SIEMENS

SIMATIC

Process Control System PCS 7 PCS 7 Industry Library for PCS 7 (V9.0 SP1)

Function Manual

Security information 1

Basics 2

Adapter blocks 3

APC - Connecting higher-level controllers 4

Conversion Blocks 5

Operator control blocks 6

Count 7

Energy management blocks 8

Function-based control blocks 9

HVAC blocks 10

Communication blocks 11

Logic blocks 12

Mathematical blocks 13

Motor and valve blocks 14

Panel blocks 15

Controller blocks 16

Continued on next page

09/2020
A5E42064404-AB

Siemens AG
Division Process Industries and Drives
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E42064404-AB
Ⓟ 09/2020 Subject to change

Copyright © Siemens AG 2020.
All rights reserved

Process Control System PCS 7 PCS 7 Industry Library for PCS 7 (V9.0 SP1)

Function Manual

Continued

System blocks 17

Monitoring blocks 18

Interlock blocks 19

Maintenance blocks 20

S88 structure interfaces and batch interface 21

Technological blocks 22

Timers 23

Appendix 24

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 AG. 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 AG
Division Process Industries and Drives
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E42064404-AB
Ⓟ 09/2020 Subject to change

Copyright © Siemens AG 2020.
All rights reserved

Table of contents

1 Security information 31

2 Basics 33

2.1 About this document 33

2.2 Overview of the function blocks 34

2.3 Specialties 38
2.3.1 Dependency on APL 38
2.3.2 Setting of the standard server 38
2.3.3 Functions of the blocks 38
2.3.4 Functions of the faceplates 38
2.3.5 Format Specification for analog value display 39

2.4 Multiple control room concept 41
2.4.1 Description of the concept 41
2.4.2 Configuring operating levels on the operator station (OS) 42
2.4.3 Configuring operating levels on the operator panel (OP) 42
2.4.4 Table with possible values of the operating levels 43

2.5 Panel integration 44
2.5.1 Description of the concept 44
2.5.2 Interface to the operator panel 46
2.5.3 Overview of IL Comfort types 47
2.5.4 Operator control and monitoring in WinCC Comfort 49
2.5.4.1 Configuration of the panel interface 49
2.5.4.2 General view of the panel blocks 50
2.5.4.3 Messages with activated function "User-configurable message classes" 52
2.5.5 Operator Control and Monitoring in WinCC 54
2.5.5.1 Block icon of the panel blocks 54
2.5.5.2 Panel block views 54

2.6 File dialog 56
2.6.1 File dialog function 56
2.6.2 Configuration of file dialog 56
2.6.3 File dialog error handling 59
2.6.4 File dialog views 59

3 Adapter blocks 61

3.1 General 61

3.2 Basics of adapter blocks 62
3.2.1 Configuration 63
3.2.2 Mode of operation 64
3.2.3 Common connections of the type-specific adapter blocks 64
3.2.4 Faceplate jumps CM block <-> Adapter block 67
3.2.5 Command process releases 68
3.2.6 ON and OFF delays 69
3.2.6.1 Block interfaces 69

3.2.7 Higher-level controlling of the unit / EQM level ... 70
3.2.7.1 Structure data connections EQM <-> CM ... 70
3.2.7.2 Command and feedback processing for 'FxCtrl' ... 72
3.2.7.3 Automatic releases at unit allocation ... 73
3.2.7.4 Higher-level simulation ... 74
3.2.7.5 Message lock ... 75
3.2.8 Enabling for manual switching operation / Block for simulation operation ... 75
3.2.8.1 Block interfaces ... 76
3.2.9 Maintenance data ... 76
3.2.9.1 Block interfaces ... 77
3.2.10 Operator control and monitoring ... 77

3.3 Adapter blocks – details ... 82
3.3.1 ApMotL ... 82
3.3.1.1 Connections of ApMotL ... 84
3.3.1.2 Operator control and monitoring ... 85
3.3.2 ApMotRevL ... 86
3.3.2.1 Connection of ApMotRevL ... 89
3.3.2.2 Operator control and monitoring ... 90
3.3.3 ApMotSpdCL ... 91
3.3.3.1 Connections of ApMotSpdCL ... 93
3.3.3.2 Operator control and monitoring ... 94
3.3.4 ApVlvL ... 95
3.3.4.1 Connections of ApVlvL ... 98
3.3.4.2 Operator control and monitoring ... 99
3.3.5 ApVlvAnL ... 100
3.3.5.1 Connections of ApVlvAnL ... 102
3.3.5.2 Operator control and monitoring ... 103
3.3.6 ApVlvDS ... 104
3.3.6.1 Connections of ApVlvDS ... 107
3.3.6.2 Operator control and monitoring ... 108
3.3.7 ApCmGen ... 108
3.3.7.1 Connections of ApCmGen ... 111
3.3.7.2 Operator control and monitoring ... 113
3.3.8 ApPIDConL ... 114
3.3.8.1 Functions of 'ApPIDConL' ... 116
3.3.8.2 Connections of 'ApPIDConL' ... 118
3.3.8.3 Operator control and monitoring ... 120

4 APC - Connecting higher-level controllers ... 125

4.1 Description of APC coupling blocks ... 125

4.2 APC_Supervisor ... 133
4.2.1 APC_Supervisor description ... 133
4.2.2 APC_Supervisor operating modes ... 135
4.2.3 APC_Supervisor functions ... 136
4.2.4 APC_Supervisor error handling ... 138
4.2.5 APC_Supervisor messaging ... 139
4.2.6 APC_Supervisor I/Os ... 140
4.2.7 APC_Supervisor block diagram ... 144
4.2.8 Operator control and monitoring ... 145
4.2.8.1 APC_Supervisor views ... 145
4.2.8.2 APC_Supervisor standard view ... 145

4.2.8.3 APC_Supervisor setpoint/controller list view ... 148
4.2.8.4 APC_Supervisor process tag list view ... 150
4.2.8.5 APC_Supervisor parameter view ... 151
4.2.8.6 APC_Supervisor preview ... 151
4.2.8.7 APC_Supervisor block icons ... 154

4.3 APC_OpSP ... 156
4.3.1 APC_OpSP description ... 156
4.3.2 APC_OpSP operating modes ... 157
4.3.3 APC_OpSP function ... 157
4.3.4 APC_OpSP error handling ... 158
4.3.5 APC_OpSP messaging ... 159
4.3.6 APC_OpSP I/Os ... 160
4.3.7 APC_OpSP block diagram ... 162
4.3.8 Operator control and monitoring ... 162
4.3.8.1 APC_OpSP views ... 162
4.3.8.2 APC_OpSP standard view ... 163
4.3.8.3 APC_OpSP block icons ... 164

4.4 APC_MV ... 166
4.4.1 Description of APC_MV ... 166
4.4.2 APC_MV operating modes ... 166
4.4.3 APC_MV function ... 166
4.4.4 APC_MV error handling ... 167
4.4.5 APC_MV messaging ... 167
4.4.6 APC_MV I/Os ... 168
4.4.7 APC_MV block diagram ... 170
4.4.8 Operator control and monitoring of APC_MV ... 170

4.5 APC_MpList ... 171
4.5.1 APC_MpList description ... 171
4.5.2 APC_MpList operating modes ... 171
4.5.3 APC_MpList functions ... 171
4.5.4 APC_MpList error handling ... 171
4.5.5 APC_MpList messaging ... 171
4.5.6 APC_MpList I/Os ... 172
4.5.7 APC_MpList block diagram ... 172
4.5.8 Operator control and monitoring of APC_MpList ... 172

5 Conversion Blocks ... 173

5.1 IntBit – Conversion of an integer to a bit mask ... 173

5.2 BitInt – Conversion of a bit mask to an integer ... 174

5.3 CmDec – Structure decoding 'CMData' ... 175

5.4 CmEnc – Structure encoding 'CMData' ... 176

5.5 MidStr – Conversion of material ID to string ... 177

5.6 StrMid – Conversion of string to material ID ... 178

5.7 OpA16Dec – Structure decoding 'Opa16' ... 179

6 Operator control blocks ... 181

6.1 Aggr16 / Aggr08 - Unit switchover of 16/08 units ... 181
6.1.1 Aggr16/ Aggr08 description ... 181

6.1.2 Aggr16/ Aggr08 operating modes ... 183
6.1.3 Aggr16/ Aggr08 functions ... 184
6.1.4 Aggr16/ Aggr08 error handling ... 187
6.1.5 Aggr16/ Aggr08 messaging ... 188
6.1.6 Aggr16/ Aggr08 I/Os ... 190
6.1.7 Aggr16/ Aggr08 block diagram ... 193
6.1.8 Operator control and monitoring ... 193
6.1.8.1 Aggr16/ Aggr08 views ... 193
6.1.8.2 Aggr16/ Aggr08 standard view ... 194
6.1.8.3 Aggr16/ Aggr08 parameter view ... 197
6.1.8.4 Aggr16/ Aggr08 preview ... 199
6.1.8.5 Aggr16/ Aggr08 block icon ... 200

6.2 OpAn16 – Operation of 16 parameters ... 203
6.2.1 Description of 'OpAn16' ... 203
6.2.2 Functions of 'OpAn16' ... 203
6.2.3 Connections of 'OpAn16' ... 205
6.2.4 Operator control and monitoring ... 207
6.2.4.1 Symbol ... 207
6.2.4.2 Faceplate ... 207

6.3 OpEnum – Operation of enumeration parameters ... 210
6.3.1 Description of 'OpEnum' ... 210
6.3.2 Functions of 'OpEnum' ... 211
6.3.3 Connections of 'OpEnum' ... 212
6.3.4 Operator control and monitoring ... 213
6.3.4.1 Symbol ... 213
6.3.4.2 Faceplate ... 214

6.4 ParaCtrl / ParaMem - Block for handling small numbers of parameters and parameter sets ... 216
6.4.1 ParaCtrl / ParaMem description ... 216
6.4.2 ParaCtrl / ParaMem operating modes ... 225
6.4.3 ParaCtrl / ParaMem functions ... 225
6.4.4 ParaCtrl / ParaMem error handling ... 230
6.4.5 ParaCtrl / ParaMem messaging ... 231
6.4.6 ParaCtrl / ParaMem I/Os ... 231
6.4.7 ParaCtrl / ParaMem block diagram ... 234
6.4.8 Operator control and monitoring ... 234
6.4.8.1 ParaCtrl / ParaMem views ... 234
6.4.8.2 ParaCtrl / ParaMem standard view ... 235
6.4.8.3 ParaCtrl / ParaMem parameter view ... 238
6.4.8.4 ParaCtrl / ParaMem preview ... 240
6.4.8.5 ParaCtrl / ParaMem block icon ... 241

6.5 SelFp - Jump distributor ... 242
6.5.1 SelFp description ... 242
6.5.2 SelFp operating modes ... 243
6.5.3 SelFp functions ... 243
6.5.4 SelFp error handling ... 243
6.5.5 SelFp messaging ... 243
6.5.6 SelFp I/Os ... 244
6.5.7 SelFp block diagram ... 244
6.5.8 Operator control and monitoring ... 244
6.5.8.1 SelFp views ... 244

6.5.8.2 SelFp standard view ... 245
6.5.8.3 SelFp block icon ... 245

6.6 UsrM - User management ... 246
6.6.1 UsrM description ... 246
6.6.2 UsrM operating modes ... 246
6.6.3 UsrM functions ... 246
6.6.4 UsrM error handling ... 247
6.6.5 UsrM messaging ... 247
6.6.6 UsrM I/Os ... 249
6.6.7 UsrM block diagram ... 250
6.6.8 Operator control and monitoring ... 250
6.6.8.1 UsrM views ... 250
6.6.8.2 UsrM standard view ... 250
6.6.8.3 UsrM block icon ... 251

7 Count ... 253

7.1 PuPa – Pulse sequence generator ... 253
7.1.1 Description of 'PuPa' ... 253
7.1.2 Operating modes of 'PuPa' ... 254
7.1.3 Functions of 'PuPa' ... 255
7.1.4 Connections of 'PuPa' ... 259
7.1.5 Operator control and monitoring ... 262
7.1.5.1 Symbol ... 262
7.1.5.2 Faceplate ... 262

8 Energy management blocks ... 269

8.1 Description of the energy management blocks ... 269

8.2 LdMgmt8 - Load management block for coordination of up to 8 consumers ... 272
8.2.1 LdMgmt8 description ... 272
8.2.2 LdMgmt8 operating modes ... 275
8.2.3 LdMgmt8 functions ... 275
8.2.4 of LdMgmt8 error handling ... 285
8.2.5 LdMgmt8 messaging ... 286
8.2.6 LdMgmt8 I/Os ... 287
8.2.7 LdMgmt8 block diagram ... 291
8.2.8 Operator control and monitoring ... 291
8.2.8.1 LdMgmt8 views ... 291
8.2.8.2 LdMgmt8 standard view ... 292
8.2.8.3 LdMgmt8 energy data view ... 295
8.2.8.4 LdMgmt8 time view ... 297
8.2.8.5 LdMgmt8 parameter view ... 297
8.2.8.6 LdMgmt8 preview ... 299
8.2.8.7 LdMgmt8 block icons ... 301

8.3 CalcPwrL - Acquisition and calculation of energy consumption values ... 303
8.3.1 CalcPwrL description ... 303
8.3.2 CalcPwrL operating modes ... 305
8.3.3 CalcPwrL functions ... 305
8.3.4 CalcPwrL error handling ... 312
8.3.5 CalcPwrL messaging ... 313
8.3.6 CalcPwrL I/Os ... 314
8.3.7 CalcPwrL block diagram ... 318

8.3.8 Operator control and monitoring ... 319
8.3.8.1 CalcPwrL views ... 319
8.3.8.2 CalcPwrL standard view ... 320
8.3.8.3 Limit view of CalcPwrL ... 323
8.3.8.4 CalcPwrL parameter view ... 325
8.3.8.5 CalcPwrL histogram view ... 327
8.3.8.6 CalcPwrL preview ... 330
8.3.8.7 CalcPwrL block icons ... 331

8.4 PeakMon - Monitoring and forecast of consumption peaks at the plant infeed ... 333
8.4.1 PeakMon description ... 333
8.4.2 PeakMon operating modes ... 335
8.4.3 PeakMon functions ... 336
8.4.4 PeakMon error handling ... 342
8.4.5 PeakMon messaging ... 342
8.4.6 PeakMon I/Os ... 344
8.4.7 PeakMon block diagram ... 348
8.4.8 Operator control and monitoring ... 348
8.4.8.1 PeakMon views ... 348
8.4.8.2 PeakMon standard view ... 349
8.4.8.3 PeakMon limit view ... 354
8.4.8.4 PeakMon parameter view ... 355
8.4.8.5 PeakMon histogram view ... 357
8.4.8.6 PeakMon preview ... 358
8.4.8.7 PeakMon block icons ... 359

8.5 PulseCon - Pulse converter ... 361
8.5.1 PulseCon description ... 361
8.5.2 PulseCon operating modes ... 361
8.5.3 PulseCon functions ... 362
8.5.4 PulseCon error handling ... 363
8.5.5 PulseCon messaging ... 363
8.5.6 PulseCon I/Os ... 364
8.5.7 PulseCon block diagram ... 364
8.5.8 Operator control and monitoring of PulseCon ... 364

9 Function-based control blocks ... 365

9.1 FxCtrl – Function-based control and monitoring block ... 365
9.1.1 Operating modes of 'FxCtrl' ... 371
9.1.2 Functions of 'FxCtrl' ... 372
9.1.2.1 Connection possibilities ... 372
9.1.2.2 Function-based activation and monitoring ... 373
9.1.2.3 Function states, transitions ... 375
9.1.2.4 Error monitorings ... 376
9.1.2.5 Overlay functions ... 378
9.1.2.6 Element-wise forced operation ... 378
9.1.2.7 Trigger for feedback simulation ... 378
9.1.2.8 Passing on the initial state request ... 379
9.1.2.9 APL standard functions of 'FxCtrl' ... 379
9.1.3 Messages of 'FxCtrl' ... 381
9.1.4 Connections of 'FxCtrl' ... 382
9.1.5 Configuration of the function definitions ... 385
9.1.5.1 Configuration dialog "Project overview screen" ... 386

9.1.5.2 Configuration dialog "Editing window" ... 387
9.1.6 Operator control and monitoring ... 390
9.1.6.1 Symbol ... 390
9.1.6.2 Faceplate ... 391

9.2 FxSigBin – Binary signal routing block for FxCtrl ... 400
9.2.1 Description of 'FxSigBin' ... 400
9.2.2 Functions of 'FxSigBin' ... 401
9.2.3 Connections of 'FxSigBin' ... 403

9.3 FxSigInt – INTEGER - Signal routing block for FxCtrl ... 405
9.3.1 Description of 'FxSigInt' ... 405
9.3.2 Functions of 'FxSigInt' ... 406
9.3.3 Connection of 'FxSigInt' ... 407

9.4 FxDef16 / 32 / 64 – Function definition block for FxCtrl ... 410
9.4.1 Description of 'FxDef16 / 32 / 64' ... 410
9.4.2 Functions of 'FxDef16 / 32 / 64' ... 410
9.4.3 Connections of 'FxDef16 / 32 / 64' ... 411

9.5 FxShare – Function definition block for FxCtrl ... 412
9.5.1 Description of 'FxShare' ... 412
9.5.2 Functions of 'FxShare' ... 414
9.5.2.1 Connection possibilities ... 414
9.5.2.2 Allocation request and control of the EQM ... 415
9.5.2.3 APL standard functions of 'FxShare' ... 416
9.5.3 Connections of 'FxShare' ... 417
9.5.4 Operator control and monitoring ... 419
9.5.4.1 Symbol ... 419
9.5.4.2 Faceplate ... 420

10 HVAC blocks ... 423

10.1 CalcWatP - Calculation of thermal power and emitted energy ... 423
10.1.1 CalcWatP description ... 423
10.1.2 CalcWatP operating modes ... 425
10.1.3 CalcWatP functions ... 425
10.1.4 CalcWatP error handling ... 427
10.1.5 CalcWatP messaging ... 427
10.1.6 CalcWatP I/Os ... 428
10.1.7 CalcWatP block diagram ... 431
10.1.8 Operator control and monitoring ... 431
10.1.8.1 CalcWatP views ... 431
10.1.8.2 CalcWatP standard view ... 432
10.1.8.3 CalcWatP parameter view ... 435
10.1.8.4 CalcWatP preview ... 436
10.1.8.5 CalcWatP block icons ... 437

10.2 HxFct - Calculation of enthalpy, absolute humidity, and saturated humidity according to Mollier ... 439
10.2.1 HxFct description ... 439
10.2.2 HxFct operating modes ... 440
10.2.3 HxFct functions ... 441
10.2.4 HxFct error handling ... 442
10.2.5 HxFct messaging ... 442
10.2.6 HxFct I/Os ... 444

10.2.7 HxFct block diagram 446
10.2.8 Operator control and monitoring 446
10.2.8.1 HxFct views 446
10.2.8.2 HxFct standard view 447
10.2.8.3 HxFct parameter view 448
10.2.8.4 HxFct preview 449
10.2.8.5 HxFct block icons 450

10.3 OptiOT - Optimization of operating time depending on outdoor temperature 452
10.3.1 OptiOT description 452
10.3.2 OptiOT operating modes 454
10.3.3 OptiOT functions 456
10.3.4 OptiOT error handling 458
10.3.5 OptiOP messaging 458
10.3.6 OptiOT I/Os 458
10.3.7 OptiOT block diagram 460
10.3.8 Operator control and monitoring 461
10.3.8.1 OptiOT views 461
10.3.8.2 OptiOT standard view 461
10.3.8.3 OptiOT parameter view 463
10.3.8.4 OptiOT preview 464
10.3.8.5 OptiOT block icons 466

10.4 ConvCF - Conversion of unit of temperature from °C to °F or from °F to °C 467
10.4.1 ConvCF description 467
10.4.2 ConvCF operating modes 467
10.4.3 ConvCF functions 467
10.4.4 ConvCF error handling 468
10.4.5 ConvCF messaging 468
10.4.6 ConvCF I/Os 468
10.4.7 ConvCF block diagram 469
10.4.8 Operator control and monitoring of ConvCF 469

10.5 ConvAbRe - Conversion of humidity from absolute to relative or relative to absolute 470
10.5.1 ConvAbRe description 470
10.5.2 ConvAbRe operating modes 470
10.5.3 ConvAbRe functions 471
10.5.4 ConvAbRe error handling 471
10.5.5 ConvAbRe messaging 471
10.5.6 ConvAbRe I/Os 471
10.5.7 ConvAbRe block diagram 472
10.5.8 Operator control and monitoring of ConvAbRe 472

10.6 CalcTHX - Calculation of setpoints for temperature and humidity controller of a ventilation system controlled by an Hx diagram 473
10.6.1 CalcTHX description 473
10.6.2 CalcTHX operating modes 476
10.6.3 CalcTHX functions 476
10.6.4 CalcTHX error handling 485
10.6.5 CalcTHX messaging 485
10.6.6 CalcTHX I/Os 487
10.6.7 CalcTHX block diagram 491
10.6.8 Operator control and monitoring 492
10.6.8.1 CalcTHX views 492

10.6.8.2 CalcTHX standard view ... 493
10.6.8.3 Limit view of CalcTHX ... 495
10.6.8.4 CalcTHX diagram view ... 496
10.6.8.5 CalcTHX parameter view ... 497
10.6.8.6 CalcTHX preview ... 499
10.6.8.7 CalcTHX block icons ... 501

11 Communication blocks ... 503

11.1 Application areas for the communication blocks ... 503

11.2 Communication between S7-41x and S7-31x / S7-15xx ... 511
11.2.1 S7SndDig - Send digital values ... 511
11.2.1.1 S7SndDig description ... 511
11.2.1.2 S7SndDig operating modes ... 512
11.2.1.3 Function of S7SndDig ... 512
11.2.1.4 S7SndDig error handling ... 512
11.2.1.5 S7SndDig messaging ... 513
11.2.1.6 S7SndDig I/Os ... 513
11.2.1.7 S7SndDig block diagram ... 513
11.2.2 S7SndAna - Send analog values ... 514
11.2.2.1 S7SndAna description ... 514
11.2.2.2 S7SndAna operating modes ... 515
11.2.2.3 Function of S7SndAna ... 515
11.2.2.4 S7SndAna error handling ... 515
11.2.2.5 S7SndAna messaging ... 516
11.2.2.6 S7SndAna I/Os ... 516
11.2.2.7 S7SndAna block diagram ... 516
11.2.3 S7RcvDig - Receive digital values ... 517
11.2.3.1 S7RcvDig description ... 517
11.2.3.2 S7RcvDig operating modes ... 518
11.2.3.3 Function of S7RcvDig ... 518
11.2.3.4 S7RcvDig error handling ... 519
11.2.3.5 S7RcvDig messaging ... 519
11.2.3.6 S7RcvDig I/Os ... 519
11.2.3.7 S7RcvDig block diagram ... 520
11.2.4 S7RcvAna - Receive analog values ... 520
11.2.4.1 S7RcvAna description ... 520
11.2.4.2 S7RcvAna operating modes ... 521
11.2.4.3 Function of S7RcvAna ... 522
11.2.4.4 S7RcvAna error handling ... 522
11.2.4.5 S7RcvAna messaging ... 522
11.2.4.6 S7RcvAna I/Os ... 522
11.2.4.7 S7RcvAna block diagram ... 523

11.3 Communication between S7-41xH and S7-31x / S7-15xx ... 524
11.3.1 SndH_DigVal - Send digital values ... 524
11.3.1.1 SndH_DigVal description ... 524
11.3.1.2 SndH_DigVal operating modes ... 526
11.3.1.3 Function of SndH_DigVal ... 526
11.3.1.4 SndH_DigVal error handling ... 526
11.3.1.5 SndH_DigVal messaging ... 526
11.3.1.6 SndH_DigVal I/Os ... 526
11.3.1.7 SndH_DigVal block diagram ... 527

11.3.2 SndH_AnaVal - Send analog values ... 528
11.3.2.1 SndH_AnaVal description ... 528
11.3.2.2 SndH_AnaVal operating modes ... 530
11.3.2.3 Function of SndH_AnaVal ... 530
11.3.2.4 SndH_AnaVal error handling ... 530
11.3.2.5 SndH_AnaVal messaging ... 530
11.3.2.6 SndH_AnaVal I/Os ... 530
11.3.2.7 SndH_AnaVal block diagram ... 532
11.3.3 RcvH_DigVal - Receive digital values ... 532
11.3.3.1 RcvH_DigVal description ... 532
11.3.3.2 RcvH_DigVal operating modes ... 533
11.3.3.3 Function of RcvH_DigVal ... 533
11.3.3.4 RcvH_DigVal error handling ... 534
11.3.3.5 RcvH_DigVal messaging ... 534
11.3.3.6 RcvH_DigVal I/Os ... 534
11.3.3.7 RcvH_DigVal block diagram ... 535
11.3.4 RcvH_AnaVal - Receive analog values ... 535
11.3.4.1 RcvH_AnaVal description ... 535
11.3.4.2 RcvH_AnaVal operating modes ... 537
11.3.4.3 Function of RcvH_AnaVal ... 537
11.3.4.4 RcvH_AnaVal error handling ... 537
11.3.4.5 RcvH_AnaVal messaging ... 537
11.3.4.6 RcvH_AnaVal I/Os ... 538
11.3.4.7 RcvH_AnaVal block diagram ... 539
11.3.5 S7SndHDig - Send digital values ... 539
11.3.5.1 S7SndHDig description ... 539
11.3.5.2 S7SndHDig operating modes ... 540
11.3.5.3 Function of S7SndHDig ... 540
11.3.5.4 S7SndHDig error handling ... 540
11.3.5.5 S7SndHDig messaging ... 541
11.3.5.6 S7SndHDig I/Os ... 541
11.3.5.7 S7SndHDig block diagram ... 542
11.3.6 S7SndHAna - Send analog values ... 542
11.3.6.1 S7SndHAna description ... 542
11.3.6.2 S7SndHAna operating modes ... 543
11.3.6.3 Function of S7SndHAna ... 543
11.3.6.4 S7SndHAna error handling ... 544
11.3.6.5 S7SndHAna messaging ... 544
11.3.6.6 S7SndHAna I/Os ... 544
11.3.6.7 S7SndHAna block diagram ... 545
11.3.7 S7RcvHDig - Receive digital values ... 545
11.3.7.1 S7RcvHDig description ... 545
11.3.7.2 S7RcvHDig operating modes ... 546
11.3.7.3 Function of S7RcvHDig ... 547
11.3.7.4 S7RcvHDig error handling ... 547
11.3.7.5 S7RcvHDig messaging ... 547
11.3.7.6 S7RcvHDig I/Os ... 547
11.3.7.7 S7RcvHDig block diagram ... 548
11.3.8 S7RcvHAna - Receive analog values ... 549
11.3.8.1 S7RcvHAna description ... 549
11.3.8.2 S7RcvHAna operating modes ... 550
11.3.8.3 Function of S7RcvHAna ... 550

11.3.8.4 S7RcvHAna error handling ... 551
11.3.8.5 S7RcvHAna messaging ... 551
11.3.8.6 S7RcvHAna I/Os ... 551
11.3.8.7 S7RcvHAna block diagram ... 552

11.4 Communication between S7-41xH and S7-41x ... 553
11.4.1 SndH_DigVal, SndH_AnaVal, RcvH_DigVal, RcvH_AnaVal ... 553
11.4.2 ASSendH - H-system communication send block ... 553
11.4.2.1 ASSendH description ... 553
11.4.2.2 ASSendH operating modes ... 555
11.4.2.3 ASSendH functions ... 555
11.4.2.4 ASSendH error handling ... 555
11.4.2.5 ASSendH messaging ... 555
11.4.2.6 ASSendH I/Os ... 557
11.4.2.7 ASSendH block diagram ... 558
11.4.2.8 Operator control and monitoring of ASSendH ... 558
11.4.3 ASRcvH - H-system communication receive block ... 558
11.4.3.1 ASRcvH description ... 558
11.4.3.2 ASRcvH operating modes ... 560
11.4.3.3 ASRcvH functions ... 560
11.4.3.4 ASRcvH error handling ... 560
11.4.3.5 ASRcvH messaging ... 560
11.4.3.6 ASRcvH I/Os ... 561
11.4.3.7 ASRcvH block diagram ... 563
11.4.3.8 Operator control and monitoring of ASRcvH ... 563

11.5 TCP communication blocks ... 564
11.5.1 TRcv_AnaVal - Receive analog values including the quality code ... 564
11.5.1.1 TRcv_AnaVal Description ... 564
11.5.1.2 TRcv_AnaVal Configuration ... 564
11.5.1.3 TRcv_AnaVal Functions ... 565
11.5.1.4 TRcv_AnaVal Error handling ... 567
11.5.1.5 TRcv_AnaVal I/Os ... 567
11.5.2 TRcv_DigVal - Receive boolean values including the quality code ... 568
11.5.2.1 TRcv_DigVal Description ... 568
11.5.2.2 TRcv_DigVal Configuration ... 568
11.5.2.3 TRcv_DigVal Functions ... 568
11.5.2.4 TRcv_DigVal Error handling ... 570
11.5.2.5 TRcv_DigVal I/Os ... 571
11.5.3 TSnd_AnaVal - Send analog values including the quality code ... 571
11.5.3.1 TSnd_AnaVal Description ... 571
11.5.3.2 TSnd_AnaVal Functions ... 572
11.5.3.3 TSnd_AnaVal Error handling ... 573
11.5.3.4 TSnd_AnaVal I/Os ... 573
11.5.4 TSnd_DigVal - Send boolean values including the quality code ... 574
11.5.4.1 TSnd_DigVal Description ... 574
11.5.4.2 TSnd_DigVal Functions ... 574
11.5.4.3 TSnd_DigVal Error handling ... 576
11.5.4.4 TSnd_DigVal IOs ... 576
11.5.5 TCPCon - TCP connection via Open TCP/IP communication ... 576
11.5.5.1 TCPCon Description ... 576
11.5.5.2 TCPCon Functions ... 577
11.5.5.3 TCPCon I/Os ... 578

12 Logic blocks 581

12.1 SelStr - Select one of two STRING values 581
12.1.1 SelStr description 581
12.1.2 SelStr operating modes 581
12.1.3 SelStr functions 581
12.1.4 SelStr error handling 582
12.1.5 SelStr messaging 582
12.1.6 SelStr I/Os 582
12.1.7 SelStr block diagram 582
12.1.8 Operator control and monitoring of SelStr 583

12.2 SelR - Select one of two REAL values 584
12.2.1 SelR description 584
12.2.2 SelR operating modes 584
12.2.3 SelR functions 584
12.2.4 SelR error handling 584
12.2.5 SelR messaging 584
12.2.6 SelR I/Os 585
12.2.7 SelR block diagram 585
12.2.8 Operator control and monitoring of SelR 585

12.3 SelB - Select one of two BOOL values 586
12.3.1 SelB description 586
12.3.2 SelB operating modes 586
12.3.3 SelB functions 586
12.3.4 SelB error handling 586
12.3.5 SelB messaging 586
12.3.6 SelB I/Os 587
12.3.7 SelB block diagram 587
12.3.8 Operator control and monitoring of SelB 587

12.4 SelI - Select one of two INTEGER values 588
12.4.1 SelI description 588
12.4.2 SelI operating modes 588
12.4.3 SelI functions 588
12.4.4 SelI error handling 588
12.4.5 SelI messaging 588
12.4.6 SelI I/Os 589
12.4.7 SelI block diagram 589
12.4.8 Operator control and monitoring of SelI 589

12.5 SelByt - Select one of two BYTE values 590
12.5.1 SelByt description 590
12.5.2 SelByt operating modes 590
12.5.3 SelByt functions 590
12.5.4 SelByt error handling 590
12.5.5 SelByt messaging 590
12.5.6 SelByt I/Os 591
12.5.7 SelByt block diagram 591
12.5.8 Operator control and monitoring of SelByt 591

12.6 SelW - Select one of two WORD values 592
12.6.1 SelW description 592
12.6.2 SelW operating modes 592

12.6.3 SelW functions ... 592
12.6.4 SelW error handling ... 592
12.6.5 SelW messaging ... 592
12.6.6 SelW I/Os ... 593
12.6.7 SelW block diagram ... 593
12.6.8 Operator control and monitoring of SelW ... 593

12.7 SelDW - Select one of two DOUBLE WORD values ... 594
12.7.1 SelDW description ... 594
12.7.2 SelDW operating modes ... 594
12.7.3 SelDW functions ... 594
12.7.4 SelDW error handling ... 594
12.7.5 SelDW messaging ... 594
12.7.6 SelDW I/Os ... 595
12.7.7 SelDW block diagram ... 595
12.7.8 Operator control and monitoring of SelDW ... 595

12.8 SelDI - Select one of two DOUBLE INTEGER values ... 596
12.8.1 SelDI description ... 596
12.8.2 SelDI operating modes ... 596
12.8.3 SelDI functions ... 596
12.8.4 SelDI error handling ... 596
12.8.5 SelDI messaging ... 596
12.8.6 SelDI I/Os ... 597
12.8.7 SelDI block diagram ... 597
12.8.8 Operator control and monitoring of SelDI ... 597

12.9 SelC - Select one of two CHAR values ... 598
12.9.1 SelC description ... 598
12.9.2 SelC operating modes ... 598
12.9.3 SelC functions ... 598
12.9.4 SelC error handling ... 598
12.9.5 SelC messaging ... 598
12.9.6 SelC I/Os ... 599
12.9.7 SelC block diagram ... 599
12.9.8 Operator control and monitoring of SelC ... 599

12.10 VoteAnL - Monitoring block for up to 8 analog process values ... 600
12.10.1 VoteAnL description ... 600
12.10.2 VoteAnL operating modes ... 606
12.10.3 VoteAnL functions ... 606
12.10.4 VoteAnL error handling ... 617
12.10.5 VoteAnL messaging ... 618
12.10.6 VoteAnL I/Os ... 620
12.10.7 VoteAnL block diagram ... 624
12.10.8 Operator control and monitoring ... 624
12.10.8.1 VoteAnL views ... 624
12.10.8.2 VoteAnL standard view ... 625
12.10.8.3 Limit view of VoteAnL ... 632
12.10.8.4 VoteAnL parameter view ... 633
12.10.8.5 VoteAnL preview ... 635
12.10.8.6 VoteAnL block icon ... 636

12.11 VoteDiL - Monitoring block for up to 8 digital process values ... 638
12.11.1 VoteDiL description ... 638

12.11.2 VoteDiL operating modes ... 644
12.11.3 VoteDiL functions ... 644
12.11.4 VoteDiL error handling ... 653
12.11.5 VoteDiL messaging ... 653
12.11.6 VoteDiL I/Os ... 654
12.11.7 VoteDiL block diagram ... 658
12.11.8 Operator control and monitoring ... 659
12.11.8.1 VoteDiL views ... 659
12.11.8.2 VoteDiL standard view ... 660
12.11.8.3 VoteDiL parameter view ... 665
12.11.8.4 VoteDiL preview ... 667
12.11.8.5 VoteDiL block icon ... 668

13 Mathematical blocks ... 671

13.1 AccuS - Accumulates measured values with a specific factor ... 671
13.1.1 AccuS description ... 671
13.1.2 AccuS operating modes ... 672
13.1.3 AccuS functions ... 672
13.1.4 AccuS error handling ... 674
13.1.5 AccuS messaging ... 674
13.1.6 AccuS I/Os ... 674
13.1.7 AccuS block diagram ... 676
13.1.8 Operator control and monitoring ... 676
13.1.8.1 AccuS views ... 676
13.1.8.2 AccuS standard view ... 677
13.1.8.3 AccuS parameter view ... 678
13.1.8.4 AccuS preview ... 679
13.1.8.5 AccuS block icons ... 679

14 Motor and valve blocks ... 681

14.1 VlvDiv - Diverter valve ... 681
14.1.1 VlvDiv description ... 681
14.1.2 VlvDiv operating modes ... 685
14.1.3 VlvDiv functions ... 687
14.1.4 VlvDiv error handling ... 694
14.1.5 VlvDiv messaging ... 695
14.1.6 VlvDiv I/Os ... 697
14.1.7 VlvDiv block diagram ... 698
14.1.8 Operator control and monitoring ... 699
14.1.8.1 VlvDiv views ... 699
14.1.8.2 VlvDiv standard view ... 699
14.1.8.3 VlvDiv parameter view ... 702
14.1.8.4 VlvDiv preview ... 704
14.1.8.5 VlvDiv block icons ... 708

14.2 VlvDsL - Double seat valve ... 712
14.2.1 VlvDsL description ... 712
14.2.2 VlvDsL operating modes ... 715
14.2.3 VlvDsL functions ... 716
14.2.4 VlvDsL error handling ... 721
14.2.5 VlvDsL messaging ... 723
14.2.6 VlvDsL I/Os ... 724
14.2.7 VlvDsL block diagram ... 727

14.2.8 Operator control and monitoring ... 728
14.2.8.1 VlvDsL views ... 728
14.2.8.2 VlvDsL standard view ... 729
14.2.8.3 VlvDsL seat lifting view ... 732
14.2.8.4 VlvDsL preview ... 734
14.2.8.5 VlvDsL block icons ... 736

15 Panel blocks ... 739

15.1 PCalcWatP - Panel communication block for CalcWatP ... 739
15.1.1 PCalcWatP description ... 739
15.1.2 PCalcWatP operating modes ... 740
15.1.3 PCalcWatP functions ... 741
15.1.4 PCalcWatP error handling ... 741
15.1.5 PCalcWatP messaging ... 741
15.1.6 PCalcWatP I/Os ... 741
15.1.7 PCalcWatP block diagram ... 743
15.1.8 Operator control and monitoring ... 743
15.1.8.1 PCalcWatP WinCC views ... 743
15.1.8.2 PCalcWatP WinCC Comfort views ... 743
15.1.8.3 PCalcWatP WinCC Comfort ... 743
15.1.8.4 PCalcWatP WinCC Comfort block icons ... 743

15.2 PHxFct - Panel communication block for HxFct ... 744
15.2.1 PHxFct description ... 744
15.2.2 PHxFct operating modes ... 745
15.2.3 PHxFct functions ... 745
15.2.4 PHxFct error handling ... 746
15.2.5 PHxFct messaging ... 746
15.2.6 PHxFct I/Os ... 746
15.2.7 PHxFct block diagram ... 747
15.2.8 Operator control and monitoring ... 748
15.2.8.1 PHxFct WinCC views ... 748
15.2.8.2 PHxFct WinCC Comfort views ... 748
15.2.8.3 PHxFct WinCC Comfort standard view ... 748
15.2.8.4 PHxFct WinCC Comfort block icons ... 748

15.3 PMotL - Panel communication block for MotL ... 749
15.3.1 PMotL description ... 749
15.3.2 PMotL operating modes ... 751
15.3.3 PMotL functions ... 751
15.3.4 PMotL error handling ... 751
15.3.5 PMotL messaging ... 751
15.3.6 PMotL I/Os ... 751
15.3.7 PMotL block diagram ... 753
15.3.8 Operator control and monitoring ... 753
15.3.8.1 PMotL WinCC views ... 753
15.3.8.2 PMotL WinCC Comfort views ... 753
15.3.8.3 PMotL WinCC Comfort standard view ... 753
15.3.8.4 PMotL WinCC Comfort block icons ... 754

15.4 PMotRevL - Panel communication block for MotRevL ... 755
15.4.1 PMotRevL description ... 755
15.4.2 PMotRevL operating modes ... 757
15.4.3 PMotRevL functions ... 757

15.4.4 PMotRevL error handling ... 757
15.4.5 PMotRevL messaging ... 757
15.4.6 PMotRevL I/Os ... 757
15.4.7 PMotRevL block diagram ... 759
15.4.8 Operator control and monitoring ... 759
15.4.8.1 PMotRevL WinCC views ... 759
15.4.8.2 PMotRevL WinCC Comfort views ... 759
15.4.8.3 PMotRevL WinCC Comfort standard view ... 759
15.4.8.4 PMotRevL WinCC Comfort block icons ... 759

15.5 PMotSpdCL - Panel communication block for MotSpdCL ... 760
15.5.1 PMotSpdCL description ... 760
15.5.2 PMotSpdCL operating modes ... 762
15.5.3 PMotSpdCL functions ... 762
15.5.4 PMotSpdCL error handling ... 762
15.5.5 PMotSpdCL messaging ... 762
15.5.6 PMotSpdCL I/Os ... 762
15.5.7 PMotSpdCL block diagram ... 764
15.5.8 Operator control and monitoring ... 764
15.5.8.1 PMotSpdCL WinCC views ... 764
15.5.8.2 PMotSpdCL WinCC Comfort views ... 764
15.5.8.3 PMotSpdCL WinCC Comfort standard view ... 764
15.5.8.4 PMotSpdCL WinCC Comfort block icons ... 765

15.6 PMotSpdL - Panel communication block for MotSpdL ... 766
15.6.1 PMotSpdL description ... 766
15.6.2 PMotSpdL operating modes ... 768
15.6.3 PMotSpdL functions ... 768
15.6.4 PMotSpdL error handling ... 768
15.6.5 PMotSpdL messaging ... 768
15.6.6 PMotSpdL I/Os ... 768
15.6.7 PMotSpdL block diagram ... 770
15.6.8 Operator control and monitoring ... 770
15.6.8.1 PMotSpdL WinCC views ... 770
15.6.8.2 PMotSpdLWinCC Comfort views ... 770
15.6.8.3 PMotSpdLWinCC Comfort standard view ... 770
15.6.8.4 PMotSpdLWinCC Comfort block icons ... 771

15.7 PMonAnL - Panel communication block for MonAnL ... 772
15.7.1 PMonAnL description ... 772
15.7.2 PMonAnL operating modes ... 773
15.7.3 PMonAnL functions ... 774
15.7.4 PMonAnL error handling ... 774
15.7.5 PMonAnL messaging ... 774
15.7.6 PMonAnL I/Os ... 774
15.7.7 PMonAnL block diagram ... 776
15.7.8 Operator control and monitoring ... 776
15.7.8.1 PMonAnL WinCC views ... 776
15.7.8.2 PMonAnL WinCC Comfort views ... 776
15.7.8.3 PMonAnL WinCC Comfort standard view ... 776
15.7.8.4 PMonAnL WinCC Comfort block icons ... 776

15.8 PMonAn08 - Panel communication block for MonAn08 ... 777
15.8.1 PMonAn08 description ... 777

15.8.2 PMonAn08 operating modes ... 778
15.8.3 PMonAn08 functions ... 778
15.8.4 PMonAn08 error handling ... 779
15.8.5 PMonAn08 messaging ... 779
15.8.6 PMonAn08 I/Os ... 779
15.8.7 PMonAn08 block diagram ... 780
15.8.8 Operator control and monitoring ... 781
15.8.8.1 PMonAn08 WinCC views ... 781
15.8.8.2 PMonAn08 WinCC Comfort views ... 781
15.8.8.3 PMonAn08 WinCC Comfort standard view ... 781
15.8.8.4 PMonAn08 WinCC Comfort block icons ... 781

15.9 PMonAnDi - Panel communication block for MonAnDi ... 782
15.9.1 PMonAnDi description ... 782
15.9.2 PMonAnDi operating modes ... 783
15.9.3 PMonAnDi functions ... 784
15.9.4 PMonAnDi error handling ... 784
15.9.5 PMonAnDi messaging ... 784
15.9.6 PMonAnDi I/Os ... 784
15.9.7 PMonAnDi block diagram ... 786
15.9.8 Operator control and monitoring ... 786
15.9.8.1 PMonAnDi WinCC views ... 786
15.9.8.2 PMonAnDi WinCC Comfort views ... 786
15.9.8.3 PMonAnDi WinCC Comfort standard view ... 786
15.9.8.4 PMonAnDi WinCC Comfort block icons ... 787

15.10 PMonDiL - Panel communication block for MonDiL ... 788
15.10.1 PMonDiL description ... 788
15.10.2 PMonDiL operating modes ... 789
15.10.3 PMonDiL functions ... 789
15.10.4 PMonDiL error handling ... 790
15.10.5 PMonDiL messaging ... 790
15.10.6 PMonDiL I/Os ... 790
15.10.7 PMonDiL block diagram ... 791
15.10.8 Operator control and monitoring ... 791
15.10.8.1 PMonDiL WinCC views ... 791
15.10.8.2 PMonDiL WinCC Comfort views ... 792
15.10.8.3 PMonDiL WinCC Comfort standard view ... 792
15.10.8.4 PMonDiL WinCC Comfort block icons ... 792

15.11 PMonDi08 - Panel communication block for MonDi08 ... 793
15.11.1 PMonDi08 description ... 793
15.11.2 PMonDi08 operating modes ... 794
15.11.3 PMonDi08 functions ... 794
15.11.4 PMonDi08 error handling ... 794
15.11.5 PMonDi08 messaging ... 795
15.11.6 PMonDi08 I/Os ... 795
15.11.7 PMonDi08 block diagram ... 796
15.11.8 Operator control and monitoring ... 796
15.11.8.1 PMonDi08 WinCC views ... 796
15.11.8.2 PMonDi08 WinCC Comfort views ... 797
15.11.8.3 PMonDi08 WinCC Comfort standard view ... 797
15.11.8.4 PMonDi08 WinCC Comfort block icons ... 797

15.12 POpAnL - Panel communication block for OpAnL ... 798
15.12.1 POpAnL description ... 798
15.12.2 POpAnL operating modes ... 799
15.12.3 POpAnL functions ... 799
15.12.4 POpAnL error handling ... 799
15.12.5 POpAnL messaging ... 800
15.12.6 POpAnL I/Os ... 800
15.12.7 POpAnL block diagram ... 801
15.12.8 Operator control and monitoring ... 801
15.12.8.1 POpAnL WinCC views ... 801
15.12.8.2 POpAnL WinCC Comfort views ... 802
15.12.8.3 POpAnL WinCC Comfort standard view ... 802
15.12.8.4 POpAnL WinCC Comfort block icons ... 802

15.13 POpD - Panel communication block for OpDi01 / OpTrig ... 803
15.13.1 POpD description ... 803
15.13.2 POpD operating modes ... 804
15.13.3 POpD functions ... 804
15.13.4 POpD error handling ... 804
15.13.5 POpD messaging ... 804
15.13.6 POpD I/Os ... 805
15.13.7 POpD block diagram ... 805
15.13.8 Operator control and monitoring ... 806
15.13.8.1 POpD WinCC views ... 806
15.13.8.2 Pulse command (OpTrig) views from POpD WinCC Comfort ... 806
15.13.8.3 Block icons - Pulse command (OpTrig) from POpD WinCC Comfort ... 806
15.13.8.4 Standard view - Pulse command (OpTrig) from POpD WinCC Comfort ... 806
15.13.8.5 Static command (OpD) views from POpD WinCC Comfort ... 806
15.13.8.6 Standard view - Static command (OpD) from POpD WinCC Comfort ... 807
15.13.8.7 Block icons- Static command (OpD) from POpD WinCC Comfort ... 807

15.14 PPIDL - Panel communication block for PIDL ... 808
15.14.1 PPIDL description ... 808
15.14.2 PPIDL operating modes ... 810
15.14.3 PPIDL functions ... 810
15.14.4 PPIDL error handling ... 810
15.14.5 PPIDL messaging ... 810
15.14.6 PPIDL I/Os ... 811
15.14.7 PPIDL block diagram ... 812
15.14.8 Operator control and monitoring ... 813
15.14.8.1 PPIDL WinCC views ... 813
15.14.8.2 PPIDL WinCC Comfort views ... 813
15.14.8.3 PPIDL WinCC Comfort standard view ... 813
15.14.8.4 PPIDL WinCC Comfort block icons ... 813

15.15 PVlvL - Panel communication block for VlvL ... 814
15.15.1 PVlvL description ... 814
15.15.2 PVlvL operating modes ... 816
15.15.3 PVlvL functions ... 816
15.15.4 PVlvL error handling ... 816
15.15.5 PVlvL messaging ... 816
15.15.6 PVlvL I/Os ... 816
15.15.7 PVlvL block diagram ... 818

15.15.8 Operator control and monitoring ... 818
15.15.8.1 PVlvL WinCC views ... 818
15.15.8.2 PVlvL WinCC Comfort views ... 818
15.15.8.3 PVlvL WinCC Comfort standard view ... 818
15.15.8.4 PVlvL WinCC Comfort block icons ... 819

15.16 PVlvMotL - Panel communication block for VlvMotL ... 820
15.16.1 PVlvMotL description ... 820
15.16.2 PVlvMotL operating modes ... 822
15.16.3 PVlvMotL functions ... 822
15.16.4 PVlvMotL error handling ... 822
15.16.5 PVlvMotL messaging ... 822
15.16.6 PVlvMotL I/Os ... 822
15.16.7 PVlvMotL block diagram ... 824
15.16.8 Operator control and monitoring ... 824
15.16.8.1 PVlvMotL WinCC views ... 824
15.16.8.2 PVlvMotL WinCC Comfort views ... 824
15.16.8.3 PVlvMotL WinCC Comfort standard view ... 824
15.16.8.4 PVlvMotL WinCC Comfort block icons ... 825

15.17 PVlvPosL - Communication block for VlvPosL ... 826
15.17.1 PVlvPosL description ... 826
15.17.2 Operating modes of PVlvPosL ... 828
15.17.3 PVlvPosL functions ... 828
15.17.4 PVlvPosL error handling ... 828
15.17.5 PVlvPosL messaging ... 828
15.17.6 PVlvPosL I/Os ... 829
15.17.7 PVlvPosL block diagram ... 830
15.17.8 Operator control and monitoring ... 830
15.17.8.1 PVlvPosL WinCC views ... 830
15.17.8.2 PVlvPosL WinCC Comfort views ... 831
15.17.8.3 PVlvPosL WinCC Comfort standard view ... 831
15.17.8.4 PVlvPosL WinCC Comfort block icons ... 831

15.18 PUsrM - Panel communication block for UsrM ... 832
15.18.1 PUsrM description ... 832
15.18.2 PUsrM operating modes ... 834
15.18.3 PUsrM functions ... 834
15.18.4 PUsrM error handling ... 834
15.18.5 PUsrM messaging ... 834
15.18.6 PUsrM I/Os ... 834
15.18.7 PUsrM ... 836
15.18.8 Operator control and monitoring ... 836
15.18.8.1 PUsrM WinCC views ... 836
15.18.8.2 PUsrM WinCC Comfort views ... 836
15.18.8.3 PUsrM WinCC Comfort standard view ... 836
15.18.8.4 PUsrM WinCC Comfort block icons ... 837

16 Controller blocks ... 839

16.1 Higher-level controlling of the unit / EQM level ... 839
16.1.1 Structure data connections EQM <-> Monitoring function ... 839

16.2 PolygonExt - Polygon line with up to 8 interpolation points ... 841
16.2.1 PolygonExt description ... 841

16.2.2 PolygonExt operating modes ... 842
16.2.3 PolygonExt functions ... 843
16.2.4 PolygonExt error handling ... 844
16.2.5 PolygonExt messaging ... 844
16.2.6 PolygonExt I/Os ... 846
16.2.7 PolygonExt block diagram ... 847
16.2.8 Operator control and monitoring ... 847
16.2.8.1 PolygonExt views ... 847
16.2.8.2 PolygonExt standard view ... 848
16.2.8.3 PolygonExt parameter view ... 850
16.2.8.4 PolygonExt preview ... 851
16.2.8.5 PolygonExt block icon ... 852

16.3 PolyCurve - Setpoint generator with variable number of curve points ... 853
16.3.1 PolyCurve description ... 853
16.3.2 PolyCurve operating modes ... 856
16.3.3 PolyCurve functions ... 856
16.3.4 PolyCurve error handling ... 860
16.3.5 PolyCurve messaging ... 860
16.3.6 PolyCurve I/Os ... 861
16.3.7 PolyCurve block diagram ... 862
16.3.8 Operator control and monitoring ... 862
16.3.8.1 PolyCurve views ... 862
16.3.8.2 PolyCurve standard view ... 863
16.3.8.3 PolyCurve parameter view ... 865
16.3.8.4 PolyCurve preview ... 866
16.3.8.5 PolyCurve block icons ... 867

16.4 SPCurve - Setpoint scheduler with a variable number of setpoints ... 868
16.4.1 SPCurve description ... 868
16.4.2 SPCurve operating modes ... 871
16.4.3 SPCurve functions ... 872
16.4.4 SPCurve error handling ... 877
16.4.5 SPCurve messaging ... 877
16.4.6 SPCurve I/Os ... 878
16.4.7 SPCurve block diagram ... 881
16.4.8 Operator control and monitoring ... 882
16.4.8.1 SPCurve views ... 882
16.4.8.2 SPCurve standard view ... 883
16.4.8.3 SPCurve parameter view ... 886
16.4.8.4 SPCurve preview ... 888
16.4.8.5 SPCurve block icons ... 889

16.5 TimeSwitch - Time switch with 8 switch points ... 892
16.5.1 TimeSwitch description ... 892
16.5.2 TimeSwitch operating modes ... 894
16.5.3 TimeSwitch functions ... 894
16.5.4 TimeSwitch error handling ... 896
16.5.5 TimeSwitch messaging ... 897
16.5.6 TimeSwitch I/Os ... 898
16.5.7 TimeSwitch block diagram ... 899
16.5.8 Operator control and monitoring ... 899
16.5.8.1 TimeSwitch views ... 899
16.5.8.2 TimeSwitch standard view ... 900

16.5.8.3 TimeSwitch parameter view ... 902
16.5.8.4 TimeSwitch preview ... 903
16.5.8.5 TimeSwitch block icons ... 904

16.6 SplRng4 signal splitter ... 907
16.6.1 SplRng4 description ... 907
16.6.2 SplRng4 operating modes ... 908
16.6.3 SplRng4 functions ... 909
16.6.4 SplRng4 error handling ... 912
16.6.5 SplRng4 messaging ... 913
16.6.6 SplRng4 I/Os ... 913
16.6.7 SplRng4 block diagram ... 914

17 System blocks ... 915

17.1 ASTimeBCD - Output the CPU time in BCD format ... 915
17.1.1 ASTimeBCD description ... 915
17.1.2 ASTimeBCD operating modes ... 915
17.1.3 ASTimeBCD functions ... 915
17.1.4 ASTimeBCD error handling ... 915
17.1.5 ASTimeBCD messaging ... 916
17.1.6 ASTimeBCD I/Os ... 916
17.1.7 ASTimeBCD block diagram ... 916
17.1.8 Operator control and monitoring of ASTimeBCD ... 916

18 Monitoring blocks ... 917

18.1 MonAn08 - Measured value monitoring with 8 limits ... 917
18.1.1 MonAn08 description ... 917
18.1.2 MonAn08 operating modes ... 920
18.1.3 MonAn08 functions ... 921
18.1.4 MonAn08 error handling ... 926
18.1.5 MonAn08 messaging ... 927
18.1.6 MonAn08 I/Os ... 929
18.1.7 MonAn08 block diagram ... 931
18.1.8 Operator control and monitoring ... 931
18.1.8.1 MonAn08 views ... 931
18.1.8.2 MonAn08 standard view ... 932
18.1.8.3 Limit view of MonAn08 ... 935
18.1.8.4 MonAn08 parameter view ... 938
18.1.8.5 MonAn08 preview ... 939
18.1.8.6 MonAn08 block icon ... 940

18.2 MonAnDi - Analog measured value monitoring with binary limits ... 942
18.2.1 MonAnDi description ... 942
18.2.2 MonAnDi operating modes ... 945
18.2.3 MonAnDi functions ... 945
18.2.4 MonAnDi error handling ... 949
18.2.5 MonAnDi messaging ... 949
18.2.6 MonAnDi I/Os ... 951
18.2.7 MonAnDi block diagram ... 953
18.2.8 Operator control and monitoring ... 953
18.2.8.1 MonAnDi views ... 953
18.2.8.2 MonAnDi standard view ... 954
18.2.8.3 Limit view of MonAnDi ... 958

18.2.8.4 MonAnDi parameter view ... 959
18.2.8.5 MonAnDi preview ... 961
18.2.8.6 MonAnDi block icon ... 962

18.3 MonDin – Monitoring of a digital input ... 963
18.3.1 Description of 'MonDin' ... 963
18.3.2 Operating modes of 'MonDin' ... 964
18.3.3 Functions of 'MonDin' ... 964
18.3.4 Connections of 'MonDin' ... 967
18.3.5 Operator control and monitoring ... 969
18.3.5.1 Symbol ... 969
18.3.5.2 Faceplate ... 969

18.4 MonMof – Combined monitoring function ... 973
18.4.1 Description of 'MonMof' ... 973
18.4.2 Operating modes of 'MonMof' ... 974
18.4.3 Functions of Mon'Mof' ... 974
18.4.4 Messages of 'MonMof' ... 980
18.4.5 Connections of 'MonMof' ... 982
18.4.6 Operator control and monitoring ... 984
18.4.6.1 Symbol ... 984
18.4.6.2 Faceplate ... 985

19 Interlock blocks ... 991

19.1 ILck4x3L - Interlock with 8 input signals and 4x3 logic ... 991
19.1.1 ILck4x3L description ... 991
19.1.2 ILck4x3L operating modes ... 997
19.1.3 ILck4x3L functions ... 997
19.1.4 ILck4x3L error handling ... 1004
19.1.5 ILck4x3L messaging ... 1004
19.1.6 ILck4x3L I/Os ... 1004
19.1.7 ILck4x3L block diagram ... 1007
19.1.8 Operator control and monitoring ... 1007
19.1.8.1 ILck4x3L views ... 1007
19.1.8.2 ILck4x3L standard view ... 1008
19.1.8.3 ILck4x3L extended standard view ... 1010
19.1.8.4 ILck4x3L parameter view ... 1015
19.1.8.5 ILck4x3L preview ... 1016
19.1.8.6 ILck4x3L block icon ... 1017

19.2 ILck8x4L - Interlock with 16 input signals and 8x4 logic ... 1019
19.2.1 ILck8x4L description ... 1019
19.2.2 ILck8x4L operating modes ... 1029
19.2.3 ILck8x4L functions ... 1029
19.2.4 ILck8x4L error handling ... 1036
19.2.5 ILck8x4L messaging ... 1037
19.2.6 ILck8x4L I/Os ... 1037
19.2.7 ILck8x4L block diagram ... 1040
19.2.8 Operator control and monitoring ... 1040
19.2.8.1 ILck8x4L views ... 1040
19.2.8.2 Intlk8x4 standard view ... 1041
19.2.8.3 ILck8x4L extended standard view ... 1044
19.2.8.4 ILck8x4L parameter view ... 1048
19.2.8.5 ILck8x4L preview ... 1050

19.2.8.6 ILck8x4L block icon 1051

20 Maintenance blocks 1053

20.1 SimAn - Simulation block for analog values 1053
20.1.1 SimAn description 1053
20.1.2 SimAn operating modes 1054
20.1.3 SimAn functions 1054
20.1.4 SimAn error handling 1055
20.1.5 SimAn messaging 1055
20.1.6 SimAn I/Os 1055
20.1.7 SimAn block diagram 1056
20.1.8 Operator control and monitoring 1056
20.1.8.1 SimAn views 1056
20.1.8.2 SimAn standard view 1057
20.1.8.3 SimAn preview 1058
20.1.8.4 SimAn block icons 1058

20.2 SimDi - Simulation block for digital values 1059
20.2.1 SimDi description 1059
20.2.2 SimDi operating modes 1060
20.2.3 SimDi functions 1060
20.2.4 SimDi error handling 1060
20.2.5 SimDi messaging 1061
20.2.6 SimDi I/Os 1061
20.2.7 SimDi block diagram 1061
20.2.8 Operator control and monitoring 1062
20.2.8.1 SimDi views 1062
20.2.8.2 SimDi standard view 1062
20.2.8.3 SimDi preview 1063
20.2.8.4 SimDi block icons 1064

21 S88 structure interfaces and batch interface 1065

21.1 UnitIf – Unit interface block 1065
21.1.1 Description of 'UnitIf' 1065
21.1.2 Functions of 'UnitIf' 1070
21.1.3 Connections of 'UnitIf' 1073
21.1.4 Operator control and monitoring 1075
21.1.4.1 Symbol 1075
21.1.4.2 Faceplate 1076

21.2 EqmIf – Equipment module interface block 1082
21.2.1 Description of 'EqmIf' 1082
21.2.2 Functions of 'EqmIf' 1084
21.2.3 Connections of 'EqmIf' 1088
21.2.4 Operator control and monitoring 1089
21.2.4.1 Symbol 1089
21.2.4.2 Faceplate 1090

21.3 BatchIf – SIMATIC BATCH interface block 1096
21.3.1 Description of 'BatchIf' 1096
21.3.2 Operating modes of 'BatchIf' 1097
21.3.3 Functions of 'BatchIf' 1097
21.3.4 Messaging of 'BatchIf' 1103
21.3.5 Connections of 'BatchIf' 1104

21.3.6 Operator control and monitoring ... 1107
21.3.6.1 Symbol ... 1107
21.3.6.2 Faceplate ... 1108

22 Technological blocks ... 1113

22.1 Qing - Queue Transfer control block ... 1113
22.1.1 Operating modes of 'Qing' ... 1116
22.1.2 Functions of 'Qing' ... 1116
22.1.3 Error correction for 'Qing' ... 1121
22.1.4 Connections of 'Qing' ... 1121
22.1.5 Operator control and monitoring ... 1125
22.1.5.1 Symbol ... 1125
22.1.5.2 Faceplate ... 1126

22.2 Xfer – Transfer routes – Control block ... 1134
22.2.1 Description of 'Xfer' ... 1134
22.2.2 Operating modes of 'Xfer' ... 1136
22.2.3 Functions of 'Xfer' ... 1137
22.2.4 Error correction for 'Xfer' ... 1141
22.2.5 Connections of 'Xfer' ... 1141
22.2.6 Configuration of the function and mask definitions ... 1148
22.2.6.1 Configuration dialog "Project overview screen" ... 1148
22.2.6.2 Configuration dialog "Editing window" ... 1150
22.2.7 Operator control and monitoring ... 1152
22.2.7.1 Symbol ... 1152
22.2.7.2 Faceplate ... 1153

22.3 Tank – Tank status block ... 1163
22.3.1 Description of 'Tank' ... 1163
22.3.2 Operating modes of 'Tank' ... 1164
22.3.3 Functions of 'Tank' ... 1165
22.3.4 Messages of 'Tank' ... 1172
22.3.5 Connections of 'Tank' ... 1173
22.3.6 Operator control and monitoring ... 1178
22.3.6.1 Symbol ... 1178
22.3.6.2 Faceplate ... 1180

22.4 TcCtrl – Cooling zones control block ... 1187
22.4.1 Description of 'TcCtrl' ... 1187
22.4.2 Operating modes of 'TcCtrl' ... 1189
22.4.3 Functions of 'TcCtrl' ... 1190
22.4.4 Error correction for 'TcCtrl' ... 1196
22.4.5 Messages of 'TcCtrl' ... 1197
22.4.6 Connections of 'TcCtrl' ... 1198
22.4.7 Operator control and monitoring ... 1203
22.4.7.1 Symbol ... 1203
22.4.7.2 Faceplate ... 1204

22.5 TcTsSel – Temperature sensor selection block ... 1213
22.5.1 Description of 'TcTsSel' ... 1213
22.5.2 Functions of 'TcTsSel' ... 1214
22.5.3 Connections of 'TcTsSel' ... 1216

22.6 TcCzSel – Cooling zone selection block ... 1217
22.6.1 Description of 'TcCzSel' ... 1217

22.6.2 Functions of 'TcCzSel' ... 1217
22.6.3 Connections of 'TcCzSel' ... 1218

22.7 TcCfg – Cooling zones configuration block ... 1220
22.7.1 Operating principle of 'TcCfg' ... 1220
22.7.2 Connections of 'TcCfg' ... 1220

23 Timers ... 1221

23.1 TimerXP – Signal delay / incremental encoder ... 1221
23.1.1 Description of 'TimerXP' ... 1221
23.1.2 Functions of 'TimerXP' ... 1221
23.1.3 Connections of 'TimerXP' ... 1223

24 Appendix ... 1225

24.1 General block I/Os ... 1225

Index ... 1237

Security information

1

Siemens provides products and solutions with industrial security 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 security 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 security measures that may be implemented, please visit
https://www.siemens.com/industrialsecurity.

