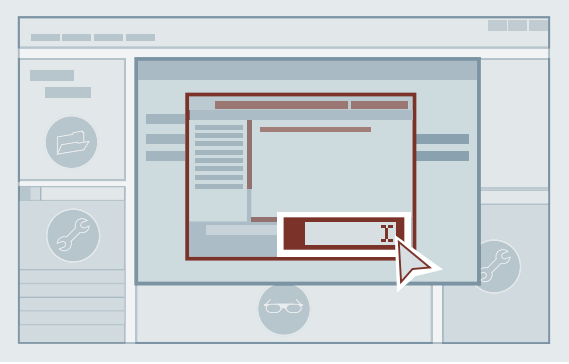
- Navigation area and work area are visible.
- An IP address has been assigned to the hardware.
- The Engineering System is connected to the hardware.
- The "Firmware update" page of the hardware is opened.
- The device password is known.
  User and password (Page 48)

Procedure

1. Select "Select firmware update".



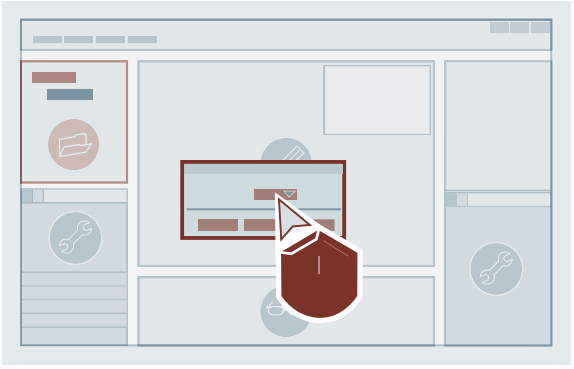
2. Select the storage location and name of the firmware update and confirm.



3. Select "Start firmware update".



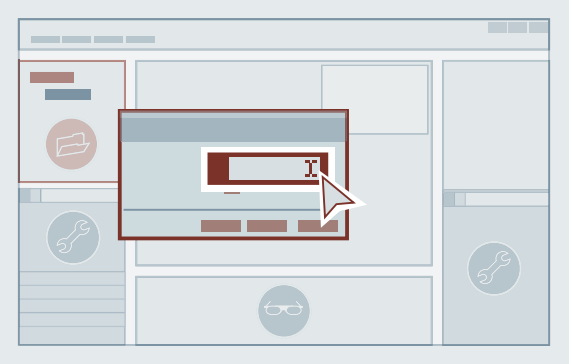
4. Select the CPU.



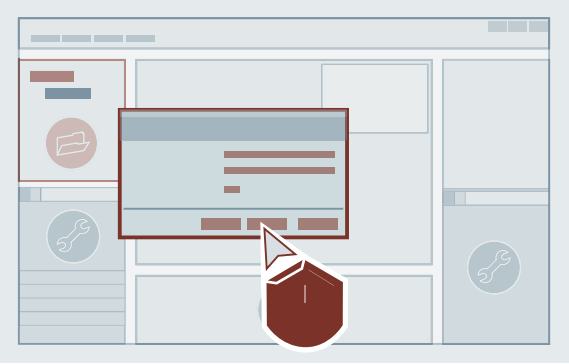
5. Confirm your selection.



6. Enter the device password.
   – Optional addition: Select "Host verifier".



7. Confirm.



Result

The firmware update is loaded to the respective CPU and the respective CPU is restarted. After the restart, the given CPU is accessible again.

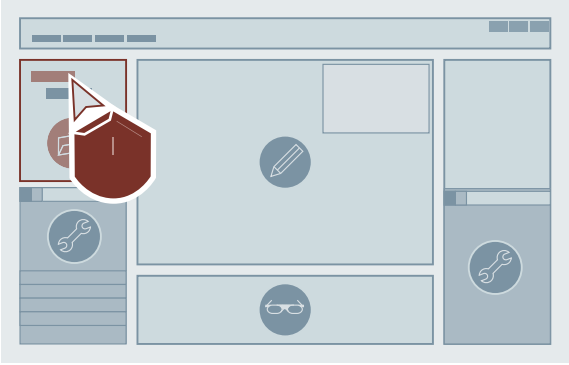
Restarting hardware

Requirement

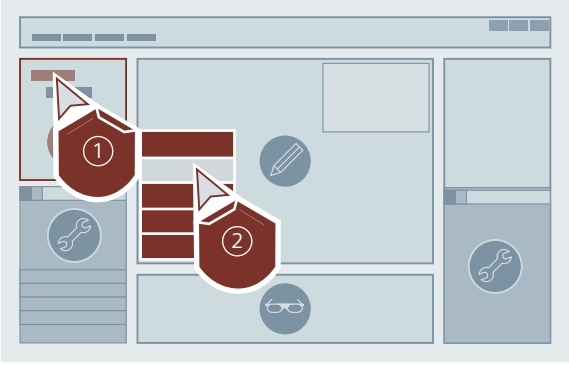
- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.
- The project (or the affected object "CN 4100") is in offline mode.
  Offline mode (Page 67)
- The device password is known.
  User and password (Page 48)

Procedure

1. Select the hardware.



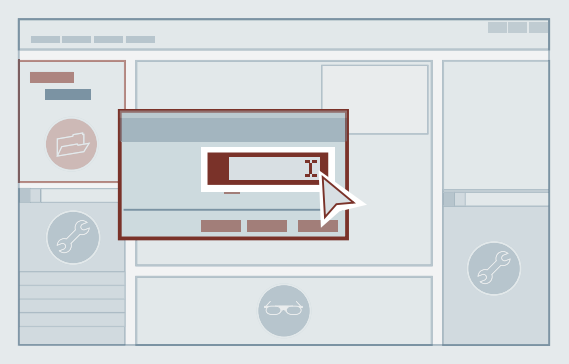
2. Select "CN 4100" > "System" > "Reboot device".



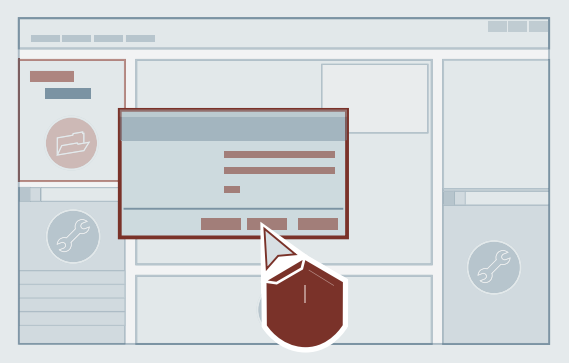
3. Confirm the restart.



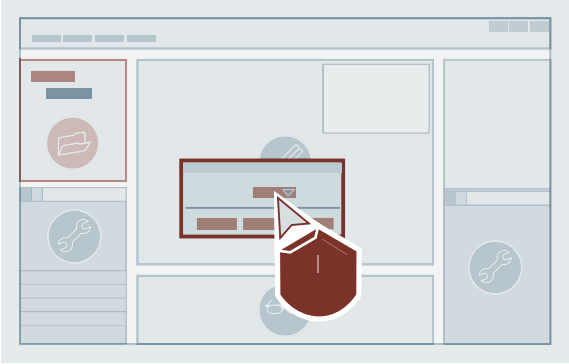
4. Enter the device password.
   – Optional addition: Select "Host verifier".



5. Select "Next".



6. Select the setting for the desired CPU.



7. Confirm.



Result

The selected hardware is rebooted.

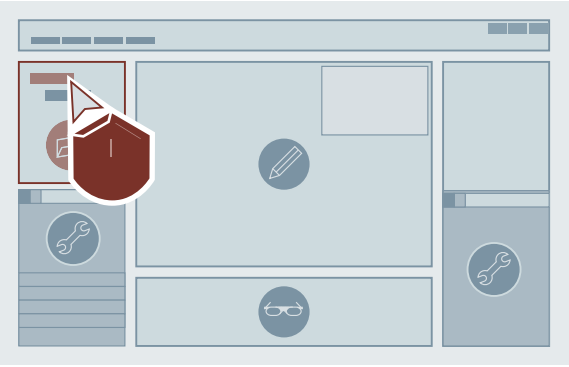
Resetting hardware

Requirement

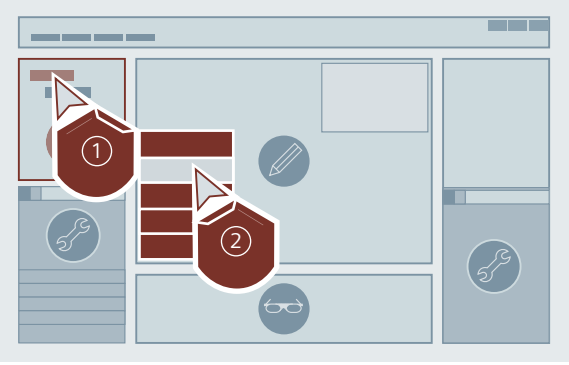
- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.
- The project (or the affected object "CN 4100") is in offline mode.
  Offline mode (Page 67)
- The device password is known.
  User and password (Page 48)

Procedure

1. Select the hardware.



2. Select "CN 4100" > "System" > "Reset device".



3. Confirm the reset.



4. Enter the password and select the CPU.
   See the steps beginning with step 4 in the following section:
   Restarting hardware (Page 101)

Result

The configuration of the selected hardware is reset to factory settings.

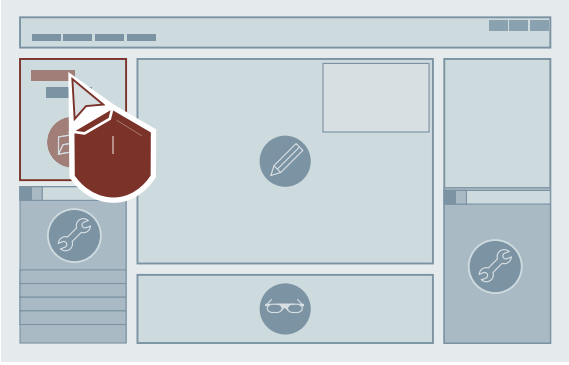
Displaying diagnostics data

Requirement

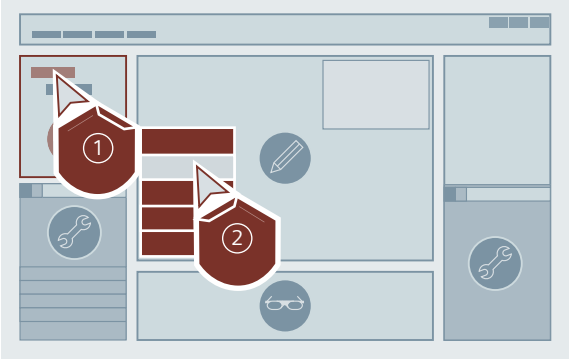
- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.

Procedure

1. Select the hardware.



2. Select "CN 4100" > "System" > "Diagnostic" > "Device Cockpit".



Result

The diagnostics data are displayed.

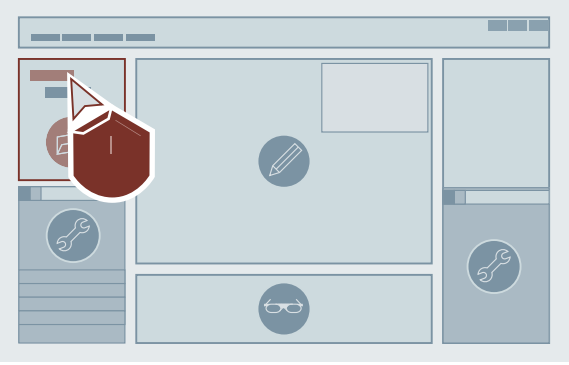
Updating diagnostics data

Requirement

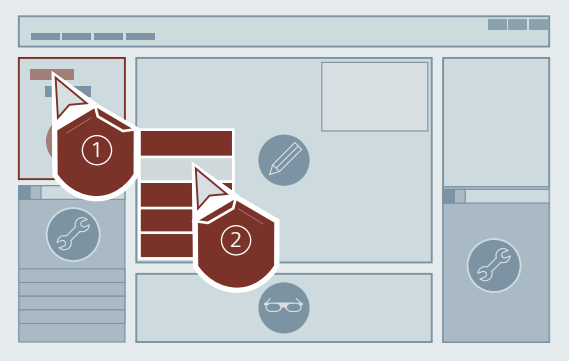
- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.

Procedure

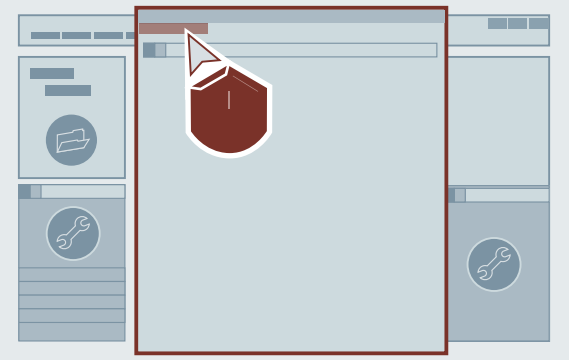
1. Select the hardware.



2. Select "CN 4100" > "System" > "Diagnostic" > "Device Cockpit".



3. Select "Update diagnostic data".



Result

The display of the diagnostics data in the Device Cockpit is updated.

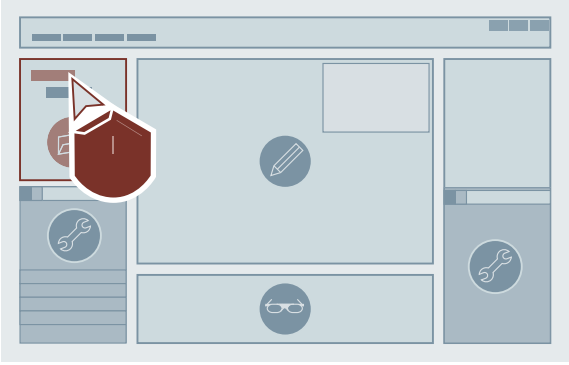
Saving a log file

Requirement

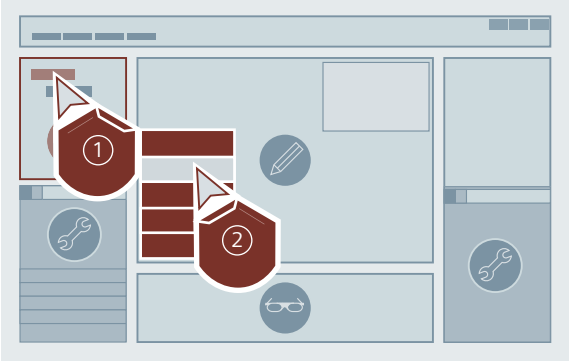
- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.
- The device password is known.
  User and password (Page 48)

Procedure

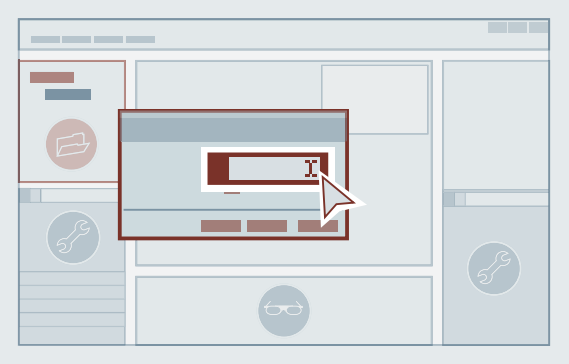
1. Select the hardware.



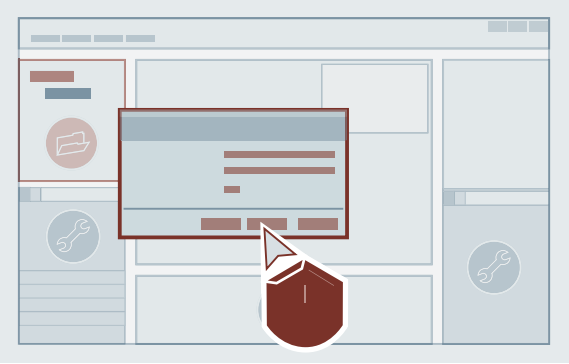
2. Select "CN 4100" > "System" > "Diagnostic" > "Logging dump".



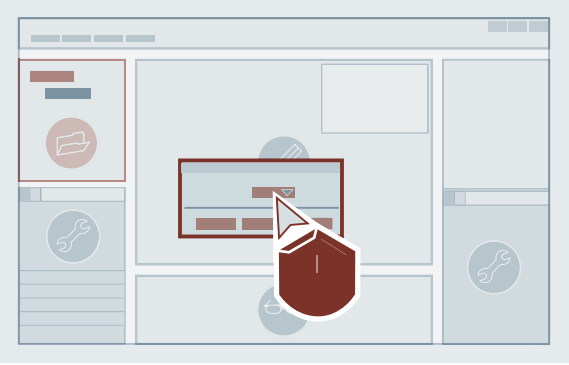
3. Enter the device password.
   – Optional addition: Select "Host verifier".



4. Select "Next".



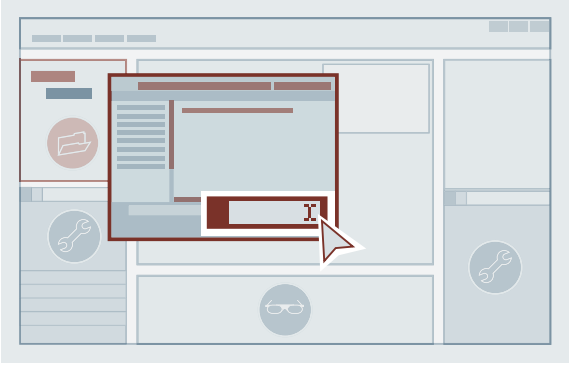
5. Select the setting for the desired CPU.



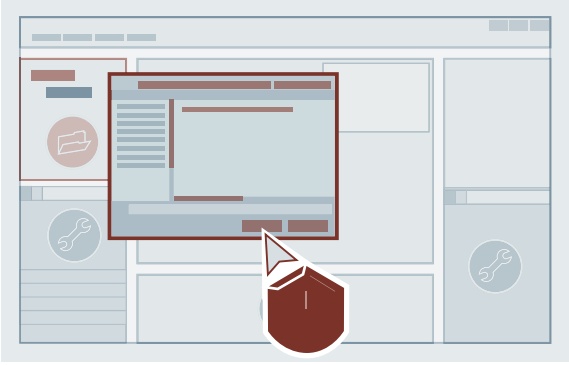
6. Select "Finish".



7. Select the storage location and enter a name for the log file.



8. Confirm.



Result

A log file has been saved in the system directory on your PC.

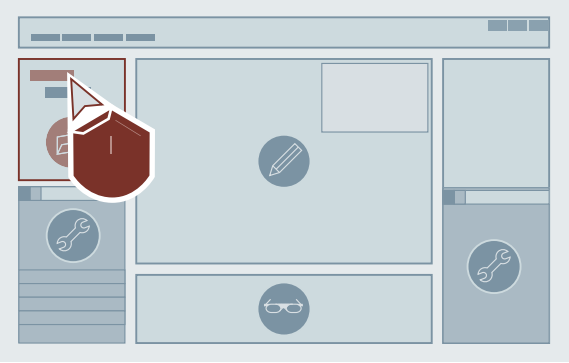
Display device status

Requirement

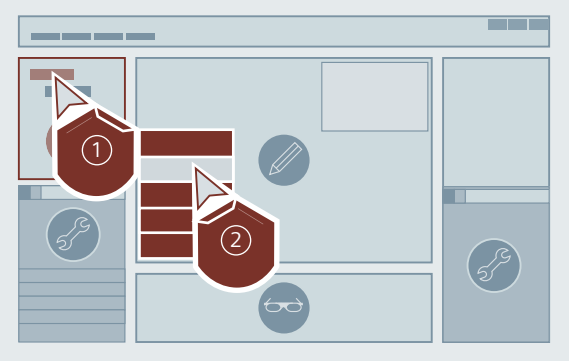
- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.
- The device password is known.
  User and password (Page 48)

Procedure

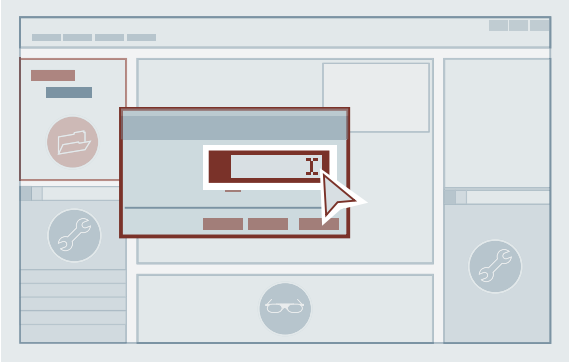
1. Select the hardware.



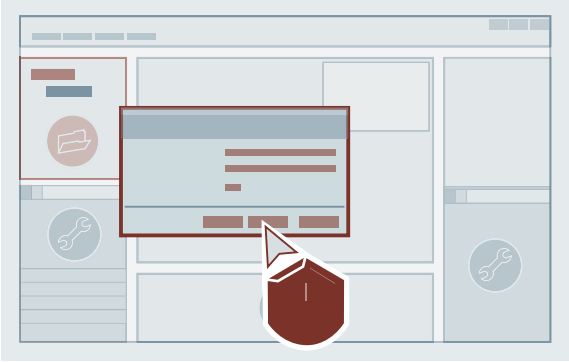
2. Select "CN 4100" > "System" > "Diagnostic" > "Update device state".



3. Enter the device password.
   – Optional addition: Select "Host verifier".



4. Select "Next".



5. Check the device status of the respective CPU.



Result

The device status of the respective CPU is displayed.

Updating device status

Requirement

- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.
- The "Update device state" dialog is open.
  Display device status (Page 111)

Procedure

1. Select "Update device state".



Result

The device status display of the CN 4100 is updated.

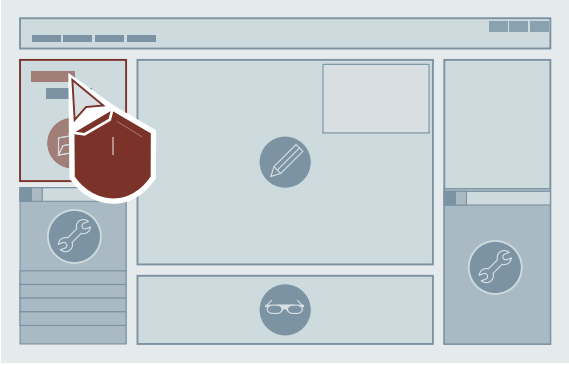
Displaying the status of the time server

Requirement

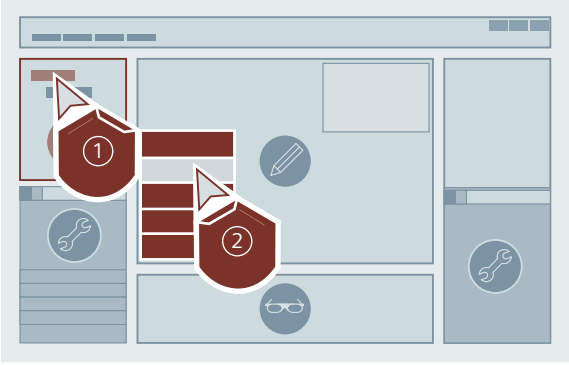
- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.
- The device password is known.
  User and password (Page 48)

Procedure

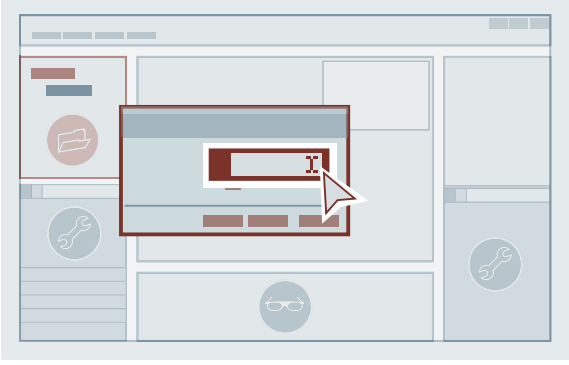
1. Select the hardware.



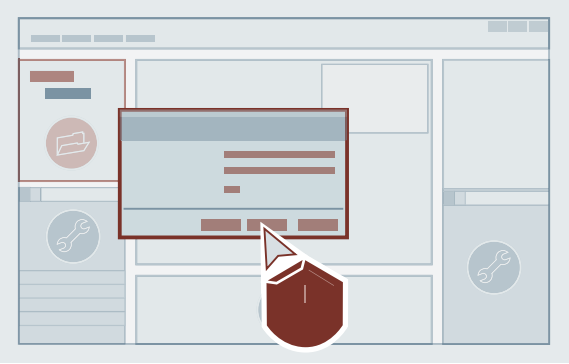
2. Select "CN 4100" > "System" > "Diagnostic" > "Time".



3. Enter the device password.
   – Optional addition: Select "Host verifier".



4. Select "Next".



5. Check the status of the time server.



Result

The status of the time server is displayed.

Updating the status of the time server

Requirement

- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.
- The "Time Server Status" dialog is open.
  Displaying the status of the time server (Page 113)

Procedure

1. Select "Update time server status".



Result

The displayed information about the time server is updated.

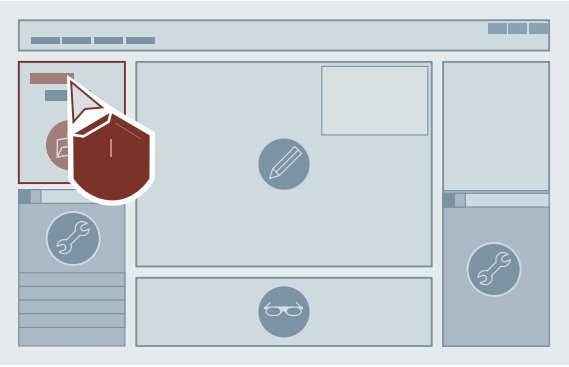
Activating automatic update of the device status

Requirement

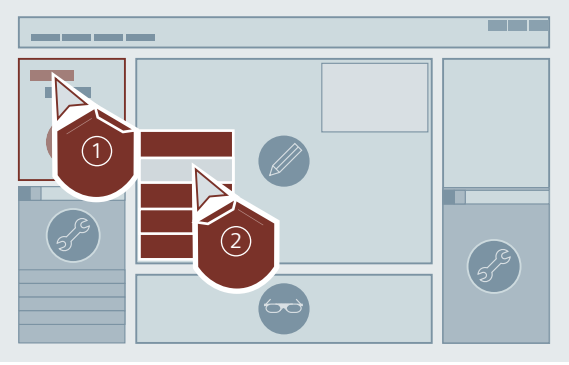
- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.

Procedure

1. Select the hardware.



2. Select "CN 4100" > "System" > "Diagnostic" > "Automatic Status Update".



Result

The automatic update of the device status is activated. The name of the hardware is displayed in color in the navigation area.

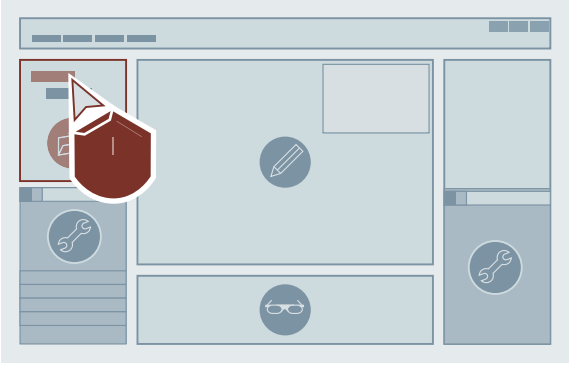
Deactivating automatic update of the device status

Requirement

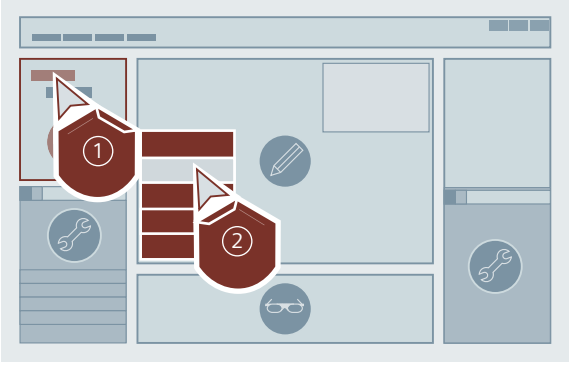
- Navigation area and work area are visible.
- The hardware has been configured and initialized.
- The Engineering System is connected to the hardware.
- Automatic Status Update has been activated for the relevant hardware.

Procedure

1. Select the hardware.



2. Select "CN 4100" > "System" > "Diagnostic" > "Automatic Status Update".



Result

The automatic update of the device status is deactivated.

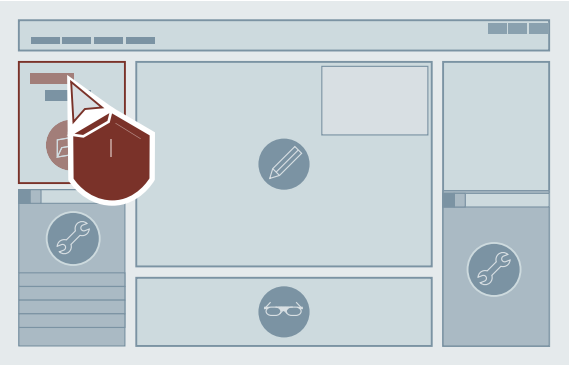
Ethernet | Show IP

Requirement

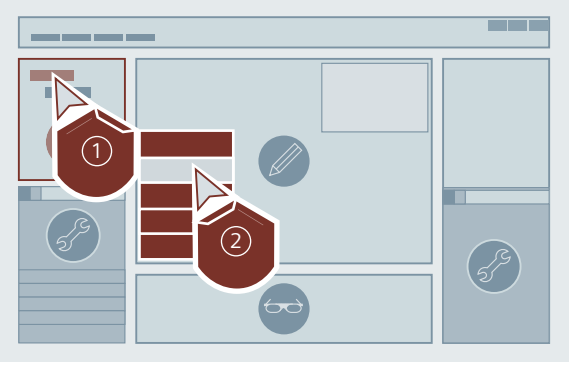
- Navigation area and work area are visible.
- An IP address has been assigned to the hardware.
- The Engineering System is connected to the hardware.
- The device password is known.
  User and password (Page 48)

Procedure

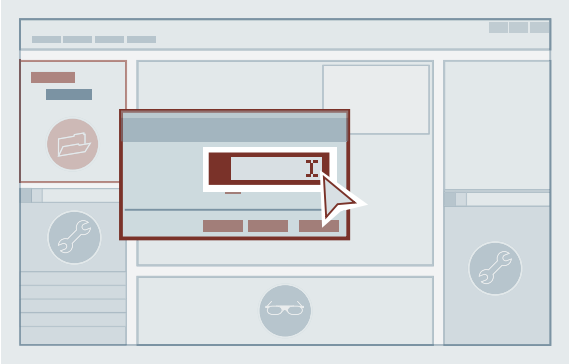
1. Select the hardware.



2. Select "CN 4100" > "System" > "Diagnostic" > "Ethernet Trace".



3. Enter the device password.
   – Optional addition: Select "Host verifier".



4. Confirm.



Result

The dialog for error logging of the Ethernet interfaces is open. The IP address of the CPU is shown.

Ethernet | Start logging

Requirement

- Navigation area and work area are visible.
- An IP address has been assigned to the hardware.
- The connections of the CPU(s) are configured.
- The Engineering System is connected to the hardware.
- The "Ethernet Trace" dialog is open.
  Ethernet | Show IP (Page 118)

Procedure

1. Select the CPU.



2. Optional: Select a different storage location for logging the connections. Confirm.

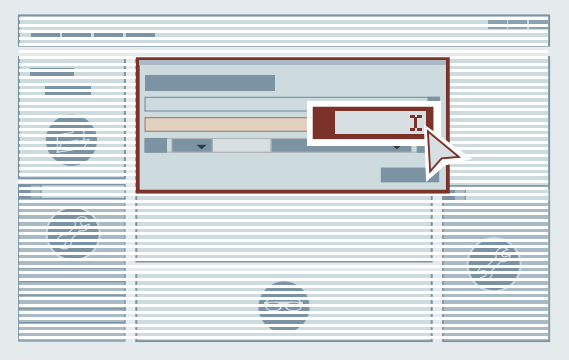


3. Optional: Enter a filter method to be applied during logging.

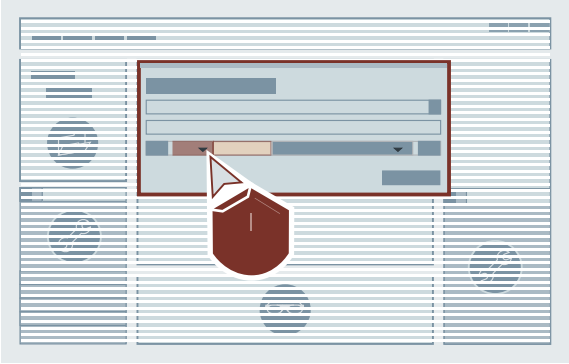
Note

Notation

Enter the name of the connection, for example. The procedure and notation are similar to that in the command line editor of the operating system.



4. Enter a name for the log file and select the connection to be monitored.



5. Select "Start".



Result

Logging for the selected connection has started.

Ethernet | Stop logging

Requirement

- Navigation area and work area are visible.
- An IP address has been assigned to the hardware.
- The connections of the CPU(s) are configured.
- The Engineering System is connected to the hardware.
- Logging for at least one Ethernet connection has started.
- The "Ethernet Trace" dialog is open.
  Ethernet | Show IP (Page 118)

Procedure

1. Select "Stop".



2. Optional: Select one of the created log files.



3. Optional: Select "Open".



Result

Logging for the selected connection has stopped. Optional: A created log file is displayed after the selection.

8.4.3.14 Step-by-step instructions (OPC UA certificate)

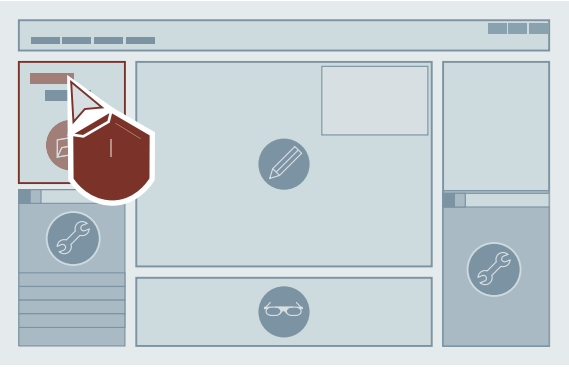
OPC UA | Opening an overview of the certificates

Requirement

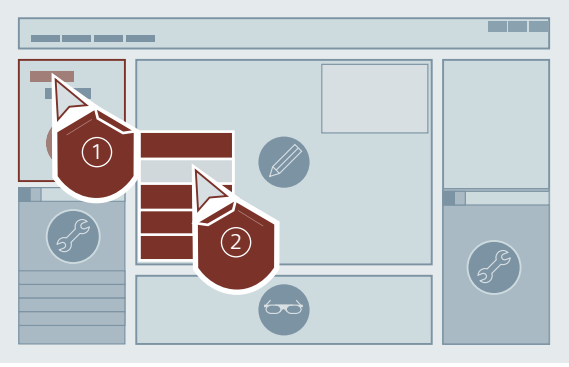
- The hardware has been created and configured.
- The configuration data are compiled and loaded.
- For OPC UA:
  - The protocol is created.
  - OPC UA blocks are created, configured and interconnected.
  - An encryption method has been set for the OPC UA blocks.
  - For OPC UA clients: A connection attempt to the OPC UA server has been made.
  - For OPC UA servers: A connection attempt to an OPC UA client has been made.
- The device password is known.
  User and password (Page 48)

Procedure

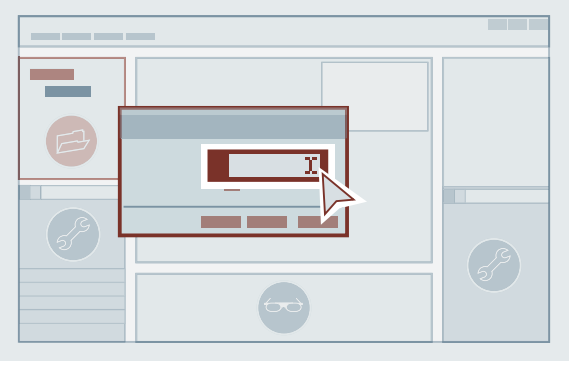
1. Select the hardware in the navigation area.



2. For OPC UA clients, select: "CN 4100" >"OPC UA certificate" > "OPC UA client certificate".
   – Alternatively: For OPC UA servers, select: "CN 4100" >"OPC UA certificate" > "OPC UA server certificate".



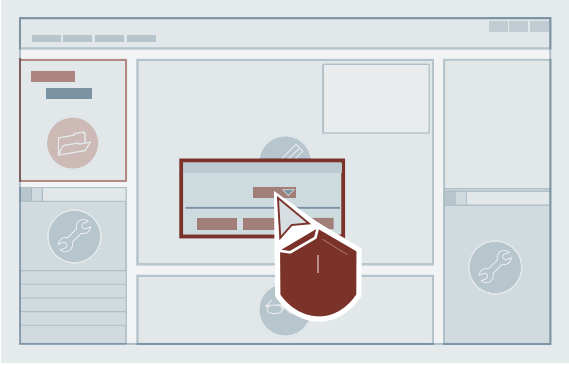
3. Enter the device password.
   – Optional addition: Select "Host verifier".



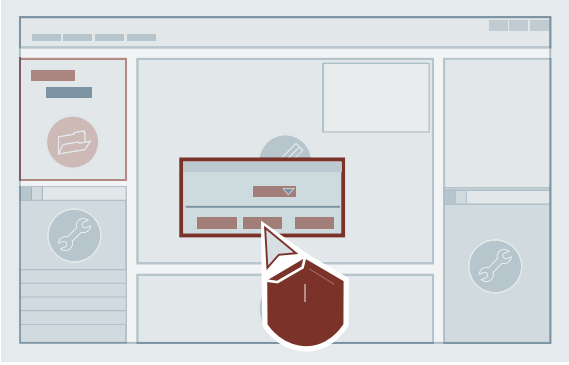
4. Select "Next".



5. Select the setting for the desired CPU.



6. Select "Finish".



Result

The overview for the OPC UA certificates is displayed.

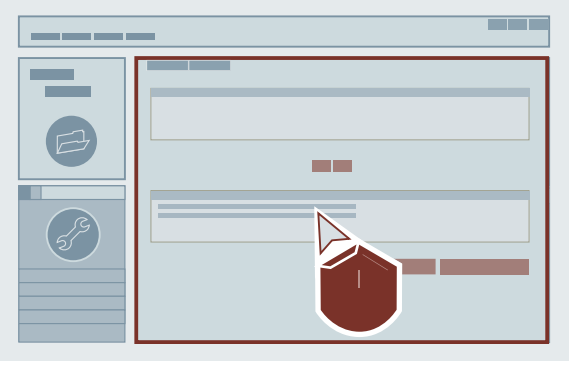
Integrating an OPC UA certificate

Requirement

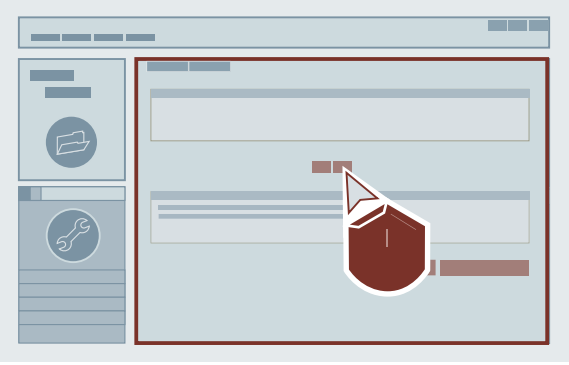
- The hardware has been created and configured.
- The configuration data are compiled and loaded.
- For OPC UA:
  - The protocol is created.
  - OPC UA blocks are created, configured and interconnected.
  - An encryption method has been set for the OPC UA blocks.
  - A connection attempt to the OPC UA server has been made.
- The overview of the OPC UA certificates is open.
  OPC UA | Opening an overview of the certificates (Page 124)
- The device password is known.
  User and password (Page 48)

Procedure

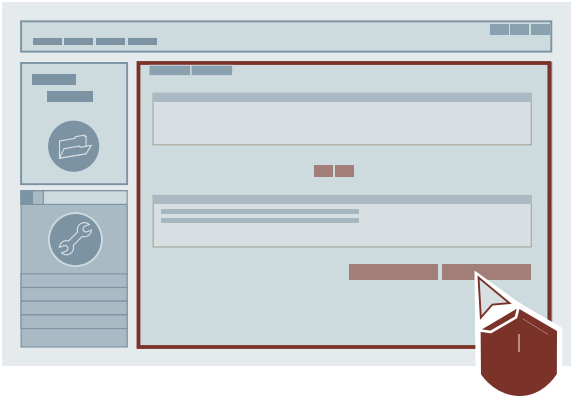
1. Select a certificate under "Rejected".



2. Move the certificate to "Trusted".



3. Select "Apply".



4. Enter the device password.
   – Proceed as described from step 6 in the following section:
     Loading the firmware update (Page 99)

Result

The settings made for the OPC UA certificates have been saved. The selected OPC UA certificate is integrated and can be used for the communication.

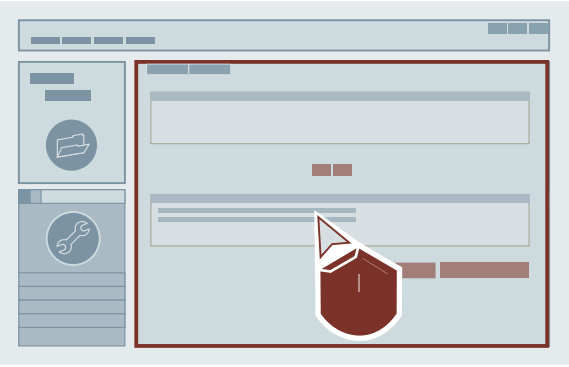
Removing an OPC UA certificate

Requirement

- The hardware has been created and configured.
- The configuration data are compiled and loaded.
- For OPC UA:
  – The protocol is created.
  – OPC UA blocks are created, configured and interconnected.
  – An encryption method has been set for the OPC UA blocks.
  – A connection attempt to the OPC UA server has been made.
- The overview of the OPC UA certificates is open.
  OPC UA | Opening an overview of the certificates (Page 124)
- The device password is known.
  User and password (Page 48)

Procedure

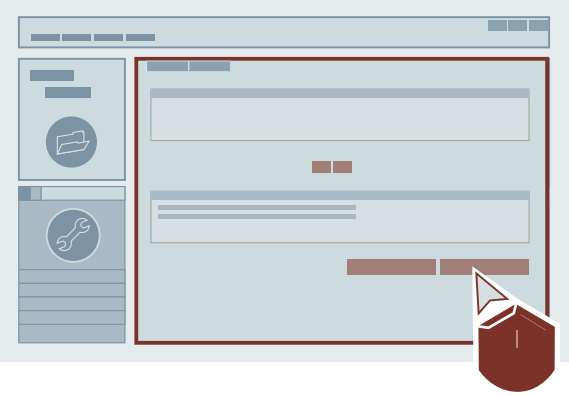
1. Select a certificate from one of the lists (e. g. under "Rejected").



2. Select "Remove".



3. Select "Apply".



4. Enter the device password.
   - Proceed as described from step 6 in the following section: Loading the firmware update (Page 99)

Result

The selected OPC UA certificate is removed from the PC.

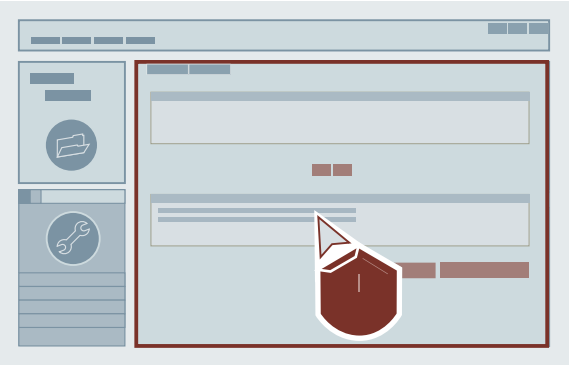
Export the OPC UA certificate

Requirement

- The hardware has been created and configured.
- The configuration data has been translated and loaded.
- For OPC UA:
  - The protocol is created.
  - OPC UA blocks are created, configured and interconnected.
  - An encryption method has been set for the OPC UA blocks.
  - A connection attempt to the OPC UA server has been made.
- The overview of the OPC UA certificates is open.
  OPC UA | Open overview of the certificates (Page 124)

Procedure

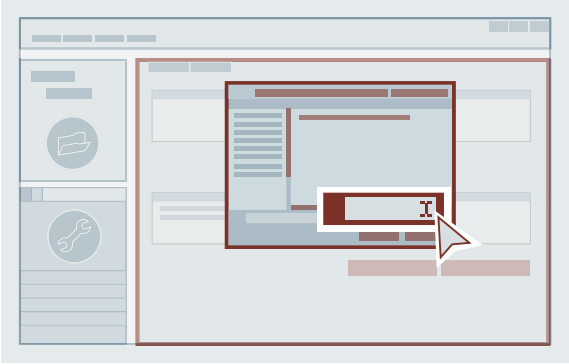
1. Select a certificate from one of the lists (e.g. under "Own").



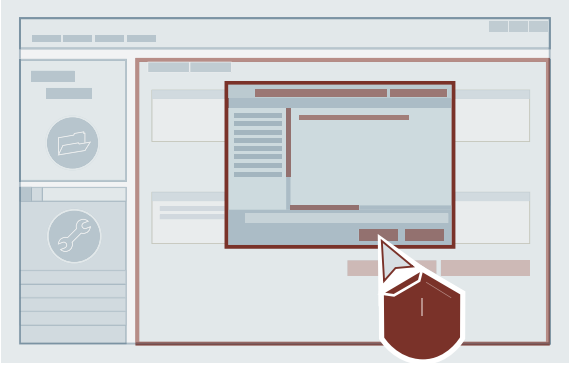
2. Select "Export".



3. Select a storage location and enter a name for the file.



4. Confirm.



Result

The selected OPC UA certificate is exported to the PC.

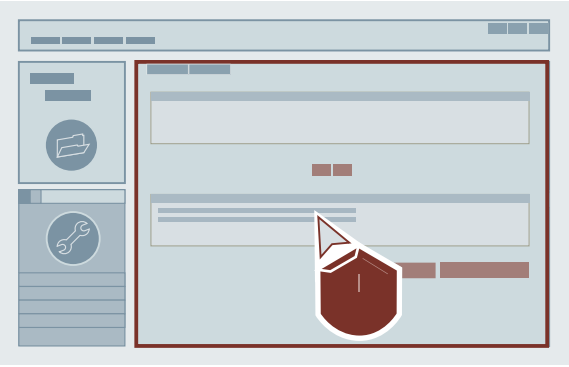
Showing the OPC UA certificate

Requirement

- The hardware has been created and configured.
- The configuration data has been translated and loaded.
- For OPC UA:
  - The protocol is created.
  - OPC UA blocks are created, parameterized and interconnected.
  - An encryption method has been set for the OPC UA blocks.
  - A connection attempt to the OPC UA server has been made.
- The overview of the OPC UA certificates is open.
  OPC UA | Open overview of the certificates (Page 124)

Procedure

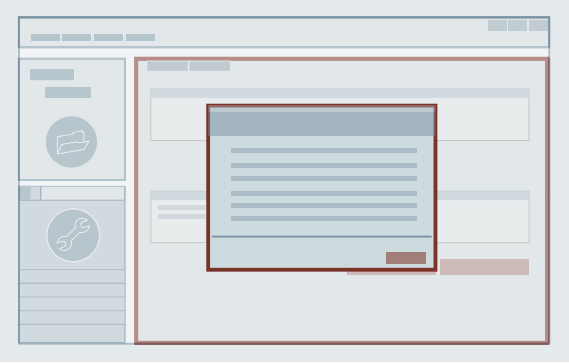
1. Select a certificate from one of the lists (e. g. under "Rejected").



2. Select "Show details".



3. Take a look at the certificate.



Result

The details of the selected OPC UA certificate can be seen.

8.4.4 CFC ("Chart")

8.4.4.1 CFC ("Diagram")

Definition

A CFC ("Diagram") is a graphical display of the program environment. Der CFC can be created as an object in the navigation area of the CNET tool.

A CN 4100 CPU can process a program of one or more CFCs consecutively.

Structure

A CFC is used as a container in which program blocks are created and can be interconnected.

Program blocks (Page 179)

CFC variables can also be created in the CFC. With these the CFCs can be interconnected.

CFC variables (Page 185)

Size

The size of the CFC can be adapted:

Size of the CFC	Height [px]	Width [px]
Minimum	2500	4000
Maximum	10000	6000

8.4.4.2 Step-by-step instructions

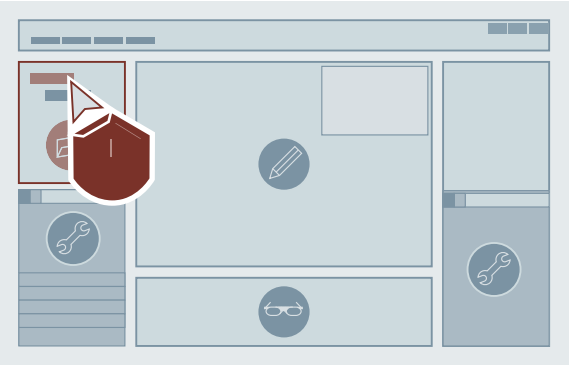
Creating the CFC

Requirement

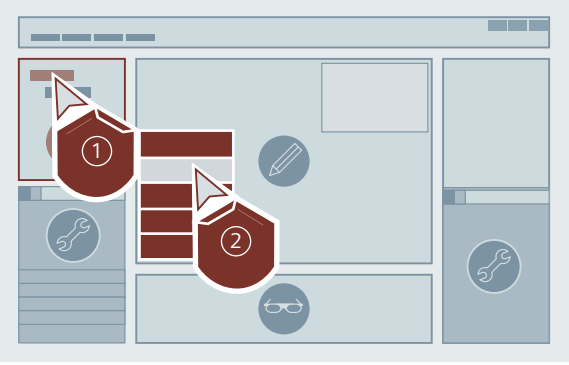
- The navigation area is visible.

Procedure

1. Select the desired hardware in the navigation area.



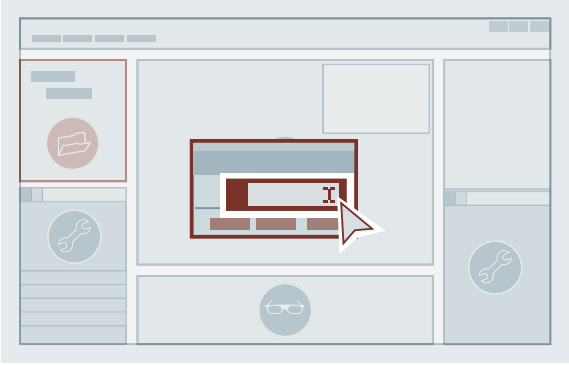
2. Select "Diagram" > "Add".



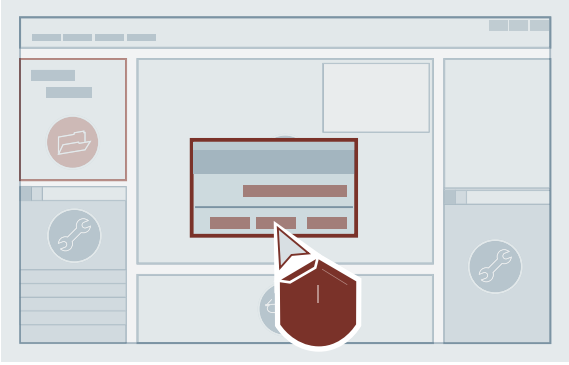
3. Enter a name and adapt the size of the CFC.

   Note

   In CNET CFCs must differ by name from each other.



4. Confirm your selection.



Result

The CFC is created.

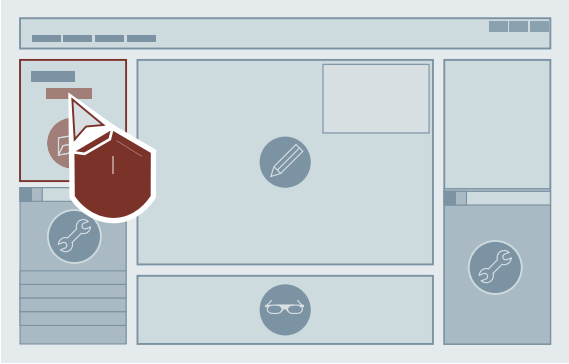
Opening a CFC

Requirement

- The navigation area is visible.
- The work area is visible.

Procedure

Select the desired CFC in the navigation area.



Result

The CFC is opened.

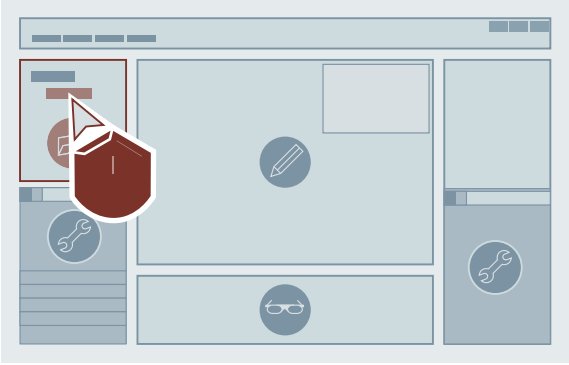
Renaming/editing a CFC

Requirement

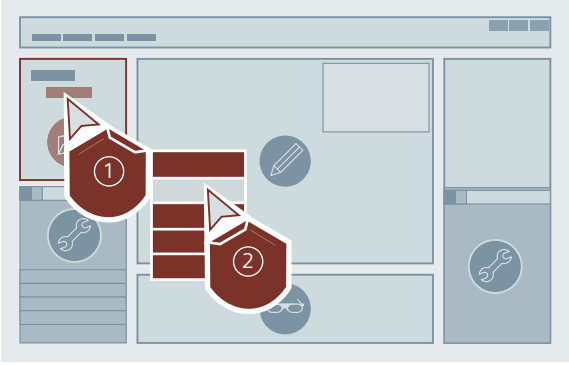
- The navigation area is visible.

Procedure

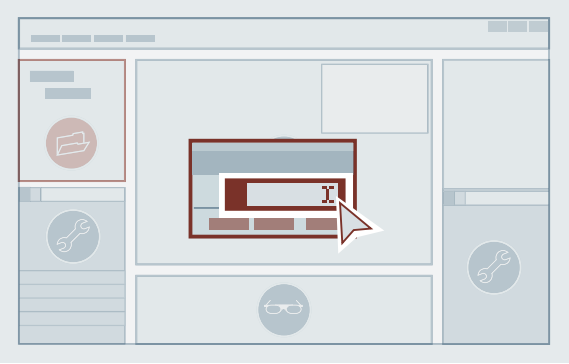
1. Select the desired CFC in the navigation area.



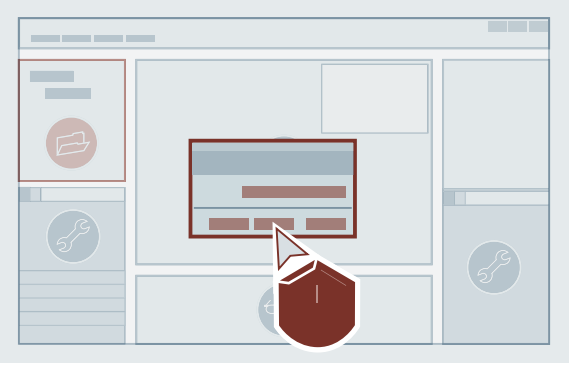
2. Select "Edit".



3. Enter a new name or change the size of the CFC.



4. Confirm your selection.



Result

The properties of the CFC have been changed.

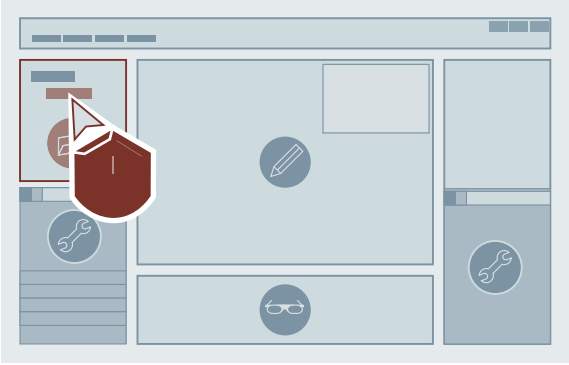
Removing a CFC

Requirement

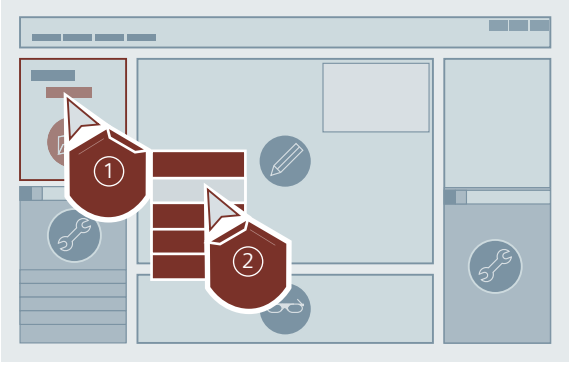
- The navigation area is visible.

Procedure

1. Select the desired CFC in the navigation area.



2. Select "Remove".



3. Confirm.



Result

The CFC has been removed.

Exporting a CFC

Requirement

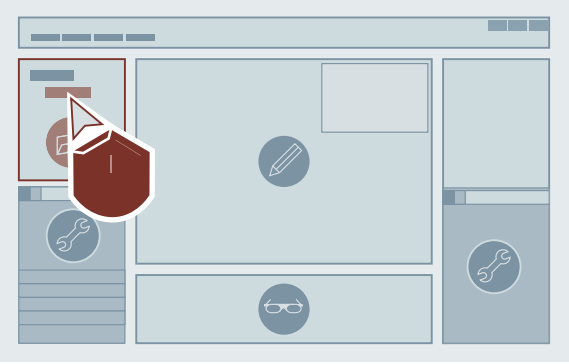
- The navigation area is visible.
- You have created a CFC in CNET.

Procedure

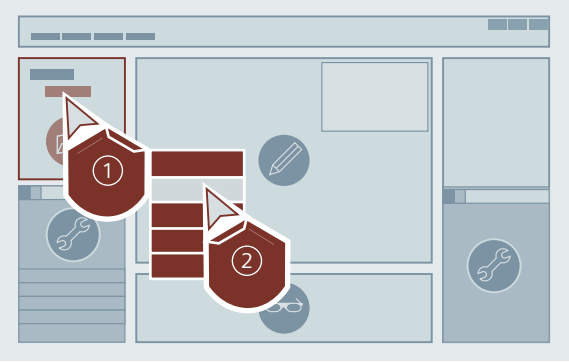
1. Select the desired CFC in the navigation area.

   Note

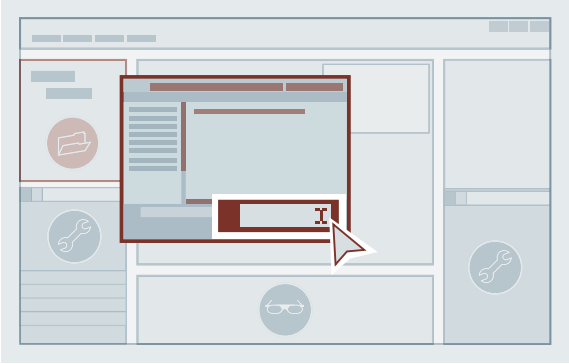
   Alternatively, you can select the hardware or a folder. Then, all lower-level CFCs are included in the export.



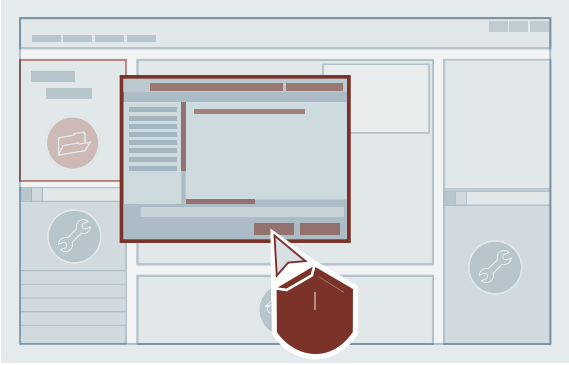
2. Select "Diagram" > "Export diagrams".



3. Select a storage location and name.



4. Confirm your selection.



Result

The CFC is exported.

Importing a CFC

Requirement

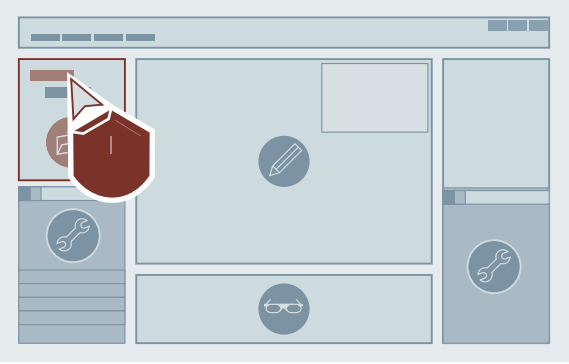
- The navigation area is visible.
- An export or a generation of a CFC is available.

Procedure

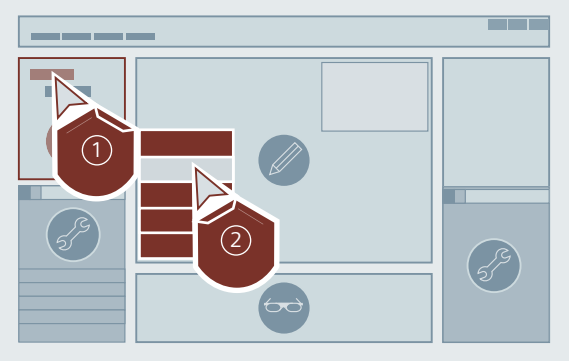
1. Select the desired hardware or a folder underneath it in the navigation area.

   Note

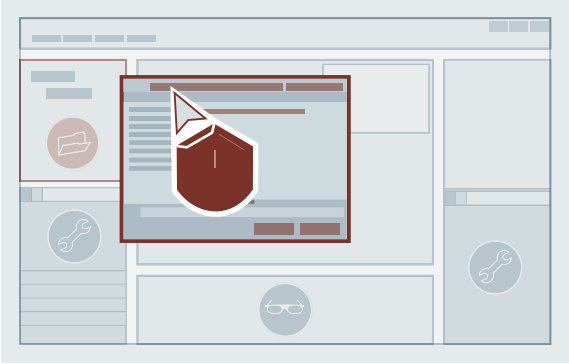
   If you instead select a previously created CFC under the desired hardware, the selection also works.



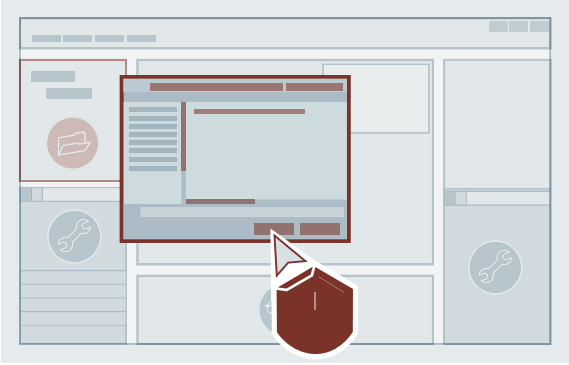
2. Select "Diagram" > "Import diagrams".



3. At the storage location of the CFC select the corresponding CSV file.



4. Confirm your selection.



Result

The CFC is imported.

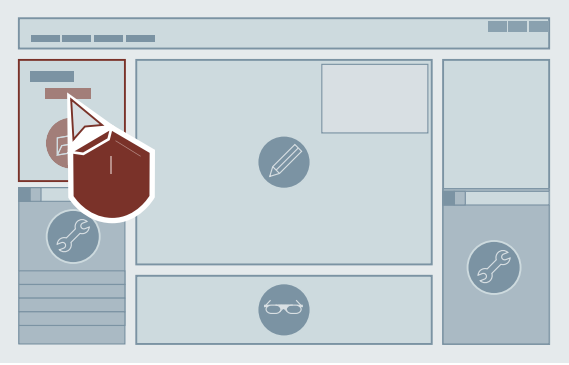
Exporting parameters

Requirement

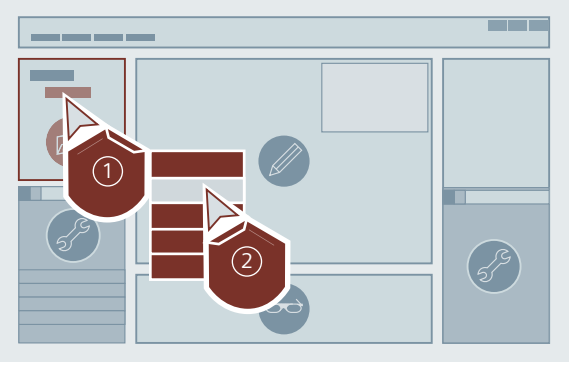
- The navigation area is visible.
- You have created program blocks in CNET.