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 Security RSS Feed under
https://www.siemens.com/industrialsecurity.

Basics

2

2.1 About this document

This manual describes the functions of the Industry Library for PCS 7 V9.0.1 blocks, hereafter referred to as IL.

The IL is an expansion of the *PCS 7 Advanced Process Library V9.0.3* block library, hereafter referred to as *APL*.

This manual describes the blocks included in the Industry Library and the functional enhancements compared to the APL.

The emphasis of this manual is on:

- "Multi-panel operation" operating strategy including coupler blocks for panel integration with WinCC Comfort
- Function of the APC coupling blocks
- Description of the included industry-specific blocks

All functions that are included in the APL and are used in the IL are described in the "APL Function manual". You can find the APL manual on your PCS 7 Engineering System and at Industry Online Support:

PCS 7 Advanced Process Library V9.0 Manual (https://support.industry.siemens.com/cs/ww/en/view/109744306)

2.2 Overview of the function blocks

The table below offers a list of the function blocks contained in the IL PCS 7 with their essential functions and features.

FB/FC name	FB/FC number	Function	OCM
SimAn	FB1100	Simulation block for analog values	J
SimDi	FB1101	Simulation block for digital values	J
MonAn08	FB1102	Measured value monitoring for 8 limits	J
VlvDsL	FB1103	Actuation of a double seat valve	J
MonAnDi	FB1104	Measured value monitoring for 4 analog and binary limits	J
SelStr	FB1106	STRING selector	N
PMotL	FB1107	Operator panel interface for single-stage motor (MotL)	J
PMotRevL	FB1108	Operator panel interface for reversing motor with 2 directions of rotation (MotRevL)	J
PMotSpdL	FB1109	Operator panel interface for frequency-controlled motor (MotSpdCL)	J
PMotSpdCL	FB1110	Operator panel interface for two-stage motor (MotSpdL)	J
AccuS	FB1111	Accumulated measured value with specific heat	J
UsrM	FB1112	Selection of operation level out of 8 levels	J
Aggr16	FB1113	Switchover for max. 16 units	J
Aggr08	FB1114	Switchover for max. 8 units	J
ASSendH	FB1116	Coordinate the sending of message frames between a redundant and non-redundant automation station	N
ASRcvH	FB1117	Coordinate the reception of message frames between a redundant and non-redundant automation station	N
PMonAn08	FB1118	Operator panel interface for measured value monitoring for 8 limits (MonAn08)	J
PMonAnDi	FB1119	Operator panel interface for measured value monitoring for 8 limits (MonAnDi)	J
PMonAnL	FB1120	Operator panel interface for measured value monitoring (MonAnL)	J
PMonDiL	FB1121	Operator panel interface for monitoring of a binary process tag (MonDiL)	J
PMonDi08	FB1122	Operator panel interface for monitoring of the 8 binary process tags (MonDi08L)	J
POpAnL	FB1123	Operator panel interface for manipulation of an analog value (OpAnL)	J
VlvDiv	FB1127	Control of a route switch with up to 8 positions	J
VoteAnL	FB1129	Monitoring block for up to 8 process values	J
POpD	FB1131	Operator panel interface for manipulation of a binary value (OpD)	J
PUsrM	FB1132	Operator panel interface for selecting one of 8 control levels (UsrM)	J
PVlvL	FB1136	Operator panel interface for a valve (VlvL)	J
PVlvPosL	FB1137	Operator panel interface for frequency-controlled motor (VlvPosL)	J

FB/FC name	FB/FC number	Function	OCM
PPIDL	FB1138	Operator panel interface for two-stage motor (PIDConL, PIDConR, PIDStepL)	J
SPCurve	FB1141	Setpoint scheduler with a variable number of setpoints	J
TimeSwitch	FB1142	Time switch with 8 switch points	J
PVlvMotL	FB1143	Operator panel interface for motor valve (VlvMotL)	J
CalcWatP	FB1144	Calculation of thermal power and emitted energy	J
PCalcWatP	FB1146	Operator panel interface for calculation of thermal power and emitted energy	J
HxFct	FB1147	Calculation of enthalpy, absolute humidity, and humidity at saturation	J
PHxFct	FB1149	Operator panel interface for calculation of enthalpy, absolute humidity, and humidity at saturation (HxFct)	J
OptiOT	FB1150	Optimization of utilization period dependent on the outdoor temperature	J
SelFp	FB1154	Jump distributor for expanding the faceplate jumps to SelFp1 and SelFp2	N
PolygonExt	FB1155	Polygon line with up to 8 interpolation points	J
PolyCurve	FB1156	Setpoint scheduler with a variable number of interpolation points	J
CalcTHX	FB1159	Calculation of the setpoints for temperature and humidity controllers of a ventilation system, which is guided according to the Hx chart	J
SndH_DigVal	FB1170	Send up to 128 digital structures from an S7-400H to an S7-300 / S7-1500 / S7-400 or from an S7-400 to an S7-400H	N
RcvH_DigVal	FB1171	Reception of up to 128 digital structures from an S7-300 / S7-1500 / S7-400 through an S7-400H or from an S7-400H through an S7-400	N
SndH_AnaVal	FB1172	Send up to 32 analog structures from an S7-400H to an S7-300 / S7-1500 / S7-400 or from an S7-400 to an S7-400H	N
RcvH_AnaVal	FB1173	Reception of up to 32 digital structures from an S7-300 / S7-1500 / S7-400 through an S7-400H or from an S7-400H through an S7-400	N
SplRng4	FB1175	Signal splitter for up to 4 actuators	N
APC_Supervisor	FB1180	Central block for managing the connection of higher-level controllers.	J
APC_OpSP	FB1181	Interface to the process value for the coupling higher-level controllers	J
APC_MV	FB1182	Interface to the process value for coupling lower-level controllers	N
APC_MpList	FB1184	Creation of process tag tables for the connection of higher-level controllers.	N
ParaCtrl	FB1185	Block for handling smaller numbers of parameters and parameter sets	J
ParaMem	FB1186	Block for handling smaller numbers of parameters and parameter sets	J
PeakMon	FB1190	Acquisition and forecast of consumption peaks at the plant infeed	J

FB/FC name	FB/FC number	Function	OCM
LdMgmt8	FB1191	Load management block for coordination of up to 8 consumers	J
PulseCon	FB1192	Pulse converter to a counter value for energy management	N
CalcPwrL	FB1193	Acquisition and calculation of energy consumption values	J
ApMotL	FB 1200	Motor	
ApMotRevL	FB 1201	Reversing motor	
ApMotSpdCL	FB 1202	Speed-controlled motor	
ApPIDConL	FB 1203	Extended PID controller	
ApVlvDS	FB 1205	Double-seat valve	
ApVlvL	FB 1206	Valve	
ApVlvAnL	FB 1207	Analog actuator	
ApCmGen	FB 1210	Generic adapter block	
CmDec	FB 1213	CM info decoding	
CmEnc	FB 1214	CM info encoding	
MidStr	FB 1215	Converter from material DINT to string	
StrMid	FB 1216	Converter from string to material DINT	
PolygonExt	FB 1217	Polyline	
TimerXP	FB 1218	TimerXP	
VlvDS	FB 1220	Double-seat valve	
BatchIf	FB 1222	SIMATIC BATCH interface	
UnitIf	FB 1223	Unit (UNIT) interface	
EqmIf	FB 1224	Equipment module (EQM) interface	
FxCtrl	FB 1226	Extended function control block	
FxShare	FB 1227	Shared equipment module (Shared EQM)	
MonDin	FB 1228	Digital input without alarming	
MonMof	FB 1229	Value monitoring with alarming and acknowledgment	
MonAn08	FB 1230	Analog input with alarming and acknowledgment	
OpAn16	FB 1231	Operation of up to 16 analog values with operating limits	
OpEnum	FB 1232	Operation of a value through enumerations	
SelString	FB 1239	1 of 2 STRING values	
PuPa	FB 1240	Pulse pause	
Qing	FB 1241	Queue transfer	
Tank	FB 1242	Tank status block	
TcCtrl	FB 1243	Cooling zone control block	
TcCzSel	FB 1244	Cooling zone selector	
TcTsSel	FB 1245	Temperature sensor selector	
Xfer	FB 1246	"Flying tank change" control block	
FxDef16	FB 1247	Function definitions max. 16	
FxDef32	FB 1248	Function definitions max. 32	
FxDef64	FB 1249	Function definitions max. 64	
FxSigBin	FB 1250	Binary signal adapter	
FxSigInt	FB 1251	Integer adapter	
TcCfg	FB 1252	Cooling configuration parameter block	

FB/FC name	FB/FC number	Function	OCM
GrabMonTm	FC 1200	- can only be used internally -	
GetIDB	FC 1201	- can only be used internally -	
BitInt	FC 1202	Converter from bit to integer	
IntBit	FC 1203	Converter from integer to bit pattern	
SelBool	FC 1204	1 of 2 Boolean values	
SelByte	FC 1205	1 of 2 BYTE values	
SelInt	FC 1206	1 of 2 INT values	
SelDint	FC 1207	1 of 2 DINT values	
SelWord	FC 1208	1 of 2 WORD values	
SelDWord	FC 1209	1 of 2 DWORD values	
SelReal	FC 1210	1 of 2 REAL values	
OpA16Dec	FC 1211	Decoder for OpA16	
HexToElemDef	FC 1212	- can only be used internally -	
HexToDW	FC 1213	- can only be used internally -	
HexElementsToS-truct	FC 1214	- can only be used internally -	
ExtCopy	FC 1215	- can only be used internally -	
OpA16Dec	FC 1216		
SelR	FC601	REAL selector	N
SelB	FC602	BOOL selector	N
SelI	FC603	INTEGER selector	N
SelByt	FC604	BYTE selector	N
SelW	FC605	WORD selector	N
SelDW	FC606	DWORD selector	N
SelDI	FC607	DOUBLE INTEGER selector	N
SelC	FC608	CHAR selector	N
ConvCF	FC609	Conversion of unit of temperature from °C to °F or °F to °C	N
ConvAbRe	FC610	Conversion of humidity from absolute to relative or relative to absolute humidity	N
ASTimeBCD	FC611	Output of CPU time in BCD format	N

2.3 Specialties

2.3.1 Dependency on APL

The Industry Library for PCS 7 represents an extension of the *PCS 7 Advanced Process Library* block library. Since interdependencies exist between the blocks of the IL and the APL, the blocks of the two libraries have to be copied into one library for the IL to be used.

To ensure conformity with PCS 7, it is advisable to use the master data library of the PCS 7 multiproject.

2.3.2 Setting of the standard server

Setting of the standard server

The Industry Library uses enumerations to display user-specific texts. To ensure that the texts of the enumerations are displayed correctly on a client, a standard server has to be specified for text libraries.

2.3.3 Functions of the blocks

Forming and outputting the signal status for interlock blocks

The IL supports the function "Forming and outputting the signal status for interlock blocks" of the APL.

You can find additional information on this function in the section "Forming and outputting the signal status for interlock blocks" of the APL manual.

As a supplement to the interlock blocks of the APL, the blocks of the IL also support the logical XOR operation (logic = 2):

General rules

- Logical XOR operation (logic = 2):
  The output value always has the signal status with the highest priority of all connected input signals, as any of the inputs can influence the output.

2.3.4 Functions of the faceplates

APL Operator Trend Control (AOTC)

The IL supports the function APL Operator Trend Control (AOTC) of the APL.

You can find additional information on function in the section "APL Operator Trend Control (AOTC)" of the APL manual

In addition to the list in the APL manual, you can add digital values from the following IL blocks:

Family	Blocks
Operate	Aggr16, Aggr08
Drives	VlvDiv, VlvDsL
Control	TimeSwitch
HVAC	OptiOT

Specialties with operator control and monitoring in WinCC

The following functions are not supported by the faceplates of the Industry Library listed below:

	Faceplate selection via process tag list	Faceplate selection via screen composition
APC Supervisor	X	X
ParaCtrl	X	X

2.3.5 Format Specification for analog value display

The format specification for the analog value display can parameterize centrally in the block icon at the attribute configurations>AnalogValueFormat1...AnalogValueFormat4. If the attribute is empty, the format specification will be taken directly from the faceplate PDL file.

The parametrization in the block icon takes effect in the standard view, and the block icon. The following table shows the analog values for which format specification can be parameterize in the block icon:

Blocks	AnalogValueFormat1 Standard View, Preview, Ramp View	AnalogValueFormat2 Standard View, Preview, Ramp View	AnalogValueFormat3 Standard View, Preview, Ramp View	AnalogValueFormat4 Standard View, Preview, Ramp View
APC_OpSP	`PV` values, `DeadBand` values	`SP` values		
PolyCurve	`PV` values			
PolygonExt	Input values	Output values		
SPCurve	`PV` values	`SP` values		
CalcPwrL	`PV` values	`SP` values	Counter1	Counter20
LdMgmt8	`PV` values, `SP` values			
PeakMon	`PV` values	`SP` values		
CalcTHx	Temperature values	Humidity values	Hum Supply Air	

Blocks	**AnalogValueFormat1** **Standard View, Preview, Ramp View**	**AnalogValueFormat2** **Standard View, Preview, Ramp View**	**AnalogValueFormat3** **Standard View, Preview, Ramp View**	**AnalogValueFormat4** **Standard View, Preview, Ramp View**
CalcWatP	Power values	Energy Values	Temperature	VolFlow Rate
HxFct	Enthalpy	AbsHum, SatHum	TempWetBulb	Atmosphere temp, Realtive Humidity
OptiOT	`PV` values	`SP` values		
VoteAnL	`PV` values			
VoteDiL	`PV` values			
SimAn	out values			
AccuS	Counter1 value	Counter2 value		
MonAn08	`PV` values			
MonAnDi	`PV` values, `DeadBand` values			
ParaCtrl	`PV` values			

Note

If the analog value display shows *****, the display field is too small for the parameterized format specification. In this case, adapt the format specification AnalogValueFormatx in the block icon.

2.4 Multiple control room concept

2.4.1 Description of the concept

There may be some PCS 7 applications in which the plants or units have to be operated from several locations.

To avoid inconsistencies caused by the operation from different locations and control who is responsible for operation, the IL offers a concept for operation of multiple control rooms to meet this requirement.

Determining the operator authorization

The multiple control room concept supplied with the Industrial Library is based on the existing APL function "Local operator authorization".

The local operator authorization is an upstream operator control permission. It determines independent of user management and authorizations, if a block may be operated from an operating station (a specified place). If local operator authorization is missing, operation of a block instance is usually blocked. Otherwise, when local operator authorization is allowed, the operator control permission is normally determined through user management and the block-specific release.

Local operator authorization can be set for each specific instance for each operator station with its own variable management; in other words, block instances can be enabled or disabled for use on an operator station independently of one another.

The concept integrated with the UsrM block consists of hierarchical operation and monitoring of control rooms and panels for up to 8 levels.

Each of these 8 levels can be assigned the control rights over a Faceplate. In addition, an operating level can be assigned a permission with highest priority (KeySwitch function) by means of a technological I/O.

The names of the operating levels are defined for specific users by means of an enumeration and are visualized on the OS.

Function and interaction of components

The `UsrM` is responsible for central management of the active operating level. The active operating level is specified at this block by the faceplate or the interconnection. By interconnecting the output `UsrM.Out` with the inputs `<Technological block>.OpSt_In` and `<Panel block>.SwitchPerm`, the active operating level is transferred to the block instance.

The local operator authorization is checked on the OS by comparing the value of the internal variable `@APLOpStation` with the value at the `OpSt_In` I/O of the block instance.

The local operator authorization on the panel is checked by comparing the values at the I/Os `SwitchPerm` and `PanelPerm`.

If these values are identical, the station has the local operator authorization.

2.4.2 Configuring operating levels on the operator station (OS)

To assign an operating level to an operator station (OS) you must perform the following steps:

1. Add the bicoded identification of the operating level to the start value of the internal variable `@APLOpStation`.
   (The variable of the data type "unsigned 32-bit value" is automatically created using the OS project editor and is located in the Split Screen Manager group). For further information, refer to "PCS 7 Advanced Process Library V7.1 SP5" manual, chapter OpStations / Engineering)
2. Interconnect the `<Technological block>.OpSt_In` input with the `UsrM.Out` output of the user manager block.
3. Set the `Feature.bit24` at the technological block.
   If the internal variable `@APLOpStation` matches the value at the `<technological block>.OpSt_In` input, the operator station can be operated.

2.4.3 Configuring operating levels on the operator panel (OP)

You need to complete the following steps to assign a relevant operating level to an operator panel (OP):

1. Configure `<Panel block>.PanelPerm` input with the appropriate operating level on the operator panel.
2. Interconnect the `<Panel block>.SwitchPerm` input with the `UsrM.Out` output of the user manager block.

If the `<Panel block>.PanelPerm` and `<Panel block>.SwitchPerm` parameters match, the operator panel can be operated.

Note

The enumeration (default: `IL_OpShort`) connected to `<panel block>.SwitchPerm_Out` is used to display the current operator permission in the symbol of the panel block.

2.4.4 Table with possible values of the operating levels

	UsrM. Switch-Perm	**UserM. Out**	**<Panel block>. Switch-Perm**	**<Panel block>. PanelPerm**	**<Panel block>. Switch-Perm_Out**	**WinCCExplorer internal variable Split Screen Manager @APLOpStation**
Parameter	IN	OUT	IN	IN	OUT	"Signless 32-bit value" (bit-coded)
Data type	INT	DWORD (bit-coded)	DWORD (bit-coded)	DWORD (bit-coded)	DWORD (bit-coded)	
Enumeration	-	IL_OpLong	IL_OpLong	IL_OpLong	IL_OpShort	
Local control panel	1	1	1	1	1	1
Control panel OS	2	2	2	2	2	2
Operator panel 1	3	4	4	4	4	4
Operator panel 2	4	8	8	8	8	8
Operator panel 3	5	16	16	16	16	16
Operator panel 4	6	32	32	32	32	32
Operator panel 5	7	64	64	64	64	64
Operator panel 6	8	128	128	128	128	128

2.5 Panel integration

2.5.1 Description of the concept

In many plants there is the requirement to operate and visualize PCS 7 units close to the machine in addition to the control room.

For this requirement, the IL provides a a concept for panel integration with TIA Portal in PCS 7. The concept offers a uniform operating and display concept and connects the alarm systems of the operator station and operator panel.

Core functionality

The PCS 7 Industry Library block library includes matching interface blocks for some of the APL technological blocks. The interface blocks are used to implement the data exchange between the operator panel and operator station and to standardize the alarm logging. The interface block also manages the operator authorization for the panel.

The interface blocks come with a library of preconfigured block icons and faceplates for visualization on the operator panel.

The functions of the IL on the panel correspond to the standards of the SIMATIC PCS 7 Advanced Process Library (APL) in their design and philosophy.

Operating philosophy

All operator stations have three standard operating levels 5, 6 and 1100.

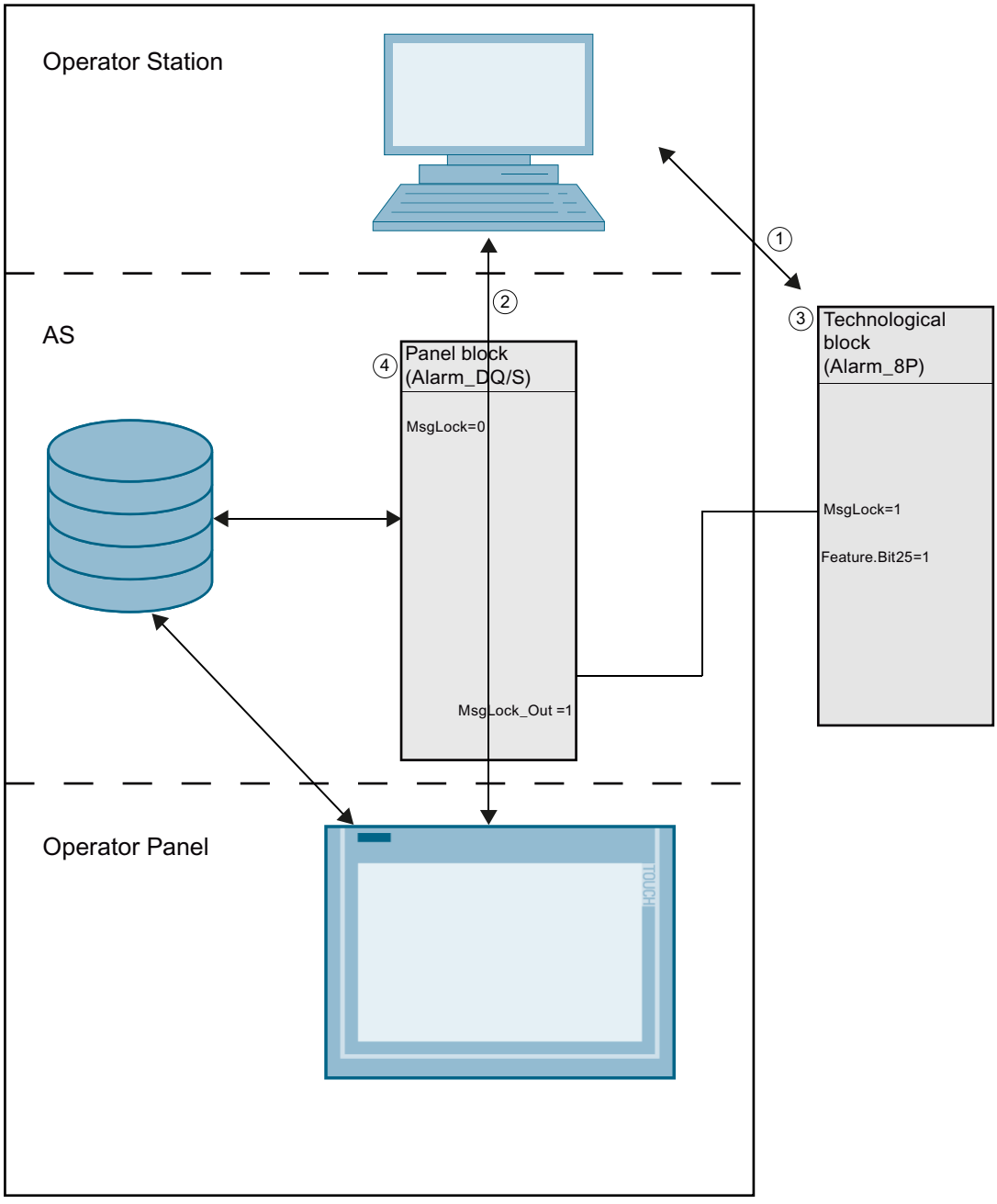
An operator with operating level 5 "Process operations" can perform all switching operations, which means all technological devices can be activated/deactivated or the operating mode can be switched from manual to automatic and vice versa.

An operator with operating level 6 "Higher-level process operations" can change parameters provided to the operator by a faceplate.

An operator with operating level 1100 "Highest-level process operations" can simulate process values and release the process tag for maintenance.

The operating philosophy implemented on the Operator Panel corresponds to operating level 5. Thus, it is not possible to change parameters on the OP or to set the equipment to "Out of service".

Message behavior



To map the message behavior of the operator station on the operator panel, it is necessary to implement the message function with Alarm_DQ **(2)** instead of the usual message procedure (via Alarm_8P **(1)**).

The panel blocks already include the preconfigured call of the message function Alarm_DQ **(4)**. This means standard messages of the technological block are already included. The message generation via ALARM_DQ can be deactivated with the MsgLock parameter at the panel block.

To prevent duplicate messages on the OS, the messages of the APL block must be suppressed when the message generation via panel block is active. To do so, set the input MsgLock=0 at

panel block **(4)**. The output structure MsgLock_Out =1 is set at the same time; it must be interconnected with the MsgLock input structure of the technological block **(3)**.

If control system messages (CSF) and external messages (ExtMsgx) are to be suppressed as well, `Feature.Bit25` ("1 = Suppress all messages if MsgLock = 1") must also be set at APL block **(3)**.

All messages of the technological block should always exist on the panel block.

In the message view of the operator panel, the message classes "S7 messages", "S7 controller control system messages" and "S7 warning" are displayed. Note that messages of the message class "S7 Process message" or "S7 Tolerance" are not displayed by the system broken down by instances in the message view of the operator panel. Messages of these message classes must be displayed by a separate message view in the operator panel project.

A filter is required to display messages broken down by instance in the message view of the operator panel. The message is given a unique value to assign the messages on the panel to an instance by means of filtering. This value consists of the instance DB number of the panel function block and the value configured at the "AS_Nr" input of the panel block.

The filter value is written to the "Op_MsgFilter" I/O and transferred to the panel with the message as message auxiliary value when the ALARM_DQ is generated. The message texts are then filtered in the message view of the faceplate according to the value at the "Op_MsgFilter" I/O of the associated panel block.

This means a value that **is unique throughout the project** must be configured at the input "AS_Nr" for each AS in the project. Values between 1 and 999 are permitted at the "AS_Nr" input.

2.5.2 Interface to the operator panel

Function and interaction of components

If you wish to control and monitor the process from an operator panel as well as an OS (operator station), you must use the operator panel blocks (Pxxx). These blocks form the OCM interface with the operator panel.

For example, if you want to control and monitor a "MOTOR" via the OS and an operator panel, it is necessary to implement the following blocks:

- Technological function block "MotL" from the Advanced Process Library (APL) to control the "MOTOR" device
- Operator panel (OP) interface block "PMotL" of the Industry Library for coupling/connecting the technological block "MotL" with the panel (WinCC Comfort)

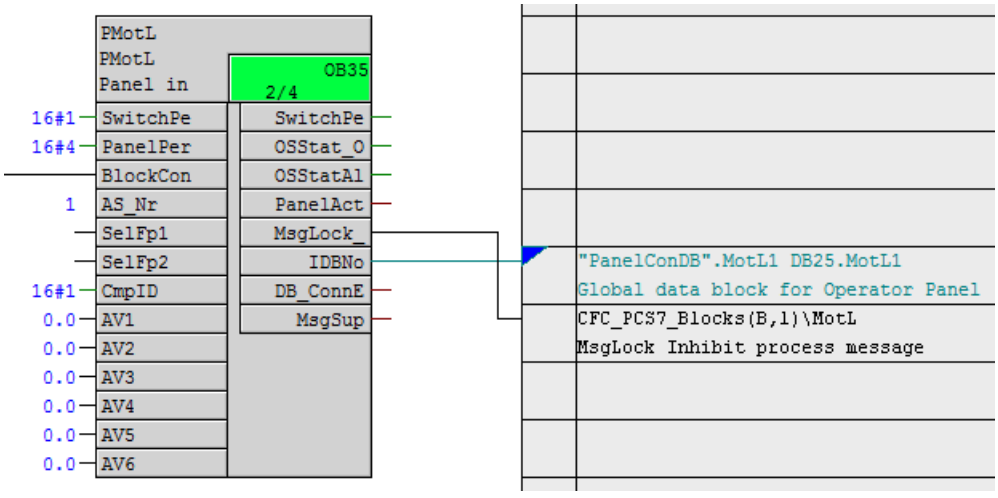
The technological block and panel block communicate using an ANY pointer connection. In other words, communication can be established simply by connecting the "BlockConnector" input of the panel block to any output (apart from "ENO") of the technological block. All necessary values are read and written using this connection.

If an error is detected in this connection (for example, if the BlockConnector input is connected to a technological block that is not associated with the panel block), the DB_ConnErr output of the panel block is set to 1.

Instance data block as interface

The instance data blocks of the operator panel form the OCM interface with the operator panel. They have a defined "structure" depending on the device.

Create a data block which serves as "communication DB" in the S7 program. You can use DB 25 from the library as template. The instance data block numbers of the used panel blocks are stored as integer in this block. Connect the "IDBNo" output of the panel interface block with the DB parameter.



Note

You only need to use one DB for each controller that includes all IDB number entries of the panel blocks used.

2.5.3 Overview of IL Comfort types

The Industry Library for WinCC Comfort provides you with a preconfigured faceplate collection. The block icons and faceplates are preconfigured with the necessary variables and functions. In addition to the graphic objects, the utilized variables and objects are also stored in the library. All you have to do is adapt them to the project-specific conditions. Please note: It is necessary to have a separate variable container for every faceplate and every block icon.

You can display the process values of several technological blocks of the same type in one display window. By clicking the corresponding block icon, you define which process values are to be displayed in the display window.

Type	IL S7	IL PCS 7	Comment
CalcWatP	S7PCalcWatP	PCalcWatP	Faceplate for panel communication block for calculating the thermal power and emitted energy
HxFct	S7PHxFct	PHxFct	Faceplate for panel communication block for calculating enthalpy, absolute humidity, and saturated humidity according to Molier

Type	IL S7	IL PCS 7	Comment
MonAnL	S7PMonAnL	PMonAnL	Faceplate for panel communication block for measured value monitoring
MonAnL_APMK	-	PMonAnL	Faceplate with activated "User-configurable message classes" function for panel communication block for measured value monitoring
MonAnGrad	S7PMonAn-Grad	-	Faceplate for panel communication block for measured value monitoring with gradients
MonAn08	S7PMonAn08	PMonAn08	Faceplate for panel communication block for measured value monitoring with 8 limits
MoAn08_APMK	-	PMonAn08	Faceplate with activated "User-configurable message classes" function for panel communication block for measured value monitoring with 8 limits
MonAnDi	S7PMonAnDi	PMonAnDi	Faceplate for panel communication block for measured value monitoring with 4 analog and binary limits
MonAn-Di_APMK	-	PMonAnDi	Faceplate with activated "User-configurable message classes" function for panel communication block for measured value monitoring with 4 analog and binary limits
MonDiL	S7PMonDiL	PMonDiL	Faceplate for panel communication block for monitoring a binary process tag
MonDiL_APMK	-	PMonDiL	Faceplate with activated "User-configurable message classes" function for panel communication block for monitoring a binary process tag
MonDi08	S7PMonDi08	PMonDi08	Faceplate for panel communication block for measured value monitoring of 8 binary process tags
Mon-Di08_APMK	-	PMonDi08	Faceplate with activated "User-configurable message classes" function for panel communication block for measured value monitoring of 8 binary process tags
Mot	S7PMot	PMot	Faceplate for panel communication block for a single-stage motor
MotRevL	S7PMotRevL	PMotRevL	Faceplate for panel communication block for a motor with 2 directions of rotation
MotSpdL	S7PMotSpdL	PMotSpdL	Faceplate for panel communication block for a two-stage motor
MotSpdCL	S7PMotSpdCL	PMotSpdCL	Faceplate for panel communication block for a frequency-controlled motor
MotSpdCL_APMK	-	PMotSpdCL	Faceplate with activated "User-configurable message classes" function for panel communication block for a frequency-controlled motor
MV3P	S7PMV3P	-	Faceplate for panel communication block for 3-point final controlling element
OpA	S7POpA	POpA	Faceplate for panel communication block of an analog measured value
OpD	S7POpD	POpD	Faceplate for panel communication block of a binary measured value
PID	S7PPID	PPID	Faceplate for panel communication block for controller

Type	IL S7	IL PCS 7	Comment
PID_APMK	-	PPID	Faceplate with activated "User-configurable message classes" function for panel communication block for a controller
UsrM	S7PUsrM	PUsrM	Faceplate for panel communication block for a user administration block
VlvL	S7PVlvL	PVlvL	Faceplate for panel communication block for a valve
VlvMotL	S7PVlvMotL	PVlvMotL	Faceplate for panel communication block for a motor valve
VlvPosL	-	PVlvPosL	Faceplate for panel communication block for a valve positioner
VlvPosL_APMK	-	PVlvPosL	Faceplate with activated "User-configurable message classes" function for panel communication block for a valve positioner

Note

You can enable the parameter view by using the EnAux input of the panel block. If EnAux = 0, the view cannot be opened and the button is not displayed.

2.5.4 Operator control and monitoring in WinCC Comfort

2.5.4.1 Configuration of the panel interface

Configuring the panel interface

The panel interface is configured in WinCC Comfort. A description of the procedure and an application example is available on the Online Support pages in entry *"Integration of Comfort Panels, Operator Panels and S7-300 Package Units in SIMATIC PCS 7 with PCS 7 Industry Library"*

Access options

You can access the documentation as follows:

- On the Internet
  - http://www.siemens.com/automation/service (http://www.siemens.com/automation/service)

2.5.4.2 General view of the panel blocks

Panel block views

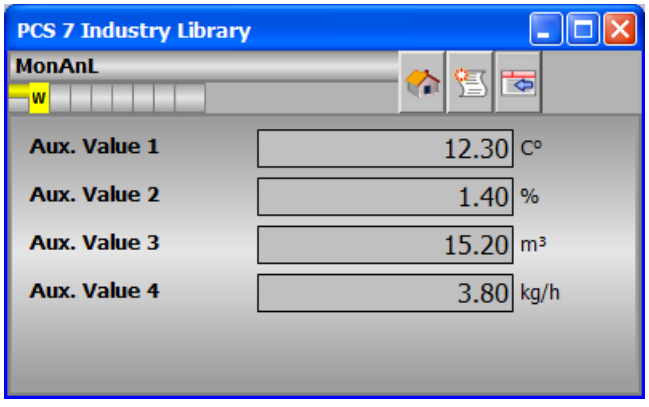
The blocks provide the following views:

- Standard view
- Message view (some blocks)
- Associated value view

Non-specific block views

These views are identical for all blocks which offer the respective view.

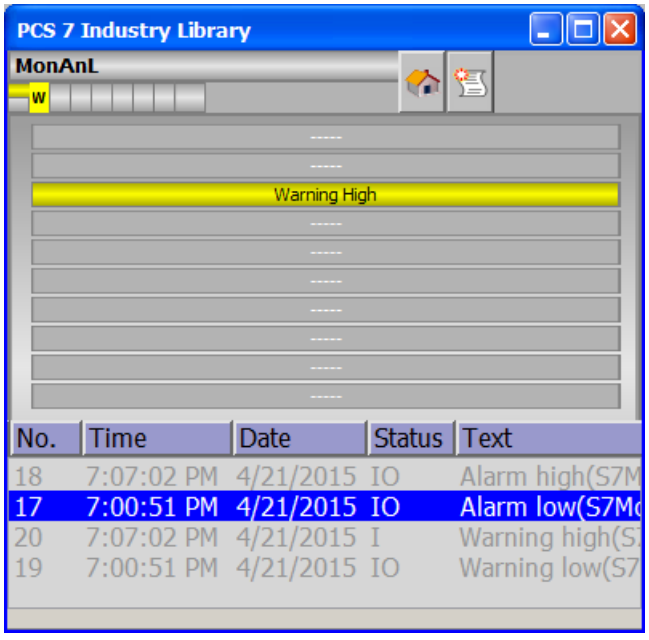
Associated value view



Note

You can enable the associated value view using the EN_AUX input of the panel block. If EN_AUX = 0, the view cannot be opened.

Message view



Note

Colors for message classes in the message window

The representation of the messages in the message window is made centrally in the TIA Portal in the message settings of the message classes of the Panel.

So that the message texts can be configured in a readable color based on the message color, each faceplate has the Message -> Alarm_TextColor attribute. Project-specific templates with an appropriate message text color can thus be created.

The color is set to gray by default so that the text is readable for warnings, which are usually yellow in PCS 7, as well as for black faults.

NOTICE
Associated values in the message texts of the panel blocks The accompanying values at the end of the message texts (for example: (PUsrM: #@3X%3u@)) must not be deleted or changed. These are required to correctly display the messages in the message view on the panel. If these texts are adapted, this can cause messages not to be displayed in the message view on the panel.

2.5.4.3 Messages with activated function "User-configurable message classes"

Implementation

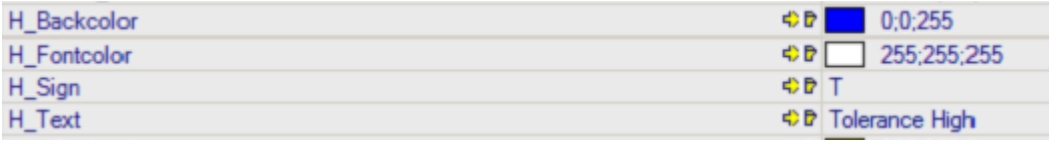
The functionality "User-configurable message classes" (UCMC) is only entered for the panel connection components of the IL for PCS 7 . Only those blocks which have a message of the type HHH, HH, H, L, LL or LLL are affected by the UCMC. This means only the panel connection components of the following blocks are affected:

- MonAnL
- MonAn08
- MonAnDi
- MonDiL
- MonDi08
- MotSpdCL
- PIDL

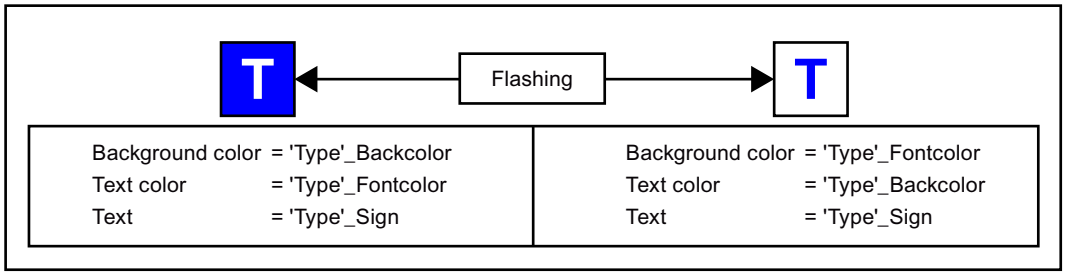
For these 7 types, extended visualization types (icon and faceplate) are available in the "APMK" folder of the WinCC Comfort library.

This extension gives users the option to adapt the colors and initials or texts of the message view in the group view as well as the discrete alarm view to the panel for specific instances to the settings of the UCMC.

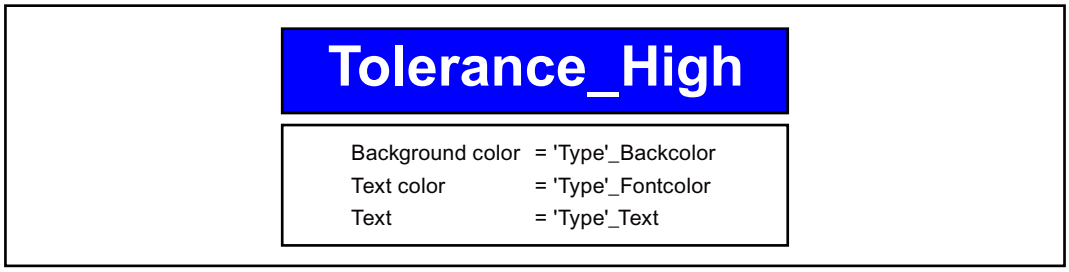
The attributes 'Type'_Backcolor, 'Type'_Fontcolor and 'Type'_Sign of the group view components are displayed for editing under Properties / General / Messages for the icon and faceplate for the corresponding message views of the type HHH, HH, H, L, LL, LLL. You also have the attribute 'Type'_Text for faceplates which represents the display text of the discrete alarm view of the message view.



Attribute assignment in the visualization of the group view (icon + faceplate):



Attribute assignment in the visualization of the discrete alarm view (faceplate):

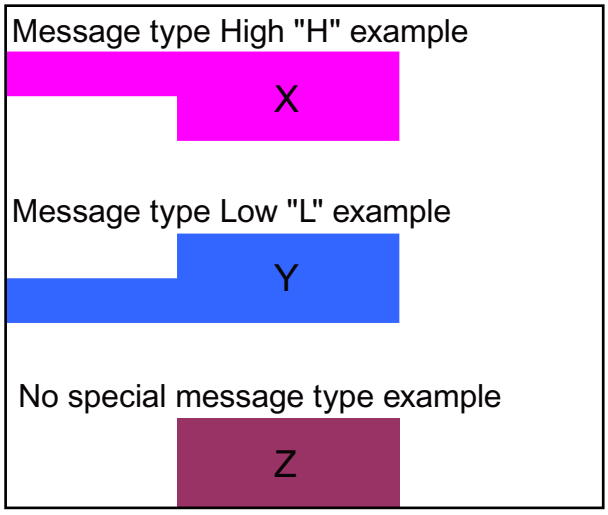


The following messages can be edited as follows for the corresponding blocks (the message assignment is displayed in parentheses):

Prio	MonAnL	MonAn08	MonAnDi	MonDiL	MonDi08	MotSpdCL	PIDL
1	PV AH (H)	Limit 1	PV AH (H)	Out	Signal 1	Rbk WH (H)	PV AH (H)
2	PosGradH (H)	Limit 2	PV AL (L)	Flutter limits	Signal 2	Rbk WL (L)	ER AH (H)
3	NegGradH (H)	Limit 3	PV WH (H)		Signal 3		PV AL (L)
4	PV AL (L)	Limit 4	PV WL (L)		Signal 4		ER AL (L)
5	AbsGrad (L)	Limit 5			Signal 5		PV WH (H)
6	PV WH (H)	Limit 6			Signal 6		Rbk WH (H)
7	PV WL (L)	Limit 7			Signal 7		PV WL (L)
8	PV TH (H)	Limit 8			Signal 8		Rbk WL (L)
9	PV TL (L)						PV TH (H)
10							PV TL (L)

The order shown in the list corresponds to the priority of the display in the group view. In contrast to visualization in the WinCC OS, you cannot change the priority of the WinCC Comfort group display. Even the assignment to the alarm box is permanently set and cannot be changed.

Depending on the message assignment High (H) or Low (L), a bar in the configured background color is additionally displayed to the left of the group view.



2.5.5 Operator Control and Monitoring in WinCC

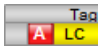
2.5.5.1 Block icon of the panel blocks

Block icon for panel block

The representation for the block icon of the panel block is the same for all panel blocks.

The icon displays the active operating level and displays the group view when the Alarm_DQ message procedure is active.

Block icon from the template @PCS7TypicalsIL_PCS_7.PDL:



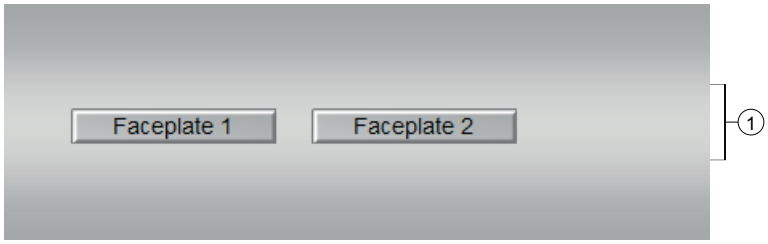
2.5.5.2 Panel block views

Panel block views

The panel blocks provide the following views:

- Standard view
- Message view (some blocks)

Standard view



(1) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

For more information, refer to the "Calling further faceplates" chapter in the APL online help.

Message view

The message view corresponds to the alarm view of the APL blocks. A description is available in the APL manual in the section "Basics of APL -> Functions of the faceplates -> Alarm view".

2.6 File dialog

2.6.1 File dialog function

The IL file dialog is used by different blocks of the Industry Library. It is used to select files for import/export or to delete files. Folders and subfolders are visible for this purpose in the IL file dialog which were explicitly released during configuration.

Folders that are not released do not contain contents/files. Only files with the file extension .csv are visible in released folders. During data export, the IL file dialog adds the .csv extension to each entered file name.

2.6.2 Configuration of file dialog

After installation of the Industry Library, you will initially not be able to select any files in the IL file dialog and not see any drives or folders.

The paths must be explicitly released. To do so, you have to enter the corresponding path in the text variable *@IL_FileDialog_PermittedFolders* in this format:

Drive letter1:\Folder1.1\Folder1.2\<Separator>Drive letter2:\Folder2.1\Folder2.2\<Separator>Drive letter3:\Folder3.1\Folder3.2

<crlf> (= line break) or <;> (= semicolon) are permitted as separators. The configuration string does not have to be concluded with a separator.

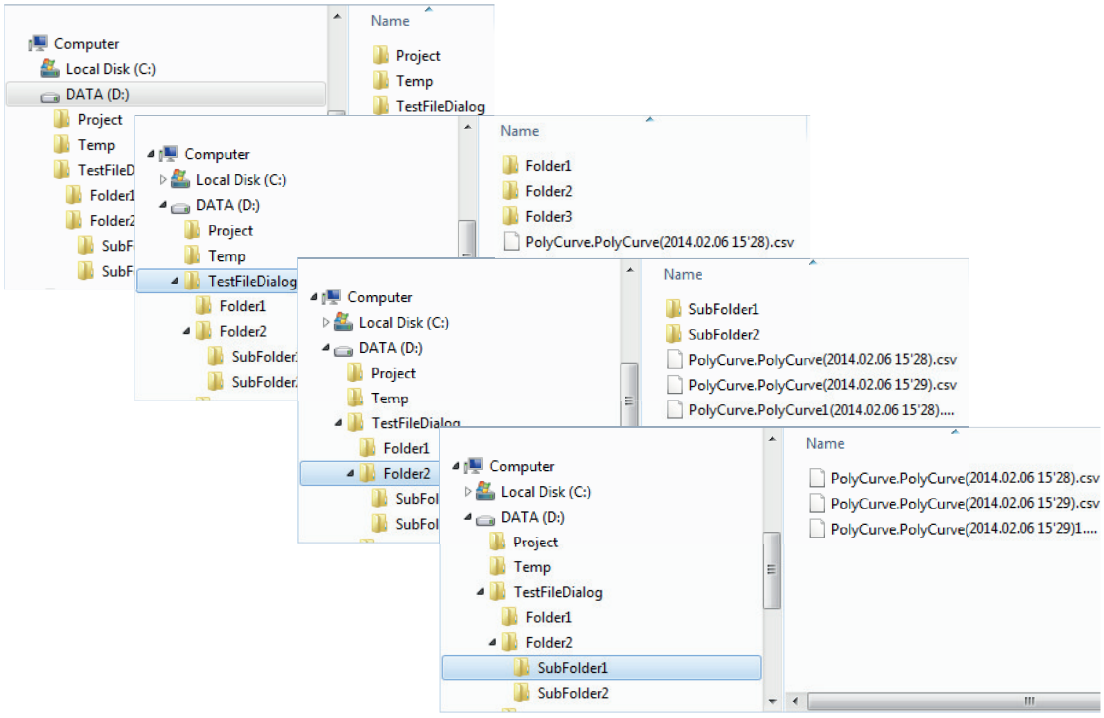
Because folder are shared by means of an internal WinCC tag, you can specify in the project if the folder is shared centrally (set tag to project-wide update) or if you want to share the folder individually for each OS computer (set tag to local computer update).

Example for sharing folders

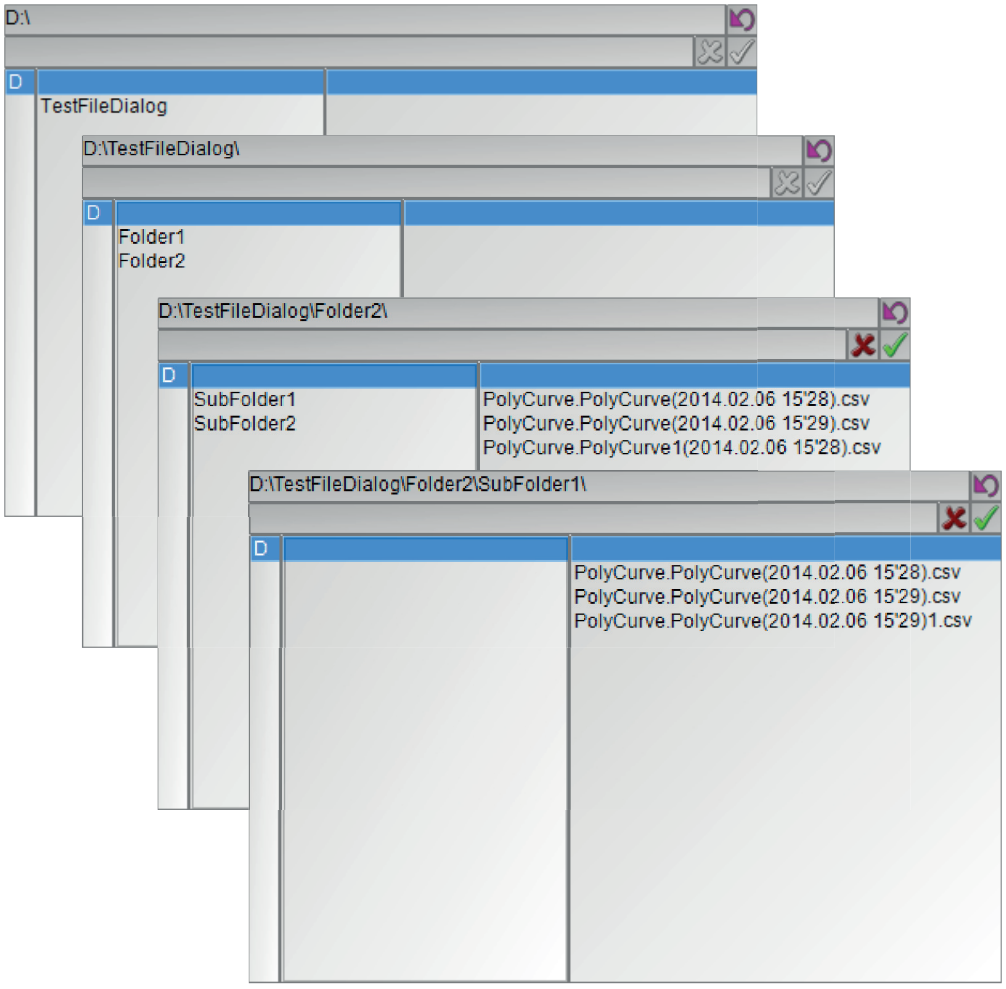
Content of variable @IL_FileDialog_PermittedFolders:

D:\TestFileDialog\Folder1\<crlf>D:\TestFileDialog\Folder2
or
D:\TestFileDialog\Folder1\;D:\TestFileDialog\Folder2

Display of folder structure:



→ Only the folders ..\Folder1 and ..\Folder2 and their contents are permitted. This means the structure would look as follows in the file dialog:



In the top level you can only see drive D:\ because there are only shared folders in drive D:\. Drive D:\ itself has not been shared, which is why you cannot see any files and because there are no files of the type .csv under D:\. No file operations of any kind are permitted. The respective buttons for deleting a file or confirming selection of a file are grayed out. Only the folder 'TestFileDialog' is visible because it holds shared folders.

The same is true for the level below with the folder 'TestFileDialog'. The folder itself is not shared. This is why the .csv file and the shared folder 'Folder3' are not visible. The shared folders 'Folder1' and 'Folder2' are visible.

If a folder is shared, its entire content as well as all subfolders of the shared folder are visible. In the example you can see all subfolders and all .csv files in 'Folder 2'. You can also see the contents of the subfolder 'Subfolder..'.

2.6.3 File dialog error handling

File dialog error handling

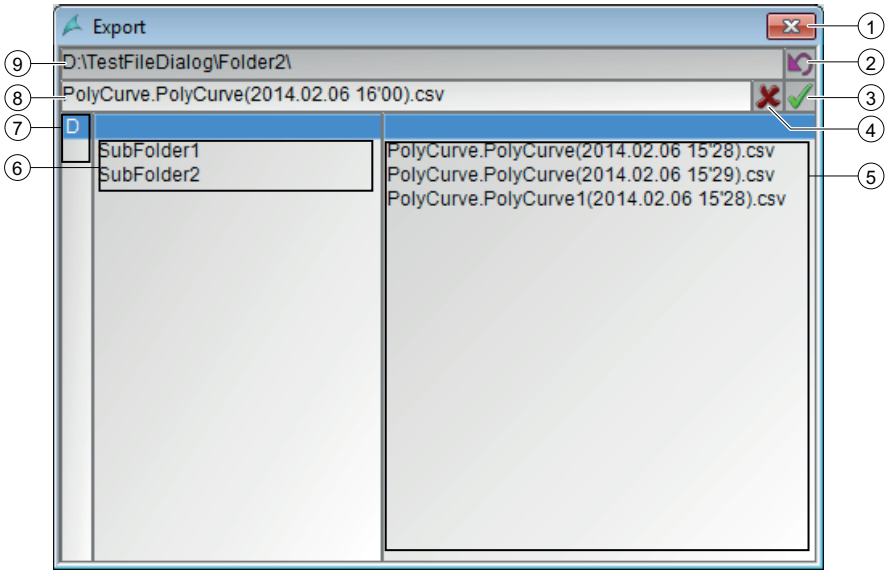
The file dialog itself does not offer error handling. But it can display errors which are transferred by the actual block-dependent import/export function. The error display could look as follows:



Click the error box to hide it.

2.6.4 File dialog views

Click the import/export button in the faceplate of the corresponding block to open the file dialog. The file dialog includes the operator controls shown in the figure below.



Click a selectable drive (7) to enter it in the selected path (9). If you click on a selectable folder (6), it is selected (appended in the "selected path"). Use the "Select higher-level folder" (2) button to move back to the parent folder.

Click a selectable file (5) to select it. With "Delete selected file" (4), the file can be deleted. With "Accept selected file" (3), the file is accepted for import/export. You can close the file dialog without an import/export with "Close Dialog" (1).

If you select the "Export" function, the file name can be edited. A file name is preset for the export depending on the block. It usually consists of the process tag name, block name and any appendix as well as the current date and time (8).

You accept an edited file name by pressing the ENTER key. If not already added, the file dialog automatically adds the .csv extension.

Adapter blocks

3

3.1 General

Topic

In Version 9.0.1 IL blocks of the category "Control Modules" were available that contained the following IL extensions on the source code level in addition to the actual block function:

- Maintenance data (operating cycle and operating hours counter)
- Simulation operation with local and higher-level (unit) activation
- Resetting of simulation operation at unit allocation
- Eight internally ORed locking inputs
- On and Off delays

In this version the blocks of the 'PCS 7 Advanced Process Library' are used for the 'Control Modules' category, with the specified function extensions being made available through the following adapter blocks:

APL block	IL adapter
MotL, MotS	ApMotL
MotRevL	ApMotRL
MotSpdCL	ApMotSCL
VlvL, VlvS	ApVlvL
VlvAnL	ApVlvAnL
VlvDsL	ApVlvDS
Allg	ApCmGen

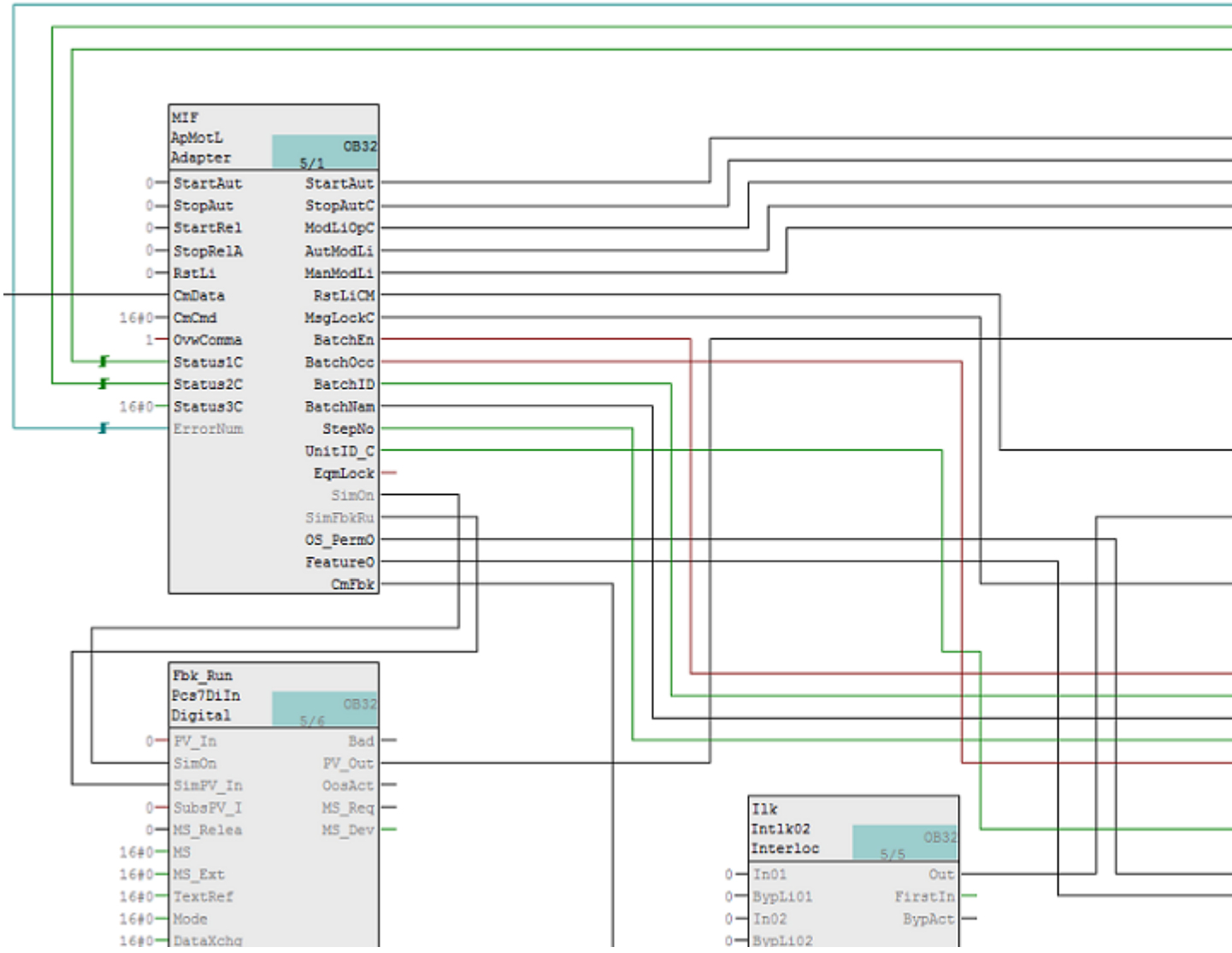
3.2 Basics of adapter blocks

The adapter blocks fulfill the following basic functions:

- Receiving the ‚CmData' data structure from the interface blocks for equipment modules EqmIf, FxCtrl) and passing on to the lower-level CM blocks
- When the centralized command block FxCtrl is used - receiving the control data structure ‚CmCmd' and passing on to the connected CM block. A type-specific break-up of the command into individual signals and transfer of the control outputs is also carried out.
- Collection of the feedback signals and formation of the feedback structure at the "CmFbk" output parameter for feedback of the CM status and of the status information at the interface block of the equipment modules. The Information is derived from the CM feedback signals Status1, Status2 and ErrNum (see CFC screenshot)
- On and Off delays
- Simulation of feedbacks during an active higher-level simulation (equipment module or unit)
- Enabling for manual switching operation
- Enabling for simulation operation
- Maintenance data
- Command transfer

3.2.1 Configuration

- Use the CFC editor to install the block together with the APL-CM clock in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).
- The interconnection is shown here, as an example, at the process tag for a single motor with one run feedback. The APL CM block MotL in addition to the corresponding IL adapter block ApMotL as well as a digital input driver Pcs7DiIn for the feedback signal and an interlocking module IntLk02 are used:



3.2.2 Mode of operation

- The adapter hooks into the "normal" signal flow for the mode switch via the connections `ModLiOp` and `AutModLi`. This means that both connections are again available to the user at the adapter.
  The same is true for the Reset function via the `RstLi` connection.
- The adapter is also switched between the SFC and CM for the control signals `OpenAut` and `CloseAut` – in as far as these are to be delayed.
  - When the control delay is used, the adapter must know the operating mode "Switch or pushbutton operation" of the APL control module. To the purpose the APL input `Feature` has to be formed by the adapter block (interconnection with the output `FeatureCMOut`). Specification of the APL feature bit is then carried out at the input `FeatureCM` of the adapter block.
  - When using the 'small' APL blocks (MotS, VlvS), please note that these blocks do not support "Switch operation". This means that the two control signals `OpenAut` and `CloseAut` must be controlled by the user program/SFC at the adapter block.
- Unit simulation is implemented with the aid of the channel driver blocks.
- The adapter reads the status of the respective CM block of the APL through the interconnections of the inputs `Status1CM`, `Status2CM` and `ErrNumCM` with the corresponding CM outputs. The status display and maintenance data are implemented by these means.
- For the implementation of a *process-conditional* manual enable through the input signal ‚MANOP_EN' or by the data structure "CmData" , the APL input `OS_Perm` has to be formed by the adapter block (interconnection with output `OS_PermCMOut`).

Note

Basic functions for CM adapters and example project

- he basic functions described in the following sections are shared by all the CM adapters.
- Process tag types are supplied for all the adapter blocks in the framework of the example project.

3.2.3 Common connections of the type-specific adapter blocks

The following section describes the common connections, i.e. those that are not type-dependent:

Input parameters

Parameter	Description	Type	Default
`CmdDesc`	Command descriptions; to be parameterized with type specific enumeration ‚Cmd???'	`INT`	
`FbkDesc`	Feedback descriptions; to be parameterized with type specific enumeration ‚Fbk???'	`INT`	

Parameter	Description	Type	Default
...	**Command signals from EQM, process releases, user feedbacks – type specific – see individual Adapter**	DigVal	
...	**...**		
ModLiOp	1=Link/Auto, 0=Manual: Input to auto/manual commands	DigVal	
AutModLi	1=Auto mode: auto mode switch via link	DigVal	
ManModLi	1=Manual mode: manual mode switch via link	DigVal	
RstLi	Linkable reset signal	DigVal	
...	**Command feedback delay times – type specific – see individual Adapter**		
...	**...**		
FbkDelay	Explicit feedback delay during higher-ranking simulation (see FeatBit1)	REAL	1.0
CmData	Control and status information for this CM	UdtCmData	
CmCmd	Command data from FxCtrl	DWORD	
OvwCommands	1=Overwrite CmCmd commands with inputs OpenAut/CloseAut	BOOL	True
OvwModSel	1=Overwrite CmData mode switches	BOOL	
OvwUnitID	1=Overwrite CmData unit ID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
OvwManOpEn	1=Overwrite CmData ManOpEn with input ManOpEn	BOOL	
UnitID	Unit ID (usually from UnitIf block)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIDLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
ManOpEn	Enable switchover to manual mode	BOOL	
OpHoursAct	Operating hours actual value [sec]	DINT	
OpCycleAct	Operating cycle actual value	DINT	
OpHoursLim	Operating hours limit	DINT	
OpCycleLim	Operating cycle limit	DINT	
OpHoursRst	Reset operating hours	BOOL	
OpCycleRst	Reset operating cycle	BOOL	
Status1CM	Status1 from CM	DWORD	
Status2CM	Status2 from CM	DWORD	
Status3CM	Status3 from CM	DWORD	
ErrorNumCM	ErrorNum from CM	ANY	
SampleTime	Sampling time [s]	REAL	1.0

Parameter	Description	Type	Default
OS_Perm	Operator permissions for adapter block - Bit assignments type specific – see individual Adapter	STRUCT_DWORD	
OpSt_In	Enabled operator stations	DWORD	
OS_PermCM	Operator permissions for CM - Bit assignments type specific – see individual APL control module	STRUCT_DWORD	
Feature	Feature parameter for adapter block - Bit assignments type specific – see individual Adapter	STRUCT_DWORD	
FeatureCM	Feature parameter for CM block - Bit assignments type specific – see individual APL control module	STRUCT_DWORD	

Output parameters

Parameter	Description	Type	Default
...	**Command outputs to CM – type specific – see individual Adapter**	DigVal	
...	**...**		
ModLiOpCM	1=Link/Auto 0=Manual: Input to auto/manual commands	DigVal	
AutModLiCM	1=Auto mode: auto mode switch via link	DigVal	
ManModLiCM	1=Manual mode: manual mode switch via link	DigVal	
RstLiCM	Reset signal	DigVal	
MsgLockCM	Message lock from unit	DigVal	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING [32]	
StepNo	Batch Stepp number	DWORD	
OpHoursActOut	Operating hours actual value	DINT	
OpCycleActOut	Operating cycle actual value	DINT	
OpHoursHit	Operating hours limit has been reached	DigVal	
OpCycleHit	Operating cycle limit has been reached	DigVal	
UnitID_CM	Connect to CM's ExtVa104 for unit message filtering	DWORD	
EqmLock	EQM lock signal from CmData	BOOL	
P_RST	Reset pulse	BOOL	
SimOn	To channel driver's SimOn	DigVal	
...	**Simulation Feedback signals – type specific – see individual Adapter**	DigVal	
...	**...**		
FbkDelayOut	Effective feedback delay during higher-ranking simulation	REAL	

Parameter	Description	Type	Default
UsrCmd1Out	Routed user command from FxCtrl (not delayed)	DigVal	
UsrCmd2Out	Routed user command from FxCtrl (not delayed)	DigVal	
CmGrpError	1=Group error is active	DigVal	
CmRdyToStart	1=Ready to start	DigVal	
...	**Control output signals** **– type specific – see individual Adapter**	DigVal	
...	**Control output feedback signals** **– type specific – see individual Adapter**	DigVal	
OS_PermOut	Operator permission: output 1 for OS	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
OS_PermLnk	Operator permission (OS info)	DWORD	
OpSt_Out	Enabled operator stations	DWORD	
OS_PermOutCM	Operator permissions for CM - Bit assignments type specific – see individual Adapter	STRUCT	
FeatureOutCM	Feature collection for connected CM - Bit assignments type specific – see individual APL Control Module	STRUCT	
Status1	Status 1 - Bit assignments type specific – see individual Adapter	DWORD	
St_Worst	Worst signal status	BYTE	16#80
CMFbk	Feedback data for EqmIf / FxCtrl	BlUdtCmFbk	

3.2.4 Faceplate jumps CM block <-> Adapter block

All the adapters dispose of the standard function for calling up further faceplates (attribute ‚BLK_Jump'). This makes it possible to call up the adapter faceplate from the standard view of the APL faceplate – and, vice versa, the associated APL faceplate from the adapter standard view.

- The interconnection SelFP1 <-> ENO between the APL block and the corresponding adapter is used to assign the first of the two free assignable buttons in the APL faceplate for calling faceplates of other blocks.
- The interconnection Status1CM <-> Status1 between the adapter and the associated APL block is used to assign the button that calls the faceplates of other blocks in the adapter faceplate / standard view.
  Special case:
  At the adapter 'ApCmGen' the input Status1CM has to be interconnected for this purpose at an output of the type BOOL (for example ENO).

See the section "Operator control and monitoring (Page 77)" for the visualization.

3.2.5 Command process releases

All the control commands in automatic mode can be interlinked with a process release. The corresponding input signals bear the designation of the commands with the addition "Rel" for "Release" in the middle.

Examples:

Control command	Interlinked with process release:
StartAut	StartRelAut
FwdAut	FwdRelAut
StopAut	StopRelAut

The process releases are of the structure type "DigValFF" (signal state = 16#FF).

Note

If the signal inputs are not interconnected, the releases are always active (standard setting) even when the signal in the CFC Online view is displayed as logic 0. If the signal inputs are interconnected, the releases are active only at logic 1. At logic 0 they are inactive.

Input parameters using the adapter ‚ApMotRevL' as an example

Parameter	Description	Type	Default
StartRelAut	1=Process release for start command	DigValFF	016#FF
FwdRelAut	1=Process release for forward direction command	DigValFF	016#FF
StopRelAut	1=Process release for stop command	DigValFF	016#FF
UsrCmd1RelAut	1=Process release for user command 1	DigValFF	016#FF
UsrCmd2RelAut	1=Process release for user command 2	DigValFF	016#FF
UsrFbk1			
UsrFbk2			

3.2.6 ON and OFF delays

Delay times for the activation of start / open and stop / close commands can be parameterized in the adapter block for the CM blocks of the APL (for example valve, motor, etc.). To this purpose the adapters have type-specific control inputs as "upstream" copies of the "actual" CM inputs.

- Passing on of the control signals to the CM block is carried out via the respective outputs of the same name supplemented by "CM". The adapter is interconnected between the SFC type and the CM block via these inputs and outputs (see below).
- The parameter names of the delay times are the same as those of the control inputs, supplemented by "Delay".

Motor example:	
Control:	StartAut (DigStruct)
Transfer:	StartAutCM (DigStruct)
On delay:	StartAutDelay (Real)
Off delay:	StopAutDelay (Real)

- As indicated by the signal designations, the delay times only function if the automatic control inputs are used.
- Parameterization of the delay times can be carried out via the process object view or the faceplate.
- The input and output delay naturally only functions if the interconnection goes from the SFC type to the adapter block – and not directly to the CM block.
- Parameterization and display of the delay times is carried out in the faceplate / "Standard" view (see section "Operator control and monitoring (Page 77)").

3.2.6.1 Block interfaces

Input parameters

Parameter	Description	Type	Default
`StartAut`	1=Start: Start command in automatic mode	`DigVal`	`0` `16#80`
`StopAut`	1=Stop: Stop command in automatic mode	`DigVal`	`0` `16#80`
`StartAutDelay`	Start delay in automatic mode	`AnaValFF`	`0.0` `16#FF`
`StopAutDelay`	Stop delay in automatic mode	`AnaValFF`	`0.0` `16#FF`

Output parameters

Parameter	Description	Type	Default
StartAutCM	1=Start: Start command in automatic mode	DigVal	16#80
StopAutCM	1=Stop: Stop command in automatic mode	DigVal	16#80

3.2.7 Higher-level controlling of the unit / EQM level

3.2.7.1 Structure data connections EQM <-> CM

Signal transmission from the EQM level to the control modules

The data connection of an equipment module with the adapter blocks of the lower-level CMs is carried out on the basis of a 1:n relationship, meaning that a connection of the EQM interface block (EqmIf, FxCtrl) from the output "CmData" to each assigned CM is required.

This contains the following individual information:

- Batch data
- Unit identifier
- Mode switch / enabling signals
- CM-type-specific control signals

Elements of the data structure CmData

Structur element	Description	Type
dwUnitID	Unit ID	DWORD
dwBatchID	Batch ID	DWORD
sBatchName	Batch name	STRING[32]
dwBatchStepNo	Batch step number	DWORD
xBatchOcc	Batch occupied	BOOL
xBatchEn	Batch enabled	BOOL
xSimEnCM	Higher-level simulation is enabled	BOOL
xSimOnParent	Higher-level simulation is active	BOOL
xEqmLock	Locking EQM / lower-level CMs	BOOL
xManOpEn	Operator can switch to MANUAL mode	BOOL
xResetPls	Reset signal (Impuls)	BOOL
xModLiOp	Switchover linked/operator mode	BOOL
xAutModLi	AUTO mode command	BOOL
xManModLi	MANUAL mode command	BOOL
xMsgLock	Lock messages	BOOL
xRes1		BOOL

Structur element	Description	Type
xRes2		BOOL
xRes3		BOOL
xRes4		BOOL
xRes5		BOOL
byFbkDelayRatio	Feedback delay factor (as a %)	BYTE

Note

This information can also be switched groupwise to external inputs via various "Overwrite" signals in order to allow the CM-individual control:

The following table lists the individual signals per category as well as the associated switchover signals:

Category	Input signal	Switchover signal
Unit identifier	UnitID	OvwUnitID
Batch data	BatchID	OvwBatchInfo
	BatchName	
	StepNo	
	BatchOcc	
	BatchEn	
Manual switching operation	ManOpEn	OvwManOpEn
Mode switch	ModLiOp	OvwModSel
	AutModLi	
Controls	(type-specific)	OvwCommands

The following signals cannot be switched over:

- Simulation enable
- Signal "Higher-level simulation is active"
- EQM lock
- Reset pulse
- Assignment requirement and lower-level function designation (only relevant for "shared equipment")
- Relationship of the feedback simulation time to the CM monitoring time

Signal transmission from the control module to the EQM

A data structure connection from the output "CmFbk" of the adapter block to the inputs "Cm01Fbk ... Cm16Fbk" is required per CM (up to 16 CMs can be interconnected at the EQM interface block "Eqm /FxCtrl"). The data structure contains the feedback information of the respective control module that is used for the status display in the EQM faceplate.

(The internal encoding of this data structure is not relevant for the user.)

3.2.7.2 Command and feedback processing for 'FxCtrl'

Command and feedback processing for ‚FxCtrl'

In order to use the extended function control block ‚FxCtrl' it is necessary for its control commands to be passed on as individual signals to the CMs via the adapter block and for its feedbacks to be collected. Depending on the CM connected, up to 4 commands can be defined, with 2 to 4 commands having a default value, depending on the CM type.

The integration of this function in the adapter blocks allows the type-specific handling of the control and feedbacks for the respective drive type in accordance with the following table:

'Drives' block	Automatic controls	Feedbacks
MotL, MotS	• StartAut	• Status2 (Bit 23): Motor is running
	• StopAut	• Status2 (Bit 20): Motor is stopped
	• UsrCmd1	• UsrFbk1
	• UsrCmd2	• UsrFbk2
MotRevL	• FwdAut	• Status2 (Bit 24): Motor runs in forward mode
	• RevAut	• Status2 (Bit 26): Motor runs in reverse mode
	• StopAut	• Status1 (Bit 09): Motor is stopped
	• UsrCmd1	• UsrFbk1
MotSpdCL	• FwdAut	• Status2 (Bit 24): Motor runs in forward mode
	• RevAut	• Status2 (Bit 26): Motor runs in reverse mode
	• StopAut	• Status2 (Bit 20): Motor is stopped
	• UsrCmd1	• UsrFbk1
VlvL, VlvS	• OpenAut	• Status2 (Bit 20): Valve is open
	• CloseAut	• Status2 (Bit 21): Valve is closed
	• UsrCmd1	• UsrFbk1
	• UsrCmd2	• UsrFbk2
VlvAnL	• OpenAut	• Status2 (Bit 20): Valve is open
	• CloseAut	• Status2 (Bit 21): Valve is closed
	• UsrCmd1	• UsrFbk1
	• UsrCmd2	• UsrFbk2
VlvDS	• OpenAut	• Status2 (Bit 20): Valve is open
	• CloseAut	• Status2 (Bit 21): Valve is closed
	• SLTopAut	• Status1 (Bit 29): Lift top seat is active
	• SLBtmAut	• Status1 (Bit 30): Lift bottom seat is active

- The bits not required for internal controls are made available as "user commands" (outputs `UsrCmd1Out/UsrCmd2Out`) or "user feedbacks" (inputs `UsrFbk1/UsrFbk2`).
- The type-specific connection designations of the commands and feedbacks are described at the respective adapter.
- The generic adapter 'ApCmGen' is to be used for all other block types.

Note

These control commands can also be switched to external inputs via the "Overwrite" input `OvwCommands = 1` in order to allow the CM-individual control. In the case of operation with the FxCtrl block these should however be set to "Internal" (`OvwCommands = 0`), since its control is otherwise ineffective. The current setting is shown in the "Preview view" of the adapter faceplate.

Command identifier

All the adapters have the two inputs `CmdDesc/FbkDesc` at which the control and feedback designations can be specified via type-specific enumerations. The enumerations of the state of delivery are contained in the 'Shared Declarations / Enumerations' folder of the library. These should be copied into the project.

Input parameters

Parameter	Description	Type	Default
CmdDesc	Command descriptions	INT	
FbkDesc	Feedback descriptions	INT	

The assignment between the command or feedback and the enumeration value is carried out via the identification numbers.

Example for reversing motor (user adaptation for "Cmd4"):

Internal	Enum value	Text
`Cmd1`	1	`StartFwd`
`Cmd2`	2	`StartRev`
`Cmd3`	3	`Stop`
`Cmd4`	4	`ReleaseTm`

If the identifiers are to be defined instance-specifically and no longer type-specifically, own enumerations have to be defined for the respective instances and parameterized at the block input.

3.2.7.3 Automatic releases at unit allocation

Automatic releases at unit allocation

In the adapter blocks the specification at the unit block `UnitIf` is implemented as follows with regard to the behavior at unit allocation:

Enabling of the mode switch (ModLiOp) and controlling of the operating mode automatic / manual operation (AutModLi) of the lower level CM blocks:

- At unit allocation
  `ModLiOp = 1` (enable for interconnection) and `AutModLi = 1` (automatic mode)
- In Hold operation
  `ModLiOp = 0` (enable for faceplate / operator)

3.2.7.4 Higher-level simulation

Higher-level simulation

- The adapter provides type-specific and time-delayed feedback signals for the implementation of an active higher-level simulation (origin: equipment module or unit) that can be passed on via the channel drivers to the APL-CM (see table of input / output parameters).
- The delay times can be specified by two methods:
  - *Locally-explicitly* using the input FbkDelay [s]
  - *Higher-level-implicitly* using the monitoring time configured at the CM and a unit-wide time reduction ratio `FbkDelayRatio` [0%...100%] specified at the block UnitIf
    Differentiation is carried out by means of a feature bit:
    `Bit1 : BOOL //Feedback delay:1 = CM's MonTiDynamic value (fraction),0 = input "FbkDelay"`
- Input of the delay time `FbkDelay` is carried out at the associated faceplate / "Standard" view (see section "Operator control and monitoring (Page 77)"). The delay time is only adopted with the start of the simulation. Value changes during the simulation are not taken into consideration.
- Activation of the higher-level simulation can be carried out via the faceplates of the unit (`UnitIf`) or the equipment module (`EqmIf`).

Input parameters

Parameter	Description	Type	Default
`FbkDelay`	Explicit feedback delay during higher-ranking simulation (see FeatBit1)	`REAL`	`1.0`

Output parameters

Parameter	Description	Type	Default
`SimOn`	To channel driver's SimOn	`DigVal`	`0` `16#80`
`SimOnFbkRun`	To channel driver's SimPV_In	`DigVal`	`0` `16#80`
`FbkDelayOut`	Effective feedback delay during higher-ranking simulation	`REAL`	`1.0`

3.2.7.5 Message lock

Message lock

- At CM blocks that are capable of issuing messages, it is possible to suppress messages via the block input `MsgLock` .
- All the adapters make the `MsgLockCM` output available to this purpose. The source signal is generated by the unit in the block `UnitIf` and distributed through the `"CmData"` structure to all the CM adapters.

Note

The complete message suppression has to be set explicitly through a feature bit at the CM block.

Output parameters

Parameter	Description	Type	Default
`MsgLockCM`	Message lock from unit	`DigVal`	`0` `16#80`

3.2.8 Enabling for manual switching operation / Block for simulation operation

At the CM blocks of the APL the process-conditional enabling of the mode switch (previously "AUTOP_EN" and "MANOP_EN") is carried out via the "OS-Permissions".

- To this purpose the respective `OS_Perm` input of the CM block is interconnected with the `OS_PermOutCM` output of the adapter. This forms the possibly limited authorization internally and passes the result through the interconnection to the APL block
- A process-conditional manual enable is implemented when required through an interconnection of the user logic to the structure input `OS_PermCM` of the adapter block+
- The authorizations for the adapter block itself have to be set at the `OS_Perm` input of the adapter.
- Simulation operation of the lower-level CM blocks can be blocked centrally on the unit level l with the exception of the adapter block for the APL block PIDConL . This does not have an OS_Perm bit for the simulation enable.

3.2.8.1 Block interfaces

Input parameters

Parameter	Description	Type	Default
OS_Perm	Operator permissions for Adapter block itself	STRUCT	
OS_PermCM	Operator permissions for APL block	STRUCT	

Output parameters

Parameter	Description	Type	Default
OS_PermOut	Operator permissions: output 1 for OS	DWORD	16#FFFFFFFF
OS_PermLog	Operator permissions: output 2 for OS	DWORD	16#FFFFFFFF
OS_Perm_Lnk	Operator permissions (OS info)	DWORD	
OS_PermOutCM	Linked and manipulated OS_PermCM	STRUCT	

Detailed representation

Detailed representation of the **input / output structures** OS_Perm/OS_PermOut

- Since these vary type-specifically, the description is carried out at the respective adapter.
- Visualization of the enables is carried out in the faceplate / "Standard" view (see section "Operator control and monitoring (Page 77)").

Detailed representation of the **input / output structures** OS_PermCM/OS_PermOutCM

- The operator authorizations of the CM blocks of the APL as well as their visualization are described in the APL documentation

3.2.9 Maintenance data

Simple operating hour and switching cycle counters are implemented in the adapter blocks.

- Limits whose exceedance is signaled at block outputs can be specified for both.
- A message is not emitted (the adapters are not capable of issuing messages). If required, the limit violation can be interconnected to free message inputs of the adapted CM block.
- Operating hours and operating cycles are counted internally directly at the block inputs in DINT format because only these can be read back in CFC.
- The resolution of the operating hours counter amounts to seconds that are counted internally in DINT format.
- The actual values are made available as outputs for project-specific evaluations.
- Operator control and monitoring of the maintenance data is carried out **in the faceplate** (see section "Operator control and monitoring (Page 77)")

- Representation of the operating hours → ddddd-hh-mm (max. approx. 68 years)
- Representation of the operating cycles → x.xxx.xxx.xxx (max approx. 2.1 billion)

3.2.9.1 Block interfaces

Input parameters

Parameter	Description	Type	Default
OpHoursAct	Operating hours actual value [sec]	DINT	0
OpCycleAct	Operating cycle actual value	DINT	0
OpHoursLim	Operating hours limit	DINT	
OpCycleLim	DINT	DINT	
OpHoursRst	1 = Reset operating hours	BOOL	0
OpCycleRst	1 = Reset operating cycle	BOOL	0

Output parameters

Parameter	Description	Type	Default
OpHoursActOut	Operating hours actual value [sec]	DINT	0
OpCycleActOut	Operating cycle actual value	DINT	0
OpHoursHit	Operating hours limit has been reached	DigVal	
OpCycleHit	Operating cycle limit has been reached	DigVal	

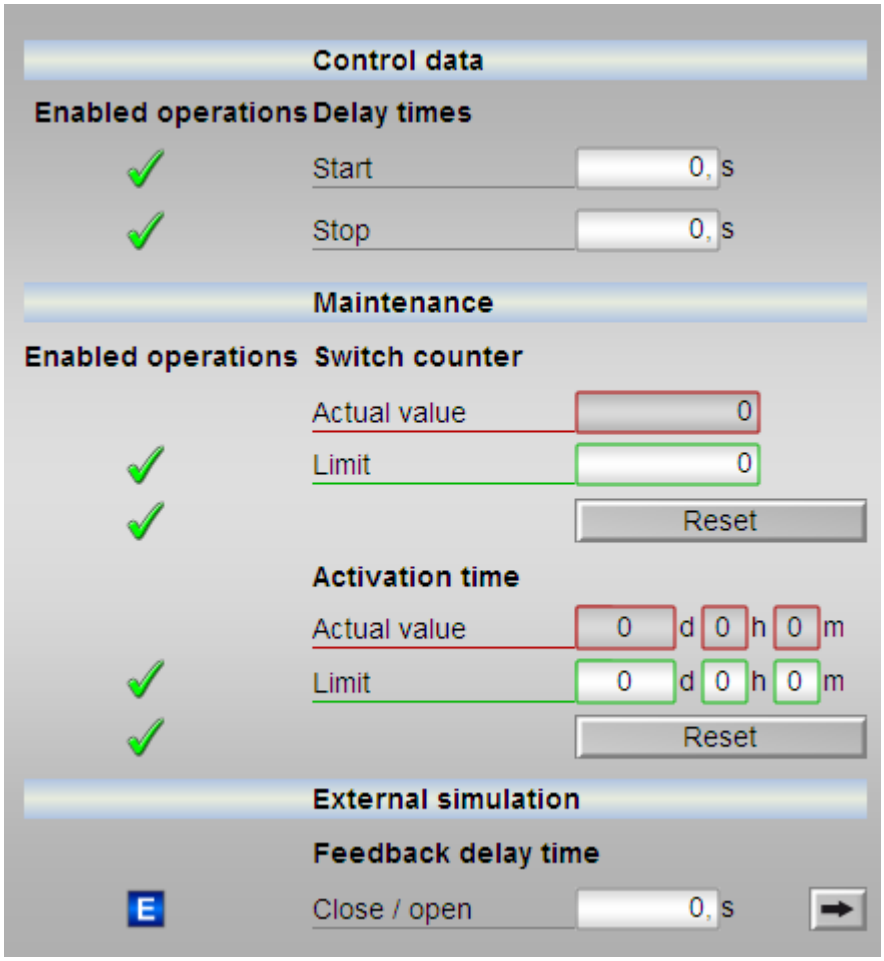
3.2.10 Operator control and monitoring

Views of the adapter blocks

Operator control and monitoring of the adapter blocks is carried out via an OS faceplate that makes the following views available

Symbol	View	Information
	Standard view	The "Standard" and "Preview" views are explained below. See below.
	Preview	
	Memo view	The memo and batch views correspond to those of the CM blocks of the APL-Library. Further information is available in the APL documentation.
	Batch view	

Standard view using "ApMotL" as an example

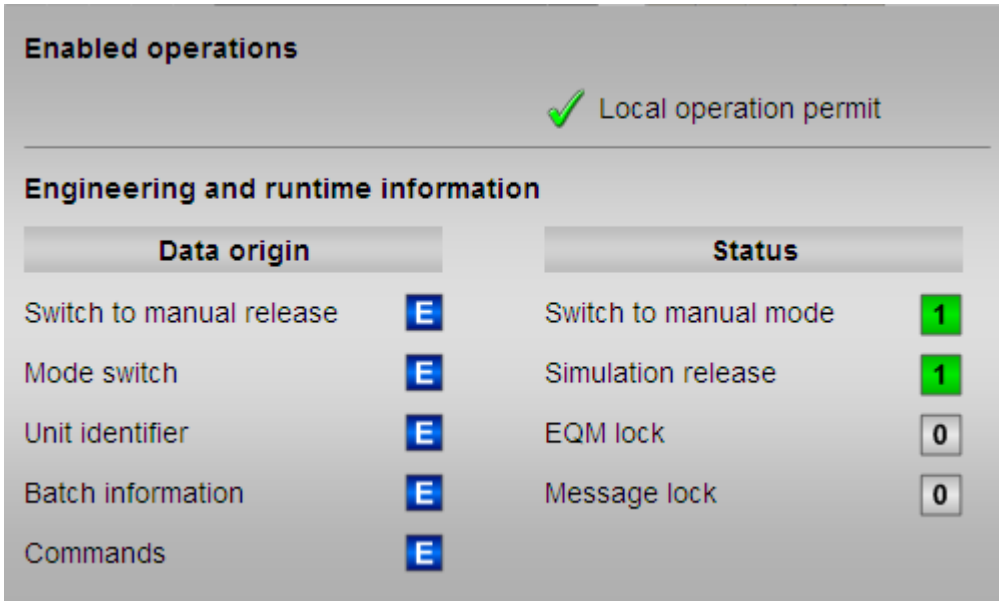


Explanation of the screen shot	
(1)	**Start / Stop delay time**
	The delay times for the excitation signals can be configured here: (Operator authorization = "Higher-order operator process controls") • Start (at valves → Open) • Stop (at valves → Close)
(2)	**Operating cycle / Operating hours counter / values**
	• Actual value (only display) • Limit value (operator authorization = "higher-order operator process controls")
(3)	**Operating cycle / Operating hours counter / resetting**
	• Operating cycle / Operating hours counter resetting (Operator authorization = "Higher-order operator process controls") • Representation of the operating hours ddddd-hh-mm (max. approx. 68 years) • Representation of the operating cycles x.xxx.xxx.xxx (max approx. 2.1 billion)

Explanation of the screen shot		
(4)	**External simulation / Feedback delays**	
	Here the delay time for starting / stopping or opening / closing can be configured, as well as displayed for the unit simulation operation: • "Internal" mode: Delay time can be configured (operator authorization = "higher-order operator process controls") • "External" mode: Delay time is calculated	
	The symbol identifies the source of the feedback delay at simulation operation:	
		The delay time is defined by the block input "FbkDelay" [s]
		The delay time is defined by the monitoring time of the connected CM block and a time reduction ratio [0 ... 100%] defined at the unit block UnitIf
(5)	**Faceplate jump**	
	Faceplate call of the connected CM block	
(6)	**Operator control enables for delay times and maintenance data**	
	The current operator control enables of the block instance are displayed in this area. They depend on the parameter OS_Perm of the adapter block.	

Preview view using "ApMotL" as an example:

The preview view shows the operator control enables, parameters and states of the adapter block. However, there are no actual possible operations in this view.



Explanation of the screen shot	
(1)	**Enabled operations**
	The state of the "Local operator permission" based on the "OpSt_In" input parameter and "Feature Bit 24" is displayed uniformly in this section at all the adapter blocks. (See APL documentation / section "Operator control permissions" Start (for valves → Open))

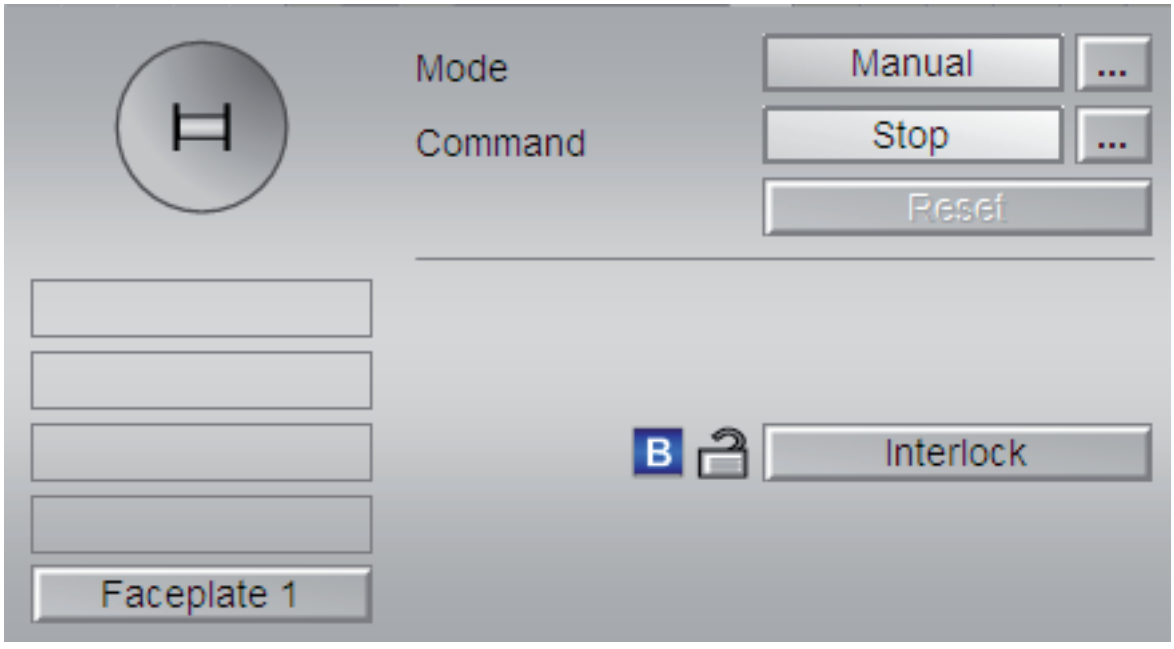
Explanation of the screen shot			
(2)	**Engineering and runtime information**		
	This section contains information from the status word `Status1`		
	• The "Data origin" column shows the setting of the overwrite signals:		
		Enable manual mode	`OvwManOpEn`
		Mode switch	`OvwModSel`
		Unit identifier	`OvwUnitID`
		Batch information	`OvwBatchInfo`
		Control	`OvwCommands`
		• The symbols have the following meaning:	
		I	Signal origin = Unit / CmData structure
		E	Signal origin = Block input Ovw<xxx>
	• The 'States' column shows the following state bits (0 / 1):		
		Enable manual mode	`ManOpEn` or `CmData.xManOpEn` depending on `OvwManOpEn`
		Simulation release	`CmData.xSimEnCM`
		EQM lock	`CmData.xEqmLock`
		Message lock	`CmData.xMsgLock`

Calling the adapter faceplate from the APL faceplate

The adapter faceplate does not have an own block icon. Calling is carried out from the associated APL block.

Example:

Call from VlvMotL faceplate:



Explanation of the screen shot	
(1)	**Faceplate button**
	Calling is carried out from the respective CM block faceplate using the faceplate button of the normal view. The button labeling can be changed in the WinCC object attributes.

3.3 Adapter blocks – details

3.3.1 ApMotL

Mode of operation

This block is used together with the CM block of the APL **‚MotL'** (control module) to control motors. Various inputs / outputs are available for controlling the motor.

For general information about the configuration and mode of operation see section: Basics of adapter blocks (Page 62)

General adapter functions

This block includes the following general functions of the adapter blocks:

- Faceplate jumps CM block <-> Adapter block (Page 67)
- Command process releases (Page 68)
- ON and OFF delays (Page 69)
- Structure data connections EQM <-> CM (Page 70)
- Enabling for manual switching operation / Block for simulation operation (Page 75)
- Maintenance data (Page 76)

Status word allocation for the output parameter ‚Status1'

For a description of the individual parameters, see the section Common connections of the type-specific adapter blocks (Page 64).

Status bit	Parameter
0	`BatchOcc`
1	`BatchEn`
2	`OpHoursHit.Value` 'Operating hours' upper limit violation
3	`OpCycleHit.Value` 'Operating cycles' upper limit violation
4	`xFeatFbkDelay` Copy of Feature Bit 1: Feedback delay: • 1 = CM's `MonTiDynamic` percentage factor • 0 = input "`FbkDelay`"
5	`CmData.xSimOnParent`
6	`SimFbkRun.Value`
7	`Reserve`
10	`OvwCommands`
11	`OvwModSel`
12	`OvwUnitId`

Status bit	Parameter
13	OvwBatchInfo
14	OvwManOpEn
15	xManOpEn
16	CmData.xSimEnCM
17	CmData.xEQMLock
18	CmData.xMsgLock
...	...

Configurable reactions using the Feature parameter

The following reactions are available for this block:

Bit	Function
0	Display of the simulation: 1 = Common display of the local and higher-level simulation
1	Feedback delay at unit simulation operation: • 1 = Percentage value of the CM monitoring time (factor = MonTiDynamic) • 0 = Input FbkDelay
2	Display of the CM instance name *) 1 = CFC name (process tag)
3	Display of the CM instance name *) 1 = Block name
4	Reserve
5	Reserve
6	Type of feedback signal UsrFbk1 • 1 = UsrFbk1 signals error • 0 = UsrFbk1 signals feedback
7	Type of feedback signal UsrFbk2 • 1 = UsrFbk2 signals error • 0 = UsrFbk2 signals feedback
8	Reserve
...	...
24	Local operator permission: 1 = Activated
...	...

*) It is also possible to have both names together!

Operator permissions

The block has the following permissions for the OS_Perm parameter:

Bit	Function
0	1 = Operator can change delay time for Start
1	1 = Operator can change delay time for Stop

Bit	Function
2	1 = Operator can change operating hours counter limit
3	1 = Operator can change switching cycle counter limit
4	1 = Operator can reset operating hours counter
5	1 = Operator can reset switching cycle counter
6	Reserve
...	...

Forming the signal status for blocks

This adapter block includes the APL standard function 'Form and output signal status for blocks'. The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `StartAutCM` (depending on `OvwCommands`)
- `StopAutCM` (depending on `OvwCommands`)
- `StartRelAut`
- `StopRelAu`
- `UsrCmd1RelAut`
- `UsrCmd2RelAut`
- `UsrFbk1`
- `UsrFbk2`
- `ModLiOpCM` (depending on `OvwModSel`)
- `AutModLiCM` (depending on `OvwModSel`)
- `ManModLiCM` (depending on `OvwModSel`)
- `RstLi`
- `StartAutDelay`
- `StopAutDelay`

3.3.1.1 Connections of ApMotL

The general (non-type-dependent) block interfaces are described in the section 'Common connections of the type-specific adapter blocks (Page 64)'. The following section describes the type-specific block interfaces:

Input parameters

Parameter	Description	Type	Default
...	...		
`StartAut`	1=Start: Start command in automatic mode	`DigVal`	
`StopAut`	1=Stop: Stop command in automatic mode	`DigVal`	
`StartRelAut`	1=Process release for start command	`DigValFF`	

Parameter	Description	Type	Default
StopRelAut	1=Process release for stop command	DigVAlFF	
UsrCmd1RelAut	1=Process release for user command 1	DigValFF	
UsrCmd2RelAut	1=Process release for user command 2	DigValFF	
UsrFbk1	Feedback for user command 1 (follows command if not connected)	DigValFF	
UsrFbk2	Feedback for user command 2 (follows command if not connected)	DigValFF	
...	...		
StartAutDelay	Start delay in automatic mode	AnaValFF	
StopAutDelay	Stop delay in automatic mode	AnaValFF	
...	...		

Output parameters

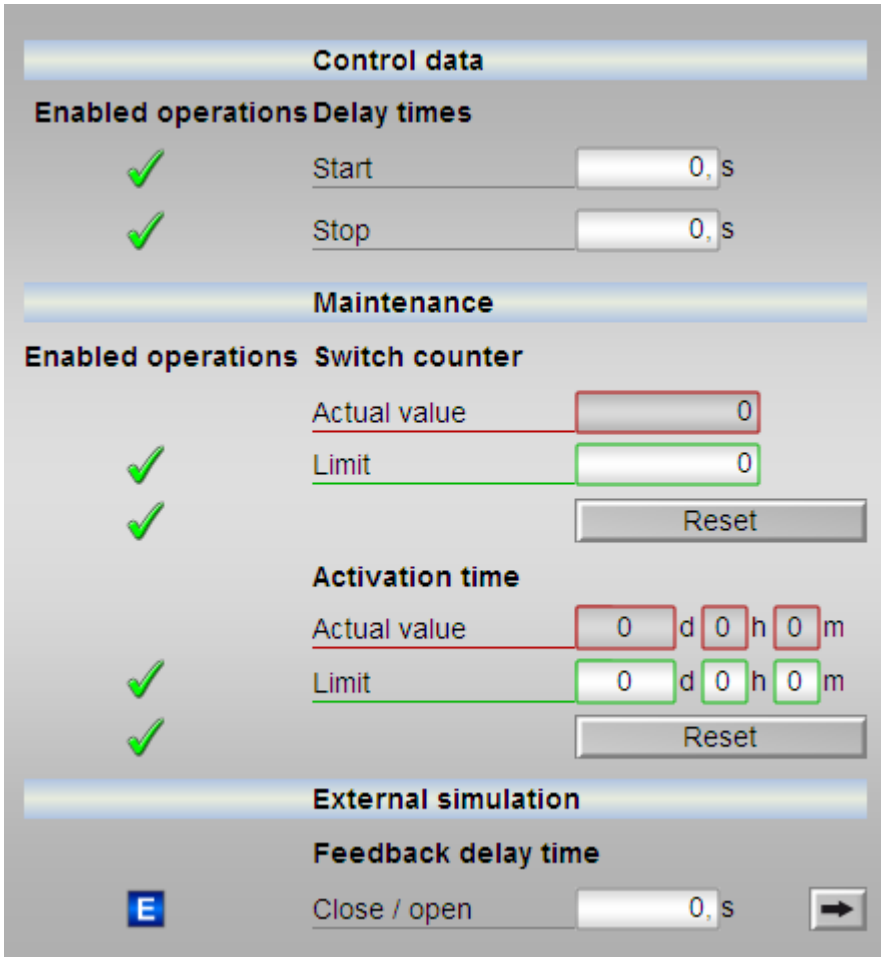
Parameter	Description	Type	Default
StartAutCM	1=Start: Start command in automatic mode	DigVal	
StopAutCM	1=Stop: Stop command in automatic mode	DigVal	
...	...		
SimFbkRun	To channel driver's SimPV_In	DigVal	
CmStart	Control output to start motor	DigVal	
CmAutAct	1=Auto: Automatic mode indicator	DigVal	
CmFbkRun	1=Run, 0=Stop: Feedback of control output	DigVal	
...			

3.3.1.2 Operator control and monitoring

For general information about the OS faceplates refer to the section "Operator control and monitoring (Page 77)".

Type-specific information for the adapter block ‚ApMotL':

- The designations of the control data are:
  'Start' / 'Stop'



3.3.2 ApMotRevL

Mode of operation

This block is used together with the CM block of **‚ApMotRevL'** (control module) to control reversible motors. Various inputs / outputs are available for controlling the motor.

For general information about the configuration and mode of operation see section: Basics of adapter blocks (Page 62)

General adapter functions

This block includes the following general functions of the adapter blocks:

- Faceplate jumps CM block <-> Adapter block (Page 67)
- Command process releases (Page 68)
- ON and OFF delays (Page 69)
- Structure data connections EQM <-> CM (Page 70)
- Enabling for manual switching operation / Block for simulation operation (Page 75)
- Maintenance data (Page 76)

Status word allocation for the output parameter ‚Status1'

For a description of the individual parameters, see the section 'Common connections of the type-specific adapter blocks (Page 64)'.

Status bit	Parameter
0	Copy of `BatchOcc`
1	Copy of `BatchEn`
2	1 = Operating hours limit reached
3	1 = Operating cycles limit reached
4	`xFeatFbkDelay` Copy of Feature Bit 1: Feedback delay: • 1 = `MonTiDynamic` (value in %) • 0 = Input parameter `FbkDelay`
5	Copy of `CmData.xSimOnParent`
6	Copy of `SimFbkFwd.Value`
7	Copy of `SimFbkRev.Value`
8	`Reserve`
...	...

Configurable reactions using the Feature parameter

The following reactions are available for this block:

Bit	Function
0	Display of the simulation: 1 = Common display of the local and higher-level simulation
1	Feedback delay at unit simulation operation: • 1 = Percentage value of the CM monitoring time `MonTiDynamic` (value as %) • 0 = Input parameter `FbkDelay`
2	Display of the CM instance name *) 1 = CFC name (process tag)
3	Display of the CM instance name *) 1 = Block name
4	`Reserve`

Bit	Function
5	Reserve
6	Reserve
7	Type of feedback signal UsrFbk1 • 1 = UsrFbk1 signals error • 0 = UsrFbk1 signals feedback
8	Reserve
...	...
24	Local operator authorization: 1 = Activated
...	...

*) It is also possible to have both names together!

Operator permissions

The block has the following authorizations for the OS_Perm parameter:

Bit	Function
0	1 = Operator can change delay time for "Forward"
1	1 = Operator can change delay time for "Reverse"
2	1 = Operator can change delay time for "Stop"
3	1 = Operator can change operating hours counter limit
4	1 = Operator can change switching cycle counter limit
5	1 = Operator can reset operating hours counter
6	1 = Operator can reset switching cycle counter
7	Reseve
...	...

Forming the signal status for blocks

This adapter block includes the APL standard function 'Form and output signal status for blocks'. The worst signal status ST_Worst for the block is formed from the following parameters:

- FwdAutCM (depending on OvwCommands)
- RevAutCM (depending on OvwCommands)
- StopAutCM (depending on OvwCommands)
- FwdRelAut
- RevRelAut
- StopRelAu
- UsrCmd1RelAut
- UsrFbk1
- UsrFbk2
- ModLiOpCM (depending on OvwModSel)

- AutModLiCM (depending on OvwModSel)
- ManModLiCM (depending on OvwModSel)
- RstLi
- FwdAutDelay
- RevAutDelay
- StopAutDelay

3.3.2.1 Connection of ApMotRevL

The general (non-type-dependent) block interfaces are described in the section 'Common connections of the type-specific adapter blocks (Page 64)'. The following section describes the type-specific block interfaces:

Input parameters

Parameter	Description	Type	Default
FwdAut	1=Start: Forward command in automatic mode	DigVal	
StopAut	1=Stop: Stop command in automatic mode	DigVal	
RevAut	1=Start: Reverse start command in automatic mode	DigVal	
FwdRelAut	1=Process release for stop command	DigValFF	
StopRelAut	1=Process release for stop command	DigVAlFF	
RevRelAut	1=Process release for reverse start command	DigValFF	
UsrCmd1RelAut	1=Process release for user command 1	DigValFF	
UsrFbk1	Feedback for user command 1 (follows command if not connected)	DigValFF	
...	...		
FwdAutDelay	Start delay in automatic mode	AnaValFF	
RevAutDelay	Reverse start delay in automatic mode	AnaValFF	
StopAutDelay	Stop delay in automatic mode	AnaValFF	
...	...		

Output parameters

Parameter	Description	Type	Default
FwdAutCM	1=Start: Forward start command in automatic mode	DigVal	
StopAutCM	1=Stop: Stop command in automatic mode	DigVal	
RevAutCM	1=Start: Reverse start command in automatic mode	DigVal	
...	...		
SimFbkFwd	To channel driver's SimPV_In (Forward)	DigVal	
SimFbkRev	To channel driver's SimPV_In (Reverse)	DigVal	
...			
CmFwd	Control output to start motor forward	DigVal	

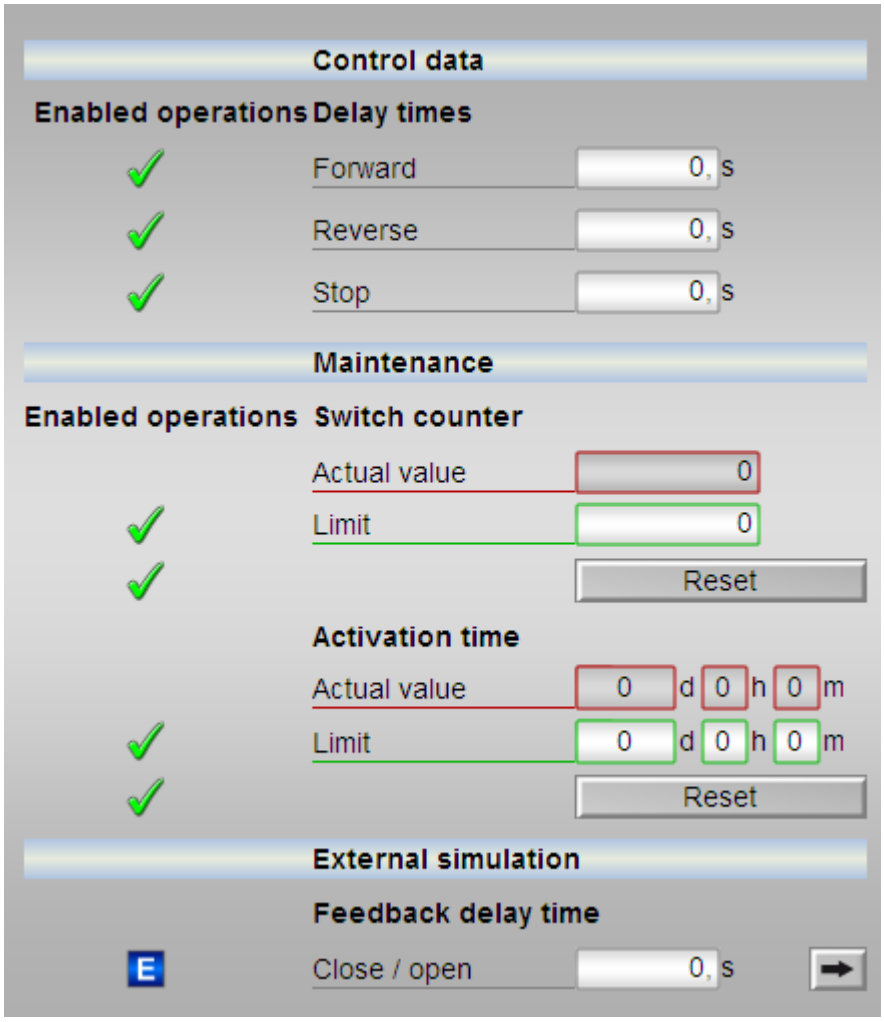
Parameter	Description	Type	Default
CmRev	Control output to start motor reverse	DigVal	
CmAutAct	1=Auto: Automatic mode indicator	DigVal	
CmFbkFwd	1=Run, 0=Stop: Feedback of control forward	DigVal	
CmFbkRev	1=Run, 0=Stop: Feedback of control reverse	DigVal	

3.3.2.2 Operator control and monitoring

For general information about the OS faceplates refer to the section "Operator control and monitoring (Page 77)".

Type-specific information for the adapter block ‚ApMotRevL':

- The designations of the control data at the reversing motor are:
  'Forward' / 'Reverse' / 'Stop'



3.3.3 ApMotSpdCL

Mode of operation

This block is used together with the CM block of the APL **‚MotSpdCL'** (control module) to control motors with two directions of rotation and various speeds. Various inputs / outputs are available for controlling the motor.

For general information about the configuration and mode of operation see section: Basics of adapter blocks (Page 62)

General adapter functions

This block includes the following general functions of the adapter blocks:

- Faceplate jumps CM block <-> Adapter block (Page 67)
- Command process releases (Page 68)
- ON and OFF delays (Page 69)
- Structure data connections EQM <-> CM (Page 70)
- Enabling for manual switching operation / Block for simulation operation (Page 75)
- Maintenance data (Page 76)

Status word allocation for the output parameter ‚Status1'

For a description of the individual parameters, see the section 'Common connections of the type-specific adapter blocks (Page 64)'.

Status bit	Parameter
0	Copy of `BatchOcc`
1	Copy of `BatchEn`
2	1 = Operating hours limit reached
3	1 = Operating cycles limit reached
4	`xFeatFbkDelay` Copy of Feature Bit 1: Feedback delay: • 1 = `MonTiDynamic` (value in %) • 0 = Input parameter `FbkDelay`
5	Copy of `CmData.xSimOnParent`
6	Copy of `SimFbkFwd.Value`
7	Copy of `SimFbkRev.Value`
8	`Reserve`
...	...

Configurable reactions using the Feature parameter

The following reactions are available for this block:

Bit	Function
0	Display of the simulation: 1 = Common display of the local and higher-level simulation
1	Feedback delay at unit simulation operation: • 1 = Percentage value of the CM monitoring time (factor = `MonTiDynamic`) • 0 = Input `FbkDelay`
2	Display of the CM instance name *) 1 = CFC name (process tag)
3	Display of the CM instance name *) 1 = Block name
4	`Reserve`
5	`Reserve`
6	`Reserve`
7	Type of feedback signal `UsrFbk1` • 1 = UsrFbk1 signals error; • 0 = UsrFbk1 signals feedback
...	...
24	Local operating permission: 1 = Activated
...	...

*) It is also possible to have both names together!

Operator permissions

The block has the following authorizations for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can change delay time for "Forward"
1	1 = Operator can change delay time for "Reverse"
2	1 = Operator can change delay time for "Stop"
3	1 = Operator can change operating hours counter limit
4	1 = Operator can change switching cycle counter limit
5	1 = Operator can reset operating hours counter
6	1 = Operator can reset switching cycle counter
7	Reserve
...	...

Forming the signal status for blocks

This adapter block includes the APL standard function 'Form and output signal status for blocks'. The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `FwdAutCM` (depending on `OvwCommands`)
- `RevAutCM` (depending on `OvwCommands`)
- `StopAutCM` (depending on `OvwCommands`)
- `FwdRelAut`
- `RevRelAut`
- `StopRelAu`
- `UsrCmd1RelAut`
- `UsrFbk1`
- `UsrFbk2`
- `ModLiOpCM` (depending on `OvwModSel`)
- `AutModLiCM` (depending on `OvwModSel`)
- `ManModLiCM` (depending on `OvwModSel`)
- `RstLi`
- `FwdAutDelay`
- `RevAutDelay`
- `StopAutDelay`

3.3.3.1 Connections of ApMotSpdCL

The general (non-type-dependent) block interfaces are described in the section 'Common connections of the type-specific adapter blocks (Page 64)'. The following section describes the type-specific block interfaces:

Input parameters

Parameter	Description	Type	Default
`FwdAut`	1=Start: Forward command in automatic mode	`DigVal`	
`StopAut`	1=Stop: Stop command in automatic mode	`DigVal`	
`RevAut`	1=Start: Reverse start command in automatic mode	`DigVal`	
`FwdRelAut`	1=Process release for forward start command	`DigValFF`	
`StopRelAut`	1=Process release for stop command	`DigVAlFF`	
`RevRelAut`	1=Process release for reverse start command	`DigValFF`	
`UsrCmd1RelAut`	1=Process release for user command 1	`DigValFF`	
`UsrFbk1`	Feedback for user command 1 (follows command if not connected)	`DigValFF`	
…	…		
`FwdAutDelay`	Start delay in automatic mode	`AnaValFF`	

Parameter	Description	Type	Default
RevAutDelay	Reverse start delay in automatic mode	AnaValFF	
StopAutDelay	Stop delay in automatic mode	AnaValFF	
...	...		

Output parameters

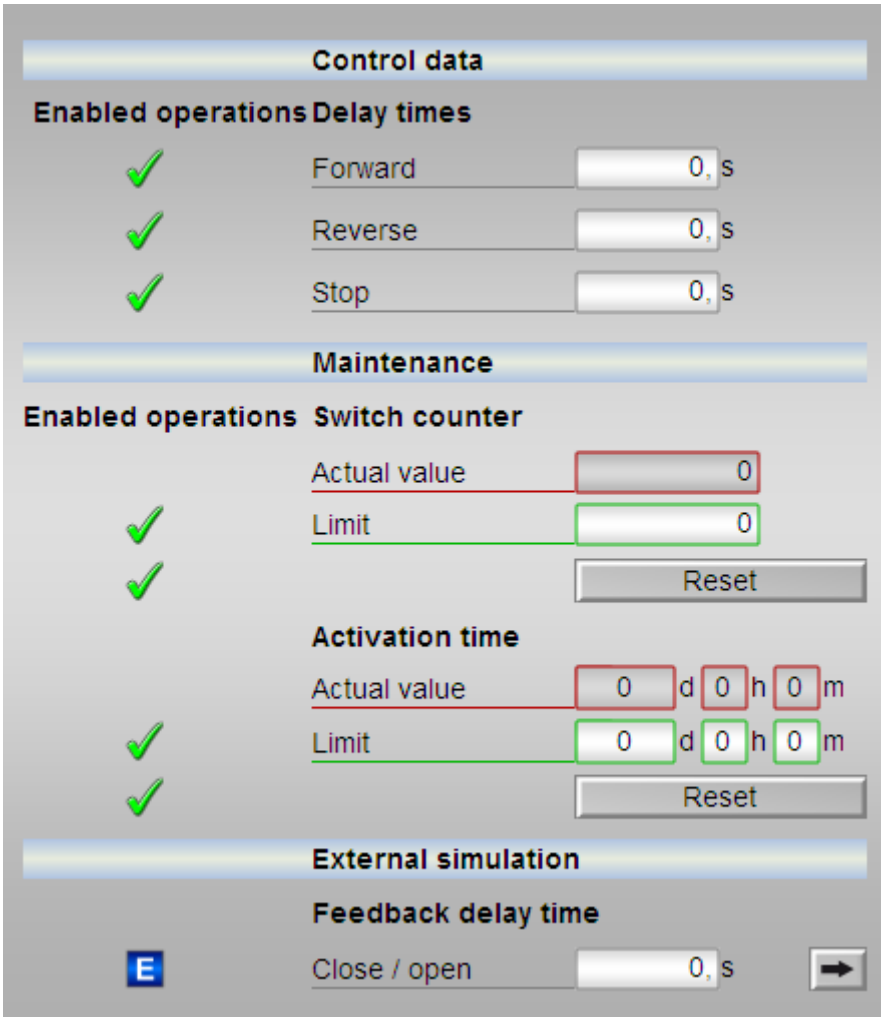
Parameter	Description	Type	Default
FwdAutCM	1=Start: Forward start command in automatic mode	DigVal	
StopAutCM	1=Stop: Stop command in automatic mode	DigVal	
RevAutCM	1=Start: Reverse start command in automatic mode	DigVal	
...	...		
SimFbkFwd	To channel driver's SimPV_In (Forward)	DigVal	
SimFbkRev	To channel driver's SimPV_In (Reverse)	DigVal	
...			
CmFwd	Control output to start motor forward	DigVal	
CmRev	Control output to start motor reverse	DigVal	
CmAutAct	1=Auto: Automatic mode indicator	DigVal	
CmFbkFwd	1=Run, 0=Stop: Feedback of control forward	DigVal	
CmFbkRev	1=Run, 0=Stop: Feedback of control reverse	DigVal	

3.3.3.2 Operator control and monitoring

For general information about the OS faceplates refer to the section "Operator control and monitoring (Page 77)".

Type-specific information for the adapter block ‚ApMotSpdL':

- The designations of the control data are:
  'Forward' / 'Reverse' / 'Stop'



3.3.4 ApVlvL

Mode of operation

This block is used together with the CM block of the APL **‚VlvL'** (control module) to control valves. Various inputs / outputs are available for controlling the valve.

For general information about the configuration and mode of operation see section: Basics of adapter blocks (Page 62)

General adapter functions

This block includes the following general functions of the adapter blocks:

- Faceplate jumps CM block <-> Adapter block (Page 67)
- Command process releases (Page 68)
- ON and OFF delays (Page 69)
- Structure data connections EQM <-> CM (Page 70)
- Enabling for manual switching operation / Block for simulation operation (Page 75)
- Maintenance data (Page 76)

Status word allocation for the output parameter ‚Status1'

For a description of the individual parameters, see the section Common connections of the type-specific adapter blocks (Page 64).

Status bit	Parameter
0	Copy of `BatchOcc`
1	Copy of `BatchEn`
2	1 = Operating hours limit reached
3	1 = Operating cycles limit reached
4	`xFeatFbkDelay` Copy of Feature Bit 1: Feedback delay: • 1 = `MonTiDynamic` (value in %) • 0 = Input parameter `FbkDelay`
5	Copy of `CmData.xSimOnParent`
6	Copy of `SimFbkOpen.Value`
7	Copy of `SimFbkClose.Value`
8	Reserve
...	...

Configurable reactions using the Feature parameter

The following reactions are available for this block:

Bit	Function
0	Display of the simulation: 1 = Common display of the local and higher-level simulation
1	Feedback delay at unit simulation operation: • 1 = Percentage value of the CM monitoring time (factor = `MonTiDynamic`) • 0 = Input `FbkDelay`
2	Display of the CM instance name *) 1 = CFC name (process tag)
3	Display of the CM instance name *) 1 = Block name
4	Reserve

Bit	Function
5	Reserve
6	Type of feedback signal `UsrFbk1` • 1 = UsrFbk1 signals error • 0 = UsrFbk1 signals feedback
7	Type of feedback signal `UsrFbk2` • 1 = UsrFbk2 signals error • 0 = UsrFbk2 signals feedback
8	Reserve
...	...
24	Local operating permission: 1 = Activated
...	...

*) It is also possible to have both names together!

Operator permissions

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can change delay time for "Open"
1	1 = Operator can change delay time for "Close"
2	1 = Operator can change operating hours counter limit
3	1 = Operator can change switching cycle counter limit
4	1 = Operator can reset operating hours counter
5	1 = Operator can reset switching cycle counter
6	Reserve
...	...

Forming the signal status for blocks

This adapter block includes the APL standard function 'Form and output signal status for blocks'. The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `OpenAutCM` (depending on `OvwCommands`)
- `CloseAutCM` (depending on `OvwCommands`)
- `OpenRelAut`
- `CloseRelAu`
- `UsrCmd1RelAut`
- `UsrCmd2RelAut`
- `UsrFbk1`
- `UsrFbk2`
- `ModLiOpCM` (depending on `OvwModSel`)

- AutModLiCM (depending on OvwModSel)
- ManModLiCM (depending on OvwModSel)
- RstLi
- OpenAutDelay
- CloseAutDelay

3.3.4.1 Connections of ApVlvL

The general (non-type-dependent) block interfaces are described in the section 'Common connections of the type-specific adapter blocks (Page 64)'. The following section describes the type-specific block interfaces:

Input parameters

Parameter	Description	Type	Initialization
OpenAut	1=Start: Open command in automatic mode	DigVal	
CloseAut	1=Stop: Close command in automatic mode	DigVal	
OpenRelAut	1=Process release for open command	DigVal	
CloseRelAut	1=Process release for close command	DigVAl	
UsrCmd1RelAut	1=Process release for user command 1	DigValFF	
UsrCmd2RelAut	1=Process release for user command 2	DigValFF	
UsrFbk1	Feedback for user command 1 (follows command if not connected)	DigValFF	
UsrFbk2	Feedback for user command 2 (follows command if not connected)	DigValFF	
...	...		
OpenAutDelay	Open delay in automatic mode	AnaValFF	
StopAutDelay	Close delay in automatic mode	AnaValFF	
...	...		

Output parameters

Parameter	Description	Type	Initialization
OpenAutCM	1=Start: Open command in automatic mode	DigVal	
CloseAutCM	1=Stop: Close command in automatic mode	DigVal	
...	...		
SimFbkOpen	To channel driver's "SimPV_In" (FbkOpen)	DigVal	
SimFbkClose	To channel driver's "SimPV_In" (FbkClose)	DigVal	
...	...		
CmCtrl	Control output to start motor	DigVal	
CmAutAct	1=Auto: Automatic mode indicator	DigVal	

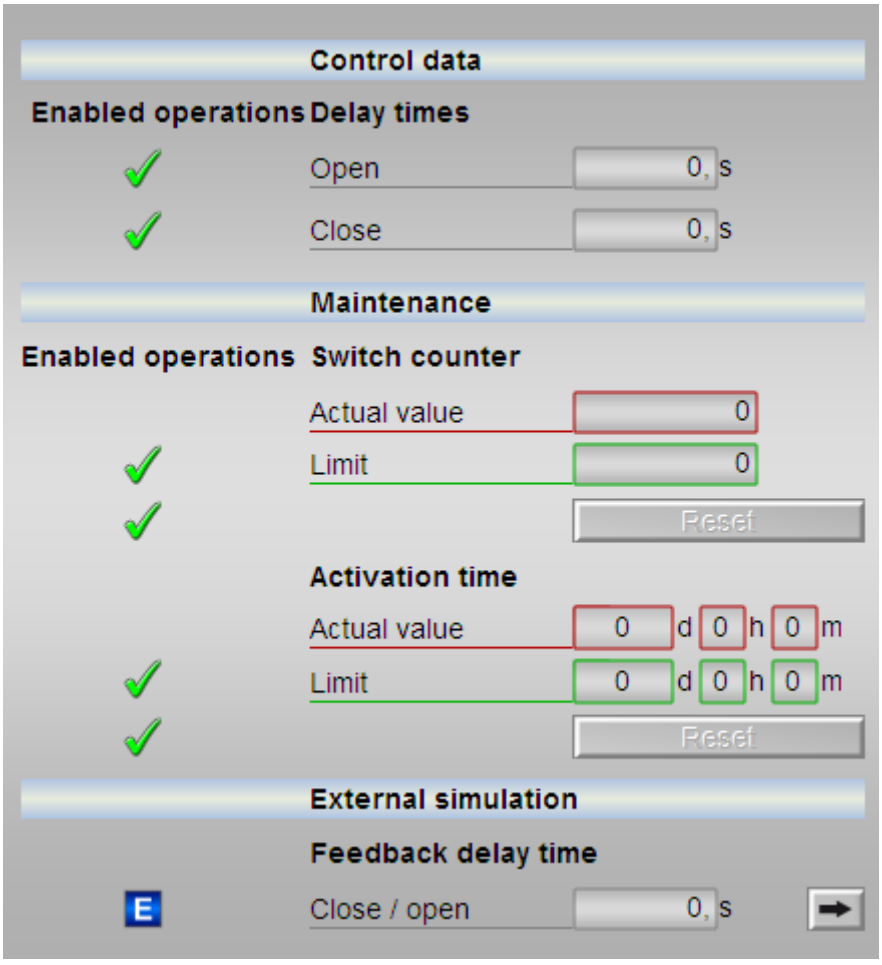
Parameter	Description	Type	Initialization
CmFbkOpen	1=Control valve is opened	DigVal	
CmFbkClose	1=Control valve is closed	DigVal	
...			

3.3.4.2 Operator control and monitoring

For general information about the OS faceplates refer to the section "Operator control and monitoring (Page 77)".

Type-specific information for the adapter block ‚ApVlvL'::

- The designations of the control data are
  'Open' / 'Close'



3.3.5 ApVlvAnL

Mode of operation

This block is used together with the CM block of the APL **‚VlvAnL'** (control module) to control an analog control valve and positioner with adjustable safety position. Various inputs / outputs are available for controlling the valve.

For general information about the configuration and mode of operation see section: Basics of adapter blocks (Page 62)

General adapter functions

This block includes the following general functions of the adapter blocks:

- Faceplate jumps CM block <-> Adapter block (Page 67)
- Command process releases (Page 68)
- ON and OFF delays (Page 69)
- Structure data connections EQM <-> CM (Page 70)
- Enabling for manual switching operation / Block for simulation operation (Page 75)
- Maintenance data (Page 76)

Status word allocation for the output parameter ‚Status1'

- For a description of the individual parameters, see the section 'Common connections of the type-specific adapter blocks (Page 64)'.

Status bit	Parameter
0	Copy of BatchOcc
1	Copy of BatchEn
2	1 = Operating hours limit reached
3	1 = Operating cycles limit reached
4	`xFeatFbkDelay` Copy of Feature Bit 1: Feedback delay: • 1 = `MonTiDynamic` (value in %) • 0 = Input parameter `FbkDelay`
5	Copy of `CmData.xSimOnParent`
6	Copy of `SimFbkOpen.Value`
7	Copy of `SimFbkClose.Value`
8	`Reserve`
...	...

Configurable reactions using the Feature parameter

The following reactions are available for this block:

Bit	Function
0	Display of the simulation: 1 = Common display of the local and higher-level simulation
1	Feedback delay at unit simulation operation: • 1 = Percentage value of the CM monitoring time (factor = `MonTiDynamic`) • 0 = Input `FbkDelay`
2	Display of the CM instance name *) 1 = CFC name (process tag)
3	Display of the CM instance name *) 1 = Block name
4	Reserve
5	Reserve
6	Type of feedback signal `UsrFbk1` • 1 = UsrFbk1 signals error • 0 = UsrFbk1 signals feedback
7	Type of feedback signal `UsrFbk2` • 1 = UsrFbk2 signals error • 0 = UsrFbk2 signals feedback
8	Reserve
...	...
24	Local operating permission: 1 = Activated
...	...

*) It is also possible to have both names together!

Operator permissions

The block has the following authorizations for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can change delay time for "Open"
1	1 = Operator can change delay time for "Close"
2	1 = Operator can change operating hours counter limit
3	1 = Operator can change switching cycle counter limit
4	1 = Operator can reset operating hours counter
5	1 = Operator can reset switching cycle counter
6	`Reserve`
...	...