Procedure

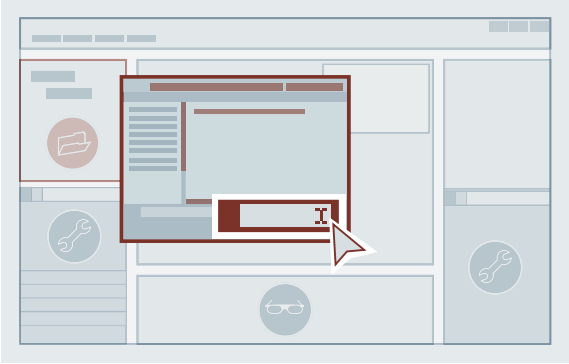
1. Select the desired CFC or the hardware in the navigation area.



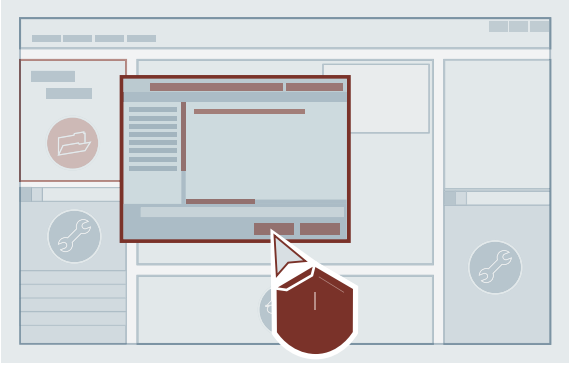
2. Select "Instance" > "Export parameters".



3. Select a storage location and name.



4. Confirm your selection.



Result

All block parameters of the selection are exported.

Importing parameters

Requirement

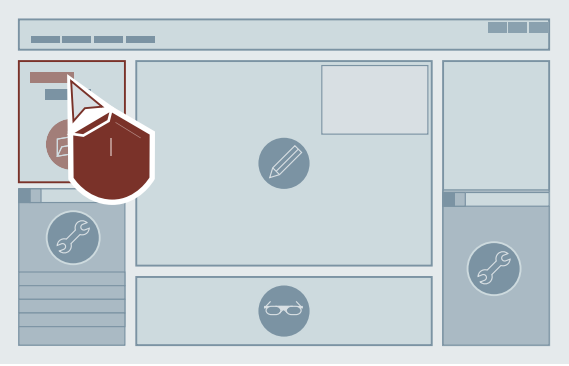
- The navigation area is visible.
- You have an export file with parameters (CSV).

Procedure

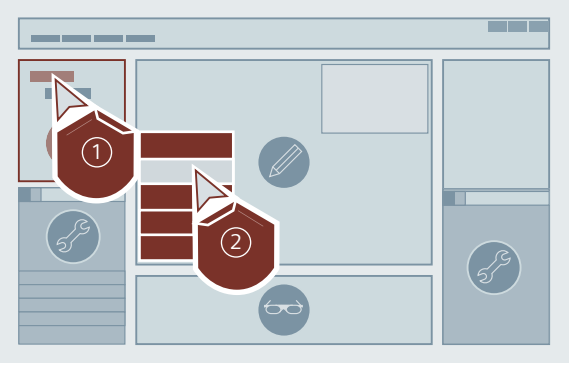
1. Select the desired hardware in the navigation area.

   Note

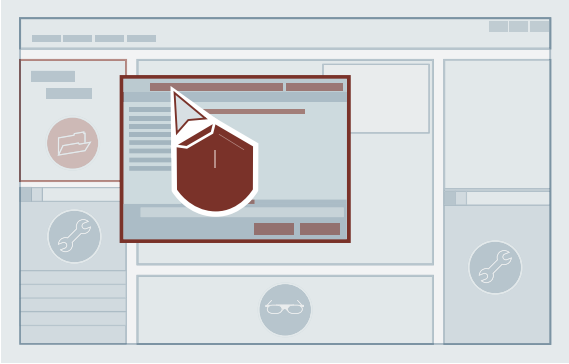
   If instead you have already selected a created CFC under the desired hardware, the selection functions in the same manner.



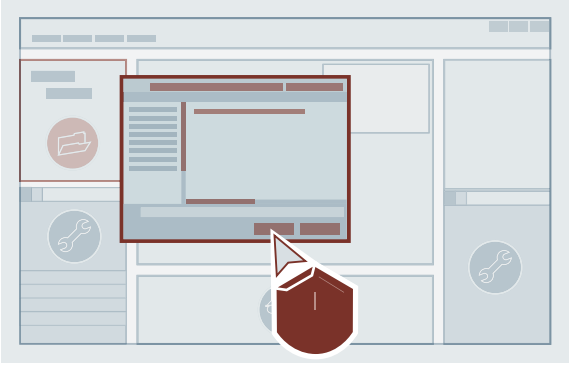
2. Select "Instance" > "Import parameters".



3. Select the relevant CSV file via its storage location.



4. Confirm your selection.



Result

The new data of the block parameters is imported and applied to the relevant block.

8.4.5 Folder

8.4.5.1 Folder

Definition

The "Folder" object is an object for grouping other objects. A Folder can be created in the navigation area of the CNET tool.

8.4.5.2 Step-by-step instructions

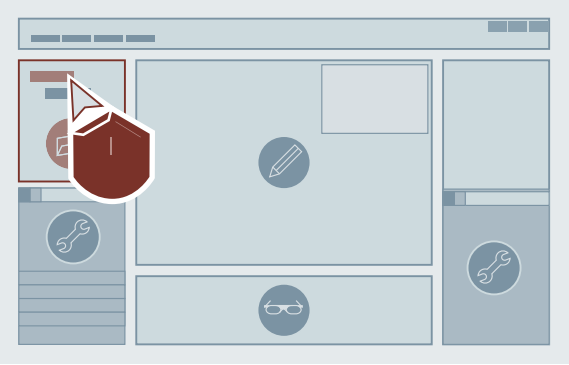
Creating a folder

Requirement

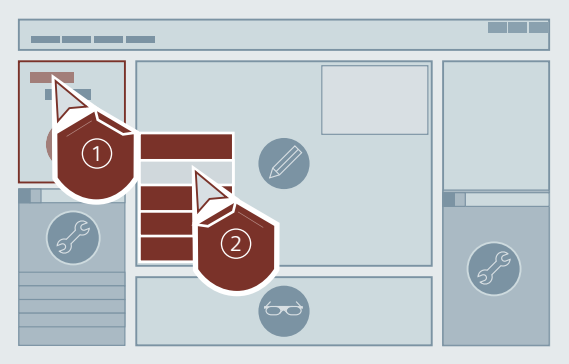
- The navigation area is visible.

Procedure

1. Select the desired hardware in the navigation area.



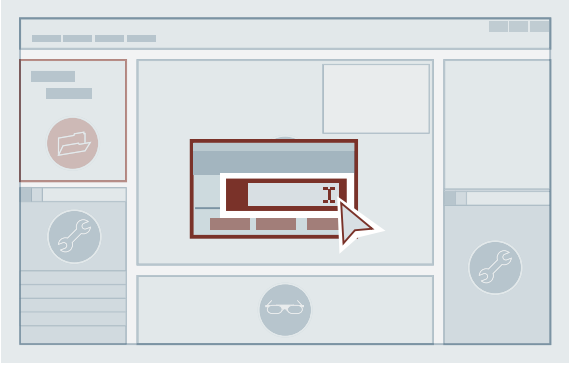
2. Select "Folder" > "Add".



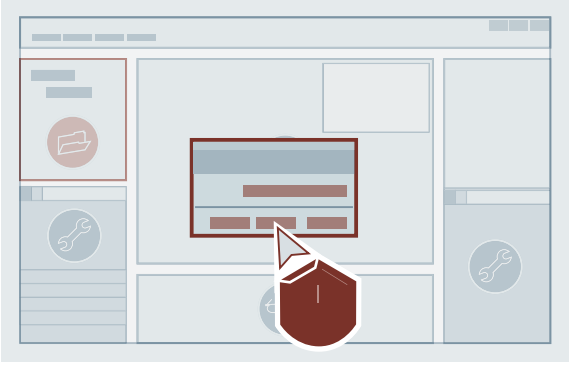
3. Enter a name.

Note

In CNET, folders do not have to be differentiated by name from each other.



4. Confirm your selection.



Result

A folder is created.

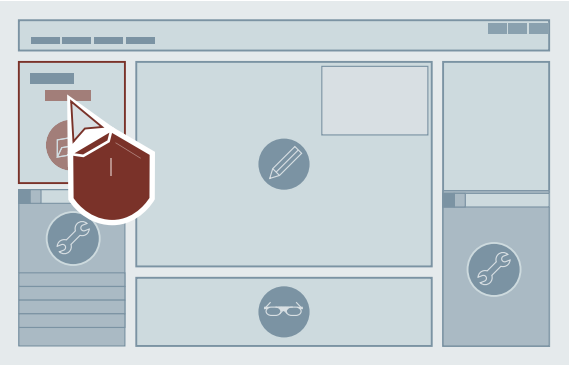
Renaming folders

Requirement

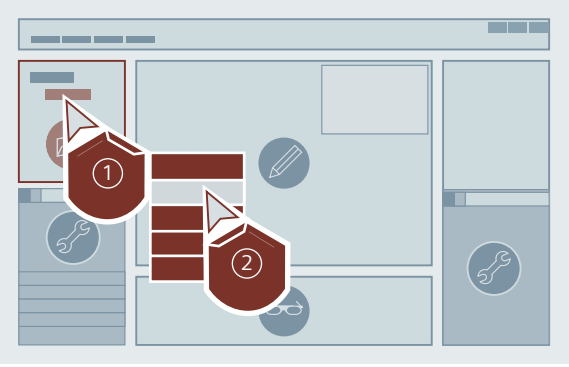
- The navigation area is visible.

Procedure

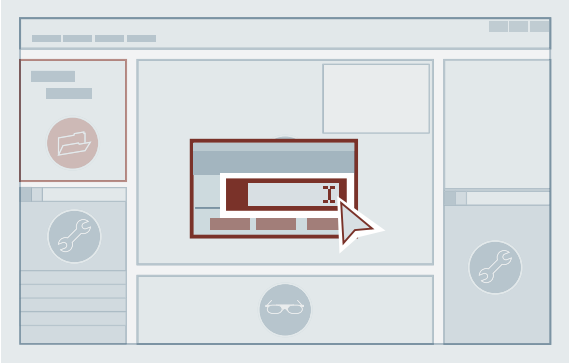
1. Select the desired folder in the navigation area.



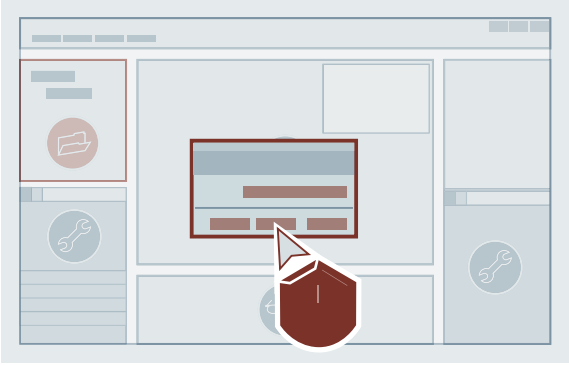
2. Select "Edit".



3. Enter a new name.



4. Confirm your selection.



Result

The folder is renamed.

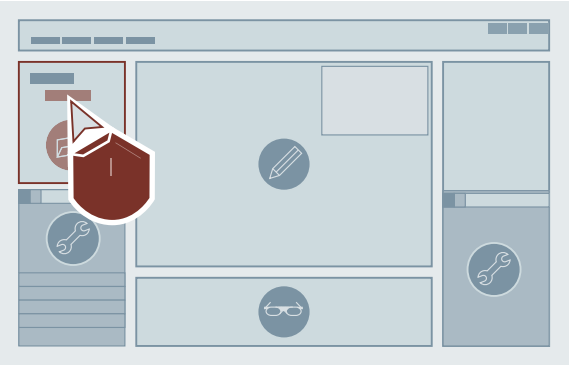
Removing a folder

Requirement

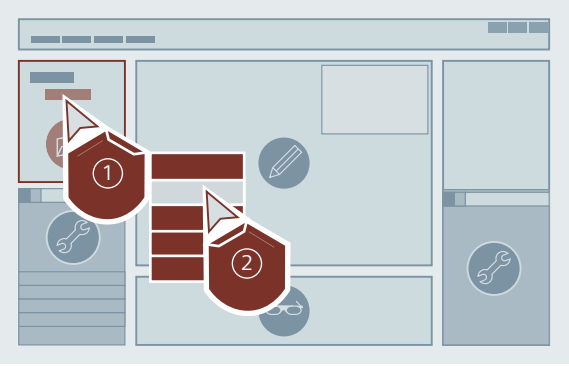
- The navigation area is visible.

Procedure

1. Select the desired folder in the navigation area.



2. Select "Remove".



3. Confirm.



Result

The folder has been removed.

8.5 Library area

8.5.1 Library area

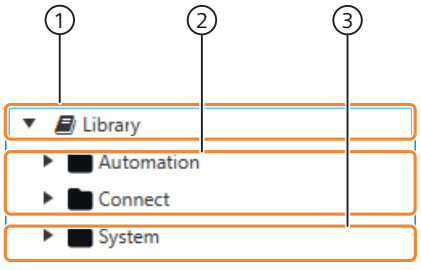
Definition

The library area is an area of the CNET tool where the user can view the block libraries and take out program blocks. The library area can also be understood as an annex to the navigation area.

Navigation area (Page 64)

Structure

The library area is structured as follows:



①	Library	Main nodes of the block libraries.
②	Automation Connect	Topic groups of the block libraries. • Automation: – Analog calculating functions: Analog time-dependent calculating functions (Page 280) Analog time-independent calculating functions (Page 245) – Binary calculating functions: Binary time-dependent calculating functions (Page 308) Binary time-independent calculating functions (Page 293) – Signal monitoring: Signal monitoring analog (Page 318) Signal monitoring binary (Page 347) – System functions: System functions (Page 355) • Connect: – Converter: Signal converter (Page 362) – Devices: BlackBox (Page 368) (S7-Put/Get) TCON (Page 385) – Protocols: CN_OPCUA_Client (Page 410) CN_OPCUA_Server (Page 432) Modbus (Page 444)
③	System	"System" library. • System (Page 482)

8.5.2 Step-by-step instructions

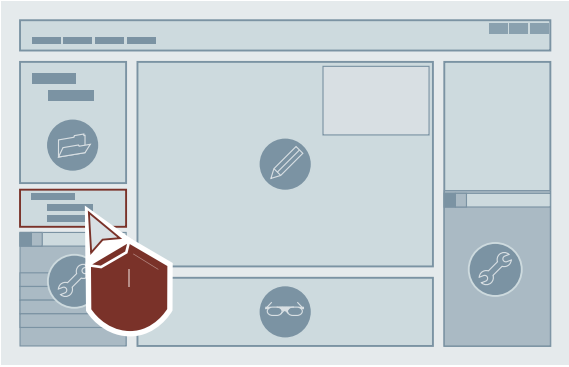
8.5.2.1 Using object from the library area

Requirement

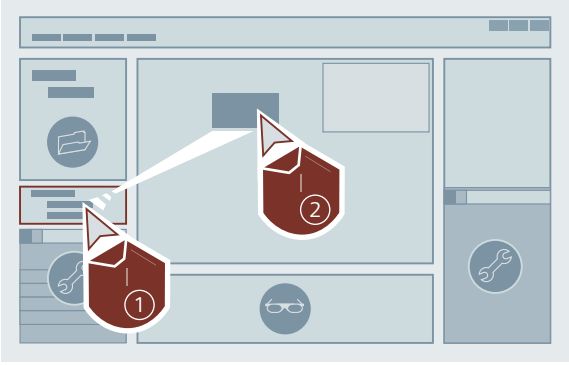
- The library area is visible.
- The work area is visible.
- You have created hardware and a CFC in CNET.

Procedure

1. In the library area, open one of the topic groups and select a program block.



2. Drag the program block into the work area.



Result

The selected program block is placed in the work area.

8.6 Configuration area

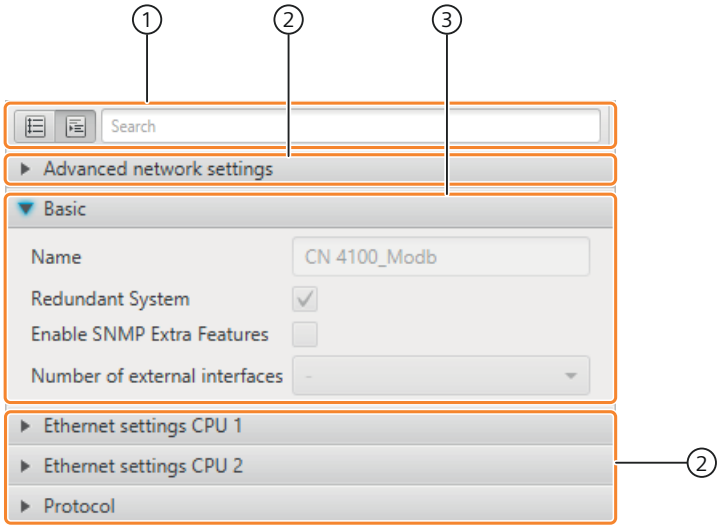
8.6.1 Configuration area

Definition

The configuration area ("Hierarchy") is an area of the CNET tool where the user is shown the properties of the selected hardware.

Design

The configuration area is structured as follows:



①	Filter range	Filter range (Page 221)
②	Categories	Shows the hardware properties divided into categories. The sections can be expanded or closed.
③	Hardware properties	Shows the hardware properties and the selected values. Hardware properties (Page 157)

8.6.2 Operation mode

Definition

- The operation mode is a status of the CN 4100, in which the CN 4100 is prepared for engineering.
- The configuration of the CN 4100 has taken place.

8.6.3 Hardware properties

8.6.3.1 Hardware properties

Definition

The hardware properties describe the respective hardware and are required for their configuration.

Structure

The hardware properties consist of the following categories:

Basic (Page 158)

Ethernet settings (Page 158)

Advanced network settings (Page 160)

Protocol (Page 161)

First/Second external interface (Page 162)

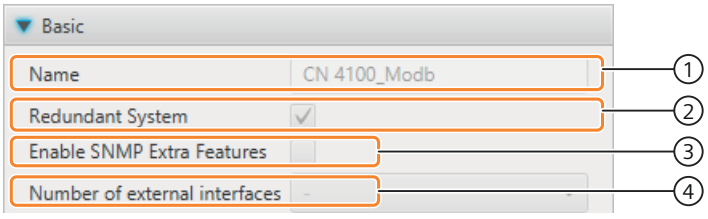
8.6.3.2 Basic

Definition

The "Basic" category shows basic settings of the hardware.

Structure

Basic is structured as follows:



①	Name	Shows the selected designation of the hardware.
②	Redundant System	Shows whether the CPU is created redundant or not.
③	Enable SNMP Extra Features	Shows whether the SNMP protocol is activated. The SNMP protocol is required for server accesses. (E.g. SIMATIC Maintenance Station)
④	Number of external interfaces	Shows the number of additional Ethernet connections.

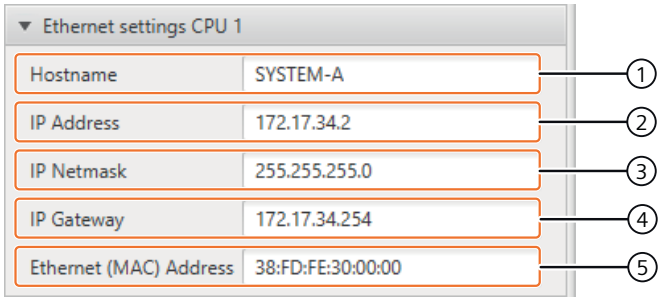
8.6.3.3 Ethernet settings

Definition

The"Ethernet settings CPU 1" and "Ethernet settings CPU 2" categories show the Ethernet settings for the created hardware.

Structure

"Ethernet settings .." is structured as follows:



①	Hostname	Show the name of the CPU.
②	IP Address ...	Shows the used IP address.
③	IP Netmask ...	Shows the used network mask.
④	IP Gateway ...	Shows the used gateway address. (Optional). This setting is necessary when using time synchronization.
⑤	Ethernet (MAC) Address...	Shows the used MAC address.

For redundantly created hardware, the settings for the second CPU are also displayed.

Note

Different subnets for each interface

In the Ethernet settings of the respective interface, a subnet must be used that is different from the other interfaces.

Example:

- The IP address of X1 is "**172.17.34**.2".
- As a separate subnet for the IP address of X2, it would be sufficient to use a value other than "34".

Note

"IP Gateway" redundant CPUs

If a CPU is to be redundant, an entry must be made for both CPUs when using the entry "IP Gateway". This also applies if no redundant plant network is used.

8.6.3.4 Advanced network settings

Definition

The "Advanced network settings" category shows settings for the time synchronization and for the use of DNS servers.

Note

Meaning of the time synchronization

The settings for the time synchronization are more important in combination with other SIMATIC products.

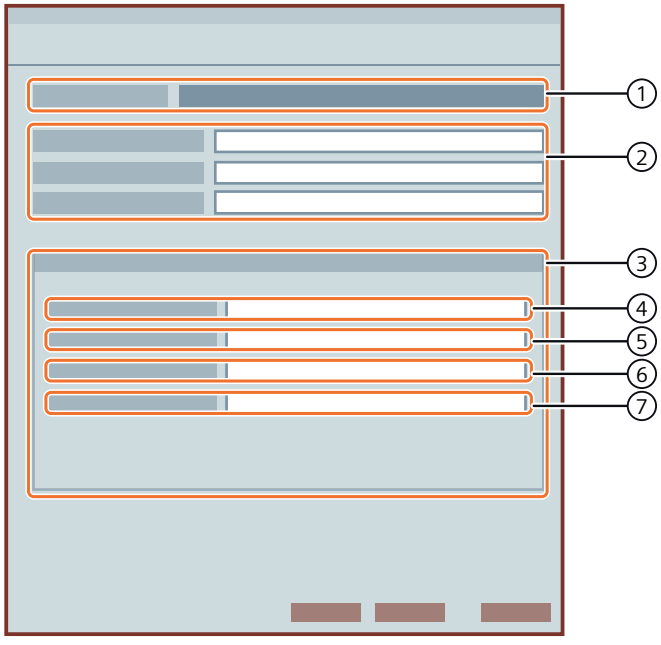
Note

Using DNS servers

Each Ethernet interface of CN 4100 supports DNS. For example, when using OPC UA, you can configure the Ethernet interface used to use DNS. Only use the specification of DNS servers if you want to use them.

Structure

Advanced network settings is structured as follows:



①	NTP or PTP Time Server		Indicates whether the NTP or PTP method is to be used for time acquisition.
②	Is displayed in the configuration area.	First DNS Server IP Address	Shows the IP for the first DNS server.
		Second DNS Server IP Address	Shows the IP for the second DNS server.
		SysLog Server IP Address	Displays the IP address for the server used for system logging.
③	NTP Settings		Area. Is displayed when the "NTP" method is selected.
④	First NTP Server IP Address		Shows the IP for the first NTP server.
⑤	First NTP Server IP Gateway		Shows the gateway for the first NTP server.
⑥	Second NTP Server IP Address		Shows the IP for the second NTP server.
⑦	Second NTP Server IP Gateway		Shows the gateway for the second NTP server.

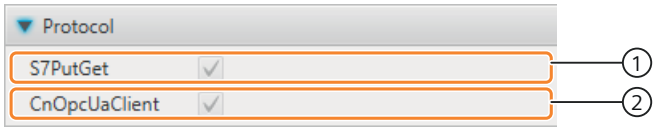
8.6.3.5 Protocol

Definition

The "Protocol" category shows the communication protocol to be used.

Structure

The protocol is structured as follows:



①	Selected PCS 7 protocol: • S7PutGet • TCON • CnOpcUaServer	Shows whether the relevant protocol is activated.
②	Protocol selected for a third-party system: • CnOpcUaClient • MODBUS	

For more information on properties of CN 4100, refer to the following documentation:

- "CN 4100 Communication System" Equipment Manual

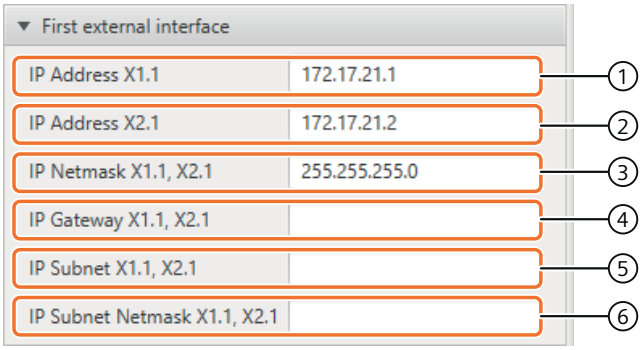
8.6.3.6 First/Second external interface

Definition

The "First external interface" and "Second external interface" categories show the Ethernet settings for the interfaces to external networks.

Structure

"First external interface" is structured as follows:



①	IP Address X1.1	Shows the IP address used for X1.1. Connection to CPU 1.
②	IP Address X2.1	Shows the IP address used for X2.1. Connection to CPU 2.
③	IP Netmask ...	Shows the network mask used for X_.1.
④	IP Gateway ...	Shows the used gateway address. (Optional). This setting is necessary when using time synchronization or a subnet. (See example.)
⑤	IP Subnet ...	Shows the IP address used for the subnet. (Optional, see example.)
⑥	IP Subnet Netmask ...	Shows the network mask used for "IP Subnet ...". (Optional.) Must be specified when using "IP Subnet ...". (See example.)

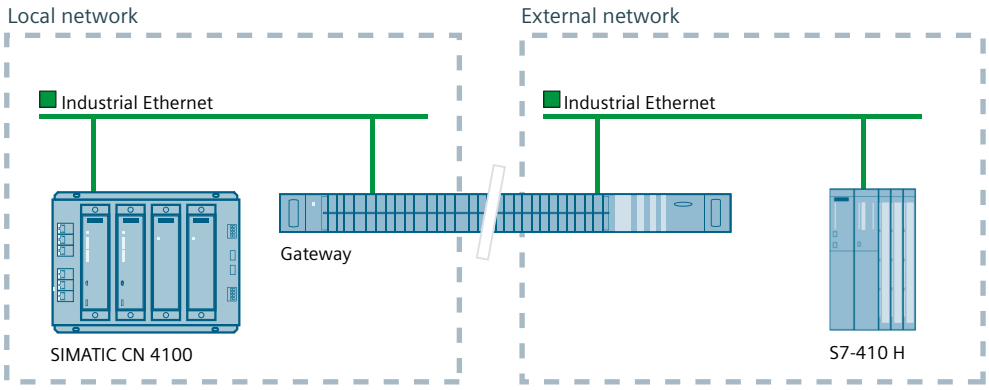
The structure for "Second external interface" is the same as "First external interface" but relates to interfaces X1.2/X2.2.

Settings for the second CPU (interface X2.x) are only displayed if hardware was created redundantly.

Example for the settings of a subnet

Objective:

A device in the local network (IP X1.1) is to be connected to a device in an external network.



Interface of the device to be accessed (in the external network):

- IP = 10.3.2.1

Set CNET (of the local device):

Interface of the local device		Router of the local network	For forwarding to external network	
IP Address X1.1	***IP Netmask X1.1***	***IP Gateway X1.1***	***IP Subnet X1.1***	***IP Subnet Netmask***
192.168.1.7	255.255.255.0	192.168.1.1	10.0.0.0	255.0.0.0

In the input mask of the X1 interface there is also an IP Gateway parameter. All telegrams that do not belong to the local network of an interface and do not belong to the subnet X1.1 or X1.2 are forwarded to this address.

8.6.4 Step-by-step instructions

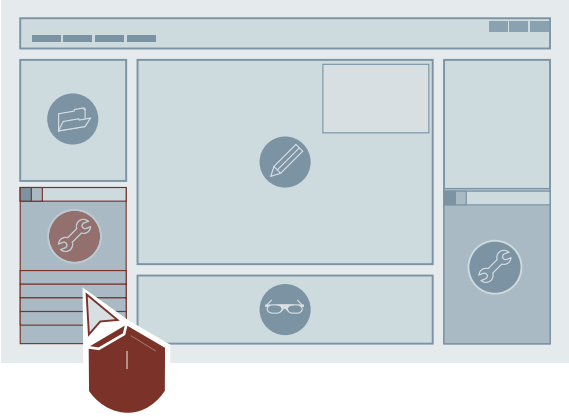
8.6.4.1 Expanding categories

Requirement

- The configuration area ("Hierarchy") is shown.
- The configuration area is displayed sorted by category.

Procedure

In the configuration area, select the desired category.



Result

The content of the selected category are displayed in the configuration area.

8.7 Work area

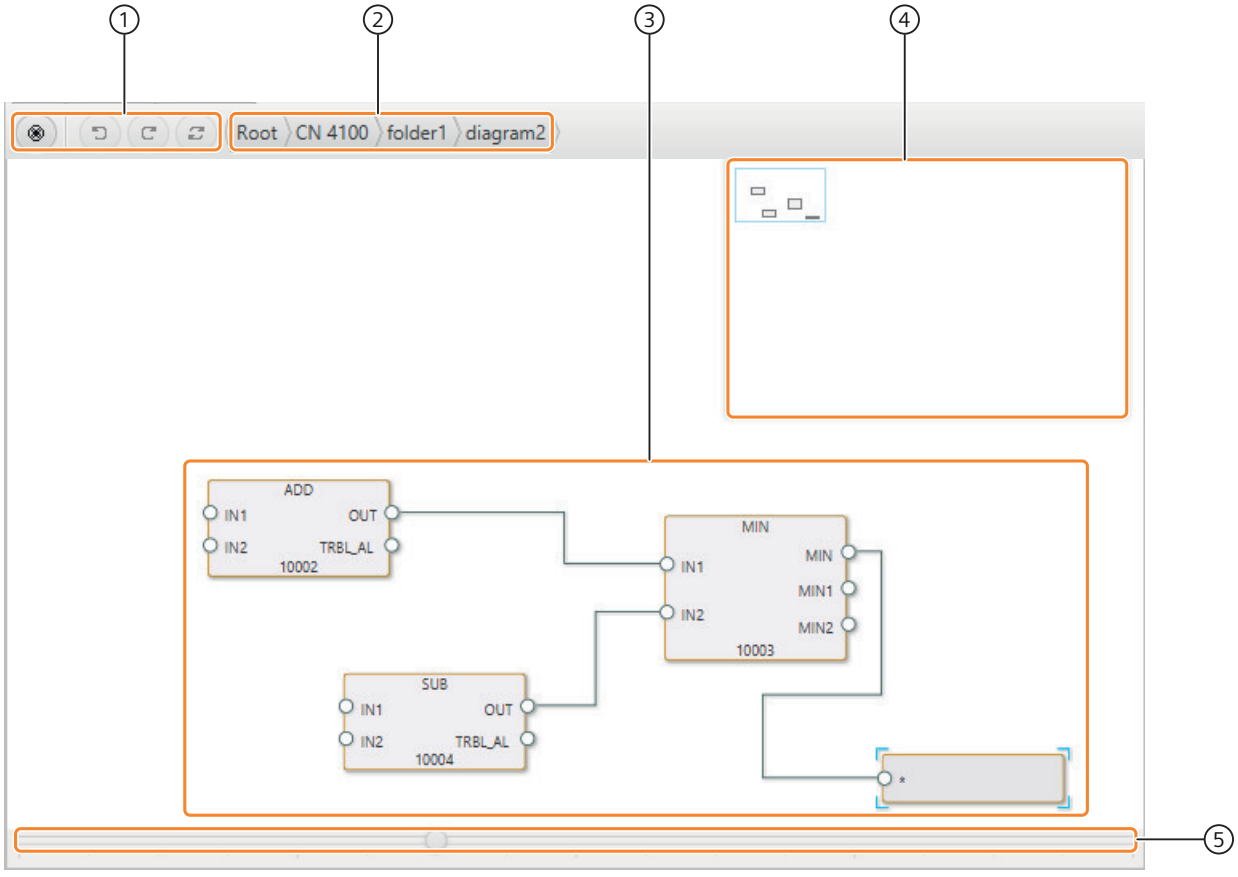
8.7.1 Work area

Definition

The work area is an area of the CNET tool where a user can have a selected CFC displayed and fill it with content.

Structure

The work area is structured as follows:



①	Influencing the work area and the editing history	Buttons. • "Point": Serves to limit the area of the work space/CFC to the area in use.  • Editing history (Page 166)
②	Name of the CFC	CFC (Page 134)
③	Interconnected objects	Program blocks (Page 179) CFC variables (Page 185)
④	Minimap	Shows the minimized version of the CFC in the work area. Minimap (Page 189) ("Diagram map")
⑤	Zoom function	Slider to reduce or enlarge the display in the work area.

8.7.2 Editing history

Definition

The editing history protocol of the CNET tool creates a list of work steps performed by the user (from the work area). CNET remembers the last 10 work steps.

Structure

CNET provides the following buttons for the editing history:

- Undo: Undoes the last work step.

- Do: Performs the undone work step again.

- Redo: Repeats the last work step. (Can be used for a copy action, for example.)

8.7.3 Step-by-step instructions

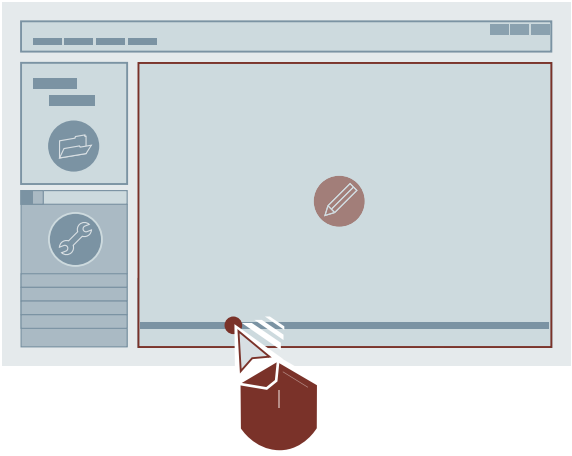
8.7.3.1 Enlarging / reducing a view

Requirement

- Work area can be seen.
- A CFC is open.

Procedure

Drag the circle until the desired enlargement of the view.



Result

You have changed the size displayed of the objects in the CFC.

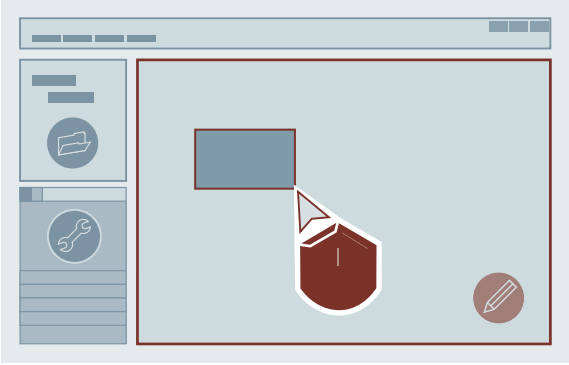
8.7.3.2 Zooming in and zoom out of an object

Requirement

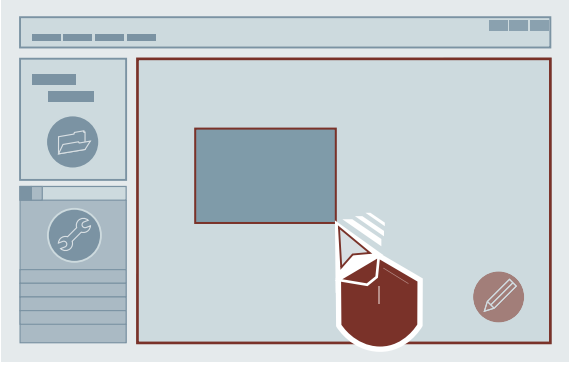
- The work area is visible.
- The desired CFC is open.
- At least one object (block or CFC variable) has been created.

Procedure

1. In the work area, select the corner of an object.



2. Drag the corner of the object to the desired position.



Result

The size of the selected object has been changed.

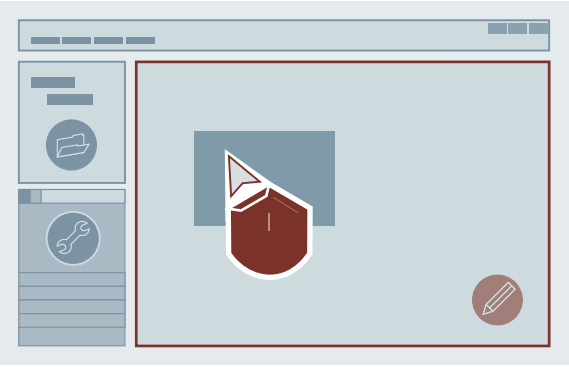
8.7.3.3 Removing an object

Requirement

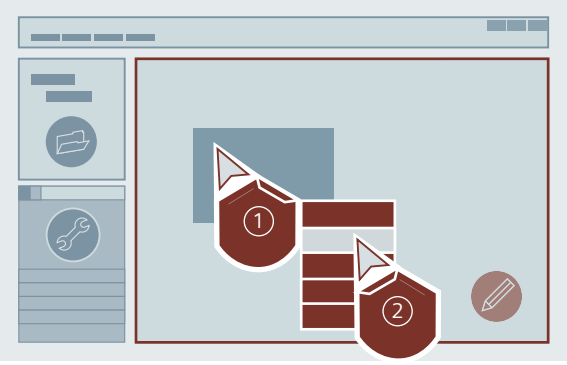
- The work area is visible.
- The desired CFC is open.
- At least one object (block or CFC variable) has been created.

Procedure

1. In the work area, select an object.



2. Select "Remove selection".



Result

The selected object has been removed.

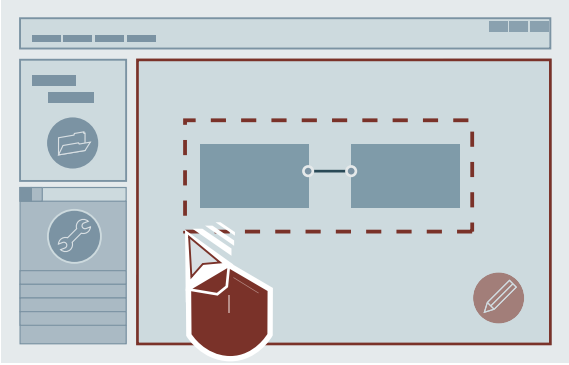
8.7.3.4 Copying a selection

Requirement

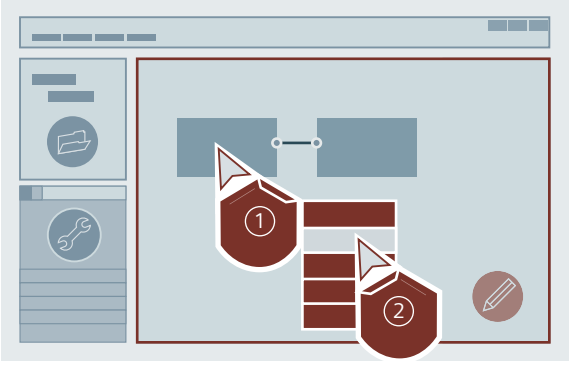
- The work area is visible.
- The desired CFC is open.
- At least one object (block) has been created.

Procedure

1. In the work area, select one or multiple objects.



2. Select "Copy selection".



Result

The selection is copied, but the copy is not yet pasted.

Note

Interconnections between objects within the selection are considered during the copy process. CFC tags are not copied.

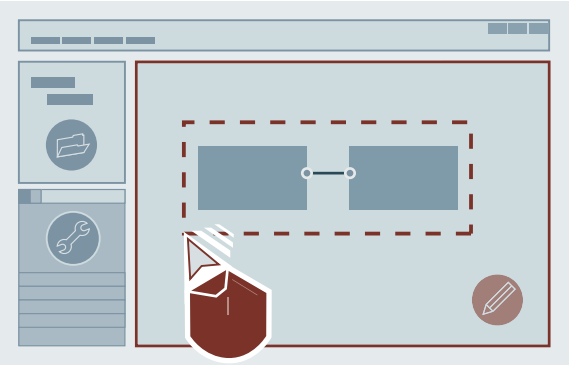
8.7.3.5 Cutting a selection

Requirement

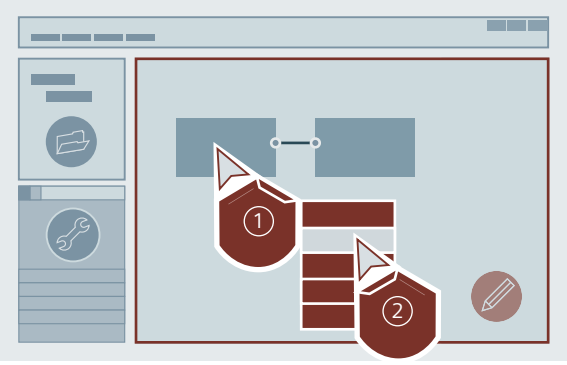
- The work area is visible.
- The desired CFC is open.
- At least one object (block) has been created.

Procedure

1. In the work area, select one or multiple objects.



2. Select "Cut selection".



Result

The selection is cut.

Note

Interconnections between objects within the selection are considered during the copy process. CFC tags are not copied.

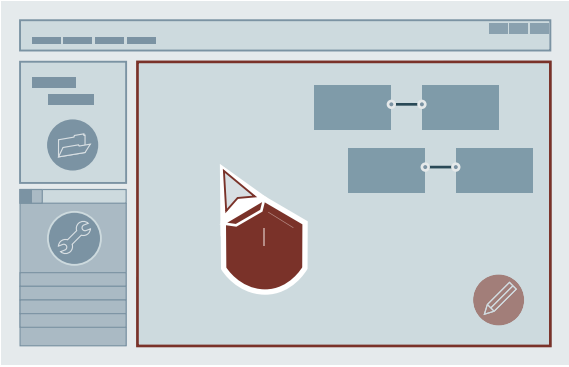
8.7.3.6 Pasting the copied selection

Requirement

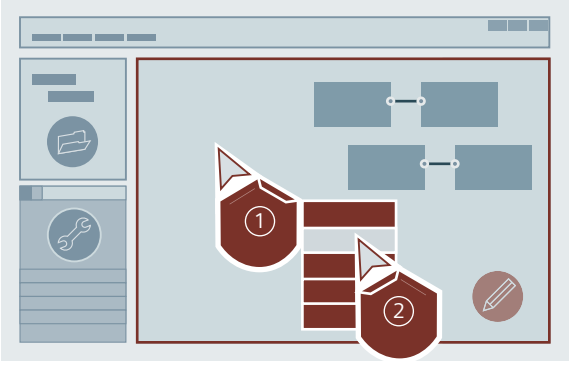
- The work area is visible.
- The desired CFC is open.
- A selection of one or more objects (block or CFC variable, with or without interconnections) was copied.

Procedure

1. Select a spot in the work area.



2. Select "Paste selection".



Result

The previously-copied selection is inserted.

Note

The procedure can be repeated as necessary.

8.7.3.7 Selecting all objects

Requirement

- The work area is visible.
- The desired CFC is open.
- At least one object (block or CFC variable) has been created.

Procedure

1. Select "Select all".



Result

All objects in the work area are selected.

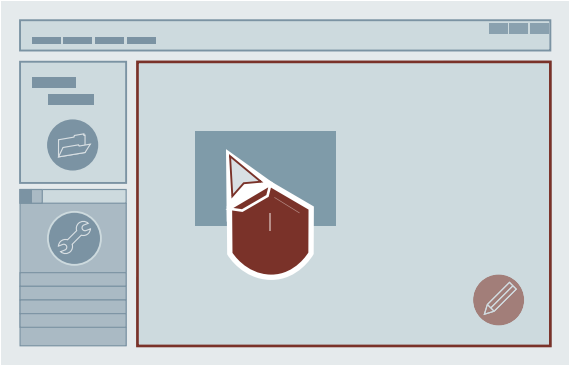
8.7.3.8 Moving an object

Requirement

- The work area is visible.
- The desired CFC is open.
- At least one object (block or CFC variable) has been created.

Procedure

1. In the work area, select an object.



2. Move the object to the new position.



Result

The object is located at a new position.

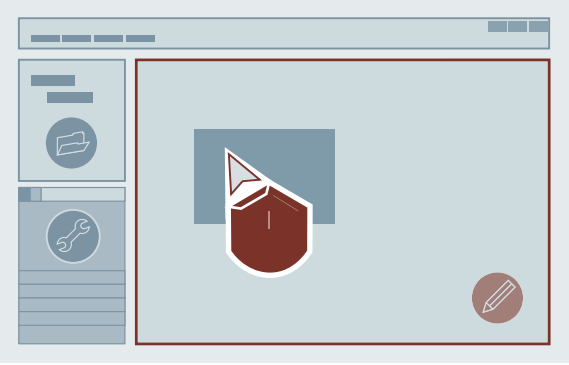
8.7.3.9 Interconnecting an object

Requirement

- The work area is visible.
- The desired CFC is open.
- Multiple objects (blocks or CFC variables) have been created.

Procedure

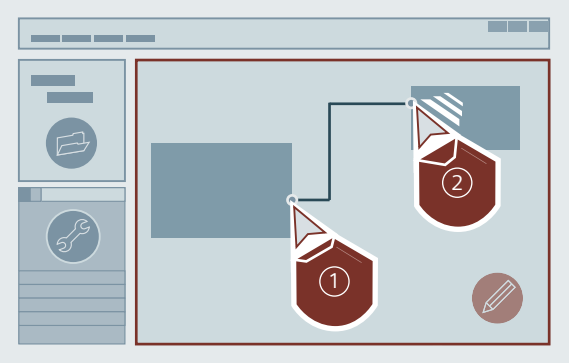
1. In the work area, select an object.



2. Drag a connection from the desired parameter to the parameter of another object.

Note

Take the data type and the interconnection direction into account.



Result

The block is interconnected with another object.

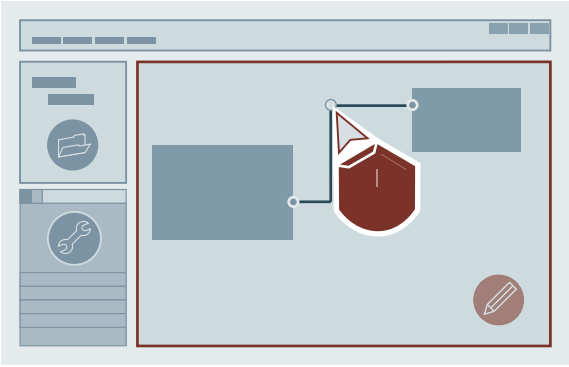
8.7.3.10 Moving an interconnection

Requirement

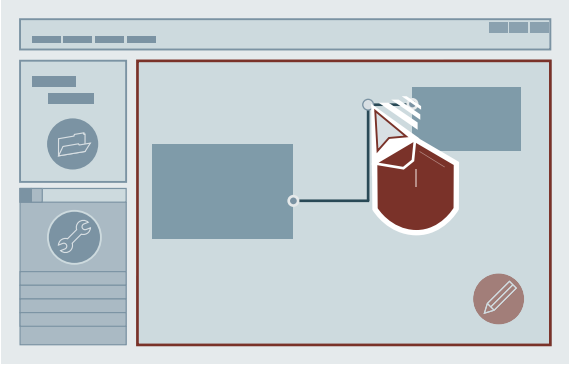
- The work area is visible.
- The desired CFC is open.
- Multiple blocks have been created and interconnected.

Procedure

1. In the work area, select any point on the line of an interconnection.



2. Move the line to the desired position.



Result

The line course of the interconnection has been changed.

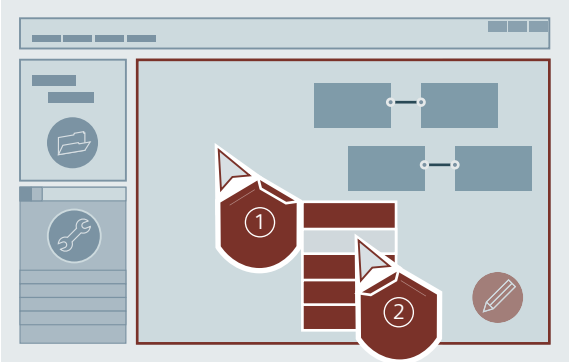
8.7.3.11 Redrawing connections

Requirement

- The work area is visible.
- The desired CFC is open.
- At least one object (block or CFC variable) has been created.

Procedure

Select "Redraw connections".



Result

All interconnections in the work area are redrawn.

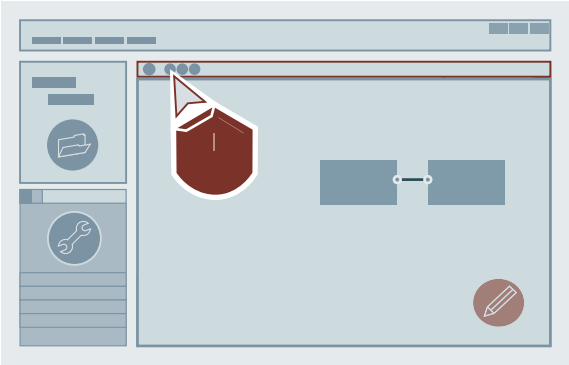
8.7.3.12 Changing the editing history

Requirement

- The work area is visible.
- The desired CFC is open.
- One or more work steps were performed.

Procedure

1. Select a function in the edit bar above the work area.



Button	Description
Undo	Undo the last work step.
Do	Perform the undone work step again.
Redo	Perform the last action again.

Result

The selected function is executed.

8.7.3.13 Limiting the work space according to use

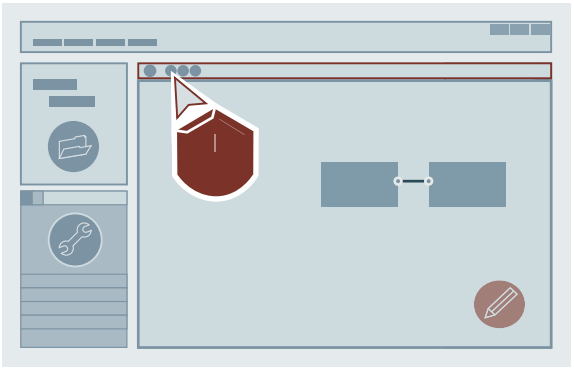
Requirement

- The work area is visible.
- The desired CFC is open.
- One or more objects were placed in the CFC.

Procedure

1. Select the "Point" in the edit bar.





Result

The work space is limited to the area in use.

8.7.4 Program blocks

8.7.4.1 Program blocks

Definition

A program block is the smallest executable part of a program. Each program block is a program in itself. Program blocks can be categorized by their functions and according to topic. Each program block can be used multiple times in the work area as an instance object ("Instance") in the CNET tool.

The program block can be provided with an individual name via its name label. This optional name may contain up to 24 characters.

Structure

Program blocks are created as part of a CFC and interconnected.

CFC (Page 134)

Interconnection of the program blocks takes place via their parameters.

Parameters (Page 180)

8.7.4.2 Parameters

Definition

Parameters are used as the storage location for the processing or transfer of a value. As part of a program block or as a CFC variable, parameters are also designated as variables. The transfer of a value can take place directly or via an interconnection with another parameter.

Note

The rules of electrical engineering apply for interconnecting.

Structure

The parameters of program blocks can be differentiated as follows:

- Input parameters: Receive a value and integrate it into the processing of the block.
- Output parameters: Output a value.
- Local variables: The internal parameters of a block.
  - Input variables = Input parameters
  - Output variables = Output parameters

The parameters of CFC variables can be differentiated as follows:

- Input parameters: Are used as input interfaces of a CFC.
- Output parameters: Are used as output interfaces of a CFC.

Data types

The parameters can have different data types.

Data types (Page 180)

8.7.4.3 Data types

Definition

A data type is the format in which the value of a parameter can be read and processed. The data type also determines the maximum size which the value at the parameter may have.

Description

Data types can be differentiated as follows:

Category	Data type	Description	Value		Converted into bit sequence
			Minimum	Maximum	
Bit sequence	BOOL	Boolean value	FALSE or "0"	TRUE or "1"	-
	BYTE	Byte Array	-	-	8 Bit / BOOL
Integer data types	INT	Integer value	-2^{31} (-2147483648)	2^{31} (2147483647)	4 BYTE (equivalent to STEP 7: DINT)
Floating point values for data types	FLOAT	Floating point value (IEEE 754)	$-3.4*10^{38}$ (-3.40282347E38)	$3.4*10^{38}$ (3.40282347E38)	(equivalent to STEP 7: REAL)
Strings	String	Character string made of CHAR	0	64	64 BYTE

8.7.4.4 Step-by-step instructions

Creating a block

Requirement

- The work area is visible.
- The desired CFC is open.

Procedure

1. In the work area, select "New" > "Instance".



2. Select a block.
   – Optional: Select a block library.



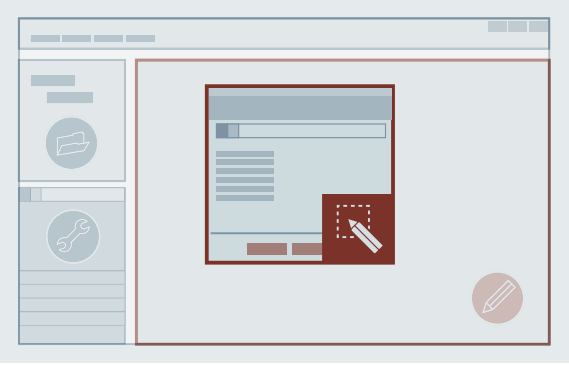
3. Select "Next".



4. Specify values for the parameters.

   Note

   With the filter range you can change the display or filter for a parameter.



5. Select "Finish".



Result

The block is created.

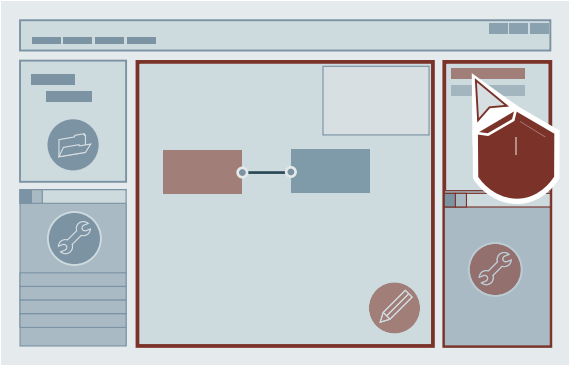
Configuring a block

Requirement

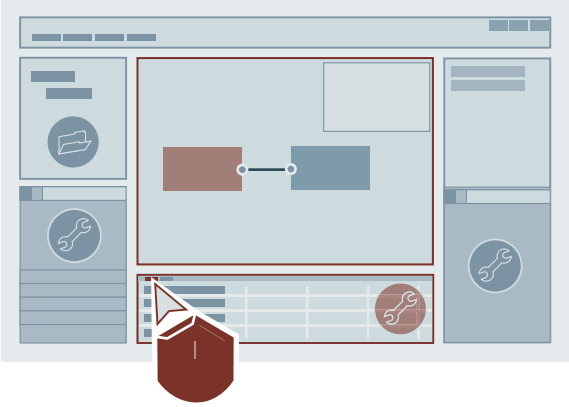
- The work area is visible.
- The desired CFC is open.
- The instance area is displayed.
- At least one block has been created.

Procedure

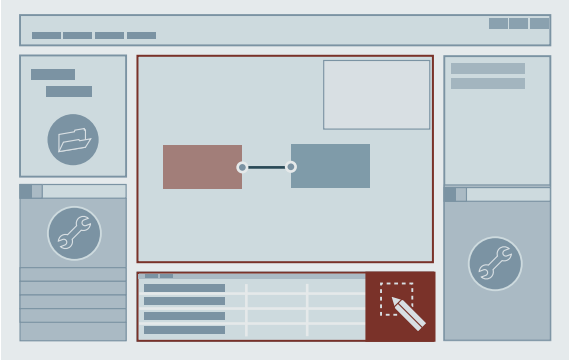
1. Select the desired entry in the instance area or in the work area.



2. Select "Properties" and the Parameter tab (e.g. "Input Output Port / Parameter").



3. Specify new values for the parameters in the parameter area.



Result

The block is configured with new values.

8.7.5 CFC variables ("Input / Output connector")

8.7.5.1 CFC variables ("Input / Output connector")

Definition

A CFC variable is the parameter of a CFC in the CNET tool. The user can create CFC variables as objects ("Input connector" or "Output connector") in the work area.

Structure

Multiple CFC variables can be created as part of a CFC.

CFC (Page 134)

The CFC variables are of the following type:

- Output interface (Output connector):
  An output interface is connected with the parameter of a program block.
- Input interface (Input connector):
  An input interface is interconnected with a previously created output interface of another CFC.

Note

Assignment recognizable

The assignment (starting from the output interface) is displayed at the object of the input interface. The names of CFC, instance (program block) and parameters are specified. If no name was used for the instance, the ID of the instance is specified.

8.7.5.2 Step-by-step instructions

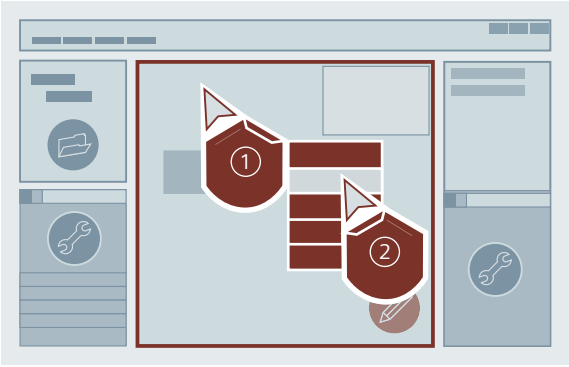
Creating and interconnecting an output interface

Requirement

- The work area is visible.
- The desired CFC is open.
- At least one program block has been created.

Procedure

1. In the work area, select "New" > "Output connector".



2. Interconnect the output interface with an output parameter of a program block.
Interconnecting an object (Page 174)

Result

The output interface has been created and can be used.

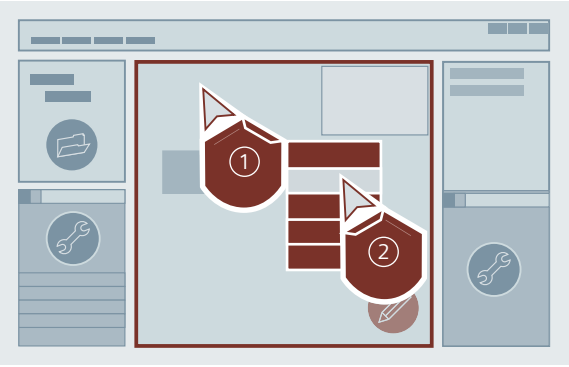
Creating the input interface

Requirement

- The work area is visible.
- Multiple CFCs have been created.
- An output interface is created in CFC A and can be used.
- CFC B is open.

Procedure

1. In the work area of the CFC B, select "New" > "Input connector".



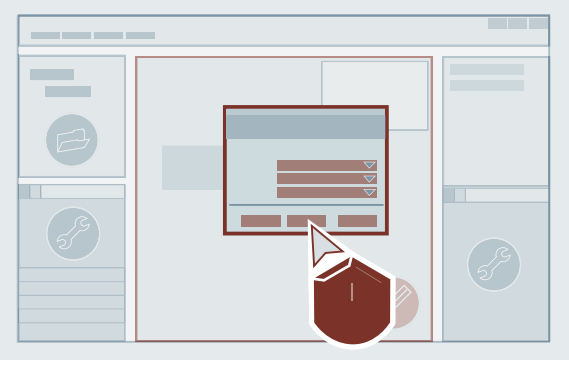
2. Select the CFC A, block and block parameters.

Note

The selection depends on the previously created output interfaces and interconnections.



3. Select "Finish".



Result

The input interface has been created and interconnected with an output interface.

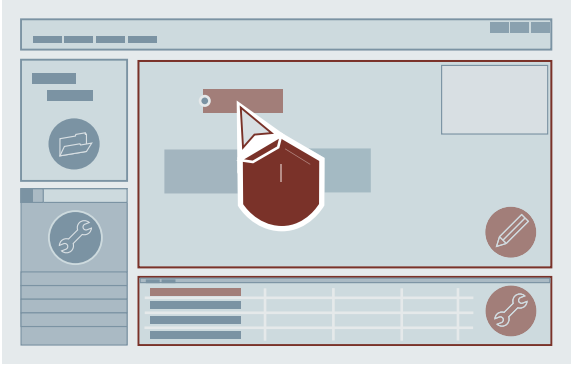
Displaying properties of a CFC variable

Requirement

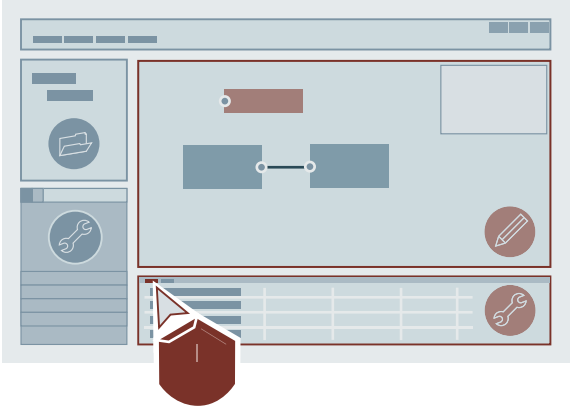
- The work area is visible.
- The desired CFC is open.
- The parameter area is displayed.
- At least one block and a CFC variable have been created.

Procedure

1. In the work area, select a CFC variable.



2. Select "Properies".
   – Optional: Select the sub-tab "Generic Property" or "Source".



Result

The properties of the CFC variable are displayed in the parameter area.

8.7.6 Minimap

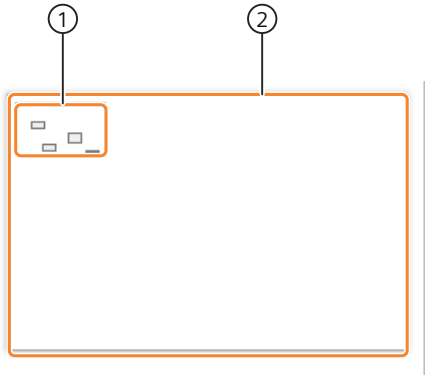
8.7.6.1 Minimap

Definition

The minimap ("Diagram map") is an area of the CNET tool which displays a zoomed out version of the entire work area. The user can get a better overview of the work area by means of the minimap.

Structure

The minimap is structured as follows:



①	Focus area	Shows the section of the work area. Moving a partial image (Page 189)
②	Entire work area	Shows the work area that is available for a CFC. Is used to provide an overview.

8.7.6.2 Step-by-step instructions

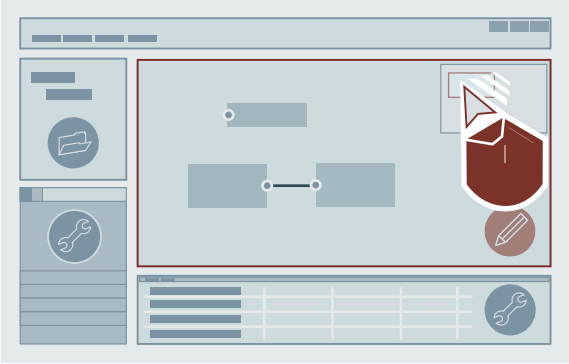
Moving a partial image

Requirement

- The work area is visible.
- The respective CFC is open.
- The minimap ("Diagram map") is shown.

Procedure

In the minimap drag the blue rectangle to the desired position.



Result

The partial image is repositioned.

8.8 Parameter area

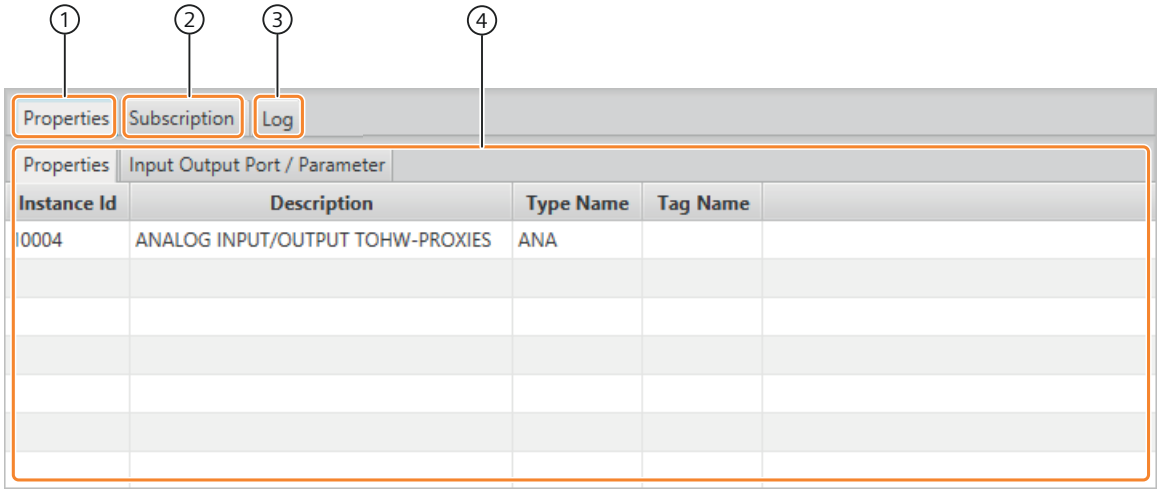
8.8.1 Parameter area

Definition

The parameter area is an area of the CNET tool where the user can switch between the 'Block and connection properties' and the watch table.