Forming the signal status for blocks

This adapter block includes the APL standard function 'Form and output signal status for blocks'. The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `OpenAutCM` (depending on `OvwCommands`)
- `CloseAutCM` (depending on `OvwCommands`)
- `OpenRelAut`
- `CloseRelAu`
- `UsrCmd1RelAut`
- `UsrCmd2RelAut`
- `UsrFbk1`
- `UsrFbk2`
- `ModLiOpCM` (depending on `OvwModSel`)
- `AutModLiCM` (depending on `OvwModSel`)
- `ManModLiCM` (depending on `OvwModSel`)
- `RstLi`
- `OpenAutDelay`
- `CloseAutDelay`

3.3.5.1 Connections of ApVlvAnL

The general (non-type-dependent) block interfaces are described in the section 'Common connections of the type-specific adapter blocks (Page 64)'. The following section describes the type-specific block interfaces:

Input parameters

Parameter	Description	Type	Default
`OpenAut`	1=Start: Open command in automatic mode	`DigVal`	
`CloseAut`	1=Stop: Close command in automatic mode	`DigVal`	
`OpenRelAut`	1=Process release for open command	`DigVal`	
`CloseRelAut`	1=Process release for close command	`DigVAl`	
`UsrCmd1RelAut`	1=Process release for user command 1	`DigValFF`	
`UsrCmd2RelAut`	1=Process release for user command 2	`DigValFF`	
`UsrFbk1`	Feedback for user command 1 (follows command if not connected)	`DigValFF`	
`UsrFbk2`	Feedback for user command 2 (follows command if not connected)	`DigValFF`	
...	...		
`OpenAutDelay`	Open delay in automatic mode	`AnaValFF`	

Parameter	Description	Type	Default
StopAutDelay	Close delay in automatic mode	AnaValFF	
...	...		

Output parameters

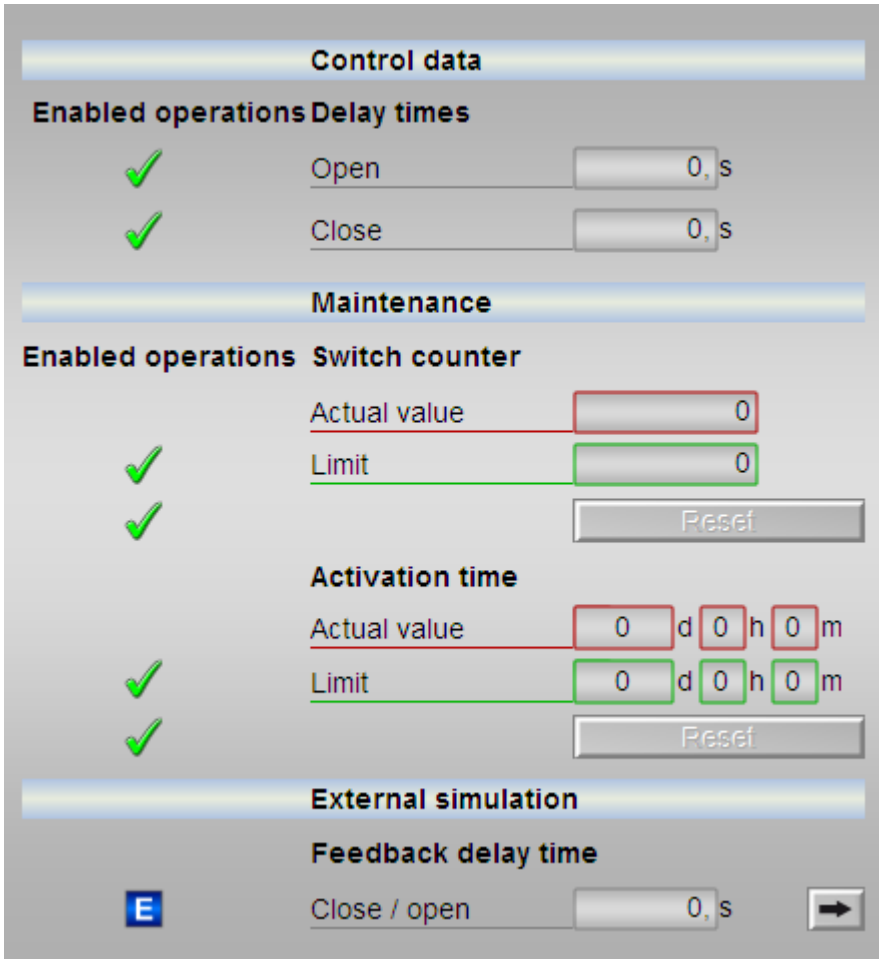
Parameter	Description	Type	Default
OpenAutCM	1=Start: Open command in automatic mode	DigVal	
CloseAutCM	1=Stop: Close command in automatic mode	DigVal	
...	...		
SimFbkOpen	To channel driver's "SimPV_In" (FbkOpen)	DigVal	
SimFbkClose	To channel driver's "SimPV_In" (FbkClose)	DigVal	
...	...		
CmCtrl	Control output to start motor	DigVal	
CmAutAct	1=Auto: Automatic mode indicator	DigVal	
CmFbkOpen	1=Control valve is opened	DigVal	
CmFbkClose	1=Control valve is closed	DigVal	

3.3.5.2 Operator control and monitoring

For general information about the OS faceplates refer to the section "Operator control and monitoring (Page 77)".

Type-specific information for the adapter block AplVlvAnL':

- The designations of the control data are
  'Open' / 'Close'



3.3.6 ApVlvDS

Mode of operation

This block is used together with the block ‚ApVlvDS' (control module) to control double-seat valves. Various inputs / outputs are available for controlling the valve.

For general information about the configuration and mode of operation see section: Basics of adapter blocks (Page 62)

General adapter functions

This block includes the following general functions of the adapter blocks:

- Faceplate jumps CM block <-> Adapter block (Page 67)
- Command process releases (Page 68)
- ON and OFF delays (Page 69)
- Structure data connections EQM <-> CM (Page 70)
- Enabling for manual switching operation / Block for simulation operation (Page 75)
- Maintenance data (Page 76)

Status word allocation for the output parameter ‚Status1'

For a description of the individual parameters, see the section 'Common connections of the type-specific adapter blocks (Page 64)'.

Status bit	Parameter
0	Copy of `BatchOcc`
1	Copy of `BatchEn`
2	1 = Operating hours limit reached
3	1 = Operating cycles limit reached
4	`xFeatFbkDelay` Copy of Feature Bit 1: Feedback delay: • 1 = `MonTiDynamic` (value in %) • 0 = Input parameter `FbkDelay`
5	Copy of `CmData.xSimOnParent`
6	Copy of `SimFbkOpen.Value`
7	Copy of `SimFbkClose.Value`
8	`Reserve`
...	...

Configurable reactions using the Feature parameter

The following reactions are available for this block:

Bit	Function
0	Display of the simulation: 1 = Common display of the local and higher-level simulation
1	Feedback delay at unit simulation operation: • 1 = Percentage value of the CM monitoring time (factor = `MonTiDynamic`) • 0 = Input `FbkDelay`
2	Display of the CM instance name *) 1 = CFC name (process tag)
3	Display of the CM instance name *) 1 = Block name
4	`Reserve`

Bit	Function
5	`Reserve`
6	Type of feedback signal `UsrFbk1` • 1 = UsrFbk1 signals error • 0 = UsrFbk1 signals feedback
7	Type of feedback signal `UsrFbk2` • 1 = UsrFbk2 signals error • 0 = UsrFbk2 signals feedback
8	`Reserve`
...	...
24	Local operator authorization: 1 = enabled
...	...

*) It is also possible to have both names together!

Operator permissions

The block has the following authorizations for the `OS_Perm` parameter:

Bit	Function
0	Reserve
1	Reserve
2	1 = Operator can change operating hours counter limit
3	1 = Operator can change switching cycle counter limit
4	1 = Operator can reset operating hours counter
5	1 = Operator can reset switching cycle counter
6	Reserve
...	...

Forming the signal status for blocks

This adapter block includes the APL standard function 'Form and output signal status for blocks'. The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `OpenAutCM (depending on OvwCommands)`
- `CloseAutCM (depending on OvwCommands)`
- `SLTopCM`
- `SLBtmCM`
- `OpenRelAut`
- `CloseRelAu`
- `SLTopRelAut`
- `SLBtmRelAut`
- `ModLiOpCM (depending on OvwModSel)`

- AutModLiCM (depending on OvwModSel)
- ManModLiCM (depending on OvwModSel)
- RstLi
- OpenAutDelay
- CloseAutDelay

3.3.6.1 Connections of ApVlvDS

The general (non-type-dependent) block interfaces are described in the section 'Common connections of the type-specific adapter blocks (Page 64)'. The following section describes the type-specific block interfaces:

Input parameters

Parameter	Description	Type	Default
OpenAut	1=Start: Open command in automatic mode	DigVal	
CloseAut	1=Stop: Close command in automatic mode	DigVal	
SLTopAut	1=Lift top seat in auto mode	DigVal	
SLBtmAut	1=Lift bottom seat in auto mode	DigVal	
OpenRelAut	1=Process release for open command	DigVal	
CloseRelAut	1=Process release for close command	DigVAl	
SLTopRelAut	1=Process release for lift top seat	DigValFF	
SLBtmRelAut	1=Process release for lift bottom seat	DigValFF	
...	...		

Output parameters

Parameter	Description	Type	Default
OpenAutCM	1=Start: Open command in automatic mode	DigVal	
CloseAutCM	1=Stop: Close command in automatic mode	DigVal	
SLTopAutCM	1=Lift top seat in auto mode	DigVal	
SLBtmAutCM	1=Lift bottom seat in auto mode	DigVal	
...	...		
SimFbkOpen	To channel driver's "SimPV_In" (FbkOpen)	DigVal	
SimFbkClose	To channel driver's "SimPV_In" (FbkClose)	DigVal	
...	...		
CmCtrl	Control output (dependent from SafePos)	DigVal	
CmSLTop	Control output for top seat lifting	DigVal	
CmSLBtm	Control output for bottom seat lifting	DigVal	
CmAutAct	1=Auto: Automatic mode indicator	DigVal	
CmFbkOpen	1=Control valve is opened	DigVal	
CmFbkClose	1=Control valve is closed	DigVal	

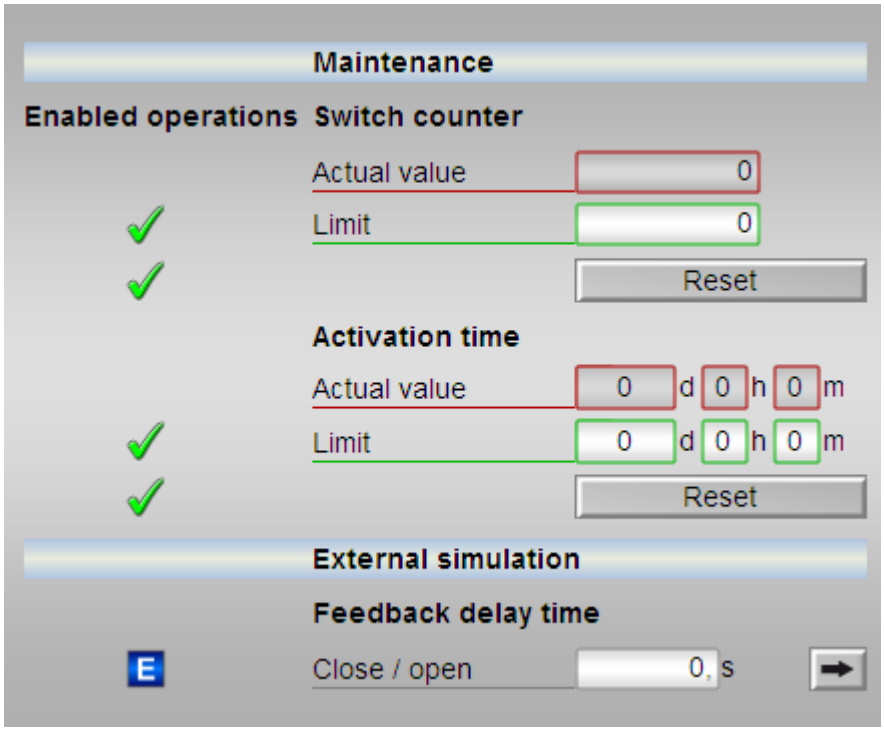
Parameter	Description	Type	Default
CmFbkSLTop	1=Top seat lifting is active	DigVal	
CmFbkSLBtm	1=Bottom seat lifting is active	DigVal	

3.3.6.2 Operator control and monitoring

For general information about the OS faceplates refer to the section "Operator control and monitoring (Page 77)".

Type-specific information for the adapter block ‚ApVlvDsL':

- The section for the ON / OFF delays does not exist at this adapter.
  Reason:
  The ON / OFF delays are already implemented in the CM block 'ApVlvDsL' and can be parameterized in the associated OS faceplate.



3.3.7 ApCmGen

Mode of operation

This block is used together with any drive blocks from other libraries. Up to 4 individually interconnectable control outputs / feedback inputs are available for controlling the drive.

For general information about the configuration and mode of operation see section: Basics of adapter blocks (Page 62)

General adapter functions

This block includes the following general functions of the adapter blocks:

- Faceplate jumps CM block <-> Adapter block (Page 67)
- Command process releases (Page 68)
- ON and OFF delays (Page 69)
- Structure data connections EQM <-> CM (Page 70)
- Enabling for manual switching operation / Block for simulation operation (Page 75)
- Maintenance data (Page 76)

Status word allocation for the output parameter ‚Status1'

For a description of the individual parameters, see the section 'Common connections of the type-specific adapter blocks (Page 64)'.

Status bit	Parameter
0	Copy of `BatchOcc`
1	Copy of `BatchEn`
2	`= false`
3	Copy of `CmData.xSimOnParent`
4	`= false`
5	`Reserve`
...	...

Configurable reactions using the Feature parameter

The following reactions are available for this block:

Bit	Function
0	Display of the simulation: 1 = Common display of the local and higher-level simulation
1	`Reserve`
2	Display of the CM instance name *) 1 = CFC name (process tag)
3	Display of the CM instance name *) 1 = Block name
4	Type of feedback signal `Fbk1` • 1 = Fbk1 signals error • 0 = Fbk1 signals feedback
5	Type of feedback signal `Fbk2` • 1 = Fbk2 signals error • 0 = Fbk2 signals feedback

Bit	Function
6	Type of feedback signal `Fbk3` • 1 = Fbk3 signals error • 0 = Fbk3 signals feedback
7	Type of feedback signal `Fbk4` • 1 = Fbk4 signals error • 0 = Fbk4 signals feedback
8	`Reserve`
9	`Reserve`
10	1 = Rest position command is derived from the parameter `Cmd1Aut`
11	1 = Rest position command is derived from the parameter `Cmd2Aut`
12	1 = Rest position command is derived from the parameter `Cmd3Aut`
13	1 = Rest position command is derived from the parameter `Cmd4Aut`
14	• 0 = Pushbutton operation • 1 = Switch operation
...	...
24	Local operating permission: 1 = Activated
...	...

*) It is also possible to have both names together!

Operator permissions

The block has the following authorizations for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can change delay time `Cmd1`
1	1 = Operator can change delay time `Cmd2`
2	1 = Operator can change delay time `Cmd3`
3	1 = Operator can change delay time `Cmd4`
4	1 = Operator can change operating hours counter limit
5	1 = Operator can change switching cycle counter limit
6	1 = Operator can reset operating hours counter
7	1 = Operator can reset switching cycle counter
8	`Reseve`
...	...

Forming the signal status for blocks

This adapter block includes the APL standard function 'Form and output signal status for blocks'. The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `Cmd1AutCM … Cmd4AutCM (depending on parameter OvwCommands)`
- `Cmd1RelAut … Cmd4RelAut`

- Fbk1 … Fbk4
- ModLiOpCM (depending on parameter OvwModSel)
- AutModLiCM (depending on parameter OvwModSel)
- ManModLiCM (depending on parameter OvwModSel)
- RstLi

3.3.7.1 Connections of ApCmGen

The general (non-type-dependent) block interfaces are described in the section 'Common connections of the type-specific adapter blocks (Page 64)'. The following section describes the type-specific block interfaces:

Input parameters

Parameter	Description	Type	Default
Cmd1Aut	1=Activate command 1	DigVal	
Cmd2Aut	1=Activate command 2	DigVal	
Cmd3Aut	1=Activate command 3	DigVal	
Cmd4Aut	1=Activate command 4	DigVal	
Cmd1RelAut	Process release for command 1 (implicitly true if not connected)	DigValFF	
Cmd2RelAut	Process release for command 2 (implicitly true if not connected)	DigValFF	
Cmd3RelAut	Process release for command 3 (implicitly true if not connected)	DigValFF	
Cmd4RelAut	Process release for command 4 (implicitly true if not connected)	DigValFF	
Fbk1	Command 1 feedback	DigValFF	
Fbk2	Command 2 feedback	DigValFF	
Fbk3	Command 3 feedback	DigValFF	
Fbk4	Command 4 feedback	DigValFF	
ModLiOp	1=Link/Auto, 0=Manual: Input to auto/manual commands	DigVal	
AutModLi	1=Auto mode: auto mode switch via link	DigVal	
ManModLi	1=Manual mode: manual mode switch via link	DigVal	
RstLi	Linkable reset signal	DigVal	
IsActive	Signals activation status (feeds operating hours and cycle)		
FbSimOn	Feedback: Simulation is on	BOOL	
FbOos	Feedback: Out of service	BOOL	
FbAutAct	Feedback: Automatic mode	BOOL	
FbLocalAct	Feedback: Local mode	BOOL	
FbGrpError	Feedback: Group error	BOOL	
FbLockAct	Feedback: Lock is active	BOOL	
FbForceAct	Feedback: Force is active	BOOL	
Status1CM	Connect to a boolean output value of your CM	BOOL	
Cmd1AutDelay	Cmd1 delay in automatic mode	REAL	

Parameter	Description	Type	Default
Cmd2AutDelay	Cmd2 delay in automatic mode	REAL	
Cmd3AutDelay	Cmd3 delay in automatic mode	REAL	
Cmd4AutDelay	Cmd4 delay in automatic mode	REAL	
Fbk1Delay	Cmd1 feedback delay during higher-ranking simulation	REAL	1.0
Fbk2Delay	Cmd2 feedback delay during higher-ranking simulation	REAL	1.0
Fbk3Delay	Cmd3 feedback delay during higher-ranking simulation	REAL	1.0
Fbk4Delay	Cmd4 feedback delay during higher-ranking simulation	REAL	1.0
CmData	Control and status information for this CM	BlUdtCmData	
CmCmd	Command data from FxCtrl	DWORD	
OvwCommands	1=Overwrite CmCmd commands with inputs OpenAut/CloseAut	BOOL	True
OvwModSel	1=Overwrite CmData mode switches	BOOL	
OvwUnitID	1=Overwrite CmData unit ID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
OvwManOpEn	1=Overwrite CmData ManOpEn with input ManOpEn	BOOL	
UnitID	Unit ID (usually from UnitIf block)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIDLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
OpHoursAct	Operating hours actual value [sec]	DINT	
OpCycleAct	Operating cycle actual value	DINT	
OpHoursLim	Operating hours limit	DINT	
OpCycleLim	Operating cycle limit	DINT	
OpHoursRst	Reset operating hours	BOOL	
OpCycleRst	Reset operating cycle	BOOL	
SampleTime	Sampling time [s]	REAL	1.0
SelFp1	Select faceplate	BOOL	
SelFp2	Select faceplate	BOOL	
OS_Perm	Operator permissions for adapter block - Bit assignments type specific	STRUCT_DWORD	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT_DWORD	

Output parameters

Parameter	Description	Type	Initialization
Cmd1AutCM	1=Cmd1 command in automatic mode	DigVal	
Cmd2AutCM	1=Cmd2 command in automatic mode	DigVal	
Cmd3AutCM	1=Cmd3 command in automatic mode	DigVal	
Cmd4AutCM	1=Cmd4 command in automatic mode	DigVal	

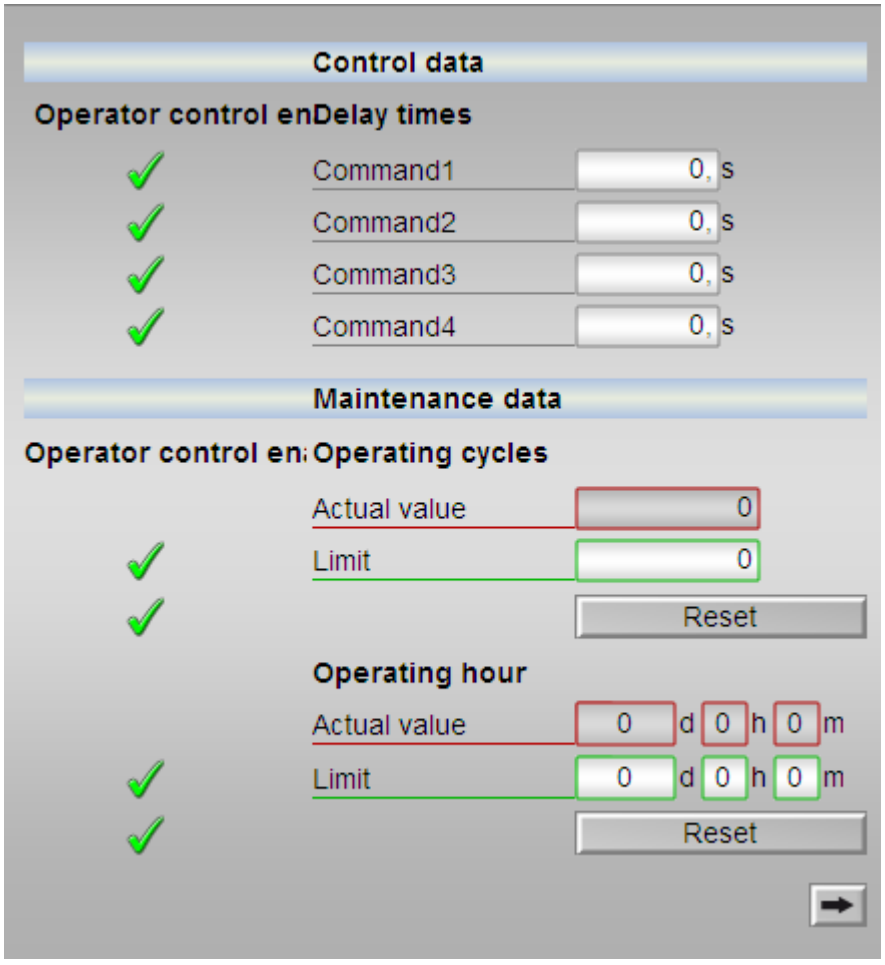
Parameter	Description	Type	Initialization
ModLiOpCM	1=Link/Auto 0=Manual: Input to auto/manual commands	DigVal	
AutModLiCM	1=Auto mode: auto mode switch via link	DigVal	
ManModLiCM	1=Manual mode: manual mode switch via link	DigVal	
RstLiCM	Reset signal	DigVal	
MsgLockCM	Message lock from unit	DigVal	
ManOpEnCM	Enable switchover to manual mode (routed CmData signal)	BOOL	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING [32]	
StepNo	Batch Stepp number	DWORD	
OpHoursActOut	Operating hours actual value	DINT	
OpCycleActOut	Operating cycle actual value	DINT	
OpHoursHit	Operating hours limit has been reached	DigVal	
OpCycleHit	Operating cycle limit has been reached	DigVal	
UnitID_CM	Connect to CM's ExtVa104 for unit message filtering	DWORD	
EqmLock	EQM lock signal from CmData	BOOL	
P_RST	Reset pulse	BOOL	
Fbk1Out	Command 1 process feedback	DigVal	
Fbk2Out	Command 2 process feedback	DigVal	
Fbk3Out	Command 3 process feedback	DigVal	
Fbk4Out	Command 4 process feedback	DigVal	
SimOn	To channel driver's SimOn	DigVal	
SimFbk1	To channel driver's SimPV_In	DigVal	
SimFbk2	To channel driver's SimPV_In	DigVal	
SimFbk3	To channel driver's SimPV_In	DigVal	
SimFbk4	To channel driver's SimPV_In	DigVal	
OS_PermOut	Operator permission: output 1 for OS	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
OS_PermLnk	Operator permission (OS info)	DWORD	
OpSt_Out	Enabled operator stations	DWORD	
Status1	Status 1	DWORD	
St_Worst	Worst signal status	BYTE	16#80
CMFbk	Feedback data for EqmIf / FxCtrl	BlUdtCmFbk	

3.3.7.2 Operator control and monitoring

For general information about the OS faceplates refer to the section "Operator control and monitoring (Page 77)".

Type-specific information for the adapter block ‚ApCmGen':

- The designations of the control data are:
  'Command 1 to Command 4'



3.3.8 ApPIDConL

Mode of operation

This block is used together with the controller block of the APL **‚PidConL'** or **‚PidConR'** to structure controller process tags. In addition to the unit-specific data supply and command propagation, various inputs / outputs for supplying with up to 4 tracking values are available.

For general information about the configuration and mode of operation see section: Basics of adapter blocks (Page 62)

General adapter functions

This block includes the following general functions of the adapter blocks:

- Faceplate jumps CM block <-> Adapter block (Page 67)
- Higher-level controlling of the unit / EQM level (Page 70)
- Enabling for manual switching operation / Block for simulation operation (Page 75)

Status word allocation for the output parameter ‚Status1'

- For a description of the individual parameters, see the section 'Common connections of the type-specific adapter blocks (Page 64)'.

Status bit	Parameter
0	BatchOcc
1	BatchEn
	…
10	OvwCommands
11	OvwModSel
12	OvwUnitId
13	OvwBatchInfo
14	OvwManOpEn
15	xManOpEn
16	CmData.xSimEnCM
17	CmData.xEQMLock
18	CmData.xMsgLock
21	ExtTrkOn
22	Trk1Mode
23	Trk2Mode
24	Trk3Mode
25	Trk4Mode
26	Trk1OnOut
27	Trk2OnOut
28	Trk3OnOut
29	Trk4OnOut
30	Reserve
31	…

Forming the signal status for blocks

This adapter block includes the APL standard function 'Form and output signal status for blocks'. The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `Cmd1CM … Cmd4CM (depending on parameter OvwCommands)`
- `Cmd1Rel … Cmd4Rel`
- `Fbk1 … Fbk4`

- ModLiOpCM (depending on parameter OvwModSel)
- AutModLiCM (depending on parameter OvwModSel)
- ManModLiCM (abhängig von Parameter OvwModSel)
- RstLi

See also

Command process releases (Page 68)

ON and OFF delays (Page 69)

Structure data connections EQM <-> CM (Page 70)

Maintenance data (Page 76)

3.3.8.1 Functions of 'ApPIDConL'

The block makes a tracking value that can be derived optionally from 4 input parameters Trk1Val–Trk4Val with time-controlled criteria or directly from the input ExtTrkVal available for the APL controller block PIDConL (input MV_Trk) at the output TrkValAct. Selection and activation of the desired input parameter is made by a changeover signal ExtTrkOn as well as by four start inputs Trk1On … Trk4On and four mode inputs Trk1Mode … Trk4Mode.

Note

The block I/O names for tracking time / tracking values are transferred to the OS faceplate / simulation view during OS compilation.

For the complete PidConL/PidConR functionality, in particular the behavior with regard to manual, automatic and follow-up mode as well as cascading → see APL documentation of the PidConL/PidConR block

External / Internal selection using the switch ExtTrkOn

- ExtTrkOn = 1
  The input parameter ExtTrkVal is written to the output TrkValAct.
- ExtTrkOn = 0
  The input parameters Trk1Val … Trk4Val are enabled

Activation through the inputs Trk1On … Trk4On

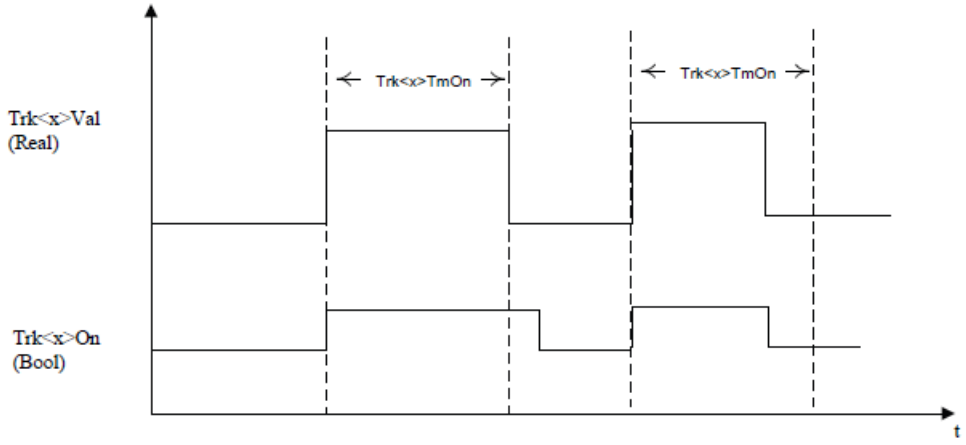
If ExtTrkOn = 0, one of the enabled values Trk1Val … Trk4Val can be switched through with the inputs Trk1On … Trk4On = 1 to the output TrkValAct. If several inputs are activated, that with the lowest ordinal number (1 to 4) "wins".

Permanent / Pulse Mode selection through the inputs Trk1Mode ... Trk4Mode

The Trk1Val ... Trk4Val values can be set for a specific period Trk1TmOn ... Trk4TmOn or for as long as the binary signal Trk1On ... Trk4On = 0. This selection is determined by Trk1Mode ... Trk4Mode:

- Trk<x>Mode = 'Permanent' → The tracking value Trk<x>Val is switched through via Trk<x>On directly to the output TrkValAct
- Trk<x>Mode = 'Pulse' → The tracking value Trk<x>Val is started by Trk<x>On and switched through for the configured duration at the input parameter Trk<x>TmOn to the output TrkValAct.
- If several signal inputs Trk1On ... Trk4On are active, the input with the lowest identification number (1 to 4) "wins".

The following diagram shows the time-based correlations:



Example for Trk<x>Mode=0

- The time that has currently expired is output at the output TrkTmAct and the active tracking value at the output TrkValAct. The latter has to be interconnected with the input MV_Trk of the controller block PidConL/PidConR.
- The activation output TrkOn has to be interconnected with the PidCon input MV_TrkOn.
- One of the outputs Trk1OnOut ... Trk4OnOut is set while the corresponding tracking value is active.

Configurable reactions using the Feature parameter

The adapter block BlPidConL includes the input with the designation "Feature". This input can be used to influence the way in which the block works.

Bit	Function
0	Reserve
1	Reserve
2	Display of the CM instance name *) 1 = CFC name (process tag = default value)

Bit	Function
3	Display of the CM instance name *) 1 = Block name
4	Type of the feedback signal `Fbk1` 1 = Fbk1 signals error, 0 = Fbk1 signals feedback
5	Type of the feedback signal `Fbk2` 1 = Fbk2 signals error, 0 = Fbk2 signals feedback
6	Type of the feedback signal `Fbk3` 1 = Fbk3 signals error, 0 = Fbk3 signals feedback
7	Type of the feedback signal Fbk4 1 = Fbk4 signals error, 0 = Fbk4 signals feedback
8	1 = Fbk1 follows Cmd1 if not interconnected
9	1 = Fbk2 follows Cmd2 if not interconnected
10	1 = Fbk3 follows Cmd3 if not interconnected
11	1 = Fbk4 follows Cmd4 if not interconnected
12	Activate 'External setpoint' at change to AUTO mode
13	Activate 'Internal setpoint' at change to MANUAL mode
...	`Reserve`
24	Enable local operating permission: 1 = Activated
...	...

*) It is also possible to have both names together!

Operator permissions

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can edit tracking value 1
1	1 = Operator can edit tracking value 2
2	1 = Operator can edit tracking value 3
3	1 = Operator can edit tracking value 4
4	1 = Operator can edit activation time for tracking value 1
5	1 = Operator can edit activation time for tracking value 2
6	1 = Operator can edit activation time for tracking value 3
7	1 = Operator can edit activation time for tracking value 4
8	1 = Operator can edit external tracking value
9	Reserve
...	...

3.3.8.2 Connections of 'ApPIDConL'

The general (non-type-dependent) block interfaces are described in the section 'Common connections of the type-specific adapter blocks (Page 64)'. The following section describes the type-specific block interfaces:

Input parameters

Parameter	Description	Type	Default
Cmd1	Custom command	DigVal	
Cmd2	Custom command	DigVal	
Cmd3	Custom command	DigVal	
Cmd4	Custom command	DigVal	
Cmd1Rel	1 = Process release for user command 1	DigValFF	
Cmd2Rel	1 = Process release for user command 2	DigValFF	
Cmd3Rel	1 = Process release for user command 3	DigValFF	
Cmd4Rel	1 = Process release for user command 4	DigValFF	
Fbk1	Custom feedback	DigValFF	
Fbk2	Custom feedback	DigValFF	
Fbk3	Custom feedback	DigValFF	
Fbk4	Custom feedback	DigValFF	
ExtTrkOn	Enable external tracking value 'ExtTrkVal'	DigVal	
ExtTrkVal	External tracking value	AnaValFF	
Trk1Val	Tracking value 1	AnaValFF	
Trk1TmOn	Activation time for tracking value 1	AnaValFF	
Trk1Mode	Operating mode: • 1=time triggered • 0=permanent	BOOL	
Trk1On	Activate Trk1Val	DigVal	
Trk2Val	Tracking value 2	AnaValFF	
Trk2TmOn	Activation time for tracking value 2	AnaValFF	
Trk2Mode	Operating mode: • 1=time triggered • 0=permanent	BOOL	
Trk2On	Activate Trk2Val	DigVal	
Trk3Val	Tracking value 3	AnaValFF	
Trk3TmOn	Activation time for tracking value 3	AnaValFF	
Trk3Mode	Operating mode: • 1=time triggered • 0=permanent	BOOL	
Trk3On	Activate Trk3Val	DigVal	
Trk4Val	Tracking value 4	AnaValFF	
Trk4TmOn	Activation time for tracking value 4	AnaValFF	
Trk4Mode	Operating mode: • 1 = time triggered • 0=permanent	BOOL	
Trk4On	Activate Trk4Val	DigVal	
...			

Output parameters

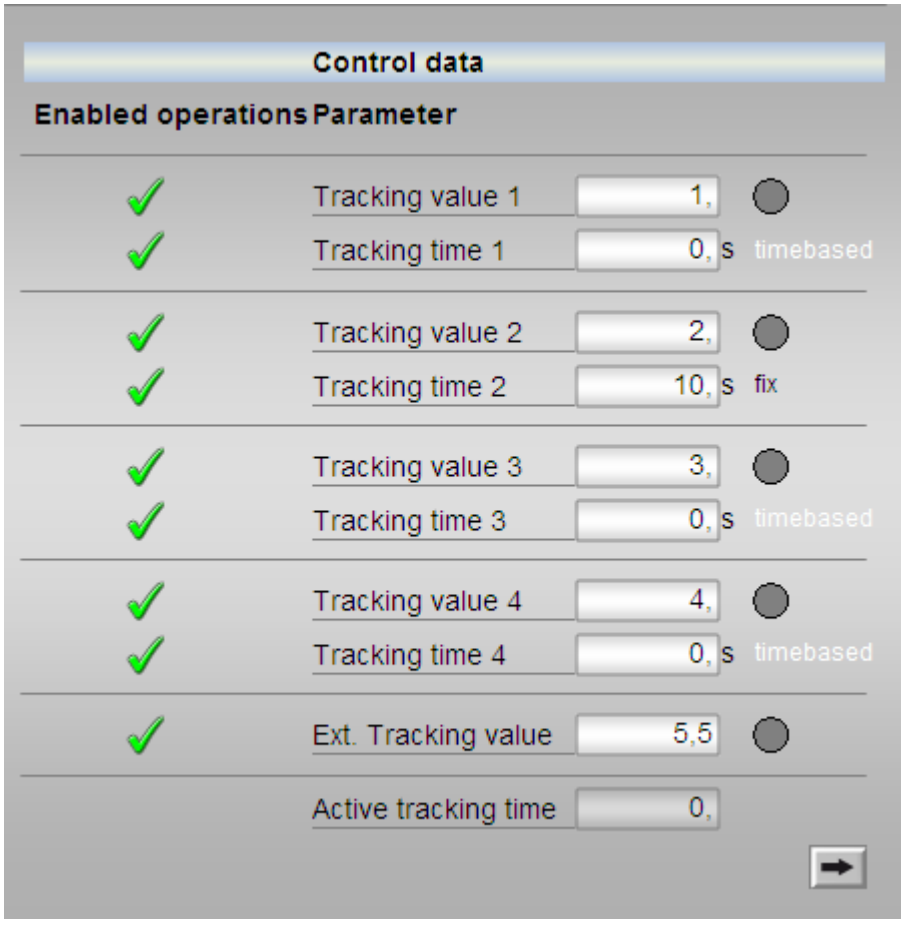
Parameter	Description	Type	Default
SP_LiOpCM	Select setpoint source • 1 = interconnection • 0 = operator	DigVal	
SP_ExtLiCM	Select external setpoint via interconnection	DigVal	
SP_IntLiCM	Select internal setpoint via interconnection	DigVal	
Trk1OnOut	Tracking value 1 is active	BOOL	
Trk2OnOut	Tracking value 2 is active	BOOL	
Trk3OnOut	Tracking value 3 is active	BOOL	
Trk4OnOut	Tracking value 4 is active	BOOL	
TrkOn	Tracking is on	DigVal	
TrkTmAct	Current tracking value's activation time	AnaVal	
TrkValAct	Current tracking value	AnaVal	
...	...	...	
Fbk1Out	Command 1 process feedback	DigVal	
Fbk2Out	Command 2 process feedback	DigVal	
Fbk3Out	Command 3 process feedback	DigVal	
Fbk4Out	Command 4 process feedback	DigVal	
Cmd1CM	Custom command	DigVal	
Cmd2CM	Custom command	DigVal	
Cmd3CM	Custom command	DigVal	
Cmd4CM	Custom command	DigVal	
...	...	...	

3.3.8.3 Operator control and monitoring

Operator control and monitoring of the adapter block is carried out via an OS faceplate that makes the following views available

Symbol	View	Information
	Standard view	See below.
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

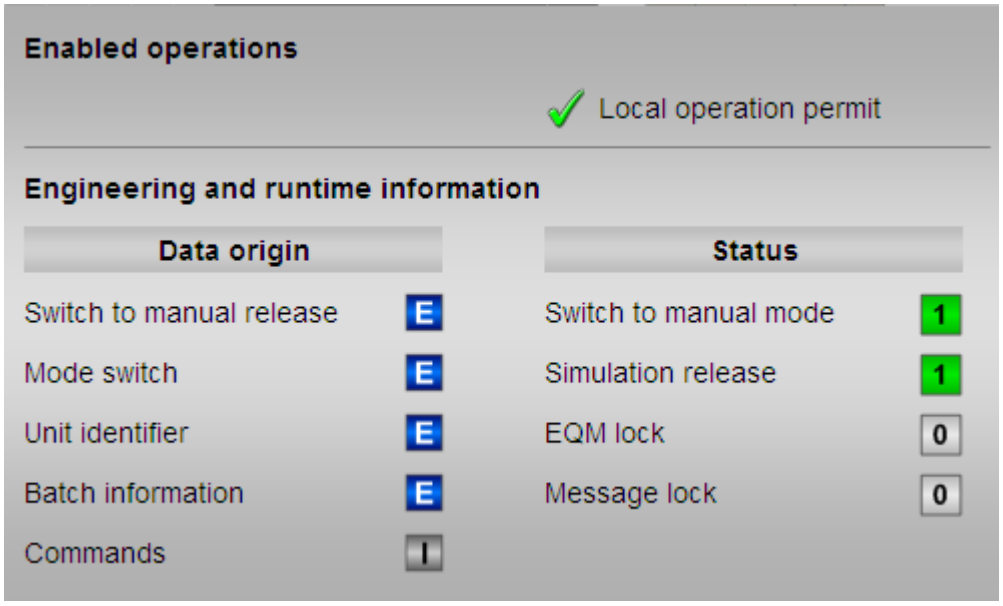
Standard view of the ApPIDConL



Explanation of the screen shot		
(1)	**Displaying and changing the internal tracking values 1 to 4**	
	The values and times of the 4 possible internal tracking values as well as of the external one are displayed in this section. If there is an operator permission, the tracking values / internal times can be changed. The following information is displayed in addition to the values:	
		Tracking value active
		Tracking value not active
	Permanent	Internal tracking value only controlled by `Trk1On` ... `Trk4On`
	Time-based	Internal tracking value started by `Trk1On` ... `Trk4On` and active on `Trk1TmOn` ... `Trk4TmOn` for the configured time
(2)	**Active tracking time**	
	In the case of time-controlled activation of a tracking value the currently expired time (output `TrkTmAct`) is displayed here.	

Explanation of the screen shot	
(3)	**"Next" button**
	The button is used to call up the faceplate of the controller block of the APL library ‚PidConL/PidConS' that is connected with the adapter block.

Preview view of the ApPIDConL



Explanation of the screen shot	
(1)	**Display and switchover of the operating mode**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration in the engineering system (ES) that applies to this block (parameter `OS_PERM`). • Symbols for operator control enable: – Green check mark The OS operator may control this parameter: – Gray check mark The OS operator cannot control this parameter at this time due to the process – Red cross The OS operator cannot control this parameter due to the configured operator permissions • The following operator control enables are shown here: Local operator permission": You can switch to the "Local operation" operating mode.

Explanation of the screen shot			
(2)	**Engineering and runtime information**		
	• The "Data origin" column shows the setting of the overwrite signals:		
		Enable manual mode	`OvwManOpEn`
		Mode switch	`OvwModSel`
		Unit identifier	`OvwUnitID`
		Batch information	`OvwBatchInfo`
		Control	`OvwCommands`
		• The symbols have the following meaning	
			Signal origin = Unit / `CmData` structure
			Signal origin = Block input `Ovw<xxx>`
	• The 'Engineering and runtime information' column shows the following state bits (0 / 1):		
		Enable manual mode	`ManOpEn` or `CmData.xManOpEn` depending on `OvwManOpEn`
		Simulation release	`CmData.xSimEnCM`
		EQM lock	`CmData.xEqmLock`
		Message lock	`CmData.xMsgLock`

APC - Connecting higher-level controllers

4

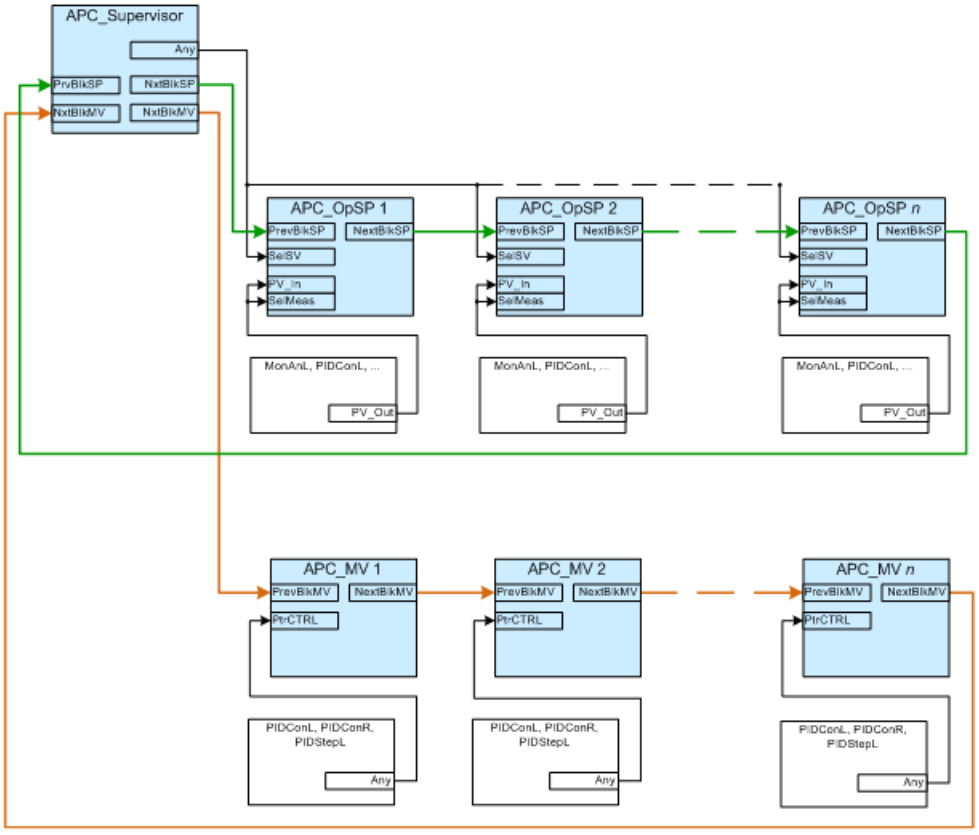
4.1 Description of APC coupling blocks

Area of application

The APC (Advanced Process Control) coupling blocks are used to connect external, higher-level controllers (referred to as Advanced Controller or AC in the following sections) to a PCS 7 process control system.
They allow the operation of external, higher-level controllers to be integrated into the control system to a large extent.
There are three different APC coupling blocks, which together represent the AC in the automation system. These have to be interconnected as described below.

Configuration

APC_Supervisor, APC_OpSP and APC_MV blocks must be interconnected as shown in the figure below. The functions of the interconnections are described in the following sections.



APC_MV and APC_OpSP loop

APC_MV and APC_OpSP blocks read data directly from the APC_Supervisor. To this end, the data block address of the APC_Supervisors is passed to each lower-level APC_MV and APC_OpSP with a separate structure. The structure inputs and outputs `PrvBlkSP/NxtBlkSP` or `PrvBlkMV/NxtBlkMV` are used for this purpose.

The structures can be used simultaneously to report specific data back to the APC_Supervisor. Therefore, the APC_MV and APC_OpSP blocks must each be installed in a loop.

The APC_OpSP structure contains the following data:

- Supervisor data block number
- Test bit to check if the loop is being processed (e.g. for interconnection across AS)
- Control signals `AdvCoEn, AdvCoMsterOn, ActiveAC, ActiveAC_Status, SPConnOk, MVConnOk for visualization and control of the APC_OpSP` blocks
- Number of APC_OpSP blocks (structure element .NumOpSP)
  ⇒ Each APC_OpSP in the loop increments .NumOpSP by one and then forwards the value. The number of APC_OpSP blocks is therefore available at the end of the loop.
- Number of APC_OpSP blocks whose status is 'OK' (structure element .NumOpSP_PVOk)
  ⇒ Each APC_OpSP in the loop increments .NumOpSP_PVOk by one and then forwards the value, if the status of the process value is 16#80 or 16#60. The number of 'good' process values is therefore available at the end of the loop.
- Worst signal status (structure element .ST_Worst)
  ⇒ Each APC_OpSP in the loops checks if its own worst signal status is worse than .ST_Worst from the input structure. The worst signal status is forwarded. The worst signal status of all APC_SpOPs is therefore available at the end of the loop.

The signals in the APC_MV structure are formed according to the same principle. The structure includes the following data:

- Supervisor data block number
- Test bit to check if the loop is being processed (e.g. for interconnection across AS)
- Control signals `AdvCoEn, AdvCoMsterOn, ActiveAC, ActiveAC_Status, SPConnOk, MVConnOk` for control of the APC_MV blocks
- Number of APC_MV blocks (structure element .NumMV)
- Number of APC_MV blocks that are ready for 'advanced control' (structure element .NumMV_AdvCoRdy)
- Number of APC_MV blocks for which 'advanced control' is active (structure element .NumMV_AdvCoAct)
- Worst signal status (structure element .ST_Worst)

The APC_Supervisor block verifies whether the loops are closed by checking whether its own data block address is returned at the end of the loop.

The APC_OpSP and APC_MV loops are used by the APC_Supervisor faceplate to generate the 'Process tag list view'. The principle used means that the last block in the loop is at the top of the list and the first block of the loop is shown at the bottom of the list. Up to 100 blocks per loop are supported in the list view.

Run sequence of APC_MV and APC_OpSP blocks

To ensure that the values described above arrive at the APC_Supervisor as fast as possible, attention must be paid to the run sequence of the APC_OpSP and APC_MV blocks. This is particularly worth mentioning since the APC coupling blocks do not require the respective APC_OpSP or APC_MV block to be installed in the same CFC as the APC_Supervisor. They can also be located at the corresponding process tag or the corresponding controller, for example.

APC_OpSP inputs

In addition to the usual SelFP inputs in APL, the APC_OpSP block has two other Select inputs for opening other faceplates:

- `SelSV` (SelectSupervisor):
  When this I/O is interconnected, the standard view of the faceplate shows a button which can be used to open the faceplate of the supervisor block.
- `SelMeas` (SelectMeasurement)
  When this I/O is interconnected, the process tag view of the supervisor shows a button next to the line of APC_OpSP, which you can use to open the faceplate of the respective process tag directly from the process tag list view of the supervisor.

APC_MV inputs

If APC_MV is used in conjunction with a known controller block (PIDConL, PIDConR, PIDStepL), only the '`PtrCTRL`' input of APC_MV must be interconnected with any output of the controller and the appropriate controller type must be set at the 'BlockType' input of APC_MV.

If APC_MV is used in conjunction with an unknown block, the 'PtrCTRL' input only serves to open a faceplate from the process tag list view (Select faceplate function). All values required for the AC must then be interconnected to the I/Os of APC_MV or configured at the I/Os.

Placing and naming APC_MV and APC_OpSP blocks

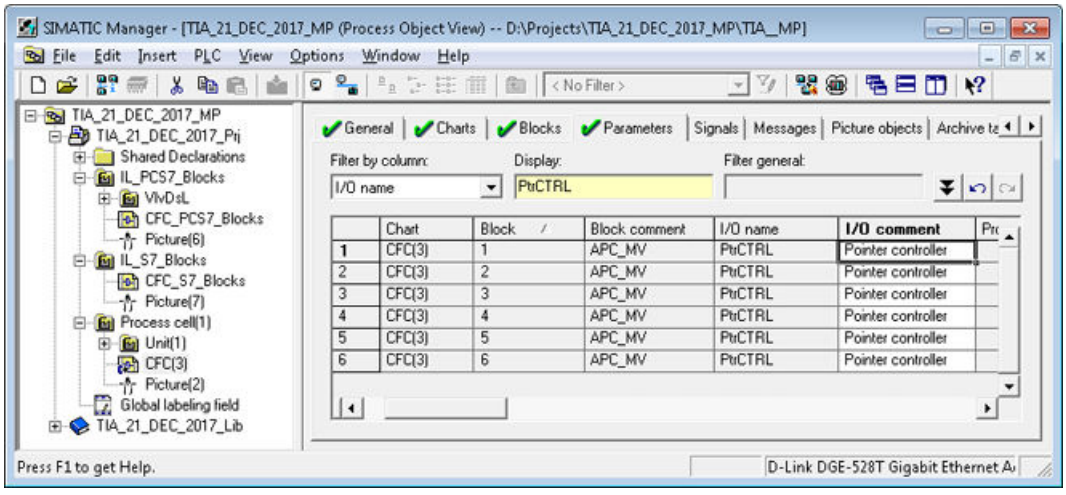
There are two ways to establish the APC coupling using the APC coupling blocks.

Existing plant

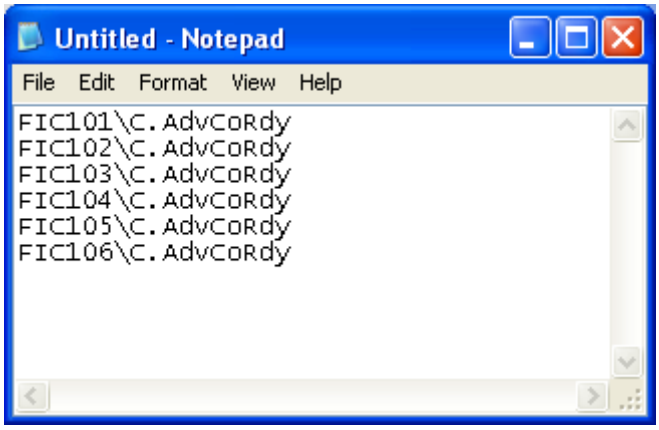
In an existing plant, you can create a CFC that represents the external control in the AS. This has the advantage that you do not have to make any changes to the existing process tags.
To connect an external controller to this CFC, however, you need to ensure that you can assign the APC_OpSP and APC_MV blocks involved to the corresponding process tags. Block names can be used to accomplish this. If you name the blocks to correspond to their process tag and their function (e.g. TI101_SP for an APC_OpSP block that is connected to the process tag TI101 and FIC105_MV for an APC_MV block that is connected to the process tag FIC105), the CFC name and the block name provide the OS with clear criteria which you can use to filter the variables for the AC coupling.

You can edit the block names manually in a small-scale project. For a large-scale project, you can use the process object view. The procedure is described below using the APC_MV blocks as an example:

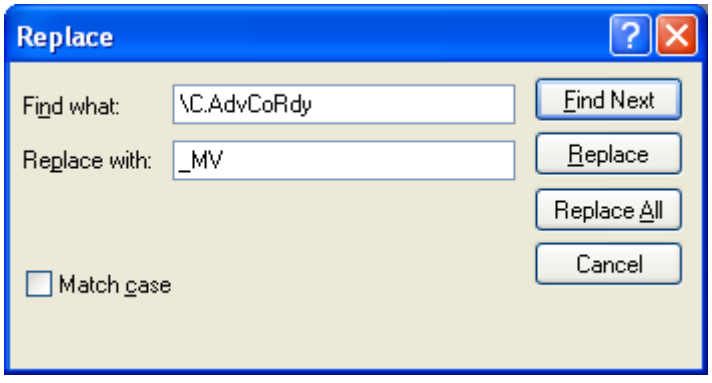
1. The APC_MV block is connected to the process tag via the `PtrCTRL` I/O. This should be identified as a parameter.
2. If the PtrCTRL I/Os are identified as parameters, they can be filtered in the 'Parameter' tab of the process object view.



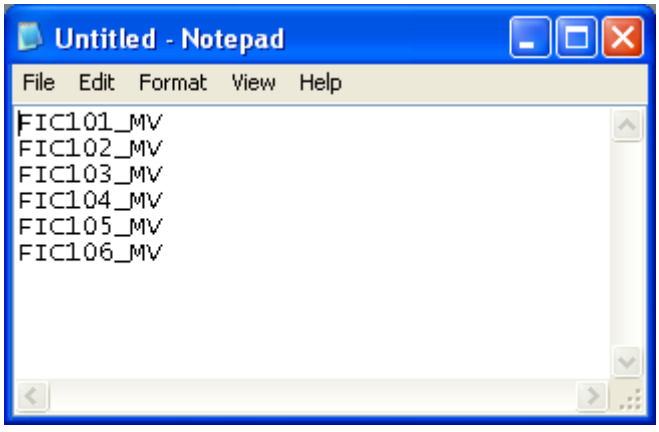
3. The list must be sorted by block name. Now you can copy the connection names to another application for editing (for example, Excel, Notepad, Write, etc.).



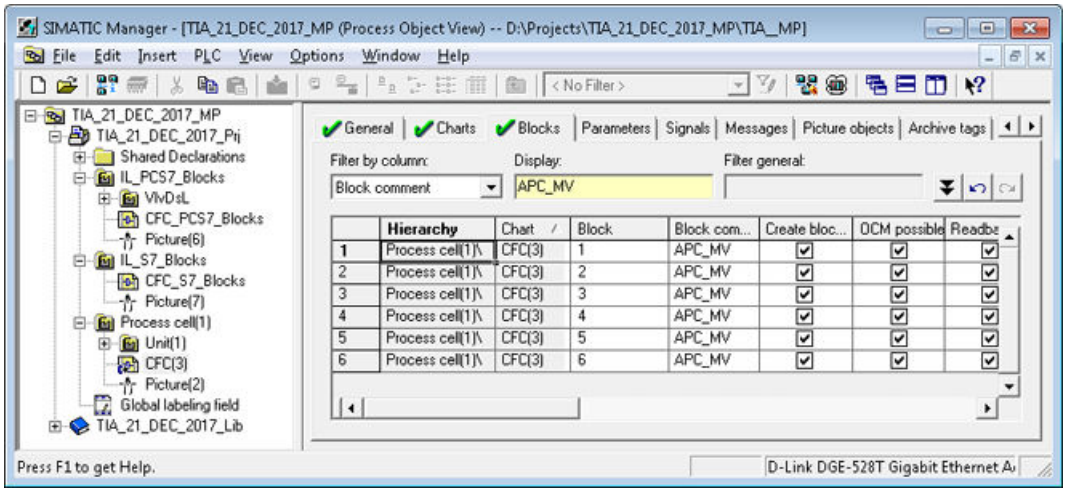
⇓



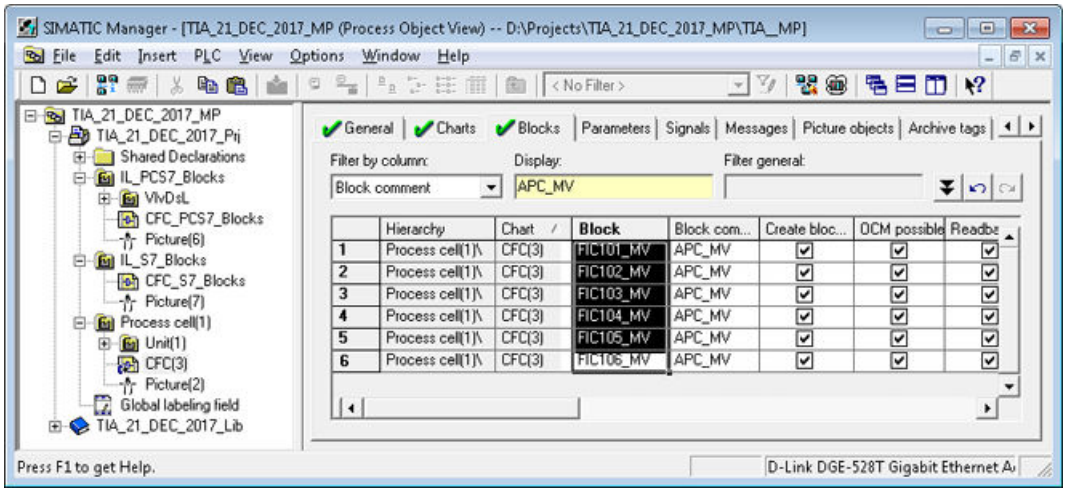
⇓



4. Once you are finished editing the list, open the 'Blocks' tab, filter by block type 'APC_MV' and sort the list by block name.



5. The edited list can now be copied into the block names column.

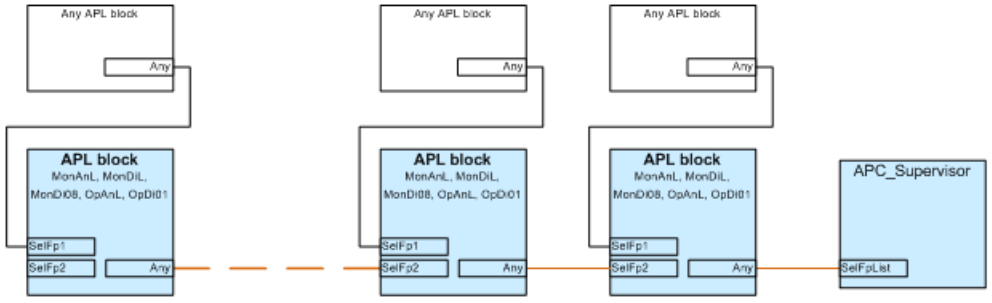


New plant with 'AC process tags'

If a new plant is being created, you can directly define the 'AC process tag types' in which the APC_OpSP and APC_MV blocks are already placed. In this case, you should make sure that the block names which you select for the APC_OpSP and APC_MV blocks distinguish them clearly from other blocks of the CFC (e.g. AC_SP for APC_OpSP and AC_MV for APC_MV). This results in chart names and block names that provide a clear criterion for filtering the tags on the OS.

Process tag list view

Process tags that influence the controller / process model can be displayed in a special view. The process tags displayed in the list serve to inform the operator but do not have any impact on the function of the APC coupling blocks.



The process tag list can only be assembled in connection with supported APL blocks. The following blocks are supported in the list view:

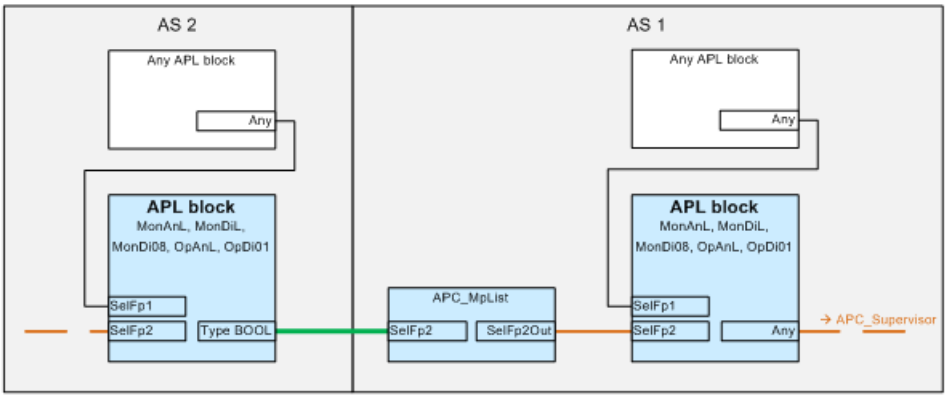
- OpDi01
- OpAnL
- MonDiL
- MonDi08
- MonAnL

To create the list, the blocks must be interconnected as shown in the figure above. This means the first block is connected with the`SelFpList` I/O of theAPC_Supervisor block. Each additional block to be displayed in the list is connected to the `SelFp2` connector of the previous block. Blocks that are interconnected with a block displayed in the list by means of the respective `SelFp1` connector are displayed as jump button in the list view. You can use the jump button to open any type of APL block faceplates.
Up to 100 blocks can be displayed in the process tag list.

AS-spanning interconnections in the list view

The controllers of an AC can be distributed across several AS. This means messages relevant for the controller / process model can be distributed across several AS as well.

The mechanism of the list view uses the SelFP I/Os of the APL blocks. These are of the 'Any' data type and can therefore be interconnected across AS limits. You still need to use the APC_MpList block to create process tag lists that span several AS. The block has a `SelFP2` connector of the BOOL type and can therefore be interconnected beyond AS limits.



The figure above shows the AS-spanning interconnection with the APC_MpList block. The list view of the APC_Supervisor block recognizes the APC_MPList block but looks for the next block instead of displaying it in the list.

4.2 APC_Supervisor

4.2.1 APC_Supervisor description

Object name (type + number) and family

Type + number: FB 1180

Family: APC

Number of process objects: 10 PO

Calling OBs

APC_Supervisor

- Cyclic interrupt OB in which the block is installed (e.g., OB32)
- OB100 (see Startup characteristics)

Startup characteristics

Based on `Feature.Bit0`, old values are retained when the block starts, or the block starts with the most recently saved values.

`RunUpCyc` indicates the number of cycles for which new alarm messages are suppressed at block startup.

Called blocks

APC_Supervisor

FC369	SelST16
SFB35	ALARM_8P
SFC6	RD_SINFO
SFC20	BLKMOV

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section APC_Supervisor I/Os (Page 140).

Status bit	Parameter
0	Occupied
1	BatchEn
2	CtrlAct.Value
3	PredAct.Value

Status bit	Parameter
4	OosAct.Value
5	OosLi.Value
6	A_Alive.Value
7	B_Alive.Value
8	A_Act.Value
9	B_Act.Value
10	Feature RedundantAC
11	AvCoRdy.Value
12	Automatic AC switchover On
13	Automatic AC switchover Off
14	Control error
15	A_Mode
16	B_Mode
17	AC not ready
18	Lower-level controller not ready (MV not ready)
19	Measurements not OK (PV not OK)
20 - 31	Not used

Status word assignment for the parameter `Status2`

For a description of the individual parameters, refer to the section APC_Supervisor I/Os (Page 140).

Status bit	Parameter
0	MsgLock = 1
1 - 31	Not used

Status word assignment for the parameter `Status3`

For a description of the individual parameters, refer to the section APC_Supervisor I/Os (Page 140).

Status bit	Parameter
0 - 29	Not used
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word assignment for the parameter `Status4`

For a description of the individual parameters, refer to the section APC_Supervisor I/Os (Page 140).

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via EventTsIn
1	Effective signal 2 of the message block interconnected via EventTsIn
2	Effective signal 3 of the message block interconnected via EventTsIn
3	Effective signal 4 of the message block interconnected via EventTsIn
4	Effective signal 5 of the message block interconnected via EventTsIn
5	Effective signal 6 of the message block interconnected via EventTsIn
6	Effective signal 7 of the message block interconnected via EventTsIn
7	Effective signal 8 of the message block interconnected via EventTsIn
8 - 31	Not used

4.2.2 APC_Supervisor operating modes

APC_Supervisor operating modes

The block can be operated using the following modes:

- "Out of service"
- "Prediction"
- "Rules"

The next section provides additional block-specific information relating to the general descriptions.

"Out of service"

You can find general information about the "out of service" mode in the "Out of service" section in the APL manual

"Prediction"

The "Prediction" operating mode deactivates the higher-level controller (Prediction mode). The lower-level controllers are enabled and are not in Program mode.

"Rules"

The "Rules" operating mode activates the higher-level controller (Control mode) and switches the lower-level controllers to the Program mode.

4.2.3 APC_Supervisor functions

The APC_Supervisor is the main block for AC coupling. It monitors the external application(s) and communication with the external application(s), and displays all the process tags involved in the control process at the PCS 7 end in a separate faceplate view. Additional selected process tags can be displayed in another view.
The APC_Supervisor generates an enable signal for AC operation, which is available to all slave controllers, and enables centralized activation of "Program mode" including switchover of all the slave controllers and activation of controller operation on the external MPC.
When redundant external APC applications are used, it manages the redundancy switchover.
The APC_Supervisor generates specific messages about the state of external programs.

Sign-of-life monitoring AC ("watchdog")

The "Watchdog Status" time interval for monitoring is specified in the faceplate and may apply to both redundant external APC programs, A and B.
.The sign-of-life "AC A alive" or signs-of-life "AC A alive" and "AC B alive" are monitored separately and in the same way as follows:
The APC_Supervisor provides a binary signal. It sets its value to FALSE and monitors the amount of time it takes for the AC to reset the signal to TRUE. If the time is longer than the "Watchdog Status" time interval, a warning is generated and a redundancy switchover may be performed.
As an alternative (`Feature.Bit5`), the APC_Supervisor does not reset the signal but only monitors the duration of the changing signal.
If a violation of the monitoring time results in termination of "Program" mode ("control"), an alarm is generated.

Sign-of-life monitoring APC-MV/APC-SP loop

The APC_Supervisor monitors the lower-level blocks by setting an "Alive" Bit to 0/1 at a structure element of the respective SP/MV loop and by monitoring the time until the signal arrives at the end of the loop. The Bit is then inverted and the entire process starts all over again. This is particularly necessary because the APC_MV/APC_OpSP blocks can be distributed across several AS.

Both loops must be closed and "alive" so that the "Program" mode ("control") can be activated.

A violation of the monitoring time of the APC_MV loop results in a termination of the "Program" mode ("control"). A corresponding alarm is generated.

A violation of the monitoring time of the APC_OpSP loop does not result in a termination of the "Program" mode ("control"). A warning is generated, however.

Setting the enable for advanced control

The APC_Supervisor enable signal is set if at least one AC is available ("alive"). If no AC is available, the enable signal is revoked.
The enable does not require external programs to be in a certain mode prediction/control).
The enable signal is forwarded to the lower-level controller via the APC_MV.

Central switching to "Program" mode

Central switching to "Program" mode is possible at APC_Supervisor level via the faceplate or the `CtrlModLi`/`PredModLi` block I/Os.
The following requirements must be met for switching to "Program" mode ("control"):

- At least one AC must be available.
- All lower-level controllers must be ready for "Program" mode (AdvCoRdy).
- All process tags relevant to controlling must have at least "Simulation" status or better.
- The APC-MV loop must be closed and "Alive".
- The APC-SP loop must be closed and "Alive".

The signal for "Program" mode is forwarded to lower-level controllers via the APC_MV. The slave controllers are switched to "Program" mode with a positive edge.
The signal for "Program" mode is also forwarded to the AC by OPC. This switches the mode from "prediction" to "control". APC_Supervisor expects feedback from "AC" mode. If the AC does not switch the mode within the "Mode switch" time interval, APC_Supervisor switches back to prediction" mode or may perform a redundancy switchover. A corresponding warning is generated. If a violation of the monitoring time results in termination of "Program" mode ("control"), an alarm is generated.

Redundancy switchover

If more than one AC is "alive", the active AC can be selected in the faceplate of APC_Supervisors. Automatic redundancy switchover can also be activated or deactivated.
If a redundancy switchover is initiated, a warning for the redundant switchover is generated and switchover to the other AC is performed.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Reserved
1	1 = Operator can switch to "Prediction" mode
2	1 = Operator can switch to control mode
3	1 = Operator can switch to "Out of service" mode
4	1 = Operator can activate AC A
5	1 = Operator can activate AC B
6	1 = Operator can activate automatic step enabling
7	1 = Operator can deactivate automatic step enabling
8	Reserved
9	1 = Operator can change the "Alive TimeOut" parameter
10	1 = Operator can change the "Control TimeOut" parameter
11 - 31	Reserved

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*.

The following modes of behavior are available for this block at the relevant bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
2	See APL function: "Reset the commands for switching over the mode"
4	See APL function: "Set switch or button mode"
5	1 = Redundant advanced controller present
6	0 = SV resets AliveTst signal; 1 = AC must reset the AliveTst signal
21	See APL function: "Bumpless switchover to 'Automatic' mode for valves, motors, and dosers"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"

Configurable reactions using the `Feature2` parameter

An overview of all the reactions provided by the `Feature` parameter can be found in the section titled *Configurable response using the Feature I/O* in the *APL Function manual*.

The following modes of behavior are available for this block at the relevant bits:

Bit	Function
0 - 31	Not used

4.2.4 APC_Supervisor error handling

Since the block does not handle process values, it only recognizes two errors:

- `MsgErr1`
- `MsgErr2`

These are generated by the `ALARM_8p` block and indicate that a message error has occurred.

Overview of error numbers

The `ErrorNum` I/O can be used to output various error numbers:

Error number	Meaning of the error number
00	No error
51	Incorrect control (e.g. "Prediction" and "Control" at the same time)

Otherwise, the error handling corresponds to the error handling of the APL_blocks.

4.2.5 APC_Supervisor messaging

Message behavior

The APC_Supervisor block uses the ALARM_8P block for generating messages. The messages can be suppressed centrally via the faceplate or via `MsgLock` (program). The free alarm inputs can be interconnected via the parameters `ExtMsg1` to `ExtMsg3`.

The associated values (`ExtVaXXX`) of the message block can be assigned freely.

Process messages

Message instance	Message ID	Message class	Event
MsgEvId1	SIG 1	High warning limit	$$BlockComment$$ AC A Mode runtime error
	SIG 2	High warning limit	$$BlockComment$$ AC B Mode runtime error
	SIG 3	High warning limit	$$BlockComment$$ AC A Not available
	SIG 4	High warning limit	$$BlockComment$$ AC B Not available
	SIG 5	High warning limit	$$BlockComment$$ Redundancy switchover A⇒B
	SIG 6	High warning limit	$$BlockComment$$ Redundancy switchover B⇒A
	SIG 7	High alarm	$$BlockComment$$ Control mode ended (AC FB mode)
	SIG 8	High alarm	$$BlockComment$$ Control mode ended (AC status)

Message instance	Message ID	Message class	Event
MsgEvId2	SIG 1	High warning limit	$$BlockComment$$ SP Loop disconnected
	SIG 2	High warning limit	$$BlockComment$$ MV Loop disconnected
	SIG 3	High alarm	$$BlockComment$$ Control mode ended (MV loop)
	SIG 4	AS control system message - Fault	$$BlockComment$$ External error has occurred
	SIG 5	AS control system message - Fault	$$BlockComment$$ External message 1
	SIG 6	AS control system message - Fault	$$BlockComment$$ External message 2
	SIG 7	AS control system message - Fault	$$BlockComment$$ External message 3
	SIG 8	< No message >	

Associated values for the message instance MsgEvId

Associated value	Block parameters
1	BatchName
2	StepNo
3	BatchID
4	ExtVa104
5	ExtVa105
6	ExtVa106
7	ExtVa107
8	ExtVa108

4.2.6 APC_Supervisor I/Os

Input parameters

Parameter	Description	Data type	Default
A_ActOp	1=Auto mode: Set advanced controller A active by operator	BOOL	0
A_AliveTst	Feedback for AC A alive	BOOL	0
A_Mode	Status AC A -> 0: Prediction only, 1: Active control	BOOL	0
AC_TimeOut	Watchdog time for AC mode change (sec)	REAL	5.0

Parameter	Description	Data type	Default
B_ActOp	1=Auto mode: Set advanced controller B active by operator	BOOL	0
B_AliveTst	Feedback for AC B alive	BOOL	0
B_Mode	Status AC B -> 0: Prediction only, 1: Active control	BOOL	0
CtrlModLi	1=Auto mode: Controlling mode by Linked or SFC • ST: Signal status • VALUE: Value	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
CtrlModOp	1=Auto mode: Controlling mode by operator	BOOL	0
LoopTimeOut	Watchdog time APC_OpSP and APC_MV loop	REAL	5.0
PredModLi	1=Manual mode: Prediction mode by Linked or SFC • ST: Signal status • VALUE: Value	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
PredModOp	1=Manual mode: Prediction mode by operator	BOOL	0
PrvBlkMV	Connection from previous block (APC_MV)	STRUCT	
PrvBlkMV.ActiveAC	Active advanced controller (0 = Controller A, 1 = Controller B)	BOOL	0
PrvBlkMV.ActiveAC_Status	Active advanced controller (0 = prediction only, 1 = active control)	BOOL	0
PrvBlkMV.AdvCoEn	Advanced control Enable	BOOL	0
PrvBlkMV.AdvCoMstrOn	Advanced control Maser On	BOOL	0
PrvBlkMV.LoopAlive	Test signal MV loop closed	BOOL	0
PrvBlkMV.MVConnOk	APC_MV connections OK	BOOL	0
PrvBlkMV.NumMV	Number of MVs	INT	0
PrvBlkMV.NumMV_AdvCoAct	Number of MVs in program mode	INT	0
PrvBlkMV.NumMV_AdvCoRdy	Number of MVs ready for advanced control	INT	0
PrvBlkMV.SPConnOk	APC_OpSP connections OK	BOOL	0
PrvBlkMV.ST_Worst	MV worst signal status	BYTE	16#00
PrvBlkMV.Supervisor_dbno	Supervisor data block number	WORD	16#0000
PrvBlkSP	Connection from previous block (APC_OpSP)	STRUCT	
PrvBlkSP.ActiveAC	Active advanced controller (0 = Controller A, 1 = Controller B)	BOOL	0
PrvBlkSP.ActiveAC_Status	Active advanced controller (0 = prediction only, 1 = active control)	BOOL	0

Parameter	Description	Data type	Default
PrvBlkSP.AdvCoEn	Advanced control Enable	BOOL	0
PrvBlkSP.AdvCoMstrOn	Advanced control Maser On	BOOL	0
PrvBlkSP.LoopAlive	Test signal OpSP loop closed	BOOL	0
PrvBlkSP.MVConnOk	APC_MV connections OK	BOOL	0
PrvBlkSP.NumOpSP	Number of OpSPs	INT	0
PrvBlkSP.NumOpSP_PVOk	Number of OpSPs with signal status 16#60 or 16#80	INT	0
PrvBlkSP.SPConnOk	APC_OpSP connections OK	BOOL	0
PrvBlkSP.ST_Worst	OpSP worst signal status	BYTE	16#00
PrvBlkSP.Supervisor_dbno	Supervisor data block number	WORD	16#0000
SelFpList	Connector for measuring point list	ANY	
SwOffOp	1=Auto mode: Set 'automatic switchover' off by operator	BOOL	0
SwOnOp	1=Auto mode: Set 'automatic switchover' on by operator	BOOL	0
TimeOut	Watchdog time for AliveA and AliveB (sec)	REAL	5.0

Output parameters

Parameter	Description	Data type	Default
A_Act	Advanced controller A is active • ST: Signal status • VALUE: Value	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 1 • ST: 16#80
A_Act	Advanced controller A is active	STRUCT	
A_Act.ST	Signal status	BYTE	16#80
A_Act.Value	Value	BOOL	1
A_Alive	Advanced controller A is alive	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
A_AliveTime	Advanced controller A alive timer	REAL	0.0
B_Act	Advanced controller B is active	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
B_Alive	Advanced controller B is alive	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
B_AliveTime	Advanced controller B alive timer	REAL	0.0
CtrlAct	Controlling mode is active	STRUCT	
CtrlAct.ST	Signal status	BYTE	16#80
CtrlAct.Value	Value	BOOL	0

Parameter	Description	Data type	Default
MV_ST_Worst	Worst signal status manipulated values	BYTE	16#80
NxtBlkMV	Connection from previous block (APC_MV)	STRUCT	
NxtBlkMV.ActiveAC	Active advanced controller (0 = Controller A, 1 = Controller B)	BOOL	0
NxtBlkMV.ActiveAC_Status	Active advanced controller (0 = prediction only, 1 = active control)	BOOL	0
NxtBlkMV.AdvCoEn	Advanced control Enable	BOOL	0
NxtBlkMV.AdvCoMstrOn	Advanced control Maser On	BOOL	0
NxtBlkMV.LoopAlive	Test signal MV loop closed	BOOL	0
NxtBlkMV.MVConnOk	APC_MV connections OK	BOOL	0
NxtBlkMV.NumMV	Number of MVs	INT	0
NxtBlkMV.NumMV_AdvCoAct	Number of MVs in program mode	INT	0
NxtBlkMV.NumMV_AdvCoRdy	Number of MVs ready for advanced control	INT	0
NxtBlkMV.SPConnOk	APC_OpSP connections OK	BOOL	0
NxtBlkMV.ST_Worst	MV worst signal status	BYTE	16#00
NxtBlkMV.Supervisor_dbno	Supervisor data block number	WORD	16#0000
NxtBlkSP	Connection from previous block (APC_OpSP)	STRUCT	
NxtBlkSP.ActiveAC	Active advanced controller (0 = Controller A, 1 = Controller B)	BOOL	0
NxtBlkSP.ActiveAC_Status	Active advanced controller (0 = prediction only, 1 = active control)	BOOL	0
NxtBlkSP.AdvCoEn	Advanced control Enable	BOOL	0
NxtBlkSP.AdvCoMstrOn	Advanced control Maser On	BOOL	0
NxtBlkSP.LoopAlive	Test signal OpSP loop closed	BOOL	0
NxtBlkSP.MVConnOk	APC_MV connections OK	BOOL	0
NxtBlkSP.NumOpSP	Number of OpSPs	INT	0
NxtBlkSP.NumOpSP_PVOk	Number of OpSPs with signal status 16#60 or 16#80	INT	0
NxtBlkSP.SPConnOk	APC_OpSP connections OK	BOOL	0
NxtBlkSP.ST_Worst	OpSP worst signal status	BYTE	16#00
NxtBlkSP.Supervisor_dbno	Supervisor data block number	WORD	16#0000
PredAct	Prediction mode is active	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 1 • ST: 16#80
SP_ST_Worst	Worst signal status setpoints	BYTE	16#80

APC_Supervisor interface to Advanced Controller

The following table lists the most important block I/Os / variables that are available for AC coupling at the APC_Supervisor. The 'AC' column indicates whether the AC reads the variable or writes to it. The signals which are absolutely necessary for coupling are marked in bold in the 'AC' column.

Input parameters

I/O name	Comment	Data type	AC
A_AliveTst	Feedback for AC A alive	BOOL	Read / Write
B_AliveTst	Feedback for AC B alive	BOOL	Read / Write
A_Mode	Status AC A -> 0: Prediction only, 1: Active control	BOOL	Write
B_Mode	Status AC B -> 0: Prediction only, 1: Active control	BOOL	Write

Table AC interface APC_Supervisor

Output parameters

I/O name	Comment	Data type	AC
CtrlAct#Value	Controlling mode is active	BOOL	Read
A_Act#Value	Advanced controller A is active	BOOL	Read
B_Act#Value	Advanced controller B is active	BOOL	Read

A specific behavior of the AC or ACs is required for the AC coupling to work correctly.

- AliveTST:
  Feature.Bit6 = 0: The AC reads the variable. As soon as the value of the variable changes to 'False', the AC sets the value back to 'True'. The AC A reads from and writes to the variable A_AliveTst, the AC B reads from and writes to the variable B_AliveTst.
  Feature.Bit6 = 1: The AC alternately sets the variable to 0 and 1.
- Mode monitoring:
  AC A reads the variables CtrlAct.Value and A_Act.Value. If both values are ‘True’, it goes to ‘Control’ mode and sets the A_Mode variable to ‘True’. Otherwise, it goes to 'Prediction' mode and sets the A_Mode tag to 'False'.
  AC B works accordingly with the CtrlAct.Value, B_Act.Value and B_Mode variables.

4.2.7 APC_Supervisor block diagram

The block has no block diagram.

4.2.8 Operator control and monitoring

4.2.8.1 APC_Supervisor views

APC_Supervisor block views

The APC_Supervisor block provides the following views:

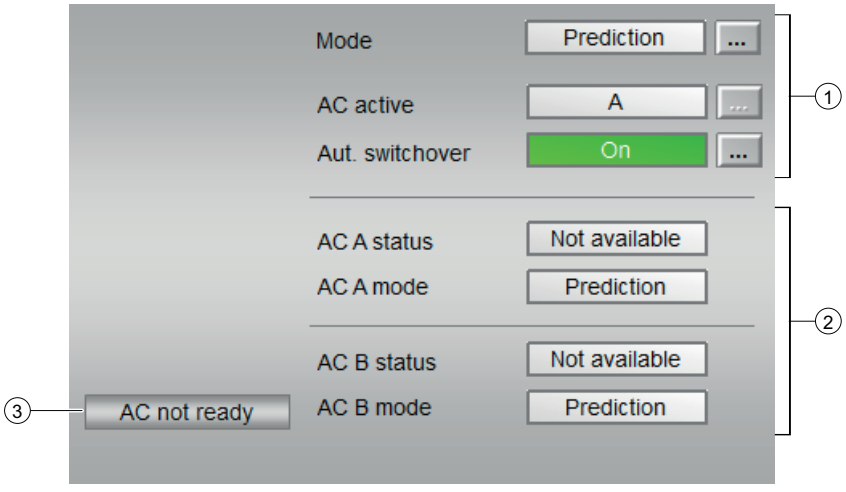
- APC_Supervisor standard view (Page 145)
- Message view
- APC_Supervisor setpoint/controller list view (Page 148)
- APC_Supervisor process tag list view (Page 150)
- APC_Supervisor parameter view (Page 151)
- APC_Supervisor preview (Page 151)
- Memo view
- Batch view
- APC_Supervisor block icons (Page 154)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

4.2.8.2 APC_Supervisor standard view

The respective elements are hidden or shown depending on whether the higher-level controller is designed for standalone or redundant operation.

Standard view for redundant higher-level control of APC_Supervisor



(1) Controlling of the higher-level controllers (AC)

This area shows the operating mode of the block, which higher-level controller is controlled by the block and whether an automatic changeover takes place when an active redundant controller fails.

- Mode
  - Prediction
  - Rules
  - Out of service
- AC active
  - A
  - B
- Aut. switchover
  - On
  - Off

(2) Status and mode of the higher-level controller (AC)

This area shows the status and mode of the higher-level controllers A and B.

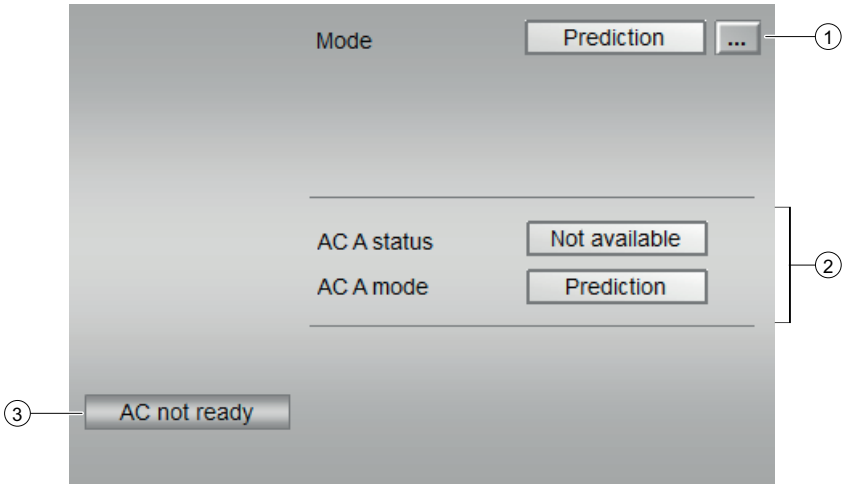
- Status:
  - Available
  - Not available
- Mode
  - Prediction
  - Rules

(3) Display area for states of the APC connection

This area provides additional information on the operating state of the APC connection:

- AC not ready
- PV not OK
- MV not ready
- SP loop faulty
- MV loop faulty

Standard view for standalone higher-level control of APC_Supervisor



(1) Controlling of the higher-level controller (AC)

This area shows the operating mode of the block.

- Mode
  - Prediction
  - Rules
  - Out of service

(2) Status and mode of the higher-level controller (AC)

This area shows the status and mode of the higher-level controller.

- Status:
  - Available
  - Not available
- Mode
  - Prediction
  - Rules

(3) Display area for states of the APC connection

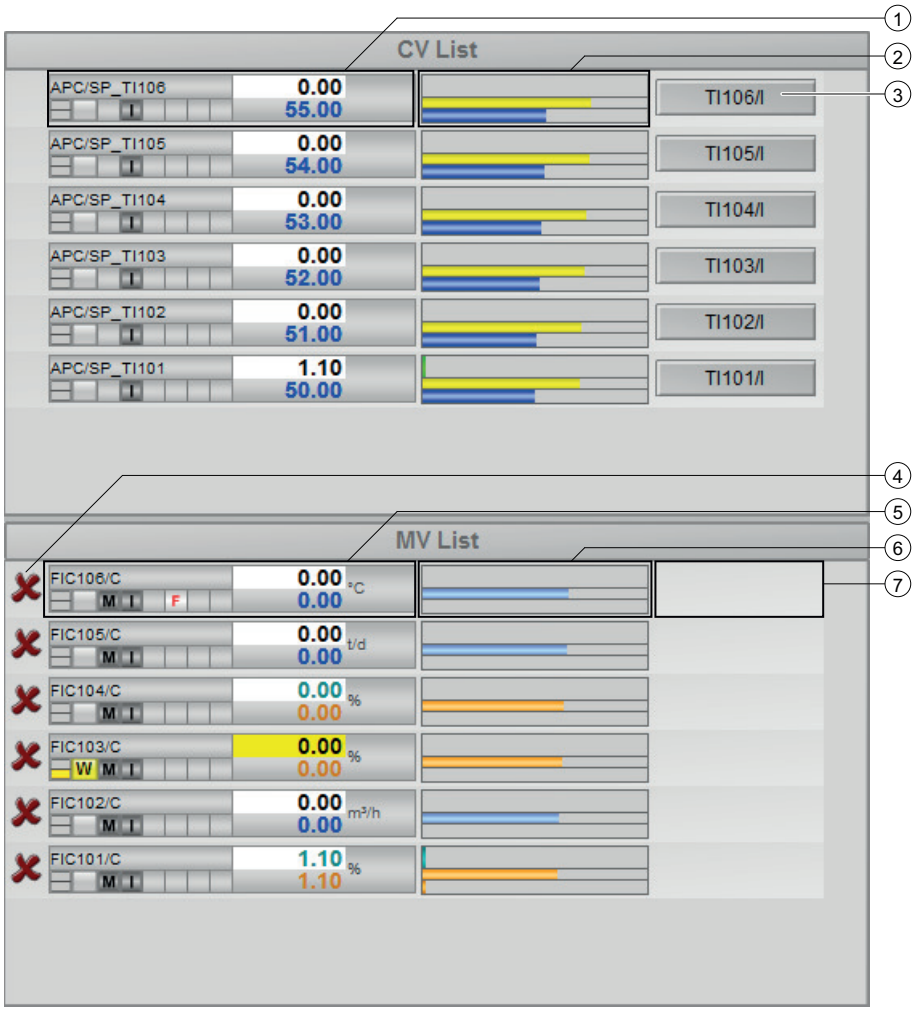
This area provides additional information on the operating state of the APC connection:

- AC not ready
- PV not OK
- MV not ready

- SP loop faulty
- MV loop faulty

4.2.8.3 APC_Supervisor setpoint/controller list view

APC_Supervisor list view



The setpoints and process values of the APC control system are displayed in the area **CV** (Controlled Value) **List**. An APC_OpSP block exists for each setpoint/process value. Each APC_OpSP block is displayed in a 'row'.

The controllers that are subordinate to the APC control system are displayed in the area **MV** (Manipulated Value) **List**. Each controller corresponds to one 'row'.

(1) Block icon of the setpoint block (APC_OpSP)

This area shows an adapted representation of the block icon of the setpoint blocks (APC_OpSP) assigned to the higher-level controller.

(2) Bar graph for the process value, target process value and setpoint

This area shows the current process value, the current target process value and the setpoint in the form of a bar graph.

(3) Button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

The navigation button (3) opens the standard view of the block interconnected with the APC_OpSP connection "SelMeas". This connection is usually interconnected with the monitoring block that also supplies the process value.

(4) Enable

This area displays whether the controller is ready for the Program mode. If the controller is ready, nothing is displayed. If the controller is not ready, a red "X" is displayed.

(5) Block icon of the controller

This area shows an adapted representation of the APL icon of the lower-level controller.

(6) Bar graph for the process value and setpoint

This area shows the current process value, the setpoint of the higher-level controller (AC) and the setpoint of the lower-level controller in the form of a bar graph.

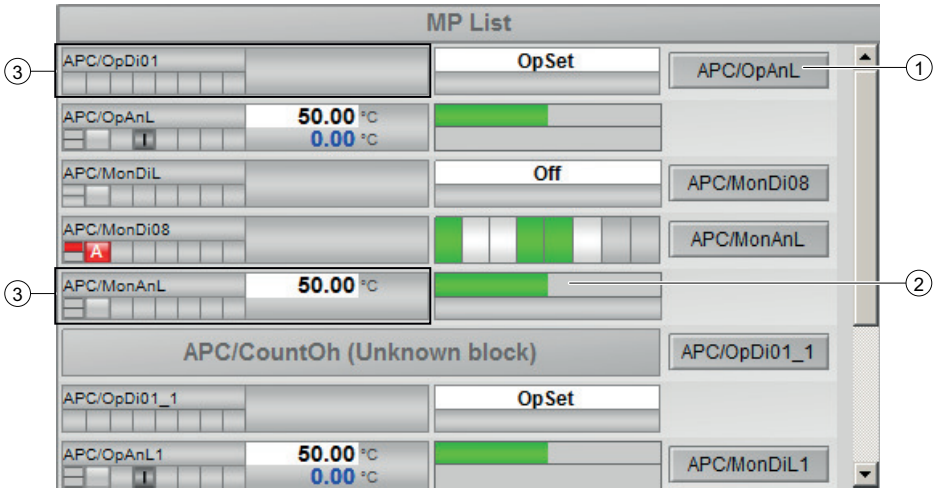
(7) Button for switching to the standard view of a faceplate

This area shows a navigation button when a controller that is not supported is connected. This navigation button allows you to open the standard view of this block. The visibility of this navigation button depends on the configuration in the engineering system (ES). The button opens the faceplate of the APL block that is connected with the APC_MV.

If a supported controller is connected, the standard view of this block is called directly via the block icon (5).

4.2.8.4 APC_Supervisor process tag list view

Process tag list view of APC_Supervisor



Different process tags can be displayed in the **MP** (Measuring Point) **list**, (see also 'Description of APC_Coupling blocks, Process tag list view). Each process tag corresponds to a 'row'.

(1) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

This navigation button opens the standard view of the block interconnected with the "`SelFP1`" connection of the measuring point.

(2) Display for process value

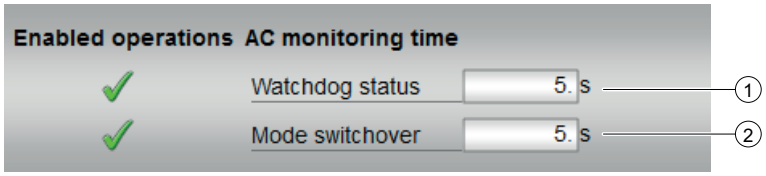
This area shows the current process value of the measuring point in various forms. For analog values, a bar graph is displayed; for digital values, a status display.

(3) Block icon of the supported APL block

This area shows an adapted representation of the block icon of the supported APL block.

4.2.8.5 APC_Supervisor parameter view

APC_Supervisor parameter view



(1) Display and change the monitoring time AC Watchdog

You specify the time for the sign-of-life monitoring here. More information is available in the section APC_Supervisor functions (Page 136) under "Sign-of-life monitoring AC ("Watchdog")"

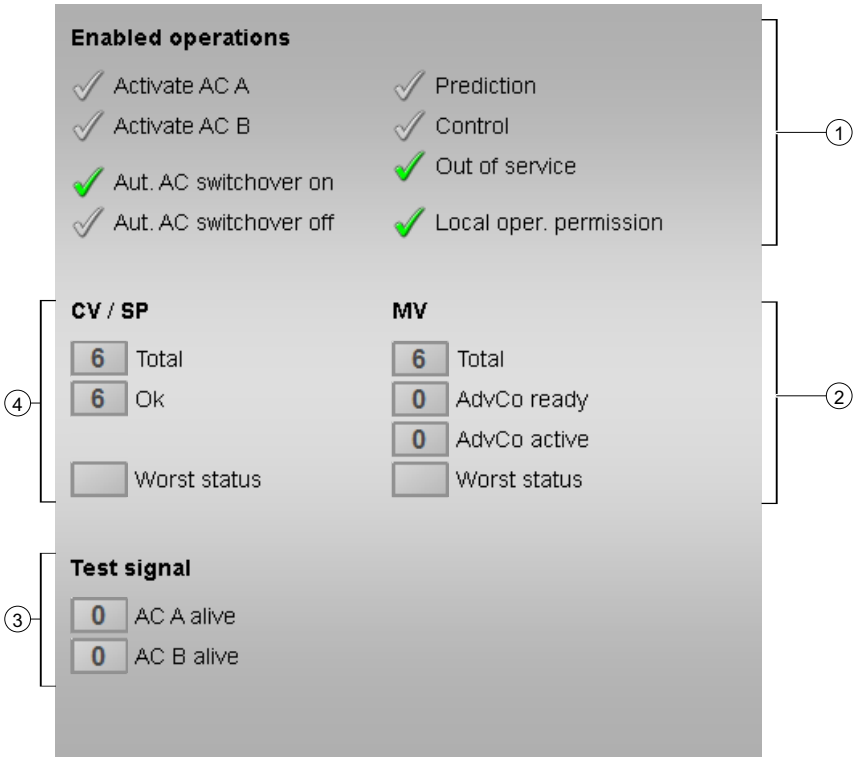
(2) Display and change the monitoring time mode changeover

Here you specify the time in which the higher-level controller (AC) has to change to the "Rules" state at an operating mode changeover.

4.2.8.6 APC_Supervisor preview

The respective elements are hidden or shown depending on whether the higher-level controller is designed for standalone or redundant operation.

APC_Supervisor preview for redundant higher-level control



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator is generally prohibited from accessing this parameter due to the configured AS operator control permissions (`OS_Perm` or `OS1Perm`)

(2) Manipulated Values

This area provides information about the lower-level controllers.

- Total: Number of lower-level controllers
- AdvCo ready: Number of lower-level controllers that are ready for the Program mode
- AdvCo active: Number of lower-level controllers that are in Program mode
- Worst status: Worst signal status of all the lower-level controllers

(3) Test Signal

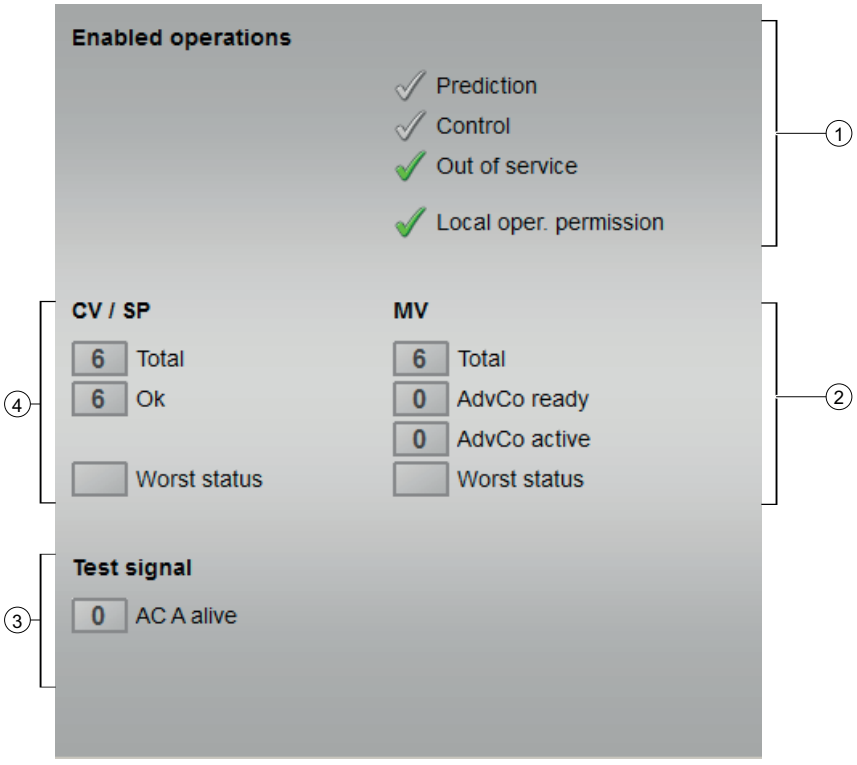
Shows the state of the watchdog signal of the higher-level redundant controller.

- AC A alive
- AC B alive

(4) Current Value / Setpoint

- Total: Number of lower-level setpoint blocks (APC_OpSP)
- OK: Number of lower-level setpoint blocks at which the signal status of the process value is "good"
- Worst status: Worst signal status of the lower-level setpoint blocks (APC_OpSP)

APC_Supervisor preview for standalone higher-level control



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator is generally prohibited from accessing this parameter due to the configured AS operator control permissions (`OS_Perm` or `OS1Perm`)

(2) Manipulated Values

This area provides information about the lower-level controllers.

- Total: Number of lower-level controllers
- AdvCo ready: Number of lower-level controllers that are ready for the Program mode
- AdvCo active: Number of lower-level controllers that are in Program mode
- Worst status: Worst signal status of all the lower-level controllers

(3) Test Signal

Shows the state of the watchdog signal of the higher-level controller.

- AC A alive

(4) Current Value / Setpoint

- Total: Number of lower-level setpoint blocks (APC_OpSP)
- OK: Number of lower-level setpoint blocks at which the signal status of the process value is "good"
- Worst status: Worst signal status of the lower-level setpoint blocks (APC_OpSP)

4.2.8.7 APC_Supervisor block icons

APC_Supervisor block icons

A variety of block icons are available with the following functions:

- Process tag type
- Violation of the alarm and warning limits as well as process control errors
- Operating modes
- Signal status, release for maintenance
- Memo display

The block icons from template @TemplateILV8.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	

You can find additional information on the block icon and the control options in the block icon in the following sections of the APL manual:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

4.3 APC_OpSP

4.3.1 APC_OpSP description

Object name (type + number) and family

Type + number: FB 1181

Family: APC

Number of process objects: 1 PO

Calling OBs

- Cyclic interrupt OB in which the block is installed (e.g., OB32)
- OB100 (see Startup characteristics)

Startup characteristics

Based on `Feature.Bit0`, old values are retained when the block starts, or the block starts with the most recently saved values.

'`RunUpCyc`' indicates the number of cycles for which new alarm messages are suppressed at block startup.

Called blocks

FC369	SelST16
SFB35	ALARM_8P
SFC6	RD_SINFO
SFC20	BLKMOV
SFC24	TEST_DB

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section APC_OpSP I/Os (Page 160).

Status bit	Parameter
0	`Occupied`
1	`BatchEn`
2	Not used
3	`OosAct.Value`
4	`OosLi.Value`
5	Not used
6	`OnAct.Value`

Status bit	Parameter
7	SP_UpRaAct, SP_DnRaAct limits active for "Gradient" mode (SP_RateOn=1)
8	SP_ExtAct.Value
9	SP_LoAct.Value
10	SP_HiAct.Value
11 - 13	Not used
14	SP_RmpOn
15	SP_RmpModTime
16 - 31	Not used

4.3.2 APC_OpSP operating modes

The block does not have any operating modes.

4.3.3 APC_OpSP function

The APC_OpSP block represents the interface between the AC and the process value. It is used to specify the AC setpoint and deadband, makes the process value available to the AC, and displays the target process value and the mode (prediction/control) of the active AC. It also passes the status of the process value to APC_Supervisor .
The APC_OpSP block is detected by the faceplate of APC_Supervisor and so appears in its CVlist. An APC_OpSP is needed for each relevant setpoint/process value of the AC.

APC_OpSP is an extended OpAnL block from the Advance Process Library. Therefore, please refer to the APL documentation for more detailed information.

In addition to the functionality of the APL OpAnL block, the APC_OpSP has the following functions:

- Display of the "target process value" of AC A or AC B in the standard Faceplates view (depending on whether AC A or AC B is active)
- Evaluation to determine whether PV_In is 'good' (status 16#60 or 16#80) or 'bad'. The result is displayed at the output of the block at PV_Bad.
- Evaluation to determine whether SP_In is 'good' (status 16#60 or 16#80) or 'bad'. The result is displayed at the output of the block at SP_Bad.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the OS_Perm parameter:

Bit	Function
0	Not used
1	1 = The operator can switch to "On" mode

Bit	Function
2	Not used
3	1 = Operator can switch to "Out of service" mode
4 - 9	Not used
10	1 = Operator can switch to "internal"
11	1 = Operator can switch to "external"
12	1 = Operator can switch to "Bumpless switchover" mode
13	1 = Operator can change SP_Int
14	1 = Operator can activate SP_RateOn
15	1 = Operator can change SP_UpRaLim
16	1 = Operator can change SP_DnRaLim
17	1 = Operator can activate the setpoint ramp function (SP_RmpOn)
18	1 = Operator can activate the time setpoint ramp function (SP_RmpModTime)
19	1 = Operator can change the ramp time (SP_RmpTime)
20	1 = Operator can change the ramp setpoint (SP_RmpTarget)
21	1 = Operator can change the parameter value of the deadband
22 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*.

The following modes of behavior are available for this block at the relevant bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
16	See APL function: "Process value with separate scale range"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"

4.3.4 APC_OpSP error handling

The error handling approach is the same as for the OpAnL block of the APL.

There is also the `BlkConErr` output. This shows whether the APC_OpSP block has access to the correct version of APC_Supervisors. It checks this based on the length of the instance data block.

4.3.5 APC_OpSP messaging

Message behavior

The APC_Supervisor block uses the ALARM_8P block for generating messages. The messages can be suppressed centrally via the faceplate or via `MsgLock` (program). The free alarm inputs can be interconnected via the parameters `ExtMsg1` to `ExtMsg3`.

The associated values (`ExtVaXXX`) of the message block can be assigned freely.

Process messages

Message instance	Message ID	Message class	Event
MsgEvId	SIG_1	AS process control message - fault	$$BlockComment$$ External message 1
	SIG_2	AS process control message - fault	$$BlockComment$$ External message 2
	SIG_3	AS process control message - fault	$$BlockComment$$ External message 3
	SIG_4	AS process control message - fault	$$BlockComment$$ External message 4
	SIG_5	< No message >	
	SIG_6	< No message >	
	SIG_7	< No message >	
	SIG_8	< No message >	

Associated values for the message instance MsgEvId

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVa104`
5	`ExtVa105`
6	`ExtVa106`
7	`ExtVa107`
8	`ExtVa108`

4.3.6 APC_OpSP I/Os

Input parameters

Parameter	Description	Data type	Default
A_StdySTTg_In *	AC steady state target prediction A	REAL	0.0
B_StdySTTg_In *	AC steady state target prediction B	REAL	0.0
MsgEvId	Message event ID	DWORD	16#000000DC
PrvBlkSP	Connection from previous block (APC_OpSP or APC_Supervisor)	STRUCT	
PrvBlkSP.ActiveAC	Active advanced controller (0 = Controller A, 1 = Controller B)	BOOL	0
PrvBlkSP.ActiveAC_Status	Active advanced controller (0 = prediction only, 1 = active control)	BOOL	0
PrvBlkSP.AdvCoEn	Advanced control Enable	BOOL	0
PrvBlkSP.AdvCoMstrOn	Advanced control Maser On	BOOL	0
PrvBlkSP.LoopAlive	Test signal OpSP loop closed	BOOL	0
PrvBlkSP.MVConnOk	APC_MV connections OK	BOOL	0
PrvBlkSP.NumOpSP	Number of OpSPs	INT	0
PrvBlkSP.NumOpSP_PVOk	Number of OpSPs with signal status 16#60 or 16#80	INT	0
PrvBlkSP.SPConnOk	APC_OpSP connections OK	BOOL	0
PrvBlkSP.ST_Worst	OpSP worst signal status	BYTE	16#00
PrvBlkSP.Supervisor_dbno	Supervisor data block number	WORD	16#0000
PV_In	Feedback analog input	STRUCT	
PV_In.ST	Signal status	BYTE	16#80
PV_In.Value	Value	REAL	0.0
SelMeas	Select connected measuring point	ANY	
SelSV	Select supervisor	ANY	
SP_OpScale	SP- bar display limits for OS	STRUCT	
SP_OpScale.High	High value	REAL	100.0
SP_OpScale.Low	Low value	REAL	0.0
SP_Unit	Engineering units of input	INT	1001

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
BlkConErr	Block connection error (supervisor)	BOOL	0
MsgAckn	Message acknowledgment status	WORD	16#0000

Parameter	Description	Data type	Default
MsgErr	1=Messaging error occurs	BOOL	0
MsgStat	Message status	WORD	16#0000
NxtBlkSP	Connection to next block (APC_OpSP or APC_Supervisor)	STRUCT	
NxtBlkSP.ActiveAC	Active advanced controller (0 = Controller A, 1 = Controller B)	BOOL	0
NxtBlkSP.ActiveAC_Status	Active advanced controller (0 = prediction only, 1 = active control)	BOOL	0
NxtBlkSP.AdvCoEn	Advanced control Enable	BOOL	0
NxtBlkSP.AdvCoMstrOn	Advanced control Maser On	BOOL	0
NxtBlkSP.LoopAlive	Test signal OpSP loop closed	BOOL	0
NxtBlkSP.MVConnOk	APC_MV connections OK	BOOL	0
NxtBlkSP.NumOpSP	Number of OpSPs	INT	0
NxtBlkSP.NumOpSP_PVOk	Number of OpSPs with signal status 16#60 or 16#80	INT	0
NxtBlkSP.SPConnOk	APC_OpSP connections OK	BOOL	0
NxtBlkSP.ST_Worst	OpSP worst signal status	BYTE	16#00
NxtBlkSP.Supervisor_dbno	Supervisor data block number	WORD	16#0000
PV_Bad	Bad status of PV_In (not 16#80 and not 16#60)	BOOL	0
SP_Bad	Bad status of SP_Out (not 16#80 and not 16#60)	BOOL	0
StdyStTg_Out	AC steady state target prediction of active AC	REAL	0.0

APC_OpSP interface to Advanced Controller

The following table lists the most important block I/Os / tags that are available for AC coupling at APC_OpSP. The 'AC' column indicates whether the AC reads the variable or writes to it. The signals which are absolutely necessary for coupling are marked in bold in the 'AC' column.

Input parameters

Parameter	Description	Data type	AC
SP_HiLim	Input high limit	REAL	Read
SP_LoLim	Input low limit	REAL	Read
SP_OpScale#High	SP- bar display limits for OS	REAL	Read
SP_OpScale#Low	SP- bar display limits for OS	REAL	Read
PV_In#Value	Value	REAL	Read
A_StdyStTg_In	AC steady state target prediction A	REAL	Write
B_StdyStTg_In	AC steady state target prediction B	REAL	Write
PV_OpScale#High	PV- bar display limits for OS	REAL	Read
PV_OpScale#Low	PV- bar display limits for OS	REAL	Read

Parameter	Description	Data type	AC
PV_Unit	Engineering units of feedback input	INT	Read
DeadBand	Setpoint deadband	REAL	Read

Output parameters

Parameter	Description	Data type	AC
PV_Bad	Bad status of PV (not 16#80 and not 16#60)	BOOL	Read
SP_Out#Value	Value	REAL	Read
SP_Bad	Bad status of SP_OutPV (not 16#80 and not 16#60)	BOOL	Read

The job of the APC_OpSP block is to supply the AC with the process value, process setpoint, and deadband for the setpoint. These values are made available to the tags '`PV_In.Value`' (process value), '`SP_Out.Value`' (setpoint) and '`DeadBand`' (deadband).
The APC_OpSP block visualizes the target process value predicted by the AC. To do this the AC A must write its target process value to the `A_StdySt Tg_In` tag and AC B must write its target process value to the `B_StdySt Tg_In` tag. AC_OpSP automatically visualizes the target process value of the active AC.

4.3.7 APC_OpSP block diagram

The block has no block diagram.

4.3.8 Operator control and monitoring

4.3.8.1 APC_OpSP views

APC_OpSp block views

The APC_OpSp block provides the following views:

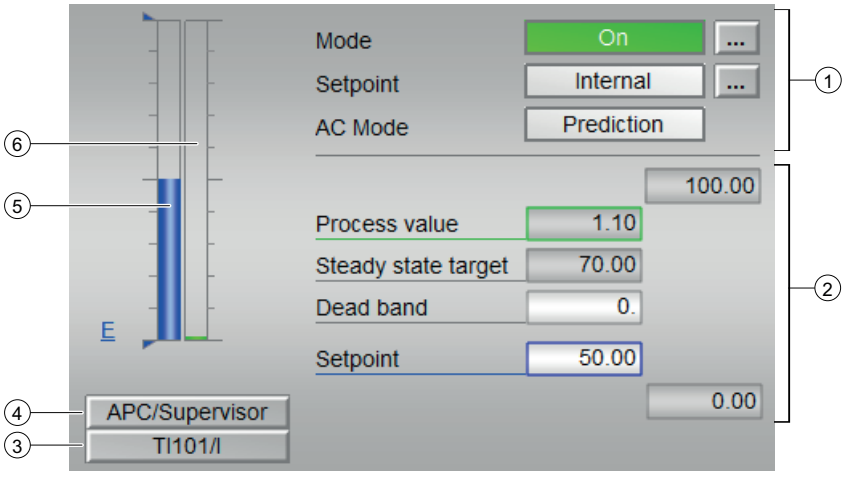
- APC_OpSP standard view (Page 163)
- Message view
- Trend view
- Ramp view
- Parameter view
- Preview
- Memo view
- Batch view
- APC_OpSP block icons (Page 164)

With the exception of the standard view, all views of the APC_OpSP block correspond to the views of the OpAnL block for APL.

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

4.3.8.2 APC_OpSP standard view

APC_OpSP standard view



(1) Display and changeover of the operating mode, setpoint specification and display of the AC mode

- Mode: This area provides information on the currently valid operating mode.
  The following operating modes can be shown here:
  - On
  - Out of service
- Setpoint: This area shows how to specify the setpoint.
  The setpoint can be specified as follows:
  - By the application ("External", CFC/SFC)
  - By the user directly in the faceplate ("Internal").
- AC mode: This area shows you the current state of the higher-level controller
  - Prediction
  - Rules

(2) Dynamic display

- Process value: Shows the current process value with the corresponding signal status.
- Target process value: The process value predicted by the higher-level controller (AC)

- Dead band: Area in which the higher-level controller does not act on this process value
- Setpoint: Setpoint for the higher-level controller specified by the operator or system (AC).

(3)+(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

The navigation button (3) opens the standard view of the block interconnected with the connection "SelMeas". This connection is usually interconnected with the monitoring block that also supplies the process value.

The navigation button (4) opens the standard view of the block interconnected with the connection "SelSV". This connection is usually interconnected with the associated APC_Supervisor block.

(5) Bar graph for the setpoint

This area shows the current setpoint in the form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

(6) Bar graph for the process value

This area shows the current process value in form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

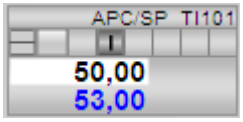
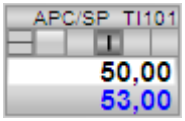
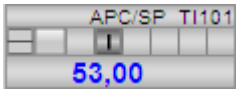
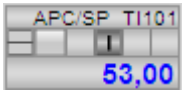
4.3.8.3 APC_OpSP block icons

APC_OpSP block icons

A variety of block icons are available with the following functions:

- Process tag type
- Operating modes
- Internal and external setpoint specification
- Signal status, release for maintenance
- Memo display
- Process value (black, with and without decimal places)
- Setpoint (blue, with and without decimal places)

The block icons from template @TemplateAPLV8.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	
	3	
	4	

You can find additional information on the block icon and the control options in the block icon in the following sections of the APL manual:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

4.4 APC_MV

4.4.1 Description of APC_MV

Object name (type + number) and family

Type + number: FB 1182

Family: APC

Number of process objects: 0 PO

Startup characteristics

- No special startup characteristics

Called blocks

FC369	SelST16
SFC24	TEST_DB

Calling OBs

- Cyclic interrupt OB in which the block is installed (e.g., OB32)

4.4.2 APC_MV operating modes

The block does not have any operating modes.

4.4.3 APC_MV function

The APC_MV block represents the standardized interface between the AC and the lower-level controller. All relevant control signals can be read directly by APC_MV, regardless of the type of controller which is connected.
Depending on the AC that is active, APC_MV passes the manipulated variable `A_AdvCoMV` or `B_AdvCoMV` to the lower-level controller.

This means either input `A_AdvCoMV_Bad` or `B_AdvCoMV_Bad` is forwarded to the `AdvCoMV_Bad` block output.

The APC_MV block reads and writes all the necessary data of known controller blocks (PIDConL, PIDConR, PIDStepL) via an Any pointer. Unknown blocks can be linked to APC_MV by means of direct interconnection of the corresponding block I/Os; parameters and measuring ranges can be set to fixed values if necessary.

With Feature.Bit5 = 1, the manipulated variable A_AdvCoMV / B_AdvCoMV is not forwarded directly to the controller via pointers; it is only made available at the AdvCoMV block output. Instead of the manipulated variable A_AdvCoMV / B_AdvCoMV, the value of the M_AdvCoMV block input is written to the controller. This allows you to manipulate the AdvCoMV value in the CFC and to forward it to the lower-level controller via M_AdvCoMV.

The APC_MV block reads and evaluates the signal status of the lower-level controller block ('good' for status 16#60 or 16#80, otherwise 'bad') and outputs it at the MV_Bad output. When AdvCoModSP = 0, the states of ST_Worst and MV_Rbk are processed; when AdvCoModSP = 1, only the ST_Worst status is evaluated.

The APC_MV block is detected by the faceplate of the APC_Supervisor block. If a known controller block is connected via an Any pointer, a corresponding icon appears in its MV list. If an unknown block is connected, an icon for the APC_MV appears in the list.

Sign-of-life monitoring of the APC_MV loop

The APC_MV block monitors the "Alive" bit in the loop structure. If the controller is in "Program" mode ("control") and the monitoring time is exceeded, the APC_MV block switches its lower-level controller block from "Program" mode back to the previous mode.

Configurable reactions using the Feature parameter

An overview of all the reactions provided by the Feature parameter can be found in the section titled *Configurable response using the Feature I/O* in the *APL Function manual*.

The following modes of behavior are available for this block at the relevant bits:

Bit	Function
0 - 4	Not used
5	0: AdvCoMV(PtrCtrl)=AdvCoMV 1: AdvCoMV(PtrlCtrl)=M_AdvCoMV.Value
6 - 31	Not used

4.4.4 APC_MV error handling

The APC_MV block only recognizes errors that concern connections to the APC_Supervisor block and to the lower-level controller block.

A connection error is displayed at the BlkConErr output. It is triggered when an incorrect version of APC_Supervisor or the controller block is connected. This is checked based on the length of the corresponding instance data block.

4.4.5 APC_MV messaging

The block has no message behavior.

4.4.6 APC_MV I/Os

Input parameters

Parameter	Description	Data type	Default
A_AdvCoMV *	Manipulated value from advanced controller A	REAL	0.0
A_AdvCoMV_Bad *	Status of manipulated value from advanced controller A is bad	BOOL	0
B_AdvCoMV *	Manipulated value from advanced controller B	REAL	0.0
B_AdvCoMV_Bad *	Status of manipulated value from advanced controller B is bad	BOOL	0
BlockType *	BlockType (1 = PIDConL, 2 = PIDConR, 3 = PIDStepL, Else = Linked)	INT	0
LoopTimeOut	Watchdog time APC_MV loop	REAL	5.0
M_AdvCoMV	Modified manipulated value	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
MV_Rbk	MV readback value	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
PrvBlkMV	Connection from previous block (APC_MV or APC_Supervisor)	STRUCT	
PrvBlkMV.ActiveAC	Active advanced controller (0 = Controller A, 1 = Controller B)	BOOL	0
PrvBlkMV.ActiveAC_Status	Active advanced controller (0 = prediction only, 1 = active control)	BOOL	0
PrvBlkMV.AdvCoEn	Advanced control Enable	BOOL	0
PrvBlkMV.AdvCoMstrOn	Advanced control Maser On	BOOL	0
PrvBlkMV.LoopAlive	Test signal MV loop closed	BOOL	0
PrvBlkMV.MVConnOk	APC_MV connections OK	BOOL	0
PrvBlkMV.NumMV	Number of MVs	INT	0
PrvBlkMV.NumMV_AdvCoAct	Number of MVs in program mode	INT	0
PrvBlkMV.NumMV_AdvCoRdy	Number of MVs ready for advanced control	INT	0
PrvBlkMV.SPConnOk	APC_OpSP connections OK	BOOL	0
PrvBlkMV.ST_Worst	MV worst signal status	BYTE	16#00
PrvBlkMV.Supervisor_dbno	Supervisor data block number	WORD	16#0000
PtrCTRL	Pointer controller	ANY	
ST_WorstIn *	Worst signal status	BYTE	16#80

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
AdvCoEn	1 = Enable program mode	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
AdvCoEn	1 = Enable program mode	STRUCT	
AdvCoMV.ST	Signal status	BYTE	16#80
AdvCoMV.Value	Value	REAL	0.0
AdvCoMV_Bad	Status of manipulated value of active advanced controller is bad	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
BlkConErr	Block connection error (supervisor or controller type)	STRUCT • VALUE: BOOL • ST: BYTE	- • VALUE: 0 • ST: 16#80
MV_Bad	Bad status of connected controller (not 16#80 and not 16#60)	BOOL	0
NxtBlkMV	Connection to next block (APC_MV or APC_Supervisor)	STRUCT	
NxtBlkMV.ActiveAC	Active advanced controller (0 = Controller A, 1 = Controller B)	BOOL	0
NxtBlkMV.ActiveAC_Status	Active advanced controller (0 = prediction only, 1 = active control)	BOOL	0
NxtBlkMV.AdvCoEn	Advanced control Enable	BOOL	0
NxtBlkMV.AdvCoMstrOn	Advanced control Maser On	BOOL	0
NxtBlkMV.LoopAlive	Test signal MV loop closed	BOOL	0
NxtBlkMV.MVConnOk	APC_MV connections OK	BOOL	0
NxtBlkMV.NumMV	Number of MVs	INT	0
NxtBlkMV.NumMV_AdvCoAct	Number of MVs in program mode	INT	0
NxtBlkMV.NumMV_AdvCoRdy	Number of MVs ready for advanced control	INT	0
NxtBlkMV.SPConnOk	APC_OpSP connections OK	BOOL	0
NxtBlkMV.ST_Worst	MV worst signal status	BYTE	16#00
NxtBlkMV.Supervisor_dbno	Supervisor data block number	WORD	16#0000

APC_MV interface to Advanced Controller

The following table lists the most important block I/Os / variables that are available for AC coupling at the APC_MV. The 'AC' column indicates whether the AC reads the variable or writes to it. The signals which are absolutely necessary for coupling are marked in bold in the 'AC' column.

Input parameters

I/O name	Comment	Data type	AC
A_AdvCoMV	Manipulated value from advanced controller A	REAL	Write
A_AdvCoMV_Bad	Status of manipulated variable from advanced controller A is 'bad'	BOOL	Write
B_AdvCoMV	Manipulated value from advanced controller B	REAL	Write
B_AdvCoMV_Bad	Status of manipulated variable from advanced controller B is 'bad'	BOOL	Write
AdvCoAct#Value	1 = Advanced control (program mode) active	BOOL	Read
MV_HiLim#Value	Limit (high) for manipulated variable MV	REAL	Read
MV_LoLim#Value	Limit (low) for manipulated variable MV	REAL	Read
MV_Rbk#Value	MV readback value	REAL	Read
MV_OpScale#High	MV - display limits for OS	REAL	Read
MV_OpScale#Low	MV - display limits for OS	REAL	Read
MV_Unit	Unit of MV	INT	Read

Table AC interface APC_MV

Output parameters

I/O name	Comment	Data type	AC
MV_Bad	Bad status of connected controller (not 16#80 and not 16#60)	BOOL	Read

The job of the APC_MV block in reference to the AC is to pass the manipulated variable of the active AC to the lower-level controller. To do this the AC A must write its manipulated variable to the `A_AdvCoMV` variable and AC B must write its manipulated variable to the `B_AdvCoMV` variable.

4.4.7 APC_MV block diagram

The block has no block diagram.

4.4.8 Operator control and monitoring of APC_MV

The block does not have any views.

4.5 APC_MpList

4.5.1 APC_MpList description

Object name (type + number) and family

Type + number: FB 1184

Family: APC

Number of process objects: 0 PO

Called blocks

- No block calls

Calling OBs

- Cyclic interrupt OB in which the block is installed (e.g., OB32)

4.5.2 APC_MpList operating modes

The block does not have any operating modes.

4.5.3 APC_MpList functions

The APC_MpList is required to create the list of selected process tags if these are in different AS.

4.5.4 APC_MpList error handling

The block does not have any error handling.

4.5.5 APC_MpList messaging

The block has no message behavior.

4.5.6 APC_MpList I/Os

Input parameters

The block does not have any input parameters.

Output parameters

Parameter	Description	Data type	Default
SelFp2Out		BOOL	0

4.5.7 APC_MpList block diagram

The block has no block diagram.

4.5.8 Operator control and monitoring of APC_MpList

The block does not have any views.

Conversion Blocks

5

5.1 IntBit – Conversion of an integer to a bit mask

Area of application and operating principle

The block converts the integer input value `In` [range 0...32] into a bit pattern (33 bit output signals). When "n" is entered, it sets exactly the output bit `BITn` to TRUE, all other bits 0 ... 32 are set to FALSE.

Example:

If, for example, `In=15`, BIT15 is set to "true" and all other output bits are set to "false" .

An invalid value at `In (In < 0` or `In > 32`) sets all the output BITs to "false".

Connections of 'IntBit'

Input parameters

Parameter	Description	Type	Default
`In`	Value to convert	`INT`	

Output parameters

Parameter	Description	Type	Default
`Bit00`	Result flag 0 ... 32	`BOOL`	
...			
`Bit32`			

5.2 BitInt – Conversion of a bit mask to an integer ...

Area of application and operating principle

This block forms the counterpart to the block IntBit. The block decodes the most significant bit or least significant bit from 33 bit input signals (Bit0 ... Bit32) into an integer value (0 ... 32).

The most significant / least significant bit is selected through the input parameter `Order`. A substitute value can be configured at the `SubsVal` input.

The numerical value is made available in the following formats at separate outputs:

- BYTE
- WORD
- INT
- DINT
- DWORD
- REAL

Connections of ‚BitInt'

Input parameters

Parameter	Description	Type	Default

`Bit00`
...
`Bit32`
	Bit to evaluate 0 ... 32	`BOOL`	

`Order`	0 := 'MSB (32 --> 0)' 1 := 'LSB (0 --> 32)'	`BOOL`	
`SubsVal`	Substitution value to use if no BitXX input is set	`INT`	`-1`

Output parameters

Parameter	Description	Type	Default
`IdxB`	Index value as a BYTE (0-32)	`BYTE`	
`IdxW`	Index value as a WORD (0-32)	`WORD`	
`IdxI`	Index value as an INT (0-32)	`INT`	
`IdxDI`	Index value as a DINT (0-32)	`DINT`	
`IdxDW`	Index value as a DWORD (0-32)	`DWORD`	
`IdxR`	Index value as a REAL (0-32)	`REAL`	

5.3 CmDec – Structure decoding 'CMData'

Area of application and operating principle

This block makes batch-specific information available when control modules from standard libraries (such as PCS7 Library blocks MOTOR, MOT_REV, VALVE, VAL_MOT) are used. It is responsible for decoding and compiling the `CmData` data structure for outputting individual signals.

Connections of „CmDec"

Input parameters

Parameter	Description	Type	Default
`CmData`	Control module info structure	`BlUdtCmData`	

Output parameters

Parameter	Description	Type	Default
`UnitID`	UnitID	`DWORD`	
`BatchID`	Current Batch ID	`DWORD`	
`BatchName`	Current Batch name	`STRING32`	
`StepNo`	Batch step number	`DWORD`	
`BatchOcc`	Occupied by Batch, enables second message	`BOOL`	
`BatchEn`	Enable remote operation of controller by Batch recipe	`BOOL`	
`SimEnCM`	Local CM simulation is enabled	`BOOL`	`True`
`SimOnParent`	Unit or equipment simulation is active	`BOOL`	
`EqmLock`	Lock equipment module	`BOOL`	
`ManOpEn`	Operator may switch to MANUAL mode	`BOOL`	
`ResetPls`	Reset signal	`BOOL`	
`ModLiOp`	1=Link/Auto,0=Manual: Input for auto/ manual commands	`BOOL`	
`AutModLi`	1=Auto mode: Auto mode by link or SFC	`BOOL`	
`ManModLi`	1=Manual mode: Manual mode by link or SFC	`BOOL`	
`Res1` … `Res5`	(Reserved)	`BOOL`	
`FbkDelayRatio`	Feedback delay ratio [%]	`BYTE`	

5.4 CmEnc – Structure encoding 'CMData'

Area of application and operating principle

This function block is responsible for encoding and compiling individual batch data interface signals for the `CMData` output data structure.