Structure

The parameter area is structured as follows:



①	Properties	button. Opens the view of the properties of a selected program block or CFC variable. Block and connection properties ("Properties") (Page 191)
②	Subscription	button. Opens the view for a previously selected watch table. Watch table ("Subscription") (Page 195)
③	Log	Button. Opens the view of the CNET event log. Event log ("Log") (Page 217)
④	Properties or parameters	Displays the content of the currently selected display. Current display: General properties of a program block.

8.8.2 Block and connection properties ("Properties")

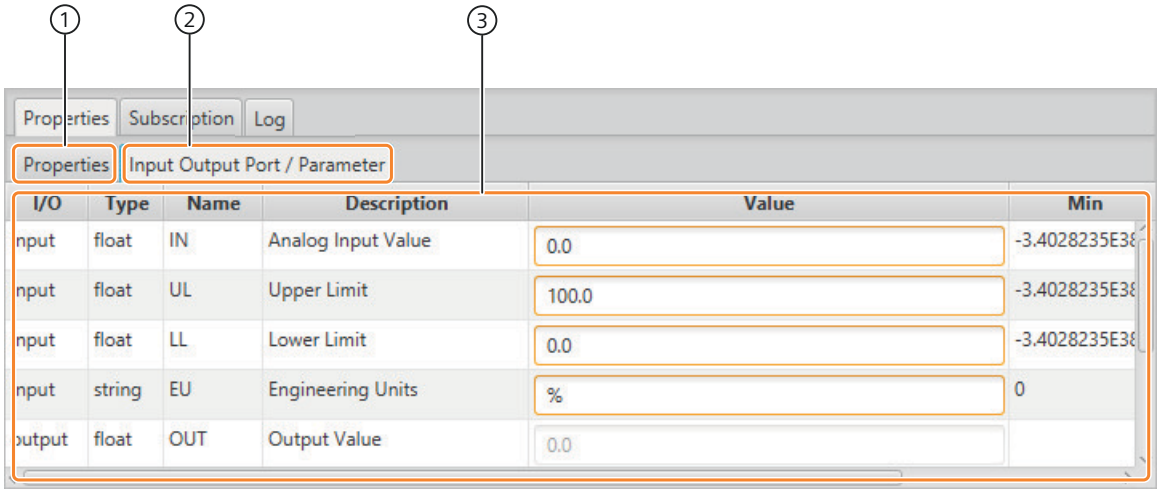
8.8.2.1 Block and connection properties ("Properties")

Definition

The block and connection properties are an area of the CNET tool where the user can display the properties of a selected program block or CFC variable.

Structure

The block and connection properties are structured as follows:



①	Properties or Generic Property	button. Opens the overview of the general properties of the program block or a variable. Properties (block) (Page 192) Generic Property (CFC variable) (Page 194)
②	Parameter properties (e.g., "Input Output Port / Parameter") or Source	button. Opens the overview of the parameters of the program block or the source specifications of a CFC variable. Input Output Port / Parameter (block) (Page 193) Source (CFC variable) (Page 194)
③	Properties or parameters	Displays the content of the currently selected display.

Note

Showing block and connection properties

The area is automatically shown as soon as a program block or a CFC variable has been selected.

The menu ("Views" > "Subscriptions") can be used to show or hide the parameter area (or the block and connection properties).

Showing / hiding a window (Page 50)

8.8.2.2 Properties (block)

Definition

Properties is an area of the CNET tool where the user can view the general properties of a selected program block.

Structure

Properties is structured as follows:

①	Instance ID	Displays the ID of the instance of the program block.
②	Description	Displays the description of the program block.
③	Type Name	Displays the type of the program block.
④	Tag Name	Displays the assigned name of the program block.

8.8.2.3 Input Output Port / Parameter (block)

Definition

Input Output Port / Parameter is an area of the CNET tool where the user can view the parameters of a selected program block. The area is a sub-tab of the block and connection properties. The titles of the tabs differ depending on the program block. In addition, the area (Input Output Port / Parameter) can also consist of several differently titled tab entries.

Structure

Input Output Port / Parameter is structured as follows:

①	I/O	Displays the connection type of the parameter. Possible values: • Input • Output
②	Type	Displays the data type of the parameter.
③	Name	Shows the name of the parameter.
④	Description	Displays the description of the parameter.
⑤	Value	Input field or drop-down list. Shows the value of the parameter. If necessary, additionally with submenu whose dialog can be accessed via "...".
⑥	Min	Displays the minimum possible value of the parameter.
⑦	Max	Displays the maximum possible value of the parameter.
⑧	Visibility	option. Allows you to hide or show the parameter on the program block. Not every parameter has this option.

Note

Checking settings (Value)

The validity of related settings for parameters of the same topic group of a program block is checked automatically.

However, for related parameters, you should pay attention to the order in which you configure them. Lastly, you should check the settings again yourself. This circumstance applies in particular to protocols and their values to be subsequently entered at other parameters.

8.8.2.4 Generic Property (CFC variable)

Definition

Generic Property is an area of the CNET tool where the user can view the general properties of an interface of a CFC variable.

Structure

Generic Property is structured as follows:

①	Diagram	Displays the CFC in which the selected interface of the CFC variable is located.
②	Connector id	Displays the port ID of the interface.
③	Connector type	Displays the type of interface.

8.8.2.5 Source (CFC variable)

Definition

Source is an area of the CNET tool where the user can view the source information of the output interface (CFC variable).

Structure

Source is structured as follows:

①	AF Container	Displays the CPU of the output interface.
②	AF Diagram	Displays the CFC in which the output interface is located.
③	AF Instance	Displays the details for the instance of the program block to which the output interface is connected: AF Instance: Displays the ID of the instance of the program block. AF Instance type: Displays the type of the program block. AF Instance port: Displays the parameter of the program block.
④	AF Instance type	
⑤	AF Instance port	

8.8.3 Watch table ("Subscription")

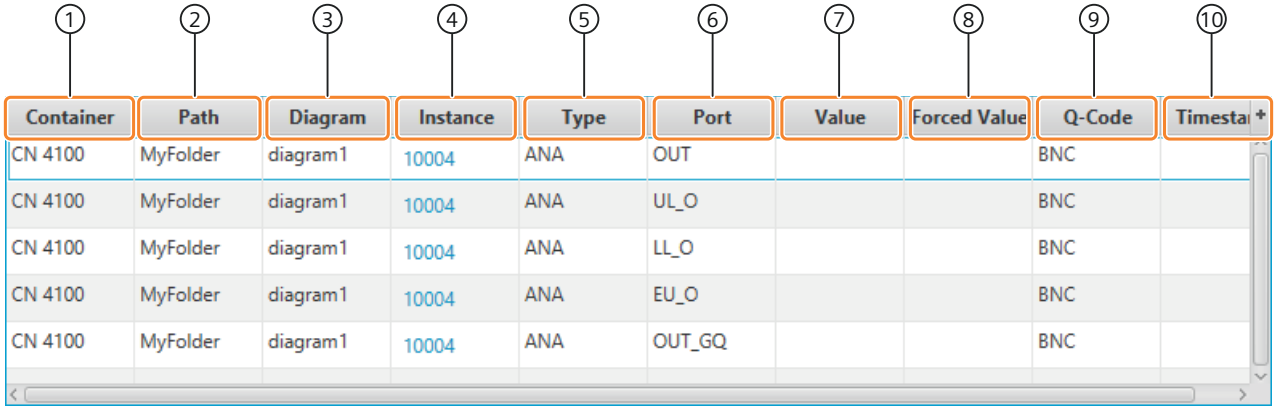
8.8.3.1 Watch table ("Subscription")

Definition

The watch table ("Subscription") is an area of the CNET tool where the user can enter selected parameters for monitoring. Monitoring of the parameters takes place in online mode of the project during the runtime of the program.

Structure

The watch table is structured as follows:



①	Container	Displays the name of the hardware which is to be accessed.
②	Path	Displays the folder path where the used CFC is located.
③	Diagram	Shows the name of the CFC which is to be accessed.
④	Instance	Shows the block which is to be accessed by means of its identification.
⑤	Type	Shows the name or type to which the block belongs.
⑥	Port	Shows the parameter which is to be monitored. Instance ports (Page 196)
⑦	Value	Shows the value which the parameter has. • In the Online view, the value is displayed in the work area (on the block) with a colored background. The display differs according to the quality (Q-code) of the value. Online view (Page 199)

⑧	Forced Value	Displays the value to be forced that the parameter is to use. • Changing this value via the watch table can only be done in online mode. • If forcing is activated, the connection is highlighted in color on the program block. • When forcing is deactivated, the parameter is reset to the original value.
⑨	Q-Code	Shows the signal status of the entry. Quality code (Page 198)
⑩	Timestamp	Shows the current time stamp of the entry.

Note

Showing the watch table

The surrounding parameter area of the watch table can be shown or hidden via the menu ("Views" > "Subscriptions").

Showing / hiding a window (Page 50)

The parameter area is automatically shown as soon as a program block or a CFC variable has been selected. The block and connection properties are displayed by default.

Switching to watch table (Page 205)

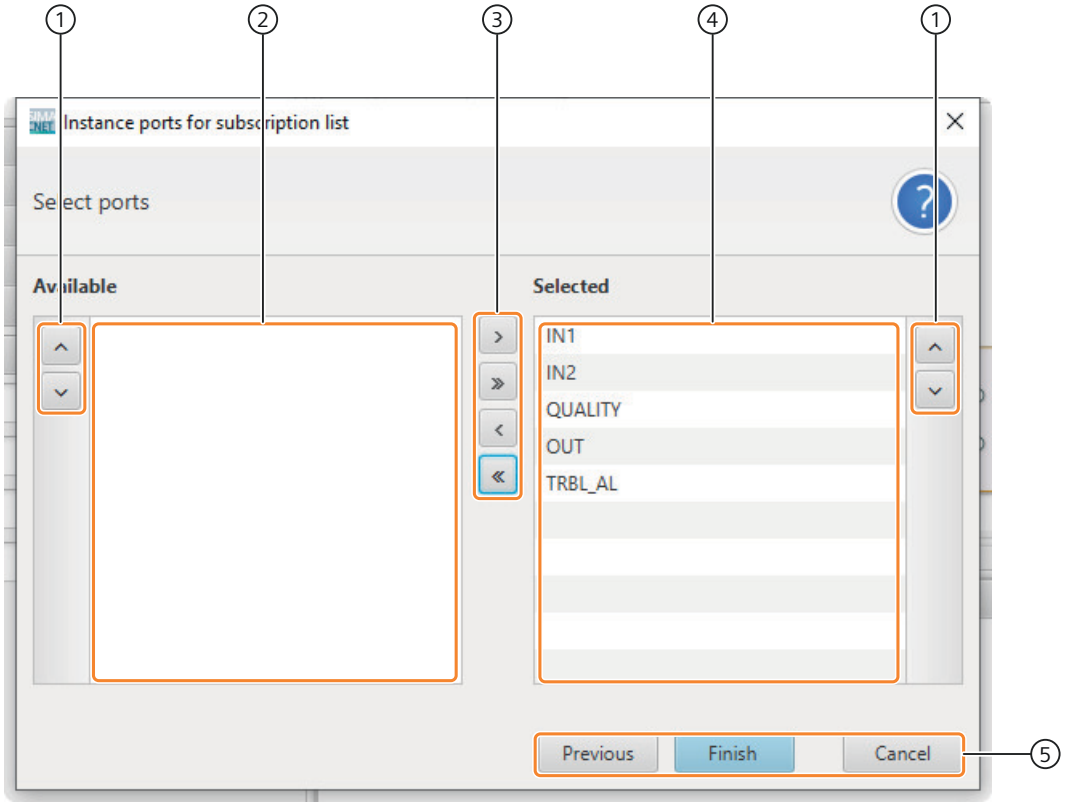
8.8.3.2 Instance ports

Definition

The "Instance ports for subscriptions" dialog provides an overview of selectable parameters for the watch table.

Structure

The overview of selectable parameters is structured as follows:



①	Buttons for the sorting function	Offers the option to sort individual parameters in the list. • Arrow high: Moves the selected parameter in the list upwards. • Arrow down: Moves the selected parameter in the list downwards.
②	List of selectable parameters	Parameters (Page 180)
③	Buttons for the selection	Offers the option to move parameters to the respective other list: ">": Moves the selected parameter to the right. "≫": Moves all parameters to the right. "<": Moves the selected parameter to the left. "≪": Moves all parameters to the left.
④	List of parameters for monitoring	Parameters (Page 180)
⑤	Navigation through the dialog.	Navigating in dialogs (Page 50)

8.8.3.3 Quality code

Definition

The quality code is a status which a signal (the value of a parameter) can have during the transfer.

List of states

The quality codes for the limit values are as follows:

Data type	Status	Status in BYTE
Byte	QUALITY_LIMIT_MASK	0x03
Byte	QUALITY_LIMIT_OK	0x00
Byte	QUALITY_LIMIT_LOW	0x01
Byte	QUALITY_LIMIT_HIGH	0x02
Byte	QUALITY_LIMIT_CONST	0x03

The quality codes for the status are as follows:

Data type	Status	Status in BYTE
Byte	QUALITY_STATE_MASK	0xC0
Byte	QUALITY_BAD	0x00
Byte	QUALITY_UNCERTAIN	0x40
Byte	QUALITY_GOOD	0xC0
Byte	QUALITY_SUBSTATE_MASK	0x3C

The quality codes for the communication are as follows:

Data type	Status	Status in BYTE
Byte	QUALITY_UNCONNECTED	0x20
Byte	QUALITY_LOCAL_OVERRIDE	0x18

The quality codes for a good status are as follows:

Data type	Status	Status in BYTE
Byte	QUALITY_GOOD_UNCONNEC-TED	0xE0
Byte	QUALITY_GOOD_LOCAL_OVER-RIDE	0xD8
Byte	QUALITY_GOOD_DANGER_TAG-OUT	0xC4
Byte	QUALITY_GOOD_CAUTION_TAG-OUT	0xC8
Byte	QUALITY_GOOD_DANGER_CAU-TION_TAGOUT	0xCC
Byte	QUALITY_GOOD_ON_TEST	0xD0

Data type	Status	Status in BYTE
Byte	QUALITY_GOOD_PADLOCKED	0xD4
Byte	QUALITY_GOOD_ON_TEST_PAD-LOCKED	0xDC

The quality codes for an unsafe state are as follows:

Data type	Status	Status in BYTE
Byte	QUALITY_UNCER-TAIN_LAST_USABLE	0x44
Byte	QUALITY_UNCERTAIN_SEN-SOR_CAL	0x50
Byte	QUALITY_UNCERTAIN_EGU_EX-CEEDED	0x54
Byte	QUALITY_UNCERTAIN_SUB_NOR-MAL	0x58
Byte	QUALITY_UNCERTAIN_SUBSTI-TUE_VALUE	0x48
Byte	QUALITY_UNCERTAIN_INI-TIAL_VALUE	0x4C

The quality codes for a bad state are as follows:

Data type	Status	Status in BYTE
Byte	QUALITY_BAD_CONFIG_ERROR	0x04
Byte	QUALITY_BAD_NOT_CONNEC-TED	0x08
Byte	QUALITY_BAD_DEVICE_FAILURE	0x0C
Byte	QUALITY_BAD_SENSOR_FAILURE	0x10
Byte	QUALITY_BAD_LAST_KNOWN	0x14
Byte	QUALITY_BAD_COMM_FAILURE	0x18
Byte	QUALITY_BAD_OUT_OF_SERVICE	0x1C

The quality codes for the transfer to a further parameter are as follows:

Data type	Status	Status (transfer)
Byte	QUALITY_GOOD_PARAMETER	QUALITY_GOOD_UNCONNEC-TED
Byte	QUALITY_GOOD_FORCED	QUALITY_GOOD_LOCAL_OVER-RIDE

8.8.3.4 Online view

Definition

The Online view is the state of the watch table in CNET in which the entries of a selected watch table display their current values in the watch table and in the work area.

Colors

The objects in the online view are colored according to their quality code:

- BAD = Red
- UNCERTAIN = Gray
- GOOD = Light orange

8.8.3.5 Step-by-step instructions

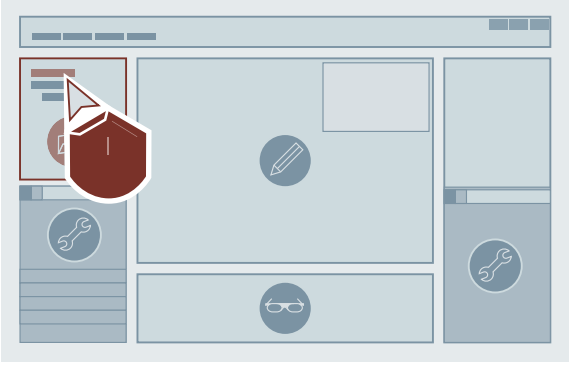
Creating a watch table

Requirement

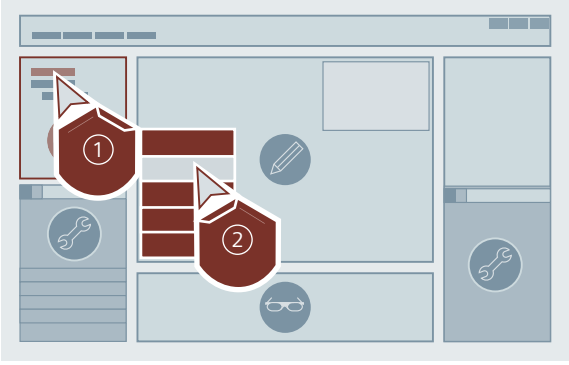
- The navigation area is visible.

Procedure

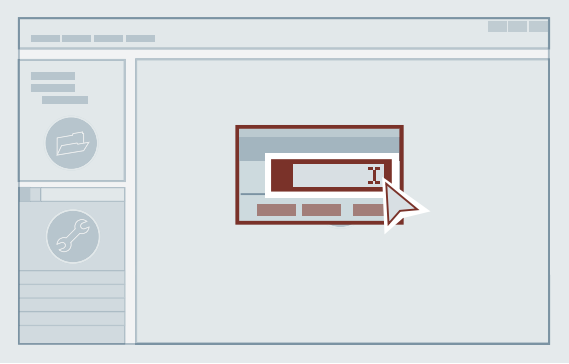
1. Select "Subscription" in the navigation area.



2. Select "Subscription" > "Add".



3. Enter a name.



4. Select "Finish".



Result

A new watch table has been created.

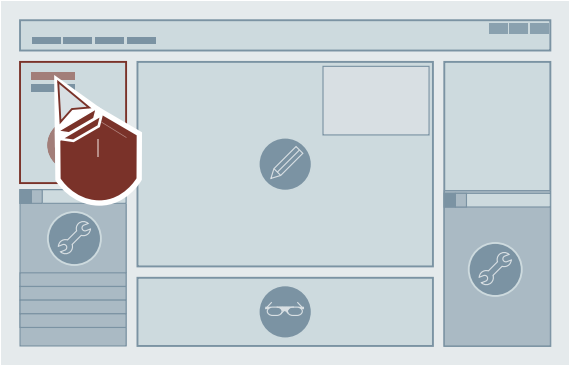
Opening a watch table

Requirement

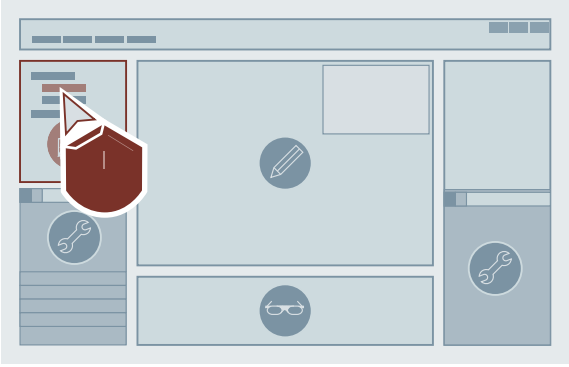
- The navigation area is visible.

Procedure

1. Open the "Subscription" entry in the navigation area.



2. Select a created watch table.



Result

The selected watch table is open.

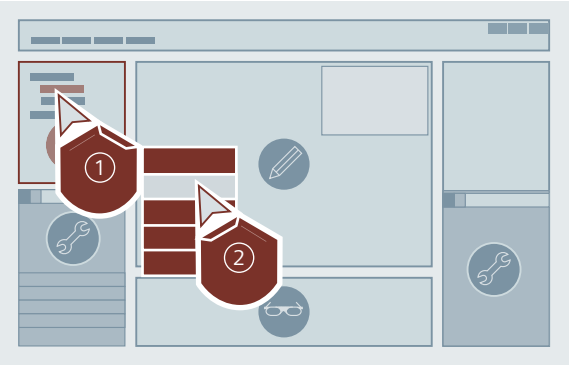
Renaming the watch table

Requirement

- The navigation area is visible.
- The desired watch table is open.

Procedure

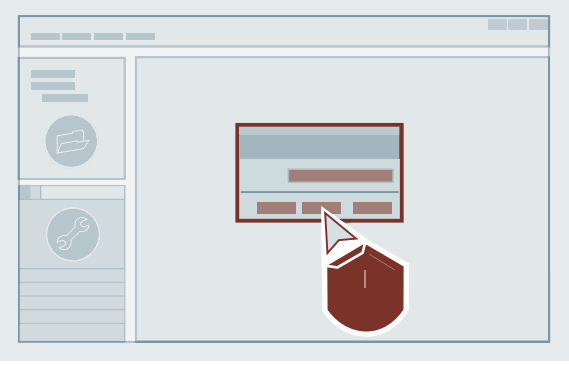
1. Select "Subscription" > "Edit" for the watch table.



2. Enter a new name.



3. Confirm.



Result

The selected watch table has been renamed.

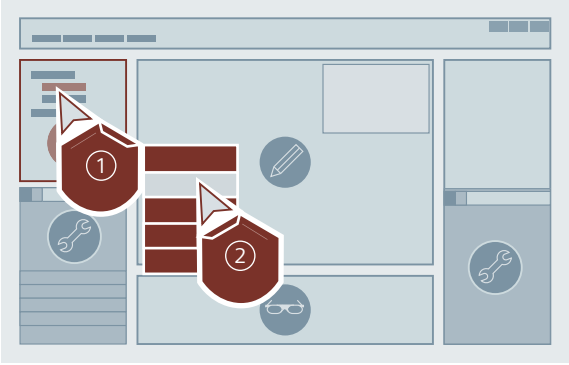
Removing a watch table

Requirement

- The navigation area is visible.
- The desired watch table is open.

Procedure

1. Select "Subscription" > "Remove" for the watch table.



2. Confirm.



Result

The selected watch table has been removed.

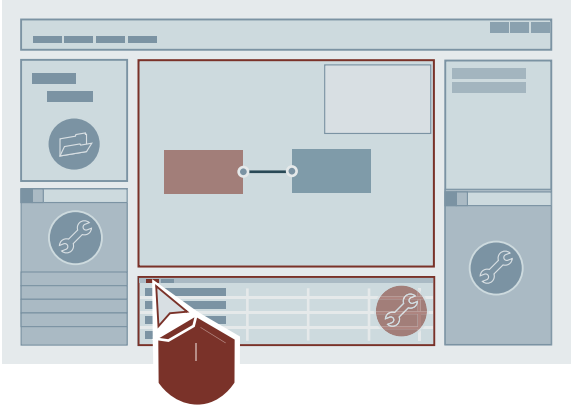
Switching to watch table

Requirement

- The work area is visible.
  - The desired CFC is open.
- The parameter area is displayed.
- A watch table has been created.

Procedure

Select "Subscription".



Result

The entries of a previously selected watch table are displayed.

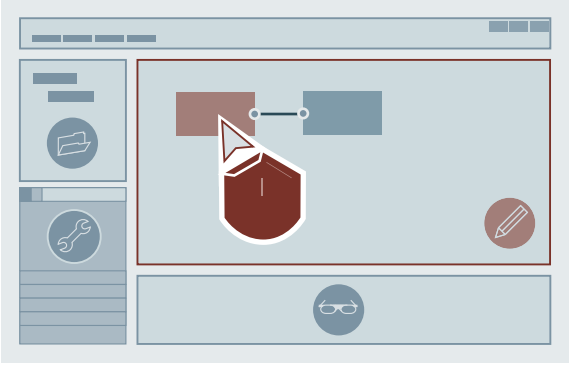
Creating entries for monitoring

Requirement

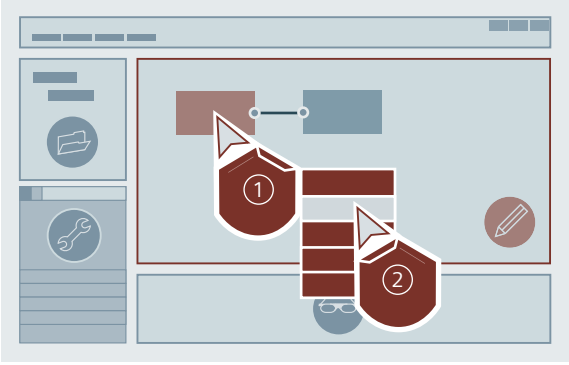
- The work area is visible.
- The respective CFC is open.
- The desired watch table is open.

Procedure

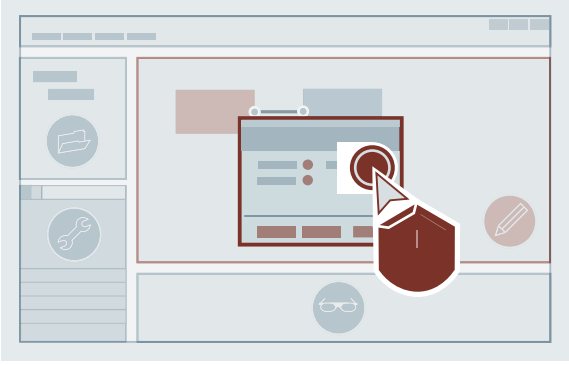
1. Select a block.



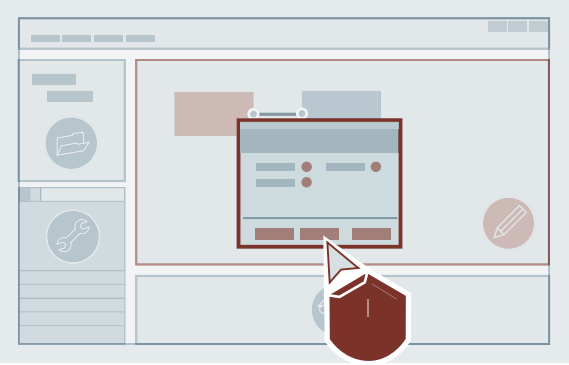
2. Select "Subscription".



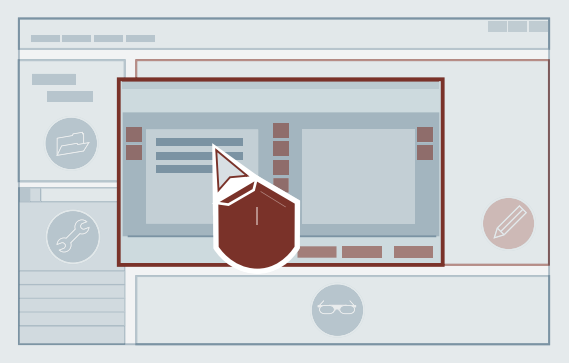
3. Select the categories to be displayed.



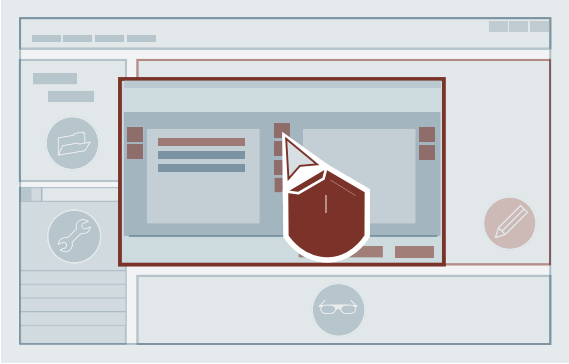
4. Select "Next".



5. Select the desired parameter.



6. Select ">" or "≫".



7. Select "Finish".



Result

The entries are created in the watch table.

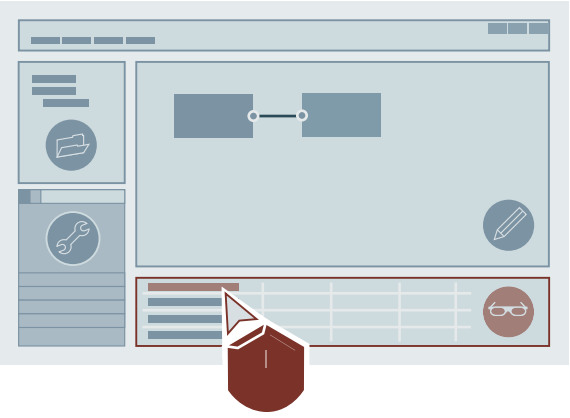
Removing entries from monitoring

Requirement

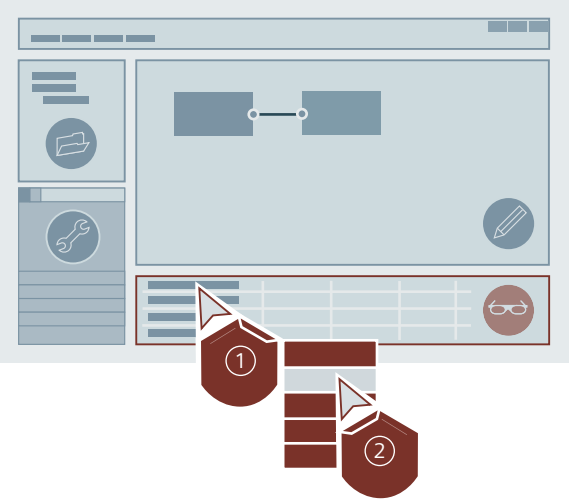
- The work area is visible.
- The desired watch table is open.
- Entries are created in the watch table.

Procedure

1. Select one or more entries in the watch table.



2. Select "Remove selection".



Result

The selected entries have been removed from the watch table.

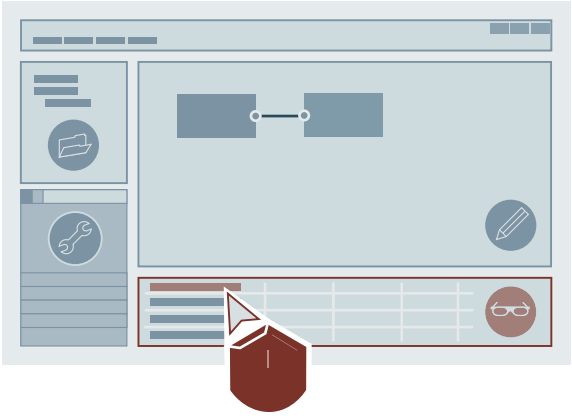
Resetting the watch table

Requirement

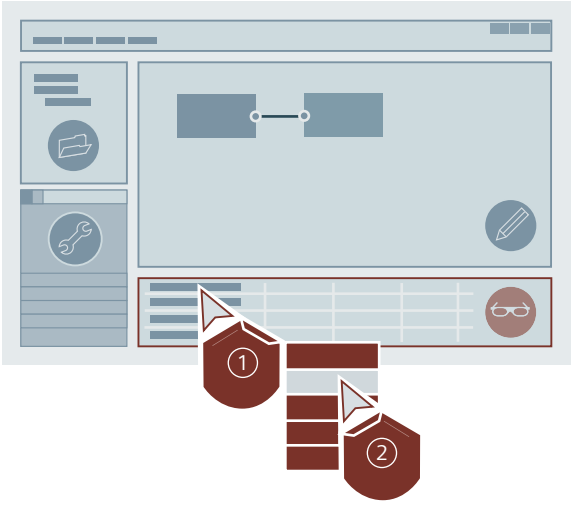
- The work area is visible.
- The desired watch table is open.
- Entries are created in the watch table.

Procedure

1. Optional: Select any entry in the watch table.



2. Select "Reset subscription".



Result

All entries have been removed from the watch table.

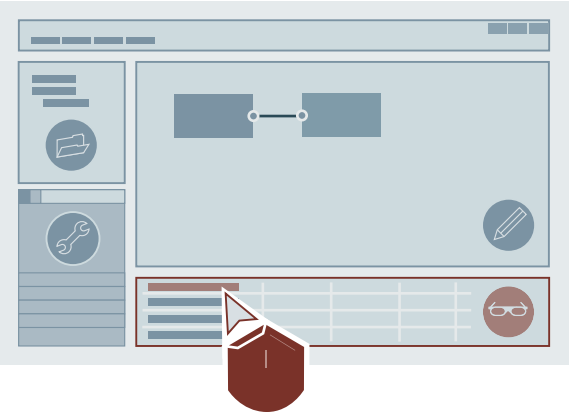
Enabling entries for monitoring

Requirement

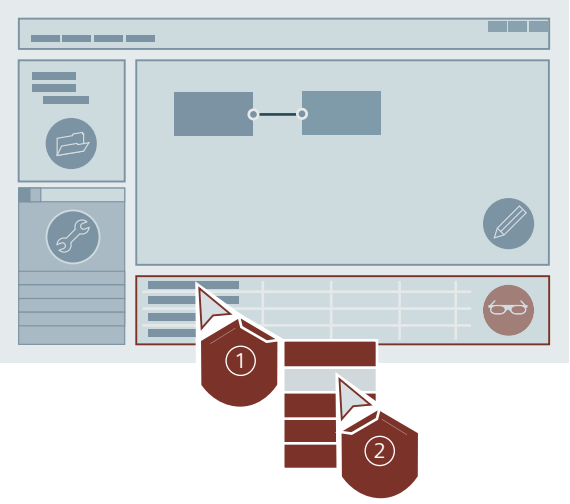
- The work area is visible.
- The desired watch table is open.
- Entries are created in the watch table.

Procedure

1. Select any entry in the watch table.



2. Select "Subscribe values".



Result

All entries of the watch table are enabled for monitoring.

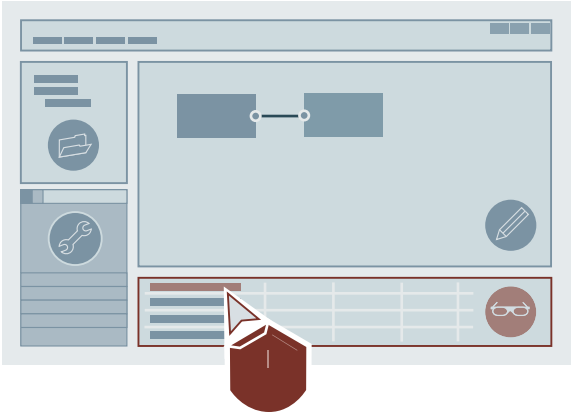
Revoking monitoring

Requirement

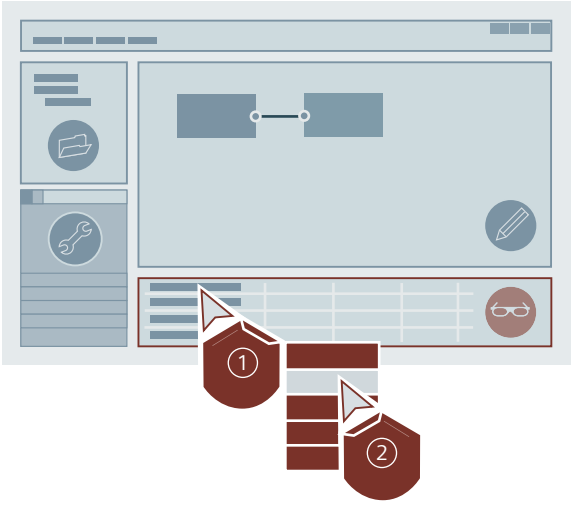
- The work area can be seen.
- The desired watch table is open.
- The project is in online mode.
- Entries are created in the watch table and under monitoring.

Procedure

1. Select any entry in the watch table.



2. Select "Unsubscribe values".



Result

All entries are not monitored anymore. The last read values are still shown in the online mode.

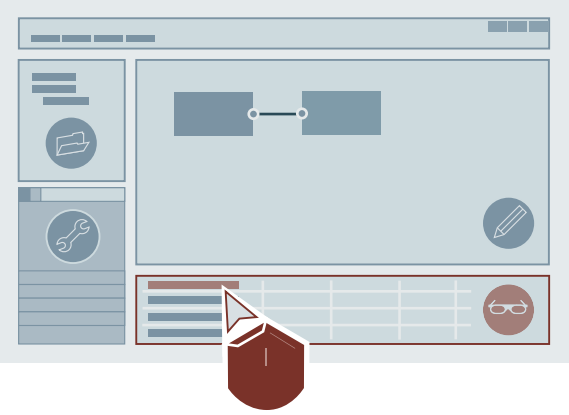
Navigating from the watch table to the instance

Requirement

- The work area is visible.
- The desired watch table is open.
- Entries are created in the watch table.

Procedure

1. In the "Instance" column of the watch table, select the specified number of an entry to be selected.



Result

In the work area, the view pans to the desired instance of a block. The block is displayed selected.

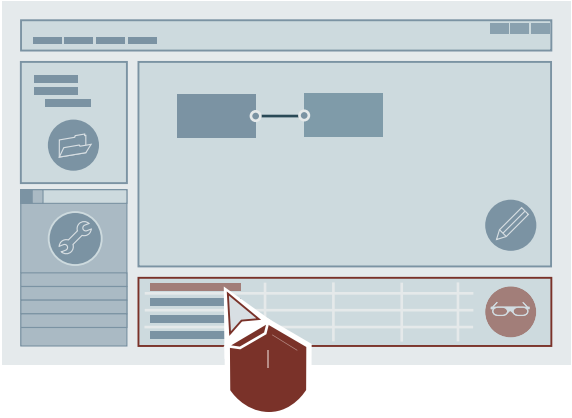
Activating forcing of the watch table

Requirement

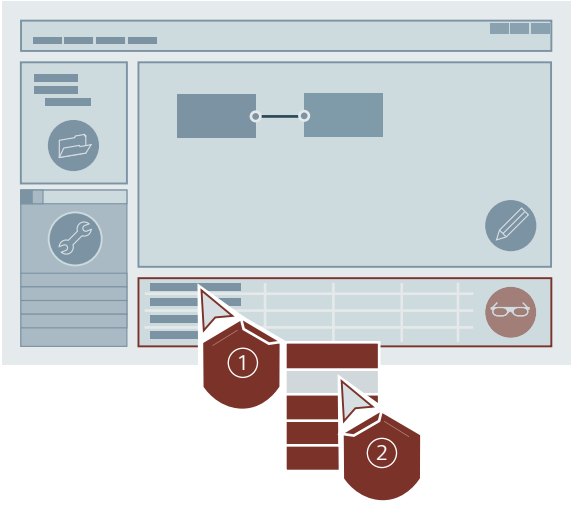
- The work area is visible.
- The desired watch table is open.
- The project is in online mode.
- Entries are created in the watch table and under monitoring.

Procedure

1. Select an entry and enter a desired value in the "Force Value" column.



2. Select "Enable Forcing".



Result

The "Force" function is activated for the watch table. The value is forced for a selected entry.

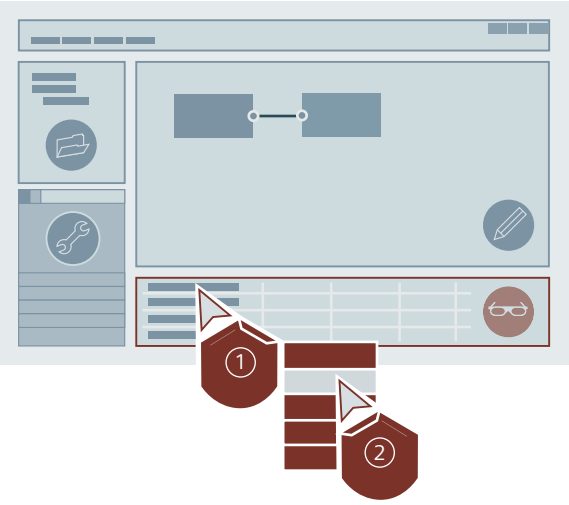
Deactivating forcing of the watch table

Requirement

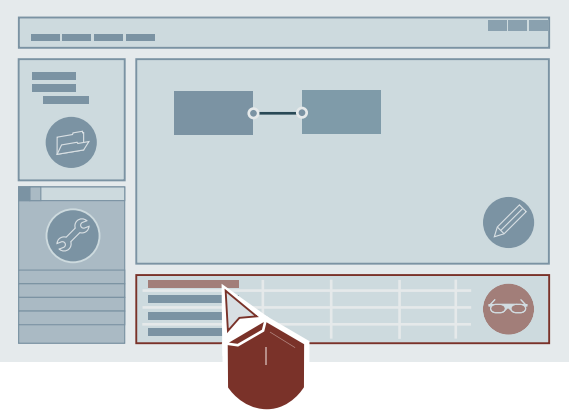
- The work area is visible.
- The desired watch table is open.
- The project is in online mode.
- Entries are created in the watch table and under monitoring.

Procedure

1. Select "Disable Forcing".



2. Optional: Remove/change to the value of the "Force Value" column of a selected entry.



Result

The "Force" function is deactivated for the watch table. The original value is displayed again for the selected entry.

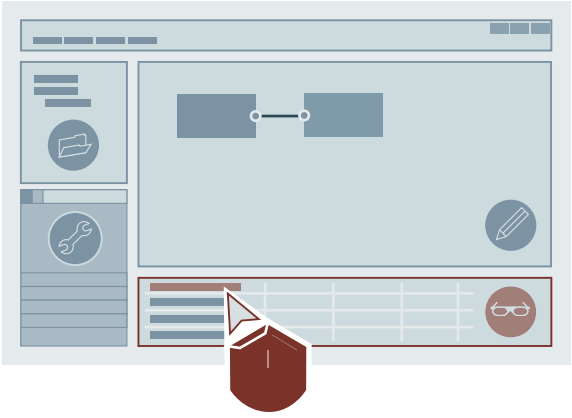
Activating online view

Requirement

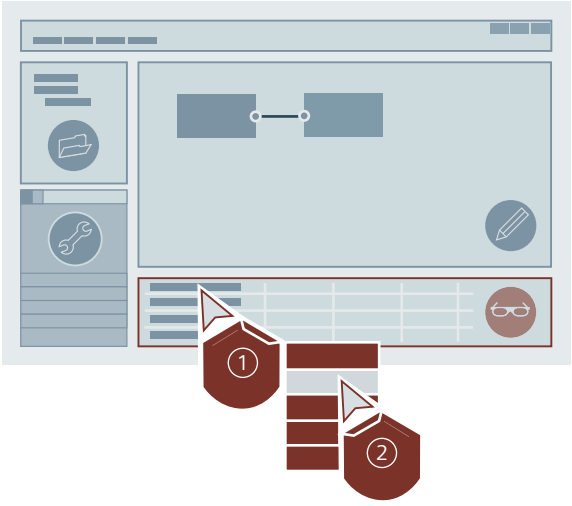
- The work area is visible.
- The desired watch table is open.
- The project is in online mode.
- Entries are created in the watch table and under monitoring.

Procedure

1. Optional: Select any entry in the watch table.



2. Select "Enable Online View".



Result

The online view of the watch table is activated. The current values are displayed in the work area. The status of the values is indicated by a color at the values.

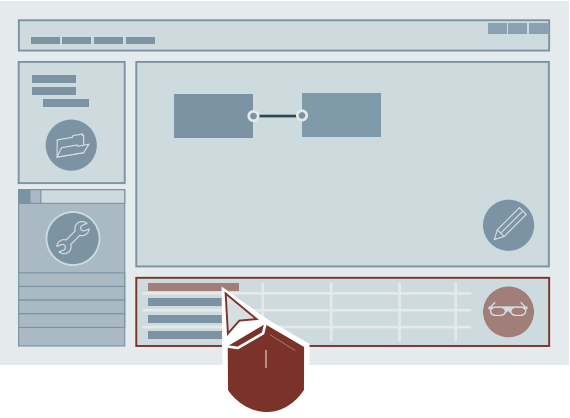
Deactivating online view

Requirement

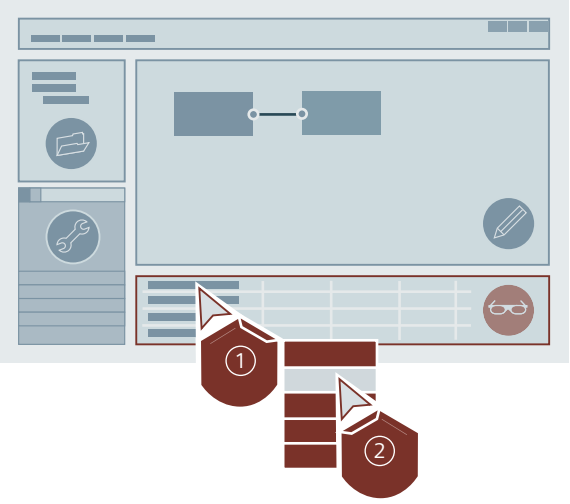
- The work area is visible.
- The desired watch table is open.
- The project is in online mode.
- Entries are created in the watch table and under monitoring.

Procedure

1. Optional: Select any entry in the watch table.



2. Select "Disable Online View".



Result

The online view for the watch table is deactivated.

8.8.4 Event log ("Log")

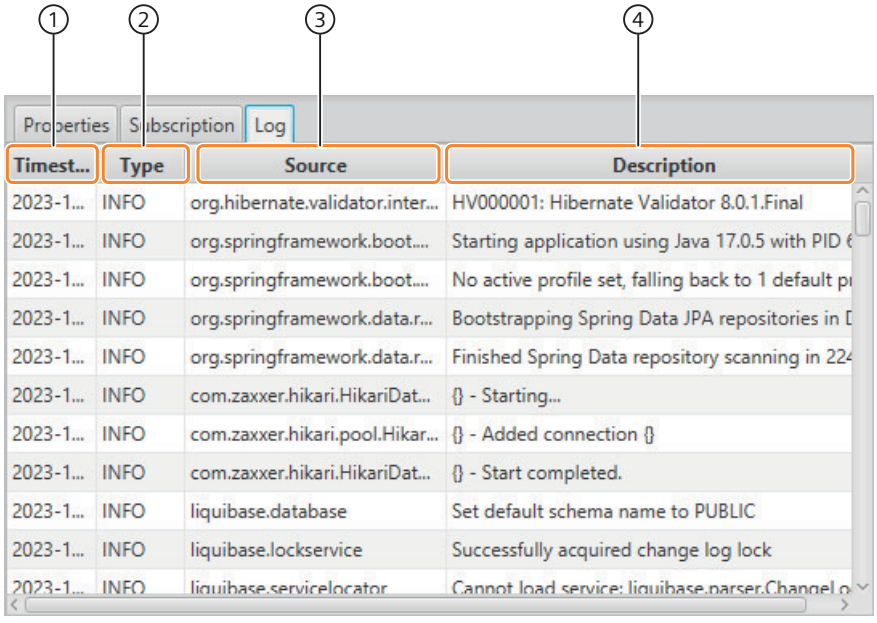
8.8.4.1 Event log ("Log")

Definition

The event log ("Log") is an area of the CNET tool where the user can view the most important CNET events.

Structure

The event log is structured as follows:



①	Timestamp	Column. Shows the time stamp of logging of the event.
②	Type	Column. Shows the type of the event. The following types exist: • INFO • ERROR
③	Source	Column. Shows the source for triggering of the event.
④	Description	Column. Shows the description of the event.

8.8.4.2 Step-by-step instructions

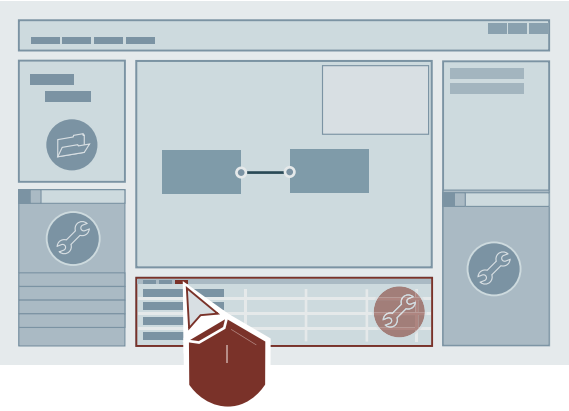
Showing the event log

Requirement

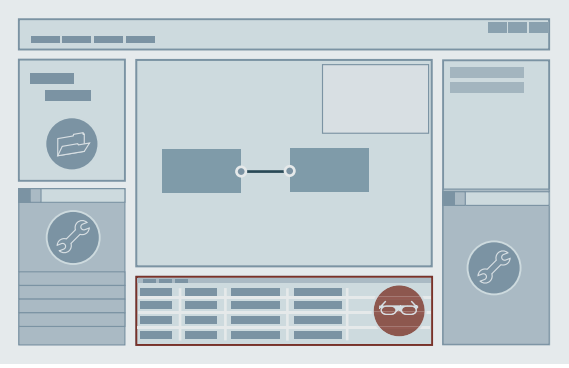
- The work area is visible.
- The parameter area is displayed.

Procedure

1. Select "Log".



2. Look at the events in the log.



Result

The events logged by CNET are shown.

8.9 Instance area

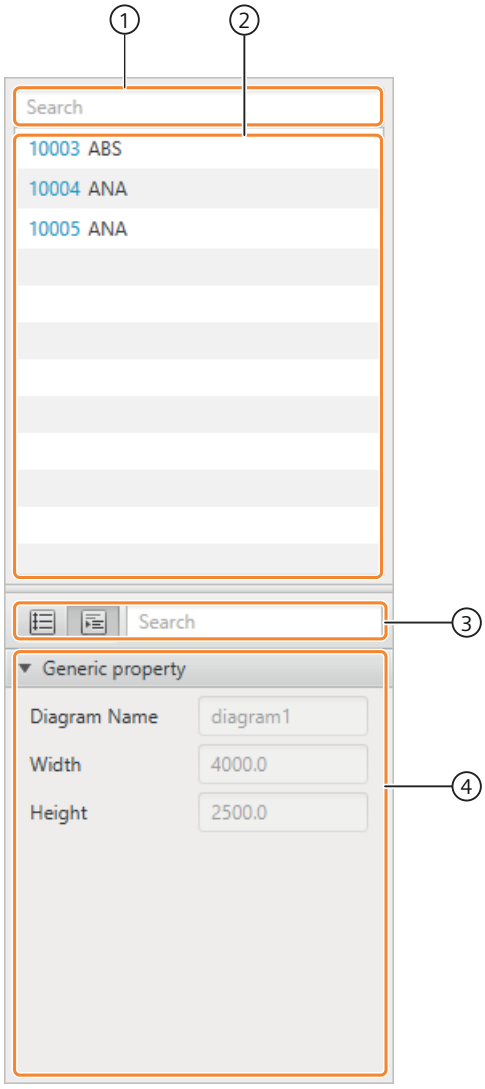
8.9.1 Instance area

Definition

The instance area is an area of the CNET tool in which the user can list all created program blocks of the CFC and display the properties of the CFC.

Structure

The instance area is structured as follows:



①	Filter area of the program blocks	The list of program blocks can be selectively searched via the input field. It is possible to search via identifier or type of instance.
②	List of program blocks	The created program blocks are listed as sorted by CNET. The program blocks are processed by the hardware in the sequence shown in the instance area. Program blocks (Page 179)
③	Filter area of the CFC	The properties of the CFC can be filtered. Filter area (Page 221)
④	Properties of the CFC	Displays the properties of the selected CFC. CFC (Page 134) ("Diagram")

8.10 Filter area

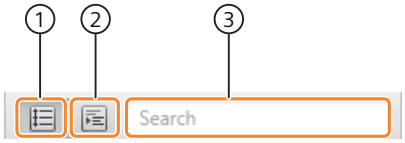
8.10.1 Filter area

Definition

The filter area is an area of the CNET tool which is used multiple times in which the user can access the options for filtering. The filtering always has an influence on the display of a subsequent listing of data.

Structure

The filter area is structured as follows:



①	Button "By Name"	Option for the display of the subsequent listing by name. The data are then displayed without categories.
②	Button "By Category"	Options for the display of the subsequent listing by category. The data are then displayed for the currently selected category. Expanding categories (Page 163)
③	Search	Option for filtering the list according to an entered name.

8.10.2 Step-by-step instructions

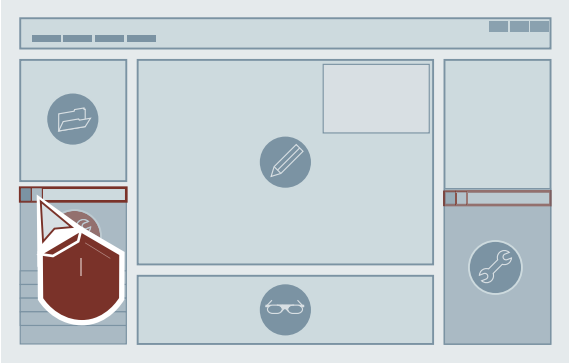
8.10.2.1 Changing the list

Requirement

- The window of the respective topic is shown.
- You have selected an object (e.g. hardware) in the respective window.

Procedure

Select the non-activated button in the respective window in the filter area.



Result

In the window the sorting of the data listing is changed.

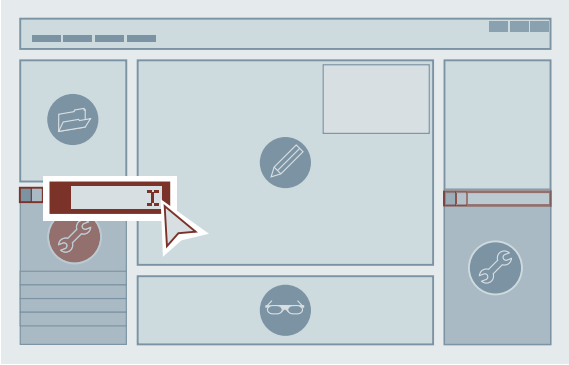
8.10.2.2 Filter by names

Requirement

- The window of the respective topic is shown.
- You have selected an object (e.g. hardware) in the respective window.

Procedure

Enter a name in the respective window > Filter area > "Search".



Result

In the window the data listing is filtered by the entered name.

Block libraries

A

A.1 Parameter tables

Definition

The IOs (ports) of the blocks are listed in the parameter tables.

Structure

The abbreviations of the parameter tables have the following meaning:

Name of the column	Meaning
Active	This text is output in the message sequence display for coming of the message.
Inactive	This text is output in the message sequence display for going of the message.
Message	This port triggers a message.
Message class	A standard or user-defined message class.
Conn	Can be connected. This parameter can be linked to the parameters of other automation functions or blocks.
Edit.	Can be edited. The value of this port can be edited in the instance area.
L	This port appears in the instance area of the AF block.
HMI (*)	This port appears in a faceplate and/or in a pictogram.
DIA (*)	This port is shown in the diagnostics view.

* The relevant information/function cannot be used in CNET.

A.2 General quality statuses

Definition

The quality statuses / quality codes of the signals are abbreviated at the description of the blocks.

Quality code (Page 198)

Structure

The abbreviations are as follows:

Quality code	Abbreviation/acronym
QUALITY_GOOD	GOD, GOOD
QUALITY_BAD	BAD
QUALITY_GOOD && QUALITY_UNCONNECTED	PAR
QUALITY_GOOD && QUALITY_LOCAL_OVERRIDE	FORCED
QUALITY_BAD_CONFIG_ERROR	BCE
QUALITY_BAD_COMM_FAILURE	BCF
QUALITY_BAD_NOT_CONNECTED	BNC
QUALITY_BAD_DEVICE_FAILURE	BDF
QUALITY_BAD_SENSOR_FAILURE	BSF
QUALITY_BAD_LAST_KNOWN	BLK
QUALITY_BAD_OUT_OF_SERVICE	BOS

A.3 Basics

A.3.1 Analog calculating functions

Definition

In general all analog values are embedded in floating point objects with value, time stamp and quality status. The transfer of an analog value within the system depends on the deviation of the analog value.

Analog time-independent calculating functions (Page 224)

Analog time-dependent calculating functions (Page 225)

A.3.2 Analog functions

A.3.2.1 Analog time-independent calculating functions

Definition

All time-independent calculating functions are executed without consideration of a deterministic time response. The analog output depends on the input signal, not on the time.

A.3.2.2 Analog time-dependent calculating functions

Definition

Analog computing functions are those in which the output signal not only depends on the input signals, but is also influenced by a time response.

A.3.3 Binary calculating function

Definition

In general all binary values are embedded in binary value objects with value, time stamp and quality status. The transfer of a binary value within the system depends on the change of the value.

Binary time-independent logic functions (Page 225)

Binary time-dependent logic functions (Page 225)

A.3.4 Binary functions

A.3.4.1 Binary time-independent logic functions

Definition

All time-independent operating functions are executed without consideration of a deterministic time response.

A.3.4.2 Binary time-dependent logic functions

Definition

All time-dependent binary options apply the setting and resetting of an output, depending on a configured time value and the status of a trigger input.

A.3.5 Monitoring

A.3.5.1 Signal monitoring analog

Definition

This section contains the automation function for the analog signal monitoring.

A.3.5.2 Signal monitoring binary

Definition

This section contains the automation function for the binary signal monitoring.

A.3.6 System functions

Definition

The library of system functions includes blocks for verifying quality states of signals or for assignment to faceplates.

A.3.7 S7PutGet (Black Box)

A.3.7.1 S7PutGet (Black Box)

Definition

The 'black box' term designates a remote controller with an AS-S7 automation system. A series of hardware proxies are intended to establish the communication between a CN 4100 and the black box. The black box communication supports Industrial Ethernet in singular or redundant configuration.

Note

Black box configuration

If a black box is configured on a CN 4100-RTC, the black box protocol must have been activated during the creation of the RTC. Proceed as follows if the black box protocol is not activated:

1. Activate the black box protocol.
2. Stop the runtime container.
3. Restart the runtime container.

Supported black boxes

A black box can be realized with each AS-S7 CPU which supports the AS-S7 PUT/GET protocol.

Supported black box CPU families							
31x	31xC	31xF	41x	41xH	41xF	12xy	15xy

Function principle of the communication

AS-S7 client/server connections with PUT and GET functions are used for communication with the black box. The CN 4100 system functions as a client, the black box as a server.

The communication function supports singular or redundant communication connections via Industrial Ethernet. CN 4100-RTCs support up to 10 black box connections.

The maximum amount of data that can be transferred in each direction over a connection is 12000 bytes.

The process data block that is transferred in a single request is limited by the type of S7 CPU.

Multiple requests are implemented to transfer the configured data volume.

If a BB_COMM proxy is configured with a READ_LEN or WRITE_LEN, which is set larger than the transfer volume, then it is not required to configure additional BB_COMM proxies. The system distributes additional volumes via multiple requests. The configured byte number is transferred.

Implementation

The black box for CN 4100-RTCs supports different hardware proxies (blocks).

Internal redundancy mechanism

The hardware proxies are only configured once. However, an instance of each runs in both redundant devices.

For signals of external systems at CNET: Each input hardware proxy instance in the master compares the own process data with the process data of the standby proxy. The best process data of the two are used.

For signals of CNET to the external system: Each output hardware proxy instance in the master and standby sends signals to the external system.

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

A.3.7.2 Black box | Network configuration

Black boxes can be connected via Industrial Ethernet in a singular or redundant configuration. Black boxes can be connected via:

- Fieldbus.
- Plant network

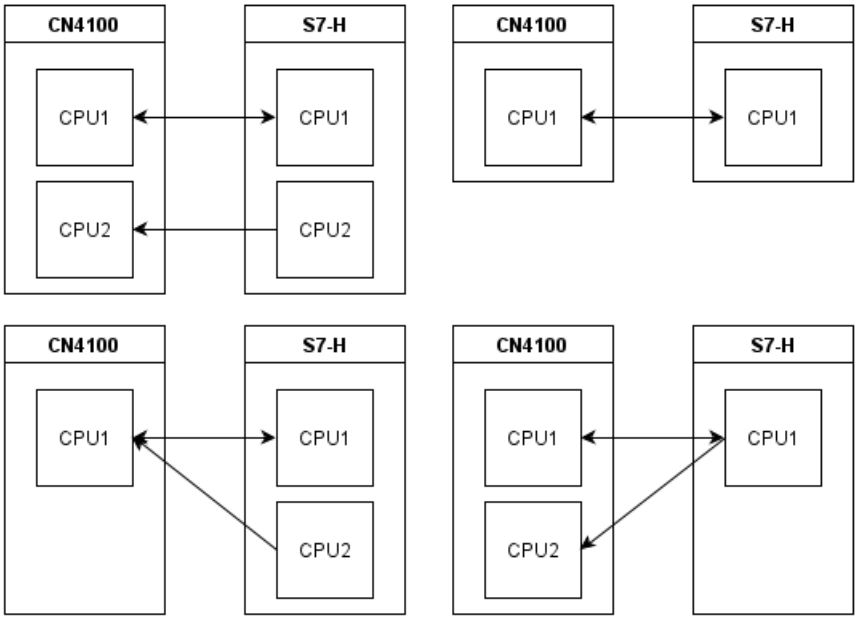
The black box must be equipped with one or two Industrial Ethernet CPs.

List of CPs:

- S7-400: CP443-1 (MLFB: 6GK7443-1EX11-0XE0 or more recent)
- S7-1200: The integrated Ethernet interface is used.

The following figure shows the possible black box configurations:

- Redundant CN 4100 as PUT/GET CLIENT with redundant S7-H as PUT/GET SERVER
- Redundant CN 4100 as PUT/GET CLIENT with singular S7 as PUT/GET SERVER
- Singular CN 4100 as PUT/GET CLIENT with redundant S7-H as PUT/GET SERVER
- Singular CN 4100 as PUT/GET CLIENT with singular S7-H as PUT/GET SERVER



A.3.7.3 Black box | Programming

The black box programming encompasses the following steps:

- Creation of DBs (data blocks).
- Reading of values from data blocks.
- Writing of values into data blocks.
- Incrementing of the sign of life counter.

Assignment of data blocks

The telegram structure is used when setting up a black box connection. The following table shows the structure of the telegrams:

Byte offset	Description	Length in bytes	Value range [HEX]
0	Sign of life counter	2	0 .. FFFF
2 : LEN-1	Data	1...n	

Data type

The data type depends on whether the data to be transferred is sent with the time stamp or quality code. If the data is sent with time stamp, 4 bytes are required. If the data are to be sent with quality code, the quality code is determined by means of the LSB value in byte. It is set to GOOD (VALID) when the LSB value is 1, or to BAD (INVALID) when the LSB value is 0. The following table specifies the length of the data package for each data type:

Description	Length in bytes	Byte offset value	Byte offset quality code	Byte offset time stamp
Single binary value	1	0.0	n.z.	n.z.
Single binary value with time stamp	5	0.0	n.z.	1.0
Group of 8 binary values	1	0.0 ... 0.7	n.z.	n.z.
Analog value (IEEE-754 32-bit)	4	0.0	n.z.	n.z.
Analog value (IEEE-754 32-bit) with time stamp	8	0.0	n.z.	4.0
Analog value (IEEE-754 32-bit) with quality code	5	0.0	4.0	n.z.
Analog value (IEEE-754 32-bit) with quality code and time stamp	9	0.0	4.0	5.0

Sign of life counter

The sign of life counter is used to check the life status of the black box. Both the send telegram as well as the receive telegram have a number in the first byte. These numbers should be incremented with each new telegram. The user must install a sign of life application.

The following examples show the two parts of the sign of life application:

Increment

The following STEP 7 example shows the incrementation of the sign of life counter in the black box:

```
NETWORK
TITLE =increment life beat counter in interface to CNET
OPN "InterfaceToCNET";
L DBW 0; //if READ_DBB = 0 => counter is in DBW 0
+ 1;
T DBW 0;
```

Check

The following STEP 7 example shows the check of the sign of life counter in the black box:

```
NETWORK
TITLE =check life beat counter in interface from CNET
OPN "InterfaceFromCNET";
L DBW 0; //if WRITE_DBB = 0 => counter is in DBW 0
L "statLifeBeatCntOld"; //static variable to save the last counter
value
==I ;
L S5T#2S; //time value should be at least 2 x cycle time of BB_COMM
SD "CNET_Timer";
A "CNET_Timer";
= "CNET_Timeout";
L DBW 0;
T "statLifeBeatCntOld"; //save the counter value
```

A.3.8 TCP | Set up the data transfer

Description

Note

TCON support

TCP/IP (or the TCON blocks) are currently not yet supported by CNET in version v1.0. In case of questions please contact the Siemens Support.