Connections of ‚CmEnc'

Input parameters

Parameter	Description	Type	Default
`BatchID`	Current Batch ID	`DWORD`	
`BatchName`	Current Batch name	`STRING32`	
`StepNo`	Batch step number	`DWORD`	
`BatchOcc`	Occupied by Batch, enables second message	`BOOL`	
`BatchEn`	Enable remote operation of controller by Batch recipe	`BOOL`	
`SimEnCM`	Local CM simulation is enabled	`BOOL`	`True`
`SimOnParent`	Unit or equipment simulation is active	`BOOL`	
`EqmLock`	Lock equipment module	`BOOL`	
`ManOpEn`	Operator may switch to MANUAL mode	`BOOL`	
`ResetPls`	Reset signal	`BOOL`	
`ModLiOp`	1=Link/Auto,0=Manual: Input for auto/ manual commands	`BOOL`	
`AutModLi`	1=Auto mode: Auto mode by link or SFC	`BOOL`	
`ManModLi`	1=Manual mode: Manual mode by link or SFC	`BOOL`	
`Res1` … `Res5`	(Reserved)	`BOOL`	
`FbkDelayRatio`	Feedback delay ratio [%]	`BYTE`	

Output parameters

Parameter	Description	Type	Default
`CmData`	Control module info structure	`BlUdtCmData`	

5.5 MidStr – Conversion of material ID to string

Area of application and operating principle

This function block converts a material ID (DINT value) into a string ID value. A specific limit range (high, low) is applied to the input value. When a limit is exceeded, the configured substitute value is used instead. The block is used mainly in combination with the 'BlTank' block.

Connections of ‚MidStr'

Input parameters

Parameter	Description	Type	Default
MatID	Material ID to convert	DINT	
MinID	Minimum ID to accept	DINT	0
MaxID	Maximum ID to accept	DINT	1024
LimitLen	Limit string to that length	INT	4
ReplVal	Replacement value	STRING[16]	0

Output parameters

Parameter	Description	Type	Default
MatSP	Converted ID	STRING[16]	, ,

5.6 StrMid – Conversion of string to material ID

Area of application and operating principle

This function block converts a String-ID into a material ID (DINT). A specific limit range (high, low) is applied to the input value. When a limit is exceeded, the configured substitute value is used instead. The block is used mainly in combination with the 'Tank' block.

Connections of ‚StrMid‘

Input parameters

Parameter	Description	Type	Default
MatSP	Material code	STRING[16]	, ,
ValMin	Minimum value	DINT	0
ValMax	Maximum value	DINT	1024
ReplVal	Replacement value	DINT	0

Output parameters

Parameter	Description	Type	Default
MatID	Converted material code	DINT	0

5.7 OpA16Dec – Structure decoding 'Opa16'

Area of application and operating principle

The block converts the input data structure with up to 16 REAL values into 16 separate output parameters of the type REAL. The block is used mainly in combination with the ‚OpA16' block. This makes available a structure output with all 16 operating values. This reduces the interconnection work required when the same parameters have to be supplied to several target blocks.

Example:

A OpA16 operator-control block supplies data for several polylines with the block 'BlPolygonExt'.

In this application the source `-->` target relation is implemented with 1:n individual connections of `OpA16 --> OpA16Dec` . The polygon turning points (input parameters of BlPolygonExt) are interconnection to the `OpA16Dec` outputs.

Connections of "OpA16Dec"

Input parameters

Parameter	Description	Type	Default
`VStruct`	Analog data structure from OpA16	`STRUCT` • `V1 : REAL` • … • `V16 : REAL`	`, ,`

Output parameters

Parameter	Description	Type	Default
`V1`	Analog output value 1	`REAL`	
…	…	`REAL`	
`V16`	Analog output value 16	`REAL`	

Operator control blocks

6

6.1 Aggr16 / Aggr08 - Unit switchover of 16/08 units

6.1.1 Aggr16/ Aggr08 description

Object name (type + number) and family

Aggr16:

Type + number: FB 1113

Family: Operate

Number of process objects: 1 PO

Aggr08:

Type + number: FB 1114

Family: Operate

Number of process objects: 1 PO

Area of application for Aggr16 / Aggr08

The block is used to switch up to 8/16 units based on the runtime or master unit criteria.

Functional principles

The block switches a certain number of units (which can be specified) on/off and tries to keep the desired number constant. If a unit is switched on or off manually, this is also taken into consideration. When runtime-dependent or priority-dependent switchover is not activated, the master unit is switched through to maintain a uniform load of the individual units. These specifications can be implemented in "Automatic" or in "Manual" mode. When runtime-dependent or priority-dependent switchover is activated and a change in the requirements occurs or the operating states change, the units with the lowest run time / highest priority are switched on and the units with the highest run time / lowest priority are switched off.

Configuration

Install the block in a cyclic interrupt OB (OB30 to OB38) in the CFC Editor. The block is also installed automatically in the startup OB (OB100).

The inputs `Aggr1RdyToStart` to `Aggr8RdyToStart` / `Aggr16RdyToStart` are to be interconnected with the outputs `RdyToStart` of the units to be controlled.

Interconnect the unit state ("Motor running") at the inputs `Aggr1Started` to `Aggr8Started` / `Aggr16Started`. This is usually the `Start` output of the unit, if necessary, in connection with the `FbkRunOut` output or your own logic.

If you need a operating hours display for the individual units (mandatory for off/on depending on operating time), the inputs `RunTi01` to `RunTi08` / `RunTi16` must be interconnected to the `TotalTime` output of a CountOh assigned to the respective unit.

For control of the units, the outputs `Start01` to `Start08` / `Start16` must be interconnected to the `StartAut` start control in automatic mode of the unit.

Startup characteristics

Use the startup characteristics Feature bit 0 "Set startup characteristics" to define the startup characteristics for this block.

Following startup, the messages are suppressed for the number of cycles assigned in the `RunUpCyc` parameter.

A startup can be simulated via the `Restart` input.

Called blocks

- SFC6 (RD_SINFO)
- SFC20 (BLKMOV)
- SFB35 (ALARM_8P)

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section Aggr16/ Aggr08 I/Os (Page 190).

Status bit	Parameter
0	Occupied
1	`BatchEn`
2	Not used
3	`OosAct.Value`
4	`OosLi.Value`
5	`AutoAct.Value`
6 - 7	Not used
8	`OnAct`
9	`OffAct`
10	`ExtSwAct`
11	`IntSwAct`
12	`TimeSwAct`
13	`ExtNoAct`
14	`IntNoAct`
15	`DelayEn`

Status bit	Parameter
16	OpTimeSwModeAct
17	PrefUnitMode = 2
18 -21	Not used
22	Invalid signal
23 - 24	Not used
25	DelayAct
26 - 31	Not used

Status word assignment for the parameter Status2

Status bit	Parameter
0	MsgLock
1 - 31	Not used

6.1.2 Aggr16/ Aggr08 operating modes

The block can be operated using the following modes:

- Automatic mode
- Manual mode
- Out of service

The next section provides additional block-specific information relating to the general descriptions.

"Automatic" mode

General information on "Automatic mode" and for switching between operating modes is available in the section *Operating modes of the blocks* in the *APL Function manual*.

In "Automatic mode" you can:

- "Power ON" (OnAut = 1)
- "Power OFF" (OffAut = 1) the unit block.

"Manual" mode

General information on "Manual mode" and for switching between operating modes is available in the section *Operating modes of the blocks* in the *APL Function manual*.

In "Manual mode" you can:

- "Power ON" (OnMan = 1)
- "Power OFF" (OffMan = 1) the unit block.

"Out of service"

General information on "Out of service" mode is available in the section *Operating modes of the blocks* in the *APL Function manual*.

If the block is in "Out of service" mode, all controls of the outputs `Start01` to `Start08` / `Start16` are reset to 0.

6.1.3 Aggr16/ Aggr08 functions

The functions for this block are listed below.

Number of units

The block can switch a maximum of 8/16 units. If a number greater than 8/16 is entered at input parameter `MaxAggr`, this is limited to 8/16.

Only available units can be switched. Information about availability and status of a unit is supplied by the inputs `AggrXRdy` and `AggrXStarted`.

Only available units in Off status can be switched on. Only available units in On status can be switched off.

If a unit is not available but switched on, it is considered for the required number and the start output (`StartXX`) is updated later.

If the required number of units cannot be started, the missing number of units is output at the `UnitMissing` output parameter for further interconnection or evaluation. The error bit `NotAch` is also output in this case.

External/internal setpoint specification

The following modes are available within the context of setpoint specification for the *number of units*: "External" and "Internal". This setting is made by the AS program (`NoLiOp` = 1) or via the OS (`NoLiOp` = 0).

The number of units is specified in external mode by means of the interconnectable input `UnitReqLi` and in internal mode with the parameter `UnitReqOp` (input by means of OS).

Switchover criterion

The unit block offers two options to switch over the units:

Leading (master) unit (`OpTimeSwMode` = 0):

The setpoint specification of the leading unit has the modes external, internal and time. This setting is done with the AS program (`SwLiOp` = 1) or by means of the OS (`SwLiOp` = 0).

The master unit is specified in internal mode with the parameter `UnitMstrOp` (input by means of OS).

In "External" mode, the master unit is determined in the block.
When the block is started for the first time, the first unit is the master. The master is switched over

by means of a positive edge at the ExtSw input. (Because the edge of the ExtSw input is evaluated in the block, the user has to ensure that it is reset.)

The master unit is also determined in the block in time mode: The switchover takes place automatically once a specified time has expired (ChangeTime). This time continues to run in external and internal mode.

The role of the master is usually passed on to the next unit during switchover. When Feature.Bit6 is activated, the role of master is passed on to the next unit that is switched on. If no units are switched on, the step enable is performed as normal.

Activation/deactivation depending on operating time (OpTimeSwMode = 1, PrefUnitMode = 1)

The deactivation/activation depending on operating time can be activated during configuration by the user with the parameters OpTimeSwMode and PrefUnitMode. If the parameters OpTimeSwMode = 1 and PrefUnitMode = 1, the unit with the lowest operating time number is switched in case of activation, and the unit with the highest operating time number is switched in case of deactivation. Only the number of required units can be set manually.

The value 0 is always output as leading unit (master unit) at the block. The display of the master unit is hidden in the faceplate.

Step enabling only takes place if the required number of units changes or if the status of a unit has changed. The current operating time of the units is configured at the inputs RunTi01 to RunTi08 / RunTi16 . Because there is no automatic step enabling, the value 0 is output as the remaining time until the next switchover at the block. The value is hidden in the faceplate.

All other functions and the operator control / monitoring options remain as described in the mode setting.

Activation/deactivation depending on priorities (OpTimeSwMode = 1, PrefUnitMode = 2)

The deactivation/activation depending on priorities can be activated during configuration by the user with the parameters OpTimeSwMode and PrefUnitMode. If the parameters OpTimeSwMode = 1 and PrefUnitMode = 2, the unit with the lowest priority (smallest PrioXX value) is switched in case of activation, and the unit with the lowest priority (largest PrioXX value) is switched in case of deactivation. Only the number of required units can be set manually.

The value 0 is always output as leading unit (master unit) at the block. The display of the master unit is hidden in the faceplate.

Step enabling only takes place if the required number of units changes or if the status of a unit has changed. The priority of the units is configured at the inputs Prio01 to Prio08 / Prio16. Priorities can be assigned from 1 to 99, with 1 being the highest and 99 being the lowest priority.

The parameter view displays an additional column with the configured priorities in the Activation/deactivation depending on priorities mode. Because there is no automatic step enabling, the value 0 is output as the remaining time until the next switchover at the block. The value is hidden in the faceplate.

All other functions and the operator control / monitoring options remain as described in the mode setting.

Delay of activation and deactivation

The activation and deactivation of units in case of request change can be delayed by assigning parameters. The parameter `DelayOn` is used for activation and the `DelayOff` for deactivation. These delay times are only active if they were enabled with Feature Bit 5 . You can set the delay times with activated operator control enable via the OS in the parameter view.

Enabling of the master unit during an interval

You can use `Feature.Bit7` to determine if the availability of the master unit should be checked during the time interval (available and started, no errors).

The default setting is 0.

Bit = 0: No evaluation of the master unit.

Bit = 1: The availability of the master unit is checked during the time interval. If the unit is not available, the next available unit becomes the master unit. If there is a switchover of the master unit, the interval time is increased by one interval period one time in the interval to minimize the operating cycles.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*.

The following modes of behavior are available for this block at the relevant bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
5	1 = Enable delay time
6	1 = Step enabling until next unit switched on
7	1 = Switchover of the master unit during an interval
19	See APL function: "Changes signal status of outputs in OOS"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can switch to "automatic mode"
1	1 = Operator can switch to "manual mode"
2	Not used
3	1 = Operator can switch to "Out of service" mode

Bit	Function
4	1 = Operator can switch to Off
5	1 = Operator can switch to On
6 - 15	Not used
16	1 = Operator can switch a number of units to external mode
17	1 = Operator can switch a number of units to internal mode
18	1 = Operator can enter the number of units
19	1 = Operator can switch the master unit to external
20	1 = Operator can switch the master unit to internal
21	1 = Operator can switch the master unit to time mode
22	1 = Operator can enter the master unit
23	1 = Operator can enter the switching time
24	1 = Operator can enter the delay time
25	1 = Operator can set the priority of the unit
24 - 31	Not used

Opening additional faceplates

This block features the standard function "Opening additional faceplates". Information is available in the section *Opening additional faceplates* in the *APL Function manual*.

6.1.4 Aggr16/ Aggr08 error handling

The block monitors the correctness of the interconnected, configured inputs received via faceplate while it is processed.

Parameter assignment error`ParamFail` Error bit if parameter at block is not within valid range.

Valid ranges:

- 0 < `ChangeTime` <= 576 hours
- 0 <= `MaxAggr` <= 8 / 16

Values with error input:

- `MaxAggr` > 8 / 16 ⇒ `MaxAggr` := 8 / 16
- `MaxAggr` < 0 ⇒ `MaxAggr` := 0
- `ChangeTime` > 576 ⇒ `ChangeTime`:= 576h (24 days)
- `ChangeTime` <= 0 ⇒ `ChangeTime` := 168h (7 days)

Interconnection error `ConErr`: Error bit if interconnected input `UnitReqLi` is not within valid range.

Valid range:

- 0 <= `UnitReqLi` <= `MaxAggr`

Values with error input:

- `UnitReqLi` ⇒ previous number is retained.

6.1.5 Aggr16/ Aggr08 messaging

Message behavior

The following messages can be generated for this block:

- Process messages
- Instance-specific messages

Process messages

Message instance	Message ID	Message class	Event
`MsgEvId 1`	SIG 1	< No message >	$$BlockComment$$ Time-driven step enabling not possible
	SIG 2	Operating message – without acknowledgment	$$BlockComment$$ Operator active

Explanation:

$$BlockComment$$: content of the instance-specific comment

The message "Time-driven step enabling not possible" is sent if the time for time-driven step enabling has expired but the block is internal or external step enabling mode.

The message is not active by default.

Instance-specific messages

You can use up to six instance-specific messages with this block.

Message instance	Message ID	Message class	Event
MsgEvId 1	SIG 1	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 2	AS process control message - fault	$$BlockComment$$ External message 2
	SIG 3	AS process control message - fault	$$BlockComment$$ External message 3
	SIG 4	AS process control message - fault	$$BlockComment$$ External message 4
	SIG 5	AS process control message - fault	$$BlockComment$$ External message 5
	SIG 6	AS process control message - fault	$$BlockComment$$ External message 6

Explanation:

$$BlockComment$$: content of the instance-specific comment

Associated values for the message instance `MsgEvId`

Associated value	Block parameters
1	BatchName
2	StepNo
3	BatchID
4	ExtVa104
5	ExtVa105
6	ExtVa106
7	ExtVa107
8	ExtVa108
9	ExtVa109
10	ExtVa110

The associated values 4 ... 10 are allocated to the parameters ExtVa104 ... ExtVa110, which you are free to use. You can find additional information on this in the "Process Control System PCS 7 - Engineering System" manual.

6.1.6 Aggr16/ Aggr08 I/Os

Input parameters

Parameter	Description	Type	Default
`Aggr1RdyToStart` … `Aggr08RdyToStart/` `Aggr16RdyToStart`	1 = Unit 1 ... 8 / 16 is ready to start	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
`Aggr1Started` … `Aggr08Started/` `Aggr16Started`	Feedback for unit 1 ... 8 / 16 is available: 1 = Start 0 = Stop	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
`ChangeTime *`	Time to change the master unit in [h]	INT	168
`DelayOff *`	Delay time for activation of unit [s]	REAL	0.0
`DelayOn *`	Delay time unit deactivation [s]	REAL	0.0
`ExtNoLi`	1 = "external number of units", via interconnection or SFC	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
`ExtNoOp *`	1 = "external number of units", via OS operator	BOOL	0
`ExtSw`	1 = External sequential switching	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
`ExtSwLi`	1 = External mode for switching: External mode via interconnection or SFC	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
`ExtSwOp *`	Ext mode for switch: Ext mode by operator	BOOL	0
`Feature`	I/O for additional functions. The description of the functions can be found in the "... functions" section for the respective block.	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 0 • .. • 0
`IntNoLi`	1 = Int mode for number of units: Int mode by Linked or SFC	STRUCT • Value: BOOL • ST: BYTE	- • 1 • 16#80
`IntNoOp *`	Int mode for number of units: Int Mode by Operator	BOOL	1
`IntSwLi`	1 = Int mode for switch: Int mode by Linked or SFC	STRUCT • Value: BOOL • ST: BYTE	- • 1 • 16#80
`IntSwOp *`	Int mode for switch: Int Mode by Operator	BOOL	1
`MaxAggr *`	Max. number of units	INT	0

Parameter	Description	Type	Default
MsgEvId	Message ID	DWORD	16#000000B9
MstrOld	Master unit from operator restored	INT	0
NoLiOp	1 = Link/Auto,0=Manual: Input to number of units	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OffAut	1 = Off: Off command in Auto mode	STRUCT • Value: BOOL • ST: BYTE	- • 1 • 16#80
OffMan *	1 = Off: Off command in manual mode	BOOL	1
OnAut	1 = On: On command in Auto mode	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OnMan *	1 = On: On command in manual mode	BOOL	0
OpTimeSwMode	0 = Master unit switch mode active 1 = Operating Time switch mode active	BOOL	0
OS_Perm	I/O for operator permissions (Page 184)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 1 • .. • 1
PrefUnitMode	1 = Switchover depending on operating time 2 = Switchover depending on priorities	INT	0
Prio01 ... Prio08/ Prio16	Switching-on priority unit 1 ... 8 / 16 (1 – 99)	INT	0
ReqOld	Number of units from operator restored	INT	0
Restart	Manual restart	BOOL	0
RunTi01 ... RunTi08/ RunTi16 *	Runtime unit 1 ... 8 / 16	DWORD	16#00000000
Start01Old ... Start08Old/ Start16Old	1 = Start unit 1 ... 8 / 16 old status	BOOL	0
SwLiOp	1 = Link/Auto 0 = Manual: Input to mode for switch	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
TimeSwLi	1 = Time mode for switch: Time mode by Linked or SFC	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
TimeSwOp *	Time mode for switch: Time Mode by Operator	BOOL	0
UnitMstrOp *	Master Unit from Operator	INT	1
UnitReqLi	External number for units via interconnection	INT	1
UnitReqOp *	Internal number of units via OS operator	INT	1

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Type	Default
ChangeTime_Out	Residual time for changing master	TIME	0ms
ConErr	1 = Connection error	BOOL	0
ExtNoAct	1 = Ext for number of units: Ext mode indicator	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
ExtSwAct	1 = Ext for switch: Ext mode indicator	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
IntNoAct	1 = Int for number of units: Int mode indicator	STRUCT • Value: BOOL • ST: BYTE	- • 1 • 16#80
IntSwAct	1 = Int for switch: Int mode indicator	STRUCT • Value: BOOL • ST: BYTE	- • 1 • 16#80
MsgAckn	Message: ACK_STATE output	WORD	16#0000
MsgErr	1 = Message Error	BOOL	0
MsgStat	Message: STATUS output	WORD	16#0000
MsgSup	1 = Message suppression active	BOOL	0
NotAch	1 = Number of units not achieved	BOOL	0
NoUnitAvail	Number of units on and available	INT	0
NoUnitRun	Compared number of units required/on	INT	0
Off	1 = Unit off	STRUCT • Value: BOOL • ST: BYTE	- • 1 • 16#80
On	1 = Unit on	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OpTimeSwModeAct	1 = OpTimeSwMode	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
ParamFail	1 = Parameter failure	BOOL	0
PrefUnitModeOut	0 = Preferred unit mode is not active 1 = Switchover depending on operating time active 2 = Switchover depending on priorities active	INT	0

Parameter	Description	Type	Default
PrioOut01 ... PrioOut08/ PrioOut16	Switching-on priority unit 1 ... 8 / 16 (1 – 99)	INT	1
Start01 ... Start08/ Start16	1 = Start unit 1 ... 8 / 16	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
TimeSwAct	1 = Time for switch: Time mode indicator	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
UnitAvail	Units available display	WORD	16#0000
UnitMissing	Units missing	INT	0
UnitMstr	Master unit	INT	0
UnitOn	Units on display	WORD	16#0000
UnitReq	Number of units required	INT	0
UnitReserved	Reserved units	INT	0

6.1.7 Aggr16/ Aggr08 block diagram

A block diagram is not provided for this block.

6.1.8 Operator control and monitoring

6.1.8.1 Aggr16/ Aggr08 views

Aggr16 / Aggr08 block views

The block provides the following views:

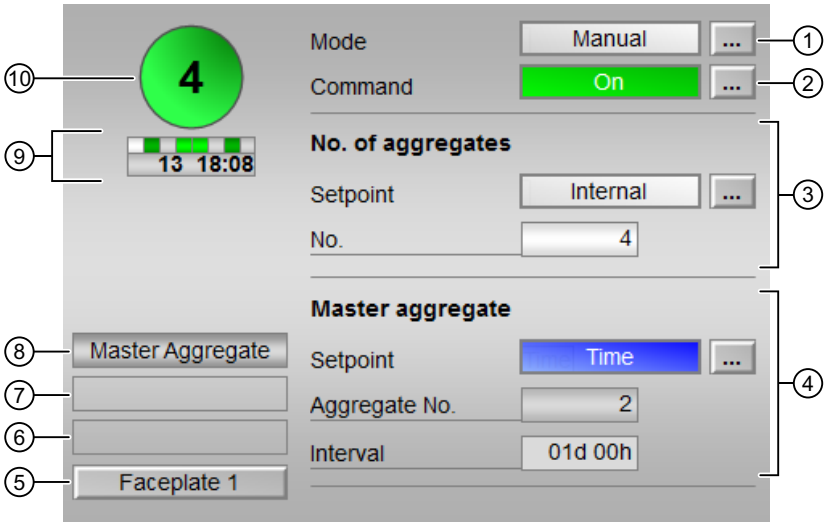
- Aggr16/ Aggr08 standard view (Page 194)
- Message view
- Aggr16/ Aggr08 parameter view (Page 197)
- Aggr16/ Aggr08 preview (Page 199)
- Memo view
- Batch view
- Aggr16/ Aggr08 block icon (Page 200)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

6.1.8.2 Aggr16/ Aggr08 standard view

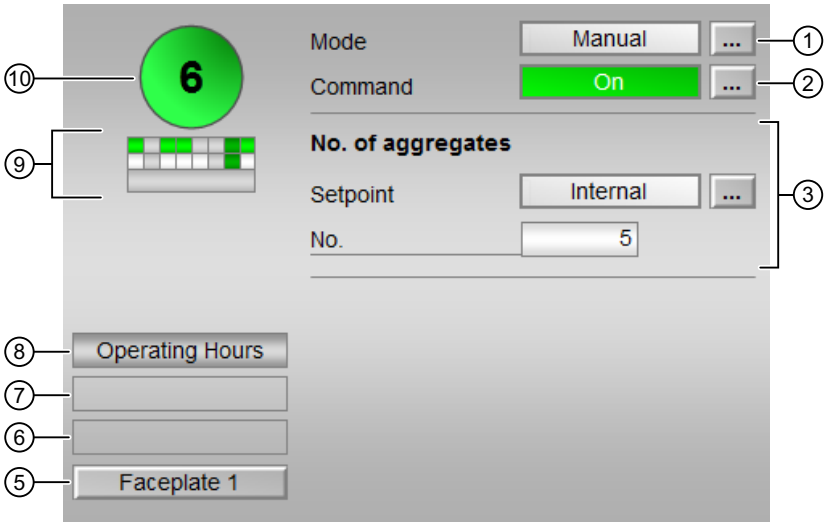
Aggr08 standard view

Connection / disconnection with master unit:



Aggr16 standard view

Runtime-dependent or priority-dependent connection / disconnection:



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- Manual mode
- Automatic mode
- Out of service

For more information on switching the operating mode, refer to the section "Switching operating states and operating modes" in the APL documentation.

(2) Display and switchover of the command

This area shows the current selection. The following operating modes can be shown here:

- "On"
- "Off"

You can find more information on this in the section "Switching operating states and operating modes" of the APL documentation.

(3) Number of units

Display and switchover of the setpoint specification for the number of units. This area shows how to specify the setpoint. The setpoint can be specified as follows:

- By the application ("External", CFC / SFC)
  By the user directly in the faceplate ("Internal").

For more information on switching the setpoint specification, refer to the chapter "Switching operating states and operating modes" in the APL manual. For more information, refer to the chapter "Setpoint specification - internal/external" in the APL manual.

(4) Master unit

Display and switchover of the setpoint specification for the master unit. This area is only visible when the block is in "Master unit" mode (OpTimeSwMode = 0) and indicates how the setpoint is specified. The setpoint can be specified as follows:

- By the application ("External", edge in the CFC/SFC)
- By the user directly in the faceplate ("Internal").
- By expiration of time at the block ("Time" block logic)

For more information on switching the setpoint specification, refer to the chapter "Switching operating states and operating modes" in the APL manual. For more information, refer to the chapter "Setpoint specification - internal/external" in the APL manual.

(5) Navigation button for switching to the standard view of any faceplate

This display is only visible when the corresponding block input is interconnected.

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

For more information, refer to the "Opening additional faceplates" section in the APL documentation.

(6) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"
- "Delay"

For more information, refer to the section "Simulating signals" and "Display of delay times" in the APL documentation.

(7) Display area for block states

This area provides additional information on the operating state of the block:

- "Invalid signal"

You can find more information about this in the section Aggr16/ Aggr08 error handling (Page 187).

(8) Display area for block states

This area provides additional information on the operating state of the block:

- "Master unit"
- "Priority"
- "Operation time"

You can find more information about this in the section Aggr16/ Aggr08 functions (Page 184).

(9) and (10) Extended status display and status display

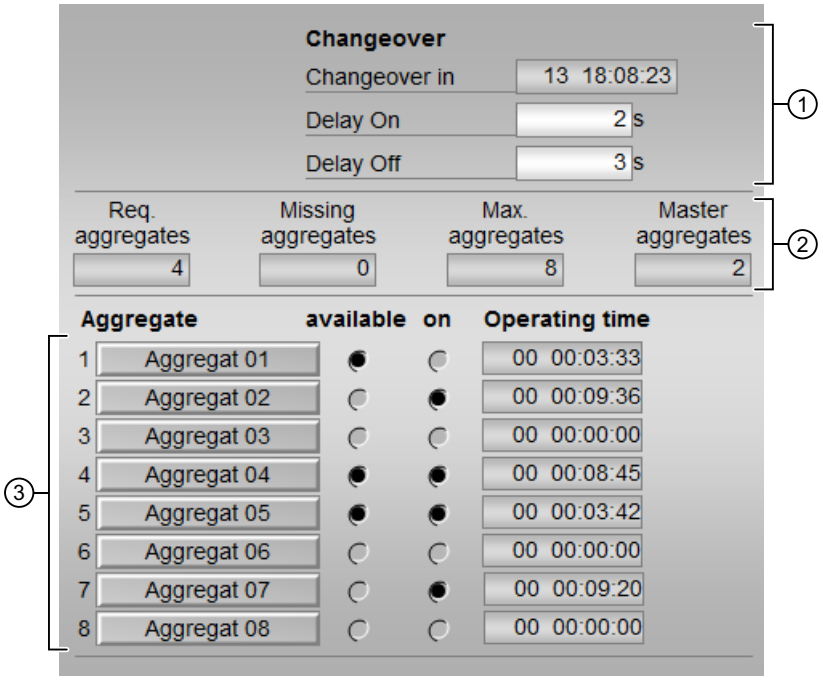
The current status of the motor is graphically displayed here.

You can find more information about this in the section Aggr16/ Aggr08 block icon (Page 200)

6.1.8.3 Aggr16/ Aggr08 parameter view

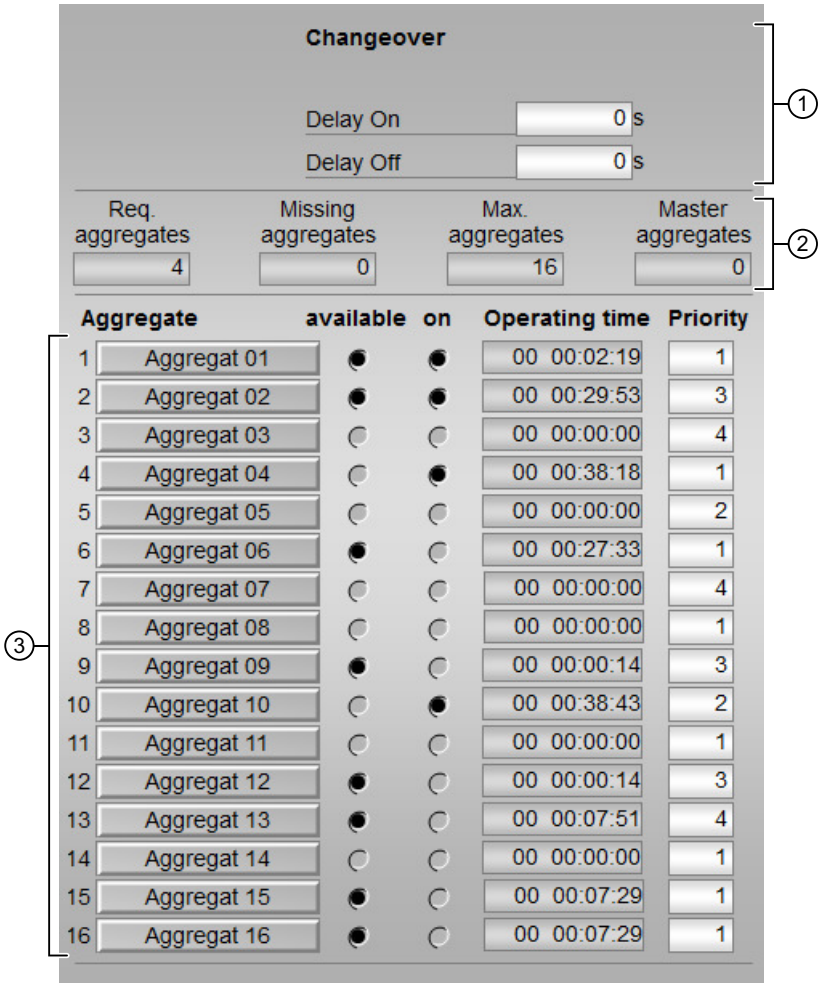
Aggr08 parameter view

Connection / disconnection with master unit:



Aggr16 parameter view

Priority-dependent connection / disconnection:



(1) Parameter for unit switchover

In this area, you change parameters and thereby influence the block.

The following parameters can be displayed:

- "Time remaining": Time until switchover of the master unit.
  The display is only visible in switchover mode "Master unit".

You can influence the following parameters:

- "Delay on": Delay time for connecting a unit.
  The display depends on the configuration in the ES.
- "Delay off": Delay time for disconnecting a unit.
  The display depends on the configuration in the ES.

(2) Statistics of the unit block

This area shows the most important parameters for this block with the current statistic:

- "Req. unit": Requested number of units.
- "Missing unit": Difference between requested and activated units. If more units are requested than can be started, they are shown as missing units at this point.
- "Max. unit": Max. number of units that can be switched by the block.
- "Master unit": Current master unit in "Master unit" switchover mode. 0 is displayed for all other switchover modes.

(3) Detail view of the unit states with navigation button for the standard view of the connected unit

The display of this area depends on the number of units configured.

Use this navigation button to open the standard view of the configured unit. The labeling of the navigation button depends on the configuration in the ES.

The display indicates whether a unit is available and/or switched on.

If an operating hour counter is available for the unit, the runtime is displayed in the form DD HH:MM:SS.

Priority: If priority-based startup/shutdown of units is activated, the operator can change the priority here depending on the operator control enable. All other switch modes do not display the priorities.

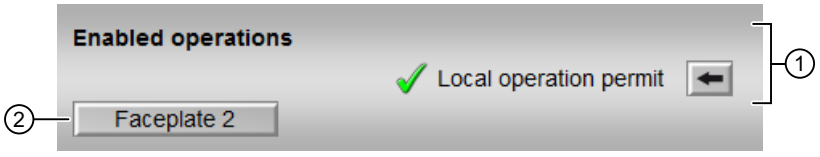
Changing the display of the text for "Unit"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the parameter (`AggrXRdyToStart`) to what you want to later see in runtime.

6.1.8.4 Aggr16/ Aggr08 preview

Aggr16 / Aggr08 preview



(1) Operator control enable

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: the OS operator cannot control this parameter due to the configured AS operator permissions (OS_Perm or OS1Perm).

The following enabled operations are shown here:

- "Local operating permission": Use the <-- button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(2) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

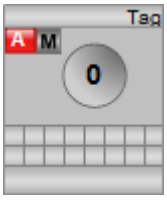
6.1.8.5 Aggr16/ Aggr08 block icon

Properties of the block icon for Agg08 / Aggr16

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of the alarm and warning limits as well as process control errors
- Operating modes
- Signal status, release for maintenance
- Memo display
- Status display

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Block icon for
	1	Aggr08
	1	Aggr16

You can find additional information on the block icon and the control options in the block icon in the following sections of the APL manual:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

Unit symbol with number of running units



Green: Number of running units > 0

Gray: Number of running units = 0

Status display of the individual units



Green: Unit running + is available

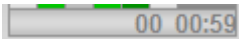
Dark green: Unit running + is not available

White: Unit not running + is available

Gray: Unit not running + is not available

Time remaining

Time remaining until switchover of the master unit in the format dd hh:mm



Gray characters: Time is running, but is not active

Black characters: Time is running + is active

No indicator: Time is not used.

6.2 OpAn16 – Operation of 16 parameters

6.2.1 Description of 'OpAn16'

Area of application

The block is used for the following applications:

- Operation of up to 16 analog values with operating limits

Mode of operation

The block checks the analog values entered in the faceplate (inputs `U1 ... U16`) for their limits (High, Low) and passes them on to the outputs `V1 ... V16` .

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'OpAn16' (Page 205)".

Status bit	Parameter
0	`BatchOcc`
1	`BatchEn`
2	`Reserve`
4	`OosLi.Value`

6.2.2 Functions of 'OpAn16'

- This block is used to operate up to 16 analog values using the OS faceplate.
- The values are monitored for plausibility of the high / low limits. Implausible values are limited to the configured high / low limits and an error is generated.
- The number of operable values is configured to an integer input parameter OpNum, number [1 ...16]. Deactivated channels are hidden in the faceplate. If OpNum is set to 0, all the outputs Vx are forced to 0.
- The block makes available a structure output with all 16 operating values. This reduces the interconnection work required when the same parameters have to be supplied to several target blocks.

Format configuration:

The user configures the format (displayed digits before and after decimal point) of each analog value at the parameters `FmtV1` to `FmtV16`.

Index	Format: predecimals, decimals
0	7,0
1	6,1
2	5,2
3	4,3
6	Scientific
Else, default	4,2

Further functions of ‚OpAn16'

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwBatchInfo`

You will find more information on this subject in the section AUTOHOTSPOT.

Calling additional faceplates

This block includes the APL standard function "Call additional faceplates".

Configurable reactions using the Feature parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in the "APL documentation" in the section "Configurable functions with the Feature I/O". The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
...	...
24	1 = "Local" operator authorization is active
...	...

Operator permissions

This block includes the APL standard function "Operator permissions".

The block has the following authorizations for the `OS_Perm` parameter:

Status bit	Parameter
0	Reserve
1	1= Operator can change the value 1

Status bit	Parameter
2	1= Operator can change the value 2
3	1= Operator can change the value 3
4	1= Operator can change the value 4
5	1= Operator can change the value 5
6	1= Operator can change the value 6
7	1= Operator can change the value 7
8	1= Operator can change the value 8
9	1= Operator can change the value 9
10	1= Operator can change the value 10
11	1= Operator can change the value 11
12	1= Operator can change the value 12
13	1= Operator can change the value 13
14	1= Operator can change the value 14
15	1= Operator can change the value 15
16	1= Operator can change the value 16
...	...

Overview of error numbers

The `ErrorNum` connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.
42	Low / high limit is not plausible `Ux_LL > Ux_HL` for `x=1…16`
43	The number of operating values is not plausible. `0 > OpNum > 16`

6.2.3 Connections of 'OpAn16'

Input parameters

Parameter	Description	Type	Default
`CmData`	Control and status information for this CM	`BlUdtCmData`	
`OvwBatchInfo`	1=Overwrite CmData batch data with BatchX input data	`BOOL`	
`BatchEnLi`	Batch is enabled (overwritable CmData element)	`BOOL`	
`BatchOccLi`	Occupied by batch (overwritable CmData element)	`BOOL`	
`BatchIdLi`	Batch ID (overwritable CmData element)	`DWORD`	

Parameter	Description	Type	Default
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
OpNum	Number of operable analog values	INT	16
U1_UL U1_LL	Upper Limit 1 Lower Limit 1	REAL	
…			
U16_UL U16_LL	Upper Limit 16 Lower Limit 16	REAL	
FmtV1 … FmtV16	Display format 1 ...16	INT	
U1 … U16	Analog value 1 ... 16	REAL	
U1_Unit … U16_Unit	Units for Analog value 1 ...16	INT	
SelFp1	Select Faceplate 1	ANY	
SelFp2	Select Faceplate 2	ANY	
OS_Perm	Operator Permission	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

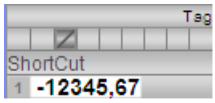
Output parameters

Parameter	Description	Type	Default
V1 … V16	Analog output value 1 ...16	REAL	
VStruct	Analog output value 1 ...16 as STRUCT	STRUCT • V1 : REAL • … • V16 : REAL	
BatchEn	Enable remote operation of controller by Batch recipe	BOOL	
BatchOcc	Occupied by Batch, enables second message	BOOL	
BatchID	Current Batch ID	DWORD	
BatchName	Current Batch name	STRING32	
StepNo	Batch step number	DWORD	
OS_PermOut	Operator permission: output 1 for OS	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
OpSt_Out	Enabled operator stations	DWORD	

Parameter	Description	Type	Default
ST_Worst	Enabled operator stations	DWORD	
Status1	Status1 data	DWORD	

6.2.4 Operator control and monitoring

6.2.4.1 Symbol



Instance name
Status bar
Connection identifier for Index (1 ...16)
Index (1...16) | Value | Unit

The symbol contains the following information:

Position	Content	
1	Empty	
2	Empty	
3	Operating mode	
		Automatic
		Manual
		"Out of service"
4...8	Empty	
9		Memo is available

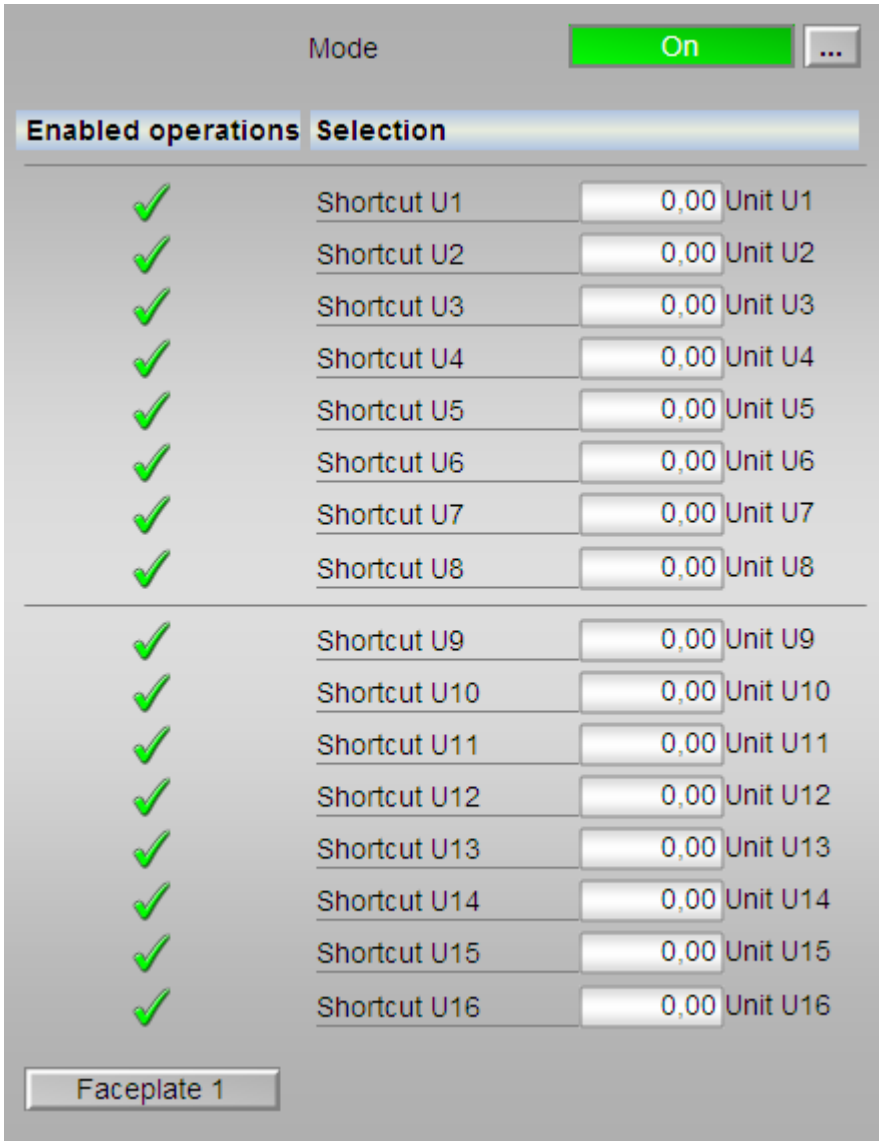
6.2.4.2 Faceplate

Views of the OpA16 block

The block OpA16 provides the following views:

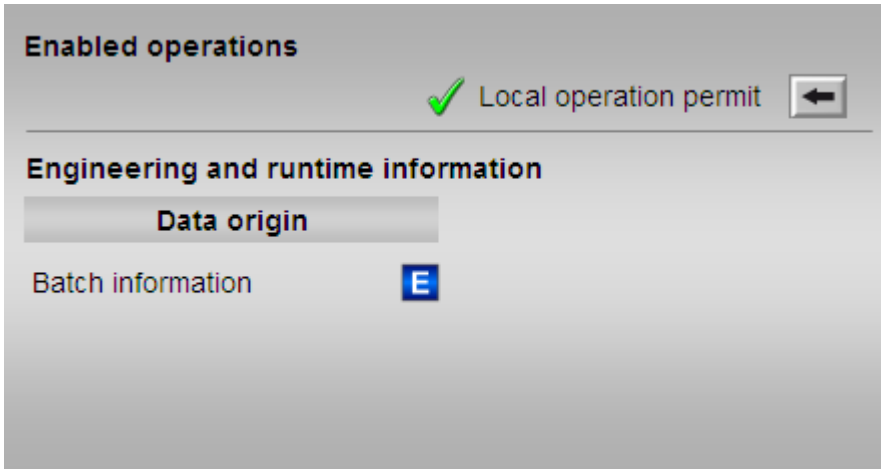
Symbol	View	
	Standard view	See below.
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

Standard view of the OpA16



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows you the current values 1 ...16 with their connection identifiers and units. The values can be changed if the required permission is available.
(2)	**Navigation button to the standard view of any faceplate**
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp1` in the CFC.

Preview of OpA16



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the `OS_PERM` parameter of this block. The operator control enables shown in the screenshot are possible here. • Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions
(2)	**Engineering and runtime information**
	This section contains information from the status word `Status1`
	• The "Data origin" column shows the setting of the overwrite signals. – Batch information `OvwBatchInfo`
(3)	**Navigation button to the standard view of any faceplate**
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp2` in the CFC.

6.3 OpEnum – Operation of enumeration parameters

6.3.1 Description of 'OpEnum'

Area of application and operating principle

The block is used for the following applications:

- Operation of an analog value through selection of the value from an enumeration. Value selection is carried out by using the drop-down box in the OS faceplate in which the enumeration texts are displayed.
- Alternatively the interconnectable input parameter `InLi` can be used for the value selection.

Configuration

- Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).
- Configure the desired enumeration at the output Out. The enumeration together with the values is defined in the shared declarations of the S7 project.

Startup characteristics:

Not available

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'OpEnum' (Page 212)".

Status bit	Parameter
0	`BatchOcc`
1	`BatchEn`
2	`InLiOp.Value`
4	`OosLi.Value`

6.3.2 Functions of 'OpEnum'

- During the execution time the operated or interconnected value is available at the output `Out` . The text configured in the enumeration is displayed in the faceplate as well as in CFC online.
- Selection between the operator input `InOp` and the interconnectable input `InLi` is carried out through the selector input `InLiOp`.
  - `InLiOp = 1`
    Interconnected value is transferred at the output `Out`
  - `InLiOp = 0`
    Operated value is transferred at the output `Out`
- Through the input `Bumpless = 1` tracking of the operator input `InOp` is activated at the interconnected value `InLi` when `InLiOp = 1`.

Further functions of OpEnum

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwBatchInfo`

You will find more information on this subject in the section AUTOHOTSPOT.

Call additional faceplates

This block includes the APL standard function "Call additional faceplates".

Forming the signal status for blocks

This block includes the APL standard function "Form and output signal status for blocks".

The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `InLi.ST` (only if currently interconnected)
- `InLiOp.ST`

Configurable reactions using the Feature parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in the *"APL documentation" in the section "Configurable response using the Feature I/O"*. The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
...	...
24	1 = "Local" operator permission is active
...	...

Operator permissions

This block includes the APL standard function "Operator permissions".

The block has the following permissions for the `OS_Perm` parameter:

Status bit	Parameter
0	1= Operator can change the Bumpless flag
1	1= Operator can change the enumeration value
2	Reserve
...	...

6.3.3 Connections of 'OpEnum'

Input parameters

Parameter	Description	Type	Default
`CmData`	Control and status information for this CM	`BlUdtCmData`	
`OvwBatchInfo`	1=Overwrite CmData batch data with BatchX input data	`BOOL`	
`BatchEnLi`	Batch is enabled (overwritable CmData element)	`BOOL`	
`BatchOccLi`	Occupied by batch (overwritable CmData element)	`BOOL`	
`BatchIdLi`	Batch ID (overwritable CmData element)	`DWORD`	
`BatchNameLi`	Batch name (overwritable CmData element)	`STRING[32]`	
`StepNoLi`	Batch step number (overwritable CmData element)	`DWORD`	
`Bumpless`	1=Bumpless changeover	`BOOL`	
`InLiOp`	1=Link/Auto, 0=Manual: Input value selection	`DigVal`	
`InOp`	Operator input	`INT`	
`InLi`	Linkable input	`INT`	
`SelFp1`	Select Faceplate 1	`ANY`	
`SelFp2`	Select Faceplate 2	`ANY`	
`OS_Perm`	Operator Permission	`STRUCT`	
`OpSt_In`	Enabled operator stations	`DWORD`	
`Feature`	Feature collection	`STRUCT`	

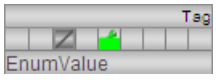
Output parameters

Parameter	Description	Type	Default
`Out`	Selected enumeration value	`INT`	
`BatchEn`	Enable remote operation of controller by Batch recipe	`BOOL`	
`BatchOcc`	Occupied by Batch, enables second message	`BOOL`	
`BatchID`	Current Batch ID	`DWORD`	
`BatchName`	Current Batch name	`STRING32`	

Parameter	Description	Type	Default
StepNo	Batch step number	DWORD	
OS_PermOut	Operator permission: output 1 for OS	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
OpSt_Out	Enabled operator stations	DWORD	
ST_Worst	Enabled operator stations	DWORD	
Status1	Status1 data	DWORD	
ST_Worst	Worst signal status	BYTE	16#80

6.3.4 Operator control and monitoring

6.3.4.1 Symbol



Instance name
Status bar
Current value as a text from the enumeration

The status bar contains the following symbol information sequentially from left to right:

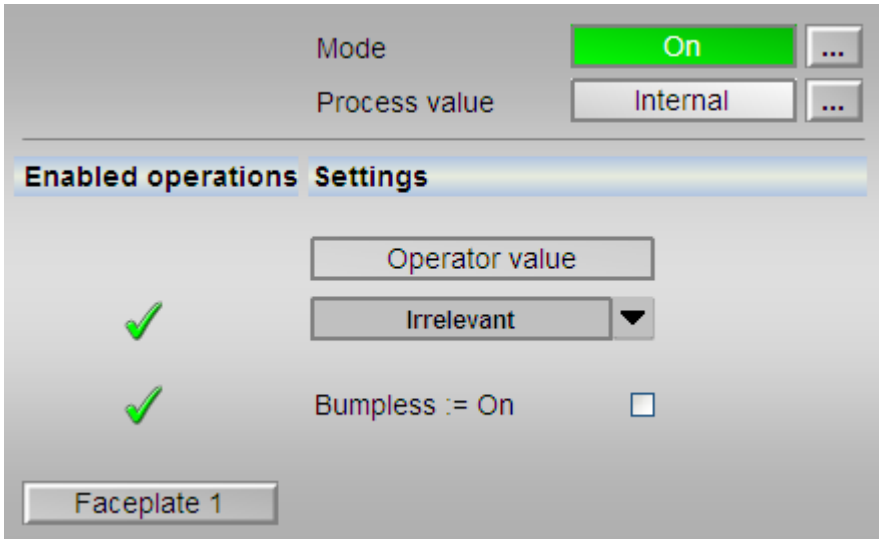
Position	Content	
1	Tag name	
2	Empty	
3	Operating mode	
		Automatic
		Manual
		"Out of service"
4	Empty	
5		Enable maintenance operation
6...8	Empty	
9		Memo is available

6.3.4.2 Faceplate

The block OpEnum provides the following views:

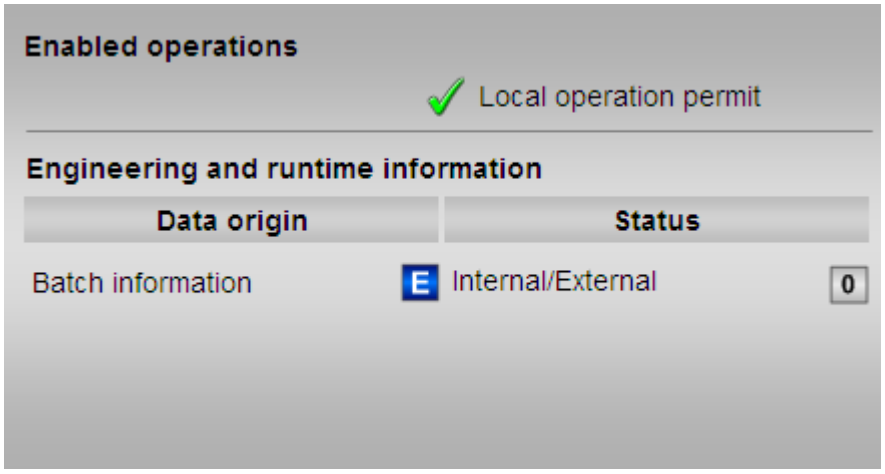
Symbol	View	
	Standard view	See below.
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

Standard view of OpEnum



Explanation of the screen shot	
(1)	**Operator value**
	This section shows you the current operating values with the designations from the enumeration. A value can be selected from the list if the required permission is available.
(2)	**'Bumpless' mode**
	The parameter `Bumpless` can be activated / deactivated here if the corresponding operator permission is available.
(3)	**Navigation button to the standard view of any faceplate**
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp1` in the CFC.

Preview of OpEnum



Explanation of the screen shot	
(1)	**Operator value**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the `OS_PERM` parameter of this block. The operator control enables shown in the screenshot are possible here. • Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions
(2)	**Engineering and runtime information**
	This section contains information from the status word `Status1`
	• The "Data origin" column shows the setting of the overwrite signals. – Batch information `OvwBatchInfo` • 'States' column displays the state of the input parameter `InLiOp (0 / 1)` – `InLiOp = 0` The faceplate operating value is used – `InLiOp = 1` The interconnectable input parameter is used
(3)	**Navigation button to the standard view of any faceplate**
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp1` in the CFC.

6.4 ParaCtrl / ParaMem - Block for handling small numbers of parameters and parameter sets

6.4.1 ParaCtrl / ParaMem description

Object name (type + number) and family

ParaCtrl:

Type + number: FB 1185

Family: Operate

Number of process objects: 10 PO

ParaMem:

Type + number: FB 1186

Family: Operate

Number of process objects: 1 PO

Area of application for ParaCtrl and ParaMem

The block is used for the following application:

- Display and edit parameters
- Managing parameter sets

Operating principle of ParaCtrl and ParaMem

The parameter blocks ParaCtrl and ParaMem are used to save smaller quantities of parameters and parameter sets in the AS and to make the parameters operable for the operator. Individual parameter sets can be exported as .csv files or imported from .csv files.

The number of parameters and parameter sets can be scaled. Up to 100 parameters per data record and any number of data records are supported.

Note

The specification "100 parameters per data record" is not a fixed limit. We do, however, strongly recommend that you do not exceed this number for reasons of operability and screen update times. The absolute limit is 1000 parameters or 100 ParaMem blocks.

Note

The number of supported parameter sets depends on the number and size of the used data blocks.

ParaCtrl

The ParaCtrl blocks controls the ParaMem blocks and provides the faceplate for operation. The parameters of the ParaMem blocks are displayed and operated in the parameter list of the faceplate.

The ParaCtrl block can be connected with a data block which is used for saving parameter sets.

ParaMem

The ParaMem block saves 10 parameters as well as their operating limits and units. There is a process value input for each parameter which can be displayed in the standard view of the ParaCtrl block.

The ParaMem block can be connected with a data block which is used for saving parameter sets.

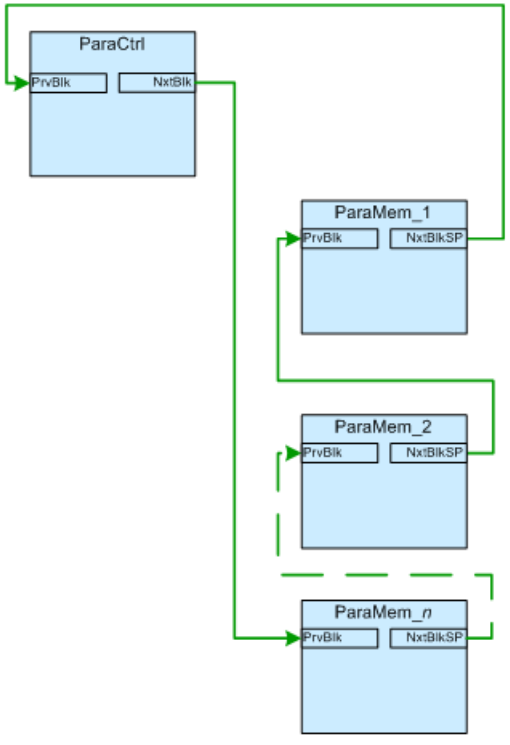
Configuration

Install the block in a cyclic interrupt OB (OB30 to OB38) in the CFC Editor. The block is also installed automatically in the startup OB (OB100).

Interconnection of ParaCtrl block and ParaMem blocks:

The ParaCtrl block must be interconnected with the ParaMem blocks as shown in the figure below to form a functioning unit. The blocks may be distributed across several AS.

Interconnection of ParaCtrl block and ParaMem blocks:



If the blocks are interconnected as shown above, the parameters of the 'ParaMem_1' block are shown at the top of the parameter list followed by the parameters of the 'ParaMem_2' block, etc. The run sequence of the parameter blocks, however, must be exactly the opposite in order for

the ParaMem blocks to be executed within one cycle. In other words, the last "ParaMem_n" block first and the "ParaMem_1" block last.

Setting parameters and parameter set names

There are general settings and parameter-specific settings when using parameter sets.

General settings are:

- Name of the active parameter set
- Name of the edited parameter set
- Number of parameter sets

Parameter-specific settings are:

- Parameter name
- Parameter value
- Parameter unit
- Parameter operating limits
- Parameter process value, if applicable

All settings are instance-specific and are therefore made at the AS blocks. The general settings are made at the ParaCtrl block; the parameter-specific settings are made at the corresponding ParaMem blocks.

Setting parameter set names

Because the user sets the number of parameter sets, the parameter set names are saved with the help of "enumerations". The assignment of the enumerations to the parameter sets takes place at the corresponding block connectors. The enumeration with the names of the active parameters is assigned at the `ActSetMan` block connector. The enumeration with the names of the edited parameters is assigned at the `ViewSetMan` block connector. It makes sense to use the same enumeration for both I/Os in most cases.

If no enumeration was defined and assigned, a substitute text is formed in the OS:

- "Parameter set" & number of the active parameter set
- E.g.: Parameter set 3

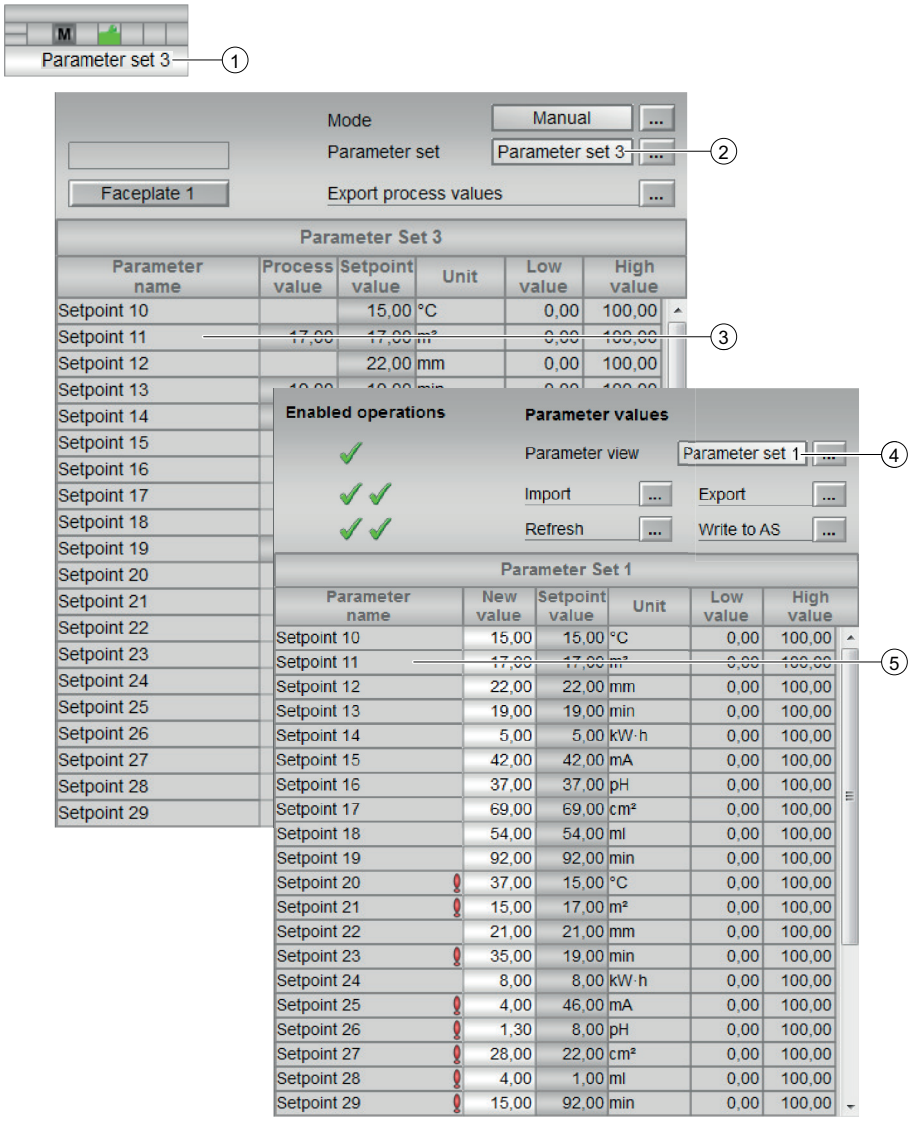
Parameter-specific settings

The ParaMem block has different block connectors with the corresponding attributes for each parameter to make all parameter-specific settings:

- `SPn_Op`:
  Current parameter value entered by the operator during runtime. If several parameter sets are used, the value of the ParaMem block is copied to the corresponding data block. The name of the parameter is configured at the "ID" attribute of the I/O.
- `SPn_OpScale`:
  The high and low operating limit is configured at the OpScale. The limits cannot be changed from the OS.

- `SPn_Unit:`
  As with the APL, you can either set the unit with an integer value or configure a text at the "Unit" attribute.
- `PVn:`
  Here you can connect the process value associated with the parameter (for example, an "actual value" of a controller if the parameter specifies the "Setpoint").

The figure below shows how the instance-specific data at the ParaCtrl and ParaMem blocks are visualized.



(1) The name of the active parameter set is displayed at the block icon.
The text is taken from the enumeration which is assigned to block connector `ActSetMan`.

(2) The same text as at the block icon is displayed in standard view.

(3) The settings of the corresponding `ParaMem` block are displayed in each parameter line. The process value is only displayed when a process value is connected to the ParaMem block.

(4) The name of the parameter set that is being edited is displayed in the parameter view. The text is taken from the enumeration which is assigned to block connector `ViewSetMan`.

(5) As in the standard view, the settings of the corresponding `ParaMem` block are displayed in each parameter line. The "new parameter value" is displayed instead of the process value. The new parameter values can be changed by the operator in the parameter view. The changed values are only written to the AS with "Write AS".

File dialog

Process values can be exported; parameters can be imported and exported. Only explicitly enabled drives and folders can be used for this purpose. Detailed information is available in the documentation of the File dialog (Page 56).

Setting parameters and parameter set names

There are general settings and parameter-specific settings when using parameter sets.

General settings are:

- Name of the active parameter set
- Name of the edited parameter set
- Number of parameter sets

Parameter-specific settings are:

- Parameter name
- Parameter value
- Parameter unit
- Parameter operating limits
- Parameter process value, if applicable

All settings are instance-specific and are therefore made at the AS blocks. The general settings are made at the ParaCtrl block; the parameter-specific settings are made at the corresponding ParaMem blocks.

Setting parameter set names

Because the user sets the number of parameter sets, the parameter set names are saved with the help of "enumerations". The assignment of the enumerations to the parameter sets takes place at the corresponding block connectors. The enumeration with the names of the active parameters is assigned at the `ActSetMan` block connector. The enumeration with the names of the edited parameters is assigned at the `ViewSetMan` block connector. It makes sense to use the same enumeration for both I/Os in most cases.