Communication blocks are used to transfer analog or digital data structures via Open TCP/IP. The following blocks are required for the data transfer between CPUs:

- At least one "TCON_COMM" connection block on which the connection parameters are configured.
  TCP | Parameter assignment of the connection block (Page 232)
- Send and receive blocks

The following table shows all send and receive blocks and their operating ranges:

Data transfer	Send block	Receive block
Analog	TCON_AO32 TCON_IO32	TCON_AI32 TCON_II32
Digital	TCON_DO128	TCON_DI128

Send and receive blocks are interconnected with the connection block:

TCP | Interconnection principle communication (Page 233)

The following rules apply to the parameter assignment of the communication paths:

TCP | Parameter assignment of the communication paths (Page 234)

The communication proceeds as follows:

TCP | Communication procedure (Page 235)

The following has to be observed for setting up redundancy

TCP | Redundancy (Page 236)

Cycle time CFC, OB

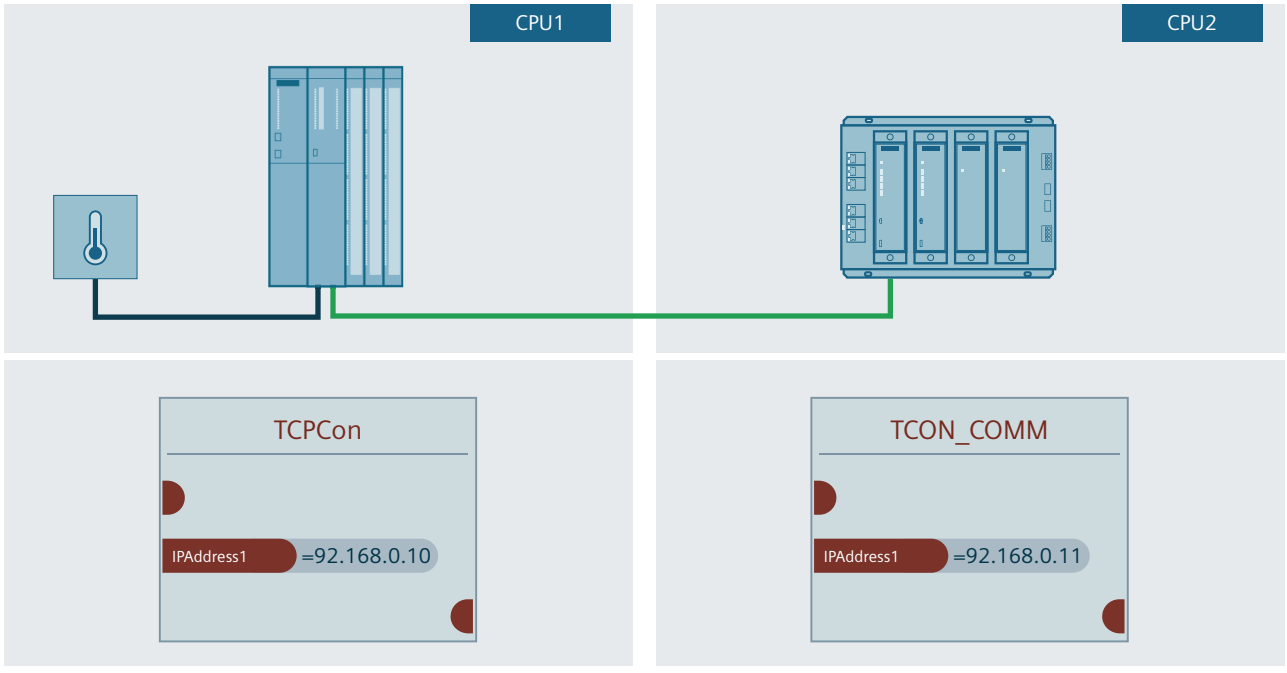
For reliable communication all CFCs within the same CPU must have the same cycle time.

A.3.9 TCP | Parameter assignment

A.3.9.1 TCP | Configuration of the connection block

Parameter assignment for Open TCP/IP

The following figure shows the parameter assignment of CPU-CPU communication using TCON_COMM as an example. The internal CPU interfaces X1 are respectively linked. Configure the respective IP address of the communication partner at the `IP_Address` parameter:



Parameter assignment with redundancy

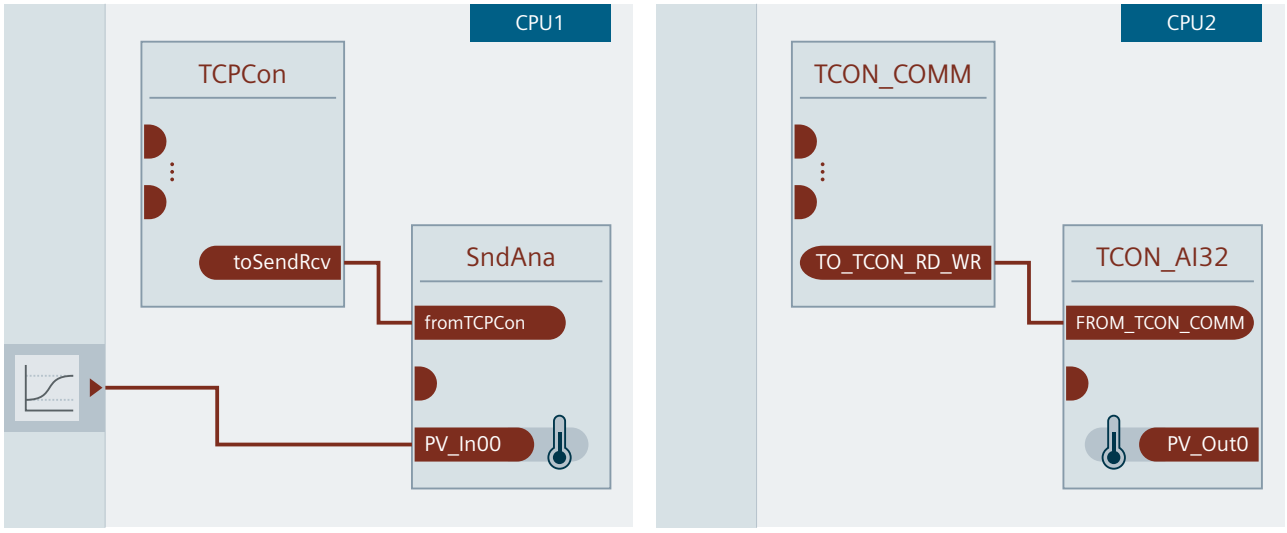
With redundantly configured connections or CPUs, parameter assignment is extended by the required parameters.

TCP | Redundancy (Page 236)

A.3.9.2 TCP | Interconnection principle for communication

Interconnection principle TCP

For the data transfer, you connect a pair of blocks with the TCON_COMM block and the value to be transferred with the send block. The following figure shows the principle of interconnection using the example of sending an analog value:



Note

Cycle time of the communication blocks

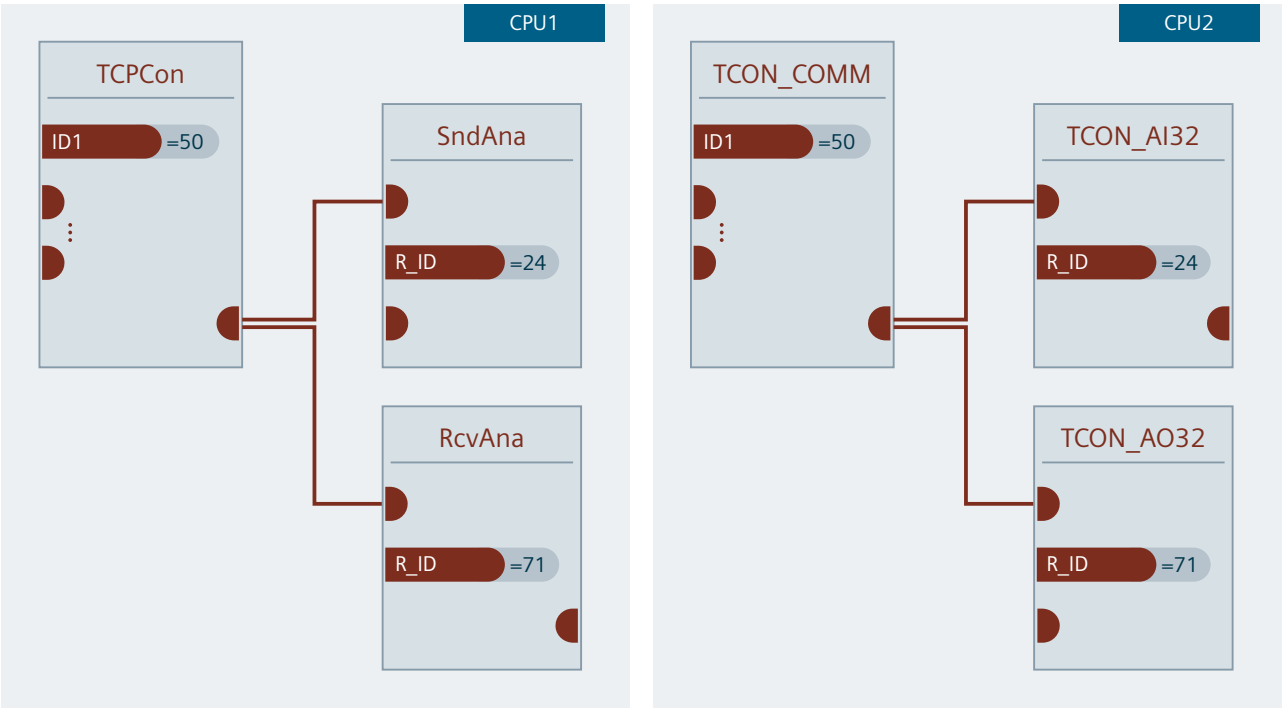
- In both controllers, the communication blocks must be executed with the same cycle time (TCON_COMM as well as send and receive blocks, e.g. in 100 ms cycle).
- Communication blocks in a fast processing cycle (e.g. 10 ms)
  It must be ensured that the time for the transfer is sufficient in the network and the time for information processing is sufficient in the controller.
  In case of a high network load, the default setting is too small for the number of cycles (`RcvMonTime` parameter).
  This can lead to a bad signal at the `PV_Outxx` output (signal status ST: 16#00=bad).
  Optimize the `RcvMonTime` parameter according to your plant situation.
  Recommendation:
  In case of a high network load, select a larger value (default setting = 12) for commissioning on the receive block for `RcvMonTime`.

A.3.9.3 TCP | Configuration of the communication paths

Parameter assignment of the communication paths

The following rules apply to the parameter assignment of the communication paths:

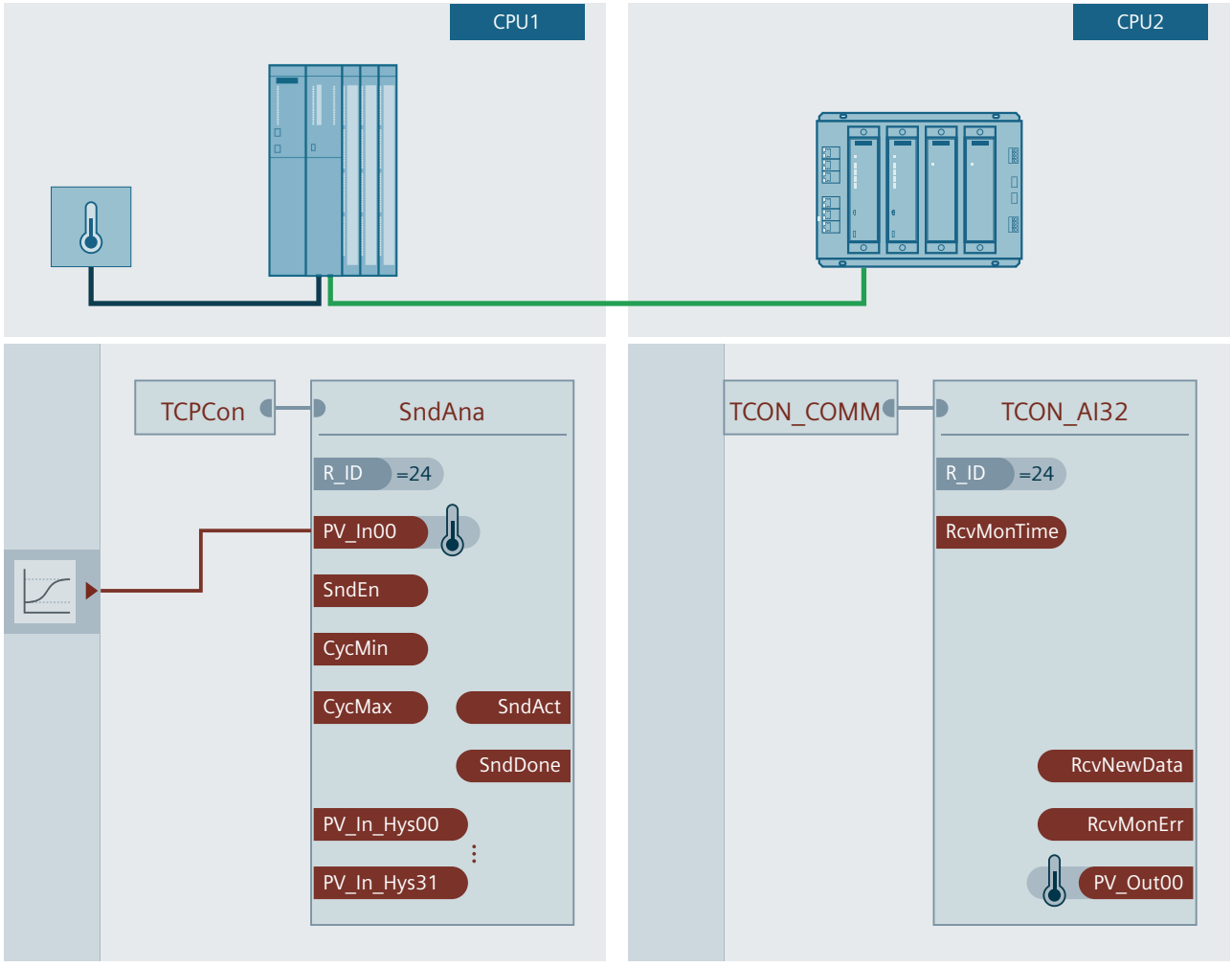
- Parameter `ID`: Must be unique within a CPU.
- Parameter `R_ID`: Must be unique for a block pair.



A.3.9.4 TCP | Communication sequence

Communication sequence

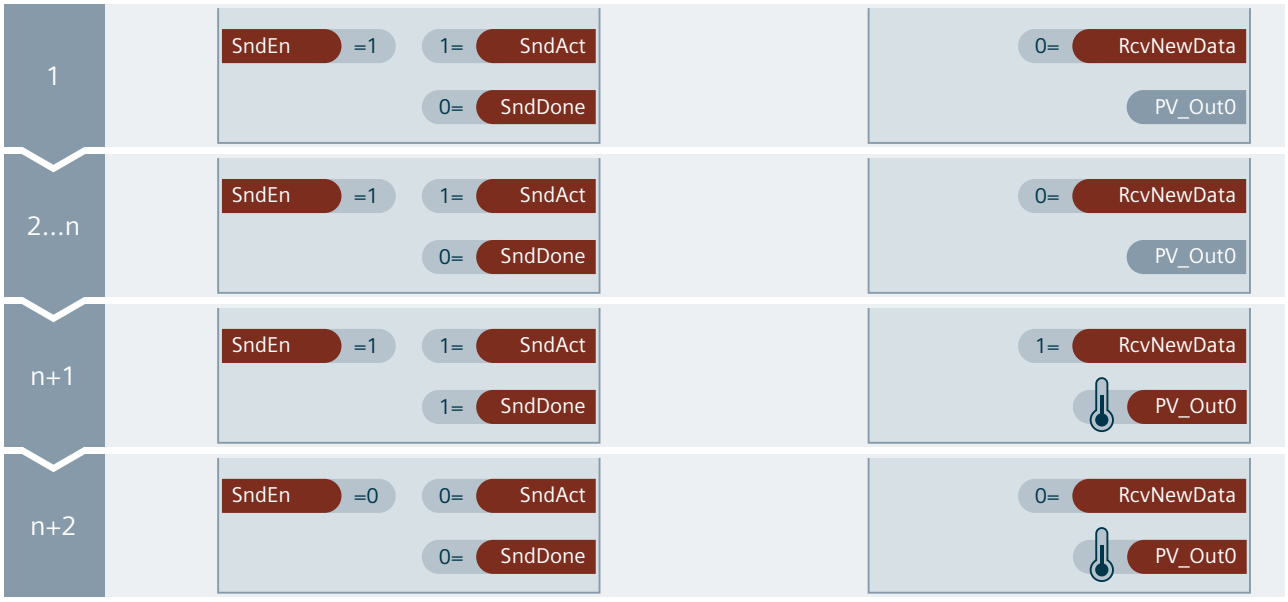
The following figure shows the communication sequence for TCON_COMM using the transfer of an analog temperature value as an example:



Data transmission is active as long as `SndEn=1`. Even if the value to be transmitted does not change, it is still sent every 10 cycles. You can control the minimum and maximum frequency via `CylMin` and `CylMax`. You can use the parameter assignment of a hysteresis `PV_In_Hys[00..31]` to suppress a value transmission in case of low fluctuations. Use the `Delta` parameter to configure a change-based value transfer.

In process mode, data transmission is started with `SndEn=1`. Data transmission is active in the next cycles. After successful data transmission, `SndDone` at the send block and `RcvNewData` at the receive block is set to 1 for one cycle.

In the next cycle, the send block automatically sends the acknowledgment to the receive block of the partner CPU.



Error case

In case of a communication error, `(RcvMonTime)` substitute values are output after the number of cycles. Bad data transmission or no data transmission are displayed via the `RcvErr` and `RcvMonErr` parameters.

Behavior in the event of an error:

- You can use one of the following strategies in case of a communication failure between the CPUs:
  - Retain last value
  - Use substitute value. You configure the substitute value on the receive block.
- The quality code of the transmitted value changes depending on the settings:

Acknowledgment

Message frames of all Send blocks must be acknowledged by the block at the receiving end. Before new message frames are sent, the message frames of all Send blocks must be acknowledged.

A.3.9.5 TCP | Configuration with redundancy

Description

To establish communication with a redundant CPU, create a "TCON_COMM" block for each connection on the send/receive side of CN 4100. (Use the "TCPCon" blocks on the side of an S7 CPU.) Interconnect the "TCON_COMM" blocks with a send/receive block that is used for both connections.

You specify the redundancy properties on the sending side at the sending block with the following parameters:

- Layout of the CPU: redundant | non-redundant: RedCPU
- Redundancy operation of the CPU: activated | deactivated: RdSysStEn

The redundancy specifications are output by the send block or mirrored at the assigned receive block with the following parameters:

- RedCPUOut
- RdSysStEnOut

A.3.10 Simulation basics

Definition

Simulation means the manipulation of a signal regardless of the actual source of the signal or logic that generates this signal. Simulation is primarily used as an aid for commissioning and service on the plant. A distinction is made between two types of simulation. If simulation is carried out directly at the field device, we refer to block-external simulation. If simulation takes place directly at the block, this is block-internal simulation.

The following block families have the function Signals to simulate:

- Communication blocks
- Logic blocks
- Monitoring blocks
- Maintenance blocks
- Counter blocks

Block-external simulation

A block receives one or more values with "Manipulated value" quality code. The technological functions of the block are not influenced. The process-relevant output signals do not obtain the "Manipulated value" status.

Block-internal simulation

The simulation is activated via the faceplate or interconnection and carried out in the block itself. Non-operable process values (e.g. PV, AV In) can also be simulated.

Behavior for technological blocks:

With technological blocks control commands are not transferred to the component. The component goes into idle/safety position. Feedback is simulated.

Behavior for control blocks:

- In manual mode, both the process value and the position feedback can be entered as simulated values in the faceplate.
- The simulated process value is tracked to the setpoint when switching to automatic mode. The control deviation therefore no longer exists, the existing manipulated variable remains constant.
- The following applies for program mode:
  - In program mode with setpoint specification, block-internal simulation responds the same as in automatic mode
  - In program mode with manipulated variable specification, block-internal simulation responds the same as in manual mode

Behavior for other blocks:

- The signal is disconnected. Process value can be changed. When the simulation is switched off, the signal is reconnected.

Simulation during runtime

Simulation at runtime takes place in the parameter view of the faceplate.

- Can only be activated with runtime authorization level system authorization
- Technological functions are not influenced
- Parameters that actively influence the process obtain the "Manipulated value" status.
- All process values displayed in the faceplate that cannot be operated in normal operation can be operated. Associated values cannot be manipulated.

Signaling of the "Simulation" state

Simulated signals are displayed in the faceplate as follows.

- Display of the quality code next to the signal in the standard view
- Status display in the parameter view
- Status display in the standard view

Substitute values

Substitute values (Page 239)

A.3.11 Simulation

A.3.11.1 Substitute values

Definition

Substitute values are output by blocks in defined states instead of the present value. Whether a substitute value is output depends on the configuration of the block.

Substitute value for defined states

Substitute values are output in the following situations.

- Simulation is active
- Interlock exclusion is active
- Error has occurred

A substitute value is marked with the "Manipulated value" quality code.

A.3.12 OPC UA

A.3.12.1 OPC UA client

Description

CNET supports the communication protocol OPC UA client.

The OPC UA client proxies (blocks) support the transfer of messages, events and data accesses of all nodes of a communication partner (Third Party System / OPC UA server) to CNET. This includes the writing to a node of an OPC UA server and the reading of values from the communication partner to CNET.

The selection of the client protocol can be seen in the configuration area of CNET. The settings of the client protocol are invisible.

The connection establishment depends on certificates:

Certificates (Page 240)

A.3.12.2 OPC UA server

Description

CNET supports the communication protocol OPC UA server.

The OPC UA server proxies (blocks) support the transmission of messages, events, and data access from all nodes of a communication partner (Third Party System / OPC UA client) to CNET. This includes writing to nodes of an OPC UA client and reading values from the communication partner to CNET.

The selection of the server protocol can be seen in the configuration area of CNET. The settings of the server protocol are invisible.

The connection establishment depends on certificates:

Certificates (Page 240)

Details

Note

Runtime inconsistency (between server and client) is possible

Editing (memory handling) of the program on the OPC UA server may exceed the set cycle time of the CN 4100 CPU. The runtimes of the program from the server and client are then inconsistent.

In this case, the following message is displayed in CNET:

- "OPC UA Server library degrades realtime capabilities"

The message is for informational purposes only. Compared to the usual synchronous program processing, the CN 4100 CPU operates asynchronously in the mentioned scenario*. *(The CN 4100 CPU individually uses a longer CPU cycle.)

- "Runtime environment = Non Realtime"

A.3.12.3 OPC UA Client, OPC UA Server | Certificates

Description

Server certificates are either accepted and are then located in the "/config/OpcUaClient/PKI/CA/" folder or they are signed by a certificate of the folder.

Trustworthy certificates

When the OPC UA client (CNET) connects with the OPC UA server for the first time, the certificates are automatically moved into the "Rejected" list of the "OPC UA Certificate" area. In this case the user has to manually move the certificates in CNET into the "Trusted" list.

A connection with the OPC UA server can only be established when the certificates are trustworthy and are located in CNET in the "Trusted" list.

Note

OPC UA certificate for all connections

CN 4100 uses only one certificate for all configured OPC UA connections.

Note

Certificates are renewable

OPC UA certificates are valid for 10 years as standard. If a certificate expires, you can renew it as follows:

1. Delete the certificate from the list.
2. Re-add the certificate and activate it.

Note

Time synchronization required

Without time synchronization between the communication partners of an OPC UA connection, the following problems may occur:

- Error when writing values
- Incorrect interpretation of certificate validity

Use NTP servers for time synchronization for all systems involved in the OPC UA environment.

A.3.13 Modbus

A.3.13.1 Modbus

Definition

Modbus blocks are hardware proxies and automation functions which enable the transfer of messages between Modbus field devices.

Details

Note

Consider security

Because the Modbus protocol can be modified/falsified, it is not considered a safe protocol from a security perspective. If possible, use the "OPC UA" protocol instead of Modbus.

Modbus | application server and CN 4100 Modbus (Page 242)

See also

Modbus | Register and BYTE swap (Page 242)

A.3.13.2 Modbus | application server and CN 4100 Modbus

CNET supports communication with devices using the Modbus TCP or Modbus COM (Serial) protocol variant via connections to CN 4100.

Modbus TCP

You can find an example configuration of a direct Ethernet connection to CN 4100 in the CN 4100 Equipment Manual under "Plant engineering > SIMATIC CN 4100 communication system configuration".

Redundancy

In a redundant configuration the system automatically determines which signals are used:

- The signals of the runtime container (either master or standby) with the best connection status are used.
- If the connection statuses are the same, the signals of the runtime container (either master or standby) with the best connection status to the station are used.
- If the connection statuses to the station are the same, the signal with the best signal status is used for each individual signal (either from master or standby runtime container).

The signals of the masters are used when all parameters mentioned above are the same for both runtime containers (master and standby).

Note

Connection loss

If the connection is lost on the master side, the signals of the standby runtime container are only used when the data traffic timeout has expired.

A.3.13.3 Modbus | Register and BYTE swap

Definition

With the Modbus protocol, all data between the communication partners is transferred via "Register". Each register has a size of 2 bytes.

Changing the order (BYTE-Swap)

The order of the data to be transferred can be changed as follows:

- Changing the byte order in the register.
  Example:
  - Before: Register 1 (with Byte 1, Byte 2) = `01 02`
  - After: In register = `02 01`
- Changing the register order.
  Example:
  - Before: Register 1 and 2: `01 02`, `03 04`
  - After: `03 04`, `01 02`
- Changing the byte and register order.
  Example:
  - Before: Register 1 and 2: `01 02`, `03 04`
  - After: `04 03`, `02 01`

A.3.13.4 Modbus | baud rate

Definition

The baud rate (step rate) indicates the number of signal states transmitted per second. Baud rates from 75 to 115200 can be set here. The baud rate must be uniformly set for both the sender and the receiver.

A.3.13.5 Modbus | data bits

Definition

For Modbus (MODBUS_COM) specify the data bits per character (7 or 8).

A.3.13.6 Modbus | stop bits

Definition

Stop bits complete the transmission of a character in an asynchronous serial line. The "Stop bits" entry indicates the duration of stop bits relative to the transmission time of an information bit.

Specify here whether 1 or 2 bits are used to complete the transmission.

A.3.13.7 Modbus | parity

Definition

Parity is used to check for errors in the transmitted bits. This allows detection of a sequence of incorrectly transmitted bits.

The parity must be uniformly set for both the sender and the receiver.

Possible values:

- Even = The number of ones, including the parity bit, is even.
- Odd = The parity bit is set so that the number of ones (including the parity bit) is odd.
- None = Parity checking should not be used.

A.3.13.8 Modbus | handshake protocol

Definition

The following protocol procedures (handshake protocols) define the receive/transmit readiness of Modbus devices:

- Hardware handshake:
  Synchronization is carried out via feature bits.
- Software handshake:
  Synchronization is carried out via the transmission of control characters.
  XON/XOFF are the most common characters used for the software handshake.

Hardware handshake

Hardware handshake consists of 2 functions:

- Request to Send ("RTS"): Request to send, RS-232-C signal.
- Clear to Send ("CTS"): Clear to send, RS-232-C signal.

The hardware handshake can also be deactivated.

Software handshake

ASCII characters are included in the communication stream by the communication partners. These control codes in the data stream signal whether the data flow is possible or not:

- XON: Data receipt is possible
- XOFF: Data receipt is not possible

The software handshake can also be deactivated.

A.3.13.9 Modbus | mode

Definition

Modbus has the following modes for information exchange:

- Modbus RTU
  This mode uses binary data for information exchange, transmitted in hexadecimal formats, along with CRC error checking.
- Modbus ASCII
  This mode is based on the less efficient but more readable transmission of ASCII characters.

A.4 ACalc4cnet | Analog time-independent calculating function library

A.4.1 ABS (ACalc4cnet) | Description

Description

The absolute value of the analog signal IN is calculated and the result is provided at the output OUT.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Input value	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

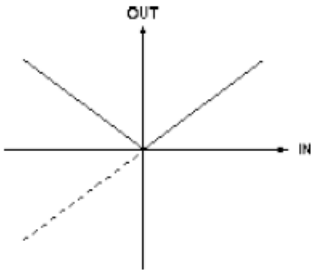
Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Absolute value			x	x	x	x

Functional description and principle of operation

The value of OUT is always the positive value of the analog signal IN:

- If IN >= 0, then OUT is equal IN.
- If IN < 0, then OUT is equal -IN.

Graphical representation:



Processing of exceptions

If the QUALITY input is set to TRUE, the quality status of input IN is transferred to the output OUT. If the QUALITY input is set to FALSE, the quality status of the output OUT is set to GOD. In case of simulation (ports simulated) the quality status is set to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.2 ADD (ACalc4cnet) | Description

Description

The addition function executes the arithmetic addition of two analog values and provides the result (sum) at the output.

Analog time-independent calculating functions (Page 224)

Parameter

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	FLOAT	Summand 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN2	FLOAT	Summand 2	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	TRUE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Sum (summation)			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The output value (summation) is calculated according to the following formula:

OUT = IN1 + IN2

Processing of exceptions

If the limit values of the output port OUT is exceeded/fall below (upper limit value UL or lower limit value LL), the value of OUT is limited to UL or LL and the digital output TRBL_AL is set to TRUE. If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see following table). If the QUALITY input is set to TRUE and the quality statuses of the inputs are not GOD, the worst quality status is transferred to the output OUT. In case of simulation (parameters simulated) the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	QUALITY		
OUT > UL	Irrelevant	OUT = UL; TRBL_AL = TRUE	OUT = BAD
OUT < LL		OUT = LL; TRBL_AL = TRUE	OUT = BAD
Any input = not GOD	FALSE	-	OUT = GOD
Any input = not GOD	TRUE	-	OUT = worst quality status of the inputs

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.3 ASIG (ACalc4cnet) | Description

Description

This function generates a constant analog value.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	FLOAT	Input signal constant	1.0	3.4028235E38	-3.4028235E38		x	x	x

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	

Functional description and principle of operation

The input can only be configured but not connected. The output is set to the analog value configured in the input.

Processing of exceptions

Not applicable

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.4 AVER (ACalc4cnet) | Description

Description

The mean value generation calculates the mean value of up to four analog signals. Signals with the quality status BAD are excluded from the mean value calculation.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	FLOAT	Input signal 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN2	FLOAT	Input signal 2	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN3	FLOAT	Input signal 3	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN4	FLOAT	Input signal 4	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
TRBL_AL	BOOL	Fault message	x /LEAD	Came in / went out	x	x	x	x
Q_AL	BOOL	Message signal quality		Came in / went out	x	x	x	x

Functional description and principle of operation

The algorithm of the mean value function always calculates the mean value depending on the quality statuses. An analog signal is only considered for calculation if its quality status is not BAD.

If the quality status of a signal is BAD, the algorithm excludes the respective point from the mean value calculation. In this case the output port Q_AL (quality message) is also set to TRUE.

Note

If an input is not to be used for the calculation of the mean value, meaning an input is "not connected", the algorithm also excludes the respective input from the mean value calculation.

Examples for the calculation of the mean value are displayed in the following table:

Example	Mean value calculation	Conditions
1	OUT = (IN1 + IN2 + IN3 + IN4)/4	• Quality status of all inputs = not BAD • All inputs are CONNECTED
2	OUT = (IN1 + IN2 + IN4)/3	• QUALITY is TRUE • Quality status of the inputs IN1, IN2, IN4 = not BAD • Quality status of the input IN3 = BAD • All inputs are CONNECTED
3	OUT = (IN1 + IN2)/2	• QUALITY is TRUE • Quality status of the inputs IN1, IN2 = not BAD • Quality status of the input IN3 = BAD • Input IN4 is NOT CONNECTED
4	OUT = (IN1 + IN2 + IN3 + IN4)/4	• QUALITY is FALSE • Quality status of the inputs IN1, IN2, IN4 = not BAD • Quality status of the input IN3 = BAD • All inputs are CONNECTED
5	OUT = (IN1 + IN2 + IN4)/3	• QUALITY is FALSE • Quality status of the inputs IN1, IN2 = not BAD • Quality status of the input IN4 = BAD • Input IN3 is NOT CONNECTED

Processing of exceptions

If the QUALITY input is set to TRUE, the worst quality status of the involved inputs is transferred to the output OUT.

If the quality statuses of all connected inputs are BAD, the quality status of OUT is also BAD. In this case output OUT keeps its last value and the output TRBL_AL is set to TRUE.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see following table). In case of simulation (parameters simulated) the quality status is set to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	QUALITY		
All connected inputs BAD	Irrelevant	OUT = OUT alt; TRBL_AL = TRUE	OUT = BAD
IN = not GOD		Q_AL = TRUE	-

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	QUALITY		
Any input = not GOD	FALSE	-	OUT = GOD
Any input = not GOD	TRUE	-	OUT = worst quality status of the inputs

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.5 AXFR (ACalc4cnet) | Description

Description

Transfers one of two analog input values to an analog output.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
NO	FLOAT	Analog input variable 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	
YES	FLOAT	Analog input variable 2	0.0	3.4028235E38	-3.4028235E38	x	x	x	
SEL	FLOAT	Preselection	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	

Functional description and principle of operation

The value of OUT is always equal to one of the two analog input values, depending on the value of SEL:

- If SEL = FALSE, then the value of OUT corresponds to the input NO
- If SEL = TRUE, then the value of OUT corresponds to the input YES

Processing of exceptions

If the input QUALITY is set to TRUE, the quality status of input IN is transferred to the output OUT. If the QUALITY input is set to FALSE, the quality status of output OUT is set to GOD.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.6 COMP (ACalc4cnet) | Description

Description

This function compares two analog input signals. The results less than, greater than or equal to are implemented as digital outputs.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	FLOAT	Analog input signal 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN2	FLOAT	Analog input signal 2	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
Q_G	BOOL	IN1 greater than IN2			x	x		
Q_E	BOOL	IN1 equal to IN2			x	x		
Q_L	BOOL	IN1 less than IN2			x	x		

Functional description and principle of operation

The analog signal in IN1 is compared with IN2. The digital outputs are set as follows:

Condition	Q_G	Q_E	Q_L
IN1 > IN2	TRUE	FALSE	FALSE
IN1 = IN2	FALSE	TRUE	
IN1 < IN2		FALSE	TRUE

Processing of exceptions

If the QUALITY input is set to TRUE, the worst quality status of the inputs is always transferred to all binary outputs. If the QUALITY input is set to FALSE, the digital outputs are always set to the quality status GOD.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.7 CONV (ACalc4cnet) | Description

Description

The scaling function scales an analog signal from a specified value range for a different value range. The output value is a linear function of the input value and is determined by 2 coordinate pairs.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Analog input variable	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
X1	FLOAT	X-coordinate breakpoint 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	
Y1	FLOAT	Y-coordinate breakpoint 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	
EU_IN	STRING	Engineering units of the input		20	0		x	x	
X2	FLOAT	X-coordinate breakpoint 2	100	3.4028235E38	-3.4028235E38	x	x		

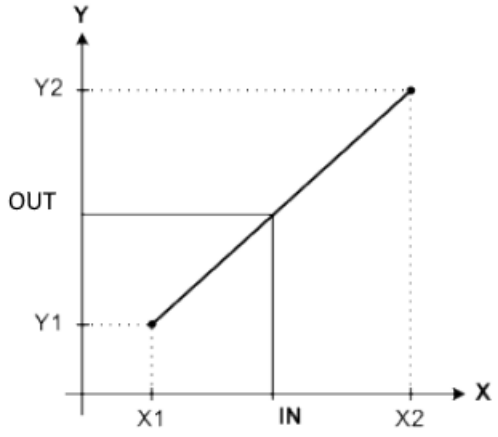
Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
Y2	FLOAT	Y-coordinate breakpoint 2	100	3.4028235E38	-3.4028235E38	x	x	x	
EU_OUT	STRING	Output engineering unit		20	0		x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The output signal OUT is a function of the input signal IN and is determined through the coordinate pairs X1/Y1 and X2/Y2:



Processing of exceptions

Normally the quality status of QUALITY is transferred to the output IN if the input OUT is set to TRUE.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see following table).

Independently of the value of QUALITY, the quality status of OUT is set to INVALID, the output OUT to TRUE and TRBL_AL is limited to UL or LL, if the value of OUT exceeds the range of the float value (upper limit value UL or lower limit value LL of the float variables) during the calculation.

In case of simulation (ports simulated) the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	QUALITY		
Overflow during the calculation	Irrelevant	OUT=UL or LL; TRBL_AL = TRUE	OUT = INVALID
Any input = not GOD	FALSE	-	OUT = GOD
Any input = not GOD	TRUE	-	OUT = worst quality status of the inputs

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.8 DB (ACalc4cnet) | Description

Description

The output of the deadband function is zero if the absolute value of the input signal is less than or equal to the specified deadband.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Input signal	0.0	3.4028235E38	-3.4028235E38	x	x	x	
DB	FLOAT	Deadband	0.0	3.4028235E38	0.0	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

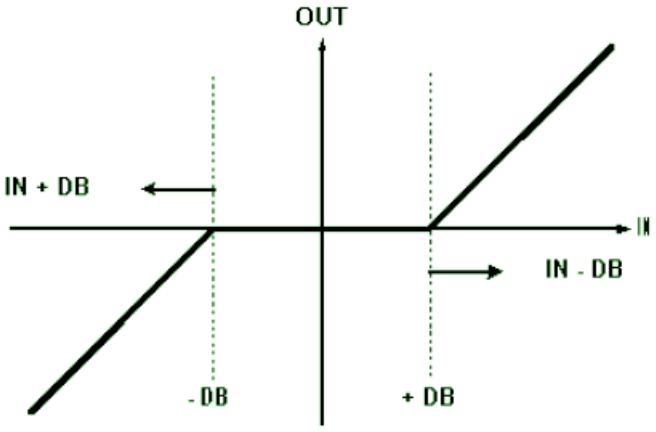
Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Output value			x	x		
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The output value OUT is calculated according to the following rules (additional constraint DB >= 0.0):

- If the absolute value of the input signal IN is less than equal to the deadband DB, the output OUT is zero.
- If IN is less than the negative DB, OUT is equal to IN plus DB.
- If IN is greater than DB, OUT is equal to IN minus DB.

Graphical representation:



Processing of exceptions

If the value of DB is less than zero, OUT is equal to IN. Normally OUT has the quality status GOD. If the QUALITY input is set to TRUE and the quality status of IN or DB is not GOD, the worst quality status is transferred to the output OUT. If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see the following table). If the value of OUT exceeds the range of the float value (upper limit value UL or lower limit value LL of the float variables) during the calculation, the quality status of OUT is set to BAD, the fault message output TRBL_AL is set to TRUE, and OUT is limited to UL or LL. In case of simulation (parameters simulated) the quality status of OUT is always changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		**Effect on further processing**	**Effect on the quality code**
Input/output	**QUALITY**		
DB < 0.0	Irrelevant	OUT = IN	-
Overflow during the calculation		TRBL_AL = TRUE	OUT = BAD
IN or DB = not GOD	FALSE	-	OUT = GOD
IN or DB = not GOD	TRUE	-	OUT = worst quality status of the inputs IN and DB

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.9 DIV (ACalc4cnet) | Description

Description

The division function executes arithmetic division of two input values and provides the result (quotient) in the output.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Dividend	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN2	FLOAT	Divisor	1.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Quotient			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x
DIV0	BOOL	Division by 0						x

Functional description and principle of operation

The output value (quotient) is calculated according to the following formula:

OUT = IN1/IN2

Processing of exceptions

If the limit values of the output parameter OUT are exceeded/fall below (upper limit value UL or lower limit value LL), the value of OUT is limited to UL or LL and the digital output port TRBL_AL is set to TRUE. If IN1 is a positive value with sign and the value of IN2 is zero (prohibited division by zero), the digital outputs TRBL_AL and DIV0 are set to TRUE and OUT receives the value UL. If IN1 is a negative value with sign and the value of IN2 is zero (prohibited division by zero), the digital outputs TRBL_AL and DIV0 are set to TRUE and OUT receives the value LL. If both input values are zero, the digital outputs TRBL_AL and DIV0 are set to TRUE and OUT receives the value zero. If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see following table). If the QUALITY input is set to TRUE and the quality status of any input is not GOD, the worst quality status is transferred to the output OUT. In case of simulation (parameters simulated) the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		**Effect on further processing**	**Effect on the quality code**
Input/output	**QUALITY**		
OUT > UL	Irrelevant	OUT = UL; TRBL_AL = TRUE	OUT = BAD
OUT < LL		OUT = LL; TRBL_AL = TRUE	OUT = BAD
> 0.0; IN2 = 0.0		OUT = UL; TRBL_AL = TRUE; DIV0 = TRUE	OUT = BAD
IN1 < 0.0; IN2 = 0.0		OUT = LL; TRBL_AL = TRUE; DIV0 = TRUE	OUT = BAD
IN1 = IN2 = 0.0		OUT = 0.0; TRBL_AL = TRUE; DIV0 = TRUE	OUT = BAD
Any input = not GOD	FALSE	-	OUT = GOD
Any input = not GOD	TRUE	-	OUT = worst quality status of the inputs

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.10 EXP (ACalc4cnet) | Description

Description

The exponential function calculates the number (base e) of an analog value and provides the result in the output.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Exponent	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Quotient			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The output value is calculated according to the following formula

$OUT = e^{IN}$, e = 2.718281

Processing of exceptions

If the upper limit value of the output OUT is exceeded (upper limit value UL) the value of OUT is limited to UL and the digital output TRBL_AL is set to TRUE. If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see the following table). If the QUALITY input is set to TRUE and the quality status of IN or DB is not GOD, the quality status of IN is transferred to the output OUT. In case of simulation (parameters simulated) the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	QUALITY		
OUT > UL	Irrelevant	OUT = UL; TRBL_AL = TRUE	OUT = BAD
IN = not GOD	FALSE	-	OUT = GOD
IN = not GOD	TRUE	-	OUT has the quality status of IN

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.11 LN (ACalc4cnet) | Description

Description

This function calculates the natural logarithm (log to base e) of an analog value and provides the result in the output.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Argument	1.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Natural logarithm			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The output value is calculated according to the following formula:

OUT = ln (IN)

Processing of exceptions

If IN is less than or equal to zero, the value of OUT is set to zero and the digital output TRBL_AL is set to TRUE.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see the following table). If the QUALITY input is set to TRUE and the quality statuses of the inputs are not GOD, the worst quality status is transferred to the output OUT.

In case of simulation (parameters simulated) the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	QUALITY		
IN <= 0.0	Irrelevant	OUT = 0.0; TRBL_AL = TRUE	OUT = BAD
IN = not GOD	FALSE	-	OUT = GOD
IN = not GOD	TRUE	-	OUT has the quality status of IN

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.12 MAX (ACalc4cnet) | Description

Description

The maximum value selection compares two analog values and makes the higher value available in the output MAX.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	FLOAT	Input signal 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN2	FLOAT	Input signal 2	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
MAX	FLOAT	Maximum value			x	x	x	
MAX1	BOOL	IN1 is the maximum value			x	x		
MAX2	BOOL	IN2 is the maximum value			x	x		

Functional description and principle of operation

The output value MAX is calculated according to the following formula:

- If IN1 >= IN2 then MAX is equal to IN1.
- If IN1 < IN2 then MAX is equal to IN2.

Depending on the selection of the maximum input value, the respective binary output MAXn is set to TRUE. If IN1 is the maximum value, MAX1 is set to TRUE. If IN2 is the maximum value, MAX2 is set to TRUE.

Condition	Value of MAX	Value of MAX1	Value of MAX2
IN1 >= IN2	IN1	TRUE	FALSE
IN2 > IN1	IN2	FALSE	TRUE

Processing of exceptions

If the QUALITY input is set to TRUE, the quality status of the maximum input value is transferred to the output MAX. If the QUALITY input is set to FALSE, the quality status of MAX is set to GOD.

In case of simulation (parameters simulated) the quality status of MAX is changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.13 MIN (ACalc4cnet) | Description

Description

The minimum value compares two analog values and makes the minimum value available in the output MIN.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parame-ter	Data type	Description	Default	Maximum value	Minimum value	Con n	Edit	L	HMI
IN1	FLOAT	Input signal 1	0.0	3.4028235E 38	-3.4028235E 38	x	x	x	
IN2	FLOAT	Input signal 2	0.0	3.4028235E 38	-3.4028235E 38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
MIN	FLOAT	Minimum value			x	x	x	
MIN1	BOOL	IN1 is the minimum value			x	x		
MIN2	BOOL	IN2 is the minimum value			x	x		

Functional description and principle of operation

The output value MIN is calculated according to the following formula:

- If IN1 <= IN2 then MIN is equal to IN1.
- If IN1 > IN2 then MIN is equal to IN2.

Depending on the selection of the minimum input value, the respective binary output MINn is set to TRUE. If IN1 is the minimum value, MIN1 is set to TRUE. If IN2 is the minimum value, MIN2 is set to TRUE.

Condition	Value of MIN	Value of MIN1	Value of MIN2
IN1 <= IN2	IN1	TRUE	FALSE
IN1 > IN2	IN2	FALSE	TRUE

Processing of exceptions

If the QUALITY input is set to TRUE, the quality status of the minimum input value is transferred to the output MIN. If the QUALITY input is set to FALSE, the quality status of MIN is set to GOD.

In case of simulation (parameters simulated) the quality status of MIN is changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.14 MINMAX (ACalc4cnet) | Description

Description

The MINMAX limiting function is used to limit an analog signal to a specified value range. The activation of the limiting function is displayed through the digital outputs.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Input signal	0.0	3.4028235E38	-3.4028235E38	x	x	x	
MAX	FLOAT	High limit	100.0	3.4028235E38	-3.4028235E38	x	x	x	
MIN	FLOAT	Low limit	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

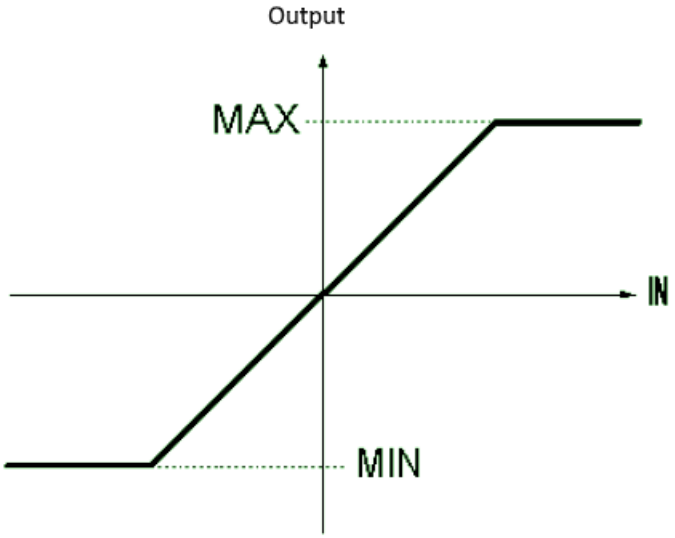
Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output variable			x	x	x	
OUT_MAX	BOOL	Output value at upper limit			x	x		
OUT_MIN	BOOL	Output value at lower limit			x	x		

Functional description and principle of operation

The output value is calculated according to the following formula:

Condition	Value of OUT	Value of OUT_MIN	Value of OUT_MAX
MIN < IN < MAX	IN	FALSE	FALSE
IN <= MIN	MIN	TRUE	FALSE
IN >= MAX	MAX	FALSE	TRUE

Graphical representation:



Processing of exceptions

If the lower limit value MIN lies above the upper limit value MAX, the output value OUT = MAX. In this case the two outputs OUT_MIN and OUT_MAX are set to TRUE.

If the QUALITY input is set to TRUE, the worst quality status of the inputs IN, MIN and MAX is transferred to the output OUT. If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see following table).

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	**QUALITY**		
MIN > MAX	Irrelevant	OUT = MAX;OUT_MIN = OUT_MAX = TRUE	-
Any input = not GOD	FALSE	-	OUT = GOD
Any input = not GOD	TRUE	-	OUT = worst quality status of the inputs

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.15 MONIT (ACalc4cnet) | Description

Description

The limit value monitoring function checks an analog signal with regard to a predefined limit value. Exceedances of the upper and lower limit value are displayed in the limit signal of the output parameter. A deadband can be assigned to the limit value.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Analog input variable	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV	FLOAT	Limit	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
DB	FLOAT	Deadband (hysteresis)	0.0	3.4028235E38	0.0	x	x	x	x
HI_LOW	BOOL	Upper/lower limit Default = True True = High False = Low	TRUE	TRUE	FALSE	x	x	x	x
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
Q	BOOL	Limit monitor Q			x	x	x	
Q_N	BOOL	Negated limit monitor			x	x		
QA	BOOL	Message signal quality			x	x	x	x

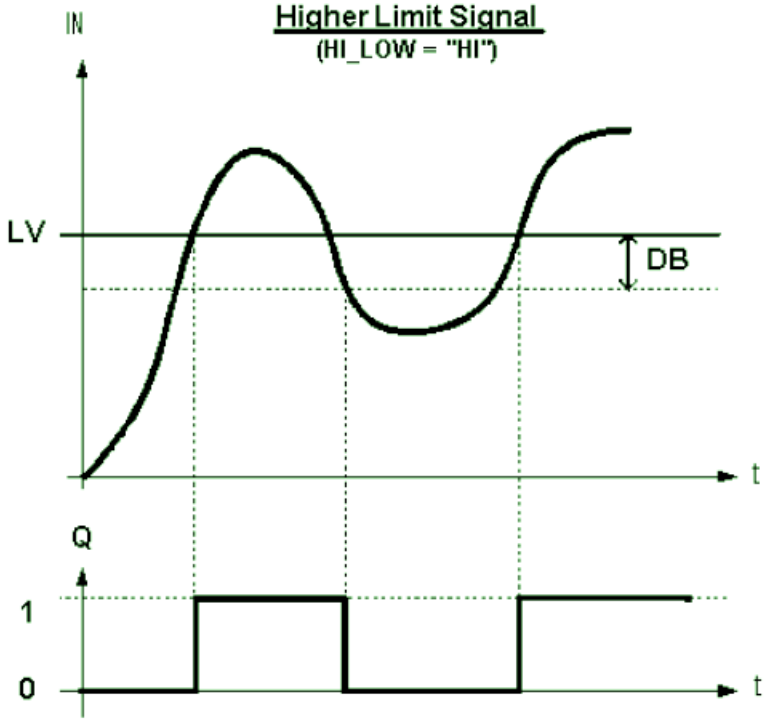
Functional description and principle of operation

The limit value monitoring function shows whether a configured limit value LV has been exceeded. The limit value signal is available at the output parameter Q. The trigger direction (upper or lower limit value) can be set by means of the HI_LOW parameter. Triggering or resetting the limit monitor, is also influenced by the deadband (physical absolute value). The differences in processing of the cases UPPER LIMIT and LOWER LIMIT are explained below:

Monitoring of the UPPER LIMIT VALUE (HI_LOW = TRUE)

- **Trigger condition:**
  If the analog value IN is greater than or equal to LV, meaning IN >= LV, the output port Q is set to TRUE and Q_N is set to FALSE.
- **Reset condition:**
  If the analog value IN is less than or equal to the value LV minus deadband DB, meaning IN < (LV - DB), the output port Q is set to FALSE and Q_N is set to TRUE.

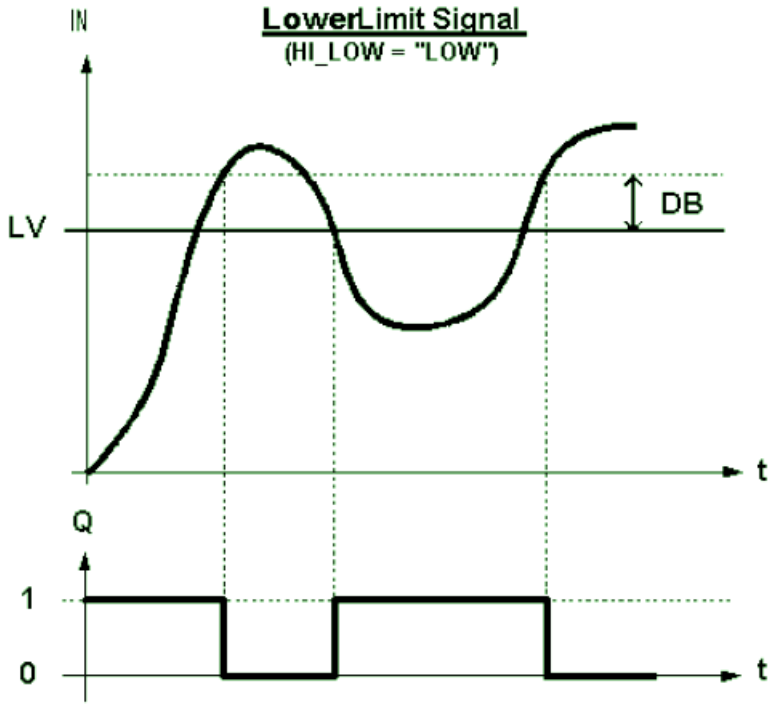
Example:



Monitoring of the LOWER LIMIT VALUE (HI_LOW = FALSE)

- **Trigger condition:**
  If the analog value IN is less than or equal to LV, meaning IN <= LV, the output port Q is set to TRUE and Q_N is set to FALSE.
- **Reset condition:**
  If the analog value IN is greater than LV plus deadband DB, meaning IN > (LV + DB), the output port Q is set to FALSE and Q_N is set to TRUE.

Example:



Processing of exceptions

If the QUALITY input is set to TRUE, the quality statuses of the output Q and Q_N are equal to the quality status of IN. If the QUALITY input is set to FALSE, the quality statuses of Q and Q_N are GOD.

In case of simulation (parameters simulated) the quality status is set to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	**QUALITY**		
IN = not GOD	Irrelevant	QA = TRUE	-
IN = not GOD	FALSE	-	Q=GOD; Q_N=GOD
IN = not GOD	TRUE	-	Q and Q_N have the quality status of IN

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.16 MULTIPLY (ACalc4cnet) | Description

Description

The multiplication function executes arithmetic multiplication of two input values and provides the result (product) in the output.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	FLOAT	Multiplicand	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN2	FLOAT	Multiplicator	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Product			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The output value (product) is calculated according to the following formula:

OT = IN1 * IN2

Processing of exceptions

If the limit values of the output parameter OUT are exceeded/fall below (upper limit value UL or lower limit value LL), the value of OUT is limited to UL or LL and the digital output TRBL_AL is set to TRUE.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see the following table). If the QUALITY input is set to TRUE and the quality statuses of the inputs are not GOD, the worst quality status is transferred to the output OUT. In case of simulation (parameters simulated) the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	**QUALITY**		
OUT > UL	Irrelevant	OUT = UL; TRBL_AL = TRUE	OUT = BAD
OUT < LL		OUT = LL; TRBL_AL = TRUE	OUT = BAD
Any input = not GOD	FALSE	-	OUT = GOD
Any input = not GOD	TRUE	-	OUT = worst quality status of the inputs

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.17 NROOT (ACalc4cnet) | Description

Description

This function calculates the n-th root of an analog value and provides the result in the output. The square root function is set as the default.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Radicand	1.0	3.4028235E38	-3.4028235E38	x	x	x	
N	FLOAT	Root	2.0	3.4028235E38	2.0		x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The output value is calculated according to the following formula:

OUT = N√IN

Processing of exceptions

If IN is less than zero, the value of OUT is set to zero and the digital output TRBL_AL is set to TRUE.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see the following table). If the QUALITY input is set to TRUE and the quality status of IN or DB is not GOD, the quality status is transferred to the output OUT. In case of simulation (parameters simulated) the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		**Effect on further processing**	**Effect on the quality code**
Input/output	**QUALITY**		
IN < 0.0	Irrelevant	OUT = 0.0; TRBL_AL = TRUE	OUT = BAD
IN = not GOD	FALSE	-	OUT = GOD
IN = not GOD	TRUE	-	OUT has the quality status of IN

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.18 POLY (ACalc4cnet) | Description

Description

The polygon function creates a curve of up to 20 breakpoint coordinate pairs. The function performs an interpolation between two breakpoints.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Analog input value	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
N_BP	INT	Number of interpolation points	10	20	2		x	x	
X1 ... X20	FLOAT	X-coordinate breakpoint	0.0	3.4028235E38	-3.4028235E38	x	x	x	
Y1 ... Y20	FLOAT	Y-coordinate breakpoint	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The polygon function is used for approximation of any analog function using any number of intervals. The latter are defined through their X/Y coordinates, whereby X is the analog input variable and Y the analog output variable. Within the approximation limits up to 20 X/Y coordinate pairs can be specified. The number of X/Y pairs must be configured via the input N_BP.

$$OUT = Y_n + (IN - X_n) * \left(\frac{Y_{n+1} - Y_n}{X_{n+1} - X_n} \right)$$

Constraints:

- $1 <= n <= 19$
- $X_n < X_{n+1}$

where:

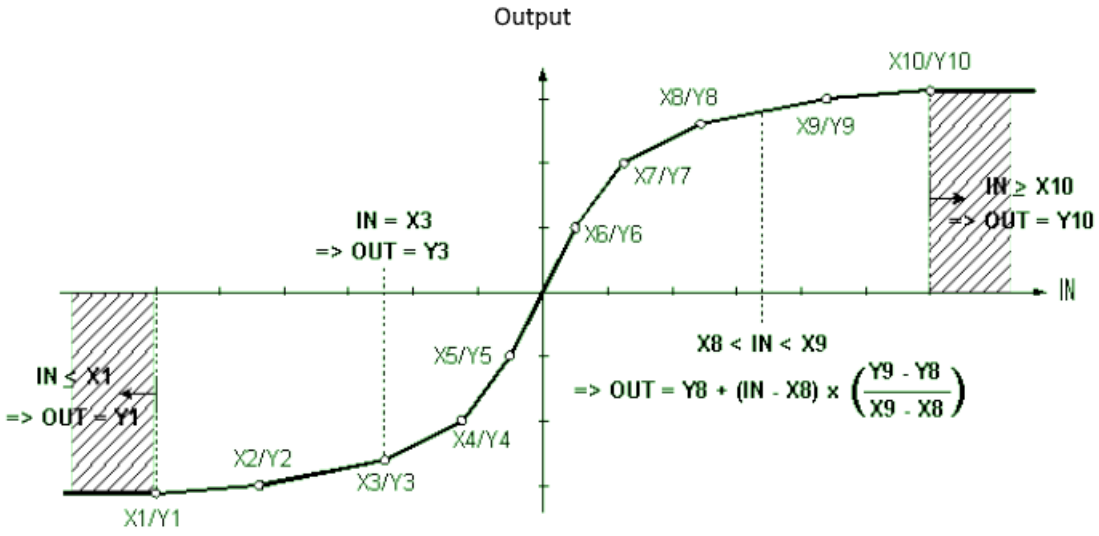
- Interpolation point n: Y_n / X_n
- Interpolation point n+1: Y_{n+1} / X_{n+1}

If the input value IN is less than the smallest X reference value, the output OUT is set to the value of the respective Y-coordinate. If the input value is greater than the largest Y reference value, the output OUT is set to the value of the respective Y-coordinate. The X-values must be entered in ascending order, meaning Xn < Xn+1. If because of a parameter assignment error Xn >= Xn+1, then OUT is set to Yn.

The conditions for the calculation of the output value OUT are reproduced in the following table:

Case	Conditions	Output value
1	Xn < IN < Xn+1	#PIC
2	N_BP = n; IN <= X1	OUT = Y1
3	N_BP = n; IN >= Xn	OUT = Yn
4	N_BP > n; IN >= Xn; Xn+1 <= Xn	OUT = Yn

Example with N_BP = 10 (number of breakpoints):



Processing of exceptions

Normally the quality status of QUALITY is transferred to the output IN if the input OUT is set to TRUE. If the value of OUT exceeds the range of the float value (upper limit value UL or lower limit value LL of the float variables) during the calculation, the quality status of OUT is set to BAD, the output TRBL_AL is set to TRUE and OUT is limited to UL or LL respectively. If because of an error during the parameter assignment Xn > Xn+1, the output TRBL_AL is also set to TRUE. If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see following table).

In case of simulation the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	QUALITY		
Xn > Xn+1	Irrelevant	OUT = Yn; TRBL_AL = TRUE	-
Overflow		OUT = UL or LL; TRBL_AL = TRUE	OUT = BAD

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	**QUALITY**		
IN = not GOD	FALSE	-	OUT = GOD
IN = not GOD	TRUE	-	OUT has the quality status of IN

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.19 POWER (ACalc4cnet) | Description

Description

The power function calculates an analog value X to potency Y and provides the result in the output.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
X	FLOAT	Base	1.0	3.4028235E38	0.0	x	x	x	
Y	FLOAT	Exponent	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The output value is calculated according to the following formula:

$OUT = X^y$

Processing of exceptions

If the upper limit value of the output parameter OUT is exceeded (upper limit value UL of the float variables) the value of OUT is limited to UL and the digital output TRBL_AL is set to TRUE. If X equals zero and the value of Y is less than or equal to zero, OUT is set to zero, TRBL_AL is set to TRUE and OUT contains the value BAD.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see following table). If the QUALITY input is set to TRUE and the quality status of X or Y is not GOD, the worst quality status of X and Y is transferred to the output port OUT.

In case of simulation (parameters simulated) the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		**Effect on further processing**	**Effect on the quality code**
Input/output	**QUALITY**		
OUT > UL	Irrelevant	OUT = UL; TRBL_AL = TRUE	OUT = BAD
X = 0.0; Y =< 0.0		OUT = 0.0; TRBL_AL = TRUE	OUT = BAD
OUT < LL		OUT = LL; TRBL_AL = TRUE	OUT = BAD
X = not GOD	FALSE	-	OUT = GOD
X or Y = not GOD	TRUE	-	OUT = worst quality status of the inputs

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.20 SUB (ACalc4cnet) | Description

Description

The subtraction function executes arithmetic subtraction of two analog values and provides the result (difference) at the output.

Analog time-independent calculating functions (Page 224)

Parameter

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	FLOAT	Minuend	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN2	FLOAT	Subtrahend	0.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Difference			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The output value (difference) is calculated according to the following formula:

OUT = IN1 - IN2

Processing of exceptions

If the limit values of the output parameter OUT are exceeded/fall below (upper limit value UL or lower limit value LL), the value of OUT is limited to UL or LL and the digital output TRBL_AL is set to TRUE.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see the following table). If the QUALITY input is set to TRUE and the quality statuses of the inputs are not GOD, the worst quality status is transferred to the output OUT.

In case of simulation (parameters simulated) the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	QUALITY		
OUT > UL	Irrelevant	OUT = UL; TRBL_AL = TRUE	OUT = BAD
OUT < LL		OUT = LL; TRBL_AL = TRUE	OUT = BAD
Any input = not GOD	FALSE	-	OUT = GOD
Any input = not GOD	TRUE	-	OUT = worst quality status of the inputs

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.4.21 TRIGON (ACalc4cnet) | Description

Description

The TRIGON block calculates the sine, cosine and tangent of an angle as well as the inverse functions arcsine, arccosine and arctangent. The result is provided at the output. The algorithm calculates the values in degrees or radians.

Analog time-independent calculating functions (Page 224)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Analog input variable	0.0	3.4028235E38	-3.4028235E38	x	x	x	
FUNCTION	INT	Trigonometrical functions Default = 0 0 = Sin 1 = Cos 2 = Tan 3 = Arcsin 4 = Arccos 5 = Arctan	0	5	0		x	x	
BASE	INT	Calculation base Default = 0 0 = Degrees 1 = Radians	0	1	0		x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	1 / 1	x	x		x

Functional description and principle of operation

For TRIGON the calculation base degrees or radians can be selected. When the sine, cosine or tangent are calculated, BASE has to correspond to the units of the input angle (input IN). When the arcsine, arccosine or arctangent are calculated, BASE is used to specify whether the calculated value OUT is output in radians or degrees.

Trigonometrical functions sine, cosine and tangent

TRIGON makes the basic functions sine, cosine and tangent available, whereby the desired function is selected via the input FUNCTION.

Since the trigonometrical functions are periodical, meaning that the period of sin and cos are 2*π and the period of tan is π, the value of input IN is not specifically limited.

Note, however, that the accuracy for input values of more than ±1*E06 declines because of the mathematical accuracy of the floating point function. Values of IN, which result in calculated values outside the range OUT are not within the permissible value range. A range violation or signal quality BAD is subject to the processing of messages and exceptions.

The following table shows the values of OUT for the respective selected values of FUNCTION and BASE for specific values of IN.

BASE = 0 (degrees)				**BASE = 1 (radians)**			
ON	**SIN(IN)**	**COS(IN)**	**TAN(IN)**	**ON**	**SIN(IN)**	**COS(IN)**	**TAN(IN)**
0°	0	1	0	0	0	1	0
30°	½	½√3	1/3 √3	π/6	½ √3	½	1/3 √3
45°	½ √2	½ √2	1	π/4	½ √2	½	1
60°	½ √3	½	√3	π/3	½	½	√3
90°	1	0	-NA-	π/2	1	0	-NA-
180°	0	-1	0	π	0	-1	0
270°	-1	0	-NA-	3π/2	-1	0	-NA-
360°	0	1	0	2π	0	1	0

-NA- = Not defined

Trigonometrical functions arcsine, arccosine, arctangent

TRIGON makes the basic inverse trigonometrical functions arcsine, arccosine, and arctangent available, whereby the desired function is selected via the input FUNCTION.

The selected inverse trigonometrical FUNCTION specifies the permissible range of IN. A range violation or signal quality BAD is subject to the processing of messages and exceptions.

The following table shows the values of OUT and the permissible values of IN for the selected values of FUNCTION and BASE.