If no enumeration was defined and assigned, a substitute text is formed in the OS:

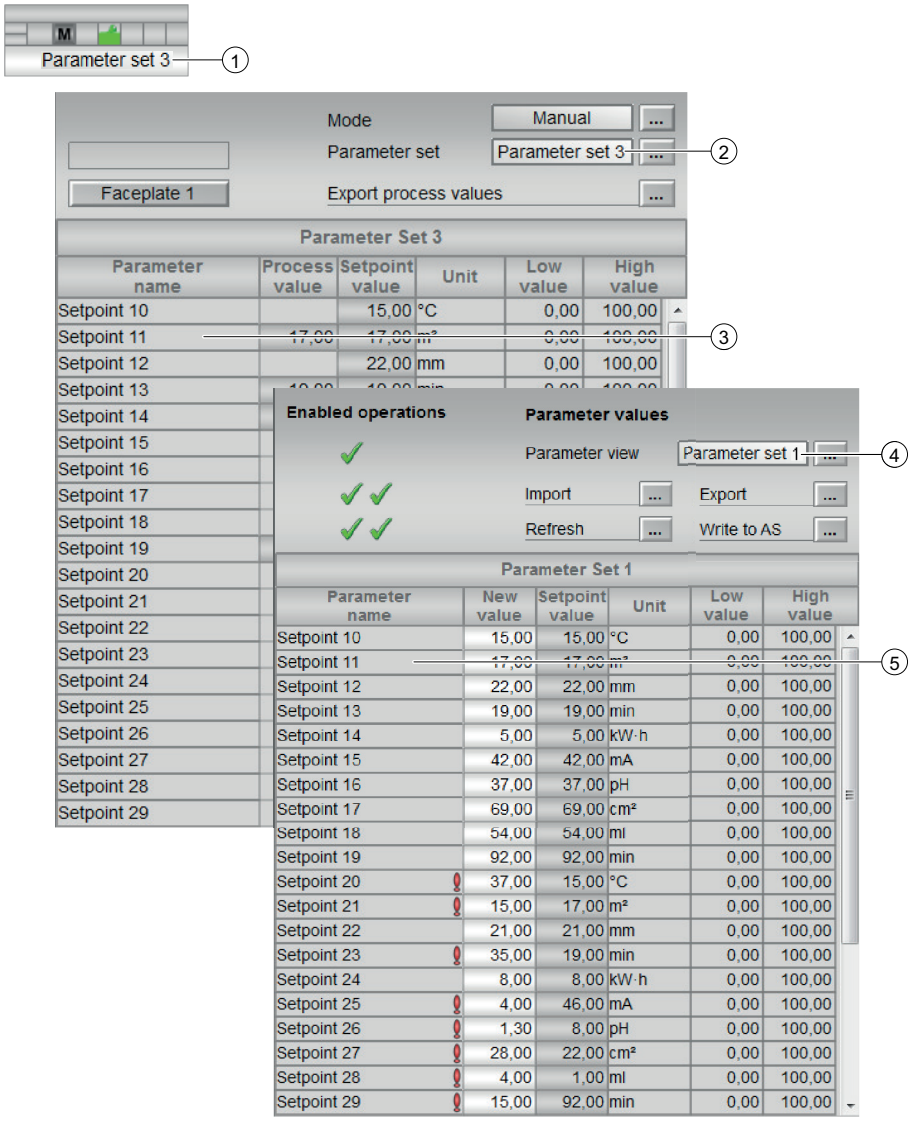
- "Parameter set" & number of the active parameter set
- E.g.: Parameter set 3

Parameter-specific settings

The ParaMem block has different block connectors with the corresponding attributes for each parameter to make all parameter-specific settings:

- `SPn_Op:`
  Current parameter value entered by the operator during runtime. If several parameter sets are used, the value of the ParaMem block is copied to the corresponding data block. The name of the parameter is configured at the "ID" attribute of the I/O.
- `SPn_OpScale:`
  The high and low operating limit is configured at the OpScale. The limits cannot be changed from the OS.
- `SPn_Unit:`
  As with the APL, you can either set the unit with an integer value or configure a text at the "Unit" attribute.
- `PVn:`
  Here you can connect the process value associated with the parameter (for example, an "actual value" of a controller if the parameter specifies the "Setpoint").

The figure below shows how the instance-specific data at the ParaCtrl and ParaMem blocks are visualized.



(1) The name of the active parameter set is displayed at the block icon.
The text is taken from the enumeration which is assigned to block connector `ActSetMan`.

(2) The same text as at the block icon is displayed in standard view.

(3) The settings of the corresponding `ParaMem` block are displayed in each parameter line. The process value is only displayed when a process value is connected to the ParaMem block.

(4) The name of the parameter set that is being edited is displayed in the parameter view.
The text is taken from the enumeration which is assigned to block connector `ViewSetMan`.

(5) As in the standard view, the settings of the corresponding ParaMem block are displayed in each parameter line. The "new parameter value" is displayed instead of the process value. The new parameter values can be changed by the operator in the parameter view. The changed values are only written to the AS with "Write AS".

File dialog

Process values can be exported; parameters can be imported and exported. Only explicitly enabled drives and folders can be used for this purpose. Detailed information is available in the documentation of the File dialog (Page 56).

Startup characteristics

ParaCtrl

Depending on the Feature.Bit0, the current operating mode is retained for the block type or the block is started manually. Regardless of Feature.Bit0, the `ParaCtrl` block outputs the `EnChgParVal=0` signal during the startup cycles `RunUpCyc` via the `ParaMem` loop and thus influences the behavior of the ParaMem blocks.

ParaMem

The startup characteristics of the ParaMem block are determined by the fact whether or not a data block is used for saving the parameter values. If no data block is used, the `ParaMem` block does not have special startup characteristics.

If a data block is used for saving the parameter values, the parameter values are read during the `RunUpCyc` startup cycles and written to the parameter inputs. The parameter values are written from the parameter inputs to the data block when the startup cycles are complete. The behavior during startup of the `ParaCtrl` block is influenced by the `EnChgParVal` signal regardless of whether a data block is used or not. For `EnChgParVal=0`, the parameter values are also read from the data block and written to the parameter inputs.

Called blocks

ParaCtrl

- FC369 (SelST16)
- SFC6 (RD_SINFO)
- SFC24(TestDB)

ParaMem

- FC369 (SelST16)
- SFC6 (RD_SINFO)
- SFC24(TEST_DB)

Calling OBs

ParaCtrl

- Cyclic interrupt OB in which the block is installed (e.g. OB32)
- OB100 (see Startup characteristics)

ParaMem

- Cyclic interrupt OB in which the block is installed (e.g. OB32)
- OB100 (see Startup characteristics)

Status word assignment for the parameter `Status1` of ParaCtrl

Status bit	Parameter
0	`Occupied`
1	`BatchEn`
2	–
3	`OosAct.Value`
4	`OosLi.Value`
5	`AutAct.Value`
6	–
7	`ManAct.Value`
8	`Control error`
9	`Data error`
10...31	–

Status word assignment for the parameter `Status1` of ParaMem

Status bit	Parameter
0	1 = Process value PV0 is interconnected
1	1 = Process value PV1 is interconnected
2	1 = Process value PV2 is interconnected
3	1 = Process value PV3 is interconnected
4	1 = Process value PV4 is interconnected
5	1 = Process value PV5 is interconnected
6	1 = Process value PV6 is interconnected
7	1 = Process value PV7 is interconnected
8	1 = Process value PV8 is interconnected
9	1 = Process value PV9 is interconnected
10...31	-

6.4.2 ParaCtrl / ParaMem operating modes

ParaCtrl/ParaMem operating modes

The block can be operated using the following modes:

- Automatic mode
- Manual mode
- Out of service

General information on the operating modes is available in the section *Operating modes of the blocks* in the *APL Function manual*.

6.4.3 ParaCtrl / ParaMem functions

Use of data blocks to save data records

This block has the function "Use of data blocks for storing data records".

If you want to use more than one data records, use data blocks to save the parameters. The ParaCtrl block and the ParaMem block both have a Feature bit (`Feature`.Bit5, 1 = Data block connected) which informs the block whether or not it is connected with a data block.

The data block must be sufficiently large to save all parameters. It is best to create a data block with an array of Real values. The number of Real values must equal the number of parameters. A ParaMem block has 10 parameters.

Example:

A ParaCtrl block with 3 ParaMem blocks is used and 3 parameter sets are to be stored. The data block should be available centrally on the ParaCtrl block.

Each of the ParaMem blocks has 10 parameters. Thus, a parameter set contains 30 parameters (30 values). To store 3 parameter sets, the array must contain at least 90 values in the data block.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DB_VAR	ARRAY[0..89]	0.000000e+000	Parameter data
*4.0		REAL		
=360.0		END_STRUCT		

Figure 6-1 Data block for saving parameters

If more data records are addressed than memory is available, an error is displayed at the error output of the block with the corresponding error number. The block does not read and write parameter values in this case.

Note

Interconnection of the data block

If only one array is defined in the data block, the `DBPointer` I/O can be connected directly to the data block. If several arrays are defined in the data block, the DBPointer connection must be connected to the corresponding array.

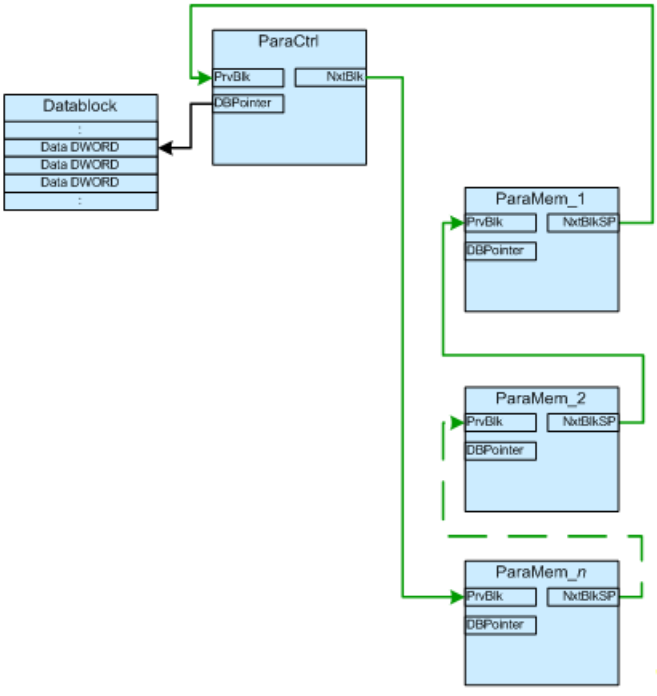
The interconnection succeeded in the CFC via the "Interconnection to operand..." command.

Application examples for interconnection of the ParaCtrl and ParaMem block with data blocks

The behavior of ParaCtrl and ParaMem blocks allows different interconnections with data blocks. The application cases are shown below:

Situation 1: ParaCtrl block, ParaMem blocks and data block in one AS

All blocks are installed in one AS so that the central data block can be connected to the central ParaCtrl block. Feature.Bit5 (Data block connected) is set to 1 at the ParaCtrl block and to 0 at all ParaMem blocks.



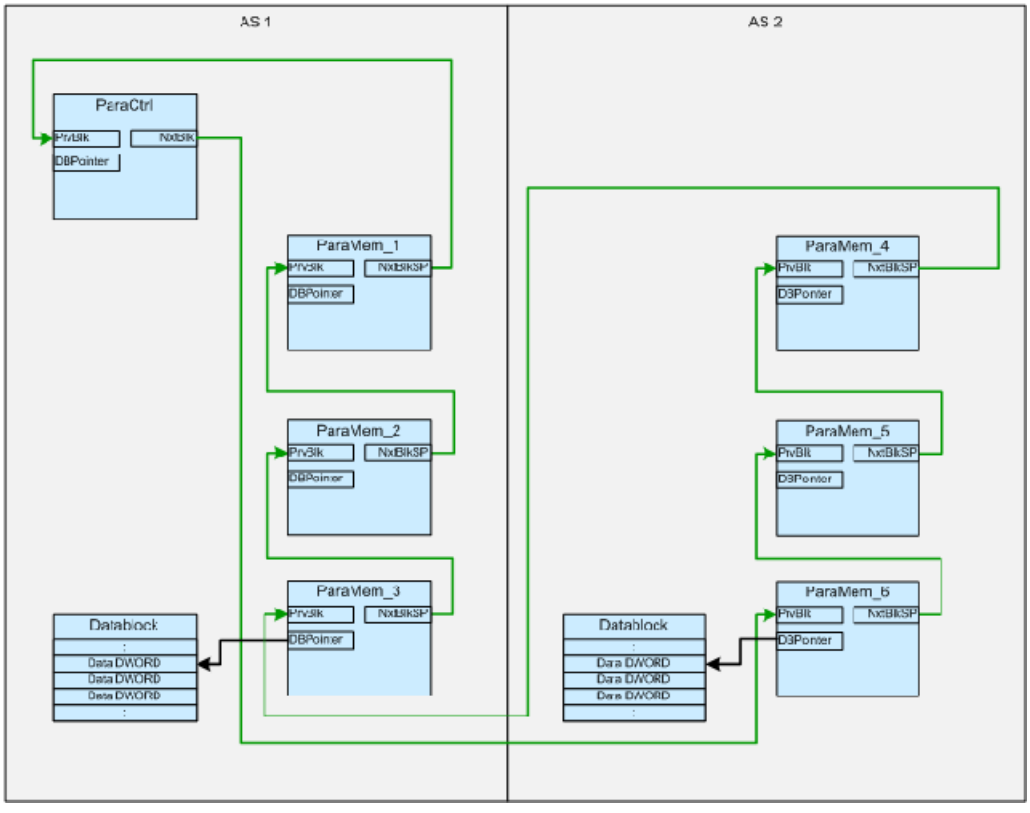
Note

Due to the way in which data storage is organized in the data block, it is not possible to increase the number of parameter sets (NumSet) later during operation.

Increasing the number of parameter sets later would have the existing parameter values show up at the wrong location, which means in the wrong parameter set at the wrong parameter.

Situation 2: ParaCtrl block, ParaMem blocks and data block distributed across several AS

If the ParaMem blocks are distributed across more than one AS, at least one data block is required for each AS. The data block must be connected with the first`ParaMem` block in the loop in each AS. The`Feature.Bit5`(Data block connected) of the `ParaCtrl` block is set to 0. The`Featrue.Bit5`of each first `ParaMem` block (in the example `ParaMem_3`and`ParaMem_6`) in the AS is set to 1 and to 0 for all other `ParaMem` blocks.



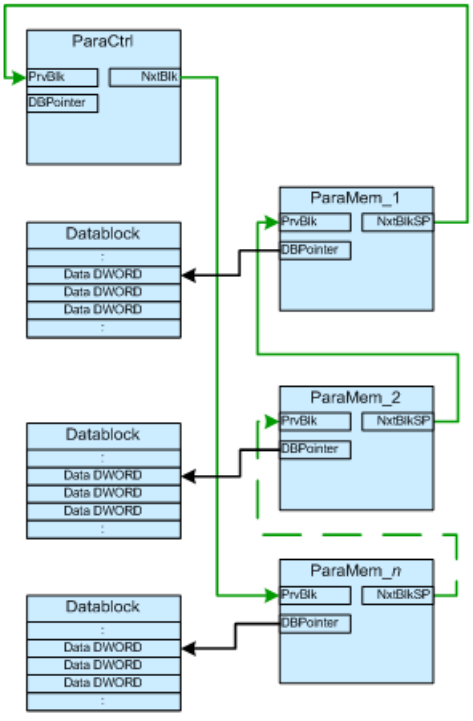
Note

Due to the way in which data storage is organized in the data block, it is not possible to increase the number of parameter sets (`NumSet`) later during operation.

Increasing the number of parameter sets later would have the existing parameter values show up at the wrong location, which means in the wrong parameter set at the wrong parameter.

Situation 3: Each ParaMem block has its own data block:

If each ParaMem block is connected to its own data block, the `ParaMem` blocks can also be distributed across several AS. The `Feature.Bit5` (Data block connected) of the `ParaCtrl` block is set to 0. `Featrue.Bit5` of each `ParaMem` block is set to 1. If the data block is sufficiently dimensioned, this configuration offers the advantage that the number of data records can be increased in runtime.



Selecting a unit of measure

This block provides the standard function 'Selecting a unit of measure'.

Opening additional faceplates

This block provides the standard APL function calling additional faceplates. Only one additional faceplate can be called up using `SelFp1` and `SelFp2`.

Operator control permissions ParaCtrl

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can switch to "automatic mode"
1	1 = Operator can switch to "manual mode"
2	1 = Operator can switch to local mode
3	1 = Operator can change the active parameter set
4	1 = Operator can change view parameter set
5	1 = Operator can copy parameter values to the AS
6	1 = Operator can import parameter values
7	1 = Operator can export parameter values
8	1 = Operators can export process values
9	1 = Operator can change parameter values
10	1 = Operator can update parameter values
11 - 31	Not used

Configurable reactions of ParaCtrl using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
2	See APL function: "Reset the commands for switching over the mode"
4	See APL function: "Set switch or button mode"
5	1 = Data block connected
6	1 = Values of the active parameter set cannot be changed
17	See APL function: "Bumpless switchover to 'Automatic' mode for valves, motors, and dosers"
21	See APL function: "Bumpless switchover to automatic mode for operator only"
24	See APL function: "Enable local operating permission"

Configurable reactions of ParaMem using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
5	1 = Data block connected

6.4.4 ParaCtrl / ParaMem error handling

The ParaCtrl and ParaMem form a shared function possibly in connection with data blocks. The individual blocks may be distributed across several AS. This means different errors can be detected. If an error occurs, the block displays an error number. The error number is only displayed at the block on which the error has occurred.

If a data block connected to a ParaCtrl block is too small for three ParaMem blocks, for example, but large enough for two ParaMem blocks, the ParaCtrl block and the first two ParaMem blocks would not display an error number. The error number would only be displayed at the third ParaMem block (and any additional ParaMem blocks, if present), because it would be unable to store its data.

The ParaCtrl block also has a 'DataError' output. This output indicates whether any block in the 'ParaMem loop' has a data block error or if the 'ParaMem loop' is open.

The error numbers with their causes are described below:

Overview of error numbers

The `ErrorNum` I/O can be used to output various error numbers:

Error number	Meaning of the error number
00	No error
40	ParaMem loop not closed There are different reasons why the ParaMem loop was not closed. It may be an incorrect configuration, for example, or the communication may be faulty with distribution across several AS. The ParaCtrl and ParaMem blocks therefore monitor the loop with a 'toggle bit' integrated into the loop structure. If the connection fails, the parameters cannot be changed (change parameter value, change active parameter set). 'Data Error' is displayed in the standard view of the faceplate. The icon 'Process-related error' is displayed at the block icon. The block output DataError is set at the ParaCtrl (it can be used to generate a message, for example).
41	Data block error If the data block error is displayed, you are dealing with an incorrect configuration. It may have the different reasons: • ParaCtrl and ParaMem: Feature.Bit5 is set and the connected data block is too small or read-only • ParaCtrl and ParaMem: Feature.Bit5 is set and no data block is connected. • ParaMem only: The block is to store its data in a data block connected to a previous block but there is not enough space in the data block.
51	Incorrect control (e.g. "Manual" and "Automatic" at the same time) "Invalid signal" is displayed in the standard view of the faceplate.

6.4.5 ParaCtrl / ParaMem messaging

Message behavior

ParaCtrl

- Not message-capable

ParaMem

- Not message-capable

6.4.6 ParaCtrl / ParaMem I/Os

Input parameters ParaCtrl

Parameter	Description	Data type	Default
ActSetAut *	Active parameter set by automatic	INT	1
ActSetMan *	Active parameter set by operator	INT	1
DBPointer	Pointer to data block	ANY	
LoopTimeOut	Watchdog time loop	REAL	5.0
MsgEvId	Message event ID	DWORD	16#000000D9
NumSet *	Number of parameter sets	INT	1
PrvBlk	Connection from pervious block (Para-Mem)	STRUCT	
PrvBlk.ActSet	Parameter set output values	INT	0
PrvBlk.DB_Error	DB error at any ParaMem block	BOOL	0
PrvBlk.EnChgParVal	Enable changing of parameter values	BOOL	0
PrvBlk.LoopAlive	Test signal ParaMem loop closed	BOOL	0
PrvBlk.LoopClosed	ParaMem loop closed	BOOL	0
PrvBlk.LoopOk	Loop alive signal from ParaMem blocks	BOOL	0
PrvBlk.NumParaDB	Number of parameter memory blocks at current data block	INT	0
PrvBlk.NumParaMem	Number of parameter memory blocks	INT	0
PrvBlk.NumParaSet	Number of parameter sets	INT	0
PrvBlk.ParaCtrlDB_dbno	Parameter control block data block number	WORD	16#0000
PrvBlk.PrvDB_adr	Previous connected data block address in DB	INT	0
PrvBlk.PrvDB_dbno	Previous connected data block DB number	WORD	16#0000
PrvBlk.PrvDB_length	Previous connected data block DB length	INT	0
PrvBlk.ST_Worst	Worst signal status	BYTE	16#80

Parameter	Description	Data type	Default
PrvBlk.ViewSet	Parameter set at input values	INT	0
ViewSetMan *	View parameter set by operator	INT	1

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters ParaCtrl

Parameter	Description	Data type	Default
ActSet	Active parameter set	INT	1
DataError	ParaMem loop or data block error	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
NxtBlk	Connection to next block (ParaMem)	STRUCT	
NxtBlk.ActSet	Parameter set output values	INT	0
NxtBlk.DB_Error	DB error at any ParaMem block	BOOL	0
NxtBlk.EnChgParVal	Enable changing of parameter values	BOOL	0
NxtBlk.LoopAlive	Test signal ParaMem loop closed	BOOL	0
NxtBlk.LoopClosed	ParaMem loop closed	BOOL	0
NxtBlk.LoopOk	Loop alive signal from ParaMem blocks	BOOL	0
NxtBlk.NumParaDB	Number of parameter memory blocks at current data block	INT	0
NxtBlk.NumParaMem	Number of parameter memory blocks	INT	0
NxtBlk.NumParaSet	Number of parameter sets	INT	0
NxtBlk.ParaCtrlDB_dbno	Parameter control block data block number	WORD	16#0000
NxtBlk.PrvDB_adr	Previous connected data block address in DB	INT	0
NxtBlk.PrvDB_dbno	Previous connected data block DB number	WORD	16#0000
NxtBlk.PrvDB_length	Previous connected data block DB length	INT	0
NxtBlk.ST_Worst	Worst signal status	BYTE	16#80
NxtBlk.ViewSet	Parameter set at input values	INT	0

Input parameters ParaMem

Parameter	Description	Data type	Default
DBPointer	Pointer to data block	ANY	
LoopTimeOut	Watchdog time loop	REAL	5.0
MsgEvId	Message event ID	DWORD	16#000000DA
PrvBlk	Connection from pervious block (ParaMem or ParaCtrl)	STRUCT	
PrvBlk.ActSet	Parameter set output values	INT	0

Parameter	Description	Data type	Default
PrvBlk.DB_Error	DB error at any ParaMem block	BOOL	0
PrvBlk.EnChgParVal	Enable changing of parameter values	BOOL	0
PrvBlk.LoopAlive	Test signal ParaMem loop closed	BOOL	0
PrvBlk.LoopClosed	ParaMem loop closed	BOOL	0
PrvBlk.LoopOk	Loop alive signal from ParaMem blocks	BOOL	0
PrvBlk.NumParaDB	Number of parameter memory blocks at current data block	INT	0
PrvBlk.NumParaMem	Number of parameter memory blocks	INT	0
PrvBlk.NumParaSet	Number of parameter sets	INT	0
PrvBlk.ParaCtrlDB_dbno	Parameter control block data block number	WORD	16#0000
PrvBlk.PrvDB_adr	Previous connected data block address in DB	INT	0
PrvBlk.PrvDB_dbno	Previous connected data block DB number	WORD	16#0000
PrvBlk.PrvDB_length	Previous connected data block DB length	INT	0
PrvBlk.ST_Worst	Worst signal status	BYTE	16#80
PrvBlk.ViewSet	Parameter set at input values	INT	0
PV0 … PV9	Process value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#FF
SP0_Op *	Operator setpoint input	REAL	0.0
SP0_OpScale * … SP9_OpScale *	Setpoint limits	STRUCT • High: REAL • Low: REAL	- • High: 100.0 • Low: 0.0
SP0_Op * … SP9_Op *	Operator setpoint input	REAL	0.0
SP0_Unit * … SP9_Unit *	Engineering units of input	INT	1001

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters ParaMem

Parameter	Description	Data type	Default
NxtBlk	Connection to next block (ParaMem or ParaCtrl)	STRUCT	
NxtBlk.ActSet	Parameter set output values	INT	0
NxtBlk.DB_Error	DB error at any ParaMem block	BOOL	0
NxtBlk.EnChgParVal	Enable changing of parameter values	BOOL	0
NxtBlk.LoopAlive	Test signal ParaMem loop closed	BOOL	0
NxtBlk.LoopClosed	ParaMem loop closed	BOOL	0
NxtBlk.LoopOk	Loop alive signal from ParaMem blocks	BOOL	0

Parameter	Description	Data type	Default
NxtBlk.NumParaDB	Number of parameter memory blocks at current data block	INT	0
NxtBlk.NumParaMem	Number of parameter memory blocks	INT	0
NxtBlk.NumParaSet	Number of parameter sets	INT	0
NxtBlk.ParaCtrlDB_dbno	Parameter control block data block number	WORD	16#0000
NxtBlk.PrvDB_adr	Previous connected data block address in DB	INT	0
NxtBlk.PrvDB_dbno	Previous connected data block DB number	WORD	16#0000
NxtBlk.PrvDB_length	Previous connected data block DB length	INT	0
NxtBlk.ST_Worst	Worst signal status	BYTE	16#80
NxtBlk.ViewSet	Parameter set at input values	INT	0
SP0_OpScaleOut … SP9_OpScaleOut	Setpoint limits	STRUCT • High: REAL • Low: REAL	- • High: 100.0 • Low: 0.0
SP0_Out … SP9_Out	Active setpoint	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
SP0_UnitOut … SP9_UnitOut	Engineering units	INT	1001

6.4.7 ParaCtrl / ParaMem block diagram

A block diagram is not provided for this block.

6.4.8 Operator control and monitoring

6.4.8.1 ParaCtrl / ParaMem views

ParaMem block views

None (visualized within ParaCtrl)

ParaCtrl block views

The ParaCtrl block offers the following views:

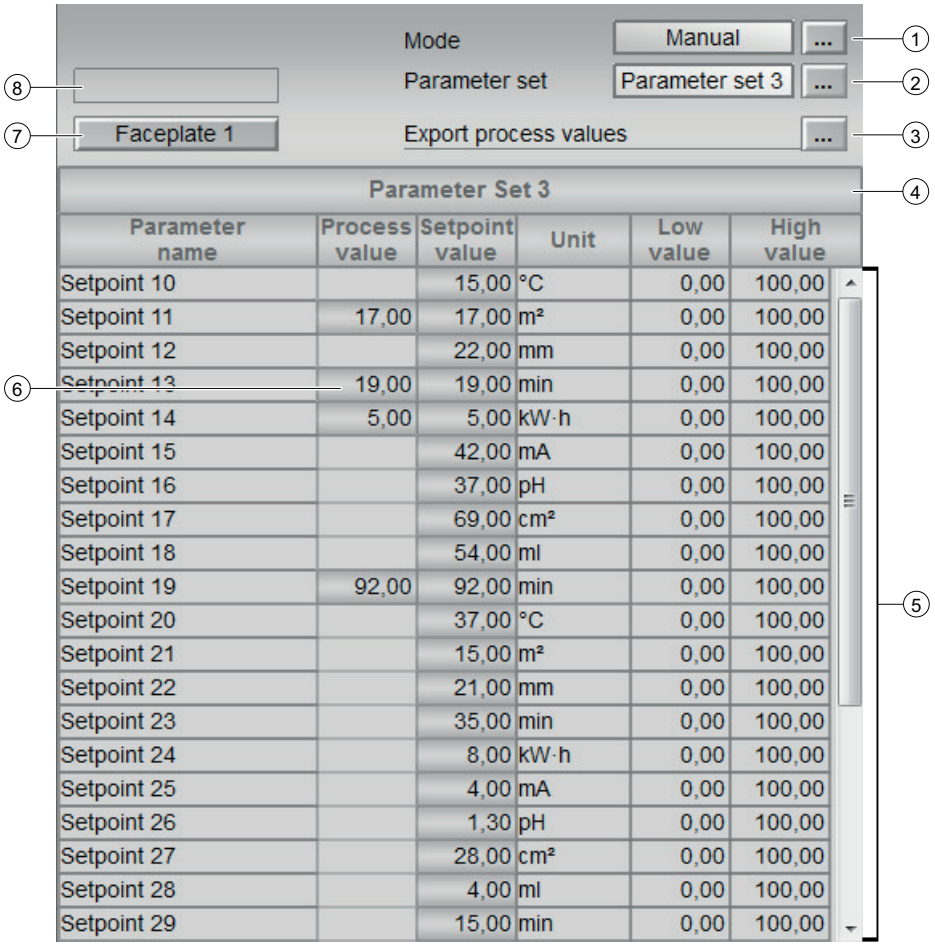
- ParaCtrl / ParaMem standard view (Page 235)
- ParaCtrl / ParaMem parameter view (Page 238)

- ParaCtrl / ParaMem preview (Page 240)
- Memo view
- Batch view
- ParaCtrl / ParaMem block icon (Page 241)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

6.4.8.2 ParaCtrl / ParaMem standard view

ParaCtrl standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- Manual mode
- Automatic mode
- Out of service

For more information on switching the operating mode, refer to the section "Switching operating states and operating modes" in the APL documentation.

(2) Current parameter set

This area shows the current parameter set if more than one parameter set is available. The area is not displayed if only one parameter set is available.

You can select and activate any configured parameter set with the input dialog.

(3) Export process values

Use the "Export process values" function to export the current process values. Process values are only exported if they are connected to the corresponding ParaMem block. This can be seen by the fact that the value in the "Process value" column is visible.

The export is performed using the "IL File Dialog". You can find more detailed information on this in the documentation of the File dialog (Page 56).

(4) Name of the current parameter set

This area shows the name of the current parameter set if more than one parameter set is available. The area is not displayed if only one parameter set is available.

(5) Display of the current parameter set

This area shows the current parameter set in form of a table. The following information is displayed:

- Parameter name (as configured at the ParaMem block)
- Process value (if interconnected)
- Current setpoint (parameter value)
- Unit of the parameter value
- Limits of the parameter value
  (The limits of the parameter value are not exceeded or undershot during output even if the setpoint should be outside this range.)

(6) Display of the process values in the parameter table

Process values are displayed if they are interconnected at the corresponding ParaMem block. This function is helpful, for example, if the parameter displays a setpoint for the controller. You can then connect the process value of the controller at the corresponding location at the ParaMem block. You can use it, for example, to create an overview of the controllers at a unit in form of a table.

(7) Navigation button for switching to the standard view of any faceplate

This display is only visible when the corresponding block input is interconnected.

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

For more information, refer to the "Opening additional faceplates" chapter in the APL documentation.

(8) Display area for block states

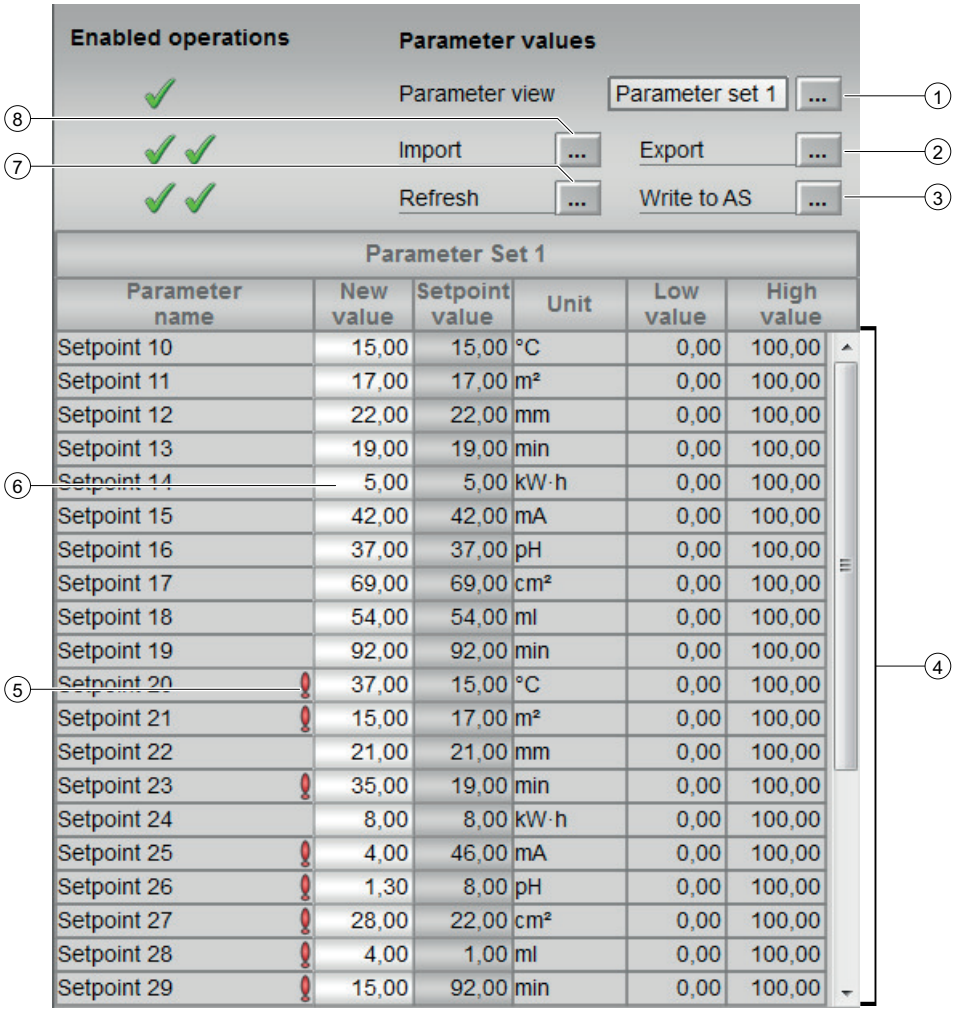
This area provides additional information on the operating state of the block:

- "Data error"
- "Invalid signal"

You can find more information about this in the section ParaCtrl / ParaMem error handling (Page 230).

6.4.8.3 ParaCtrl / ParaMem parameter view

ParaCtrl parameter view



(1) Current parameter set for processing

This area shows the currently processed parameter set if more than one parameter set is available. The area is not displayed if only one parameter set is available.

You can select any configured parameter set with the input dialog and activate it for processing.

(2) Export parameter set

Use the "Export parameter set" function to export the current parameter set. The values from the "New value" column are exported with this. You can find more detailed information on this in the section Basics of APL (Page 33).

(3) Write parameter set to AS

Use the "Write AS" function to write the parameter from the "New value" column to the AS. Only those values in which the new value deviates from the setpoint are written. This can be identified by the red exclamation mark in the corresponding line.

(4) Tabular representation of the execution of the parameter set

This area shows the current parameter set in form of a table. The following information is displayed:

- Parameter name (as configured at the ParaMem block)
- New value
  The "setpoints" are copied to the "new values" when the parameter view opens.
  Changes to the "new values" only take effect when they are written to the AS using the "Write AS" function.
- Current setpoint (parameter value)
- Unit of the parameter value
- Limits of the parameter value

(5) Display of changed parameter values

A red exclamation mark indicates that the current and the new parameter value are different.

(6) Display new parameter values

The "setpoints" are copied to the "new values" when the parameter view opens. By clicking on "New value", you can change the "new value" via the corresponding input dialog.

A red exclamation mark next to the new value indicates differences between the new and current value. The new values are written to the AS by the "Write AS" function.

(7) Update new parameter values

The "Update" function copies the current parameter values from the "Setpoint" column into the "New value" column

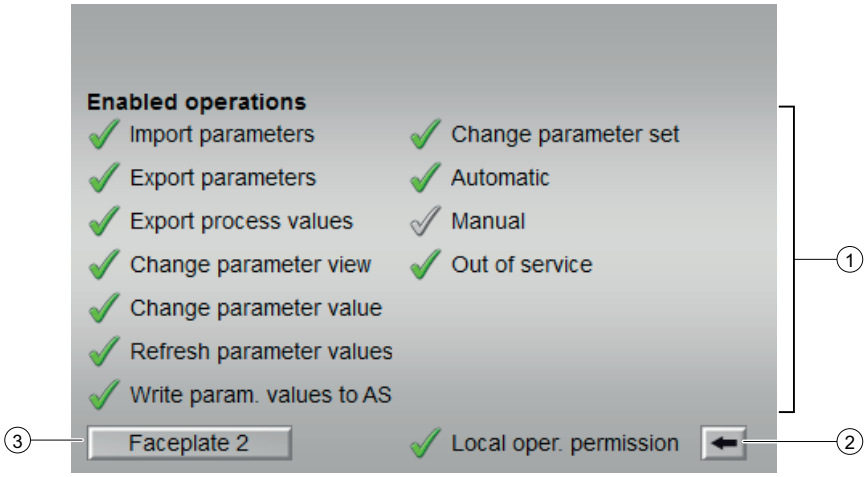
(8) Import parameter set

New parameter values can be imported using the "Import parameter set" function. These are displayed in the "New value" column. They only take effect once they are written to the AS using the "Write AS" function.

The import is performed using the "IL File Dialog". You can find more detailed information on this in the documentation of the "IL File Dialog (Page 56)".

6.4.8.4 ParaCtrl / ParaMem preview

ParaCtrl preview



(1) and (2) Operator control enable

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: the OS operator cannot control this parameter due to the configured AS operator permissions (OS_Perm or OS1Perm).

(2) The following operator control enables are displayed here:

- "Local operating permission": Use the <-- button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(3) Button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

6.4.8.5 ParaCtrl / ParaMem block icon

Properties of the block icon for ParaCtrl

A block icon with the following functions is available:

- Process tag type
- Operating modes
- Signal status, release for maintenance
- Memo display
- Status display

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Block icon for
	1	ParaCtrl

You can find additional information on the block icon and the control options in the block icon in the following sections of the APL manual:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

6.5 SelFp - Jump distributor

6.5.1 SelFp description

Object name (type + number) and family

Type + number: FB 1154

Family: Operate

Number of process objects: 0 PO

Area of application of SelFp

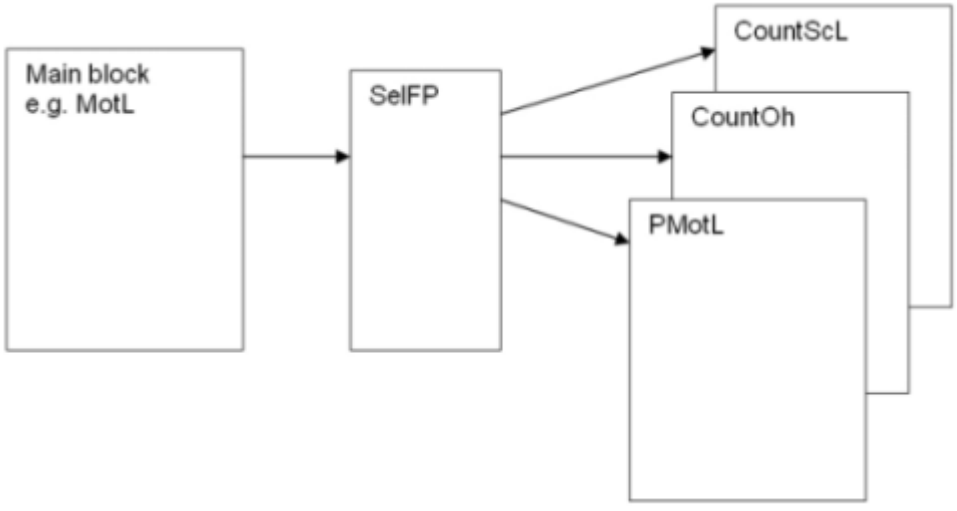
The SelFp block provides the menu navigation for the various function blocks within a typical. Up to 5 jump destinations can be configured on the block.

Configuration

The SelFp block uses the "Calling additional faceplates" function of the APL. To define the jump destinations, the `SelFpX` inputs must be interconnected with any output parameter of the destination block.

The buttons in the faceplate are labeled with the "OS Text" attribute of the `SelFpX` block inputs. The button is labeled with this text. If no text is entered, the button is labeled with the CFC/block name of the connected block.

The input for the `SwitchPerm` operating level is passed through to the `SwitchPerm_Out` output.



Startup characteristics

The block has no startup characteristics.

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

6.5.2 SelFp operating modes

The block does not have any operating modes.

6.5.3 SelFp functions

The SelFp block uses the "Calling additional faceplates" function of the APL. You will find more detailed information in the APL documentation.

See also

SelFp description (Page 242)

6.5.4 SelFp error handling

The block does not have any error handling.

6.5.5 SelFp messaging

The block has no message behavior.

6.5.6 SelFp I/Os

Input parameters

Parameter	Description	Data type	Default
`SelFp1 ... SelFp5`	Select faceplate 1 ... 5	ANY	
`SwitchPerm`	Switch permission	DWORD	1

Output parameters

Parameter	Description	Data type	Default
`SwitchPerm_Out`	Switch permission	DWORD	1

6.5.7 SelFp block diagram

The block has no block diagram.

6.5.8 Operator control and monitoring

6.5.8.1 SelFp views

SelFp block views

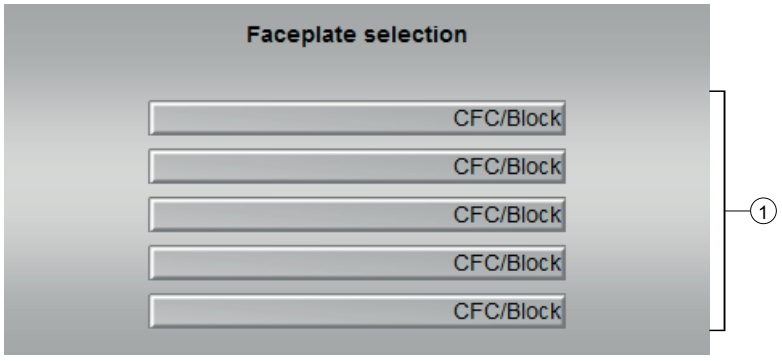
The block provides the following views:

- SelFp standard view (Page 245)
- SelFp block icon (Page 245)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

6.5.8.2 SelFp standard view

SelFp standard view



(1) Navigation button for standard view of any block

This display is only visible when the corresponding block input is interconnected. Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

6.5.8.3 SelFp block icon

SelFp block icons

There is no block icon for this block.

6.6 UsrM - User management

6.6.1 UsrM description

Object name (type + number) and family

Type + number: FB 1112

Family: Operate

Number of process objects: 1 PO

Area of application

Then UsrM block manages the operator control of up to 8 operator stations.

Startup characteristics

The block has no startup characteristics.

Called blocks

SFB35	ALARM_8P
SFC6	RD_SINFO

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

6.6.2 UsrM operating modes

The block does not have any operating modes.

6.6.3 UsrM functions

The operator stations can be enabled via the `DevxxAct` inputs or marked as available.
The names of the operating levels are created as enumeration and assigned at the `Out` output parameter.

The output parameters `Out` and `DevxxActOut` display the active operating level.
The output at the `Out` output is bit-coded. This means the output value is compatible with the `OpSt_In` input of a technological block for further interconnection.

You select the active operating level in the faceplate (KeySwitch.Value = FALSE) or in the program by means of interconnection (KeySwitch.Value = TRUE).
If a selected operating level fails, a switch is made to the next available backup level BackUp1 / BackUp2. If both backup levels are unavailable, the first available operating level is selected.

If no operating level is available, the NoLevel output is set and the Out output is set to 16#00000000.

The value 3 at the input, for example, will produce the following output values:

- Dev03ActOut = 1
- Out = 16#00000004 (bit 2 is set)
  = 2#00000000 00000000 00000000 00000100
- Operation (KeySwitch.Value = FALSE)
  The operating level is defined by the operable input SwitchPerm.
- Interconnection (KeySwitch.Value = TRUE)
  The operating level is defined by the interconnectable input KeySwLvl.

6.6.4 UsrM error handling

UsrM error handling

For information about error handling for all blocks, see also section "Error handling" in the Basics of APL manual.

The following error messages can be generated at this block:

- Error numbers

Overview of error numbers

The ErrCode I/O can be used to output various error numbers:

Error number	Meaning of the error number
0	There is no error.
2	Parameter assignment error – Backup1 > MaxLevel or Backup1 < 1
3	Parameter assignment error – Backup2 > MaxLevel or Backup2 < 1
4	Parameter assignment error – MaxLevel < 1 or MaxLevel > 8
5	Parameter assignment error - KeySwLvl > MaxLevel or KeySwLvl < 1

6.6.5 UsrM messaging

Message behavior

The UsrM block uses the ALARM_8P block for generating messages.

Message triggers are:

- Failure/change in operating levels
- Parameter assignment error
- Quality code of input `KeySwitch` (KeySwitch.ST).

The messages can be suppressed with the `MsgLock` input.

Process messages

The following control system fault messages can be output:

Message instance	Message ID	Message class	Event
MsgEvId	SIG 1	Process message – with acknowledgment	$$BlockComment$$ Key switch activated
	SIG 2	AS process control message - fault	$$BlockComment$$ Parameter assignment error
	SIG 3	Process message – with acknowledgment	$$BlockComment$$ Operating level changed
	SIG 4	Process message – with acknowledgment	$$BlockComment$$ Backup level 1 selected
	SIG 5	Process message – with acknowledgment	$$BlockComment$$ Backup level 2 selected
	SIG 6	Process message – with acknowledgment	$$BlockComment$$ Next available level selected
	SIG 7	Process message – with acknowledgment	$$BlockComment$$ No operating level available
	SIG 8	AS process control message - fault	$$BlockComment$$ External error

Explanation:

$$BlockComment$$: content of the instance-specific comment

You can interconnect an external fault (signal) to the input parameter `CSF`. If this signal changes to `CSF` = 1, a control system fault is triggered ((MsgEvId1; SIG 3).).

Associated values for the message instance **`MsgEvId1`**

Associated value	Block parameters
1	`ErrCode`
2	`SwitchPerm_Out` Old value
3	`SwitchPerm_Out`
4	`KeySwLvl`
5	`BackUp1`
6	`BackUp2`
7	`ExtVa107`

Associated value	Block parameters
8	ExtVa108
9	ExtVa109
10	ExtVa110

The associated values 7 ... 10 are allocated to the parameters ExtVa107 ... ExtVa110 , which you are free to use. You can find additional information on this in the "Process Control System PCS 7 - Engineering System" manual.

6.6.6 UsrM I/Os

Input parameters

Parameter	Description	Data type	Default
BackUp1	Backup permission level #1	INT	1
BackUp2	Backup permission level #2	INT	1
Dev01Act ... Dev08Act *	Device level 01 ... 08 available	BOOL	0
KeySwitch	Enable 1=Key switch	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
KeySwLvl	Device level for key switch	INT	1
MaxLevel	Maximum number of input devices	INT	8
MsgEvId	Message ID	DWORD	16#000000CB
SwitchPerm	Switch permission	INT	1

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
Dev01ActOut ... Dev08ActOut	1=Device 01 ... 08 is activated for operating	BOOL	0
ErrCode	Error code	INT	0
GrErr	1=Error	BOOL	0
MsgAckn	Message: ACK_STATE output	WORD	16#0000
MsgErr	1=Message error	BOOL	0
MsgStat	Message: STATUS output	WORD	16#0000
MsgSup	Message suppressed	BOOL	0
NoLevel	No operation level available	BOOL	0
Out	Output enabled operator stations	DWORD	0

6.6.7 UsrM block diagram

The block has no block diagram.

6.6.8 Operator control and monitoring

6.6.8.1 UsrM views

UsrM block views

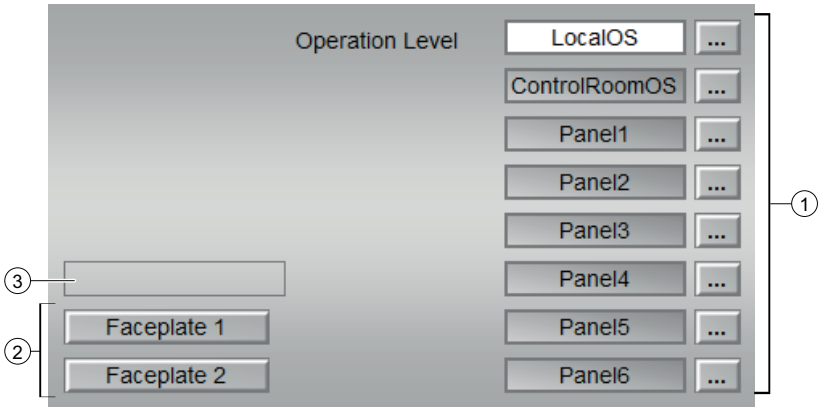
The block provides the following views:

- UsrM standard view (Page 250)
- Message view
- Memo view
- Batch view
- UsrM block icon (Page 251)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

6.6.8.2 UsrM standard view

UsrM standard view



(1) Operating level

Selection of the active operating level. In this area you select the active operation level. You can use customized texts instead of the default texts.
Proceed as follows:

- Create an enumeration called "ILOpLong" in the global declarations of SIMATIC Manager or copy the included template from the library. You can find additional information on this in the "Process Control System PCS 7 - Engineering System" configuration manual "How to save global declarations".
  The values described in the section Table with possible values of the operating levels (Page 43) are permitted for the enumeration. Other values are not recorded.

(2) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

(3) Display area for block states

This area provides additional information on the operating state of the block:

- "Key switch On"

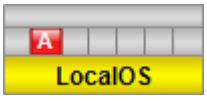
6.6.8.3 UsrM block icon

UsrM block icon

A variety of block icons are available with the following functions:

- Process tag type
- Violation of the alarm and warning limits as well as process control errors
- Signal status
- Memo display
- Status display

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	

You can find additional information on the block icon and the control options in the block icon in the following sections of the APL manual:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

Count 7

7.1 PuPa – Pulse sequence generator

7.1.1 Description of 'PuPa'

Area of application and operating principle

The function block is used mainly to sequentially pulse valve groups (double-seat valves), meaning the cyclic-continuous controlling of the seat lifting functions for cleaning purposes. The number as well as the duration of the pulses as well as the number of cycles can be specified using interconnectable inputs or also by operation.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).

Startup characteristics

Use the Feature Bit 'Set startup characteristics' to define the startup characteristics of this block.

After startup the output `CurLpDone` is switched inactive for the number of cycles set in the value `RunUpCyc` .

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'PuPa' (Page 259)".

Status bit	Parameter
0	`BatchOcc`
1	`BatchEn`
2	`Reserve`
3	`Reserve`
4	`OosLi.Value`
5	`ManAct.Value`
6	`AutAct.Value`
7	`OosAct.Value`
8	`Permanent`
9	`xIntlock`
10	`Reserve`

Status bit	Parameter
11	ActiveOut
12	StartOut.Value
13	AllLpDone
...	...

Status word allocation ‚Status2'

For a description of the individual parameters, see the section "Connections of 'PuPa' (Page 259)".

Status bit	Parameter
0	Reserve
1	ActGrp01 (pulse signal output)
2	ActGrp02
3	ActGrp03
4	ActGrp04
5	ActGrp05
6	ActGrp06
7	ActGrp07
8	ActGrp08
9	ActGrp09
10	ActGrp10
11	Act01 (signal 'Group active')
12	Act02
13	Act03
14	Act04
15	Act05
16	Act06
17	Act07
18	Act08
19	Act09
20	Act10
...	...

7.1.2 Operating modes of 'PuPa'

The block can be operated using the following standard operating modes:

- Automatic mode
- Manual mode
- Out of service

The next section provides additional block-specific information relating to the general descriptions.

Automatic mode

You will find general information about the automatic mode, changing modes in the APL documentation in the section 'Manual and automatic mode for motors, valves and dosers'.

In automatic mode you can carry out the following actions on pulse generation:

- Switch on `(StartAut = 1)`
- Switch off `(StartAut = 0)`

Manual mode

You will find general information about the manual mode, changing modes in the APL documentation in the section 'Manual and automatic mode for motors, valves and dosers'.

Valve actions you can control in manual mode:

- Switch on `(StartMan = 1)`
- Switch off `(StartMan = 0)`

Out of service

You will find general information about the out of service mode in the APL documentation, section "Out of service".

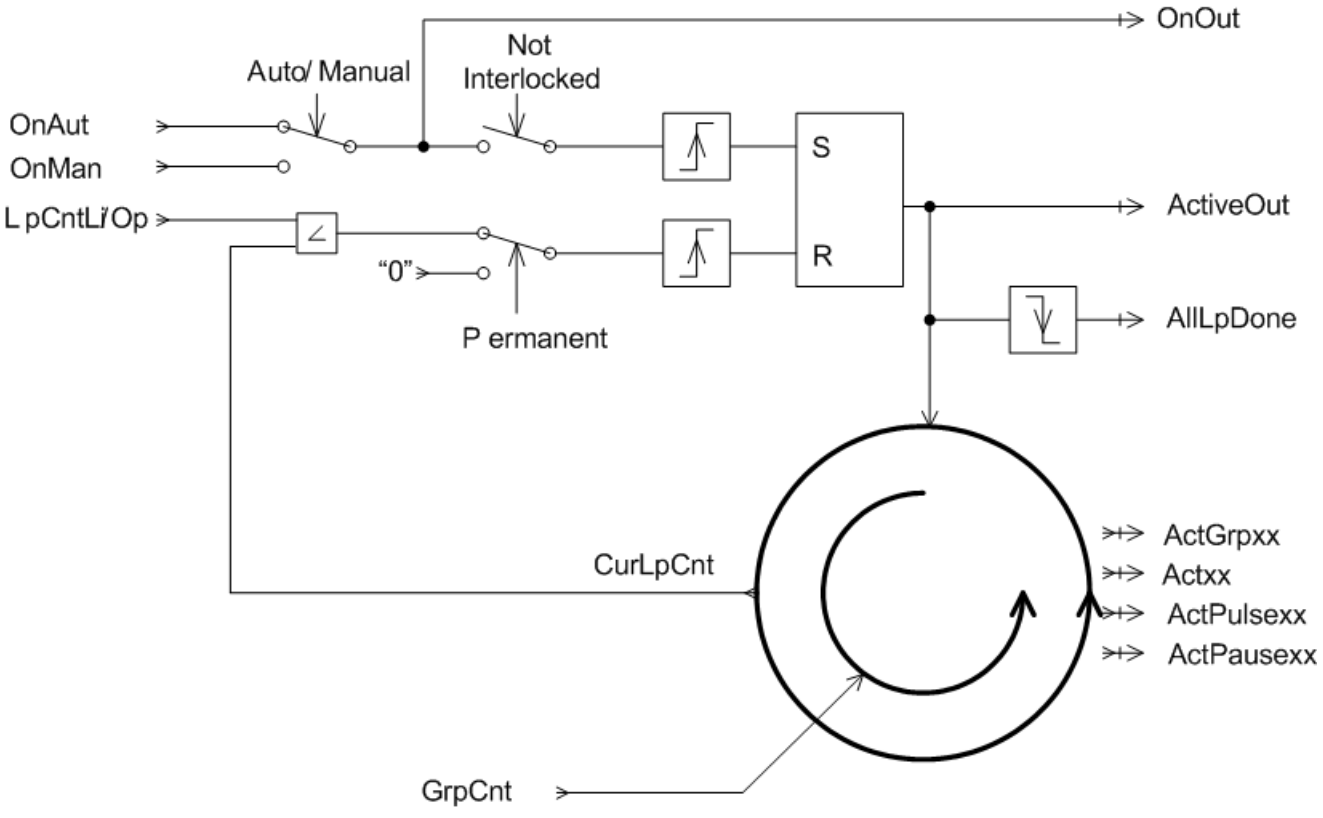
7.1.3 Functions of 'PuPa'

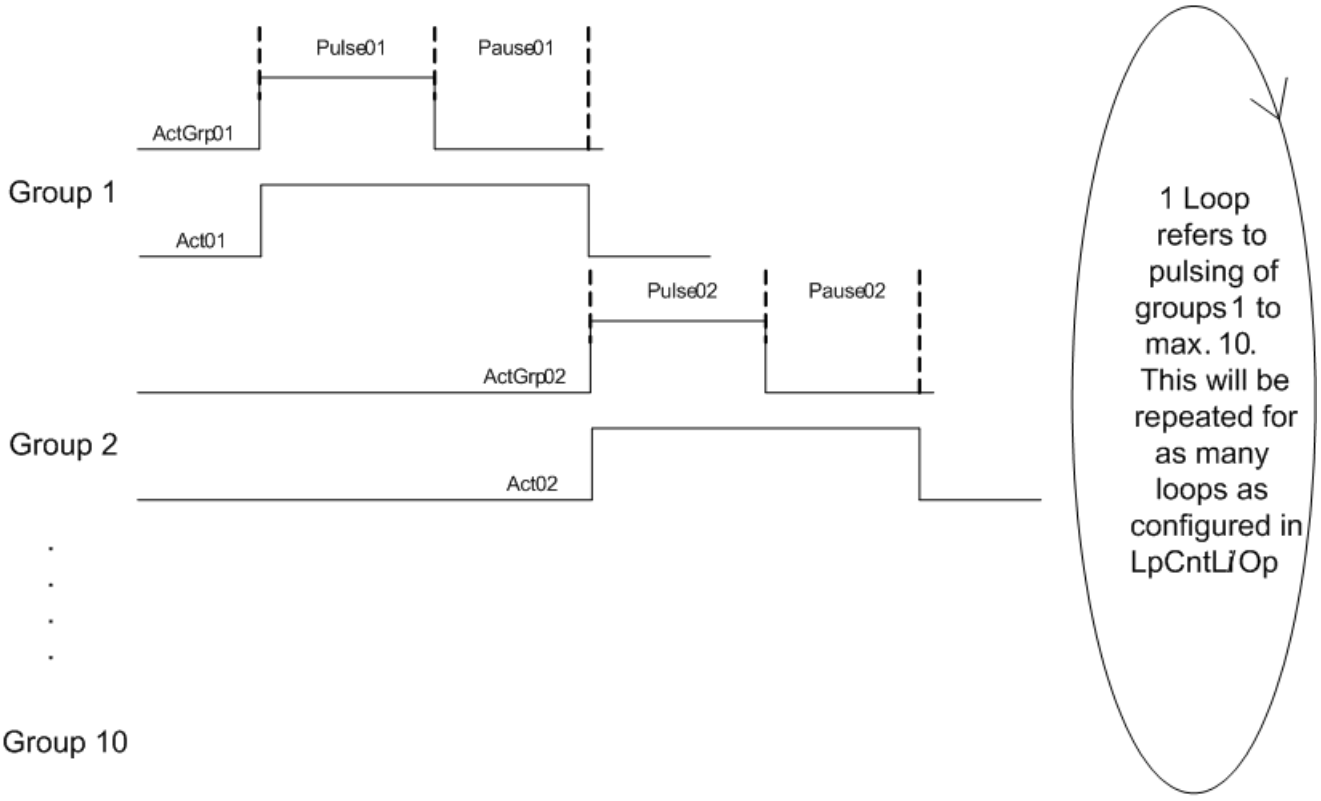
This function block implements sequential pulsing for a maximum of 10 valve groups. The number of valve groups (= active pulse outputs) is defined at the input `GroupCnt` . Generation of the continuous pulse runs in a loop, either in the mode LOOP (with configured number of loops) or PERM (continuous without limitation). All the important parameters, modes and states can be operated and controlled by the operator using a faceplate.

- The active pulses for interconnecting to the valves are made available at the outputs `ActGrpxx` .
- The operating mode is set in the faceplate (using the parameter `Permanent`):
  - `Permanent = 1`
    Endless loop
  - `Permanent = 0`
    The number of loops is defined using the parameter `LoopCnt` (interconnectable or operable using the faceplate). The current number of loops is signaled at the output CurLpCnt .
- The pulse length is defined using the parameter `Pulsexx`, the pause length using `Pausexx` (both interconnectable or operable using the faceplate)

- When the edge to the input `Leavexx` is positive, the corresponding pulse block is left for the current loop, `ActGrpxx` and `Actxx` are reset,
- If the function block is in manual mode and has not been started, the signal for the connected valve groups can be set using the faceplate by means of the parameters `ManActxx` (that are written to the outputs `ActGrpxx`).
- If the function block is running and the operating mode is set to automatic, the internal operator start command is forced to active - allowing the smooth switchover from automatic to manual operating mode.
- When the function block is started, the output `StartOut` is set and `ActiveOut` becomes `TRUE` by the end of the loop.
- After each loop the output `CurLpDone` is set for one cycle. When the loop has been terminated, `ActiveOut` is reset and `AllLpDone` is set until a new start command becomes active.

Block diagram





Further functions of BlMof

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwModSel`
- `OvwUnitID`
- `OvwBatchInfo`
- `OvwManOpEn`

You will find more information on this subject in the section AUTOHOTSPOT

Call additional faceplates

This block includes the APL standard function "Call additional faceplates".

Forming the signal status for blocks

This block includes the APL standard function "Form and output signal status for blocks".

The worst signal status for the block is formed from the following parameters:

- Pulse01
- ...
- Pulse10
- Pause1
- ...
- Pause10
- LoopCnt
- GroupCnt
- ModLiOp
- StartAut
- StartMan
- Intlock

Configurable reactions using the Feature parameter

An overview of all the reactions that are provided by the Feature parameter is available in the "APL documentation" in the section "Configurable functions with the Feature I/O". The following reactions are available for this block at the relevant bits.

Feature bits

Bit	Comment
0	Start-up without call in OB100 (default = 0)
1	Enable switchover to "Out of service" via OosLi (default = 0)
2	Reserve
...	...
24	1 = "Local" operator permission is active
...	...

Operator permissions

This block includes the APL standard function "Operator permissions". The block has the following permissions for the OS_Perm parameter:

Status bit	Parameter
0	1= Operator can switch to "Auto" mode
1	1=Operator can switch to "Manual" mode
2	1= Operator can switch to "Out of service" mode
3	1=Operator can switch pulse generation to "On"
4	1= Operator can switch pulse generation to "Off"
5	1 = Operator can change pulse / pause times
6	1 = Operator can switch group on / off manually

Status bit	Parameter
7	1 = Operator can change pulse mode LOOP/PERM
8	1 = Operator can change the loop counter
9	1 =Operator can change the number of groups
10	Reserve
...	...