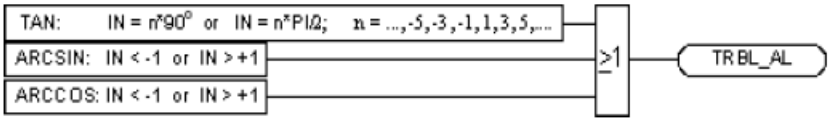
BASE = 0 (degrees)			
ON	**SIN(IN)**	**COS(IN)**	**TAN(IN)**
-1 ≤ IN ≤ +1	-90° ≤ OUT ≤ +90°	+180° ≥ OUT ≥ 0°	-45° ≤ OUT ≤ +45°
IN < -1	-NA-	-NA-	-90° < OUT < -45°
+1 > IN ≤ $3.4 * 10^{38}$	-NA-	-NA-	+45° < OUT < +90°

-NA- = Not defined

BASE = 1 (radians)			
ON	**SIN(IN)**	**COS(IN)**	**TAN(IN)**
-1 ≤ IN ≤ +1	- π/2 ≤ OUT ≤ + π/2	π ≥ OUT ≥ 0	- π/4 ≤ OUT ≤ + π/4
IN < -1	-NA-	-NA-	- π/2 < OUT < - π/4
+1 > IN ≤ $3.4 * 10^{38}$	-NA-	-NA-	+ π/4 < OUT < + π/2

Message processing

The permissible range of the input values is monitored for each selected trigonometrical function. The fault message output TRBL_AL functions as the message indicator of TRIGON. Each time a range violation exist, the output TRBL_AL is set to TRUE. The logic of the fault message is illustrated in the following scheme:



Note

TRBL_AL

TRBL_AL is only set for the following condition if TRIGON is assigned to an S7 runtime container:

TAN: IN=n*PI/2; n...,-5, -3, -1, 1, 3, 5,... >=1; where BASE= **RAD**

Processing of exceptions

If the value of IN results in a value of Not defined for the selected trigonometrical function, the value of OUT is kept and the output TRBL_AL is set to TRUE. Otherwise the value of TRBL_AL is FALSE.

Normally OUT has the quality status GOD. If the QUALITY input is set to TRUE and the quality code of IN is not GOD, the quality status of IN is transferred to the output parameter OUT. If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see following table).

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		**Effect on further processing**	**Effect on the quality code**
Input/output	**QUALITY**		
IN lies outside the permissible range for the selected FUNCTION	Irrelevant	OUT=OUT alt or; TRBL_AL = TRUE	OUT = BAD
IN = not GOD	FALSE	-	OUT = GOD
IN = not GOD	TRUE	-	OUT = Quality status of IN

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.5 ACalcTimeR4cnet | Analog time-dependent calculating function library

A.5.1 DERIV (ACalcTimeR4cnet) | Description

Description

The differentiation function uses the trapezoidal rule to calculate the speed of change of an analog signal.

Analog time-dependent calculating functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Input signal	0.0	3.4028235E38	-3.4028235E38	x	x	x	
TD	FLOAT	Time difference [s]	1.0	30000.0	0.0	x	x	x	
T1	FLOAT	Delay time [s]	10.0	30000.0	0.0	x	x	x	
UL	FLOAT	High limit	100.0	3.4028235E38	-3.4028235E38	x	x	x	
LL	FLOAT	Low limit	-100.0	3.4028235E38	-3.4028235E38	x	x	x	
RES	BOOL	Reset of OUT	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
OUT_UL	BOOL	Output high limit value			x	x		
OUT_LL	BOOL	Output low limit value			x	x		
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

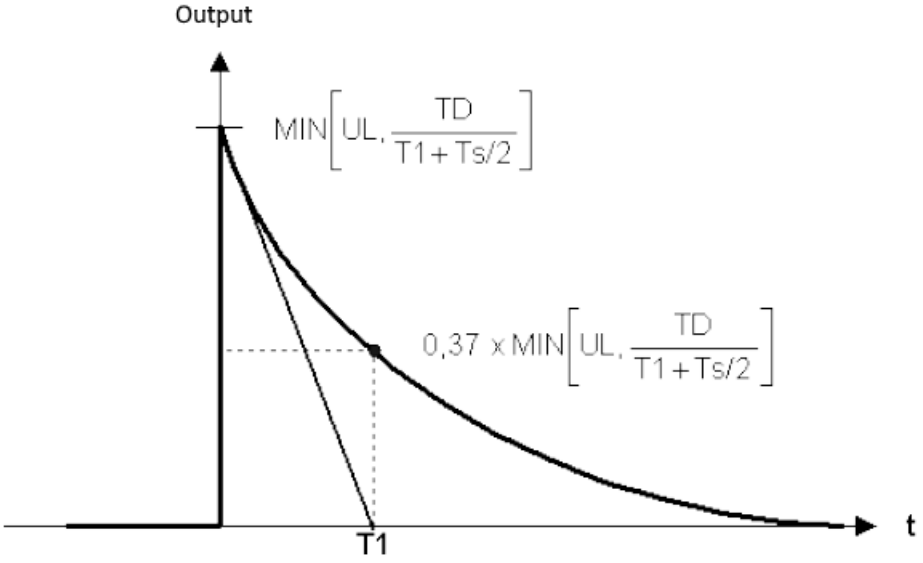
The calculation of the output value takes place according to the following equation:

$$OUT_n = OUT_{n-1} \times \left(1 - \frac{Ts}{Ts/2 + T1}\right) + \frac{TD}{Ts/2 + T1} \times (IN_n - IN_{n-1})$$

where:

- Analog value of the sampling interval n:
  - OUT
  - n / IN n
- Analog value of the sampling interval n-1:
  - OUT n-1 / IN n-1
- Programmed runtime cycle:
  - Ts (sampling time)

The step reaction of an amplitude with the value IN = 1 is:



where:

- Programmed runtime cycle: Ts (sampling time)
- Time difference [s]: TD
- Delay time [s]: T1

The following furthermore applies:

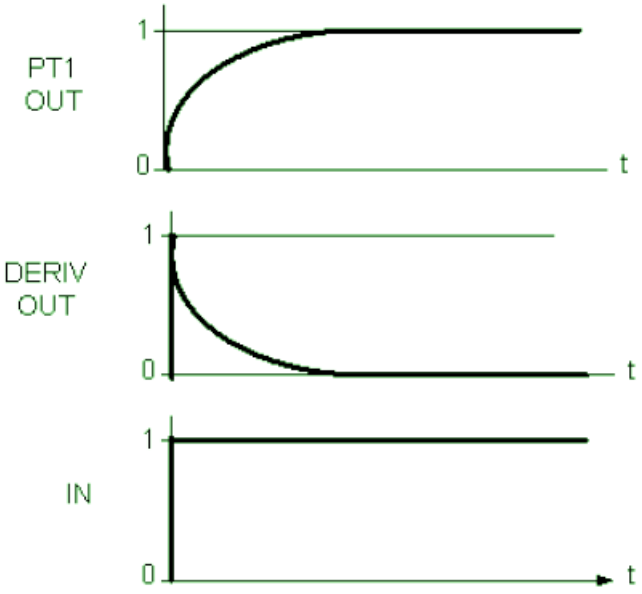
- The upper and lower range limit of the output OUT is specified by UL and LL.
- With activated limitation the respective output OUT_UL or OUT_LL is set additionally.
- If the RESET input RES is TRUE, the output OUT is set to 0.0.

Inverted response to first-level delay

The DERIV block can be set so that it delivers the inverted response of the PT1 function. To do this, the parameters of DERIV must be specified as follows:

- T1 of the function DERIV = T1 of the function PT1
- In the DERIV function: T1 = TD.

An example of the resulting response is shown below:



Note

In this application the function DERIV is sometimes referred to as the DT1 function.

Processing of exceptions

If the lower limit value LL exceeds the upper limit value UL, the internal calculation is executed with UL = LL and the digital outputs OUT_UL and OUT_LL are set to TRUE. If the value of T1 is less than Ts/2, the internal calculation is executed with T1 = Ts/2 (no delay). In each case the quality status of input IN is transferred to the output OUT when the QUALITY input is set to TRUE.

If any value exceeds the float value range during the calculation, the quality status of OUT is set to BAD, the fault message output TRBL_AL is set to TRUE and OUT keeps the last value.

If the QUALITY input is set to FALSE, the quality status of OUT is GOD, except in cases where it is not relevant (see following table).

In case of simulation the quality status of OUT is always changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		**Effect on further processing**	**Effect on the quality code**
Input/output	**QUALITY**		
LL > UL	Irrelevant	UL is set internally to LL.	-
T1 < Ts/2		T1 is set internally to Ts/2.	-
Overflow during the calculation		TRBL_AL = TRUE	OUT = BAD
IN = not GOD	FALSE	-	OUT = GOD
IN = not GOD	TRUE	-	OUT = Quality status of the input IN

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.5.2 INT (ACalcTimeR4cnet) | Description

Description

The integration function offers two execution modes:

1. Integration operation
   The output signal rises with a positive and falls with a negative input signal.
2. Tracking mode
   The TRACKV input signal is looped through the function to the output OUT.

Analog time-dependent calculating functions (Page 225)

Parameters

Inputs

Parameter	**Data type**	**Description**	**Default**	**Maximum value**	**Minimum value**	**Conn**	**Edit**	**L**	**HMI**
TRACKV	FLOAT	External tracking variable	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN	FLOAT	Input signal	0.0	3.4028235E38	-3.4028235E38	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
TN	FLOAT	Integration time	1.0	3.4028235E38	0.0	x	x	x	
UL	FLOAT	High limit	100.0	3.4028235E38	-3.4028235E38	x	x	x	
LL	FLOAT	Low limit	-100.0	3.4028235E38	-3.4028235E38	x	x	x	
C_TRACK	BOOL	Tracking command	FALSE	TRUE	FALSE	x	x	x	
EN_INC	BOOL	Output increase permitted	TRUE	TRUE	FALSE	x	x	x	
EN_DEC	BOOL	Output decrease permitted	TRUE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
OUT_UL	BOOL	Output high limit value			x	x		
OUT_LL	BOOL	Output low limit value			x	x		
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

The calculation of the output value takes place according to the following equation:

$$OUT_n = OUT_{n-1} + \frac{Ts}{TN} \times IN_n$$

where:

- Analog value of the sampling interval n:
  OUT n / IN n
- Analog value of the sampling interval n-1:
  OUT n-1 / IN n-1
- Programmed runtime cycle:
  Ts (sampling time)

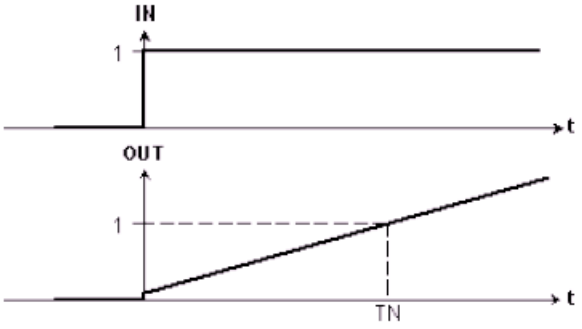
Note

The output signal is limited by LL and UL.

Integration operation

The integration function must be enabled via the inputs EN_INC (increase) and EN_DEC (decrease).

The step response of an amplitude with the value 1 (IN: 0 -> 1) is shown in the following scheme:



where:

- Integration time in seconds:
  TN

Tracking mode

The TRACKV input signal is looped through the function to the output OUT. The mode is activated via the digital input C_TRACK. The follow-up mode can be used to preset the integration function.

Processing of exceptions

If the value of TN is less than or equal to 0.0 or if any value exceeds the float value range during the calculation, the quality status of OUT is set to BAD, the fault message output parameter TRBL_AL is set to TRUE and OUT keeps the last value.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see following table). If the QUALITY input is set to TRUE, the quality status of input IN is transferred to the output OUT in integration operation, and the quality status of TRACKV is transferred to the output OUT in tracking mode.

In case of simulation the quality status of OUT is always changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		**Effect on further processing**	**Effect on the quality code**
Input/output	**QUALITY**		
LL > UL	Irrelevant	UL is set internally to LL	-
TN =< 0.0		OUT keeps the last value; TRBL_AL = TRUE	OUT = BAD
Overflow during the calculation			

Conditions for error handling			Effect on further processing	Effect on the quality code
Input/output		QUALITY		
Any input = not GOD		FALSE	-	OUT = GOD
Any input = not GOD	C_TRACK = FALSE	TRUE	-	OUT = Quality status of the input IN
	C_TRACK = TRUE			OUT = Quality status of TRACK_V

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.5.3 LEADLAG (ACalcTimeR4cnet) | Description

Description

The LEADLAG block executes a phase lead or phase lag function.

Analog time-dependent calculating functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Input signal	0.0	3.4028235E38	-3.4028235E38	x	x	x	
T_LEAD	FLOAT	LEAD time constant [s]	1.0	30000.0	0.0	x	x	x	
T_LAG	FLOAT	LAG time constant [s]	1.0	30000.0	0.0	x	x	x	
GAIN	FLOAT	Gain	1.0	3.4028235E38	-3.4028235E38	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

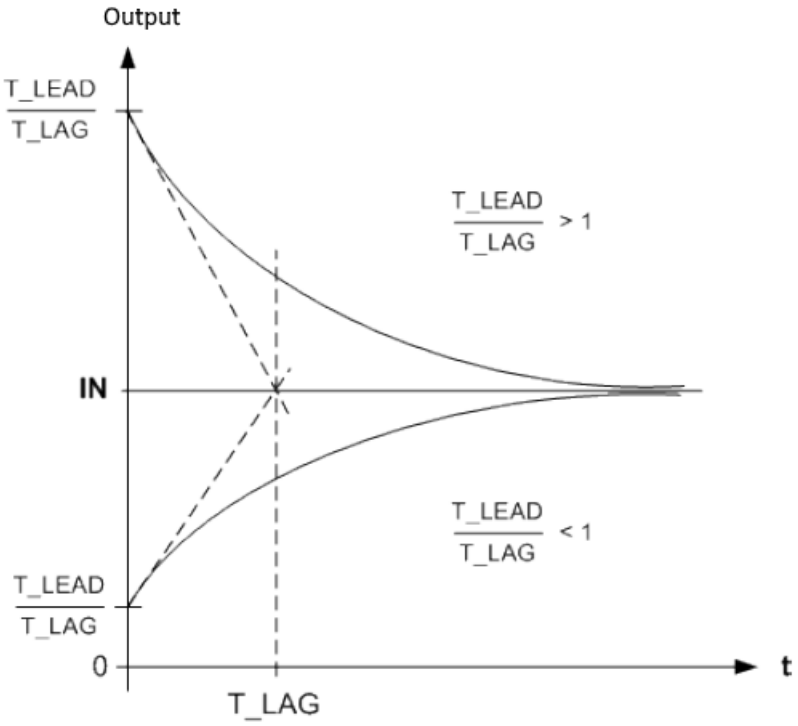
Functional description and principle of operation

A phase lead function is performed when the time value of LEAD is greater than the value of LAG. A lag function is performed when the time value of LAG is greater than the value of LEAD.

LEADLAG is a first-level transfer of the quotient of LEAD and LAG (reciprocal of the zeros / reciprocal of the poles).

If T_LEAD is zero, a first-level delay takes place (see delay function PT1). The quotient T_LEAD/ T_LAG defines the T(0) response to a step change whereby the fade-out speed of the step change is an exponential function of the delay time.

The step response of an amplitude with the value IN = 1.0 (GAIN = 1.0) is thus:



Processing of exceptions

If the value of T_LAG is less than Ts, the internal calculation is executed with T_LAG = Ts. In each case the quality status of input IN is transferred to the output OUT when the QUALITY input is set to TRUE.

If any value exceeds the float value range during the calculation, the quality status of OUT is set to BAD, the fault message output TRBL_AL is set to TRUE and OUT keeps the last value. If the QUALITY input is set to FALSE, the quality status of OUT is GOD, except in cases where it is not relevant (see following table).

In case of simulation the quality status of OUT is always changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	**QUALITY**		
T_LAG < Ts	Irrelevant	T_LAG is set internally to Ts	-
Overflow during the calculation		TRBL_AL = TRUE	OUT = BAD
IN = not GOD	FALSE	-	OUT = GOD
IN = not GOD	TRUE	-	OUT = Quality status of the input IN

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.5.4 PT1 (ACalcTimeR4cnet) | Description

Description

The first-level delay function implements a delay of the signal rise with the time constant T1.

Analog time-dependent calculating functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Input signal	0.0	3.4028235E38	-3.4028235E38	x	x	x	
T1	FLOAT	Delay time	10.0	1000000.0	0.0	x	x	x	
D_OFF	BOOL	Delay off	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x		x

Functional description and principle of operation

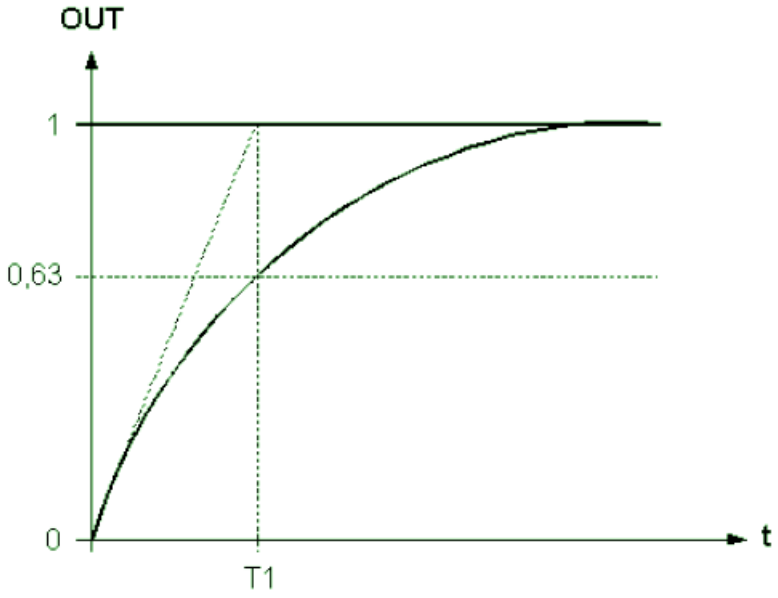
The delay function PT1 can be used to smoothen analog signals. The calculation of the output value takes place according to the following equation:

$$OUT_n = OUT_{n-1} + \frac{Ts}{Ts/2 + T1} \times (IN_n - OUT_{n-1})$$

where:

- Analog value of the sampling interval n:
  OUT_n / IN_n
- Analog value of the sampling interval n-1:
  OUT_{n-1}
- Programmed runtime cycle:
  Ts (sampling time)

The step response of an amplitude with the value IN = 1 is shown in the following figure:



where:

- Delay time:
  T1

The delay can be switched off via the binary input D_OFF: The input signal IN is then looped through the function to the output OUT.

Processing of exceptions

If the value of T1 is less than Ts/2, the internal calculation is executed with T1 = Ts/2 (no delay). If any value exceeds the float value range during the calculation, the quality status of OUT is set to BAD, the fault message output TRBL_AL is set to TRUE and OUT keeps the last value.

If the QUALITY input is set to TRUE, the quality status of input IN is transferred to the output OUT.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD, except in cases where it is not relevant (see the following table).

In case of simulation the quality status of OUT is always changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		**Effect on further processing**	**Effect on the quality code**
Input/output	**QUALITY**		
T1 < Ts/2	Irrelevant	T1 is set internally to Ts/2.	-
Overflow during the calculation		TRBL_AL = TRUE	OUT = INVALID
IN = not GOD	FALSE	-	OUT = GOD
IN = not GOD	TRUE	-	OUT = Quality status of the input IN

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.5.5 RAXFR (ACalcTimeR4cnet) | Description

Description

The block RAXFR switches one of the two analog inputs to an analog output with the possibility to specify transfer speeds for both ramp directions.

Analog time-dependent calculating functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
NO	FLOAT	Analog input variable 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	
YES	FLOAT	Analog input variable 2	0.0	3.4028235E38	-3.4028235E38	x	x	x	
SEL	BOOL	Preselection	FALSE	TRUE	FALSE	x	x	x	
RR_NO	FLOAT	Ramp gradient for input 1 in EU/BASE	0.0	3.4028235E38	0.0	x	x	x	
RR_YES	FLOAT	Ramp gradient for input 2 in EU/BASE	0.0	3.4028235E38	0.0	x	x	x	
ALV	FLOAT	Limit value for comparison in EU	0.0	3.4028235E38	0.0	x	x	x	
BASE	BOOL	Time base Default = False True = Min. False = Sec.	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	

Functional description and principle of operation

The output value OUT is either equal to one of the two analog inputs or executes an increasing/falling ramp function with one of the two analog input values.

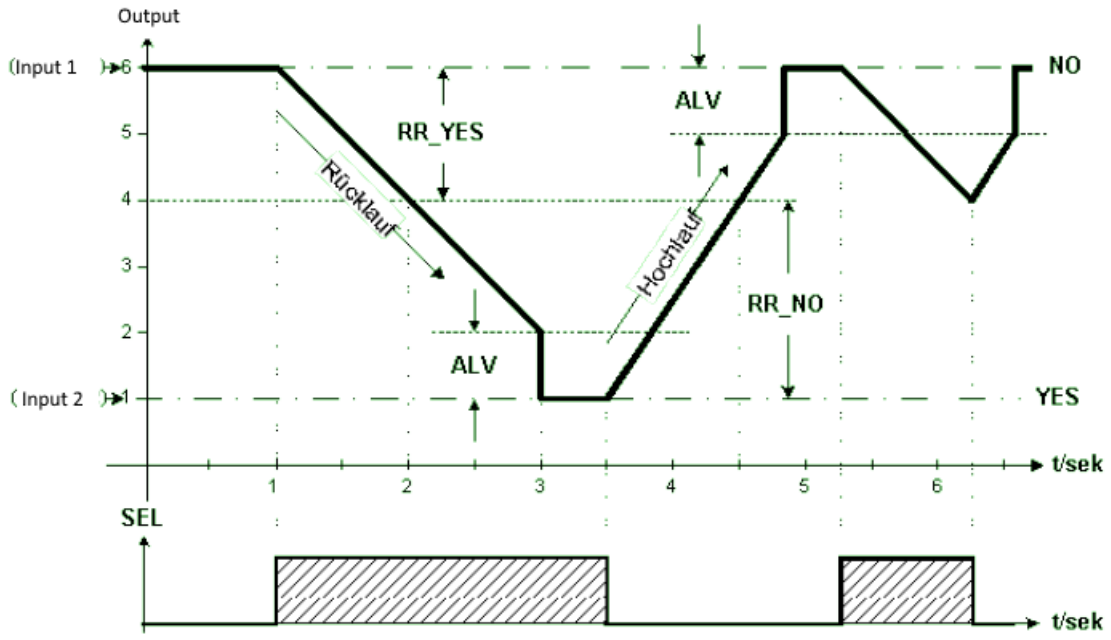
- If SEL = FALSE, OUT takes on the value of the NO input.
- If SEL = TRUE, OUT takes on the value of the YES input.

If the input status of the selection signal SEL changes, the output signal OUT performs a ramp function with the specified ramp gradient RR_NO or RR_YES until the selected input value is reached.

As long as the difference between OUT and the selected input value is greater than the value configured in ALV, the adaptation is performed via the ramp function. If the difference between OUT and the selected input falls below the value configured in ALV, OUT is immediately set to the value of the selected input.

Graphical representation

The processing of the output signal OUT is illustrated in the following figure:



The following values are applied in the process:

- Input NO:
  - Actual value = 6.0 (constant); ramp gradient
  - RR_NO = 3.0 (per second)
- Input YES:
  - Actual value = 1.0 (constant); ramp gradient
  - RR_YES = 2.0 (per second)
- Adjustment limit:
  - ALV = 1.0

Processing of exceptions

Normally the quality status of the selected input is transferred to the output OUT if the input QUALITY is set to TRUE.

If the QUALITY input is set to FALSE, the quality status of OUT is GOD, except in cases where it is not relevant (see following table). If the value of one of the ramp gradients is less than or equal to zero, the function is switched off, meaning the value of OUT takes on the value of the selected input. If the value of ALV is less than zero, ALV is set to zero.

In case of simulation the quality status of OUT is always changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling		Effect on further processing	Effect on the quality code
Input/output	QUALITY		
RR_NO <= 0.0; SEL = FALSE	Irrelevant	OUT = NO	-
RR_YES <= 0.0; SEL = TRUE		OUT = YES	-
ALV < 0.0		ALV = 0.0	-
Selected input = not GOD	FALSE	-	OUT = GOD
	TRUE	-	OUT = Quality status of the selected input

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6 BCalc4cnet | Binary time-independent calculating function library

A.6.1 AND (BCalc4cnet) | Description

Description

The AND block enables the connecting operation of at least 2 to a maximum of 8 binary inputs.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1 ... IN8	BOOL	Binary input	TRUE	TRUE	FALSE	x		x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		

Functional description and principle of operation

The binary output is TRUE when all input values are TRUE.

Truth table:

IN1	IN2	to	IN8	OUT
0	0	to	0	0
1	0	to	0	0
1	1	to	0	0
:	:	:	:	:
1	1	1	1	1

Processing of exceptions

If the QUALITY input is set to TRUE:

- If the value of OUT is TRUE, the worst quality status is transferred to the output OUT.
- If the value of OUT is FALSE, the quality status of OUT is GOD.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD (valid).

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.2 BSEL (BCalc4cnet) | Description

Description

The "Binary signal selection" block monitors up to 16 binary inputs for the TRUE status. The binary output OUT is set to TRUE when at least the specified number of inputs is TRUE.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1 ... IN16	BOOL	Binary input	FALSE	TRUE	FALSE	x	x	x	
N_IN	INT	Number of inputs with signal state TRUE	2	16	1		x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x	x	

Functional description and principle of operation

The binary signal selection function monitors all connected inputs for the TRUE status. At least the number of inputs defined in N_IN must have the status TRUE. Then the binary output OUT is set to TRUE.

Only inputs whose quality status is GOD are considered. If the quality status is not GOD, the respective input is excluded from the binary signal selection.

Example:

In the following example the three inputs IN1 to IN3 are connected with any binary signal sources. The minimum number of two signals with the TRUE status is set with N_IN = 2. The quality status of IN2 is assumed as BAD. Under these conditions the output OUT behaves according to the following truth table:

IN1 GOD	IN2 BAD	IN3 GOD	OUT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	**0**
1	0	0	0
1	0	1	1
1	1	0	**0**
1	1	1	1

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD.

If the QUALITY input is set to TRUE, each input with a quality status which is not GOD is excluded from the selection.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.3 BSIG (BCalc4cnet) | Description

Description

The binary signal transmitter generates a constant binary value.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	BOOL	Binary constant	TRUE	TRUE	FALSE		x	x	x

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		

Functional description and principle of operation

The input can only be configured but not connected. The output is set to the binary value configured in the input.

Processing of exceptions

Not applicable

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.4 BXFR (BCalc4cnet) | Description

Description

The BXFR block switches one of two binary input signals to a binary output.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
NO	BOOL	Binary input 1	FALSE	TRUE	FALSE	x	x	x	
YES	BOOL	Binary input 2	FALSE	TRUE	FALSE	x	x	x	
SEL	BOOL	Preselection	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		

Functional description and principle of operation

The value of OUT is always equal to one of the two binary input values, depending on the state of SEL:

- If SEL = FALSE, then the value of OUT corresponds to the input NO
- If SEL = TRUE, then the value of OUT corresponds to the input YES

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD. If the QUALITY input is set to TRUE, the quality status of the selected input is transferred to the output OUT.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.5 COUNTER (BCalc4cnet) | Description

Description

The counter function offers an edge-controlled upwards and downwards counter for positive and negative values. The counter output can be predefined.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
CU	BOOL	Count up	FALSE	TRUE	FALSE	x	x	x	
CD	BOOL	Count down	FALSE	TRUE	FALSE	x	x	x	
EN_UP	BOOL	Enable UP	TRUE	TRUE	FALSE	x	x	x	
EN_DOWN	BOOL	Enable DOWN	TRUE	TRUE	FALSE	x	x	x	
S	BOOL	Set counter to SV	FALSE	TRUE	FALSE	x	x	x	
R	BOOL	Reset counter to LL	FALSE	TRUE	FALSE	x	x	x	
SV	FLOAT	Specification for counter	0.0	1.6E7	-1.6E7	x	x	x	x
UL	FLOAT	High limit counter	0.0	1.6E7	-1.6E7	x	x	x	x
LL	FLOAT	Low limit counter	0.0	1.6E7	-1.6E7	x	x	x	x

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Counter output			x	x	x	
OUT_UL	BOOL	Output high limit value			x	x	x	
OUT_LL	BOOL	Output low limit value			x	x	x	

Functional description and principle of operation

The function range of the counter can be defined by setting an upper or lower limit value. The behavior of the counter output OUT depends on the following conditions:

- If the binary input CU switches its status from FALSE to TRUE while EN_UP is TRUE, the output OUT is counted up.
- If the binary input CD switches its status from FALSE to TRUE, while EN_DO is TRUE, the output parameter OUT is counted down.
- If the setpoint value input S is TRUE, the output OUT is set to the value of SV (OUT remains limited to the upper and lower limit value UL or LL).
- If the reset input R is TRUE, the output OUT is set to the lower limit LL with the highest priority.
- If OUT is limited by UL, the binary output OUT_UL is set to TRUE.

- If OUT is limited by LL, the binary output OUT_LL is set to TRUE.
- If the binary inputs CU and CD switch their statuses simultaneously from FALSE to TRUE, the output OUT keeps its last value.

Truth table:

CU*	**CD***	**S**	**R**	**OUT $_n$**
0	0	0	0	OUT $_{n-1}$
0	1	0	0	OUT $_{n-1}$-1
1	0	0	0	OUT $_{n-1}$+1
1	1	0	0	OUT $_{n-1}$
x	x	1	0	SV
x	x	x	1	LL

where:

- No effect:
  x
- Current sampling cycle:
  n
- Previous sampling cycle:
  n-1
- Trigger condition, EN_UP = EN_DO = TRUE:
  *

Note

With the first activation the internal counter is set to the value of LL. Further processing depends on the current input statuses.

Processing of exceptions

For the counting algorithm, inputs of the data type FLOAT (SV, UL, LL) have to be limited to the decimal point. If, for example, the value of SV is set to 1000.5, SV is set internally to 1000.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.6 DYN_OR (BCalc4cnet) | Description

Description

The DYN_OR block enables the OR logic operation of up to 8 binary inputs. Both static and dynamic outputs are provided.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1 ... IN8	BOOL	Binary input	FALSE	TRUE	FALSE	x		x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT_STAT	BOOL	Static binary output			x	x		
OUT_DYN	BOOL	Dynamic binary output			x	x		

Functional description and principle of operation

The function has two outputs, one static and one dynamic.

- **Static output**:
  The static binary output OUT_STAT is TRUE when at least one of the input values is TRUE.
  Truth table**:**

IN1	In2	to	In8	OUT_STAT
0	0	to	0	0
1	0	to	0	1
1	1	to	0	1
:	:	:	:	:
1	1	1	1	1

- **Dynamic output**:
  The dynamic binary output OUT_DYN becomes TRUE for 1 execution cycle, as soon as a rising edge of one of the inputs values INx is detected. A rising edge for an input value occurs during the transfer from FALSE to TRUE.

Processing of exceptions

If the QUALITY input is set to TRUE:

- If the value of OUT_STAT is TRUE, the best quality status of the inputs that have the value TRUE, are transferred to the output OUT_STAT.
- If the value of OUT_STAT is FALSE, the quality status of OUT_STAT is GOD.

If the QUALITY input is set to FALSE, the quality status of OUT_STAT is always set to GOD (valid).

In case of simulation the quality status of OUT_STAT is always changed to GFO.

The quality status of OUT_DYN is always equal to the state of OUT_STAT.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.7 EXOR (BCalc4cnet) | Description

Description

The output is TRUE if only one input is TRUE.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	BOOL	Binary input 1	FALSE	TRUE	FALSE	x	x	x	
IN2	BOOL	Binary input 2	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		

Functional description and principle of operation

The output OUT is only TRUE when one single input is TRUE.

Truth table:

IN1	IN2	OUT
0	0	0
0	1	1
1	0	1
1	1	0

Processing of exceptions

If the QUALITY input is set to TRUE:

- If the value of OUT is TRUE, the worst quality status is transferred to the output OUT.
- If the value of OUT is FALSE, the quality status of OUT is GOD.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD (valid).

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.8 JK_FF (BCalc4cnet) | Description

Description

The JK flip-flop executes the function of an isochronous trigger element which can take on two internal statuses.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
J	BOOL	J input	FALSE	TRUE	FALSE	x	x	x	
K	BOOL	K input	FALSE	TRUE	FALSE	x	x	x	
C	BOOL	Cycle input	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

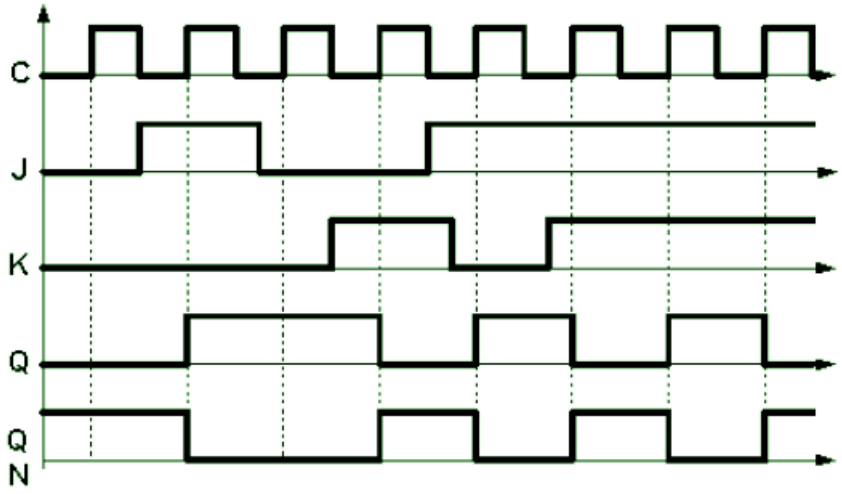
Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
Q	BOOL	Output			x	x		
Q_N	BOOL	Negated output			x	x		

Functional description and principle of operation

The JK flip-flop function is used to store binary values. The status of the value is available in the output Q while Q_N contains the relevant complementary value.

The inputs J and K behave like the inputs S and R of the RS flip-flop function by setting and resetting the JK flip-flop respectively. The status change of Q and Q_N is triggered by a rising edge in input C. If the two inputs J and K are FALSE the outputs remain unchanged. If both inputs are TRUE, the outputs Q and Q_N switch for every rising edge to cycle input C into their complementary status.

Graphical representation:



The transition state of the JK flip-flop function is reproduced in the following table:

J	K	C	Q	Q_N
0	0	X	Q_{n-1}	Q_N_{n-1}
0	1	0	Q_{n-1}	Q_N_{n-1}
0	1	1	0	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

where:

- No effect:
  X
- Last sampling cycle:
  n-1

With the first activation, Q has the default setting FALSE (Q_N = TRUE).

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD.

If the QUALITY input is set to TRUE and the value of J or K is TRUE with a quality status of not GOD, this quality status is transferred to Q and Q_N.

If the QUALITY input is set to TRUE and both inputs are TRUE, the worst quality status is looped through. In case of simulation the quality status of Q or Q_N is changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.9 NOT (BCalc4cnet) | Description

Description

The NOT logic reverses the status of a binary signal.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	BOOL	Binary input	TRUE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		

Functional description and principle of operation

The binary output OUT is TRUE when the input signal is FALSE.

Truth table:

IN	OUT
0	1
1	0

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD.

In case of simulation the quality status of OUT is always changed to GFO. If the QUALITY input is set to TRUE, the quality status of IN is transferred to the output OUT.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.10 OR (BCalc4cnet) | Description

Description

The OR block enables an OR logic operation of at least 2 to a maximum of 8 binary inputs.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1 ... IN8	BOOL	Binary input	FALSE	TRUE	FALSE	x		x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		

Functional description and principle of operation

The binary output OUT is TRUE when at least one of the input values is TRUE.

Truth table:

IN1	IN2	to	IN8	OUT
0	0	to	0	0
1	0	to	0	1
1	1	to	0	1
:	:	:	:	:
1	1	1	1	1

Processing of exceptions

If the QUALITY input is set to TRUE:

- If the value of OUT is TRUE, the best quality status of the inputs that have the value TRUE, is transferred to the output OUT.
- If the value of OUT is FALSE, the quality status of OUT is GOD.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD (valid).

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.6.11 RS_FF (BCalc4cnet) | Description

Description

The RS flip-flop executes the function of a bistable element which can take on two internal statuses.

Binary time-independent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
S	BOOL	Set input	FALSE	TRUE	FALSE	x	x	x	
R	BOOL	Reset input	FALSE	TRUE	FALSE	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
DOM	BOOL	Preference position Default = False True = Set False = Reset	FALSE	TRUE	FALSE		x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
Q	BOOL	Output			x	x		
Q_N	BOOL	Negated output			x	x		

Functional description and principle of operation

The "RS flip-flop" block is used to store binary values. The status of the value is available in the output Q while Q_N contains the relevant complementary value.

Dominance behavior

If both inputs are TRUE, the output is set Q when DOM is TRUE, and the output Q_N is set when DOM is FALSE. The transition state of the RS flip-flop function is reproduced in the following table:

Truth table:

S	R	Q	Q_N	DOM
0	0	Q_{n-1}	Q_N_{n-1}	X
0	1	0	1	X
1	0	1	0	X
1	1	0	1	FALSE
1	1	1	0	TRUE

where:

- No effect:
  X
- Last sampling cycle:
  n-1

If both inputs (R and S) are FALSE, Q and Q_N remain in the status of the last sampling cycle. With the first activation, Q has the default setting FALSE (Q_N = TRUE).

Processing of exceptions

Normally the ports Q and Q_N have the quality status GOD.

If the QUALITY input is set to FALSE, the quality status of OUT is always set to GOD. If the QUALITY input is set to TRUE, the value of R or S is TRUE and both quality statuses are GOD; the corresponding quality status is transferred to Q and Q_N.

If the QUALITY input is set to TRUE and both inputs are TRUE, the quality status of the dominant input is looped through.

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.7 BCalcTimeR4cnet | Binary time-dependent calculating function library

A.7.1 LPULSE (BCalcTimeR4cnet) | Description

Description

The "Pulse generation storing" delivers the signal TRUE in the output during a predefined period when the input signal switches to TRUE. If the input becomes TRUE again before the period has expired, the output remains TRUE for an additional period.

Binary time-dependent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	BOOL	Binary input	FALSE	TRUE	FALSE	x	x	x	
TIME	FLOAT	Time period	1.0	30000.0	0.0	x	x	x	
BASE	INT	Time base Default = 0 0 = Sec. 1 = Min.	0	1	0		x	x	
RESET	BOOL	Reset input	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

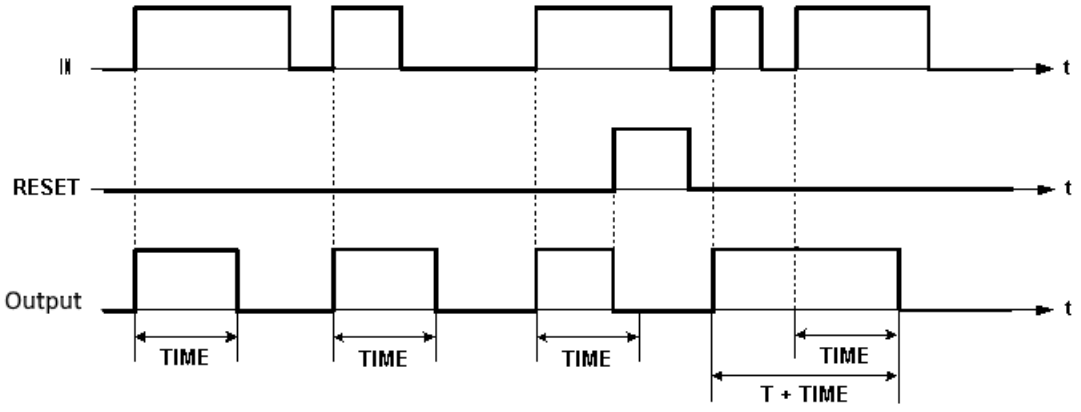
Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		
RT	FLOAT	Remaining time			x	x		x

Functional description and principle of operation

The behavior of OUT depends on the following conditions:

- If the binary input IN switches to TRUE, the output OUT also becomes TRUE until the time has expired. Subsequently OUT is set to FALSE.
- OUT remains TRUE, until the time duration in TIME has expired.
- If the time has not expired yet and IN switches a second time to TRUE, the time function is restarted with the time value of the input TIME (retriggered).
- If the RESET input is TRUE, OUT is set to FALSE.

Example:



As long as the timer runs the remaining runtime is output at the output RT.

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is set to GOD. If the QUALITY input is set to TRUE, the quality status of input IN is transferred to the output OUT.

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.7.2 MINPULS (BCalcTimeR4cnet) | Description

Description

The output of the minimum pulse or one-shot function is set to TRUE for a preset time period when the input signal switches to TRUE, unless the input signal is reset before the expiration of the time period.

Binary time-dependent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	BOOL	Binary input	FALSE	TRUE	FALSE	x	x	x	
TIME	FLOAT	Time period	1.0	30000.0	0.0	x	x	x	
BASE	INT	Time base Default = 0 0 = Sec. 1 = Min.	0	1	0		x	x	
RESET	BOOL	Reset input	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

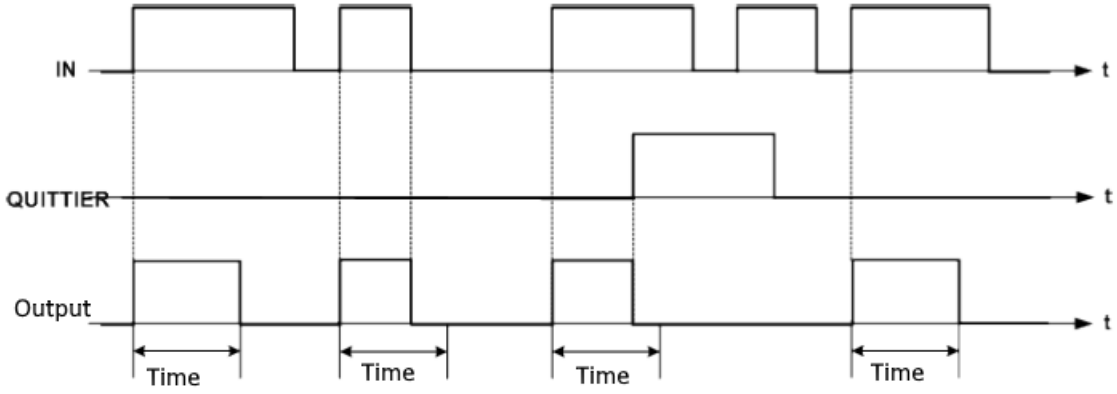
Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		
RT	FLOAT	Remaining time			x	x		x

Functional description and principle of operation

The behavior of OUT depends on the following conditions:

- If the binary input IN switches to TRUE, the output OUT also becomes TRUE until the time has expired. Subsequently OUT is set to FALSE.
- If IN switches to FALSE before the time has expired, OUT is set to FALSE.
- If the RESET input is TRUE, OUT is set to FALSE.

Example:



As long as the timer runs the remaining runtime is output at the output RT.

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is set to GOD. If the QUALITY input is set to TRUE, the quality status of input IN is transferred to the output OUT.

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.7.3 PULSE (BCalcTimeR4cnet) | Description

Description

The "Monoflop (one-shot)" block delivers the signal TRUE in the output during a predefined time period when the input signal switches to TRUE.

Binary time-dependent logic functions (Page 225)

Parameters

Inputs

Parame-ter	Data type	Description	Default	Maximum value	Minimum value	Con n	Edit	L	HMI
IN	BOOL	Binary input	FALSE	TRUE	FALSE	x	x	x	x
TIME	FLOAT	Time period	1.0	30000.0	0.0	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
BASE	INT	Time base Default = 0 0 = Sec. 1 = Min.	0	1	0		x	x	
RESET	BOOL	Reset input	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

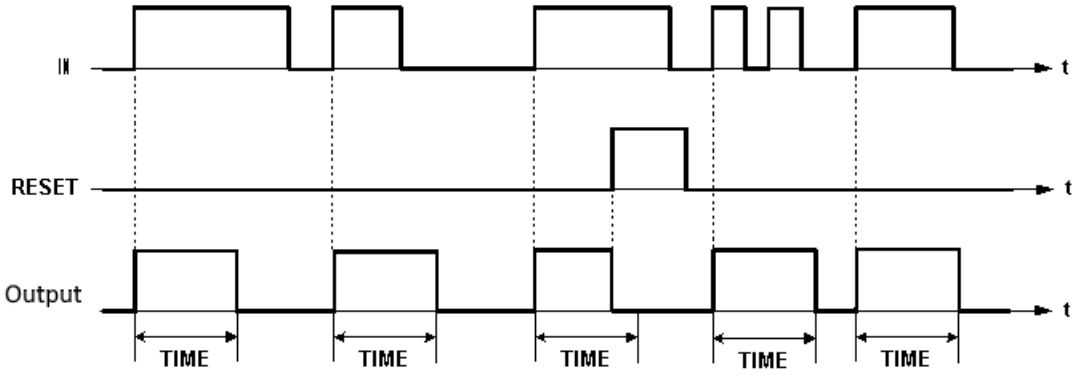
Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		
RT	FLOAT	Remaining time			x	x		x

Functional description and principle of operation

The behavior of OUT depends on the following conditions:

- If the binary input IN switches to TRUE, the output OUT also becomes TRUE until the time has expired. Subsequently OUT is set to FALSE.
- OUT remains TRUE, until the time duration in TIME has expired.
- If the RESET input is TRUE, OUT is set to FALSE.

Example:



As long as the timer runs the remaining runtime is output at the output RT.

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is set to GOD. If the QUALITY input is set to TRUE, the quality status of input IN is transferred to the output OUT.

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.7.4 T_LON (BCalcTimeR4cnet) | Description

Description

The block "ON delay latching" sets the output after a specific time period to TRUE when the trigger input switches to TRUE. After setting to TRUE the output remains TRUE until it is reset through a corresponding signal.

Binary time-dependent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	BOOL	Binary input	FALSE	TRUE	FALSE	x	x	x	x
TIME	FLOAT	Time period	1.0	30000.0	0.0	x	x	x	
BASE	INT	Time base Default = 0 0 = Sec. 1 = Min.	0	1	0		x	x	
RESET	BOOL	Reset input	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		
RT	FLOAT	Remaining time			x	x		x

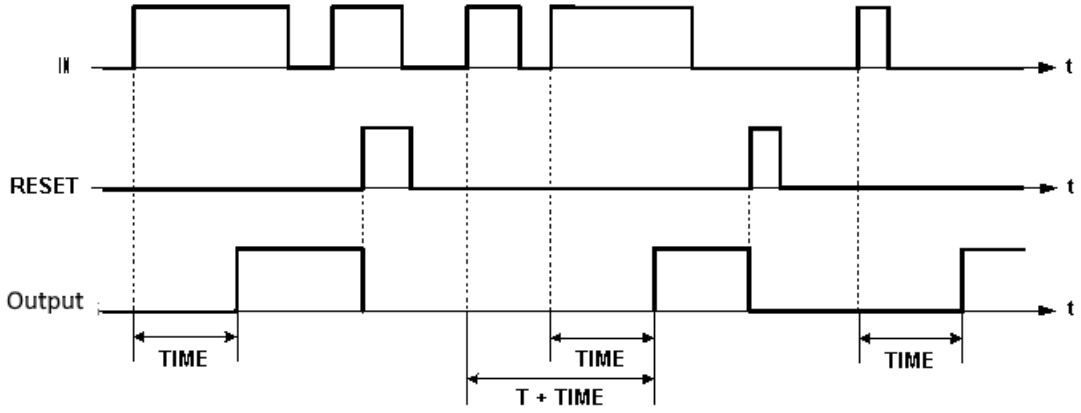
Functional description and principle of operation

The behavior of OUT depends on the following conditions:

- If the binary input IN switches to TRUE, the output OUT only becomes TRUE after expiration of the time duration TIME.
- If IN switches to FALSE before the time has expired, OUT is still set to TRUE after the time has expired.

- If IN switches a second time to TRUE before the time has expired, the delay is restarted with the time specified in TIME (retriggered).
- If the RESET input is TRUE, OUT is set to FALSE. This is the only method for resetting the output signal.

Example:



Note

Depending on the time period defined for the block it is possible that the signal appears in the output with a significant delay, after the recalling of the trigger input signals. When using this block you should therefore proceed with special caution so that no unexpected events occur during its execution.

As long as the timer runs the remaining runtime is output at the output RT.

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is set to GOD. If the QUALITY input is set to TRUE, the quality status of input IN is transferred to the output OUT.

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.7.5 T_OFF (BCalcTimeR4cnet) | Description

Description

The "OFF delay" block sets the output to TRUE when the input is TRUE. When the input switches to FALSE, the output switches with delay for a specified time period.

Binary time-dependent logic functions (Page 225)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	BOOL	Binary input	FALSE	TRUE	FALSE	x	x	x	
TIME	FLOAT	Time period	1.0	30000.0	0.0	x	x	x	
BASE	INT	Time base Default = 0 0 = Sec. 1 = Min.	0	1	0		x	x	
RESET	BOOL	Reset input	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

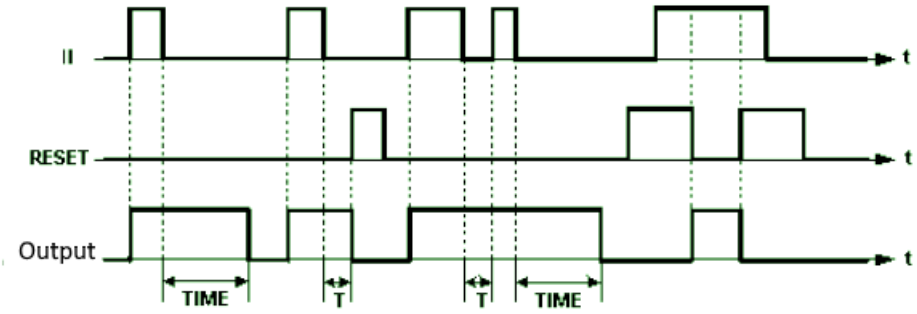
Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		
RT	FLOAT	Remaining time			x	x		x

Functional description and principle of operation

The behavior of OUT depends on the following conditions:

- If the binary input IN takes on the TRUE state the output OUT is also TRUE.
- If the binary input IN switches to FALSE, the output OUT only becomes FALSE after the time duration TIME has expired.
- If the IN input switches to TRUE again before the time has expired, the output OUT remains TRUE.
- If the RESET input is TRUE, OUT is set to FALSE.

Example:



When the timer runs the remaining runtime can be called at the output RT.

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is set to GOD. If the QUALITY input is set to TRUE, the quality status of input IN is transferred to the output OUT.

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.7.6 T_ON (BCalcTimeR4cnet) | Description

Description

The block "ON delay" sets the output to TRUE after a specific time period when the input switches to TRUE. If the input switches to FALSE, the output also switches to FALSE.

Binary time-dependent logic functions (Page 225)

Parameters

Inputs

Parame-ter	Data type	Description	Default	Maximum value	Minimum value	Con n	Edit	L	HMI
IN	BOOL	Binary input	FALSE	TRUE	FALSE	x	x	x	
TIME	FLOAT	Time period	1.0	30000.0	0.0	x	x	x	
BASE	INT	Time base Default = 0 0 = Sec. 1 = Min.	0	1	0		x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
RESET	BOOL	Reset input	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Outputs

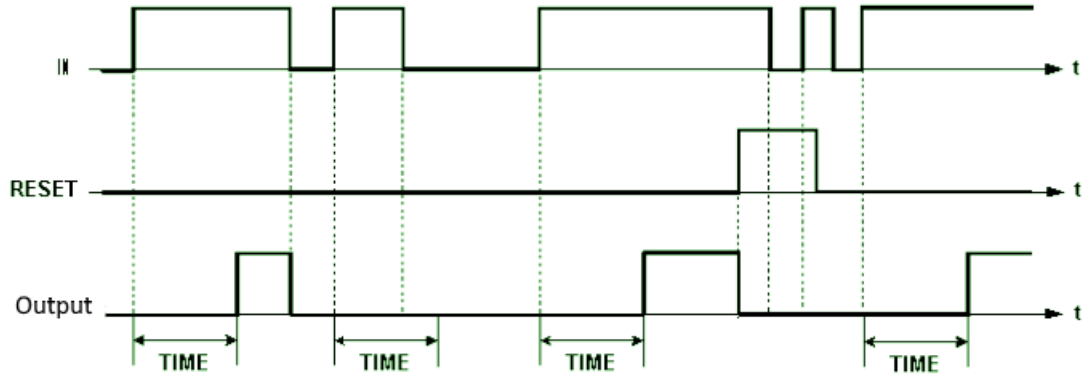
Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		
RT	FLOAT	Remaining time			x	x		x

Functional description and principle of operation

The behavior of OUT depends on the following conditions:

- If the binary input IN switches to TRUE, the output OUT only becomes TRUE after expiration of the time duration TIME.
- If IN switches to FALSE before the time has expired, OUT remains in the FALSE status.
- If IN switches a second time to TRUE before the time has expired, the time function is restarted (retriggered).
- If the RESET input is TRUE, OUT is set to FALSE.

Example:



As long as the timer runs the remaining runtime is output at the output RT.

Processing of exceptions

If the QUALITY input is set to FALSE, the quality status of OUT is set to GOD. If the QUALITY input is set to TRUE, the quality status of input IN is transferred to the output OUT.

In case of simulation the quality status of OUT is always changed to GFO.

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.8 ASignMon4cnet | Signal monitoring analog library

A.8.1 ANA (ASignMon4cnet) | Description

Description

The block "Analog input/output" can be used to connect analog values with HW proxies.

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Analog input variable	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
UL	FLOAT	High limit	100.0	3.4028235E38	-3.4028235E38	x	x	x	x
LL	FLOAT	Low limit	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
EU	STRING	Engineering units	%	20	0	x	x	x	x

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output variable			x	x		
UL_O	FLOAT	High limit			x	x		
LL_O	FLOAT	Low limit			x	x		
EU_O	STRING	Engineering units			x	x		
OUT_GQ	FLOAT	Output variable with fixed signal quality 'valid'			x	x		

Functional description and principle of operation

The analog value at the input IN is transferred to the outputs OUT and in addition to OUT_GQ, whereby the quality status is always set to GOD:

```
OUT = IN
OUT_GQ = IN
```

The other inputs are transferred to the corresponding outputs, this means:

- Input UL = Output UL
- Input LL = Output LL
- Input EU = Output EU

Processing of exceptions

The quality status of OUT_GQ is always set to GOD. The quality statuses of all other inputs are allocated to the corresponding outputs.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.8.2 ASEL2 (ASignMon4cnet) | Description

Description

The "Selection for 2 analog values" block monitors analog signals for their quality statuses and a specified deviation of the signals from each other.

The output can be as following:

- The mean value (default setting)
- The highest input value
- The lowest input value
- Input 1 or 2

Each engineering unit (EU) can be specified for an analog input range. The default setting of the EU is %.

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	FLOAT	Input signal 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	
IN2	FLOAT	Input signal 2	0.0	3.4028235E38	-3.4028235E38	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
MODE	FLOAT	Output mode Default = 3.0 1.0 = Input 1 2.0 = Input 2 3.0 = Mean value 4.0 = Higher value 5.0 = Lower value 6.0 = Mean value, higher value if message deviation input signals 7.0 = Mean value, lower value if message deviation input signals	3.0	7.0	1.0	x	x	x	x
DEV_DB	FLOAT	Deadband for input signal difference	2.0	3.4028235E38	0.0	x	x	x	x
DB	FLOAT	Deadband	1.0	3.4028235E38	0.0	x	x	x	
UL	FLOAT	High limit	100.0	3.4028235E38	-3.4028235E38	x	x	x	x
LL	FLOAT	Low limit	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
EU	STRING	Engineering units	%	20	0	x	x	x	x
BL_IN1	BOOL	Lock input signal 1	FALSE	TRUE	FALSE	x	x	x	
BL_IN2	BOOL	Lock input signal 2	FALSE	TRUE	FALSE	x	x	x	
SUB_V	FLOAT	Substitute value	0.0	3.4028235E38	-3.4028235E38	x	x	x	
SUB_VS	INT	Substitute value selection Default = 1 0 = NONE 1 = Last valid value 2 = Substitute value SUB_V 3 = Valid input value	1	3	0		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output variable			x	x	x	
O_MODE	FLOAT	Active output mode			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x	x	x
OUT_AL	BOOL	Output signal quality			x	x	x	x

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
DEV_AL	BOOL	Message deviation input signals			x	x	x	x
Q_AL1	BOOL	Message signal quality IN1			x	x		x
Q_AL2	BOOL	Message signal quality IN2			x	x		x

Functional description and principle of operation

The function selection for 3 analog values executes the following tasks:

- Checking and display of the quality statuses of the inputs
- Blocking of the inputs
- Processing and display of the deviation of input signals
- Processing of two valid inputs
- Processing of one valid input
- Processing a substitute value

An input is valid when it is not blocked and its quality status is GOD.

Checking and display of the quality statuses of the inputs

The function "Selection for analog value" calculates the output signal according to the selected output mode (MODE) and the quality statuses of the inputs. An input value is seen as valid when the quality status is GOD. If the quality status is BAD or UNC, the respective input is ignored.

An invalid input status is specified at output Q_ALn and shown in the diagnostics view.

Blocking of the inputs

An analog input INn can be blocked by the binary input BL_INn. The respective input is excluded from the calculation. A blocked input is not specifically reported.

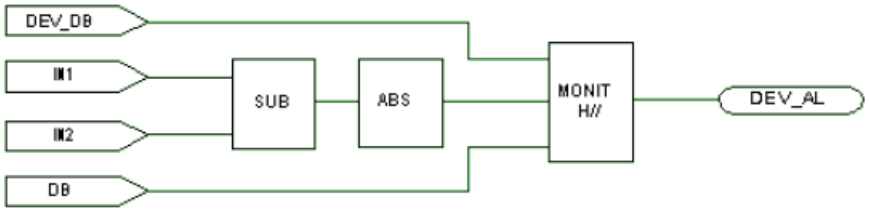
Processing and display of the deviation of input signals

A message DEV_ALn is released when the difference is exceeded between the analog inputs DEV_DB and is reset when the deviation falls below DEV_DB minus DB.

Note

If the deadband for the input signal difference DEV_DB is equal to or less than zero, the respective consistency monitoring function is switched off and all assigned output ports are set to FALSE.

The logic of the consistency check is illustrated in the following scheme:



The trigger conditions for setting and resetting the deviation message DEV_AL are displayed in the following table:

Parameter	Set condition (TRUE status)	Reset condition (FALSE status)
DEV_AL	|IN1 - IN2| >= DEV_DB	|IN1 - IN2| < (DEV_DB - DB)

With DEV_DB and DB in physical values.

Processing of two valid inputs

As long as the quality statuses of both analog inputs are GOD and no input is blocked, the output signal OUT is calculated according to the value of MODE:

MODE	Description	Value of OUT	Conditions
1.0	Input 1	IN1	-
2.0	Input 2	IN2	-
3.0 (Standard)	Mean value	(IN1+IN2)/2	-
4.0	Higher value	INx	INx > INy
5.0	Lower value	INy	INx > INy
6.0	Mean value, higher value if message deviation input signals	(IN1+IN2)/2	DEV_AL = FALSE
		Higher value	DEV_AL = TRUE
7.0	Mean value, lower value if message deviation input signals	(IN1+IN2)/2	DEV_AL = FALSE
		Lower value	DEV_AL = TRUE

One valid input

Under conditions where only one of the inputs is valid, the output depends on the selected operating mode.

Conditions	MODE	Description	Output
The quality status of an input is BAD or UNC. or An input is blocked.	1.0	Input 1	No effect, unless the selected the input is the invalid one. In this case see the substitute value below.
	2.0	Input 2	
	3.0	Mean value	The faulty signal is excluded and the output is switched to the remaining effective input.
	4.0	Higher value	
	5.0	Lower value	
	6.0	Mean value, higher value if message deviation input signals	
	7.0	Mean value, lower value if message deviation input signals	

Substitute value selection

The application of the substitute value depends on the selected operating mode.

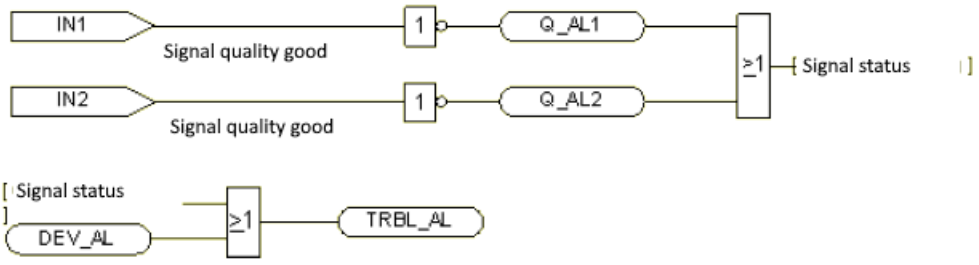
MODE	Description	Conditions
1.0	Input 1	For the selected input: The quality status is BAD or UNC. or The input is blocked.
2.0	Input 2	
3.0	Mean value	For both inputs: The quality status is BAD or UNC. or The input is blocked.
4.0	Higher value	
5.0	Lower value	
6.0	Mean value, higher value if message deviation input signals	
7.0	Mean value, lower value if message deviation input signals	

The substitute value is defined through SUB_VS:

- SUB_VS = 0
  The output is set to zero.
- SUB_VS = 1
  The output keeps the last valid value.
- SUB_VS = 2
  The output reads the substitute value SUB_V.
- SUB_VS = 3
  The output switches to the remaining valid input.
  If no valid input is available, the output is set to zero.

Processing of messages

The fault message output TRBL_AL functions as a group signal indicator. At an input quality message Q_AL or deviation message DEV_AL, the output TRBL_AL is set to TRUE.



Output signal quality

The output signal quality OUT_AL is set to TRUE when:

- The preselected input INn is available in a message signal quality (meaning Q_ALn is TRUE or INn is blocked) and SUB_VS is less than 3 (this means the next valid input is not required).
- All inputs are locked and all inputs are available in a message in signal quality.

Furthermore, OUT_AL is set to TRUE, when MODE > 5 and an entry has a deviation message.

Processing of exceptions

The quality status of OUT is usually GOD. In case of processing of a substitute value the quality status of OUT depends on the value SUB_VS:

- If SUB_VS = 0, the value of OUT is set to zero and the quality status of OUT to BAD.
- If SUB_VS is set to 1 or 2, the quality status of OUT is set to UNC.

Note

If SUB_VS = 3 and all inputs are not GOD or the last valid input is blocked, OUT is set to zero and the quality status of OUT to BAD.

In case of simulation the quality status is always changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling	Effect on further processing	Effect on the quality code
MODE < 1.0	MODE = 1.0	-
MODE > 7.0	MODE = 7.0	-
SUB_VS = 0	OUT = 0.0	OUT = BAD
SUB_VS = 1	OUT = OUT_{old}	OUT = UNC
SUB_VS = 2	OUT = SUB_V	OUT = UNC

Conditions for error handling	Effect on further processing	Effect on the quality code
SUB_VS = 3	OUT = INn (INn is GOD and BL_INn = FALSE)	-
SUB_VS = 3	OUT = 0.0 (all inputs are not GOD or are blocked)	OUT = BAD

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.8.3 ASEL3 (ASignMon4cnet) | Description

Description

The "Selection for 3 analog values" block monitors analog signals for their quality statuses and a specified deviation of the signals from each other.

The output can be the following:

- The median value (default setting)
- The mean value
- The highest input value
- The lowest input value
- Input 1, 2 or 3

Each engineering unit (EU) can be specified for an analog input range. The default setting of the EU is %.

Parameter

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1 ... IN3	FLOAT	Input signal	0.0	3.4028235E38	-3.4028235E38	x	x	x	
MODE	FLOAT	Output mode Default = 4.0 1.0 = Input 1 2.0 = Input 2 3.0 = Input 3 4.0 = Median value 5.0 = Mean value 6.0 = Highest value 7.0 = Lowest value 8.0 = Mean value with consideration of the deviation of the input signals 9.0 = Highest value with consideration of the deviation of the input signals 10.0 = Lowest value with consideration of the deviation of the input signals	4.0	10.0	1.0	x	x	x	x
DEV_DB	FLOAT	Deadband for input signal difference	2.0	3.4028235E38	0.0	x	x	x	x
DB	FLOAT	Deadband	1.0	3.4028235E38	0.0	x	x	x	
UL	FLOAT	High limit	100.0	3.4028235E38	-3.4028235E38	x	x	x	x
LL	FLOAT	Low limit	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
EU	STRING	Engineering units	%	20	0	x	x	x	x
BL_IN1 ... BL_IN3	BOOL	Block input signal	FALSE	TRUE	FALSE	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
SUB_V	FLOAT	Substitute value	0.0	3.4028235E38	-3.4028235E38	x	x	x	
SUB_VS	INT	Substitute value selection Default = 1 0 = None 1 = Last valid value 2 = Substitute value SUB_V 3 = Valid input value	1	3	0		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output variable			x	x	x	
O_MODE	FLOAT	Active output mode			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x	x	x
OUT_AL	BOOL	Output signal quality			x	x	x	x
DEV_AL	BOOL	Message deviation input signals			x	x	x	x
Q_AL1 ... Q_AL3	BOOL	Message signal quality IN1 ... IN3			x	x		x
DEV_AL1 ... DEV_AL3	BOOL	MSG difference IN1 ... In3			x	x		x

Functional description and principle of operation

The function selection for 3 analog values executes the following tasks:

- Checking and display of the quality statuses of the inputs
- Blocking of the inputs
- Processing and display of the deviation of input signals
- Processing of three valid inputs
- Processing of two valid inputs
- Processing a substitute value

An input is valid when it is not blocked and its quality status is GOD.

Checking and display of the quality statuses of the inputs

The function "Selection for analog value" calculates the output signal according to the selected output mode (MODE) and the quality statuses of the inputs. An input value is seen as valid when the quality status is GOD. If the quality status is BAD or UNC, the respective input is suspended (except if MODE MEDIAN VALUE is selected). An invalid input status is specified at output Q_ALn and shown in the diagnostics view.

Blocking of the inputs

An analog input INn can be blocked by the binary input BL_INn. The respective input is excluded from the calculation. A blocked input is not specifically reported.

Processing and display of the deviation of input signals

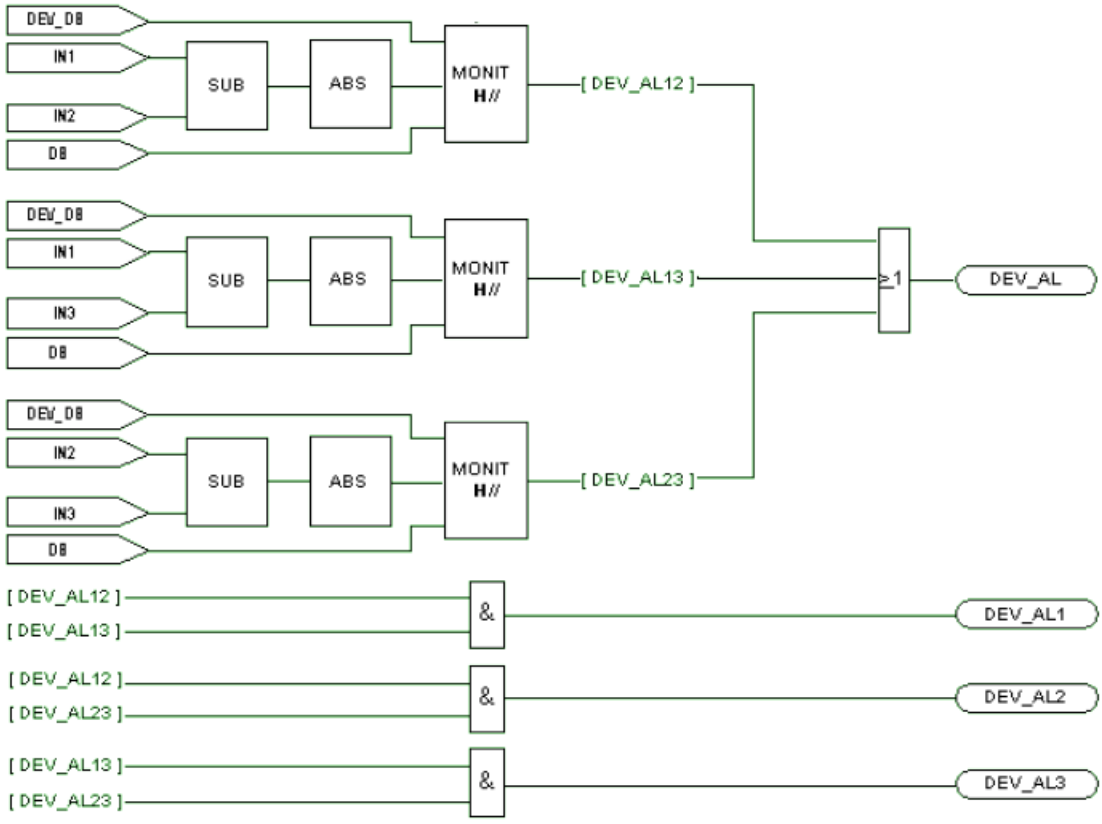
A message DEV_ALn is enabled when the difference between two analog inputs exceeds DEV_DB and is reset when the deviation falls below DEV_DB minus DB.

If an input signal INn exists in a deviation message and deviates from the two others, the assigned output port DEV_ALn is set to TRUE.

Note

If the deadband for the input signal difference DEV_DB is equal to or less than zero, the respective consistency monitoring function is switched off and all assigned output ports are set to FALSE.

The logic of the consistency check is illustrated in the following scheme:



The trigger conditions for setting and resetting a deviation message DEV_ALxy are displayed in the following table:

Parameter	Set condition (TRUE status)	Reset condition (FALSE status)
DEV_AL1	|IN1 - IN2| >= DEV_DB	|IN1 - IN2| < (DEV_DB - DB)
DEV_AL2	|IN1 - IN3| >= DEV_DB	|IN1 - IN3| < (DEV_DB - DB)
DEV_AL3	|IN2 - IN3| >= DEV_DB	|IN2 - IN3| < (DEV_DB - DB)

With DEV_DB and DB in physical values.

Three valid inputs

As long as the quality statuses of all three analog inputs are GOD and no input is blocked, the output signal OUT is calculated according to the value of MODE:

MODE	Description	Value of OUT	Conditions
1.0	Input 1	IN1	-
2.0	Input 2	IN2	-
3.0	Input 3	IN3	-
4.0	Median value	INy	INx < INy < INz
5.0	Mean value	(IN1+IN2+IN3)/3	-

MODE	Description	Value of OUT	Conditions
6.0	Highest value	INz	INx < INy < INz
7.0	Lowest value	INx	INx < INy < INz
8.0	Mean value with consideration of the deviation of the input signals	(IN1+IN2+IN3)/3	DEV_AL = FALSE
9.0	Highest value with consideration of the deviation of the input signals	INz	DEV_AL = FALSE and INx < INy < INz
10.0	Lowest value with consideration of the deviation of the input signals	INx	DEV_AL = FALSE and INx < INy < INz

Two valid inputs

Under conditions where only 2 of the inputs are valid, the output depends on the selected operating mode.

Conditions	MODE	Description	Output
The quality status of an input is BAD or UNC. or An input is blocked.	1.0	Input 1	No effect, unless the selected the input is the invalid one. In this case see the substitute value below.
	2.0	Input 2	
	3.0	Input 3	
	4.0	Median value	The output signal is still calculated as the median value of all three inputs.
	5.0	Mean value	The output signal is calculated as the mean value of the two remaining inputs.
	6.0	Highest value	The failing signal is excluded and calculation takes place with the two remaining signals.
	7.0	Lowest value	
The quality status of an input is BAD or UNC. or An input is blocked. or A deviation message is pending for an input.	8.0	Mean value with consideration of the deviation of the input signals	The output signal is calculated as the mean value of the two remaining inputs.
	9.0	Highest value with consideration of the deviation of the input signals	The failing signal is excluded and calculation takes place with the two remaining signals.
	10.0	Lowest value with consideration of the deviation of the input signals	

Substitute value selection

The application of the substitute value depends on the selected operating mode.

MODE	Description	
1.0	Input 1	For the selected input: The quality status is BAD or UNC. or The input is blocked.
2.0	Input 2	
3.0	Input 3	
4.0	Median value	For any 2 inputs: The quality status is BAD or UNC. or The input is blocked.
5.0	Mean value	
6.0	Highest value	
7.0	Lowest value	
8.0	Mean value with consideration of the deviation of the input signals	For any 2 inputs: The quality status is BAD or UNC. or The input is blocked. or The input has a deviation message.
9.0	Highest value with consideration of the deviation of the input signals	
10.0	Lowest value with consideration of the deviation of the input signals	

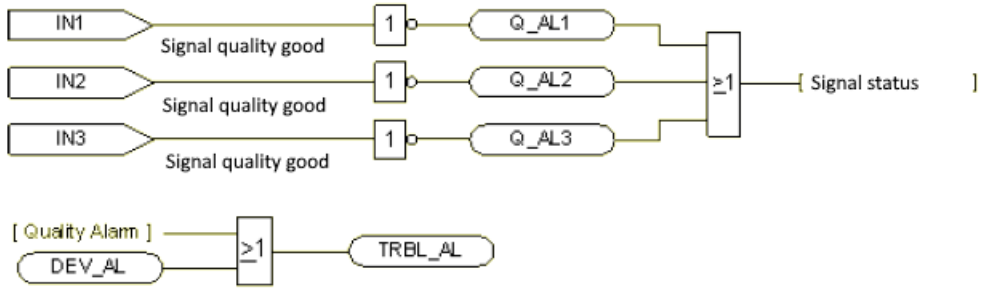
The substitute value is defined through SUB_VS:

- SUB_VS = 0
  The output is set to zero.
- SUB_VS = 1
  The output keeps the last valid value.
- SUB_VS = 2
  The output reads the substitute value SUB_V.
- SUB_VS = 3
  The output switches to the next valid and non-blocked input in the sequence: IN1 -> IN2 -> IN3 -> IN1
  If no active input is available, the output is set to zero.

Processing of messages

Fault message

The fault message output TRBL_AL functions as a group signal indicator. At an input quality message Q_AL or deviation message DEV_AL, the output TRBL_AL is set to TRUE.



Output signal quality

The output signal quality OUT_AL is set to TRUE when:

- The preselected input INn is available in a message signal quality (meaning Q_ALn is TRUE or INn is blocked) and SUB_VS is less than 3 (this means the next valid input is not required).
- Two inputs are blocked or are available in a message signal quality and SUB_VS is less than 3 (this means the last effective input is not required).
- All inputs are locked and all inputs are available in a message in signal quality.

Furthermore, OUT_AL is set to TRUE, when MODE > 7 and 2 entries have a deviation message.

Processing of exceptions

The quality status of OUT is usually GOD. In case of processing of a substitute value the quality status of OUT depends on the value SUB_VS:

- If SUB_VS = 0, the value of OUT is set to zero and the quality status of OUT to BAD.
- If SUB_VS is set to 1 or 2 the quality status of OUT is set to UNC.

Note

If SUB_VS = 3 and all inputs are not GOD or the last valid input is blocked, OUT is set to zero and the quality status of OUT to BAD.

In case of simulation the quality status is always changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling	Effect on further processing	Effect on the quality code
MODE < 1.0	MODE = 1.0	-
MODE > 10.0	MODE = 10.0	-
SUB_VS = 0	OUT = 0.0	OUT = BAD
SUB_VS = 1	OUT = OUT_{old}	OUT = UNC
SUB_VS = 2	OUT = SUB_V	OUT = UNC
SUB_VS = 3	OUT = INn (INn is GOD and BL_INn = FALSE)	-
SUB_VS = 3	OUT = 0.0 (all inputs are not GOD or are blocked)	OUT = BAD

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.8.4 ASMON (ASignMon4cnet) | Description

Description

The main application of the analog signal monitoring function is the use of up to 6 limit checks of an analog value. The analog signal monitoring also ensures smoothing (filtering) of potential fault signals.

Parameter of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Input signal	0.0	3.4028235E38	-3.4028235E38	x	x	x	
LV1	FLOAT	Limit 1	99.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV2	FLOAT	Limit 2	95.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV3	FLOAT	Limit 3	5.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV4	FLOAT	Limit 4	1.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV5	FLOAT	Limit 5	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV6	FLOAT	Limit 6	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
DB	FLOAT	Deadband absolute	0.0	3.4028235E38	0.0	x	x	x	
UL	FLOAT	High limit	100.0	3.4028235E38	-3.4028235E38	x	x	x	x
LL	FLOAT	Low limit	0.0	3.4028235E38	-3.4028235E38	x	x	x	x

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
LSG1	BOOL	Formation limit value 1 Default = True, True = Yes, False = No	TRUE	TRUE	FALSE	x	x	x	x
HI_LOW1	BOOL	Upper / lower limit value 1; True = High; False = Low	TRUE	TRUE	FALSE	x	x	x	x

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
LSG2	BOOL	Formation limit value 2; True = Yes; False = No	TRUE	TRUE	FALSE	x	x	x	x
HI_LOW2	BOOL	Upper / lower limit value 2; True = High; False = Low	TRUE	TRUE	FALSE	x	x	x	x
LSG3	BOOL	Formation limit value 3; True = Yes; False = No	TRUE	TRUE	FALSE	x	x	x	x
HI_LOW3	BOOL	Upper / lower limit value 3; True = High; False = Low	FALSE	TRUE	FALSE	x	x	x	x
LSG4	BOOL	Formation limit value 4; True = Yes; False = No	TRUE	TRUE	FALSE	x	x	x	x
HI_LOW4	BOOL	Upper / lower limit value 4; True = High; False = Low	FALSE	TRUE	FALSE	x	x	x	x
LSG5	BOOL	Formation limit value 5; True = Yes; False = No	FALSE	TRUE	FALSE	x	x	x	x
HI_LOW5	BOOL	Upper / lower limit value 5; True = High; False = Low	FALSE	TRUE	FALSE	x	x	x	x
LSG6	BOOL	Formation limit value 6; True = Yes; False = No	FALSE	TRUE	FALSE	x	x	x	x
HI_LOW6	BOOL	Upper / lower limit value 6; True = High; False = Low	FALSE	TRUE	FALSE	x	x	x	x
EU	STRING	Engineering unit	%	20	0	x	x	x	x

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
T1	FLOAT	Smoothing time in seconds	0.0	30000.0	0.0		x	x	
SUB_V	FLOAT	Substitute value	0.0	3.4028235E38	-3.4028235E38	x	x	x	
AL_SUPQ1	BOOL	Message suppression output Q1 Default = False; False = NO; True = YES	FALSE	TRUE	FALSE	x	x	x	
SUB_VS	INT	Selection substitute value Default = 1, 0 = NONE, 1 = Last valid value, 2 = Substitute value	1	2	0		x	x	x

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
LS_GQ	BOOL	Assign limit signal outputs signal quality 'valid' in a fixed manner, Default = False, False = NO, True = YES	FALSE	TRUE	FALSE	x	x	x	
AL_SUPQ2 ... AL_SUPQ6	BOOL	Message suppression output Q2 ... Q6 Default = False False = NO, True = YES	FALSE	TRUE	FALSE	x	x	x	

Parameter of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output variable			x	x	x	
Q1 ... Q6	BOOL	Limit monitor 1 ... 6			x	x	x	
CF	BOOL	Calculation error detected	x / LEAD	Came in / went out	x	x	x	x
Q_AL	BOOL	Message signal quality		Came in / went out	x	x	x	x
OUT_GQ	FLOAT	Output variable with fixed signal quality 'valid'			x	x	x	

Functional description and principle of operation

The analog signal monitoring function performs the following tasks:

- Formation of the upper/lower limit value
- Signal smoothing
- Substitute value processing
- Message suppression

Signal smoothing

The smoothing function is used to attenuate the analog signal IN. It is used as a first-level delay element and is implemented in the same way as the automation function PT1. The smoothing time is entered at port T1. A time setting of T1 = 0.0 s deactivates the smoothing function.

After smoothing the analog signal is provided at the OUT output. In addition the analog signal is provided at the output OUT_GQ, whereby the quality status is always set to GOD.

Formation of the upper/lower limit value

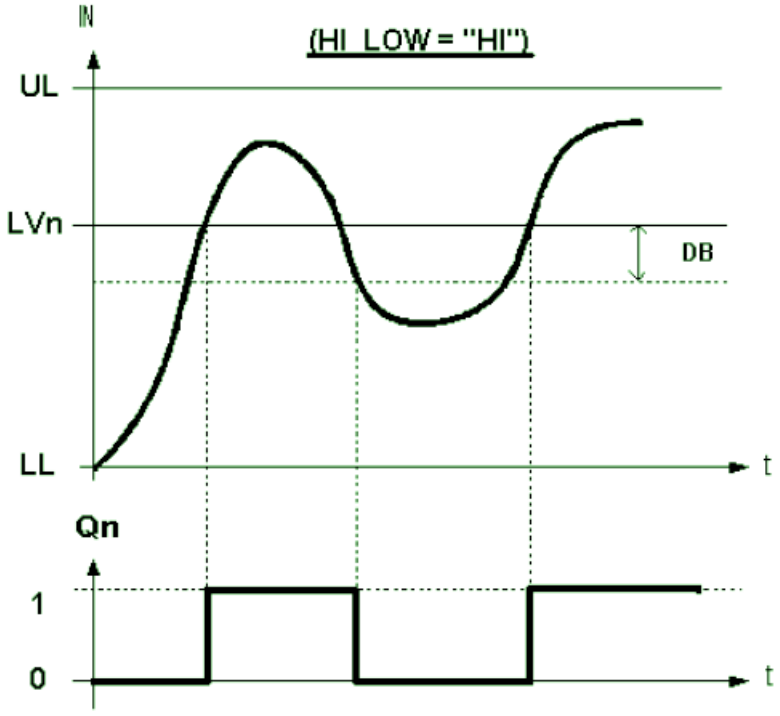
The limit value signal function monitors whether the input signal IN exceeds the limit value LVn. It shows the limit violation at the corresponding binary output Qn. It is enabled by the parameter LSGn.

The trigger direction (upper or lower limit value) can be set by means of the HI_LOWn parameter. TRUE triggers with upper limit values, FALSE with lower limit values.

a) Monitoring the UPPER LIMIT VALUE SIGNAL (HI_LOWn = HIGH)

- **Trigger condition:**
  If the input signal IN is greater than or equal to the value of LV, meaning IN >= LVn, the corresponding output Qn is set to TRUE.
- **Reset condition:**
  If the input signal IN is less than the value of LVn minus the deadband DB, meaning IN < (LVn - DB), the corresponding output Qn is set to FALSE.

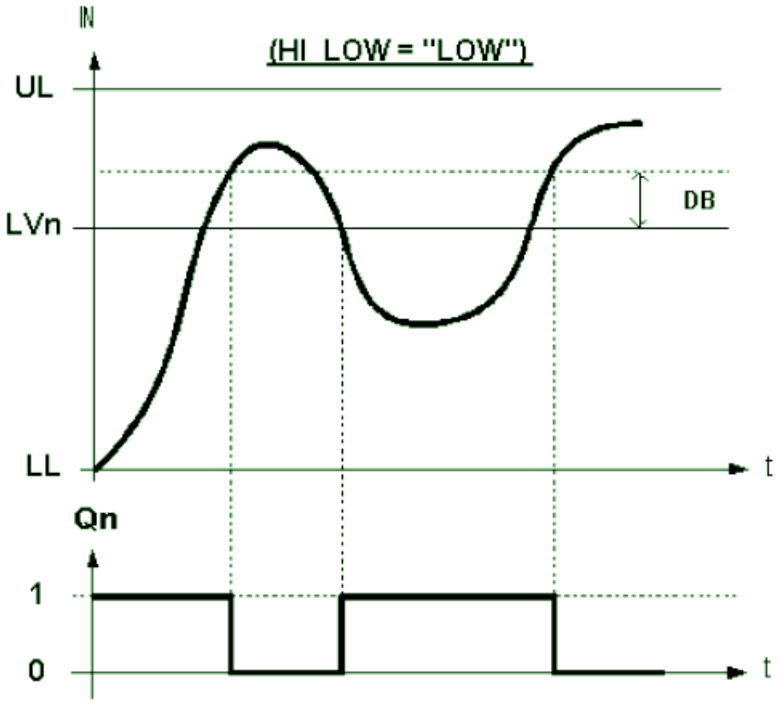
Graphical representation:



a) Monitoring the LOWER LIMIT VALUE SIGNAL (HI_LOWn = LOW)

- **Trigger condition:**
  If the input signal IN is less than or equal to the value of LVn, meaning IN <= LVn, the corresponding output Qn is set to TRUE.
- **Reset condition:**
  If the input signal IN is greater than the value of LVn plus the deadband DB, meaning IN > (LVn + DB), the corresponding output Qn is set to FALSE.

Graphical representation:



Substitute value processing

If the quality status of IN is BAD or UNC, the analog signal is considered as follows:

- Not changed (SUB_VS = 0),
- Replaced by the last input value with quality status GOD (SUB_VS = 1),
- Replaced by the substitute value SUB_V (SUB_VS = 2).

Note

The smoothing function is only applied if no substitute value was selected (SUB_VS = 0).

Message suppression

The generation of messages for the limit value signals Q1 to Q6 can be suppressed by setting the inputs AL_SUPQ1 to AL_SUPQ6. The suppression of the messages does not affect the processing of the limit signals for Q1 to Q6.

Overwriting the quality status

Setting the input LS_GQ to YES overwrites the quality statuses of all limit value outputs Q1 to Q6 with GOD.

Processing of exceptions

If T1 is smaller than Ts/2 (except T1 = 0.0), T1 is set internally to Ts/2, meaning no smoothing takes place (here Ts is the programmed runtime cycle). The quality status of OUT_GQ is always set to GOD. Normally OUT has the quality status GOD.

If the quality status of IN is BAD or UNC, the quality status is only transferred to the OUT output if SUB_VS = 0. If SUB_VS = 1 or 2, the quality status of OUT is replaced by UNC.

If LS_GQ is FALSE, then the quality status of the outputs Q1 to Q6 is equal to the quality status of OUT. If LS_GQ = TRUE, then the quality status of the outputs Q1 to Q6 is always set to GOD.

If the quality status of IN is BAD or UNC, the output Q_AL is set to TRUE. If any value exceeds the float value range during the calculation, the quality status is set to BAD and the calculation error output CF is set to TRUE; OUT keeps the last value.

In case of simulation the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling	Effect on further processing	Effect on the quality code
T1 < Ts/2	T1 is set internally to Ts/2.	-
Overflow during the calculation	CF = TRUE	OUT = BAD
SUB_VS = 0	OUT = IN	-
SUB_VS = 1	OUT = OUT_{old}	OUT = UNC
SUB_VS = 2	OUT = SUB_V	OUT = UNC

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.8.5 ASMON_R (ASignMon4cnet) | Description

Description

The ASMON_R block monitors redundant analog inputs (transmitters) for their quality statuses and specific deviation from each other. The output of the algorithm can either be the maximum input value (MAX) or the minimum input value (MIN). In addition ASMON_R checks at least 1 and at most 6 preconfigured limit values. Values outside the upper or lower limit values (adjustable) are displayed as corresponding output ports (with message functions). The analog signal monitoring also ensures smoothing (filtering) of potential fault variables.

Parameter of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1 ... IN2	FLOAT	Input signal	0.0	3.4028235E38	-3.4028235E38	x	x	x	
LV1	FLOAT	Limit 1	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV2	FLOAT	Limit 2	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV3	FLOAT	Limit 3	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV4	FLOAT	Limit 4	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV5	FLOAT	Limit 5	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
LV6	FLOAT	Limit 6	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
DB	FLOAT	Deadband absolute	0.5	3.4028235E38	0.0	x	x	x	
UL	FLOAT	High limit	100.0	3.4028235E38	-3.4028235E38	x	x	x	x
LL	FLOAT	Low limit	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
EU	STRING	Engineering unit	%	20	0	x	x	x	x

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
LSG1	BOOL	Formation limit value 1 Default = True, True = Yes, False = No	FALSE	TRUE	FALSE	x	x	x	
HI_LOW1	BOOL	Upper / lower limit value 1; True = High; False = Low	TRUE	TRUE	FALSE	x	x	x	x
LSG2	BOOL	Formation limit value 2; True = Yes; False = No	FALSE	TRUE	FALSE	x	x	x	
HI_LOW2	BOOL	Upper / lower limit value 2; True = High; False = Low	TRUE	TRUE	FALSE	x	x	x	x
LSG3	BOOL	Formation limit value 3; True = Yes; False = No	FALSE	TRUE	FALSE	x	x	x	
HI_LOW3	BOOL	Upper / lower limit value 3; True = High; False = Low	FALSE	TRUE	FALSE	x	x	x	x
LSG4	BOOL	Formation limit value 4; True = Yes; False = No	FALSE	TRUE	FALSE	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
HI_LOW4	BOOL	Upper / lower limit value 4; True = High; False = Low	FALSE	TRUE	FALSE	x	x	x	x
LSG5	BOOL	Formation limit value 5; True = Yes; False = No	FALSE	TRUE	FALSE	x	x	x	
HI_LOW5	BOOL	Upper / lower limit value 5; True = High; False = Low	FALSE	TRUE	FALSE	x	x	x	x
LSG6	BOOL	Formation limit value 6; True = Yes; False = No	FALSE	TRUE	FALSE	x	x	x	
HI_LOW6	BOOL	Upper / lower limit value 6; True = High; False = Low	FALSE	TRUE	FALSE	x	x	x	x

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
MINMAX	BOOL	Selection of the minimum/maximum value, Default = True, True = Maximum, False = Minimum	TRUE	TRUE	FALSE	x	x	x	
T1	FLOAT	Smoothing time [s]	0.0	30000.0	0.0		x	x	
DEV_DB	FLOAT	Deadband for input signal difference	1.0	3.4028235E38	0.0	x	x	x	
T_MON	FLOAT	Delay time input signal consistency monitoring [s]	1.0	30000.0	0.0	x	x	x	
SUB_V	FLOAT	Substitute value	0.0	3.4028235E38	-3.4028235E38	x	x	x	
SUB_VS	INT	Selection substitute value; 0 = NONE; 1 = Last valid value; 2 = Substitute value	1	2	0		x	x	x
AL_SUPQ1 ... AL_SUPQ6	BOOL	Message suppression output Q1 ... Q6 False = NO, True = YES	FALSE	TRUE	FALSE	x	x	x	
LS_GQ	BOOL	Assign limit signal outputs signal quality 'valid' in a fixed manner; False = NO; True = YES	FALSE	TRUE	FALSE	x	x	x	

Parameter of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output variable			x	x	x	
OUT_IN1 ... OUT_IN2	BOOL	Output equivalent to IN1 ... IN2			x	x		
Q1 ... Q6	BOOL	Limit monitor 1 ... 6			x	x	x	
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x	x	x
OUT_AL	BOOL	Output signal quality		Came in / went out	x	x	x	x
DEV_AL	BOOL	Message deviation input signals			x	x	x	x
Q_AL1 ... Q_AL2	BOOL	Message signal quality IN1 ... IN2			x	x	x	x
CF	BOOL	Calculation error detected			x	x	x	x
OUT_GQ	FLOAT	Output variable with fixed signal quality 'valid'			x	x	x	

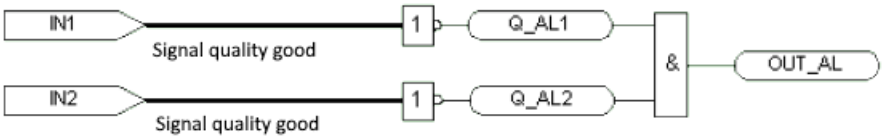
Functional description and principle of operation

The redundant analog signal monitoring function performs the following tasks:

- Checking and display of the quality statuses of the inputs
- Selection of the minimum/maximum value
- Processing and display of the deviations of inputs (with absolute deadband for input signal difference DEV_DB)
- Signal smoothing
- Formation of upper/lower limit value (with absolute deadband for limit values DB)
- Substitute value processing
- Message suppression

Checking and display of the quality statuses of the inputs

The algorithm ASMON_R processes the output according to the selected output mode (value of MINMAX) and the quality status of the inputs. An input value INn is usually considered, meaning INn is valid if the quality status is valid, meaning the quality status is GOD. If the quality status is not GOD (BAD or UNC), the algorithm ignores the respective input IN. An invalid input status is displayed each time at the assigned output port Q_ALn of the respective point. If the quality status of both inputs is furthermore not GOD, the message output OUT_AL is set to TRUE.



Selection of the minimum/maximum value

If both inputs are valid, meaning both quality statuses are GOD, the output value OUT is created according to the following rule:

- If MINMAX is FALSE, meaning MIN is selected, OUT corresponds to the minimum value of the inputs.
- If MINMAX is TRUE, meaning MAX is selected, OUT corresponds to the maximum value of the inputs.

Furthermore the selected input is specified at any time at the corresponding output OUT_INn.

The trigger conditions and effects on the corresponding outputs are summarized in the following table:

MINMAX	Condition	Value of OUT	Value of OUT_IN1	Value of OUT_IN2
FALSE (MIN)	IN1 <= IN2	IN1	TRUE	FALSE
FALSE (MIN)	IN1 > IN2	IN2	FALSE	TRUE
TRUE (MAX)	IN1 >= IN2	IN1	TRUE	FALSE
TRUE (MAX)	IN1 < IN2	IN2	FALSE	TRUE

Processing and display of the deviation of input signals

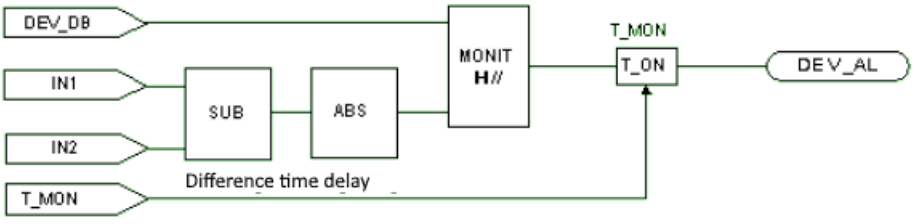
The algorithm of the ASMON_R function monitors the deviation between the analog inputs when both inputs are valid:

- If the input IN1 deviates from IN2 by more than DEV_DB, the delay time T_MON for consistency monitoring is started. When the time has expired, the DEV_AL output is set to TRUE.

  Note

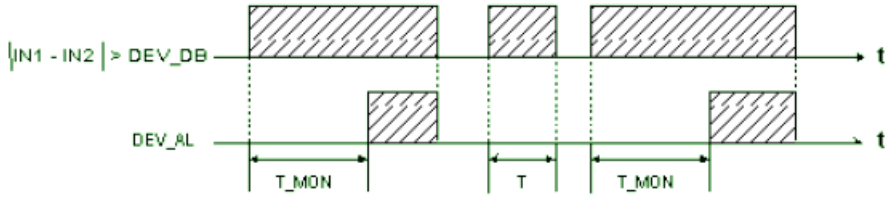
  If the deadband for the input signal difference DEV_DB is equal to or greater than zero, the consistency monitoring function is switched off and all assigned outputs are set to FALSE.

The principle of processing of the deviation of inputs then illustrated in the following representation:



The trigger conditions for setting and resetting DEV_AL are displayed in the following table or in the following scheme:

Parameter	Set condition (TRUE status)	Reset condition (FALSE status)
DEV_AL	| IN1 - IN2 | >= DEV_DB and T_MON expired	| IN1 - IN2 | < DEV_DB



Signal smoothing

The smoothing function is used for additional attenuation of the analog signals IN 1 and IN2. It is used as a first-level delay element and is implemented in the same way as the automation function PT1. The smoothing time is specified by means of port T1. A smoothing time of T1 = 0.0 s deactivates the smoothing function.

After smoothing the analog signal is provided to the OUT output depending on the parameter MINMAX. In addition the analog signal is provided at the output OUT_GQ, whereby the quality status is always set to GOD.

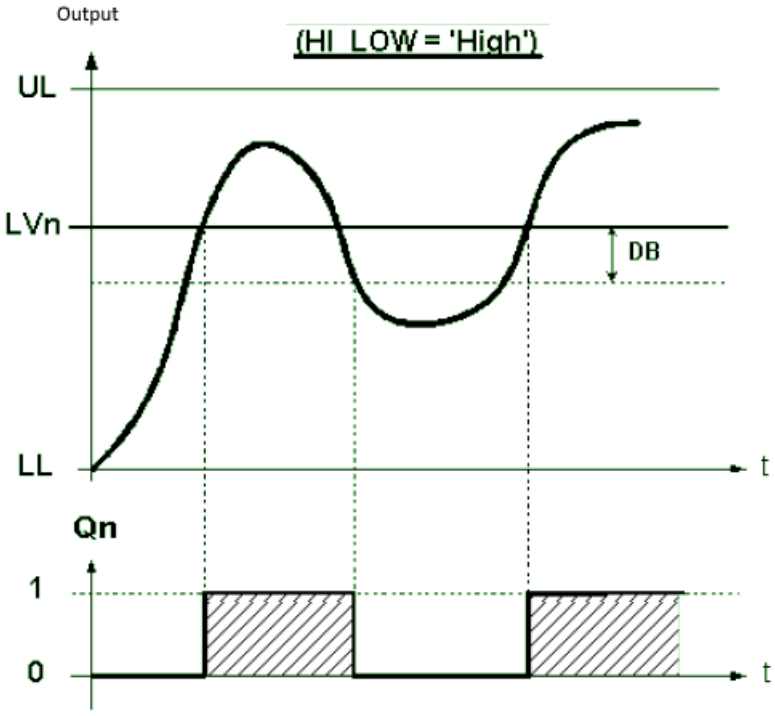
Formation of the upper/lower limit value

The function formation upper/lower limit value monitors whether a configured limit value LVn is exceeded and shows the limit violation at the corresponding binary output Qn. It is enabled by the parameter LSGn. The upper limit value is set via the parameter HI_LOW = High. The lower limit value is set via the parameter HI_LOW = Low. The triggering and resetting of a limit value signal is also specified via the deadband DB.

a) Monitoring the UPPER LIMIT VALUE SIGNAL (HI_LOWn = HIGH)

- **Trigger condition:**
  If the output Out has a value greater than or equal to LVn, the corresponding binary output Qn is set to TRUE.
- **Reset condition:**
  If the output signal OUT is less than LVn minus deadband DB, the corresponding binary output Qn is set to FALSE.

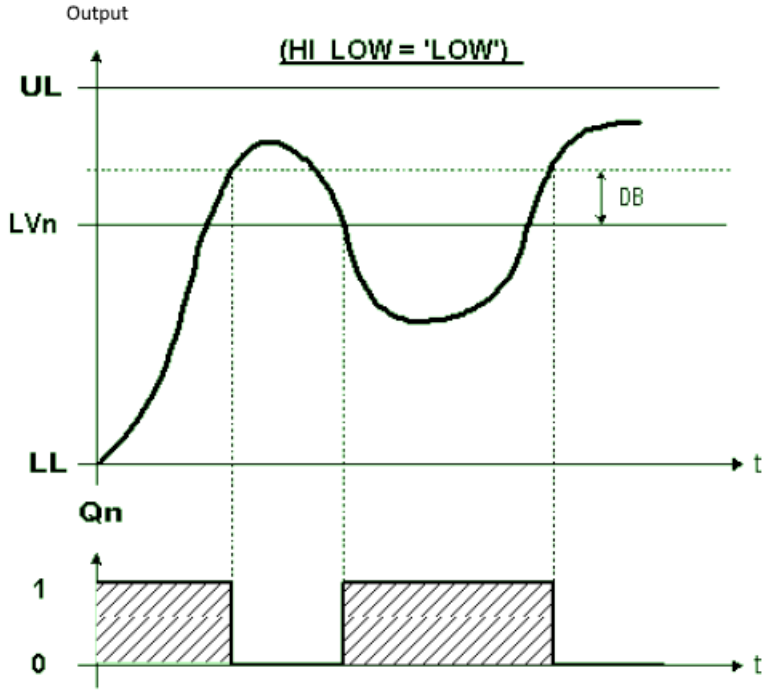
Graphical representation:



a) Monitoring the LOWER LIMIT VALUE SIGNAL (HI_LOWn = LOW)

- **Trigger condition:**
  If the output OUT has a value less than or equal to LVn the corresponding digital output Qn is set to TRUE.
- **Reset condition:**
  If the output signal OUT is larger than LVn plus deadband DB, the corresponding binary output Qn is set to FALSE.

Graphical representation:



If no limit value formation was selected (LSG = NO) the corresponding output Qn is set to FALSE.

Substitute value processing

If the quality statuses of both analog inputs IN are BAD or UNC, the analog signal OUT is:

- Not changed (SUB_VS = 0),
- Replaced by the last valid value (SUB_VS = 1),
- Replaced by the substitute value SUB_V (SUB_VS = 2).

Note

The smoothing function is only applied if no substitute value was selected (SUB_VS = 0).

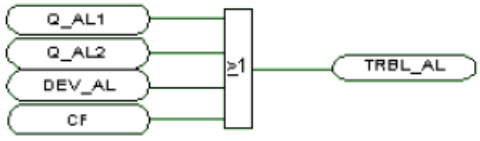
Message suppression

The generation of messages for the limit value signals Q1 to Q6 can be suppressed by setting the inputs AL_SUPQ1 to AL_SUPQ6. The suppression of the messages does not affect the processing of the limit signals for Q1 to Q6.

Processing of messages

The fault message output TRBL_AL functions as a group signal indicator of the ASMON_R function. If a quality message of an input or a message on the deviation of input signals or a calculation error occurs, the output TRBL_AL is set to TRUE.

The implementation of the fault message is illustrated in the following representation:



Overwriting the quality status

Setting the input LS_GQ to YES overwrites the quality statuses of all limit value outputs Q1 to Q6 with GOD.

Processing of exceptions

If T1 is less than Ts/2 (except T1 = 0.0), T1 is set internally to Ts/2, meaning no smoothing (here Ts is the programmed runtime cycle).

The quality status of OUT_GQ is always set to GOD. Normally OUT has the quality status GOD.

If the quality status of both analog inputs, IN1 and IN2, is BAD or UNC, the quality status depends on the mode of SUB_VS:

- If SUB_VS = 0 the value and the quality status of the last selected input are supplied to the output port OUT.
- If SUB_VS = 1 or 2, the quality status of OUT is replaced by UNC.

If LS_GQ has the status FALSE, the quality statuses of the outputs Q1 to Q6 are equal to the quality status of OUT. If LS_GQ is TRUE, the quality statuses of the outputs Q1 to Q6 are always set to GOD.

If a value exceeds the float value range during the calculation, the quality status is set to BAD, the calculation error output CF is set to TRUE and OUT keeps the last value.

In case of simulation the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling	Effect on further processing	Effect on the quality code
T1 < Ts/2	T1 is set internally to Ts/2.	-
Overflow during the calculation	CF = TRUE	OUT = BAD
SUB_VS = 0	OUT = INn	OUT = INn
SUB_VS = 1	OUT = OUT_{old}	OUT = UNC
SUB_VS = 2	OUT = SUB_V	OUT = UNC

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.9 BSignMon4cnet | Signal monitoring binary library

A.9.1 BIN (BSignMon4cnet) | Description

Description

The BIN block can be used to connect binary values with hardware proxies.

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	BOOL	Binary input value	FALSE	TRUE	FALSE	x	x	x	x

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Output variable			x	x		
OUT_GQ	BOOL	Output variable with fixed signal quality 'valid'			x	x		

Functional description and principle of operation

The binary value at the input IN is transferred to the output OUT and also OUT_GQ, whereby the quality status is always set to GOD:

```
OUT = IN
OUT_GQ = IN
```

Processing of exceptions

The quality status of IN is transferred to the output OUT. The quality status of OUT_GQ is always set to GOD.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.9.2 BSMON (BSignMon4cnet) | Description

Description

The binary signal monitoring function processes and monitors individual and changeover contacts.

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	BOOL	Binary input 1 (NO contact)	FALSE	TRUE	FALSE	x	x	x	
IN2	BOOL	Binary input 2 (NO contact)	TRUE	TRUE	FALSE	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
T_MON	FLOAT	Delay time contact monitoring [s]	0.2	30000.0	0.0		x	x	
SUB_VS	INT	Selection substitute value; 0 = NONE; 1 = Last valid value 2 = FALSE; 3 = TRUE	0	3	0		x	x	
SUBV_OUTN	INT	Replacement value OUTN; 0 = Always FALSE; 1 = Always the negated value of AUSG	0	1	0		x	x	
AL_SUP	BOOL	Message suppression	FALSE	TRUE	FALSE	x	x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x	x	
OUTN	BOOL	Negated output			x	x		
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x	x	x
BOTH_T	BOOL	Both inputs FALSE			x	x		x
BOTH_T	BOOL	Both inputs TRUE			x	x		x
OUT_GQ	BOOL	Output variable with fixed signal quality 'valid'			x	x		
OUTN_GQ	BOOL	Negated output signal quality 'valid'			x	x		

Functional description and principle of operation

The binary signal monitoring function performs the following tasks:

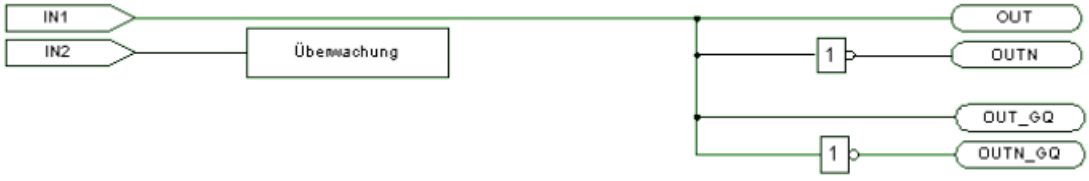
- Processing of changeover contacts
- Monitoring of changeover contacts
- Processing of contact points
- Substitute value processing
- Message suppression

The signal at input IN1 is provided to the output OUT and in addition to the output OUT_GQ, where the quality status is always set to GOD. A negated value is provided to OUTN or OUTN_GQ.

Processing of changeover contacts

The binary signal monitoring function processes an input signal pair, meaning a changeover contact, and generates a suitable binary output signal OUT. The status of the IN1 input is decisive for the generation of the output signal OUT. The negated status of OUT is transferred to the output OUTN.

The second input IN2 is only used for monitoring purposes.



Monitoring of changeover contacts

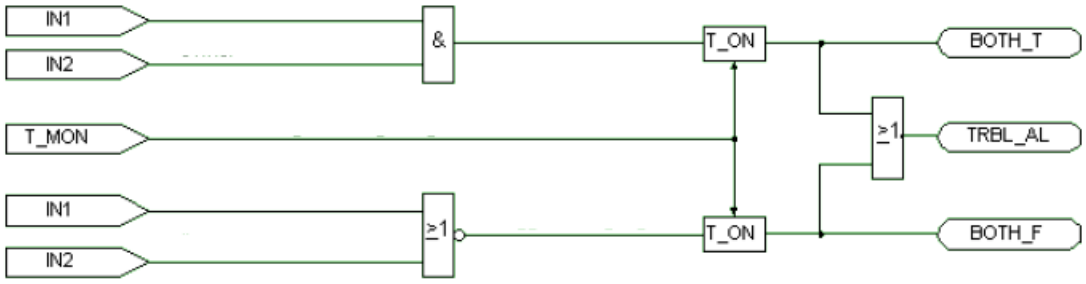
The inputs are monitored for plausibility, meaning depending on the equivalent values of IN1 and IN2 the fault message output TRBL_AL is set to TRUE when the monitoring delay time T_MON has expired. The corresponding binary output BOTH_F or BOTH_T is also set to TRUE for detailed information in the diagnostics view.

The behavior of the monitoring function depends on the following conditions:

- If both inputs are TRUE (trigger condition), the time duration T_MON is started. When the time has expired, the BOTH_T output becomes TRUE.
- If both inputs are FALSE (trigger condition), the time duration T_MON is started. When the time has expired, the BOTH_F output becomes TRUE.
- If the trigger conditions are FALSE, the corresponding outputs BOTH_T or BOTH_F are set to FALSE.

The fault message output TRBL_AL functions as a group signal indicator of the binary signal monitoring function. If the quality status of the inputs is BAD or UNC, the output TRBL_AL is set to TRUE. In this case the plausibility check is switched off and the outputs BOTH_T and BOTH_F are set to FALSE. If TRBL_AL is set to TRUE, further processing of OUT depends on the value of SUB_VS and the processing of OUTN on the value of SUBV_OUTN (see substitute value processing further below).

The implementation of the plausibility check is illustrated in the following scheme



Note

The T_MON monitoring time must be greater than the changeover time of the contact.

Processing of contact points

The binary signal monitoring function only processes a connected input signal, meaning a contact point, and generates a suitable binary output signal OUT. In this case the status of the connected input INx, meaning IN1 or IN2, is decisive for the generation of the output signal OUT. The negated status of OUT is also provided at the output OUTN.



If the quality status of the connected input is BAD or UNC, the output TRBL_AL is set to TRUE. In this case further processing of OUT depends on the value of SUB_VS and the processing of OUTN depends on the value of SUBV_OUTN (see substitute value processing further below).

Substitute value processing

If the fault message output TRBL_AL is TRUE, the processing of OUT and OUTN can be controlled by the selection of the substitute value SUB_VS and SUBV_OUTN.

- OUT is:
  - Not changed (SUB_VS = 0),
  - The last output value with quality status GOD (SUB_VS = 1)
  - FALSE (SUB_VS = 2),
  - TRUE (SUB_VS = 3).
- OUTN is:
  - FALSE (SUBV_OUTN = 0),
  - The negated value of OUT (SUBV_OUTN = 1).

Message suppression

By setting the input AL_SUP, the generation of messages for the outputs OUT, OUTN, OUT_GQ and OUTN_GQ can be suppressed.

The message suppression does not affect the signal processing at the outputs OUT, OUTN, OUT_GQ and OUTN_GQ.

Processing of exceptions

The quality status of OUT_GQ and OUTN_GQ are always set to GOD. Usually the quality statuses of OUT and OUT_N are also GOD. When processing a substitute value, the quality status of OUT depends on the value of SUB_VS. If SUB_VS = 0, the value and quality status of OUT correspond to input IN1 when both inputs are connected. If only one input is connected, the value and the quality status of OUT correspond to the input INx. In every case the quality status of OUTN is equal to the quality status of OUT.

In case of simulation the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling	Effect on further processing	Effect on the quality code
SUB_VS = 0	OUT = IN1 or IN (a connected input x)	OUT = IN1 or INx
SUB_VS = 1	OUT = OUT_{old}	OUT = UNC
SUB_VS = 2	OUT = FALSE	OUT = UNC
SUB_VS = 3	OUT = TRUE	OUT = UNC
SUBV_OUTN = 0	OUTN = FALSE	OUTN = OUT
SUBV_OUTN = 1	OUTN = OUT	OUTN = OUT

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.9.3 BSMON_R (BSignMon4cnet) | Description

Description

The BSMON_R block monitors two redundant binary inputs for their quality statuses and a deviation from each other.

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN1	BOOL	Binary input 1	FALSE	TRUE	FALSE	x	x	x	
IN2	BOOL	Binary input 2	TRUE	TRUE	FALSE	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
T_MON	FLOAT	Delay time input signal consistency monitoring [s]	0.2	30000.0	0.0		x	x	
SUB_VS	INT	Selection substitute value; 0 = NONE; 1 = Last valid value 2 = FALSE; 3 = TRUE	0	3	0		x	x	
AL_SUP	BOOL	Message suppression	FALSE	TRUE	FALSE	x	x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x	x	
OUT_IN1 ... OUT_IN2	BOOL	Output equivalent to IN1 ... IN2			x	x		
TRBL_AL	BOOL	Fault message	x / LEAD	Came in / went out	x	x	x	x
OUT_AL	BOOL	Output signal quality			x	x	x	x
DEV_AL	BOOL	Message deviation input signals			x	x	x	
Q_AL1 ... Q_AL2	BOOL	Message signal quality IN1 ... IN2			x	x	x	x
OUT_GQ	BOOL	Binary signal output with fixed signal quality 'valid'			x	x		

Functional description and principle of operation

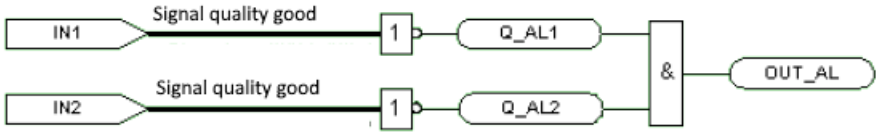
The binary signal monitoring function performs the following tasks:

- Checking and display of the quality statuses of the inputs
- Output processing
- Processing and display of the deviation of an input

- Substitute value processing
- Message suppression

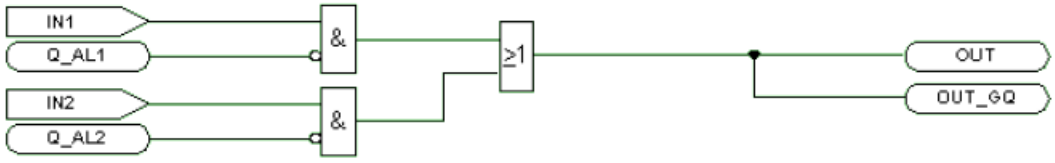
Checking and display of the quality statuses of the inputs

The algorithm BSMON_R processes the output depending on the quality status of the inputs. An input value INn is usually considered, meaning INn is valid if the quality status is valid, meaning the quality status is GOD. If the quality status is not GOD, the algorithm ignores the respective input IN. An invalid input status is displayed at the assigned output Q_ALn. If the quality status of both inputs is not GOD, the message output OUT_AL is set to TRUE. In this case further processing of OUT depends on the value SUB_VS.



Output processing

The function BSMON_R processes the redundant input signals and generates a suitable binary output signal OUT for further processing. If both inputs are active, the output OUT is always TRUE as long as only one output is TRUE. If OUT_AL is TRUE, further processing of OUT depends on the value SUB_VS.



The actual value of OUT is provided at the output port OUT_GQ, whereby the quality status is always set to GOD.

OUT_IN1 is TRUE, when IN1 is the source for OUT: OUT_IN2 is TRUE, when IN2 is the source for OUT:

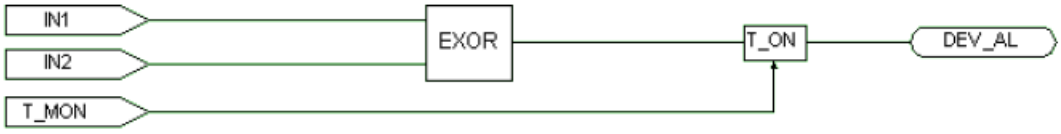
Processing and display of the deviation of an input

The algorithm of BSMON_R monitors the deviation of the two valid inputs. If IN1 is not equal IN2, the delay time for the consistency monitoring is started. The output to the message of the deviation DEV_AL is not set to TRUE until the T_MON has expired.

The behavior of the consistency monitoring function depends on the following conditions:

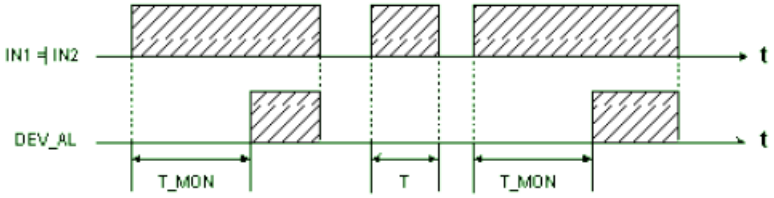
- If IN1 is not equal to IN2 (trigger condition), the duration T_MON is started. If the time has expired and the trigger condition is still TRUE, the output DEV_AL becomes TRUE.
- If the trigger condition is FALSE, the corresponding output DEV_AL is set to FALSE.

The principle of processing of the deviation of inputs then illustrated in the following representation:



The trigger conditions for setting and resetting DEV_AL are displayed in the following table or in the following scheme:

Parameter	Set condition (TRUE status)	Reset condition (FALSE status)
DEV_AL	IN1 >< IN2 and T_MON expired	IN1 = IN2



Substitute value processing

By selecting the substitute value SUB_VS, further processing of OUT can be assigned when both inputs are not GOD, meaning OUT_AL is TRUE. Depending on SUB_VS the output OUT takes on the following value:

- Not changed (SUB_VS = 0),
- Last valid value (SUB_VS = 1),
- FALSE (SUB_VS = 2),
- TRUE (SUB_VS = 3).

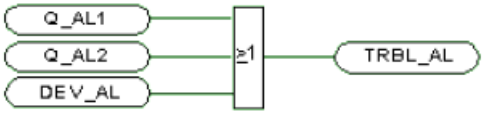
Message suppression

The generation of messages for the output OUT can be suppressed by setting the inputs AL_SUP. The message suppression does not affect the signal processing at the output OUT.

Processing of messages

The fault message output TRBL_AL functions as a group signal indicator of the BSMON_R function. At an input quality message or deviation message the output TRBL_AL is set to TRUE.

The logic of the fault message is illustrated in the following scheme:



Processing of exceptions

The quality status of OUT_GQ is always GOD. Normally the quality status of OUT is also GOD. When processing a substitute value, the quality status of OUT depends on the value of SUB_VS. If SUB_VS = 0 the value and the quality status of the last selected input INn are supplied to the output OUT. If SUB_VS > 0, OUT sets the quality status to UNC.

In case of simulation the quality status of OUT is changed to GFO.

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling	Effect on further processing	Effect on the quality code
SUB_VS = 0	OUT = INn	OUT = INn
SUB_VS = 1	OUT = OUT old	OUT = UNC
SUB_VS = 2	OUT = FALSE	OUT = UNC
SUB_VS = 3	OUT = TRUE	OUT = UNC

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.10 SysFcts4cnet | System function library

A.10.1 ASQMON (SysFcts4cnet) | Description

Description

The ASQMON block checks the quality status of an analog signal and provides the result at the corresponding binary outputs. In addition the processing of a substitute variable is offered, if the quality status of the signal is not GOD. The analog signal is provided in the output OUT.

System functions (Page 226)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Analog input	0.0	3.4028235E38	-3.4028235E38	x	x	x	
SUB_V	FLOAT	Output substitute value	0.0	3.4028235E38	-3.4028235E38	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
SUB_QS	BOOL	Output signal quality substitute value; False = Uncertain; True = Valid	FALSE	TRUE	FALSE		x	x	
SUB_VS	INT	Selection substitute value; 0 = NONE; 1 = Last valid value 2 = Substitute value SUB_V	0	2	0		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Analog output			x	x		
QG	BOOL	Signal quality VALID			x	x		
QB	BOOL	Signal quality INVALID			x	x		
QNG	BOOL	Signal quality NOT VALID			x	x		

Functional description and principle of operation

Note

This automation function cannot be used in an S7 runtime container.

Quality monitoring

The input signal IN is checked with regard to its quality status:

- If the quality status is GOD, the output QG is set to TRUE, QB and QNG to FALSE.
- If the quality status is BAD, the outputs QB and QNG are set to TRUE and QG to FALSE.
- If the quality status is neither GOD nor BAD, thus UNC, the output QNG is set to TRUE, QB and QG to FALSE.

Truth table:

Quality status of IN	QG	QB	QNG
GOD	1	0	0
BAD	0	1	1
UNC	0	0	1

Substitute value processing

Usually the analog signal IN is transferred to the output OUT. The selection of the substitute value SUB_VS can determine the value of OUT when the quality status of IN is not GOD.

Depending on SUB_VS, OUT takes on the following value:

- Not changed (SUB_VS = 0),
- Last valid value (SUB_VS = 1),
- Value of SUB_V (SUB_VS = 2)

In addition the quality status can be influenced by OUT, if the substitute value processing is activated (this means SUB_VS > 0). The quality status of OUT is dependent on SUB_QS:

- UNC, when SUB_QS is FALSE (default)
- GOD, if SUB_QS is TRUE (with the secondary status local_override or simulated)

Processing of exceptions

Normally OUT is equal to IN and the quality outputs have the status GOD. When processing a substitute value the quality status of OUT depends on the value of SUB_VS:

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling	Effect on further processing	Effect on the quality code
SUB_VS = 0	OUT = IN	-
SUB_VS = 1; SUB_QS = FALSE	OUT = OUT_{old}	OUT = UNC
SUB_VS = 1; SUB_QS = TRUE	OUT = OUT_{old}	OUT = GFO
SUB_VS = 2; SUB_QS = FALSE	OUT = SUB_V	OUT = UNC
SUB_VS = 2; SUB_QS = TRUE	OUT = SUB_V	OUT = GFO

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.10.2 BSQMON (SysFcts4cnet) | Description

Description

The block BSQMON checks the quality status of a binary signal and provides the result at the corresponding binary outputs. In addition the processing of a substitute variable is offered, if the quality status of the signal is not GOD. The binary signal is provided in the output OUT.

System functions (Page 226)

Parameter

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	BOOL	Binary input	FALSE	TRUE	FALSE	x	x	x	
SUB_V	BOOL	Output substitute value	FALSE	TRUE	FALSE	x	x	x	
SUB_QS	BOOL	Output signal quality substitute value; False = Uncertain; True = Valid	FALSE	TRUE	FALSE		x	x	
SUB_VS	INT	Selection substitute value; 0 = NONE; 1 = Last valid value 2 = Substitute value SUB_V	0	2	0		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Binary output			x	x		
QG	BOOL	Signal quality VALID			x	x		
QB	BOOL	Signal quality INVALID			x	x		
QNG	BOOL	Signal quality NOT VALID			x	x		

Functional description and principle of operation

Note

This automation function cannot be used in an S7 runtime container.

Quality monitoring

The input signal IN is checked with regard to its quality status:

- If the quality status is GOD, the output QG is set to TRUE, QB and QNG to FALSE.
- If the quality status is BAD, the outputs QB and QNG are set to TRUE and QG to FALSE.
- If the quality status is neither GOD nor BAD, thus UNC, the output QNG is set to TRUE, QB and QG to FALSE.

Truth table:

Quality status of IN	QG	QB	QNG
GOD	1	0	0
BAD	0	1	1
UNC	0	0	1

Substitute value processing

Usually the binary signal IN is transferred to the output port OUT. The selection of the substitute value SUB_VS can determine the value of OUT when the quality status of IN is not GOD (is indicated by the quality status BAD or UNC).

Depending on SUB_VS, OUT takes on the following value:

- Not changed (SUB_VS = 0),
- Last valid value (SUB_VS = 1),
- Value of SUB_V (SUB_VS = 2)

In addition the quality status can be influenced by OUT, if the substitute value processing is activated (this means SUB_VS > 0). The quality status of OUT is dependent on SUB_QS:

- UNC, when SUB_QS is FALSE (default)
- GOD, if SUB_QS is TRUE (with the secondary status local_override or simulated)

Processing of exceptions

Normally OUT is equal to IN and the quality outputs have the status GOD. When processing a substitute value the quality status of OUT depends on the value of SUB_VS:

The trigger conditions for error handling and the effects on further processing are summarized in the following table:

Conditions for error handling	Effect on further processing	Effect on the quality code
SUB_VS = 0	OUT = IN	-
SUB_VS = 1; SUB_QS = FALSE	OUT = OUT_{old}	OUT = UNC
SUB_VS = 1; SUB_QS = TRUE	OUT = OUT_{old}	OUT = GFO
SUB_VS = 2; SUB_QS = FALSE	OUT = SUB_V	OUT = UNC
SUB_VS = 2; SUB_QS = TRUE	OUT = SUB_V	OUT = GFO

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.10.3 OP_ANA (SysFcts4cnet) | Description

Description

The output is set to the analog value configured in the faceplate.

System functions (Page 226)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
M_OP	FLOAT	Manual release	0.0	3.4028235E38	-3.4028235E38				x
LL	FLOAT	Operation low limit	-100	3.4028235E38	-3.4028235E38	x	x	x	x
UL	FLOAT	Operation high limit	100	3.4028235E38	-3.4028235E38	x	x	x	x
EU	STRING	Engineering units	%	20	0	x	x	x	x

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	

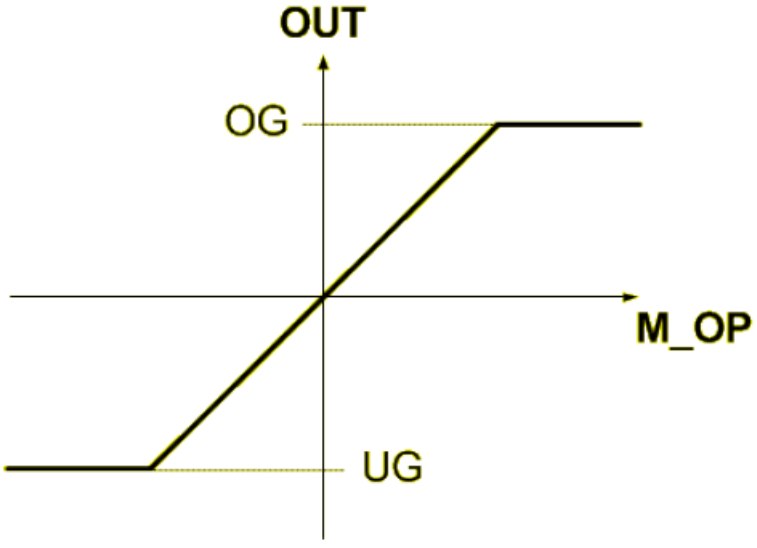
Functional description and principle of operation

The output is set to the analog value configured in the faceplate.

The output value is calculated according to the following formula:

Condition	Value of OUT
LL < M_OP < UL	M_OP
M_OP <= LL	LL
M_OP >= UL	UL

Graphical representation:



Processing of exceptions

Not applicable

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.10.4 OP_BIN (SysFcts4cnet) | Description

Description

The OP_BIN block delivers the signal TRUE in the output during the duration of a runtime container cycle when the input signal of the faceplate switches to TRUE.

System functions (Page 226)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
M_OP	BOOL	Manual release	FALSE	TRUE	FALSE				x

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Output value			x	x	x	

Functional description and principle of operation

OP_BIN delivers the signal TRUE in the output during the duration of a runtime container cycle when the input signal of the faceplate switches to TRUE.

Processing of exceptions

Not applicable

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.11 CONV4cnet | Signal converter library

A.11.1 CONV_AT (CONV4cnet) | Description

Description

Triggers a message when an integer values corresponds to the defined specified value.

Parameter of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	INT	Input value	0	2147483647	-2147483648	x	x	x	x
TRG_VAL	INT	Trigger value	0	2147483647	-2147483648		x	x	x
INV	BOOL	Inverter trigger	FALSE	TRUE	FALSE		x	x	x
QUALITY	BOOL	Processing of the quality status	TRUE	TRUE	FALSE		x	x	x

Parameter of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	BOOL	Message triggered	x / ALARM		x	x	x	

Functional description and principle of operation

If the value of IN is equal to the value of TRG_VAL, the output OUT is set to TRUE.

If the INV input is set to TRUE, the value of OUT is inverted.

The following table shows the values of OUT:

Condition	INV	OUT
IN = TRG_VAL	FALSE	TRUE
IN <> TRG_VAL	FALSE	FALSE
IN = TRG_VAL	TRUE	FALSE
IN <> TRG_VAL	TRUE	TRUE

The time stamp of the input IN is transferred to the OUT output.

Processing of exceptions

If the input QUALITY has the value TRUE, the quality status of the input IN is transferred to the output OUT. If the input QUALITY has the value FALSE, the quality status of the output OUT is always GOD.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.11.2 CONV_B2F (CONV4cnet) | Description

Description

Converts 16 bits in a FLOAT value.

Parameter of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
DI01_PV ... DI16_PV	BOOL	Input value	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	TRUE	TRUE	FALSE		x	x	x

Parameter of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
FLOAT_OUT	FLOAT	Output value			x	x		x

Functional description and principle of operation

The output FLOAT_OUT can display of a value of 0..65535.

The time stamp of the FLOAT_OUT input is output during the calculation.

Processing of exceptions

If the input QUALITY has the value TRUE, the quality status of the input DIxx_PV is transferred to the output FLOAT_OUT. If the input QUALITY has the value FALSE, the quality status of the output FLOAT_OUT is always GOD.

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.11.3 CONV_B2I (CONV4cnet) | Description

Description

Converts 16 bits into an integer value.

Parameter of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN_1 ... IN_32	BOOL	Input value	FALSE	TRUE	FALSE	x	x	x	x
QUALITY	BOOL	Processing of the quality status	TRUE	TRUE	FALSE		x	x	x

Parameter of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	INT	Output value			x	x	x	

Functional description and principle of operation

The time stamp of the output OUT is output during the calculation.

Processing of exceptions

If the input QUALITY has the value TRUE, the quality status of the input IN_xx is transferred to the output OUT. If the input QUALITY has the value FALSE, the quality status of the output OUT is always GOD.

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.11.4 CONV_F2B (CONV4cnet) | Description

Description

Converts a FLOAT value into 16 bits

Parameter of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FLOAT_IN	FLOAT	Input value	0.0	65535	0	x	x	x	
QUALITY	BOOL	Processing of the quality status	TRUE	TRUE	FALSE		x	x	x

Parameter of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
DO01_PV ... DO16_PV	BOOL	Output value			x	x		x

Functional description and principle of operation

The time stamp of the input FLOAT_IN is transferred to the outputs during the value change.

Processing of exceptions

If the input QUALITY has the value TRUE, the quality status of the input FLOAT_IN is transferred to the output OUT. If the input QUALITY has the value FALSE, the quality status of the output DOxx_PV is always GOD.

If the value of the input FLOAT_IN is less than 0, the outputs are set to FALSE. If the value of the input FLOAT_IN is greater than 65535, the outputs are set to TRUE and receive the quality status "BAD".

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.11.5 CONV_F2I (CONV4cnet) | Description

Description

Converts a FLOAT value into an integer value.

Parameter of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	FLOAT	Input value	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
QUALITY	BOOL	Processing of the quality status	TRUE	TRUE	FALSE		x	x	x

Parameter of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	INT	Output value			x	x	x	

Functional description and principle of operation

If the value at the input IN lies outside the limit values (INT), the value of the output OUT corresponds to the maximum or minimum (INT).

The time stamp of the input IN is transferred to the output.

Processing of exceptions

If the input QUALITY has the value TRUE, the quality status of the input IN is transferred to the output OUT. If the input QUALITY has the value FALSE, the quality status of the output OUT is always GOD.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.11.6 CONV_I2B (CONV4cnet) | Description

Description

Converts an integer value into 32 bits.

Parameter of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	INT	Input value	0	2147483647	-2147483648	x	x	x	x
QUALITY	BOOL	Processing of the quality status	TRUE	TRUE	FALSE		x	x	x

Parameter of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT_1 ... OUT_32	BOOL	Output value			x	x	x	

Functional description and principle of operation

The time stamp of the input IN is transferred to the outputs during the value change.

Processing of exceptions

If the input QUALITY has the value TRUE, the quality status of the input IN is transferred to the output OUT_xx. If the input QUALITY has the value FALSE, the quality status of the output OUT_xx is always GOD.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.11.7 CONV_I2F (CONV4cnet) | Description

Description

Converts an integer value into a FLOAT value

Parameter of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IN	INT	Input value	0	2147483647	-2147483648	x	x	x	x
QUALITY	BOOL	Processing of the quality status	TRUE	TRUE	FALSE		x	x	x

Parameter of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
OUT	FLOAT	Output value			x	x	x	

Functional description and principle of operation

The time stamp of the input IN is transferred to the output.

Processing of exceptions

If the input QUALITY has the value TRUE, the quality status of the input IN is transferred to the output OUT. If the input QUALITY has the value FALSE, the quality status of the output OUT is always GOD.

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.12 BB4cnet | Devices.BlackBox-Library

A.12.1 CN_BB_AI8 (BB4cnet) | Description

Description

The block CN_BB_AI8 receives 8 analog values in 32 bits from a black box system.

S7PutGet (Black Box) (Page 226)

Parameter

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_BB_COMM	BYTE	Connection of BB_COMM	BBLNK	10	10	x	x	x	
OFFSET	INT	Byte offset in the interface	2	5968	0	x	x	x	
LEN	INT	Quantity of the signals	8	8	1	x	x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	
AI00_sEU ... AI07_sEU	STRING	Specification engineering unit (EU)	%	10 AI04_sEU: 20	0		x	x	
AI00_sLL ... AI07_sLL	FLOAT	Specification low limit	0.0	3.4028235E38	-3.4028235E38		x	x	
AI00_sUL ... AI07_sUL	FLOAT	Specification high limit	0.0	3.4028235E38	-3.4028235E38		x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
AI00_PV ... AI07_PV	FLOAT	Process value 0 ... 7			x	x	x	
AI00_EU ... AI07_EU	STRING	Engineering unit (EU)				x	x	
AI00_LL ... AI07_LL	FLOAT	Low limit				x	x	
AI00_UL ... AI07_UL	FLOAT	High limit				x	x	

Functional description and principle of operation

CN_BB_AI8 receives 8 analog values in 32 bits from a black box system which are specified through the inputs FROM_BB_COM and OFFSET. Subsequently the upper and lower limit value are checked and the quality code is added.

The input OFFSET specifies where the first of the two bytes is placed in the receive telegram (interface).

If the input TIME_STAMP is set to TRUE, the process variables are provided with time stamps.

If the input QUALITY_CODE is set to TRUE, the process variables are provided with quality codes.

Signal types and areas

The following signal types and areas are available:

- AIxx_sEU: The engineering units for the signal (also available at output AIxx_EU).
- AIxx_sLL and AIxx_sUL: Low and high limit of the signal in engineering units (also available at the output AIxx_LL and AIxx_UL).
- AIxx_PV: The current process variable of the input signal.

Structure of the data block

Byte offset	Description
0	Amount of used process values, will be compared with the LEN parameter
1..4	Analog Process Value 1, will be routed to AI00_PV
5..8	Digital Process Value 2, will be routed to AI01_PV
...	
LEN*4+1	Quality code for AI00_PV if QUALITY_CODE = True
LEN*4+2	Quality code for AI01_PV if QUALITY_CODE = True
...	

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)

Parameter tables (Page 223)

A.12.2 CN_BB_AO8 (BB4cnet) | Description

Description

The CN_BB_AO8 block sends 8 analog values in 32 bits to a black box system.

S7PutGet (Black Box) (Page 226)

Parameter

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_BB_COMM	BYTE	Connection of BB_COMM	BBLNK	10	10	x	x	x	
OFFSET	INT	Byte offset in the interface	2	5968	0	x	x	x	
LEN	INT	Quantity of the signals	8	8	1	x	x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	
AO00_PV ... AO07_PV	FLOAT	Process value 0 ... 7	0.0	3.4028235E38	-3.4028235E38	x	x	x	

Functional description and principle of operation

CN_BB_AO8 sends 8 analog values in 32 bits to a black box system which is specified through the inputs FROM_BB_COMM and OFFSET.

The input OFFSET specifies where the first of the two bytes is placed in the receive telegram (interface).

The input AOxx_PV acquires the analog signal to be sent.

Data buffer usage

Byte offset	Description
0	LEN
1..4	Input port AO00_PV
5..8	Input port AO01_PV
...	
29..32	Input port AO07_PV
33	Q-Code input port AO00_PV
34	Q-Code input port AO01_PV
...	
40	Q-Code input port AO07_PV

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)

Parameter tables (Page 223)

A.12.3 CN_BB_COMM (BB4cnet) | Description

Description

The block CN_BB_COMM establishes the connection to a black box system and manages the data exchange.

S7PutGet (Black Box) (Page 226)

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
IP_ADDRESS1 ... IP_ADDRESS2	STRING	IP address. Remote CP		64	0	IP_ADDRESS1	x	x	
RACK1 ... RACK2	INT	Rack no. of the remote CPU 1 ... 2	RACK1: 0 RACK2: 1	1	0		x	x	
SLOT1 ... SLOT2	INT	Slot no. of the remote CPU 1 ... 2	3	18	1		x	x	
READ_DB	INT	Read DB number in the black box	1	16383	1		x	x	
READ_DBB	INT	Byte offset for reading (only even)	0	65532	0		x	x	
READ_LEN	INT	Number of bytes to be read	0	6000	0		x	x	
WRITE_DB	INT	DB number in the black box for writing	1	16383	1		x	x	
WRITE_DBB	INT	Byte offset for reading (only even)	0	65532	0		x	x	
WRITE_LEN	INT	Number of bytes to be written	0	6000	0		x	x	
T_MON_LB	FLOAT	Monitoring time for life beat counter [s]	1.0	60.0	0.1		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error or sign of life error	x / LEIT	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Switchover to detect further messages	x / LEIT	Came in / went out	x	x		x
TO_BB_RD_WR	BYTE	Connection to BB READ/WRITE modules			x	x		x

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TCP_CONN_STATE1 ... TCP_CONN_STATE2	INT	Connection status connection 1 ... 2; Default = 2; 1 = Connected; 2 = Not connected; 3 = Connecting; 4 = Disconnecting			x			x
TP_LAYER_STATE1 ... TP_LAYER_STATE2	INT	TP layer status 1 ... 2; Default = 3; 0 = Connected; 1 = Connecting; 2 = Disconnecting; 3 = Not connected			x			x

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
APPL_LAYER_STATE1 ... APPL_LAYER_STATE2	INT	Application layer status 1 ... 2; Default = 0; 0 = TPDU negotiated; 1 = TPDU being negotiated; 2 = Unknown			x			x
LB_FAULT_CON1 ... LB_FAULT_CON2	BOOL	Sign of life error connection 1 ... 2			x			x
READ_ACCESS_RESULT1 ... READ_ACCESS_RESULT2	INT	Return value of the remote S7 for reading; Default = 0; 0 = Unknown; 1 = Hardware error; 3 = Access violation; 5 = Invalid address; 6 = Unknown type; 7 = Inconsistent object; 10 = Unknown object; 15 = Config error; 16 = No-load operation; 32 = Unknown; 255 = Successful			x			x

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
WRITE_ACCESS_ RESULT1 ... WRITE_ACCESS_ RESULT2	INT	Return value of the remote S7 for writing; 1 Default = 0; 0 = Unknown; 1 = Hardware error: 3 = Access violation; 5 = Invalid address; 6 = Unknown type; 7 = Inconsistent object; 10 = Unknown object; 15 = Config error; 16 = No-load operation; 32 = Unknown; 255 = Successful			x			x
DATA_TRANSFER_ DELAY	BOOL	Send and receive tasks are not concluded within the instance cycle			x	x		x
NUMBER_OF_TELEGRAMS	FLOAT	Number of all transferred telegrams via an S7 connection			x	x		x

Functional description and principle of operation

CN_BB_COMM manages a singular or redundant connection to the black box system. Here the inputs x_DB, x_DBB and x_LEN specify the read / write interface in the black box system. If an input x_LEN is set to zero, the assigned function is not called up. The interface block starts with the sign of life counter.

The inputs IP_ADDRESSx contain the IP addresses of the black box. The inputs RACKx and SLOTx show the installation location of the black box CPU.

The output TO_BB_RD_WR defines the connection to the write and read modules of the black box.

The output TRANSFER_TIME_FAULT shows whether the time for the exchange of the data exceeds the available time of the black box partner in the plug-in.

Note

Avoiding a timeout

A timeout during the data exchange can be avoided by reducing the data volume or extending the cycle time.

Message and diagnostics

The output TRBL_AL is TRUE if one of the following outputs does not have the setpoint state:

Parameter	Desired state
TCP_CONN_STATEx	Connected
TP_LAYER_STATEx	Connected
APPL_LAYER_STATEx	TPDU (Transport Protocol Data Unit) negotiated
READ_ACCESS_RESULTx	Successful
WRITE_ACCESS_RESULTx	Successful
LB_FAULT_CONx	FALSE

MULTIPLE_AL is TRUE, if TRBL_AL has the status TRUE and is acknowledged and a new error occurs.

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)

Parameter tables (Page 223)

A.12.4 CN_BB_DI8 (BB4cnet) | Description

Description

The block CN_BB_DI8 receives eight bits from a black box system.

S7PutGet (Black Box) (Page 226)

Parameters

Inputs

Parame-ter	Data type	Description	Default	Maximum value	Minimum value	Con n	Edit	L	HMI
FROM_BB _COMM	BYTE	Connection of BB_COMM	BBLINK	10	10	x		x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
OFFSET	INT	Byte offset in the interface	2	5999	0		x	x	
LEN	INT	Quantity of the signals	8	8	1		x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
DI00_PV ... DI07_PV	BOOL	Process value			x	x	x	

Functional description and principle of operation

CN_BB_DI8 reads 8 bytes from a black box system which are a specified through the inputs FROM_BB_COMM, OFFSET and LEN. The quality code is added.

LEN specifies the number of signals to be read.

The input OFFSET specifies where the first of the two bytes is placed in the receive telegram (interface).

The output DIxx_PV delivers the binary signal.

Structure of the data block

Byte offset	Description
0	Amount of used process values, will be compared with the LEN parameter
1.0	Digital Process Value 1, will be routed to DI00_PV
1.1	Digital Process Value 2, will be routed to DI01_PV
...	
2	Quality code for DI00_PV if QUALITY_CODE = True
3	Quality code for DI01_PV if QUALITY_CODE = True
...	

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)

Parameter tables (Page 223)

A.12.5 CN_BB_DO8 (BB4cnet) | Description

Description

The block CN_BB_DO8 sends eight bits to a black box system.

S7PutGet (Black Box) (Page 226)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_BB_COMM	BYTE	Connection of BB_COMM	BBLINK	10	10	x		x	
OFFSET	INT	Byte offset in the interface	2	5999	0		x	x	
LEN	INT	Quantity of the signals	8	8	1		x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	
DO00_PV ... DO7_PV	BOOL	Process value 0 ... 15	FALSE	TRUE	FALSE	x	x	x	

Functional description and principle of operation

CN_BB_DO8 sends 8 binary signals in one byte to a black box system which are specified through the inputs FROM_BB_COMM, OFFSET and LEN.

LEN specifies the number of signals to be sent.

The input OFFSET specifies where the first of the two bytes is placed in the receive telegram (interface).

The input DOxx_PV acquires the binary signal to be sent.

Structure of the data block

Byte offset	Description
0	LEN
1.0	Input port DO00_PV
1.1	Input port DO01_PV
...	
1.7	Input port DO07_PV
2	Q-Code input port DO00_PV
3	Q-Code input port DO01_PV
...	
9	Q-Code input port DO07_PV

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)

Parameter tables (Page 223)

A.12.6 CN_BB_II8 (BB4cnet) | Description

Description

The block CN_BB_II8 receives eight integer values with the associated quality code of an S7-4xx.

S7PutGet (Black Box) (Page 226)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_BB_COMM	BYTE	Connection of BB_COMM	BBLNK	10	10	x	x	x	
OFFSET	INT	Byte offset in the interface	2	5968	0	x	x	x	
LEN	INT	Quantity of the signals	8	8	1	x	x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
IO00_PV ... IO07_PV	INT	Process value 0 ... 7			x	x		

Functional description and principle of operation

CN_BB_II8 receives eight integer values of an S7-4xx, which are specified through the inputs FROM_BB_COM and OFFSET.

Input OFFSET specifies from which address the information is to be copied out of the DB.

LEN specifies the number of signals to be read.

If the input QUALITY_CODE is set to TRUE, the process variables are provided with quality codes.

Data buffer usage

Byte offset	Description
0	LEN
1..4	Input port IO00_PV
5..8	Input portIO01_PV
...	
29..32	Input port IO07_PV
33	Q-Code input port IO00_PV
34	Q-Code input port IO01_PV
...	
40	Q-Code input port IO07_PV

See also

Black box | Network configuration (Page 228)
Black box | Programming (Page 228)
General quality statuses (Page 223)
Parameter tables (Page 223)

A.12.7 CN_BB_IO8 (BB4cnet) | Description

Description

The block CN_BB_IO8 sends eight integer values with the associated quality code to an S7-4xx.

S7PutGet (Black Box) (Page 226)

Parameter

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_BB_COMM	BYTE	Connection of BB_COMM	BBLNK	10	10	x	x	x	
OFFSET	INT	Byte offset in the interface	2	5968	0	x	x	x	
LEN	INT	Quantity of the signals	8	8	1	x	x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	
IO00_PV ... IO07_PV	BOOL	Process value 0 ... 7	FALSE	TRUE	FALSE	x	x	x	

Functional description and principle of operation

CN_BB_IO8 sends eight integer values of an S7-4xx, which are specified through the inputs FROM_BB_COM and OFFSET.

Input OFFSET specifies from which address the information is to be copied into the DB.

LEN specifies the number of signals to be sent.

The input IOxx_PV acquires the signal to be sent.

Data buffer usage

Byte offset	Description
0	LEN
1..4	Input port IO00_PV
5..8	Input port IO01_PV
...	
29..32	Input port IO07_PV
33	Q-Code input port IO00_PV
34	Q-Code input port IO01_PV
...	
40	Q-Code input port IO07_PV

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)

Parameter tables (Page 223)

A.12.8 CN_BB_STRI (BB4cnet) | Description

Description

The CN_BB_STRI block receives a string with the associated quality code of an S7-4xx.

S7PutGet (Black Box) (Page 226)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_BB_COMM	BYTE	Connection of BB_COMM	BBLNK	10	10	x	x	x	
OFFSET	INT	Byte offset in the interface	2	5968	0	x	x	x	
MAX_LEN	INT	Maximum length of the STRING	64	64	1	x	x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
STR00_PV	STRING	Process value			x	x		

Functional description and principle of operation

CN_BB_STRI receives a string with the associated quality code of an S7-4xx, which are specified through the inputs FROM_BB_COM and OFFSET.

Input OFFSET specifies from which address the information is to be copied from the DB.

MAX_LEN specifies the length of the character string. The character string must have a maximum length of 64 characters.

If the input QUALITY_CODE is set to TRUE, the process variables are provided with quality codes.

Data buffer usage

Byte offset	Description
0	LEN
1	strlen STR00PV
2 - LEN+2	Input port STR00_PV
LEN+2+1	Q-Code output port STR00_PV

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)

Parameter tables (Page 223)

A.12.9 CN_BB_STRO (BB4cnet) | Description

Description

The block CN_BB_STRO sends a string with the associated quality code to an S7-4xx.

S7PutGet (Black Box) (Page 226)

Parameter

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_BB_COMM	BYTE	Connection of BB_COMM	BBLNK	10	10	x	x	x	
OFFSET	INT	Byte offset in the interface	2	5999	0	x	x	x	
MAX_LEN	INT	Maximum length of the STRING	64	64	1	x	x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	
STR00_PV	STRING	Process value	%	32	0	x	x	x	x

Functional description and principle of operation

CN_BB_STRO sends eight integer values of an S7-4xx, which are specified through the inputs FROM_BB_COM and OFFSET.

Input OFFSET specifies from which address the information is to be copied into the DB.

MAX_LEN specifies the length of the character string. The character string must have a maximum length of 64 characters.

The input STR00_PV acquires the signal to be sent.

Data buffer usage

Byte offset	Description
0	LEN
1	strlen STR00PV
2 - LEN+2	Input port STR00_PV
LEN+2+1	Q-Code input port STR00_PV

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)
Parameter tables (Page 223)

A.12.10 CN_BB_STRI32x8 (BB4cnet) | Description

Description

The block CN_BB_STRI32x8 receives eight strings with the associated quality code of an S7-4xx.

S7PutGet (Black Box) (Page 226)

Parameter

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_BB_COMM	BYTE	Connection of BB_COMM	BBLNK	10	10	x	x	x	
OFFSET	INT	Byte offset in the interface	2	5968	0	x	x	x	
LEN	INT	Number of strings	8	8	1	x	x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	

Outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
STR00_PV ... STR07_PV	STRING	Process value			x	x	x	

Functional description and principle of operation

CN_BB_STRI32x8 receives eight strings with the associated quality code from an S7-4xx, which are specified through the inputs FROM_BB_COM and OFFSET.

Input OFFSET specifies from which address the information is to be copied from the DB.

LEN specifies the number of strings to be read. A string must have maximum length of 32 characters.

If the input QUALITY_CODE is set to TRUE, the process variables are provided with quality codes.

Data buffer usage

Byte offset	Description
0	LEN
1	strlen STR00PV
2 - 33	Input port STR00_PV
34	strlen STR01PV
35 - 66	Input port STR01_PV
...	
LEN*33+1	Q-Code input port STR00_PV
LEN*33+2	Q-Code input port STR01_PV
...	Q-Code input port STR07_PV

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)

Parameter tables (Page 223)

A.12.11 CN_BB_STRO32x8 (BB4cnet) | Description

Description

The block CN_BB_STRO32x8 sends a string with the associated quality code to an S7-4xx.

S7PutGet (Black Box) (Page 226)

Parameters

Inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_BB_COMM	BYTE	Connection of BB_COMM	BBLNK	10	10	x	x	x	
OFFSET	INT	Byte offset in the interface	2	5999	0	x	x	x	
LEN	INT	Number of strings	8	8	1	x	x	x	
QUALITY_CODE	BOOL	Value with quality code	TRUE	TRUE	FALSE	x	x	x	
STR00_PV ... STR07_PV	STRING	Process value		32	0	x	x	x	

Functional description and principle of operation

CN_BB_STRO32x8 sends eight strings with the associated quality code to an S7-4xx, which are specified through the inputs FROM_BB_COM and OFFSET.

Input OFFSET specifies from which address the information is to be copied into the DB.

LEN specifies the number of strings to be read. A string must have maximum length of 32 characters.

The input STRxx_PV acquires the signal to be sent.

Data buffer usage

Byte offset	Description
0	LEN
1	strlen STR00PV
2 - 33	Input port STR00_PV
34	strlen STR01PV
35 - 66	Input port STR01_PV
...	
LEN*33+1	Q-Code input port STR00_PV
LEN*33+2	Q-Code input port STR01_PV
...	Q-Code input port STR07_PV

See also

Black box | Network configuration (Page 228)

Black box | Programming (Page 228)

General quality statuses (Page 223)

Parameter tables (Page 223)

A.13 TCON4cnet | Devices.TCON-Library

A.13.1 Blocks

A.13.1.1 TCON_COMM (TCON4cnet) | Description

Description

The block TCON_COMM contains all required parameters for establishing a connection and for data exchange with an S7 CPU. The IP addresses are specified as a string (e.g. "123.4.5.67").

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
ID1 ... ID2	INT	Connection ID (E.g. "ID1": ID of the first connection)	0	2147483647	0	x		x	
IPAddress1 ... IPAddress2	STRING	IP address (E.g. "IPAddress1": IP of the first connection)	0	24	0	x		x	
Port1 ... Port2	INT	Port-ID (E.g. "Port1": Port-ID of the first connection)	1	65535	0	x		x	
LocalTSAP1 ... LocalTSAP2	INT	Local TSAP (E.g. "LocalTSAP1": Local TSAP of the first connection)	0	2147483647	0	x			
RemoteTSAP1 ... RemoteTSAP2	INT	Remote TSAP (E.g. "RemoteTSAP1": Remote TSAP of the first connection)	0	2147483647	0	x		x	
ConnectionType1... ConnectionType2	INT	Connection type; 0: Native; 1: IsoOnTcp (E.g. "ConnectionType1": Type of the first connection)	0	1	0	x		x	
Timeout1 ... Timeout2	INT	Timeout (E.g. "Timeout1": Timeout of the first connection)	3	100	1	x		x	
Connect	BOOL	Accept a connection	TRUE	TRUE	FALSE	x		x	
Active	BOOL	Active for the client, passive for the server	TRUE	TRUE	FALSE	x		x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TO_TCON_RD_WR	BYTE	Value for the transfer			x	x		
Connected	BOOL	"Connected" status			x	x		
Connected1 ... Connected2	BOOL	"Connected" status for the respective connection			x	x		

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
Error1 ... Error2	BOOL	Error message for the respective connection			x	x		
Status1 ... Status2	INT	Status of the respective connection			x	x		

Functions

Monitoring communication (Page 403)

Establishing a connection (Page 402)

Configuring a connection (Page 402)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.13.1.2 TCON_AO32 (TCON4cnet) | Description

Description

Sends up to 32 analog structures to another CPU using an Industrial Ethernet connection.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_TCON_COMM	BYTE	Connection to communication block	TCON			x		x	
R_ID	INT	Telegram connection ID	0	2147483647	-2147483647	x		x	
RdSysStEn	BOOL	Activates the redundancy for the system (plant network)	FALSE	TRUE	FALSE		x	x	
RedCPU	BOOL	Activates the redundancy for the CPU	FALSE	TRUE	FALSE		x	x	
SndEn	BOOL	1 = Send continuously	FALSE	TRUE	FALSE	x		x	
Min_Time	INT	Minimum wait time	0	2147483647	-2147483647	x		x	
Max_Time	INT	Maximum wait time	0	2147483647	-2147483647	x		x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
PVHys_in00 ... PVHys_in31	FLOAT	EDC hysteresis of process value PV_In[00..31]	0.0	3.4028235E38	-3.4028235E38	x [0..15]		x	
PV_In00 ... PV_In31	FLOAT	Input parameters [0..31]	0.0	3.4028235E38	-3.4028235E38	x [0..15]		x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
SndAct	BOOL	1 = Command is active			x	x		
SndDone	BOOL	1 = Command executed			x	x		
SndErr	BOOL	1 = Command executed with error			x	x		
SndStat	INT	Type of error			x	x		
RdSysStEnOut	BOOL	1 = Redundancy system status active			x	x		
RedCPUOut	BOOL	Controller: 1 = Redundant controller 0 = Single controller			x	x		
Rack0Mstr ... Rack1Mstr	BOOL	1 = CPU rack [0..1] is master			x	x		
Rack0Run ... Rack1Run	BOOL	1 = CPU rack [0..1] is running			x	x		
SyncLink	BOOL	1 = Synchronization possible (redundant CPU)			x	x		
ERR	BOOL	Internal error (redundant CPU)			x	x		
ErrCode	INT	Error code of the redundancy system status (redundant CPU)			x	x		

Functions

Configuring error handling (Page 409)
Configuring the mode of operation (Page 406)
Configuring the process value (Page 408)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.13.1.3 TCON_AI32 (TCON4cnet) | Description

Description

Receives up to 32 analog structures from another CPU using an Industrial Ethernet connection.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_TCON_COMM	BYTE	Connection to communication block	TCON			x		x	
R_ID	INT	Telegram connection ID	0	2147483647	-2147483647	x		x	
RcvMonTime	INT	Number of cycles until substitute value is output in the event of a communication error	0	2147483647	-2147483647	x		x	
SimOn	BOOL	1 = Simulation active	FALSE	TRUE	FALSE	x		x	
Feature	INT	Input for additional functions	0	2147483647	-2147483647	x		x	
Sim_PV00 ... Sim_PV31	FLOAT	Simulation value	0.0	3.4028235E38	-3.4028235E38	x [0..15]		x	
Sub_PV00 ... Sub_PV31	FLOAT	Substitute value	0.0	3.4028235E38	-3.4028235E38	x [0..15]		x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
PV_Out00 ... PV_Out31	FLOAT	Output process value [00..31]			x [0..15]	x		
SimAct	BOOL	1 = Simulation active			x	x		
RcvMonErr	BOOL	1 = No data received via connection			x	x		
RcvNewData	BOOL	1 = New data received via connection			x	x		

Parameter	Data type	Description	Message / message class	Active / inactive	Con n	L	HMI	DIA
RcvErr	BOOL	1 = Command executed with error			x	x		
RcvStat	INT	Type of error			x	x		
RdSys-StEnOut	BOOL	1 = Redundancy system status active			x	x		
RedCPU-Out	BOOL	Controller: 1 = Redundant controller 0 = Single controller			x	x		
Rack0Mstr ... Rack1Mstr	BOOL	1 = CPU rack [0..1] is master			x	x		
Rack0Run ... Rack1Run	BOOL	1 = CPU rack [0..1] is running			x	x		
SyncLink	BOOL	1 = Synchronization is possible (redundant CPU)			x	x		
ERR	BOOL	Internal error (redundant CPU)			x	x		
ErrCode	INT	Error code of the redundancy system status (redundant CPU)			x	x		

Functions

Configuring substitute values (Page 405)

Configuring error handling (Page 408)

Configuring the mode of operation (Page 404)

Configuring simulation (Page 400)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.13.1.4 TCON_IO32 (TCON4cnet) | Description

Description

Sends up to 32 integers via Industrial Ethernet connection to another CPU.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_TCON_COMM	BYTE	Connection to communication block	TCON			x		x	
R_ID	INT	Telegram connection ID	0	100	0		x	x	
RdSysStEn	BOOL	Activates the redundancy for the system (plant network)	FALSE	TRUE	FALSE		x	x	
RedCPU	BOOL	Activates the redundancy for the CPU	FALSE	TRUE	FALSE		x	x	
SndEn	BOOL	1 = Send continuously	TRUE	TRUE	FALSE		x	x	
Min_Time	INT	Minimum wait time	1	32767	1		x	x	
Max_Time	INT	Maximum wait time	10	32767	1		x	x	
PVHys_in00 ... PVHys_In31	INT	EDC hysteresis of process value PV_In[00..31]	0	2147483647	0		x	x	
PV_In00 ... PV_In31	INT	Input parameters [0..31]	0	2147483647	0	x	x	x[0..15]	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
SndAct	BOOL	1 = Command is active			x	x		
SndDone	BOOL	1 = Command executed			x	x		
SndErr	BOOL	1 = Command executed with error			x	x		
SndStat	INT	Type of error			x	x		
RdSys-StEnOut	BOOL	1 = Redundancy system status active			x	x		
RedCPU-Out	BOOL	Controller: 1 = Redundant controller 0 = Single controller			x	x		
Rack0Mstr ... Rack1Mstr	BOOL	1 = CPU rack [0..1] is master			x	x		
Rack0Run ... Rack1Run	BOOL	1 = CPU rack [0..1] is running			x	x		

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
SyncLink	BOOL	1 = Synchronization possible (redundant CPU)			x	x		
ERR	BOOL	Internal error (redundant CPU)			x	x		
ErrCode	INT	Error code of the redundancy system status (redundant CPU)			x	x		

Functions

Configuring error handling (Page 409)
Configuring the mode of operation (Page 406)
Configuring the process value (Page 408)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.13.1.5 TCON_II32 (TCON4cnet) | Description

Description

Receives up to 32 integers via Industrial Ethernet connection from another CPU.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_TCON_COMM	BYTE	Connection to communication block	TCON			x		x	
R_ID	INT	Telegram connection ID	0	100	0		x	x	
RcvMonTime	INT	Number of cycles until substitute value is output in the event of a communication error	12	2147483647	1		x	x	
SimOn	BOOL	1 = Simulation active	FALSE	TRUE	FALSE		x	x	
Feature	INT	Input for additional functions	0	536870912	0		x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
Sim_PV00 ... Sim_PV31	INT	Simulation value	0	2147483647	0	x [0..15]	x	x	
Sub_PV00 ... Sub_PV31	INT	Substitute value	0	2147483647	0	x [0..15]	x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
PV_Out00 ... PV_Out31	INT	Output process value [00..31]			x [0..15]	x		
SimAct	BOOL	1 = Simulation active			x	x		
RcvMonErr	BOOL	1 = No data received via connection			x	x		
RcvNewData	BOOL	1 = New data received via connection			x	x		
RcvErr	BOOL	1 = Command executed with error			x	x		
RcvStat	INT	Type of error			x	x		
RdSysStEnOut	BOOL	1 = Redundancy system status active			x	x		
RedCPUOut	BOOL	Controller: 1 = Redundant controller 0 = Single controller			x	x		
Rack0Mstr ... Rack1Mstr	BOOL	1 = CPU rack [0..1] is master			x	x		
Rack0Run ... Rack1Run	BOOL	1 = CPU rack [0..1] is running			x	x		
SyncLink	BOOL	1 = Synchronization is possible (redundant CPU)			x	x		
ERR	BOOL	Internal error (redundant CPU)			x	x		
ErrCode	INT	Error code of the redundancy system status (redundant CPU)			x	x		

Functions

Configuring substitute values (Page 405)

Configuring error handling (Page 408)

Configuring the mode of operation (Page 404)

Configuring simulation (Page 400)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.13.1.6 TCON_DO128 (TCON4cnet) | Description

Description

Sends up to 128 digital structures to another CPU using an Industrial Ethernet connection.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_TCON_COMM	BYTE	Connection to communication block	TCON			x		x	
R_ID	INT	Telegram connection ID	0	2147483647	-2147483647	x		x	
RdSysStEn	BOOL	Activates the redundancy for the system (plant network)	FALSE	TRUE	FALSE		x	x	
RedCPU	BOOL	Activates the redundancy for the CPU	FALSE	TRUE	FALSE		x	x	
SndEn	BOOL	1 = Send continuously	FALSE	TRUE	FALSE	x		x	
Min_Time	INT	Minimum wait time	0	2147483647	-2147483647	x		x	
Max_Time	INT	Maximum wait time	0	2147483647	-2147483647	x		x	
PV_In00 ... PV_In127	BOOL	Input parameters [0..127]	FALSE	TRUE	FALSE	x [0..15]		x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
SndAct	BOOL	1 = Command is active			x	x		
SndDone	BOOL	1 = Command executed			x	x		

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
SndErr	BOOL	1 = Command executed with error			x	x		
SndStat	INT	Type of error			x	x		
RdSysStEnOut	BOOL	1 = Redundancy system status active			x	x		
RedCPUOut	BOOL	Controller: 1 = Redundant controller 0 = Single controller			x	x		
Rack0Mstr ... Rack1Mstr	BOOL	1 = CPU rack [0..1] is master			x	x		
Rack0Run ... Rack1Run	BOOL	1 = CPU rack [0..1] is running			x	x		
SyncLink	BOOL	1 = Synchronization possible (redundant CPU)			x	x		
ERR	BOOL	Internal error (redundant CPU)			x	x		
ErrCode	INT	Error code of the redundancy system status (redundant CPU)			x	x		

Functions

Configuring error handling (Page 410)
Configuring the mode of operation (Page 406)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.13.1.7 TCON_DI128 (TCON4cnet) | Description

Description

Receives up to 128 digital structures from a further CPU using an Industrial Ethernet connection.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_TCON_COMM	BYTE	Connection to communication block	TCON			x		x	
R_ID	INT	Telegram connection ID	0	2147483647	-2147483647	x		x	
RcvMonTime	INT	Number of cycles until substitute value is output in the event of a communication error	0	2147483647	-2147483647	x		x	
SimOn	BOOL	1 = Simulation active	FALSE	TRUE	FALSE	x		x	
Feature	INT	Input for additional functions	0	2147483647	-2147483647	x		x	
Sim_PV00 ... Sim_PV127	BOOL	Simulation value	FALSE	TRUE	FALSE	x [0..15]		x	
Sub_PV00 ... Sub_PV127	BOOL	Substitute value	FALSE	TRUE	FALSE	x [0..15]		x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
PV_Out00 ... PV_Out127	FLOAT	Output process value [0..127]			x [0..15]	x		
SimAct	BOOL	1 = Simulation active			x	x		
RcvMonErr	BOOL	1 = No data received via connection			x	x		
RcvNewData	BOOL	1 = New data received via connection			x	x		
RcvErr	BOOL	1 = Command executed with error			x	x		
RcvStat	INT	Type of error			x	x		
RdSysStEnOut	BOOL	1 = Redundancy system status active			x	x		
RedCPUOut	BOOL	Controller: 1 = Redundant controller 0 = Single controller			x	x		

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
Rack0Mstr ... Rack1Mstr	BOOL	1 = CPU rack [0..1] is master			x	x		
Rack0Run ... Rack1Run	BOOL	1 = CPU rack [0..1] is running			x	x		
SyncLink	BOOL	1 = Synchronization is possible (redundant CPU)			x	x		
ERR	BOOL	Internal error (redundant CPU)			x	x		
ErrCode	INT	Error code of the redundancy system status (redundant CPU)			x	x		

Functions

Configuring substitute values (Page 406)

Configuring error handling (Page 408)

Configuring the mode of operation (Page 404)

Configuring simulation (Page 401)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.13.1.8 TCON_SO (TCON4cnet) | Description

Description

Sends a string via Industrial Ethernet connection to another CPU.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_TCON_COMM	BYTE	Connection to communication block	TCON			x		x	
R_ID	INT	Telegram connection ID	1	100	1	x		x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
RdSysStEn	BOOL	Activates the redundancy for the system (plant network)	FALSE	TRUE	FALSE		x	x	
RedCPU	BOOL	Activates the redundancy for the CPU	FALSE	TRUE	FALSE		x	x	
SndEn	BOOL	1 = Send continuously	FALSE	TRUE	FALSE	x		x	
Min_Time	FLOAT	Minimum wait time	0,1	3.4028235E38	0	x		x	
Max_Time	FLOAT	Maximum wait time	60,0	3.4028235E38	0	x		x	
PV_In00	STRING	Input parameters		64	0	x		x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
SndAct	BOOL	1 = Command is active			x	x		
SndDone	BOOL	1 = Command executed			x	x		
SndErr	BOOL	1 = Command executed with error			x	x		
SndStat	INT	Type of error			x	x		
RdSysStEnOut	BOOL	1 = Redundancy system status active			x	x		
RedCPUOut	BOOL	Controller: 1 = Redundant controller 0 = Single controller			x	x		
Rack0Mstr ... Rack1Mstr	BOOL	1 = CPU rack [0..1] is master			x	x		
Rack0Run ... Rack1Run	BOOL	1 = CPU rack [0..1] is running			x	x		
SyncLink	BOOL	1 = Synchronization possible (redundant CPU)			x	x		
ERR	BOOL	Internal error (redundant CPU)			x	x		
ErrCode	INT	Error code of the redundancy system status (redundant CPU)			x	x		

Functions

Configuring error handling (Page 409)

Configuring the mode of operation (Page 406)
Configuring the process value (Page 408)

A.13.1.9 TCON_SI (TCON4cnet) | Description

Description

Receives a string via Industrial Ethernet connection from another CPU.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_TCON_COMM	BYTE	Connection to communication block	TCON			x		x	
R_ID	INT	Telegram connection ID	1	100	1	x		x	
RcvMonTime	FLOAT	Number of cycles until substitute value is output in the event of a communication error	12,0	32767.0	1,0	x		x	
SimOn	BOOL	1 = Simulation active	FALSE	TRUE	FALSE	x		x	
Feature	INT	Input for additional functions	0	536870912	0	x		x	
Sim_PV00	STRING	Simulation value	0	64	0	x		x	
Sub_PV00	STRING	Substitute value	0	64	0	x		x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
PV_Out00	STRING	Output process value			x	x		
SimAct	BOOL	1 = Simulation active			x	x		
RcvMonErr	BOOL	1 = No data received via connection			x	x		
RcvNewData	BOOL	1 = New data received via connection			x	x		
RcvErr	BOOL	1 = Command executed with error			x	x		
RcvStat	INT	Type of error			x	x		
RdSysStEnOut	BOOL	1 = Redundancy system status active			x	x		

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
RedCPU-Out	BOOL	Controller: 1 = Redundant controller 0 = Single controller			x	x		
Rack0Mstr ... Rack1Mstr	BOOL	1 = CPU rack [0..1] is master			x	x		
Rack0Run ... Rack1Run	BOOL	1 = CPU rack [0..1] is running			x	x		
SyncLink	BOOL	1 = Synchronization is possible (redundant CPU)			x	x		
ERR	BOOL	Internal error (redundant CPU)			x	x		
ErrCode	INT	Error code of the redundancy system status (redundant CPU)			x	x		

Functions

Configuring substitute values (Page 405)

Configuring error handling (Page 408)

Configuring the mode of operation (Page 404)

Configuring simulation (Page 400)

A.13.2 Simulation

A.13.2.1 TCON_AI32, TCON_II32, TCON_SI (TCON4cnet) | Configure simulation

Description

Activates the simulation at the block.

Relevant input parameters

"Switch on simulation" command:

- `SimOn=1`

Configure simulation value:

- `Sim_PV[00..31]`

Relevant output parameters

Simulation is active:

- `SimAct=1`

Received process values:

- `PV_Out[00..31]`

See also

Simulation basics (Page 237)

A.13.2.2 TCON_DI128 (TCON4cnet) | Configure simulation

Description

Activates the simulation at the block.

Relevant input parameters

"Switch on simulation" command:

- `SimOn=1`

Configure simulation value:

- `Sim_PV[00..127]`

Relevant output parameters

Simulation is active:

- `SimAct=1`

Received process values:

- `PV_Out[00..127]`

See also

Simulation basics (Page 237)

A.13.3 Parameter assignment

A.13.3.1 TCON_COMM (TCON4cnet) | Configure connection

Description

Defines the communication partners and connects the blocks S7Put and S7Get.

Principle of operation

TCP | Set up the data transfer (Page 230)

Relevant input parameters

Configure ID of connection 1-2:

- ID[1..2]

Configure the IP addresses of the partner CPU:

- IPAddress[1..2]

Configure local TSAP ID of connection 1-2:

- LocalTSAP[1..2]

Parametrizing remote TSAP-IP of connection 1-2:

- RemoteTSAP[1..2]

Select connection type:

- ConnectionType[1..2]

Configure port number for connection 1-4:

- Port[1..2]

Note

Port number depending on connection type

- ConnectionType.. = "Native": The specified port number ("Port..") is used for a TCP connection.
- ConnectionType.. = "IsoOnTcp": The port number is automatically set internally to the value "102". A port number ("Port..") specified by the user is ignored.

A.13.3.2 TCON_COMM (TCON4cnet) | Establish connection

Description

Activates communication and specifies the communication mode.

Relevant input parameters

Establish connection:

- Connect = 1

Disconnect the connection:

- Connect = 0

Activate communication via TCPCon:

- Active: Active = 1
- Not active: Active = 0

Relevant output parameters

Connection established:

- Connected = 1

Connection [1..2] established

- Connected[1..2] = 1

Output for connection to blocks for sending and receiving:

- TO_TCON_RD_WR

A.13.3.3 TCON_COMM (TCON4cnet) | Monitor communication

Description

Monitors connections and the communication between devices.

Relevant input parameters

Configure wait time for connection setup or telegram receipt via connection 1-2:

- Timeout[1..2]

Relevant output parameters

Status of the connection is pending:

- Status[1..2]
  = 0: No problem
  = 2: No connection

Connection problem:

- Error[1..2] = 1

A.13.3.4 TCON_AI32, TCON_II32, TCON_SI (TCON4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant input parameters

Configure ID of telegram connection:

- `R_ID`

Configure connection to the block TCON_COMM :

- `FROM_TCON_COMM`

Relevant output parameters

Received process values:

- `PV_Out[00..31]`

New data is received:

- `RcvNewData = 1`

Communication status of the redundant controller pair:

Controller	Rack 0	Rack 1
Master	`Rack0Mstr = 1`	`Rack1Mstr = 1`
State = `RUN`	`Rack0Run = 1`	`Rack1Run = 1`

RdSysSt is active (redundancy for the system is activated):

- `RdSysStEnOut = 1`

Controller:

- Redundant CPU: `RedCPUOut = 1`
- Non-redundant CPU: `RedCPUOut = 0`

Synchronization between redundant CPUs is possible:

- `SyncLink = 1`

A.13.3.5 TCON_DI128 (TCON4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant input parameters

Configure ID of telegram connection:

- `R_ID`

Configure connection to the block TCON_COMM:

- `FROM_TCON_COMM`

Relevant output parameters

Received process values:

- `PV_Out[00..127]`

New data is received:

- `RcvNewData = 1`

Communication status of the redundant controller pair:

Controller	Rack 0	Rack 1
Master	`Rack0Mstr = 1`	`Rack1Mstr = 1`
State = `RUN`	`Rack0Run = 1`	`Rack1Run = 1`

RdSysSt is active:

- `RdSysStEnOut = 1`

Controller:

- Redundant CPU: `RedCPUOut = 1`
- Non-redundant CPU: `RedCPUOut = 0`

Synchronization between redundant CPUs is possible:

- `SyncLink = 1`

A.13.3.6 TCON_AI32, TCON_II32, TCON_SI (TCON4cnet) | Configure substitute values

Description

Specifies substitute values:

- Simulation active
- Error handling

Principle of operation

Simulation basics (Page 237)

Relevant input parameters

Configure simulation value:

- `Sim_PV[00..31]`

Configure substitute values in the event of communication errors:

- `Sub_PV[00..31]`

Relevant output parameters

-

A.13.3.7 TCON_DI128 (TCON4cnet) | Configure substitute values

Description

Specifies substitute values:

- Simulation active
- Error handling

Principle of operation

Simulation basics (Page 237)

Relevant input parameters

Configure simulation value:

- `Sim_PV[00..127]`

Configure substitute values in the event of communication errors:

- `Sub_PV[00..127]`

Relevant output parameters

-

A.13.3.8 TCON_AO32, TCON_IO32, TCON_DO128, TCON_SO (TCON4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant input parameters

Configure ID of telegram connection:

- `R_ID`

Configure connection to the block TCON_COMM :

- `FROM_TCON_COMM`

Activate redundancy for the system:

- `RdSysStEn = 1`

Activate redundancy for the CPU:

- `RedCPU = 1`

Activate continuous sending:

- `SndEn = 1`

Configure the number of cycles until suppression of transmission:

- `Min_Time`

Configure the number of cycles until resumption of transmission:

- `Max_Time`

Relevant output parameters

Sending active:

- `SndAct`

Sending completed without error:

- `SndDone`

Communication status of the redundant controller pair:

Controller	Rack 0	Rack 1
Master	`Rack0Mstr = 1`	`Rack1Mstr = 1`
`State = RUN`	`Rack0Run = 1`	`Rack1Run = 1`

RdSysStEn is active:

- `RdSysStEnOut = 1`

Controller:

- Redundant CPU: `RedCPUOut = 1`
- Non-redundant CPU: `RedCPUOut = 0`

Synchronization between redundant CPUs is possible:

- `SyncLink = 1`

A.13.3.9 TCON_AO32, TCON_IO32, TCON_SO (TCON4cnet) | Configure process value

Description

Configures the process value.

Relevant input parameters

Configure process value:

- `PV_In[00..31]`

Configure EDC hysteresis

- `PVHys_In[00..31]`

Relevant output parameters

-

A.13.4 Error handling

A.13.4.1 TCON_AI32, TCON_II32, TCON_DI128, TCON_SI (TCON4cnet) | Configure error handling

Description

Configures troubleshooting.

Relevant input parameters

Number of cycles until the substitute value is output.

- `RcvMonTime`

Relevant output parameters

Internal error

- `ERR=1`

Error code of RdSysSt:

- `ErrCode`

No data received:

- `RcvMonErr=1`

Data transfer completed with error:

- `RcvErr=1`

Type of error for transmission:

- `RcvStat =`
  - 0x0000: No problems
  - 0x0012: The R_ID is used by another send block.
  - Confirmation to send:
  - 0x0002: R_ID was not configured for the partner CPU
  - 0x0006: Wrong configuration of the receive block

A.13.4.2 TCON_AO32, TCON_IO32, TCON_SO (TCON4cnet) | Configure error handling

Description

Indicates whether an error information is active at least at one input or output parameter.

Relevant input parameters

-

Relevant output parameters

Data transfer completed with error:

- `SndErr=1`

Type of error for transmission:

- `SndStat =`
  - 0x0000: No problems
  - 0x8xxx: Error code of block
  - 0x0001: The value of R_ID is not within the permitted range.
  - 0x0002: R_ID was not configured for the partner CPU
  - 0x0003: Access error for the data record in the partner CPU
  - 0x000A: No confirmation received before timeout of the connection
  - 0x0012: The R_ID is used by another send block.

Internal error:

- `ERR=1`

Error code of RdSysSt:

- `ErrCode`

A.13.4.3 TCON_DO128 (TCON4cnet) | Configure error handling

Description

Configures troubleshooting.

Relevant input parameters

-

Relevant output parameters

Internal error:

- `ERR=1`

Error code of RdSysSt:

- `ErrCode`

Data transfer completed with error:

- `SndErr=1`

Type of fault:

- `SndStat =`
  - 0x0000: No problems
  - 0x8xxx: Error code of block
  - 0x0001: The value of R_ID is not within the permitted range.
  - 0x0002: R_ID was not configured for the partner CPU
  - 0x0003: Access error for the data record in the partner CPU
  - 0x000A: No confirmation received before timeout of the connection
  - 0x0012: The R_ID is used by another send block.

A.14 OPCUAC4cnet | Protocols.OPC UA.Client-Library

A.14.1 Blocks

A.14.1.1 CN_OPCUAC_AI (OPCUAC4cnet) | Description

Description

Reads the analog values of the selected node from an OPC UA server.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OP-CUAC_COMM	BYTE	Connection to communication block for OPC UA	CN_OP-CUAC			x			
NODE_ID	STRING	OPC UA node ID in the server	0	200	0	x	x	x	x
PUB_INT_GR	INT	Selection of the group of the publishing interval 0 = PUB_INT_1 1 = PUB_INT_2 2 = PUB_INT_3 3 = PUB_INT_4 4 = DEFAULT	4	4	0		x	x	
SAMPL_INT_GR	INT	Selection of the group of the sampling interval 0 = SAMPL_INT_1 1 = SAMPL_INT_2 2 = SAMPL_INT_3 3 = SAMPL_INT_4 4 = DEFAULT	4	4	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
AI01_PV	FLOAT	Analog input value		(came in / went out)	x	x	x	

Functions

Configuring the mode of operation (Page 422)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.14.1.2 CN_OPCUAC_AO (OPUAC4cnet) | Description

Description

Writes an analog value from an OPC UA client to the selected node in an OPC UA server.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OPCUAC_COMM	BYTE	Connection to communication block for OPC UA	CN_OPCUAC			x			x
NODE_ID	STRING	OPC UA node ID		200	0	x	x	x	x
TYPE_ID	STRING	OPC UA Type ID	i=10 (FLOAT)	200	0		x	x	x
AO01_PV	FLOAT	Analog output value	0.0	3.4028235E38	-3.4028235E38	x	x	x	x
ENABLE	BOOL	Write enable	FALSE	TRUE	FALSE	x	x	x	x
MODE	INT	Write mode 0 = CYCLE 1 = CHANGE	0	1	0		x	x	x

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
FAIL	BOOL	Write error	6	Came in / went out	x	x	x	
MSG	STRING	Status message	(6)	(came in / went out)		x	x	

Functions

Configuring the mode of operation (Page 423)
Monitoring communication (Page 430)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.14.1.3 CN_OPCUAC_DI (OPCUAC4cnet) | Description

Description

Reads the digital values of the selected node from an OPC UA server.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OPCUAC_COMM	BYTE	Connection to communication block for OPC UA	CN_OPCUAC			x			x
NODE_ID	STRING	OPC UA node ID		200	0	x	x	x	x
PUB_INT_GR	INT	Selection of the group of the publishing interval 0 = PUB_INT_1 1 = PUB_INT_2 2 = PUB_INT_3 3 = PUB_INT_4 4 = DEFAULT	4	4	0		x	x	
SAMPL_INT_GR	INT	Selection of the group of the sampling interval 0 = SAMPL_INT_1 1 = SAMPL_INT_2 2 = SAMPL_INT_3 3 = SAMPL_INT_4 4 = DEFAULT	4	4	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
DI01_PV	BOOL	Digital input value	(6)	(came in / went out)	x	x	x	

Functions

Configuring the mode of operation (Page 422)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.14.1.4 CN_OPCUAC_DO (OPCUAC4cnet) | Description

Description

Writes digital values from an OPC UA client to the selected node in an OPC UA server.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OPCUAC_COMM	BYTE	Connection to communication block for OPC UA	CN_OPCUAC			x			x
NODE_ID	STRING	OPC UA node ID		200	0	x	x	x	x
TYPE_ID	STRING	OPC UA Type ID	i=1 (BOOLEAN)	200	0		x	x	x
DO01_PV	BOOL	Digital output value	FALSE	TRUE	FALSE	x	x	x	x
ENABLE	BOOL	Write enable	FALSE	TRUE	FALSE	x	x	x	x
MODE	INT	Write mode 0 = CYCLE 1 = CHANGE	0	1	0		x	x	x

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
FAIL	BOOL	Write error	6	Came in / went out	x	x	x	
MSG	STRING	Status message	(6)	(came in / went out)		x	x	

Functions

Configuring the mode of operation (Page 424)
Monitoring communication (Page 430)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.14.1.5 CN_OPCUAC_II (OPCUAC4cnet) | Description

Description

Reads the integer values of the selected node from an OPC UA server.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OPCUAC_COMM	BYTE	Connection to communication block for OPC UA	CN_OPCUAC			x			
NODE_ID	STRING	OPC UA node ID	0	200	0	x	x	x	x
PUB_INT_GR	INT	Selection of the group of the publishing interval 0 = PUB_INT_1 1 = PUB_INT_2 2 = PUB_INT_3 3 = PUB_INT_4 4 = DEFAULT	4	4	0		x	x	
SAMPL_INT_GR	INT	Selection of the group of the sampling interval 0 = SAMPL_INT_1 1 = SAMPL_INT_2 2 = SAMPL_INT_3 3 = SAMPL_INT_4 4 = DEFAULT	4	4	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
II01_PV	INT	Integer input value			x	x	x	

Functions

Configuring the mode of operation (Page 422)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.14.1.6 CN_OPCUAC_IO (OPCUAC4cnet) | Description

Description

Writes an integer value from an OPC UA client to the selected node in an OPC UA server.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OPCUAC_COMM	BYTE	Connection to communication block for OPC UA	CN_OPCUAC			x			
NODE_ID	STRING	OPC UA node ID		200	0	x	x	x	x
TYPE_ID	STRING	OPC UA Type ID	i=6 (INT32)	200	0		x	x	x
IO01_PV	INT	Integer output value	0	2147483647	-2147483647	x	x	x	x
ENABLE	BOOL	Write enable	TRUE	TRUE	FALSE	x	x	x	x
MODE	INT	Write mode 0 = CYCLE 1 = CHANGE	0	1	0		x	x	x

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
FAIL	BOOL	Write error	6	Came in / went out	x	x	x	
MSG	STRING	Status message	(6)	(came in / went out)		x	x	

Functions

Configuring the mode of operation (Page 423)
Monitoring communication (Page 430)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.14.1.7 CN_OPCUAC_SI (OPCUAC4cnet) | Description

Description

Reads the string values of the selected node from an OPC UA server.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OPCUAC_COMM	BYTE	Connection to communication block for OPC UA	CN_OPCUAC			x			x
NODE_ID	STRING	OPC UA node ID	0	200	0	x	x	x	x
PUB_INT_GR	INT	Selection of the group of the publishing interval 0 = PUB_INT_1 1 = PUB_INT_2 2 = PUB_INT_3 3 = PUB_INT_4 4 = DEFAULT	4	4	0		x	x	
SAMPL_INT_GR	INT	Selection of the group of the sampling interval 0 = SAMPL_INT_1 1 = SAMPL_INT_2 2 = SAMPL_INT_3 3 = SAMPL_INT_4 4 = DEFAULT	4	4	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
SI01_PV	STRING	String input value		(came in / went out)	x	x	x	

Functions

Configuring the mode of operation (Page 422)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

A.14.1.8 CN_OPCUAC_SO (OPCUAC4cnet) | Description

Description

Writes a string value from an OPC UA client to the selected node in an OPC UA server.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OP-CUAC_COMM	BYTE	Connection to communication block for OPC UA	CN_OP-CUAC			x			
NODE_ID	STRING	OPC UA node ID		200	0	x	x	x	x
TYPE_ID	STRING	OPC UA Type ID	STRING	200	0		x	x	x
SO01_PV	STRING	String output value		64	0	x	x	x	x
ENABLE	BOOL	Write enable	FALSE	TRUE	FALSE	x	x	x	x
MODE	INT	Write mode 0 = CYCLE 1 = CHANGE	0	1	0		x	x	x

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
FAIL	BOOL	Write error	6	Came in / went out	x	x	x	
MSG	STRING	Status message	(6)	(came in / went out)		x	x	

Functions

Configuring the mode of operation (Page 425)

Monitoring communication (Page 430)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.14.1.9 CN_OPCUAC_COMM

CN_OPCUAC_COMM (OPCUAC4cnet) | Description

Description

Establishes the connection to the OPC UA Server.

Parameters

Input parameters (Page 419)
Output parameters (Page 421)

Functions

Configuring a connection (Page 426)
Establishing a connection (Page 428)
Configuring a time stamp (Page 429)
Configuring communication (Page 429)
Monitoring communication (Page 431)

CN_OPCUAC_COMM (OPCUAC4cnet) | Input parameters

Parameters	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
ENABLE	BOOL	Enabling a connection	TRUE	TRUE	FALSE		x	x	x
SRV_1	STRING	OPC UA server URL 1		200	0		x	x	x
SRV_2	STRING	OPC UA server URL 2		200	0		x	x	x

Parameters	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
RED_MODE	INT	OPC UA redundancy mode: 0 = NONE 1 = COLD 2 = WARM 3 = HOT_1 4 = HOT_2	0	4	0		x	x	x
SEC_MODE	INT	OPC UA safety mode: 0 = NONE 1 = BASIC128RSA15_SIGN_ENCRYPT 2 = BASIC128RSA15_SIGN 3 = BASIC256_SIGN_ENCRYPT 4 = BASIC256_SIGN 5 = BASIC256SHA256_SIGN_ENCRYPT 6 = BASIC256SHA256_SIGN 7 = AES128_SIGN_ENCRYPT 8 = AES128_SIGN 9 = AES256_SIGN_ENCRYPT 10 = AES256_SIGN	0	10	0		x	x	x

Parameters	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
TIMEOUT	INT	OPC UA connection timeout (ms)	10000	60000	1000		x	x	x
USER	STRING	OPC UA user name		200	0		x	x	x
EPWD	STRING	Encrypted OPC UA password		200	0		x	x	x
LANG	STRING	Connection language	en_US	16	0		x	x	x
T_ORVD	BOOL	Overwrite the time of origin	FALSE	TRUE	FALSE		x	x	x
W_TIME	BOOL	Send timestamp	TRUE	TRUE	FALSE		x	x	x
W_QUAL	BOOL	Send Quality Code	TRUE	TRUE	FALSE		x	x	x

Parameters	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
PUB_INT_1	INT	Group (1..4) of the publishing interval (ms)	100	10000	100		x	x	x
PUB_INT_2			250				x	x	x
PUB_INT_3			500				x	x	x
PUB_INT_4			1000				x	x	x
SAMPL_INT_1	INT	Group (1..4) of the sampling interval (ms)	100	10000	100		x	x	x
SAMPL_INT_2			250				x	x	x
SAMPL_INT_3			500				x	x	x
SAMPL_INT_4			1000				x	x	x
MAX_SUB_ITEM_CALL	INT	Maximum number of objects to watch per call	500	5000	1		x	x	x
MAX_WR_ITEM_CALL	INT	Maximum number of write objects per call	500	5000	1		x	x	x

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

CN_OPCUAC_COMM (OPCUAC4cnet) | Output parameters

*The parameters involved are invisible in the workspace and visible only in watch tables.

Parameters	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TO_OPCUA_CLIENT_WR_RD	BYTE	Connection to OPC UA blocks for read or write commands	x / CN_OPCUAC		x		x	
TRBL_AL	BOOL	Connection failure	x / 6	Came in / went out	x		x	
RED_LOSS	BOOL	Redundancy loss	x / 6	Came in / went out	x		x	

Parameters	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
STATUS_1 ... STATUS_2 (*)	INT	Connection status (1..2) Default = 0 0 = INACTIVE 1 = ESTABLISHED 2 = ACTIVE_SUBSCRIBE 3 = ACTIVE_WRITE 4 = ACTIVE_SUBSCRIBE_AND_WRITE	(x / 6)	(came in / went out)			x	
FAIL_1 ... FAIL_2	BOOL	Connection failure (1..2)	x / 6	Came in / went out	x		x	
MSG_1 ... MSG_2 (*)	STRING [64]	Status message (1..2)	(x / 6)	(came in / went out)			x	

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.14.2 Configuration

A.14.2.1 CN_OPCUAC_AI, CN_OPCUAC_DI, CN_OPCUAC_SI, CN_OPCUAC_II (OPCUAC4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant input parameters

Create connection to communication block for OPC UA:

- `FROM_CN_OPCUAC_COMM`

Configure address of the monitored node in the server:

- `NODE_ID`

Configure selection of the group:

- of the publishing interval:
  - `PUB_INT_GR = (0..4)`

  Note
  PUB_INT_GR = 4 (DEFAULT)
  DEFAULT depends on the runtime cycle.

- of the sampling interval:
  - `SAMPL_INT_GR = (0..4)`

  Note
  SAMPL_INT_GR = 4 (DEFAULT)
  DEFAULT depends on the publishing interval.

Relevant output parameters

Read input value:

- ..01_PV
  (e.g. `AI01_PV`, `DI01_PV`, `SI01_PV`, `II01_PV`)

A.14.2.2 CN_OPCUAC_AO, CN_OPCUAC_IO (OPCUAC4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant input parameters

Create connection to communication block for OPC UA:

- `FROM_CN_OPCUAC_COMM`

Enable writing:

- `ENABLE = TRUE`

Configure address of the monitored node in the server:

- `NODE_ID`

Configure value to be written:

- ..01_PV
  (e.g. `AO01_PV`, `IO01_PV`)

Configure data type of the value to be written:

- `TYPE_ID =`
  - i=1 (BOOLEAN)
  - i=2 (SBYTE)
  - i=3 (BYTE)
  - i=4 (INT16)
  - i=5 (UINT16)
  - i=6 (INT32)
  - i=7 (UINT32)
  - i=8 (INT64)
  - i=9 (UINT64)
  - i=10 (FLOAT)
  - i=11 (DOUBLE)
  - i=12 (STRING)
  - i=21 (LOCALIZEDTEXT)
  - i=290 (DURATION)

Configure write operation:

- Cyclic value transmission: `MODE = 0`
- Value transmission only when value changes: `MODE = 1`

Relevant output parameters

-

A.14.2.3 CN_OPCUAC_DO (OPCUAC4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant input parameters

Create connection to communication block for OPC UA:

- `FROM_CN_OPCUAC_COMM`

Enable writing:

- `ENABLE = TRUE`

Configure address of the monitored node in the server:

- `NODE_ID`

Configure digital value to be written:

- `DO01_PV`

Configure data type of the value to be written:

- `TYPE_ID`
  - i=1 (BOOLEAN)
  - i=2 (SBYTE)
  - i=3 (BYTE)
  - i=4 (INT16)
  - i=5 (UINT16)
  - i=6 (INT32)
  - i=7 (UINT32)
  - i=8 (INT64)
  - i=9 (UINT64)
  - i=10 (FLOAT)
  - i=11 (DOUBLE)
  - i=12 (STRING)
  - i=21 (LOCALIZEDTEXT)
  - i=290 (DURATION)

Configure write operation:

- Cyclic value transmission: `MODE = 0`
- Value transmission only when value changes: `MODE = 1`
- Value transmission only when `IN` changes from 0 to 1 or with `IN = 1` and connection change: `MODE = 2`

Relevant output parameters

-

A.14.2.4 CN_OPCUAC_SO (OPCUAC4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant input parameters

Create connection to communication block for OPC UA:

- `FROM_CN_OPCUAC_COMM`

Enable writing:

- `ENABLE = TRUE`

Configure address of the monitored node in the server:

- `NODE_ID`

Configure string value to be written:

- `SO01_PV`

Configure data type of the value to be written:

- `TYPE_ID`
  - i=1 (BOOLEAN)
  - i=12 (STRING)
  - i=15 (BYTESTRING)
  - i=21 (LOCALIZEDTEXT)

Configure write operation:

- Cyclic value transmission: `MODE = 0`
- Value transmission only when value changes: `MODE = 1`

Relevant output parameters

-

A.14.2.5 CN_OPCUAC_COMM (OPCUAC4cnet) | Configure connection

Description

Defines communication connection to the OPC UA Server.

Principle of operation

OPC UA (Page 239)

Relevant input parameters

Configure URL of the primary or secondary OPC UA server (1..2):

- SRV_[1..2]

Note

URL

The URL must have the following format: "opc.tcp://host name or ip address of server:port"

The host name of the OPC UA Server must be entered in the host file of the CPU of the OPC UA Client.

Configure redundancy mode for the Server: (*= The second Server URL is mandatory.)

- No redundancy. The connection only consists of a connection link. The second Server URL is not used.
  `RED_MODE = NONE`
- Cold redundancy. The second Server is not connected. (*)
  `RED_MODE = COLD`
- Warm redundancy. The second Server is not connected and all monitored elements are placed but inactive. (*)
  `RED_MODE = WARM`
- HOT_1 redundancy. The second Server is not connected and all monitored elements are placed but the data is not called. The publication is still deactivated. (*)
  `RED_MODE = HOT_1`
- HOT_2 redundancy. The second Server is not connected and transfers data. (*)
  `RED_MODE = HOT_2`

Configure encryption for the connection:

- `SEC_MODE = BASIC128RSA15_SIGN_ENCRYPT`
- `SEC_MODE = BASIC128RSA15_SIGN`
- `SEC_MODE = BASIC256_SIGN_ENCRYPT`
- `SEC_MODE = BASIC256_SIGN`
- `SEC_MODE = BASIC256SHA256_SIGN_ENCRYPT`
- `SEC_MODE = BASIC256SHA256_SIGN`
- `SEC_MODE = AES128_SIGN_ENCRYPT`
- `SEC_MODE = AES128_SIGN`
- `SEC_MODE = AES256_SIGN_ENCRYPT`
- `SEC_MODE = AES256_SIGN`

Note

Secure connection for receiving the safety mode from the OPC UA Server.

First select the user name and password before you select the redundancy mode or before you establish a connection to the secondary Server.

Note

Use security functions

Decide on an authentication method. The use of authentication by password or certificate is recommended.

Configure the user name for the login at OPC UA Server:

- `USER`

Note

User name

This is the same as for CNET.

Configure encrypted password for the connection:

- `EPWD`

Note

Password

You enter the password in the password area.

Configure connection language:

- `LANG`

Relevant output parameters

Connection to the OPC UA block for read or write commands:

- `TO_OPCUA_CLIENT_WR_RD`

A.14.2.6 CN_OPCUAC_COMM (OPCUAC4cnet) | Establish connection

Description

Activates communication and specifies the communication mode.

Relevant input parameters

Activate communication for the block:

- Active: `ENABLE = TRUE`
- Not active: `ENABLE = FALSE`

A.14.2.7 CN_OPCUAC_COMM (OPCUAC4cnet) | Configure time stamp

Description

Configures the time stamp.

Relevant input parameters

(*Effect on blocks CN_OPCUAC_xI and CN_OPCUAC_xC:)

Configure handling of the remote time stamp: (*)

- Overwrite time stamp from the Server by local system time:
  `T_OVRD = TRUE`
- Accept time stamp from Server at block:
  `T_OVRD = FALSE`

(**Effect on blocks CN_OPCUAC_xO:)

Configure transfer of the time stamps for write operations: (**)

- Transfer output value with the time stamp from CNET:
  `W_TIME = TRUE`
- Transfer output value without time stamp:
  `W_TIME = FALSE`

Configure transfer of the quality codes for write operations: (**)

- Transfer output value with the quality code from CNET:
  `W_QUAL = TRUE`
- Transfer output value without quality code:
  `W_QUAL = FALSE`

Note

Quality codes from Server

The quality codes deriving from the OPC UA Server are modified for clear assignment at CNET. (For example the quality code "LIMIT_LOW" is set in CNET for "LIMITBITS_LOW".) If quality codes of the Server cannot be specified in CNET, they are assigned to the parent classes "BAD", "UNCERTAIN" or "GOOD" instead in CNET.

A.14.2.8 CN_OPCUAC_COMM (OPCUAC4cnet) | Configure communication

Description

Configures the communication.

Relevant input parameters

Configure the inspection interval for connection establishment between OPC UA Server and Client:

- `TIMEOUT`

Note

Timeout influences quality code

When a timeout occurs, all quality codes of the block types OPCUAC_xI and OPCUAC_xC are set to "BAD_COMMUNICATION_ERROR".

Note

Adjustment

The CNET OPC UA Client can usually read the Server capabilities and automatically apply the highest possible value.

Configure group (1..4) of the publishing interval (ms):

- `PUB_INT_` [1..4]

Configure group (1..4) of the sampling interval (ms):

- `SAMPL_INT_` [1..4]

Configure maximum number of objects:

- For watch per call:
  - `MAX_SUB_ITEM_CALL`
- Write objects per call:
  - `MAX_WR_ITEM_CALL`

A.14.3 Communication

A.14.3.1 CN_OPCUAC_AO, CN_OPCUAC_DO, CN_OPCUAC_IO, CN_OPCUAC_SO (OPCUAC4cnet) | Monitor communication

Description

Monitors connections and communication between devices.

Relevant input parameters

-

Relevant output parameters

Send to the OPC UA server failed:

- `FAIL = TRUE`

Status of the last send command:

- `MSG`

A.14.3.2 CN_OPCUAC_COMM (OPCUAC4cnet) | Monitor communication

Description

Monitors connections and communication between devices.

Relevant output parameters

Connection failed:

- `TRBL_AL = TRUE`

Connection failed to primary (1) or secondary (2) OPC UA server:

- FAIL_[1..2] = TRUE

Loss of redundancy between the OPC UA Client and Server:

- `RED_LOSS = TRUE`

Status of the connection to the primary (1) or secondary (2) OPC UA server:

- The connection is not redundant and consists of only one channel. This channel is inactive. (INACTIVE)
  STATUS_[1..2] = 0
- The channel is connected. (ESTABLISHED)
  STATUS_[1..2] = 1
- The channel is not connected but ready for activation as backup in the event of a failure. (ACTIVE_SUBSCRIBE)
  STATUS_[1..2] = 2
- The channel is connected. All monitored elements are placed but inactive. (ACTIVE_WRITE)
  STATUS_[1..2] = 3
- The channel is connected. All monitored elements are placed but the data are not called. The publication is still deactivated. (ACTIVE_SUBSCRIBE_AND_WRITE)
  STATUS_[1..2] = 4

Note

Restoring the connection

If the connection to server 1 has been re-established after a connection failure, the link status of connection 2 shows the current status and the link status of server 2.

Generated status message for the connection to the server (1..2): (Applies for the connection failure, ID error or exceeded connection limit.)

- MSG_[1..2]

A.15 OPCUAS4cnet | Protocols.OPC UA.Server-Library

A.15.1 Blocks

A.15.1.1 CN_OPCUAS_AI (OPCUAS4cnet) | Description

Description

Reads the analog values of the selected node from an OPC UA client.

Parameters

Input parameters (Page 436)
Output parameters (Page 437)

Functions

Configuring the mode of operation (Page 439)

A.15.1.2 CN_OPCUAS_AO (OPUAS4cnet) | Description

Description

Writes an analog value from an OPC UA server to the selected node in an OPC UA client.

Parameters

Input parameters (Page 437)

Functions

Configuring the mode of operation (Page 440)

A.15.1.3 CN_OPCUAS_DI (OPCUAS4cnet) | Description

Description

Reads the digital values of the selected node from an OPC UA client.

Parameters

Input parameters (Page 437)

Output parameters (Page 437)

Functions

Configuring the mode of operation (Page 439)

A.15.1.4 CN_OPCUAS_DO (OPCUAS4cnet) | Description

Description

Writes digital values from an OPC UA server to the selected node in an OPC UA client.

Parameters

Input parameters (Page 438)

Functions

Configuring the mode of operation (Page 440)

A.15.1.5 CN_OPCUAS_II (OPCUAS4cnet) | Description

Description

Reads the integer values of the selected node of an OPC UA client.

Parameters

Input parameters (Page 438)

Output parameters (Page 438)

Functions

Configuring the mode of operation (Page 439)

A.15.1.6 CN_OPCUAS_IO (OPCUAS4cnet) | Description

Description

Writes an integer value from an OPC UA client to the selected node in an OPC UA server.

Parameters

Input parameters (Page 438)

Functions

Configuring the mode of operation (Page 440)

A.15.1.7 CN_OPCUAS_SI (OPCUAS4cnet) | Description

Description

Reads the string values of the selected node of an OPC UA client.

Parameters

Input parameters (Page 439)
Output parameters (Page 439)

Functions

Configuring the mode of operation (Page 439)

A.15.1.8 CN_OPCUAS_SO (OPCUAS4cnet) | Description

Description

Writes a string value from an OPC UA server to the selected node in an OPC UA client.

Parameters

Input parameters (Page 439)

Functions

Configuring the mode of operation (Page 440)

A.15.1.9 CN_OPCUAS_COMM (OPCUAS4cnet) | Description

Description

Establishes the connection to the OPC UA client.

Parameters

Input parameters (Page 435)

Output parameters (Page 436)

Functions

Configuring a connection (Page 440)

Configuring communication (Page 442)

Monitoring communication (Page 443)

A.15.2 Connections

A.15.2.1 CN_OPCUAS_COMM (OPCUAS4cnet) | Input parameters

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
SRV_PORT_1 ... SRV_PORT_2	STRING	OPC UA Client URL (1..2)	52520	60000	1000		x	x	x
TAG_ID	STRING	String for the structure of a node ID	tag-id	200	0		x	x	x
Discovery_Server_URL		URL of the Discovery Server		200	0		x	x	x

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
USER	STRING	OPC UA user name		200	0		x	x	x
EPWD	STRING	Encrypted OPC UA password		200	0		x	x	x

A.15.2.2 CN_OPCUAS_COMM (OPCUAS4cnet) | Output parameters

*The parameters involved are invisible in the work area and visible only in watch tables.

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TO_OPCUA_SERVER_WR_RD	BYTE	Connection to OPC UA blocks for read or write commands	x / CN_OPCUAS		x		x	
TRBL_AL	BOOL	Connection failure	x / 6	Came in / went out	x		x	
RED_LOSS	BOOL	Redundancy loss	x / 6	Came in / went out	x		x	
STATUS_1 ... STATUS_2 (*)	INT	Connection status (1..2) Default = 0 0 = INACTIVE 1 = ESTABLISHED 2 = ACTIVE_SUBSCRIBE 3 = ACTIVE_WRITE 4 = ACTIVE_SUBSCRIBE_AND_WRITE	(x / 6)	(came in / went out)			x	
FAIL_1 ... FAIL_2	BOOL	Connection failure (1..2)	x / 6	Came in / went out	x		x	
MSG_1 ... MSG_2 (*)	STRING [64]	Status message (1..2)	(x / 6)	(came in / went out)			x	

A.15.2.3 CN_OPCUAS_AI (OPCUAS4cnet) | Input parameters

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OPCUAS_COMM	BYTE	Connection to communication block for OPC UA	CN_OPCUAS			x			
TAG_ID	STRING	String for the structure of a node ID	tag-id	200	0		x	x	x

A.15.2.4 CN_OPCUAS_AI (OPCUAS4cnet) | Output parameters

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
AI01_PV	FLOAT	Analog input value		(came in / went out)	x	x	x	

A.15.2.5 CN_OPCUAS_AO (OPCUAS4cnet) | Input parameters

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OP-CUAS_COMM	BYTE	Connection to communication block for OPC UA	CN_OP-CUAS			x			x
TAG_ID	STRING	String for the structure of a node ID	tag-id	200	0		x	x	x
AO01_PV	FLOAT	Analog output value	0.0	3.4028235E38	-3.4028235E38	x	x	x	x

A.15.2.6 CN_OPCUAS_DI (OPCUAS4cnet) | Input parameters

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OP-CUAS_COMM	BYTE	Connection to communication block for OPC UA	CN_OP-CUAS			x			x
TAG_ID	STRING	String for the structure of a node ID	tag-id	200	0		x	x	x

A.15.2.7 CN_OPCUAS_DI (OPCUAS4cnet) | Output parameters

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
DI01_PV	BOOL	Digital input value	(6)	(came in / went out)	x	x	x	

A.15.2.8 CN_OPCUAS_DO (OPCUAS4cnet) | Input parameters

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OP-CUAS_COMM	BYTE	Connection to communication block for OPC UA	CN_OP-CUAS			x			x
TAG_ID	STRING	String for the structure of a node ID	tag-id	200	0		x	x	x
DO01_PV	BOOL	Digital output value	FALSE	TRUE	FALSE	x	x	x	x

A.15.2.9 CN_OPCUAS_II (OPCUAS4cnet) | Input parameters

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OP-CUAS_COMM	BYTE	Connection to communication block for OPC UA	CN_OP-CUAS			x			
TAG_ID	STRING	String for the structure of a node ID	tag-id	200	0		x	x	x

A.15.2.10 CN_OPCUAS_II (OPCUAS4cnet) | Output parameters

Parameter	Data type	Description	Message / Message class	Active / inactive	Conn	L	HMI	DIA
II01_PV	INT	Integer input value			x	x	x	

A.15.2.11 CN_OPCUAS_IO (OPCUAS4cnet) | Input parameters

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OP-CUAS_COMM	BYTE	Connection to communication block for OPC UA	CN_OP-CUAS			x			
TAG_ID	STRING	String for the structure of a node ID	tag-id	200	0		x	x	x
IO01_PV	INT	Integer output value	0	2147483647	-2147483647	x	x	x	x

A.15.2.12 CN_OPCUAS_SI (OPCUAS4cnet) | Input parameters

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OP-CUAS_COMM	BYTE	Connection to communication block for OPC UA	CN_OP-CUAS			x			x
TAG_ID	STRING	String for the structure of a node ID	tag-id	200	0		x	x	x

A.15.2.13 CN_OPCUAS_SI (OPCUAS4cnet) | Output parameters

Parameter	Data type	Description	Message / Message class	Active / inactive	Conn	L	HMI	DIA
SI01_PV	STRING	String input value		(came in / went out)	x	x	x	

A.15.2.14 CN_OPCUAS_SO (OPCUAS4cnet) | Input parameters

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_CN_OP-CUAS_COMM	BYTE	Connection to communication block for OPC UA	CN_OP-CUAS			x			
TAG_ID	STRING	String for the structure of a node ID	tag-id	200	0		x	x	x
SO01_PV	STRING	String output value		64	0	x	x	x	x

A.15.3 Parameter assignment

A.15.3.1 CN_OPCUAS_AI, CN_OPCUAS_DI, CN_OPCUAS_SI, CN_OPCUAS_II (OPCUAS4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant input parameters

Create connection to communication block for OPC UA:

- `FROM_CN_OPCUAS_COMM`

Configure string for the structure of a node ID:

- `TAG_ID`

Relevant output parameters

Read input value:

- ..01_PV
  (e.g. `AI01_PV`, `DI01_PV`, `SI01_PV`, `II01_PV`)

A.15.3.2 CN_OPCUAS_AO, CN_OPCUAS_DO, CN_OPCUAS_IO, CN_OPCUAS_SO (OPCUAS4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant input parameters

Create connection to communication block for OPC UA:

- `FROM_CN_OPCUAS_COMM`

Configure string for the structure of a node ID:

- `TAG_ID`

Configure value to be written:

- ..01_PV
  (e.g. `AO01_PV`, `IO01_PV`)

Relevant output parameters

-

A.15.3.3 CN_OPCUAS_COMM (OPCUAS4cnet) | Configure connection

Description

Defines communication connection to the OPC UA client.

Principle of operation

OPC UA server (Page 239)

Relevant input parameters

Configure URL of the primary or secondary OPC UA client (1..2):

- SRV_PORT_[1..2]

Note

URL

The URL must have the following format: "opc.tcp://host name or ip address of client:port"

The host name of the OPC UA client must be entered in the host file of the CPU of the OPC UA server.

Configure string for the structure of a node ID:

- `TAG_ID`

Configure security settings of the protocol:

- Parameter area > "`Security Settings`" tab

Note

Use security functions

Decide on an authentication method. The use of authentication by password or certificate is recommended.

Select authentication method:

- Activate "Anonymous dial-in":
  `Anonymous = TRUE`
- Activate "Dial-in with user data":
  `Username & Password = TRUE`
- Activate "Dial-in with certificate":
  `Certificate = TRUE`

Select security mode:

- No encryption or signature required:
  `None = TRUE`
- Signing required:
  `Sign = TRUE`
- Signing and encryption required:
  `Sign & Encrypt = TRUE`

Select security method:

- `Basic128Rsa15 = TRUE`
- `Basic256 = TRUE`
- `Basic256Sha256 = TRUE`

- `Aes128_Sha256_RsaOaep = TRUE`
- `Aes256_Sha256_RsaPss = TRUE`

Note
Secure connection for receiving the security modes from the OPC UA client

First select the user name and password before you select the redundancy mode or establish a connection to the secondary client.

Configure user data:

Configure the user name for the login to the OPCUA client:

- `USER`

Note
User name

This is the same as for CNET.

Configure encrypted password for the connection:

- `EPWD`

Note
Password

You enter the password in the password area.

Relevant output parameters

Connection to the OPC UA server block for read or write commands:

- `TO_OPCUA_SERVER_WR_RD`

A.15.3.4 CN_OPCUAS_COMM (OPCUAS4cnet) | Configure communication

Description

Configures the communication.

Relevant input parameters

Configure advanced settings for the connection establishment:

- Parameter area > "`Advanced Settings`" tab

Configure test interval for the connection establishment (from OPC UA server to client):

- `MAX_SESSION_TIMEOUT`

Configure maximum number of clients:

- `MAX_SESSION_COUNT`

Configure minimum sampling interval (ms):

- `MIN_SAMPLING_INTERVAL`

Configure minimum publishing interval (ms):

- `MIN_PUBLISHING_INTERVAL`

Maximum number of subscriptions:

- `MAX_SUBSCRIPTION_COUNT`

Configure maximum number of objects:

- For watch per call:
  `MAX_ITEMS_PER_SUBSCRIPTION`

A.15.4 Communication

A.15.4.1 CN_OPCUAS_COMM (OPCUAS4cnet) | Monitor communication

Description

Monitors connections and communication between devices.

Relevant output parameters

Connection failed:

- `TRBL_AL = TRUE`

Connection failed to primary (1) or secondary (2) OPC UA client:

- FAIL_[1..2] = TRUE

Loss of redundancy between the OPC UA server and client:

- `RED_LOSS = TRUE`

Status of the connection to the primary (1) or secondary (2) OPC UA client:

- The connection is not redundant and consists of only one channel. This channel is inactive. (INACTIVE)
  STATUS_[1..2] = 0
- The channel is connected. (ESTABLISHED)
  STATUS_[1..2] = 1
- The channel is not connected but ready for activation as backup in the event of a failure. (ACTIVE_SUBSCRIBE)
  STATUS_[1..2] = 2
- The channel is connected. All monitored elements are placed but inactive. (ACTIVE_WRITE)
  STATUS_[1..2] = 3
- The channel is connected. All monitored elements are placed but the data are not called. The publication is still deactivated. (ACTIVE_SUBSCRIBE_AND_WRITE)
  STATUS_[1..2] = 4

Note

Restoring the connection

When the connection to client 1 has been re-established after a connection failure, the link status of connection 2 shows the current status and the link status of client 2.

Generated status message for the connection to the client (1..2): (Applies for the connection failure, ID error or exceeded connection limit.)

- MSG_[1..2]

See also

OPC UA server (Page 239)

A.16 MODB4cnet | Protocols.Modbus-Library

A.16.1 Blocks

A.16.1.1 MODBUS_AI01 (MODB4cnet) | Description

Description

The block can be used to perform the following tasks:

- Reading an analog value from the MODBUS receive buffer. The start value for the outputs is 0.0 until the first telegram is received.
- Reading the event buffer data of the ABB Triguard system using the MODBUS protocol.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_MODBUS_COMM	BYTE	Connection of MODBUS_COMM block		2147483647	0	x		x	
STATION_ADDRESS	INT	Standby/Block identifier address	1	247	1		x	x	
START_ADDRESS	INT	Start address	0	65535	0		x	x	
COMMAND_TYPE	INT	Command type	3	45	3		x	x	
AI01_TYPE	INT	Data type of the transferred values	1	6	0		x	x	
AI01_SWAP	INT	Data exchange	0	3	0		x	x	
ABB_BIN_SCAN	INT	ABB binary scan interval [s]	60	65535	1		x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
AI01_rLL	FLOAT	Device lower measuring range	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_rUL	FLOAT	Device upper measuring range	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_sLL	FLOAT	Setting: Low limit (EU)	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_sUL	FLOAT	Setting: High limit (EU)	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_sEU	STRING	Setting: Engineering unit (EU)	%	10	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error	x / LEAD	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Multiple message	x / LEAD	Came in / went out	x	x	x	x
AI01_EU	STRING	Engineering unit (EU)			x	x		
AI01_LL	FLOAT	Engineering unit lower limit			x	x		
AI01_UL	FLOAT	Engineering unit upper limit			x	x		

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
AI01_PV	FLOAT	Process variable			x	x		x
PROXY_STATUS1 ... PROXY_STATUS2	INT	Proxy status value; Default = 0;			x	x		x

Functions

Configuring a connection (Page 460)
Configuring communication (Page 460)
Configuring a time stamp (Page 463)
Configuring error handling (Page 480)
Monitoring communication (Page 476)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)
Modbus (Page 241)

A.16.1.2 MODBUS_AI08 (MODB4cnet) | Description

Description

The block reads up to 8 analog values from the MODBUS receive buffer.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_MODBUS_COMM	BYTE	Connection of MODBUS_COMM block		2147483647	0	x		x	
STATION_ADDRESS	INT	Standby/Block identifier address	1	247	1		x	x	
START_ADDRESS	INT	Start address	0	65535	0		x	x	
COMMAND_TYPE	INT	Command type	3	4	3		x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
AI01_TYPE ... AI08_TYPE	INT	Data type of the transferred values	1	6	0		x	x	
AI01_SWAP ... AI08_SWAP	INT	Data exchange	0	3	0		x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
AI01_rLL ... AI08_rLL	FLOAT	Device lower measuring range	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_rUL ... AI08_rUL	FLOAT	Device upper measuring range	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_sLL ... AI08_sLL	FLOAT	Setting: Low limit (EU)	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_sUL ... AI08_sUL	FLOAT	Setting: High limit (EU)	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_sEU ... AI08_sEU	STRING	Setting: Engineering unit (EU)	%	10	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error	x / LEAD	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Multiple message	x / LEAD	Came in / went out	x	x	x	x
AI01_EU ... AI08_EU	STRING	Engineering unit (EU)			x	x		
AI01_LL ... AI08_LL	FLOAT	Engineering unit lower limit			x	x		
AI01_UL ... AI08_UL	FLOAT	Engineering unit upper limit			x	x		

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
AI01_PV ... AI08_PV	FLOAT	Process variable			x	x		x
PROXY_STATUS1 ... PROXY_STATUS2	INT	Proxy status value; Default = 0;			x	x		x

Functions

Configuring a connection (Page 460)
Configuring communication (Page 460)
Configuring error handling (Page 480)
Monitoring communication (Page 476)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)
Modbus (Page 241)

A.16.1.3 MODBUS_AI16 (MODB4cnet) | Description

Description

The block reads 16 analog values from the MODBUS receive buffer.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_MODBUS_COMM	BYTE	Connection of MODBUS_COMM block		2147483647	0	x		x	
STATION_ADDRESS	INT	Standby/Block identifier address	1	247	1		x	x	
START_ADDRESS	INT	Start address	0	65535	0		x	x	
COMMAND_TYPE	INT	Command type	3	4	3		x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
AI01_TYPE ... AI08_TYPE	INT	Data type of the transferred values	1	6	0		x	x	
AI01_SWAP ... AI16_SWAP	INT	Data exchange	0	3	0		x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
AI01_rLL ... AI16_rLL	FLOAT	Device lower measuring range	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_rUL ... AI16_rUL	FLOAT	Device upper measuring range	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_sLL ... AI16_sLL	FLOAT	Setting: Low limit (EU)	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_sUL ... AI16_sUL	FLOAT	Setting: High limit (EU)	0.0	3.4028235E38	-3.4028235E38		x	x	
AI01_sEU ... AI16_sEU	STRING	Setting: Engineering unit (EU)	%	10	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error	x / LEAD	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Multiple message	x / LEAD	Came in / went out	x	x	x	x
AI01_EU ... AI16_EU	STRING	Engineering unit (EU)			x	x		
AI01_LL ... AI16_LL	FLOAT	Engineering unit lower limit			x	x		
AI01_UL ... AI16_UL	FLOAT	Engineering unit upper limit			x	x		

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
AI01_PV ... AI16_PV	FLOAT	Process variable			x	x		x
PROXY_STATUS1 ... PROXY_STATUS2	INT	Proxy status value; Default = 0;			x	x		x

Functions

Configuring a connection (Page 460)

Configuring communication (Page 460)

Configuring error handling (Page 480)

Monitoring communication (Page 476)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

Modbus (Page 241)

A.16.1.4 MODBUS_AO8 (MODB4cnet) | Description

Description

The block writes 8 analog values into a MODBUS transmit buffer.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_MODBUS_COMM	BYTE	Connection of MODBUS_COMM block		2147483647	0	x		x	
STATION_ADDRESS	INT	Modbus station address	1	247	1		x	x	
AO01_ADDRESS ... AO08_ADDRESS	INT	Device address	0	65535	0		x	x	

Parame-ter	Data type	Description	Default	Maximum value	Minimum value	Con n	Edit	L	HMI
AO01_COMM_TYPE ... AO08_COMM_TYPE	INT	Command type:	0	16	0		x	x	
AO01_PV ... AO08_PV	FLOAT	Process value	0.0	3.4028235E38	-3.4028235E38		x	x	
AO01_TYPE ... AO08_TYPE	INT	Data type:	0	5	0		x	x	
AO01_SWAP ... AO08_SWAP	INT	Data exchange:	0	3	0		x	x	
AO01_RSND ... AO08_RSND	BOOL	Resend activation during start	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Parame-ter	Data type	Description	Default	Maximum value	Minimum value	Con n	Edit	L	HMI
AO01_rLL ... AO08_rLL	FLOAT	Low value range	0.0	3.4028235E38	-3.4028235E38		x	x	
AO01_rUL ... AO08_rUL	FLOAT	High value range	0.0	3.4028235E38	-3.4028235E38		x	x	
AO01_sLL ... AO08_sLL	FLOAT	Setting: Low limit (EU)	0.0	3.4028235E38	-3.4028235E38		x	x	
AO01_sUL ... AO08_sUL	FLOAT	Setting: High limit (EU)	0.0	3.4028235E38	-3.4028235E38		x	x	
AO01_sEU ... AO08_sEU	STRING	Setting: Engineering unit (EU)	%	10	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error	x / LEAD	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Multiple message	x / LEAD	Came in / went out	x	x	x	x
PROXY_STATUS1 ... PROXY_STATUS2	INT	Proxy status value; Default = 0;			x	x		x

Functions

Configuring a connection (Page 464)

Configuring communication (Page 464)

Monitoring communication (Page 477)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

Modbus (Page 241)

A.16.1.5 MODBUS_COMM (MODB4cnet) | Description

Description

The MODBUS_COMM block initiates and manages a redundant communications interface.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
PROT_VAR	INT	Protocol selection;	1	3	1		x	x	x
COM_PORT1 ...COM_PORT2	STRING	Communication port		32	0		x	x	x
STATION_TYPE	INT	Station type;	10	11	10		x	x	x

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
T_RES	FLOAT	Time out for feedback messages [s]	1.0	120.0	0.1		x	x	x
T_INTER	FLOAT	Delay between messages [s]	0.0	10.0	0.0		x	x	x
T_SCAN	FLOAT	Delay between scanning for object change [s]	0.05	10.0	0.02		x	x	x
RETRY_NUM	INT	Number of connection attempts	0	10	0		x	x	x
TRAFFIC_TIMEOUT	FLOAT	Data traffic timeout [s]	6.0	10000.0	3.0		x	x	x
EX_RETRY_NUM	INT	Number of times an exception is overlooked	0	10	0		x	x	x

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
GAP_SPACE	INT	Size of the gap in the address range of a request	0	2147483647	0		x	x	
RW_MODE	INT	Read/Write operating mode	0	1	0		x	x	
ENABLE_LENGHT_CHECK	BOOL	Activates checking the length of a message as a response	TRUE	TRUE	FALSE		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error	x / LEAD	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Multiple message	x / LEAD	Came in / went out	x	x	x	x
TO_MODBUS_RD_WR	BYTE	Connection to MODBUS READ/WRITE blocks			x	x		x
STATUS_COM 1 ... STATUS_COM 2	INT	Communication error on communications interface; Default = 0;		1 / 1		x		x
CYCLE_A	INT	Expired time through read request for master (in milliseconds)			x	x		x

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
CYCLE_B	INT	Time consumed through read request for Standby (in milliseconds)			x	x		x
REQ_COUNT	INT	Number of the request for scanning the read data			x	x		x

Functions

Configuring a connection (Page 468)

Configuring communication (Page 471)

Monitoring communication (Page 478)

Configuring error handling (Page 481)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

Modbus (Page 241)

A.16.1.6 MODBUS_DI01 (MODB4cnet) | Description

Description

The block reads a binary value from a MODBUS receive buffer.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_MODBUS_COMM	BYTE	Connection of MODBUS_COMM block		2147483647	0	x		x	
STATION_ADDRESS	INT	Standby/Block identifier address	1	247	1		x	x	
POINT_LENGTH	INT	Number of points	1	1	1				
START_ADDRESS	INT	Start address	0	65535	0		x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
COMMAND_TYPE	INT	Command type:	1	2	1		x	x	
DI_TYPE	INT	Data type of the values	0	1	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error	x / LEAD	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Multiple message	x / LEAD	Came in / went out	x	x	x	x
DI01_PV	BOOL	Process variable			x	x		
PROXY_STATUS1 ... PROXY_STATUS2	INT	Proxy status value; Default = 0;			x	x		x

Functions

Configuring a connection (Page 473)
Configuring communication (Page 473)
Monitoring communication (Page 479)
Configuring error handling (Page 481)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)
Modbus (Page 241)

A.16.1.7 MODBUS_DI08 (MODB4cnet) | Description

Description

The block reads 8 binary values from a MODBUS receive buffer.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_MODBUS_COMM	BYTE	Connection of MODBUS_COMM block		2147483647	0	x		x	
STATION_ADDRESS	INT	Standby/Block identifier address	1	247	1		x	x	
POINT_LENGTH	INT	Number of points	8	8	1		x	x	
START_ADDRESS	INT	Start address	0	65535	0		x	x	
COMMAND_TYPE	INT	Command type	1	2	1		x	x	
DI_TYPE	INT	Data type of the values	0	1	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error	x / LEAD	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Multiple message	x / LEAD	Came in / went out	x	x	x	x
DI01_PV ... DI08_PV	BOOL	Process variable			x	x		
PROXY_STATUS1 ... PROXY_STATUS2	INT	Proxy status value; Default = 0;			x	x		x

Functions

Configuring a connection (Page 473)
Configuring communication (Page 473)
Monitoring communication (Page 479)
Configuring error handling (Page 481)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)
Modbus (Page 241)

A.16.1.8 MODBUS_DI16 (MODB4cnet) | Description

Description

The block reads 16 binary values from a MODBUS receive buffer.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_MODBUS_COMM	BYTE	Connection of MODBUS_COMM block		2147483647	0	x		x	
STATION_ADDRESS	INT	Standby/Block identifier address	1	247	1		x	x	
POINT_LENGTH	INT	Number of points	16	16	1		x	x	
START_ADDRESS	INT	Start address	0	65535	0		x	x	
COMMAND_TYPE	INT	Command type	1	2	1		x	x	
DI_TYPE	INT	Data type of the values	0	1	0		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error	x / LEAD	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Multiple message	x / LEAD	Came in / went out	x	x	x	x
DI01_PV ... DI16_PV	BOOL	Process variable			x	x		
PROXY_STATUS1 ... PROXY_STATUS2	INT	Proxy status value; Default = 0;			x	x		x

Functions

Configuring a connection (Page 473)
Configuring communication (Page 473)
Monitoring communication (Page 479)
Configuring error handling (Page 481)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)
Modbus (Page 241)

A.16.1.9 MODBUS_DO8 (MODB4cnet) | Description

Description

The block writes 8 binary values into a MODBUS transmit buffer.

Parameters of inputs

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
FROM_MODBUS_COMM	BYTE	Connection of MODBUS_COMM block		2147483647	0	x		x	
STATION_ADDRESS	INT	Modbus station address	1	247	1		x	x	
DO01_ADDRESS ... DO08_ADDRESS	INT	Start address	0	65535	0		x	x	
DO01_COMM_TYPE ... DO08_COMM_TYPE	INT	Command type:	0	9	0		x	x	
DO01_PV ... DO08_PV	BOOL	Process value	FALSE	TRUE	FALSE	x	x	x	

Parameter	Data type	Description	Default	Maximum value	Minimum value	Conn	Edit	L	HMI
DO01_RSND ... DO08_RSND	BOOL	Resend activation during start	FALSE	TRUE	FALSE	x	x	x	
QUALITY	BOOL	Processing of the quality status	FALSE	TRUE	FALSE		x	x	

Parameters of outputs

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
TRBL_AL	BOOL	Communication error	x / LEAD	Came in / went out	x	x	x	x
MULTIPLE_AL	BOOL	Multiple message	x / LEAD	Came in / went out	x	x	x	x
PROXY_STATUS1 ... PROXY_STATUS2	INT	Proxy status value; Default = 0;			x	x		x

Functions

Configuring a connection (Page 474)

Configuring communication (Page 475)

Monitoring communication (Page 477)

See also

General quality statuses (Page 223)

Parameter tables (Page 223)

Modbus (Page 241)

A.16.2 Parameter assignment

A.16.2.1 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Configure connection

Description

Defines communication connection to the Modbus device.

Relevant input parameters

Connection to the block MODBUS_COMM:

- `FROM_MODBUS_COMM`

Configure Modbus device address:

- `STATION_ADDRESS`

Configure the address of the first register of the analog input values:

- `START_ADDRESS`

See also

Modbus | application server and CN 4100 Modbus (Page 242)

A.16.2.2 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Configure communication

Description

Configures the communication.

Relevant input parameters

Configure the command type for calling the register from the Modbus devices:

- Read multiple register (fc 3):
  `COMMAND_TYPE = 3`
- Read input register (fc 4):
  `COMMAND_TYPE = 4`
- Read ABB system buffer (fc 3): (only applies for MODBUS_AI01)
  `COMMAND_TYPE = 45`

Configure the sequence for reading the data from the buffer:

Modbus | Register and BYTE swap (Page 242)

- Deactivated:
  `AI0x_SWAP = 0`
- Swapping of the byte sequence:
  `AI0x_SWAP = 1`
- Swapping of the register sequence:
  `AI0x_SWAP = 2`
- Swapping of the byte and register sequence:
  `AI0x_SWAP = 3`

Note

Same data for data exchange

The data exchange for reading the data from the storage must correspond to the data exchange for writing in the buffer.

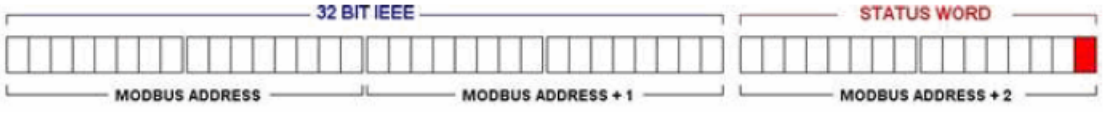
Configure data type of the transferred values:

- 16 BITS without sign:
  `AI0x_TYPE = 1`
- 16 BITS with sign
  `AI0x_TYPE = 2`
- 32 BITS without sign
  `AI0x_TYPE = 3`
- 32 BITS with sign
  `AI0x_TYPE = 4`
- 32 BIT IEEE standard
  `AI0x_TYPE = 5`
- 48 BIT IEEE_WITHSTATUS
  `AI0x_TYPE = 6`

Note

48 Bit IEEE_WITHSTATUS

- Data packages of the type 48 bit IEEE_WITHSTATUS encompass 4 bytes for the signal values and two bytes for the quality of the signal. If the LSB of the status word (the last two bytes) has the 1, the signal quality is invalid. If the LSB has the value 0, the signal quality is valid. The LSB is displayed in red in the following figure.
- The data type 48 BIT IEEE_WITHSTATUS is only supported in primary stations (Modbus client) and not in secondary stations (Modbus server).



Configure signal range:

Limit values for the Modbus line:

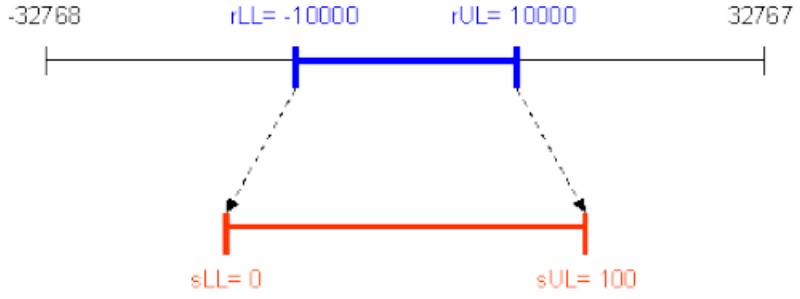
- Low limit:
  `AI0x_rLL`
- High limit:
  `AI0x_rUL`.

Limit values for available scaled values:

- Low limit:
  `AI0x_sLL`
- High limit:
  `AI0x_sUL`.

Example

The signals are scaled as shown in the following example:



The example applies to an input type of 16 bits with sign. A raw value range of -10000 to 10000 is scaled to a range of 0 to 100. The scaling of the raw values in the example is specified in the table.

Raw value	Scaled value
-10000	0
0	50
10000	100

Predefined signal ranges

The scaling is performed as follows if specific settings are made for the signal range:

- AI0x_rLL, AI0x_rUL, AI0x_sLL and AI0x_sUL are all =**0**
  The raw and scaled value ranges are set to the entire range of the selected data type. No scaling takes place between the raw values and the scaled values.
- AI0x_rLL and AI0x_rUL are =**0**, and AI0x_sLL and AI0x_sUL are **not**=**0**
  The raw value range is set to the entire range of the selected data type. The scaled range is limited through the lower and upper limit values AI0x_sLL and AI0x_sUL. No scaling takes place between the raw values and the scaled values.
- AI0x_rLL and AI0x_rUL are **not**=**0**, and AI0x_sLL and AI0x_sUL are =**0**
  The raw value range is set to the entire range of the selected data type. The scaling of the raw value range on the scaled range takes place between the lower and upper limit values AI0x_rLL and AI0x_rUL.

The overall ranges of the data types are as follows:

Type	Low limit	High limit
16 bits without sign	0	65535
16 bits with sign	-32768	32767
32 bits without sign	0	4294967295
32 bits with sign	-2147483648	2147483647

Note

32-bit IEEE standard

For the 32-Bit-IEEE standard, the limits of the raw value range AI0x_rLL and AI0x_rUL have to be set equal 0. Only the limit values for the scaled range AI0x_sLL and AI0x_sUL should be set. No scaling of the input takes place.

A.16.2.3 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Configure time stamp

Description

Configures the time stamp.

Relevant input parameters

Read time-stamped binary data of the event buffer of the ABB Triguard:

- `COMMAND_TYPE  = 45`

Configure time interval between two successive scans of the event buffer of the ABB Triguard:

- `ABB_BIN_SCAN`

Note

Read ABB system buffer (fc 3)

The command type Read ABB system buffer (fc 3) is only supported in Modbus in primary stations (Modbus client) and not in secondary stations (Modbus server).

A.16.2.4 MODBUS_AO8 (MODB4cnet) | Parameterize a connection

Description

Defines a communication connection to the Modbus device.

Relevant input parameters

Connection to the MODBUS_COMM block:

- `FROM_MODBUS_COMM`

Configure a Modbus device address:

- `STATION_ADDRESS`

Configure device address of the analog values:

- `AO0[1..8]_ADDRESS`

Send respective output again for new connection establishment:

- `AO0[1..8]_RSND = TRUE`

Note

Node as Modbus client (primary)

Outputs are only sent again if the node is set as Modbus client (primary). If the station is set as Modbus device (secondary), the setting of the port AO0x_RSND is ignored.

See also

Modbus | application server and CN 4100 Modbus (Page 242)

A.16.2.5 MODBUS_AO8 (MODB4cnet) | Parameterize communication

Description

Configures the communication.

Relevant input parameters

Configure process variables for writing in the send buffer:

- `AO0[1..8]_PV`

Configure the command type for calling the register (for respective channel) from the Modbus devices:

- CNET as Modbus client (primary):
  - Deactivated:
    `AO0[1..8]_COMM_TYPE = 0`
  - Write single register (fc 6):
    `AO0[1..8]_COMM_TYPE = 6`
  - Write multiple registers (fc 16):
    `AO0[1..8]_COMM_TYPE = 16`
- CNET as Modbus server (secondary):
  - Read analog input (fc 9):
    `AO0[1..8]_COMM_TYPE = 9`

Note

CNET as Modbus server

The values can then be read by the Modbus client as follows:

- Read holding register of the analog outputs:
  `AO0[1..8]_COMM_TYPE = 3`
- Read registers of the analog inputs:
  `AO0[1..8]_COMM_TYPE = 4`

Note

Individual address area

In CNET there is a single address area for analog holding registers and analog input registers – they are not separated. The external Modbus client reads the same data from the same address, irrespective of whether function code 3 or 4 is used.

Configure the sequence for the writing of the data in the buffer:

Modbus | Register and BYTE swap (Page 242)

- Data outsourcing deactivated:
  `AO0[1..8]_SWAP = 0`
- Swapping of the byte sequence:
  `AO0[1..8]_SWAP = 1`
- Swapping of the register sequence:
  `AO0[1..8]_SWAP = 2`
- Swapping of the byte and register sequence:
  `AO0[1..8]_SWAP = 3`

Note

Same data for data exchange

The data exchange for writing the data into the storage must correspond to the data exchange for reading from the buffer.

Configure data type of the transferred values:

- Deactivated:
  `AO0[1..8]_TYPE = 0`
- 16 BITS without sign:
  `AO0[1..8]_TYPE = 1`
- 16 BITS with sign
  `AO0[1..8]_TYPE = 2`
- 32 BITS without sign
  `AO0[1..8]_TYPE = 3`
- 32 BITS with sign
  `AO0[1..8]_TYPE = 4`
- 32 BIT IEEE standard
  `AO0[1..8]_TYPE = 5`

Quality processing:

- Quality code of the process value is ignored and all process values are sent by the proxy, including the process values with invalid quality code:
  `QUALITY = FALSE`
- Block only sends the process values with valid quality code:
  `QUALITY = TRUE`

Configure signal range:

Limit values for the Modbus line:

- Low limit:
  `AO0[1..8]_rLL`
- High limit:
  `AO0[1..8]_rUL`.

Limit values for available scaled values:

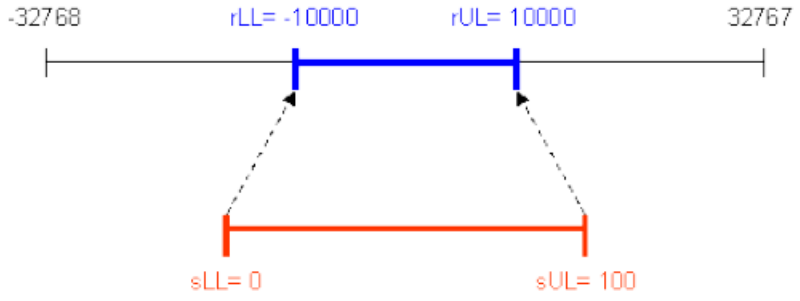
- Low limit:
  `AO0[1..8]_sLL`
- High limit:
  `AO0x[1..8]_sUL`.

Settings of the engineering unit of the (x) signal:

- `AO0[1..8]_sEU`

Example

The signals are scaled as shown in the following example:



The example applies to an input type of 16 bits with sign. A raw value range of -10000 to 10000 is scaled to a range of 0 to 100. The scaling of the raw values in the example is specified in the table.

Scaled value	Raw value
0	-10000
50	0
100	10000

Predefined signal ranges

The scaling is performed as follows if specific settings are made for the signal range:

- AO0x_rLL, AO0x_rUL, AO0x_sLL and AO0x_sUL are all = **0**
  The raw and scaled value ranges are set to the entire range of the selected data type. No scaling takes place between the raw values and the scaled values.
- AO0x_rLL and AO0x_rUL are = **0** and AO0x_sLL and AO0x_sUL are **not** = **0**
  The raw value range is set to the entire range of the selected data type. The scaled range is limited by the lower and upper limit values AO0x_sLL and AO0x_sUL. No scaling takes place between the raw values and the scaled values.
- AO0x_rLL and AO0x_rUL are **not** = **0**, and AO0x_sLL and AO0x_sUL are = **0**
  The raw value range is set to the entire range of the selected data type. The scaling on the raw value range takes place between the low and high limit values AO0x_rLL and AO0x_rUL.

The overall ranges of the data types are as follows:

Type	Low limit	High limit
16 bits without sign	0	65535
16 bits with sign	-32768	32767

Type	Low limit	High limit
32 bits without sign	0	4294967295
32 bits with sign	-2147483648	2147483647

Note

32-bit IEEE standard

For the 32 bit IEEE standard the limit values of the raw value range AO0x_rLL and AO0x_rUL have to be set equal to 0. Only the values for the scaled range AO0x_sLL and AO0x_sUL should be set. No scaling of the input takes place.

A.16.2.6 MODBUS_COMM (MODB4cnet) | Parameterize a connection

Description

Defines a communication connection to the Modbus device.

Modbus | application server and CN 4100 Modbus (Page 242)

Modbus (Page 241)

Relevant input parameters

Configure the station type:

- CN 4100 as Modbus client (primary):
  `STATION_TYPE = 10`
- CN 4100 as Modbus server (secondary):
  `STATION_TYPE = 11`

Configure protocol variant:

- For MODBUS-TCP:
  `PROT_VAR = 1`
- For MODBUS COM (RTU/Serial):
  `PROT_VAR = 2`

Assign parameters to configuration of the Modbus communication: (*) (**)

- `COM_PORT[1..2]`

* The configuration depends on the selection of PROT_VAR.

Note

MODBUS_COMM as Modbus server

If MODBUS_COMM is configured as Modbus server, the fields of COM_PORTx can be entered in one of the following formats:

- PORT
- IP-ADDRESS:PORT

** Only input COM_PORT1 is configured when establishing the communication with a non-redundant system. For connections with redundant systems the inputs COM_PORT1 and COM_PORT2 are referred as follows to the devices (A and B) of the redundant pair:

- The COM_PORT1 input refers to the device with the lower IP address.
- The COM_PORT2 input refers to the device with the higher IP address.

Note

Online changes at a COM_PORTx restart the Modbus communication of both redundancies.

Relevant output parameters

Configure connection to MODBUS devices.

- `TO_MODBUS_RD_WR`

Operation of MODBUS_COMM

Note

Activate / deactivate

The activation of MODBUS_COMM by itself (meaning the activation of MODBUS_COMM without the connected Modbus read/write proxy) has to be avoided. Otherwise flooding of the network is possible. In such a case the device has to be restarted.

The deactivation of connected Modbus read/write proxies can trigger a message at MODBUS_COMM and the other activated Modbus proxies. The message is output for 1 to 2 seconds, depending on the basic cycle time of the RTC, the cycle time of the proxies and the current Modbus communication partner.

Note

Configuring a sign of life signal

If CN 4100 is the Modbus client and this Modbus client only writes values (MODBUS_ AO/DO) and does not read values (MODBUS_AI/DI) and the values do not change for a long time, the block cannot detect a connection interruption or failure of the Modbus server.

To monitor this, a sign of life signal has to be configured or at least one value has to be read since the read requests are sent cyclically.

PROT_VAR = MODBUS TCP

The following configurations can be made in the dialog if PROT_VAR = MODBUS TCP:

Example	
IP_ADDRESS	10.116.28.52
PORT	502

PROT_VAR = MODBUS COM

The following configurations can be carried out in the dialog if PROT_VAR = MODBUS COM:

Topic	Properties	Value
DEVICE		/dev/ttyS1
BAUD (Page 243)		9600
DATA BITS (Page 243)	DATABITS_7 =	7
	DATABITS_8 =	8
STOP BITS (Page 243)	STOPBITS_1 =	1
	STOPBITS_2 =	2
PARITY (Page 244)	PARITY_NONE =	0
	PARITY_ODD =	1
	PARITY_EVEN =	2
FLOW CONTROL HARDWARE (Page 244)	FLOWCONTROL_NONE =	0
	FLOWCONTROL_RTSCTS_IN =	1
	FLOWCONTROL_RTSCTS_OUT =	2
FLOW CONTROL SOFTWARE (Page 244)	FLOWCONTROL_NONE =	0
	FLOWCONTROL_XONXOFF_IN =	4
	FLOWCONTROL_XONXOFF_OUT =	8
MODE (Page 245)	Serial ASCII mode =	ascii
	Serial RTU mode =	rtu

Note

Device configuration

Upper- and lower-case spelling has to be monitored during device configuration in the port COM_PORTx. The spelling of the configuration must correspond exactly to the format in the example shown above.

Note

Successful communication

To establish a successful communication the baud rate has to be the same as the one configured for the third party system.

Example

Example for a specific port string for serial RTU communication:

`/dev/ttyS0:9600:8:1:0:0:rtu.`

Note

Data bits

8 data bits can be used for RTU.

7 and 8 data bits can be used for ASCII.

A.16.2.7 MODBUS_COMM (MODB4cnet) | Parameterize communication

Description

Configures the communication.

Relevant input parameters

Configure 'Read/Write" operating mode:

- R-W-R-W sequence:
  With this operating mode the read and write requests have the same priority. The read request is always executed first and if a write request is pending, it is executed before the next read request.
  `RW_MODE = 0`
- Scan-based reading:
  With this operating mode the read request is executed for all nodes and then the write request for all the required nodes. In this mode, the data can be read from the Modbus server (secondary) in a fixed time interval.
  `RW_MODE = 1`

Configure dummy points as a request:

- As an analog value:
  `GAP_SPACE = 0`
- As a binary value:
  `GAP_SPACE = FALSE`

Note

Reducing read/write requests

The read/write requests are reduced with the dummy points.

Example: Addresses 1, 2, 3, 4, 5, 8, 9 and 10 are configured and two requests for read/write of all addresses are required. When the dummy parameters are set to 2, all addresses together with dummy points for 6 and 7 are read/written with only one request.

Configure confirmation of the length of a message:

- Send confirmation:
  `ENABLE_LENGHT_CHECK = TRUE`
- Do not send a confirmation:
  `ENABLE_LENGHT_CHECK = FALSE`

Time and retry settings:

Assigning parameters of the number of overlooked exceptions: (Ignore exceptions)

- `EX_RETRY_NUM`

Note

Scenario

Some systems send exceptions during normal operation.

- If the specified number has been exceeded, a reply timeout (with T-RES) is initiated.
- If a fault free reply is received, the number of the current exceptions is reset to zero.

Configure the maximum permissible duration for the reply to a request:

- `T_RES`

Configure the number of retries:

- `RETRY_NUM`

Note

Retries

It is possible that replies are not received directly after the first request. If no reply has been received within T_RES, the system tries again to send the request.

If the retries are not successful:

- The quality code of all input signals in the telegram is set to BAD.
- The TRBL_AL output of all proxies in the telegram connected with signals is set to TRUE.

Configure the data traffic timeout:

- `TRAFFIC_TIMEOUT`
- Conditions:
  - TRAFFIC_TIMEOUT >= (T_RES * RETRY_NUM)
  - Or: TRAFFIC_TIMEOUT > T_RES > T_INTER

Note

Message

If no data traffic (either requests or replies) occurs during the time duration of TRAFFIC_TIMEOUT, the connection is interrupted and a message is generated.

Configure the time interval between two queries:

- T_INTER

Configure the time for the query of input signals from the device:

- T_SCAN
- Conditions:
  - RW_MODE = 1
  - T_SCAN > 'Time required to read all data from the Modbus device (secondary)'

Note

Write data to the Modbus server (secondary)

For example, if T_SCAN is set to 20 ms and the Modbus client (primary) reads the data in 15 ms, the remaining time, i.e. 5 ms, is used to write the data to the Modbus server (secondary).

A.16.2.8 MODBUS_DI01, MODBUS_DI08, MODBUS_DI16 (MODB4cnet) | Parameterize a connection

Description

Defines a communication connection to the Modbus device.

Relevant input parameters

Connection to MODBUS_COMM block:

- FROM_MODBUS_COMM

Parameterizing a Modbus device address:

- STATION_ADDRESS

Parameterizing the address of the first register of the digital input values:

- START_ADDRESS

See also

Modbus | application server and CN 4100 Modbus (Page 242)

A.16.2.9 MODBUS_DI01, MODBUS_DI08, MODBUS_DI16 (MODB4cnet) | Parameterize communication

Description

Configures the communication.

Relevant input parameters

Configure the command type for calling the register from the Modbus devices:

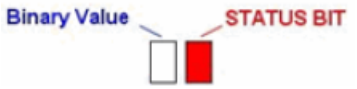
- Read coils (fc 1):
  `COMMAND_TYPE = 1`
- Read discrete inputs (fc 2):
  `COMMAND_TYPE = 2`

Configure the number of values which are to be read from the Modbus device:

- `POINT_LENGTH = 1`

Configure data type of the transferred values:

- Transfer a bit which contains the signal (standard signal):
  `DI_TYPE = 0`
- Additionally transfer a bit for the quality status of the signal (SIGNAL_WITHSTATUS):
  `DI_TYPE = 1`



Note

Data type SIGNAL_WITHSTATUS

The SIGNAL_WITHSTATUS data type is only supported in primary stations (Modbus client) and not in secondary stations (Modbus server).

A.16.2.10 MODBUS_DO8 (MODB4cnet) | Parameterize a connection

Description

Defines a communication connection to the Modbus device.

Relevant input parameters

Connection to the MODBUS_COMM block:

- `FROM_MODBUS_COMM`

Configure a Modbus device address:

- `STATION_ADDRESS`

Configure a device address of the digital values:

- `DO0[1..8]_ADDRESS`

Send respective output again for new connection establishment:

- `DO0[1..8]_RSND = TRUE`

Note

Node as Modbus client (primary)

Outputs are only sent again if the node is set as Modbus client (primary). If the station is set as Modbus server (secondary), the setting of the port DO0x_RSND is ignored.

See also

Modbus | application server and CN 4100 Modbus (Page 242)

A.16.2.11 MODBUS_DO8 (MODB4cnet) | Parameterize communication

Description

Configures the communication.

Relevant input parameters

Configure process variables for writing in the send buffer:

- `DO0[1..8]_PV`

Configure the command type for calling the register (for respective channel) from the Modbus devices:

- CNET as Modbus client (primary):
  - Deactivated:
    `DO0[1..8]_COMM_TYPE = 0`
  - Writing a single register (fc 5):
    `DO0[1..8]_COMM_TYPE = 1`
  - Writing multiple registers (fc 15):
    `DO0[1..8]_COMM_TYPE = 2`
- CNET as Modbus server (secondary):
  - Reading a binary input (fc 9):
    `DO0[1..8]_COMM_TYPE = 9`

Note

CNET as Modbus server (secondary)

The values can then be read as follows by the Modbus client (primary):

- Reading discrete outputs (coils):
  `DO0[1..8]_COMM_TYPE = (fc1)`
- Reading discrete inputs (contacts):
  `DO0[1..8]_COMM_TYPE = (fc2)`

Note

Individual address area

In CNET there is a single address area for discrete coils and discrete inputs – they are not separated. The external Modbus client reads the same data from the same address, irrespective of whether function code 1 or 2 is used.

Quality processing:

- Quality code of the process value is ignored and all process values are sent by the proxy, including the process values with invalid quality code:
  `QUALITY = FALSE`
- Block only sends the process values with valid quality code:
  `QUALITY = TRUE`

A.16.3 Communication

A.16.3.1 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Monitor communication

Description

Monitors connections and communication between devices.

Relevant output parameters

Make (x) process variables available:

- `AI0x_PV`

Show limit values of the (x) signal:

- High limit:
  `AI0x_UL`
- Low limit:
  `AI0x_LL`

Show setting of the engineering unit of the (x) signal:

- `AI0x_EU`

Show status of the redundant communications interfaces:

- No fault:
  `PROXY_STATUS[1..2] = 0`
- Station address not available:
  `PROXY_STATUS[1..2] = 1`
- Point address not found:
  `PROXY_STATUS[1..2] = 8`
- Invalid command:
  `PROXY_STATUS[1..2] = 16`
- Invalid command and address not found:
  `PROXY_STATUS[1..2] = 24`
- Station address not ready and invalid command:
  `PROXY_STATUS[1..2] = 17`
- Station address not ready and address not found:
  `PROXY_STATUS[1..2] = 9`
- Station address not ready and address not found and invalid command:
  `PROXY_STATUS[1..2] = 25`

A.16.3.2 MODBUS_AO8, MODBUS_DO8 (MODB4cnet) | Monitor communication

Description

Monitors connections and the communication between devices.

Relevant output parameters

Show status of the redundant communications interfaces:

- No fault:
  `PROXY_STATUS[1..2] = 0`
- Station address not available:
  `PROXY_STATUS[1..2] = 1`
- Point address not found:
  `PROXY_STATUS[1..2] = 8`
- Invalid command:
  `PROXY_STATUS[1..2] = 16`
- Invalid command and address not found:
  `PROXY_STATUS[1..2] = 24`
- Station address not ready and invalid command:
  `PROXY_STATUS[1..2] = 17`

- Station address not ready and address not found:
  `PROXY_STATUS[1..2] = 9`
- Station address not ready and address not found and invalid command:
  `PROXY_STATUS[1..2] = 25`

Communication interrupted or faulty:

- `TRBL_AL = TRUE`

TRBL_AL = TRUE and the other communication channel has also failed:

- `MULTIPLE_AL = TRUE`

MULTIPLE_AL is reset to FALSE as soon as the message has been acknowledged.

A.16.3.3 MODBUS_COMM (MODB4cnet) | Monitor communication

Description

Monitors connections and the communication between devices.

Relevant output parameters

(The following parameters apply for the diagnostics view.)

Show status of the protocol and communications interfaces:

- No fault:
  `STATUS_COM[1..2] = 0`
- Connected:
  `STATUS_COM[1..2] = 1`
- Not connected:
  `STATUS_COM[1..2] = 2`
- Connects:
  `STATUS_COM[1..2] = 3`
- Disconnecting:
  `STATUS_COM[1..2] = 4`
- Driver cannot be connected:
  `STATUS_COM[1..2] = -1`
- Connection to RTU not established:
  `STATUS_COM[1..2] = -2`
- Invalid connection:
  `STATUS_COM[1..2] = -3`
- Connection not available:
  `STATUS_COM[1..2] = -7`

- Monitored:
  `STATUS_COM[1..2] = 9`
- Interrupt(!):
  `STATUS_COM[1..2] = 14`
- Interrupted(!):
  `STATUS_COM[1..2] = 24`

Shows the required number of the requests for scanning the read data:

- `REQ_COUNT`

Shows the last required time [ms] for processing all read requests:

- For part device A:
  `CYCLE_A`
- For part device B:
  `CYCLE_B`

Note

Time

The specification can be checked at CYCLE_A, CYCLE_B if T_SCAN was set to 20 ms.

Note

Without redundancy

In case of a non-redundant configuration, the value 0 is output for CYCLE_B.

A.16.3.4 MODBUS_DI01, MODBUS_DI08, MODBUS_DI16 (MODB4cnet) | Monitor communication

Description

Monitors connections and the communication between devices.

Relevant output parameters

Make (x) process variables available:

- `DIxx_PV`

Show status of the redundant communications interfaces:

- No fault:
  `PROXY_STATUS[1..2] = 0`
- Station address not available:
  `PROXY_STATUS[1..2] = 1`

- Point address not found:
  `PROXY_STATUS[1..2] = 8`
- Invalid command:
  `PROXY_STATUS[1..2] = 16`
- Invalid command and address not found:
  `PROXY_STATUS[1..2] = 24`
- Station address not ready and invalid command:
  `PROXY_STATUS[1..2] = 17`
- Station address not ready and address not found:
  `PROXY_STATUS[1..2] = 9`
- Station address not ready and address not found and invalid command:
  `PROXY_STATUS[1..2] = 25`

A.16.4 Alarming, error handling

A.16.4.1 MODBUS_AI01, MODBUS_AI08, MODBUS_AI16 (MODB4cnet) | Parameterize troubleshooting

Description

Configures troubleshooting.

Relevant output parameters

Communication interrupted or faulty:

- `TRBL_AL = TRUE`

TRBL_AL = TRUE and the other communication channel has also failed:

- `MULTIPLE_AL = TRUE`

MULTIPLE_AL is reset to FALSE as soon as it has been acknowledged.

Conditions for error handling	Effect on further processing	Effect on the quality code
No telegram received	-	AI01_PV = BCF
Only applies for MODBUS_AI01: For a station of type Modbus device (secondary), the COMMAND_ TYPE input is set to ABB system buffer read (fc 3).	-	AI01_PV = BAD

A.16.4.2 MODBUS_COMM (MODB4cnet) | Parameterize troubleshooting

Description

Configures troubleshooting.

Relevant output parameters

Communication interrupted or faulty:

- `TRBL_AL = TRUE`

TRBL_AL = TRUE and the other communication channel has also failed:

- `MULTIPLE_AL = TRUE`

MULTIPLE_AL is reset to FALSE as soon as it has been acknowledged.

A.16.4.3 MODBUS_DI01, MODBUS_DI08, MODBUS_DI16 (MODB4cnet) | Parameterize troubleshooting

Description

Configures troubleshooting.

Relevant output parameters

Communication interrupted or faulty:

- `TRBL_AL = TRUE`

TRBL_AL = TRUE and the other communication channel has also failed:

- `MULTIPLE_AL = TRUE`

MULTIPLE_AL is reset to FALSE as soon as it has been acknowledged.

Conditions for troubleshooting	Effect on further processing	Effect on the quality code
No telegram received	-	DI01_PV = BCF

A.17 SYS4cnet | System-Library

A.17.1 Blocks

A.17.1.1 BW_RTC_DIAG (SYS4cnet) | Description

Description

Calls up diagnostics data for the runtime container.

Parameter of inputs

--

Parameter of outputs

(*) = Optional parameter

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
WORKLOAD	INT	Workload [%] (max. 80%)			x	x	x	
WORKLOAD_MAX	INT	Maximum workload [%] (max. 80%)			x	x	x	
WORKLOAD_AVG	INT	Average workload [%] (max. 80%)			x	x	x	
CYCLE_AL	BOOL	Cycle exceeded	x	Came in / went out	x	x	x	

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
STATE_SYSTEM_A ... STATE_SYSTEM_B	INT	Redundancy status of the system (A..B) (*)	x	(came in / went out)	x	x	x	
COMM_ERROR_A ... COMM_ERROR_B	BOOL	System error for the communication connection	x	Came in / went out	x	x	x	

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
STATE_RT_COM	INT	Status of the redundant connection (*)	x	(came in / went out)	x	x	x	
RED_LOST_AL	BOOL	Redundancy loss	x	Came in / went out	x	x	x	
SYNC_BUFFER_SIZE	INT	Transferred synchronization buffer: [BYTE]	x	(came in / went out)	x	x	x	
SYNC_BUFFER_USAGE	INT	Reserved synchronization buffer [BYTE]	x	(came in / went out)	x	x	x	

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
NR_OF_SUBSCRIPTIONS	INT	Number of current monitorings	x	(came in / went out)	x	x	x	
NR_OF_PUBLISHER	INT	Number of current publishers	x	(came in / went out)	x	x	x	
NR_OF_MODULES	INT	Number of the modules	x	(came in / went out)	x	x	x	
DELTA_FACTOR	INT	Delta factor for the hysteresis	x	(came in / went out)	x	x	x	
PUBL_BUFFER_USAGE	INT	Use of the publisher buffer [%]	x	(came in / went out)	x	x	x	
ENTRY_BUFFER_USAGE	INT	Use of the buffer of the inputs [%]	x	(came in / went out)	x	x	x	

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
ACTIVATION_TRANSACTION	BOOL	Activation of pending transaction	x	(came in / went out)	x	x	x	
STATE_COM_INTF [0..2]_[A..B]	INT	Communication protocol for plugin (0..2) of the system (A..B) (*)	x	(came in / went out)	x	x	x	

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
RT_COM_FAILURE	BOOL	Failure of one of the redundancy connections	x	Came in / went out	x	x	x	
COM_INTF_ERROR_A ... COM_INTF_ERROR_B	BOOL	Error of the communication interfaces of the system (A..B)	x	Came in / went out	x	x	x	

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
ARCH_EVENT_RATE_AVG	INT	Average archive result rate per second (smoothed)	x	(came in / went out)	x	x	x	
ARCH_EVENT_RATE_MAX	INT	Maximum archive event rate per second (smoothed)	x	(came in / went out)	x	x	x	
ARCH_BUFFERED_EVENTS	INT	Number of buffered archive events	x	(came in / went out)	x	x	x	
ARCH_EVENT_TIME_DELAY	INT	Time delay of the oldest archive result:	x	(came in / went out)	x	x	x	

Parameter	Data type	Description	Message / message class	Active / inactive	Conn	L	HMI	DIA
ARCH_EVENT_DELAY_AL	BOOL	Transfer to the archive event delayed	x	Came in / went out	x	x	x	
ARCH_BUFFER_OVERFLOW_AL	BOOL	Buffer overflow for archive events	x	Came in / went out	x	x	x	
ARCH_RECOVER_ABORTED_AL	BOOL	Restoring of the archive event aborted	x	Came in / went out	x	x	x	
ARCH_BUFFER_THRESHOLD_AL	BOOL	Buffer limit for archive event exceeded	x	Came in / went out	x	x	x	

Functions

Configuring the mode of operation (Page 485)

See also

General quality statuses (Page 223)
Parameter tables (Page 223)

A.17.2 Configuration

A.17.2.1 BW_RTC_DIAG (SYS4cnet) | Configure mode of operation

Description

Configures the mode of operation.

Relevant output parameters

Show workload:

- `WORKLOAD`
- `WORKLOAD_MAX`
- `WORKLOAD_AVG`

Message of a cycle overrun:

- `CYCLE_AL = TRUE`

System error for the communication connection (A..B):

- COMM_ERROR_[A..B] `= TRUE`

Redundancy loss:

- `RED_LOST_AL = TRUE`

BYTE of the synchronization buffer:

- Transferred:
  `SYNC_BUFFER_SIZE`
- Reserved:
  `SYNC_BUFFER_USAGE`

Configuration limits to...:

- Observations:
  `NR_OF_SUBSCRIPTIONS`
- Publisher:
  `NR_OF_PUBLISHER`
- Module:
  `NR_OF_MODULES`
- Delta factor:
  `DELTA_FACTOR`

- Publisher buffer:
  `PUBL_BUFFER_USAGE`
- Buffer of the inputs:
  `ENTRY_BUFFER_USAGE`

Messages for communication connection:

- Pending transaction:
  `ACTIVATION_TRANSACTION = TRUE`
- Failure of redundancy:
  `RT_COM_FAILURE = TRUE`
- Error at the interface:
  COM_INTF_ERROR_[A..B] `= TRUE`

Measuring values for archive events:

- ARCH_EVENT_RATE_AVG
- ARCH_EVENT_RATE_MAX
- ARCH_BUFFERED_EVENTS
- ARCH_EVENT_TIME_DELAY

Alarms for archive events:

- Delay:
  `ARCH_EVENT_DELAY_AL = TRUE`
- Buffer overflow:
  `ARCH_BUFFER_OVERFLOW_AL = TRUE`
- No restoring:
  `ARCH_RECOVER_ABORTED_AL = TRUE`
- Limit violation:
  `ARCH_BUFFER_THRESHOLD_AL = TRUE`

Optional output parameters

Redundancy status of the system (A..B)

STATE_SYSTEM_[A..B] =

- Default:
  `0`
- Shutdown:
  `1`
- Starting:
  `2`
- Stopped:
  `3`
- Started:
  `4`

- Synchronization started:
  5
- Standby:
  6
- Master stand-alone:
  7
- Master synchronizes standby:
  8
- Master:
  9
- Freezed:
  10

Status of the redundant connection:

STATE_RT_COM =

- Default:
  0
- Connection 1 down:
  1
- Connection 2 down:
  2
- Not available:
  3

Communication protocol for plugin (0..2) of the system (A..B):

STATE_COM_INTF[0..2]_[A..B] =

- Default:
  0
- Starting:
  1
- Running:
  2
- Shutdown required:
  3
- Error:
  4