Overview of error numbers

The `ErrorNum` connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.
55	If the input signals `AutAct` and `ManAct` are set simultaneously

7.1.4 Connections of 'PuPa'

Input parameters

Parameter	Description	Type	Default
`ModLiOp`	• 1=Link/Auto • 0=Manual: Input to auto/manual commands	`DigVal`	
`AutModOp`	1=Auto mode: auto mode by operator	`BOOL`	
`ManModOp`	1=Manual mode: manual mode by operator	`BOOL`	`True`
`AutModLi`	1=Auto mode: auto mode switch via link	`DigVal`	
`ManModLi`	1=Manual mode: manual mode switch via link	`DigVal`	
`StartAut`	1=Start: Start command in Auto Mode	`DigVal`	
`StartMan`	1=Start: Start command in Manual Mode	`DigVal`	
`OosOp`	1=Oos mode: Oos mode by operator	`BOOL`	
`OosLi`	1=Oos mode: Oos mode switch via link	`DigVal`	
`AutOpEn`	Operator may switch to AUTO mode	`BOOL`	`True`
`ManOpEn`	Operator may switch to MANUAL mode	`BOOL`	`True`
`IntLock`	1=Valve is not interlocked 0=Valve is interlocked (no reset is required)	`DigVal`	
`Intl_En`	1=Interlock enabled, 0=Interlock disabled	`BOOL`	`True`
`GroupCnt`	Group counter to be activated	`STRUCT` • `DINT` • `Byte`	`2` `16#FF`

Parameter	Description	Type	Default
LoopCnt	Loop number for groups to be activated	STRUCT • DINT • Byte	2 16#FF
Permanent	Permanent signal (0= counter active,1=always run)	BOOL	False
Pulse01	Pulse time for group 01 [s]	STRUCT • REAL • Byte	10.0 16#FF
Pause01	Pause time for group 01 [s]	STRUCT • REAL • Byte	10.0 16#FF
…	…		
Pulse10	Pulse time for group 10 [s]	STRUCT • REAL • Byte	10.0 16#FF
Pause10	Pause time for group 10 [s]	STRUCT • REAL • Byte	10.0 16#FF
CmData	Control and status information for this CM	BlUdtCmData	
OvwUnitID	1=Overwrite CmData.UnitID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
OvwManOpEn	1=Overwrite CmData.ManOpEn with input ManOpEn	BOOL	
UnitID	Unit ID (usually from BlUnitIf block)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIdLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
ManOpEn	Enable switchover to manual mode	BOOL	
ManAct01 … ManAct10	Manual activation ACT01 • 1: Cmd = On • 0: Cmd = Off	BOOL	False
Leave01 … Leave10	Leave group prematurely	BOOL	False
RunUpCyc	Number of cycles for which all messages are suppressed	INT	3
SampleTime	Sample time for block	REAL	1.0
SelFp1	Select Faceplate 1	ANY	
SelFp2	Select Faceplate 2	ANY	
OS_Perm	Operator Permission	STRUCT	

Parameter	Description	Type	Default
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

Output parameters

Parameter	Description	Type	Default
AutAct	1=Automatic mode is active	DigVal	
ManAct	1= Manual mode is active	DigVal	
OosAct	Out of service is active	DigVAl	
BatchEn	Enable remote operation of controller by Batch recipe	BOOL	
BatchOcc	Occupied by Batch, enables second message	BOOL	
BatchID	Current Batch ID	DWORD	
BatchName	Current Batch name	STRING32	
StepNo	Batch step number	DWORD	
StartOut	Status of command 0 - OFF, 1 – ON	DigVal	
ActiveOut	Status of loop 0 - stop, 1 – start	BOOL	
ActGrp01	Activate devices group 01	BOOL	
Act01	Current active device group 01	BOOL	
ActPulse01	Active pulse TIME group 01	REAL	
ActPause01	Active pause TIME group 01	REAL	
...	...		
ActGrp10	Activate devices group 10	BOOL	
Act10	Current active device group 10	BOOL	
ActPulse10	Active pulse TIME group 10	REAL	
ActPause10	Active pause TIME group 10	REAL	
ActPuTime	Active pulse TIME	AnaVal	
ActPaTime	Active pause TIME	AnaVal	
CurLpCnt	Actual loop number	DINT	0
CurLpDone	One loop done, signal for cascading another block (1-cycle signal)	BOOL	
AllLpDone	All loops done	BOOL	
OS_PermOut	Operator permission: output 1 for OS	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
OS_PermLnk	Operator permission (OS info)	DWORD	16#0
OpSt_Out	Enabled operator stations	DWORD	
ST_Worst	Enabled operator stations	DWORD	
Status1	Status1 data	DWORD	
Status2	Status2 data	DWORD	
ErrorNum	Error Number	INT	-1

7.1.5 Operator control and monitoring

7.1.5.1 Symbol



The symbol contains the following information:

Position	Content	
1	Tag name	
2	Operating mode	
		Automatic
		Manual
		Out of service
3	Empty	
4	Empty	
5		Enable maintenance operation
6	Interlock status	
		Interlock deactivated
		Not interlocked
		Interlocked
7		Memo is available

7.1.5.2 Faceplate

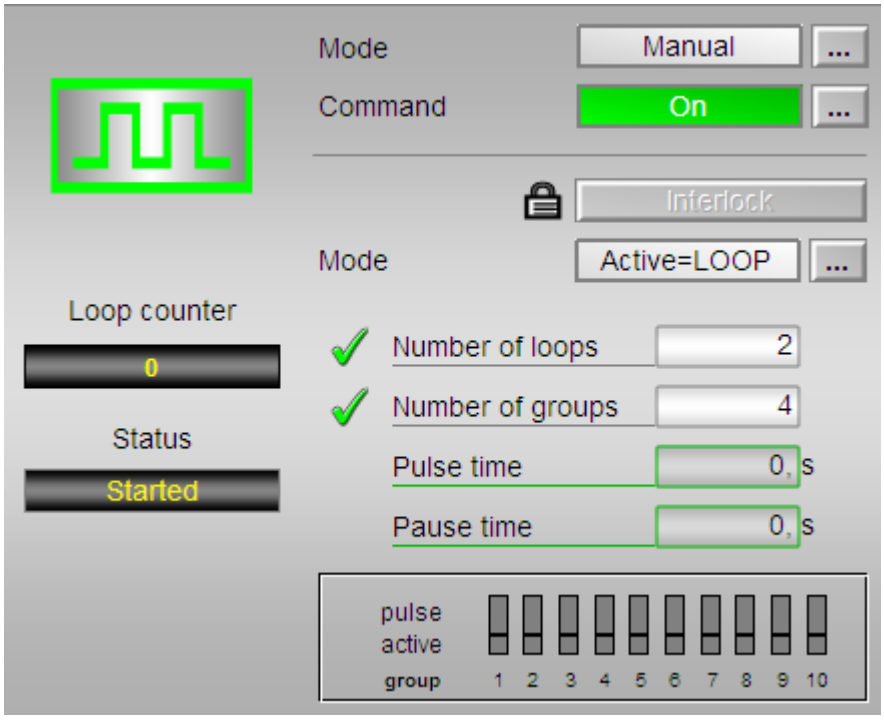
Views of the PuPa block

The block PuPa provides the following views:

Symbol	View	Information
	Standard view	See below.
	Message view	
	Trend view	Standard trend view of all the blocks of the Advanced Process Library
	Parameter view	See below.

Symbol	View	Information
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

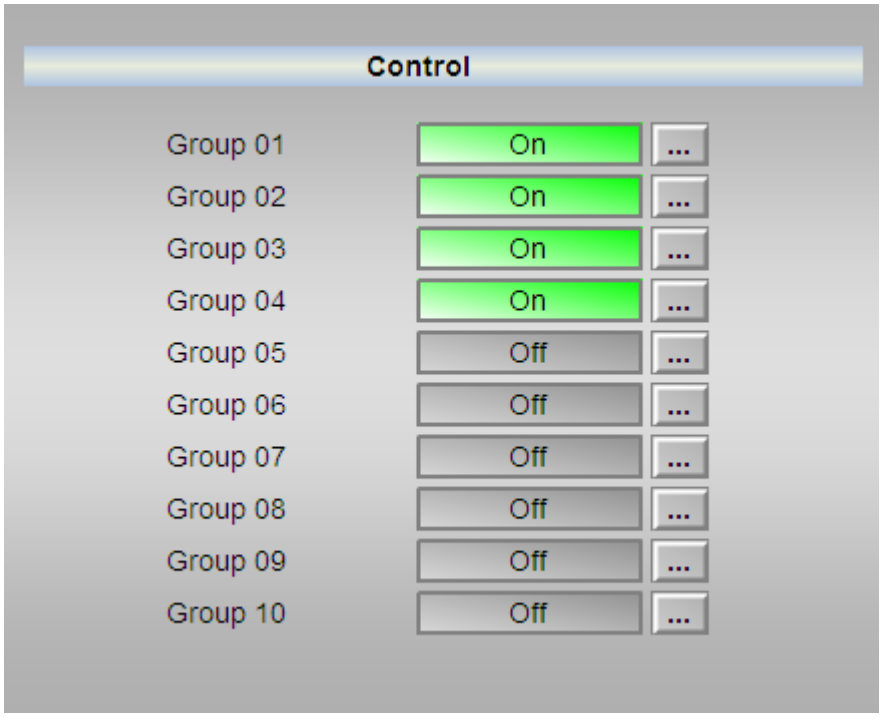
Standard view of the PuPa



Explanation of the screen shot	
(1)	**Displaying and switching the operating mode**
	This area provides information on the currently valid operating mode. The following operating modes can be shown and executed here: • Automatic mode • Manual mode • Out of service
(2)	**Switching the pulse generation on / off**
	This section shows you the operating state of the pulse generator. The following states can be shown and executed here: • On • Off

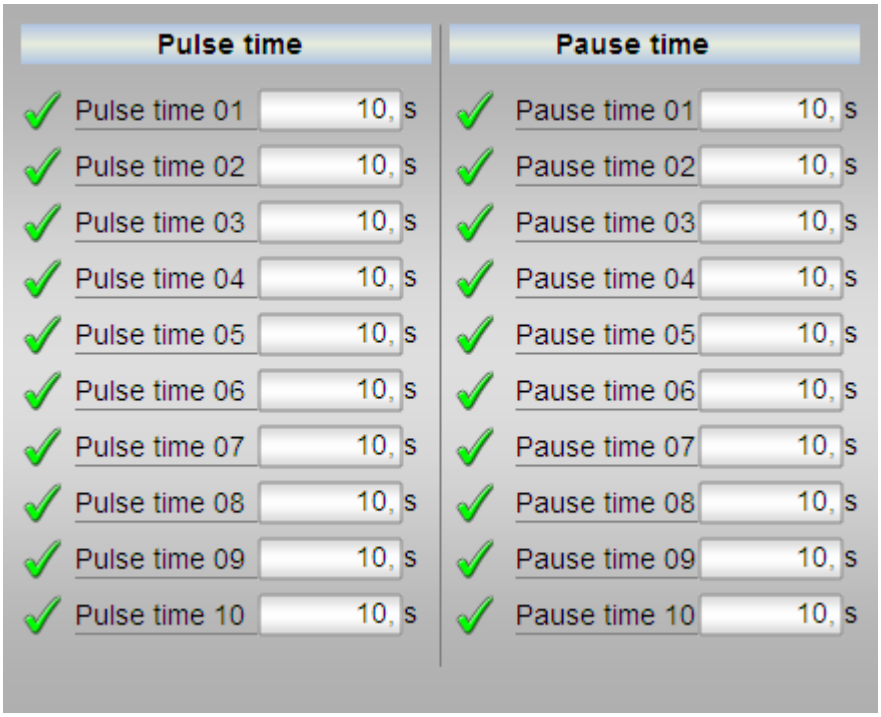
Explanation of the screen shot	
(3)	**Mode of the block**
	This section shows you the current execution mode for the pulse generator. The following modes can be shown and executed here: • `Active=LOOP` • `Active=PERM`
(4)	**Parameters of the pulse generation**
	• Number of loop executions for `Active=LOOP` • Number of groups (= connectable valve groups) • Pulse time [s] Current output signal = `True` • Pause time [s] Current output signal = `False`
(5)	**Online status of the pulse generation**
	The output status is displayed in a vertical bar symbol per group. • Both partial bars green Output signal = `True` • Lower partial bar green Output signal = `False` • Both partial bars gray Output is inactive
(6)	**Block status**
	Shows the current state of the loop counter • Off Pulse generation is not started • Active At mode ,`Active=LOOP'` and pulse generation has been held • Started Pulse generation is started • Ended Loop counter has expired
(7)	**Loop counter**
	Displays the current value of the loop counter for the mode ,`Active=LOOP'`
(8)	**Navigation button to the standard view of any faceplate**
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp1` in the CFC.

"Manual operation" view



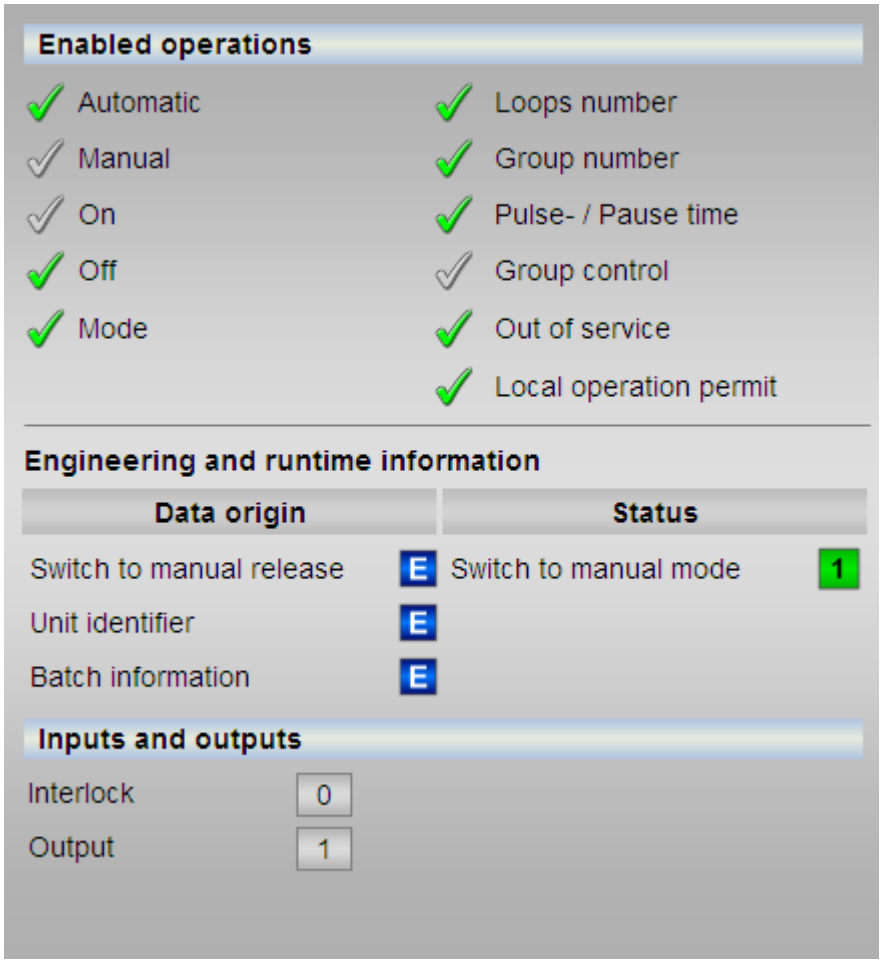
In the group control section the Outputs 1 – 10 can be switched on / off individually (parameter `ManActxx`) when pulse generation is deactivated. When the pulse generation is started, all the parameters are reset and the pulse outputs are generated by the loop execution in the block.

Parameter view of the PuPa



This section shows you the pulse and idle times for the maximum of 10 groups in [s]. The values can be changed if the required permission is available.

Preview of PuPa



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the `OS_PERM` parameter of this block. The operator control enables shown in the screenshot are possible here. • Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions

Explanation of the screen shot		
(2)	**Engineering and runtime information**	
	This section contains information from the status word `Status1`	
	• The "Data origin" column shows the setting of the overwrite signals. – Enable manual mode `OvwManOpEn` – Unit identifier `OvwUnitID` – Batch information `OvwBatchInfo` • The symbols have the following meaning	
		Signal origin = Unit / `CmData` structure
		Signal origin = Block input `Ovw<xxx>`
	• 'States' column displays the state of the status bit `Status1.x17 := xManOpEn` – `xManOpEn = 0` Enable manual mode deactivated – `xManOpEn = 1` Enable manual switching operation active	
(3)	**Inputs and Outputs**	
	The status of the interlock signal `Intlock` and of the output signal `StartOut` is displayed in this section.	
(4)	**Navigation button to the standard view of any faceplate**	
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp2` in the CFC.	

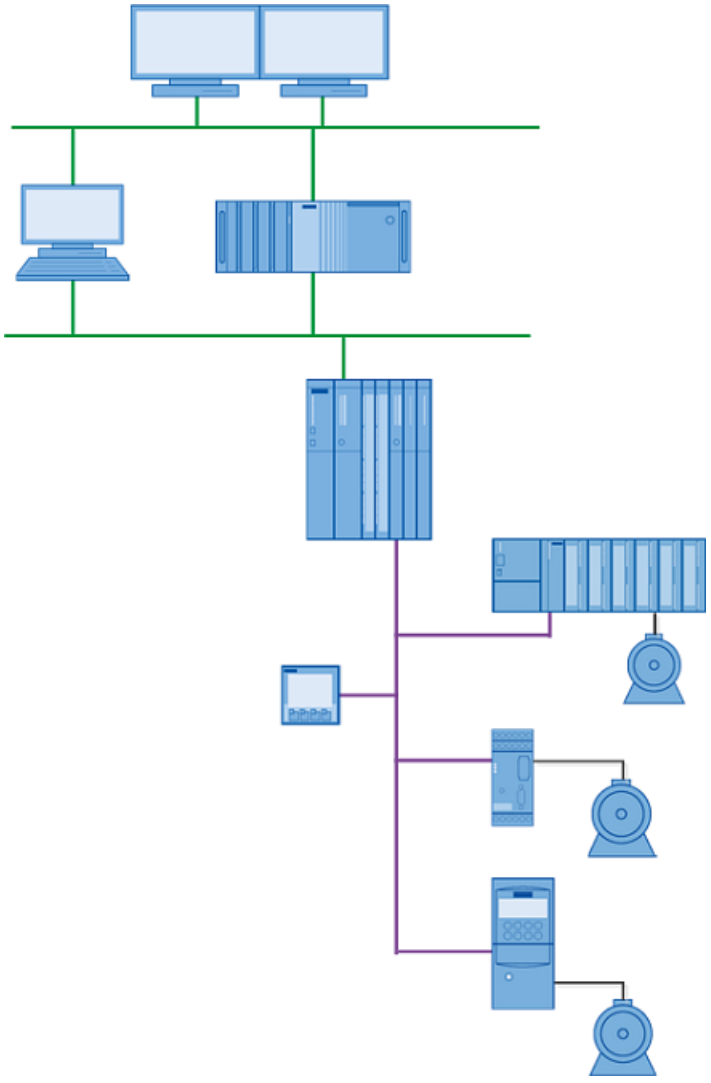
Energy management blocks

8

8.1 Description of the energy management blocks

Area of application

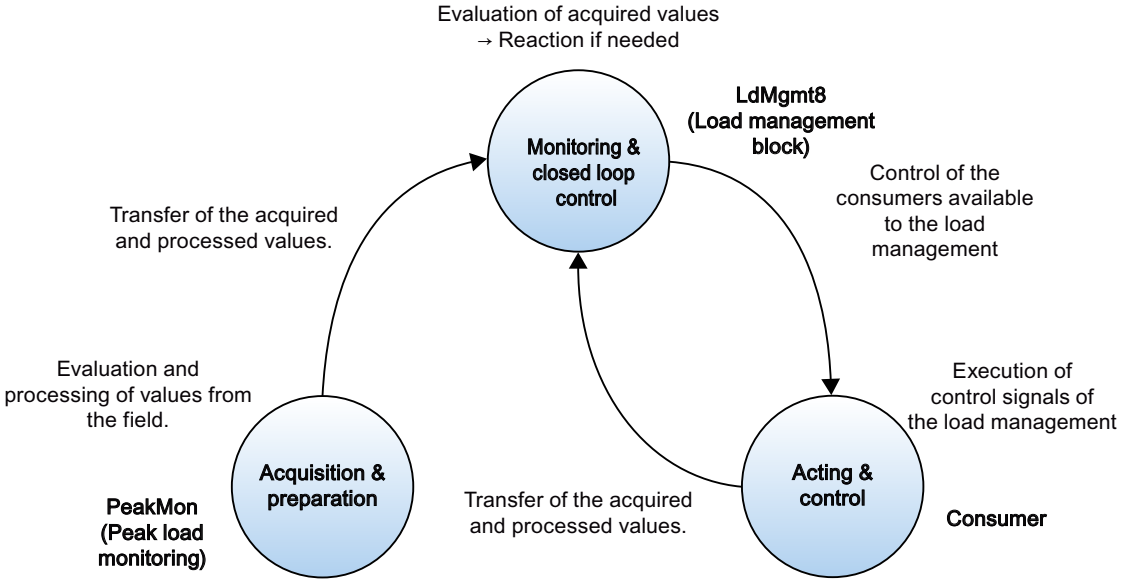
This production-related load management is used in all applications where consumers/ generators in the process industry are controlled via load management. It includes data acquisition, analysis and closed loop control of consumers/generators. The focus of this production-related load management is the adherence to the limits of power supply companies.



Function blocks are provided for acquiring and processing performance data, as well as for load shedding/connection of consumers/generators based on a configured load limit. (Power supply company limit)

The processed values are made available to the actual load management function. Based on this data and the characteristics of consumers/generators, load management calculates when a consumer/generator is connected/disconnected and when a consumer/generator is released for being switched on or shut down.

Consumers (e.g. motors) are configured with APL standard functions.



Configuration

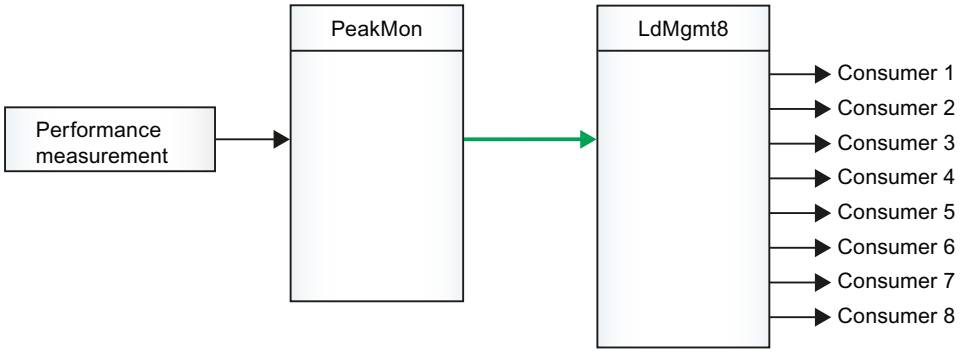
Load management consists of three functional blocks:

- PeakMon – Acquisition and processing of consumption values at the central infeed
- CalcPwrL – Acquisition and calculation of energy values for units and plant sections
- LdMgmt8 – Control of consumers/generators according to load criteria
- PulseCon – Recording of performance data via the pulse output of a volume meter.

PeakMon, LdMgmt8 and PulseCon blocks must be interconnected as shown in the figure below.

The following sections describe in detail the functions of each block and the required interconnections.

PulseCon is only required for the application scenario in which performance data is acquired via the pulse output of a volume meter.


PeakMon
LdMgmt8
Performance measurement
Consumer 1
Consumer 2
Consumer 3
Consumer 4
Consumer 5
Consumer 6
Consumer 7
Consumer 8

8.2 LdMgmt8 - Load management block for coordination of up to 8 consumers

8.2.1 LdMgmt8 description

Object name (type + number) and family

Type + number: FB 1191

Family: EngyMgmt

Number of process objects: 100 PO

Area of application for LdMgmt8

The block is used for coordination of up to 8 consumers in the load management.

Mode of operation

The block coordinates at least 1 and up to 8 consumers and/or generators in load management together with the PeakMon block.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB100).

Startup characteristics

Use the Feature Bit 'Setting the startup characteristics' to define the startup characteristics for this block.

The messages are suppressed after startup for the number of cycles set at `RunUpCyc`.

Status word allocation for `Status1` parameter

You can find a description for each parameter in the section LdMgmt8 I/Os (Page 287).

Status bit	Parameter
0	`Occupied`
1	`BatchEn`
2-4	Not used
5	`OosAct`.Value
6	`AutAct`.Value
7	`OnAct`
8	`OffAct`

Status bit	Parameter
9	Not used
10	`OosLi`.Value
11-13	Not used
14	Delay time start active
15	Delay time stop active
16	Primary PeakMon block is out of service
17-19	Not used
20	1= `PrefLoadMode`: true
21	1= `PrefLoadMode`: false
22	1= `TimeMode`: true
23	1= `TimeMode`: false
24	Not used
25	No aggregate available
26	Aggregates available for
27	Automatic preview for "active"
28	Automatic preview for "passive"
29	Not used
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word allocation for `Status2` parameter

You can find a description for each parameter in the section LdMgmt8 I/Os (Page 287).

Status bit	Parameter
0	`MsgLock`
1	Load 1 connected
2	Load 2 connected
3	Load 3 connected
4	Load 4 connected
5	Load 5 connected
6	Load 6 connected
7	Load 7 connected
8	Load 8 connected
9	Parameter change
10	Parameter change `TimeMode`
11	Parameter change `PrefLoadMode`
12	Parameter change `LoadMOde`
13	Parameter change `DelayOn`
14	Parameter change `DelayOff`
15	Parameter change number of aggregates
16	Parameter change `Load1Prio`
17	Parameter change `Load2Prio`

Status bit	Parameter
18	Parameter change `Load3Prio`
19	Parameter change `Load4Prio`
20	Parameter change `Load5Prio`
21	Parameter change `Load6Prio`
22	Parameter change `Load7Prio`
23	Parameter change `Load8Prio`
24	Parameter change `Load1Device` or connection parameter Load 1
25	Parameter change `Load2Device` or connection parameter Load 2
26	Parameter change `Load3Device` or connection parameter Load 3
27	Parameter change `Load4Device` or connection parameter Load 4
28	Parameter change `Load5Device` or connection parameter Load 5
29	Parameter change `Load6Device` or connection parameter Load 6
30	Parameter change `Load7Device` or connection parameter Load 7
31	Parameter change `Load8Device` or connection parameter Load 8

Status word allocation for parameter `Status3`

For a description of the individual parameters, refer to section LdMgmt8 I/Os (Page 287).

Status bit	Parameter
0-30	Not used
31	`MS_RelOp`

Status word allocation for parameter `Status4`

You can find a description for each parameter in the section LdMgmt8 I/Os (Page 287).

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via `EventTsIn`
1	Effective signal 2 of the message block interconnected via `EventTsIn`
2	Effective signal 3 of the message block interconnected via `EventTsIn`
3	Effective signal 4 of the message block interconnected via `EventTsIn`
4	Effective signal 5 of the message block interconnected via `EventTsIn`
5	Effective signal 6 of the message block interconnected via `EventTsIn`
6	Effective signal 7 of the message block interconnected via `EventTsIn`
7	Effective signal 8 of the message block interconnected via `EventTsIn`
8 - 23	Effective signal 8 to 16 of the message block connected via `Event16TsIn`
24-31	Not used

8.2.2 LdMgmt8 operating modes

The block can be operated using the following modes:

- Automatic mode
- Manual mode
- Out of service

The next section provides additional block-specific information relating to the general descriptions.

"Automatic mode"

You can find general information on automatic mode, changing modes and bumpless transfer in the section "Manual and automatic mode for motors, valves and dosers" of the APL documentation.

In automatic mode, the block evaluates the input parameters independently, displays them in the faceplate and switches the consumers according to the specified behavior.

"Manual mode"

You can find general information on manual mode, changing modes and bumpless transfer in the section "Manual and automatic mode for motors, valves and dosers" of the APL documentation.

In manual mode, the block evaluates the input parameters independently and displays them in the faceplate. However, there are no switching actions. All available values are displayed. Manual mode is basically a stop of the automatic mode.

"Out of service"

You can find general information about the "out of service" mode in the "Out of service" section of the APL manual.

8.2.3 LdMgmt8 functions

LdMgmt8 functions

The functions for this block are listed below.

Release for maintenance

This block has the standard APL function, "Release for maintenance".

Select unit of measure

This block includes the standard APL function, Select unit of measure.

Calling additional blocks

This block includes the standard function Call additional faceplates. Information is available in the section *Calling additional faceplates* in the *APL Function Manual*.

This block also has an additional button in the standard view which calls the faceplate of the PeakMon interconnected at the `BlkPeakMon` input.

Suppressing messages using the `MsgLock` parameter

This block includes the standard APL function, "Suppress messages using the MsgLock parameter".

Group display SumMsgAct for limit monitoring, CSF and ExtMsgx

The block has the standard APL function, Group display for limit monitoring, CSF and ExtMsgx.

Displaying auxiliary values

This block provides the standard APL function, "Display auxiliary values".

Maximum rated power

The rated power of the unit is required to calculate the enabling time.

The maximum rated power is the maximum power that the connected consumer can retrieve/ the maximum power that the connected generator can supply. It is crucial for this power to be as precise as possible for the enabling time so that load shedding does not occur immediately after enabling.

Current power consumption

The current power consumption is required to forecast the power. The result determines the time when the consumer is locked/ the generator is connected to observe the limit.

Power difference

The power difference is the difference between forecast and available power. The LdMgmt8 block receives this information from the assigned PeakMon block.

The parameter is calculated as follows:

- See PeakMon (Page 336)

Power reserve

The power reserve indicates how much higher the power still may be to stay within the available power limit. The LdMgmt8 block receives this information from the assigned PeakMon block.

The parameter is calculated as follows: (power maximum = power reserve)

- See PeakMon (Page 336)

Availability

Consumer

The availability of the consumer is usually not relevant for the LdMgmt8 block, but the information whether or not the consumer is operating so that it can be turned off, if necessary.

Generator

Availability indicates if the generator is accessible for the load management block and can be turned on. To do so, the `LoadXRdyToStart` input must be connected with an output that provides this information, for example, the `RdyToStart` output of a ApMotL.

Note

If the `LoadXRdyToStart` information is required for an unit can be defined at the `Feature`.Bit8-15 (1 = `LoadXRdyToStart` is required).

Locking consumers

The "LoadXLock" output must be interconnected with the respective consumer.

The strategy for locking consumers can be different in each case. This parameter can, for example, be interconnected to an input of the locking block, which means it has a different effect on the behavior of the consumer depending on the locking input used.

Note

Locking and enabling of consumers

The use of a locking input that requires acknowledgment often makes sense for locking of a consumer to reduce unnecessarily recurring switching operations (regular switch-on and switch-off every few cycles).

When the limit is exceeded, all non-active consumer are locked. A consumer is also locked if the power reserve were to be exceeded by connecting the consumer.

The consumer is not released until the limit is undershot again, and the available power is sufficient for the respective consumer.

Parameter application (`Feature`.Bit26)

Different strategies are possible at the block for applying changed parameters. The strategy is set at the parameter application `Feature`.Bit26 and affects the following parameters:

- `TimeMode`
- `PrefLoadMode`
- `LoadMode`
- Change of the device type or number of units
- `Prio`
- `DelayOn/DelayOff`

Automatic application of changed parameters (`Feature`.Bit26 = 1)

The block applies parameters automatically as soon as it is in a state in which these can be applied.

Parameter application by operator intervention (`Feature`.Bit26 = 0)

Operator intervention is necessary to apply changed parameters (passivation of the load management block)

Priority

The priority of the consumers determines, depending on the mode, the sequence in which the consumers are locked. Even when consumers are enabled, the sequence is determined based on the priority. The lowest priority can be a three-digit number.

Internal calculation

- Residual power per consumer:
  Calculation of the power that the consumer consumes in the time remaining of the interval (per consumer).
- Residual power of all active consumers:
  Calculation of the total power of all active consumers to determine exceeding or undershooting of the target point.
- Enable power per consumer:
  Calculation of the power that the consumer would consume with enable until the end of the interval.

Shutdown sequence (mode)

Based on priority (`PrefLoadMode` = 1)

The consumer with the lowest priority is shut down first; the one with the highest priority last.

Priority 1 has the highest priority. If the same priority is specified multiple times, the consumer with the highest number is shut down first.

This function mandates that a priority is assigned to each consumer. A corresponding error number is output at the ErrorNum parameter when a priority is missing.

Power-oriented (`PrefLoadMode` = 0)

The priority is secondary. The first consumer to be shut down, is the consumer whose current power is closest to the power to be shut down. This is usually the smallest consumer. If there are major jumps in the power consumption, however, larger consumers can also be shut down first.

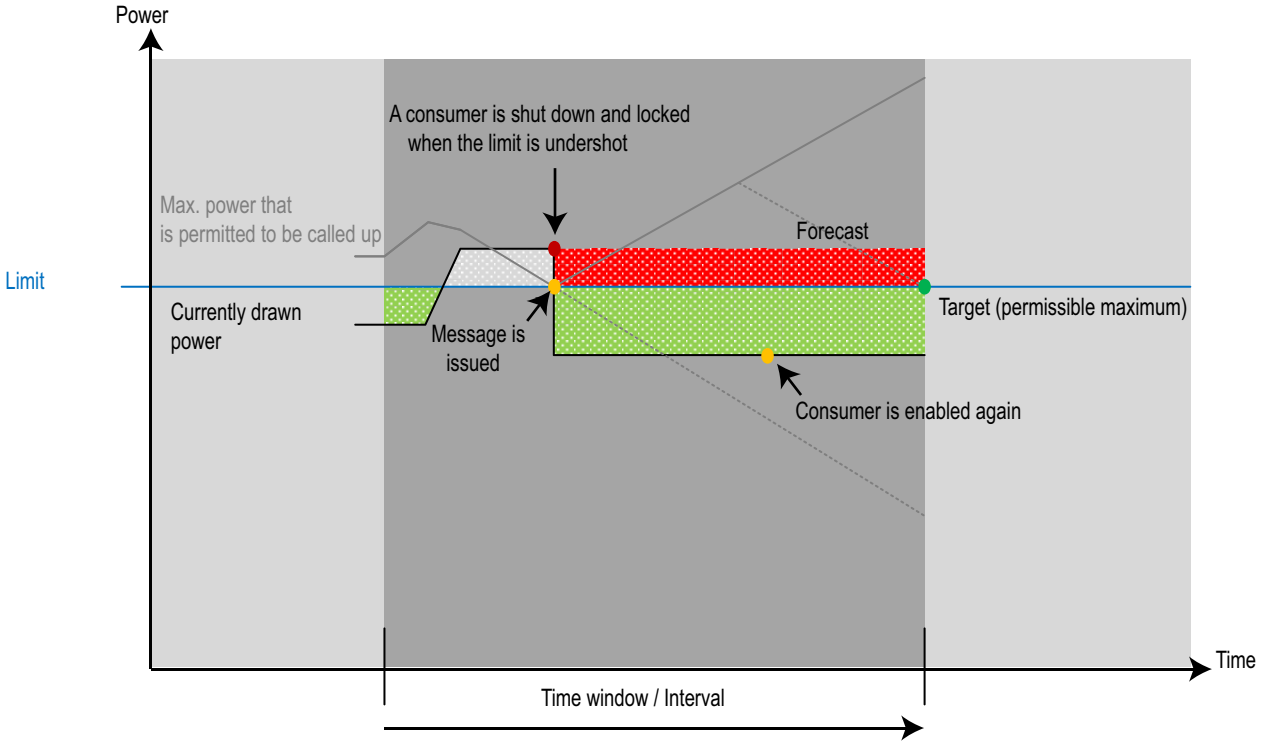
If multiple consumers have the same current power, the consumer with the lowest priority is shut down first. If no priority was specified, the consumer with the highest number is shut down first.

If multiple consumers have the same priority and when it is the turn for one of the consumers in the shutdown sequence, the consumers are considered as a group. This means the power is added and forecast. When it comes to shutdown, all consumers are shut down simultaneously.

Shutdown scenarios

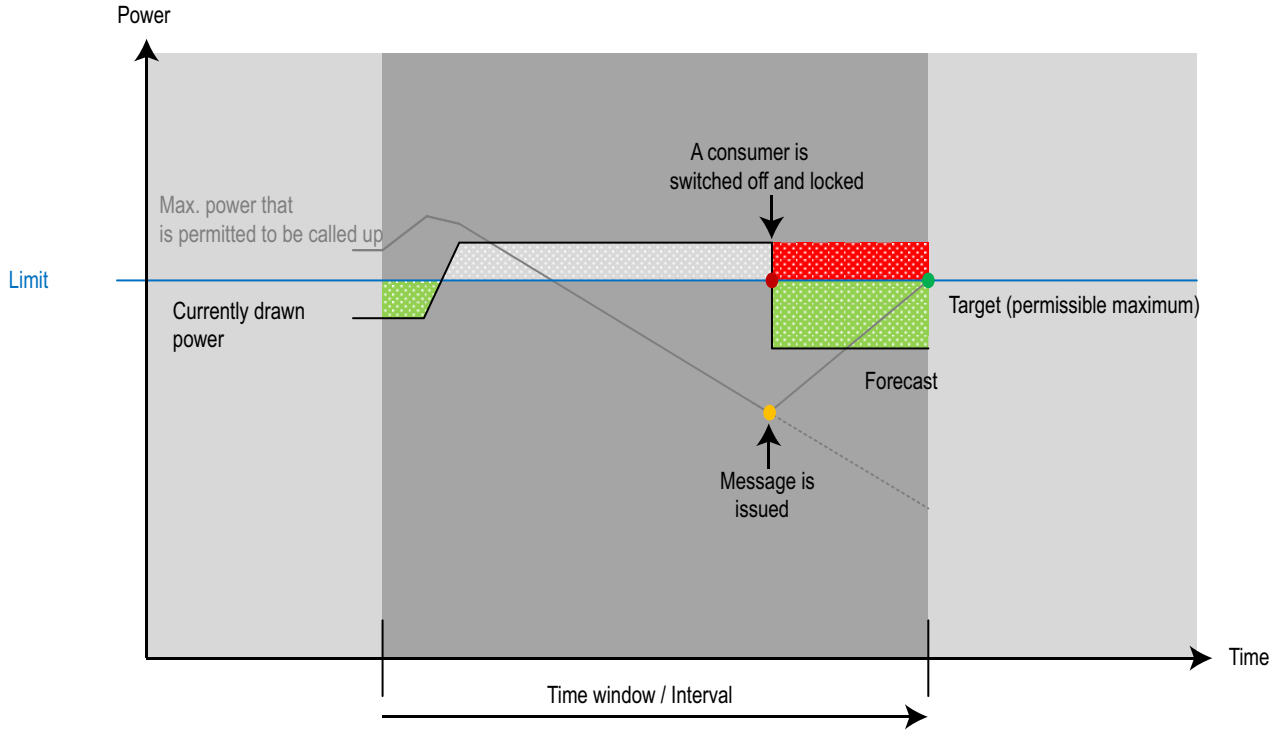
Early shutdown of loads (`TimeMode` = 0)

With early shutdown, the first consumer is locked as soon as the limit can no longer be maintained as determined by the forecast.



Late shutdown of loads (`TimeMode` = 1)

With late shutdown, the first consumer is shut down as late as possible. The goal is that the mean value is once again within the limit at the end of the time window and that the consumer can work as long as possible.



Wait time for restart (`LoadXIdleTime`)

The wait time after a shutdown until consumers and generators are enabled and switched on again can be configured with parameter „`LoadXIdleTime`". The consumer is not available for the load management during a wait time and is identified as "not available".

This may have an effect on the switch-on enable. If the wait time is still in progress and the next available consumers meets all requirements for an enable, it will be enabled. If the wait time expires and the enabled consumer is not switched on, the shut-down consumer is enabled as well.

If the consumer that was first enabled is switched on, however, the consumer ready to be switched on again must first fall below the forecast limit before it can be enabled again.

The operator can intervene here by bypassing the load management locking for this consumer and switching on the consumer manually. Load management is now forced to switch off a different consumer according to the set criteria.

With a generator, the operator can force the generator to switch off so that load management can connect a different generator.

Generator or consumer

Parameter `LoadXDevice` is used to set whether a generator (`LoadXDevice` =1) or consumer (`LoadXDevice` = 0) is connected to the output of the load management channel.

The block will behave exactly opposite, depending on the setting: consumers are shut down, generators are connected.

The settings for shutdown scenarios and shutdown sequence apply to both cases. With the help of the setting at input LoadMode, the user can define whether, in the case of a power-oriented shutdown for serious reasons, the first switching operation of the block:

- Shutdown of consumers (LoadMode = 1).
- A connection of generators (LoadMode = 2).
- Regardless of the type of the connected component, only according to power/priority (LoadMode = 0).

Time until starting and stopping

The meaning of the (LoadXOffTi and LoadXOnTi) is dependent of the generator or load setting.

The value configured at the LoadXOnTi I/O defines the time that the consumer/generator needs until it is switched on.

The value configured at the LoadXOffTi I/O defines the time that the consumer/generator needs until it is switched off.

Note

The values LoadXOffTi and LoadXOnTi are times that the block needs for calculation. These times are not monitored.

If these times are not specified correctly, it may have an effect on the switching operations of the block.

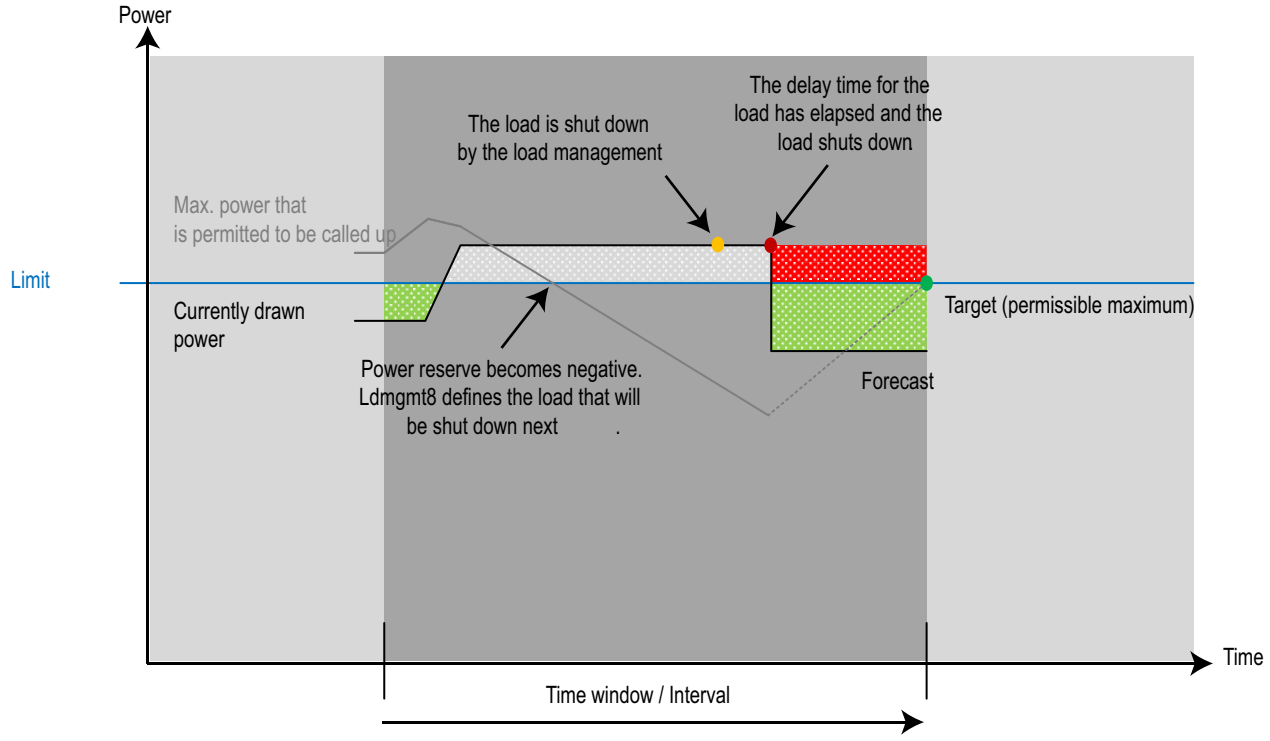
Consumer (LoadXOffTi):

Here, you can set a time for consumers that is located between the shutdown command of the LdMgmt8 block and the actual shutdown of the consumer. The time can, for example, be determined by the fact that a conveyor belt has to be emptied before it can be shut down. It may not be possible to switch on a full conveyor belt again.

Generator (LoadXOnTi):

For generators, the time set is that needed for the generator to start up after the start command of the LdMgmt8 block and for it to synchronize with the network. Only then can the generator be connected to the network.

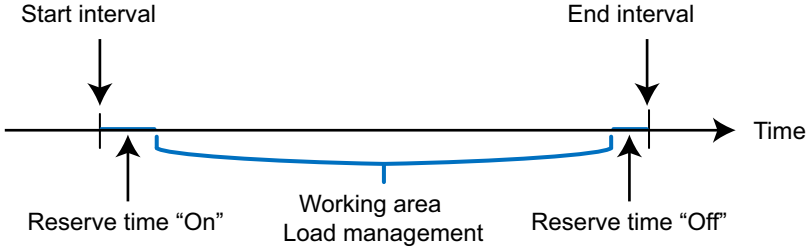
Generally speaking, consumers can only be shut down and generators connected when the delay time is greater than the residual time in the time interval. Because the times vary for each consumer or generator, a separate time can be configured for each channel.



Reserve time

With the reserve time, a time can be configured at the end (`DelayOff`) and start (`DelayOn`) of the time window during which no switching operations occur. The reserve time is deducted from the starting time. As a result, the switching points of the load management are delayed.

Otherwise this time is not taken into consideration when calculating the block.



Note

These reserve times should be kept rather short. (`DelayOn` max. ca. 10s/ `DelayOff` max. ca. 5s)

Control signals

The block has the following control signals (outputs that can be connected to the consumer) per consumer:

LoadXStart = Starting and stopping of the consumer in switch mode

LoadXLock = Locking or enabling of the consumer

Using the APL block "Trigger", the output Load1Start can also be used in button mode.

Number of units

A maximum of 8 units per block is possible. The number of units is automatically determined using an existing interconnection at the `LoadXRdyToStart` (only for generators: Feature.Bit8-15 = true) and `LoadXStarted` I/Os. The number of units is output at the `MaxLoad` output.

Interface to PeakMon

The LdMgmt8 block needs information from a PeakMon block for its function.

A connection **must** be created between the `BlkPeakMon` input of the LdMgment8 block and the `BlkLdMgmt` output of the PeakMon block.

This connection transfers the following I/O information in form of a structure:

- `PV` incl. ST
- `PV_UnitOut`
- `SP` incl. ST
- `SP_UnitOut`
- `PowerFor` incl. ST
- `PowerDif` incl. ST
- `PowerRes` incl. ST
- `PFor_UnitOut`
- `TimeSlot`
- `Time_Out`
- `CalcTime`
- `Status1`

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can switch to "automatic mode"
1	1 = Operator can switch to "manual mode"
2	Not used
3	1 = Operator can switch to "Out of service" mode

Bit	Function
4	1 = Operator can switch to passive
5	1 = Operator can switch to active
6-11	Not used
12	1 = Operator can activate the Release for maintenance function
13	1 = Operator can define the wait time for unit 1
14	1 = Operator can define the wait time for unit 2
15	1 = Operator can define the wait time for unit 3
6	1 = Operator can define the wait time for unit 4
17	1 = Operator can define the wait time for unit 5
18	1 = Operator can define the wait time for unit 6
19	1 = Operator can define the wait time for unit 7
20	1 = Operator can define the wait time for unit 8
21	1 = Operator can define the start/stop time for unit 1
22	1 = Operator can define the start/stop time for unit 2
23	1 = Operator can define the start/stop time for unit 3
24	1 = Operator can define the start/stop time for unit 4
25	1 = Operator can define the start/stop time for unit 5
26	1 = Operator can define the start/stop time for unit 6
27	1 = Operator can define the start/stop time for unit 7
28	1 = Operator can define the start/stop time for unit 8
29	1 = Operator can specify the delay time
30	1 = Operator can activate the "Delay time" function
31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
8	`Load1RdyToStart` active (device type: 0 = load)
9	`Load2RdyToStart` active (device type: 0 = load)
10	`Load3RdyToStart` active (device type: 0 = load)
11	`Load4RdyToStart` active (device type: 0 = load)
12	`Load5RdyToStart` active (device type: 0 = load)
13	`Load6RdyToStart` active (device type: 0 = load)
14	`Load7RdyToStart` active (device type: 0 = load)
15	`Load8RdyToStart` active (device type: 0 = load)
16	Forecast with separate display area and custom unit
19	See APL function: "Changes signal status of outputs in OOS"

Bit	Function
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"
26	Automatic parameter acceptance

8.2.4 of LdMgmt8 error handling

For troubleshooting all blocks, refer to the "Troubleshooting" section of the APL documentation in the basic information.

The following errors can be displayed for this block:

- Error numbers

Overview of error numbers

The ErrorNum I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
51	Invalid signal for `ModLiOp` = 1: • `ManModLi` = 1 and `AutModLi` = 1 `AutAct` = 1: • `OffAut` = 1 and `OnAut` = 1
72	No PeakMon block connected.
73	`MaxLoad` > 8
74	`MaxLoad` = 0
75	`MaxLoad` < 0
76	Invalid `SamleTime`: • `CalcTime` <= `SampleTime`
77	Invalid `TimeSlot`: • `CalcTime` >= `TimeSlot` • `DelayOn` >= `TimeSlot` or `DelayOn` < 0 • `DelayOff` >= `TimeSlot` or `DelayOff` < 0

8.2.5 LdMgmt8 messaging

Messaging

The following messages can be generated for this block:

- Process messages
- Control system fault

Control system fault

You can interconnect an external fault (signal) to input parameter `CSF`. If it changes to `CSF` = 1, a process control fault is triggered (`MsgEvId2`, SIG 2).

Process messages

Message instance	Message identifier	Message class	Event
MsgEvID1	SIG 1	Alarm - high	$$BlockComment$$ No consumer can be shut down
	SIG 2	AS process control message - fault	$$BlockComment$$ Consumer 1 locked
	SIG 3	AS process control message - fault	$$BlockComment$$ Consumer 2 locked
	SIG 4	AS process control message - fault	$$BlockComment$$ Consumer 3 locked
	SIG 5	AS process control message - fault	$$BlockComment$$ Consumer 4 locked
	SIG 6	AS process control message - fault	$$BlockComment$$ Consumer 5 locked
	SIG 7	AS process control message - fault	$$BlockComment$$ Consumer 6 locked
	SIG 8	AS process control message - fault	$$BlockComment$$ Consumer 7 locked
MsgEvID2	SIG 1	AS process control message - fault	$$BlockComment$$ Consumer 8 locked
	SIG 2	AS process control message - fault	$$BlockComment$$ External error has occurred

Explanation:

`$$BlockComment$$`: content of the instance-specific comment

You have the option of connecting an external fault (signal) to input parameter `CSF`. If this `CSF` = 1, a control system error is triggered (`MsgEvId2`, SIG 2).

Instance-specific messages

You can use up to four instance-specific messages with this block.

Message instance	Message identifier	Message class	Event
MsgEvID2	SIG 3	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 4	AS process control message - fault	$$BlockComment$$ External message 2
	SIG 5	AS process control message - fault	$$BlockComment$$ External message 3
	SIG 6	AS process control message - fault	$$BlockComment$$ External message 4

Explanation:

`$$BlockComment$$`: content of the instance-specific comment

Associated values for message instance MsgEvId1

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVal05`
5	`ExtVal06`
6	`ExtVal07`
7	`ExtVal08`
8	Reserved
9	Reserved
10	Reserved

8.2.6 LdMgmt8 I/Os

Input parameters

Parameter	Description	Data type	Default
`BlkPeakMon`	Connection to PeakMon Block	STRUCT	
`BlkPeakMon.BlockID`	BlockID PeakMon	INT	0
`BlkPeakMon.CalcTime`	Sample Time for the Power Forecast calculation [s]	DINT	0
`BlkPeakMon.PFor_UnitOut`	Power Forecast unit	INT	1190

Parameter	Description	Data type	Default
BlkPeakMon.PowerDif	Power Difference	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.PowerFor	Power Forecast	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.PowerRes	Power Rest	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.PV	Process Value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.PV_UnitOut	Process Value unit	INT	1190
BlkPeakMon.SP	Setpoint	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.SP_UnitOut	Setpoint unit	INT	1190
BlkPeakMon.Status1	Status1 Word	DWORD	16#00000000
BlkPeakMon.Time_Out	Current time [s]	DINT	0
BlkPeakMon.TimeSlot	Interval Time [s]	DINT	0
DelayEn	//1=Delay ON, 0=Delay OFF	BOOL	1
DelayOff	Delay Time End [s]	REAL	0.0
DelayOn	Delay Time Start [s]	REAL	10.0
Load1Device … Load8Device	Aggregat 1 ... 8 - Device Typ: 0 = Consumer / 1 = Generator	BOOL	0
Load1IdleTime … Load8IdleTime	Aggregat 1 ... 8 - IdleTime [s]	REAL	5.0
Load1OffTi … Load8OffTi	Aggregat 1 ... 8 - Operator input for Time Off [s]	REAL	5.0
Load1OnTi … Load8OnTi	Aggregat 1 ... 8 - Operator input for Time On [s]	REAL	5.0
Load1Prio … Load8Prio	Aggregat 1 ... 8 - Priority (1 - 99)	INT	1
Load1PV … Load8PV	Aggregat 1 ... 8 - act. Power	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#FF
Load1RdyToStart … Load8RdyToStart	Aggregat 1 ... 8 - ready to start	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
Load1SP … Load8SP	Aggregat 1 ... 8 - max. Power	REAL	0.0

Parameter	Description	Data type	Default
Load1Started ... Load8Started	Aggregat 1 ... 8 - started	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
LoadMode	0 = Normal / 1 = Consumers first / 2 = Generators first (PrefLoadMode = 0)	INT	0
OffAut	1=Off: Off Command in Auto Mode	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OffMan	1=Off: Off Command in Manual Mode	BOOL	1
OnAut	1=On: On Command in Auto Mode	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OnMan	1=On: On Command in Manual Mode	BOOL	0
PFor_OpScale	PowerFor - Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 0
PrefLoadMode	0 = Performance-oriented / 1 = Priority switch mode active	BOOL	1
PV_OpScale	PV - Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 0
SP_OpScale	SP- Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 0
TimeMode	0 = early / 1 = late shut-down of the Aggreates	BOOL	1

Output parameters

Parameter	Description	Data type	Default
DelayOffOut	Delay Time End [s]	REAL	0.0
DelayOnOut	Delay Time Start [s]	REAL	10.0
Load1Ctrl ... Load8Ctrl	Aggregat 1 ... 8 - Ctrl = 1	BOOL	0
Load1DeviceOut ... Load8DeviceOut	Aggregat 1 ... 8 - Device Typ: 0 = Consumer / 1 = Generator	BOOL	0
Load1IdleTiOut ... Load8IdleTiOut	Aggregat 1 ... 8 - IdleTime [s]	REAL	0.0
Load1Lock ... Load8Lock	Aggregat 1 ... 8 - Lock = 0	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
Load1PrioOut ... Load8PrioOut	Aggregat 1 ... 8 - Priority (1 - 99)	INT	1

Parameter	Description	Data type	Default
Load1PVAvOut ... Load8PVAvOut	Aggregat 1 ... 8 - act. Power act. interval (Analog Output/Average)	REAL	0.0
Load1PVOut ... Load8PVOut	Aggregat 1 ... 8 - act. Power (Analog Output)	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Load1SPAvOut ... Load8SPAvOut	Aggregat 1 ... 8 - max. Power act. interval (Analog Output/Average)	REAL	0.0
Load1SPOut ... Load8SPOut	Aggregat 1 ... 8 max. Power (Analog Output)	REAL	0.0
Load1Start ... Load8Start	Aggregat 1 ... 8 - Start = 1 / Stop 0 (Button Mode)	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
LoadAvail	Aggregates available Display	BYTE	16#00
LoadDevice	Aggregates Device Typ	BYTE	16#00
LoadLock	Aggregates lock Display	BYTE	16#00
LoadModeOut	0 = Performance-oriented / 1 = Priority switch mode active	INT	0
LoadNextSwitch	Aggregates switching Display	BYTE	16#00
LoadOn	Aggregates on Display	BYTE	16#00
LoadStart	Aggregates start Display	BYTE	16#00
MaxLoad	max. Number of aggregates	INT	0
NoLoadAvail	Number of aggregates on and available	INT	0
NoLoadRun	Compared Number of aggregates required/on	INT	0
Off	1= Aggr off	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
On	1= Aggr on	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PFor_UnitOut	PowerFor unit	INT	1190
PowerDif	Power Difference Value (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PowerFor	Power Forecast Value (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PowerRes	Power Rest Value (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PrefLoadModeOut	0 = Performance-oriented / 1 = Priority switch mode active	BOOL	0
PV_UnitOut	Process Value unit	INT	1190

Parameter	Description	Data type	Default
SP_UnitOut	Setpoint Value unit	INT	1190
Time_Out	Current time	DINT	0
TimeModeOut	0 = early / 1 = later shut-down of the aggregates	BOOL	0
TimeSlot_Out	Interval Time	DINT	0

8.2.7 LdMgmt8 block diagram

A block diagram is not provided for this block.

8.2.8 Operator control and monitoring

8.2.8.1 LdMgmt8 views

LdMgmt8 block views

The LdMgmt8 block has the following views:

- LdMgmt8 standard view (Page 292)
- Message view
- LdMgmt8 energy data view (Page 295)
- Trend view
- LdMgmt8 time view (Page 297)
- LdMgmt8 parameter view (Page 297)
- LdMgmt8 preview (Page 299)
- Notes view
- Batch view
- LdMgmt8 block icons (Page 301)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

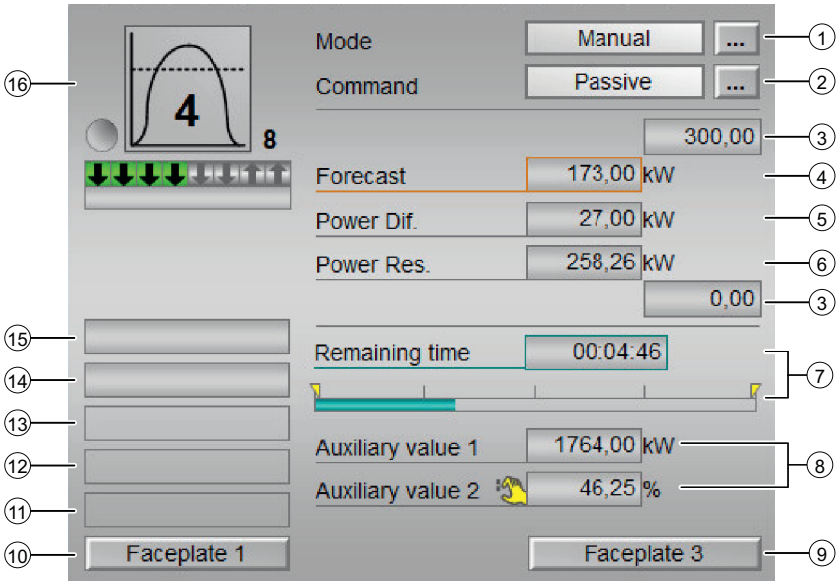
Additional displays within the views:

The clock indicates that a time is active at the block or at a specific unit.

The red exclamation mark indicates that there was a parameter change. Depending on the parameter assignment (Feature.Bit26), this change may require an operator intervention to apply the change.

8.2.8.2 LdMgmt8 standard view

LdMdmt8 standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can

be displayed here:

- Manual
- Automatic
- Out of service

For more information on changing the operating mode, refer to the "Switching operating states and operating modes" section in the APL manual.

(2) Displaying and switching the command

This area shows the specified operating state.

- Active
- Passive

For information on switching the state, refer to the section "Switching operating states and operating modes" in the APL manual.

(3) High and low scale range for the process value

These values provide information on the display range for the bar graph of the process value. The scale range is defined in the engineering system.

(4) Forecast

This area shows the forecast of the current energy consumption value with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "PowerFor" parameter.

(5) Power difference

This area shows the difference between forecast and available power with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "PowerDif" parameter.

(6) Power reserve

This area shows how much higher the power can be to stay within the available power limit with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "PowerRes" parameter.

(7) Time window

This area shows the remaining time for the active time window.

(8) Display of auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system.

For more information, refer to the section "Displaying auxiliary values" in the APL manual.

(9) Navigation button for switching to the standard view of any faceplate

Use this navigation button to switch to the standard view of the associated PeakMon.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "BlkPeakMon" parameter.

(10) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL manual.

(11) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

(12) Display area for block states

This area provides additional information on the operating state of the block:

- "PeakMon not active"

(13) Display area for block states

This area provides additional information on the operating state of the block:

- "Delay start"
- "Delay end"

(14) Display area for block states

This area provides additional information on the operating state of the block:

- "Earliest shutdown"
- "Latest shutdown"

(15) Display area for block states

This area provides additional information on the operating state of the block:

- "Priority"
- "Power"

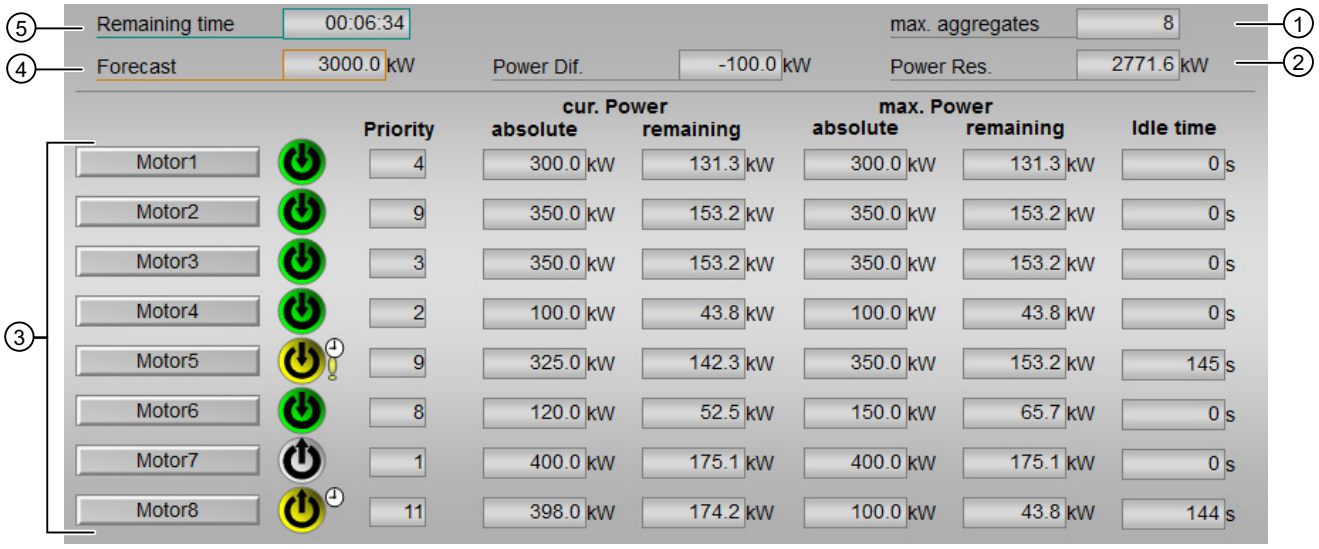
(16) Status display

The current status of the block and the assigned consumers and generators is graphically displayed here.

You can find more information in section LdMgmt8 block icons (Page 301).

8.2.8.3 LdMgmt8 energy data view

LdMgmt8 energy data view



(1) Max. aggregates

Maximum number of units that can be switched by the block

(2) Power reserve / power difference

Power difference

This area shows the difference between forecast and available power with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "PowerDif" parameter.

Power reserve

This area shows how much higher the power can be to stay within the available power limit with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "PowerRes" parameter.

(3) Navigation button for switching to the standard view of the connected unit/detail view of the unit states

- The display of this area depends on the maximum number of units configured in the CFC.
- Use these navigation buttons to reach the standard view of the connected block in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).
  Refer to the section "Opening additional faceplates" in the APL manual.
  The text in the navigation button can be configured at the OS additional text parameter of the "LoadXStarted" I/O. If no text is configured, the standard text "Motor X" is displayed.
- The display indicates whether a unit is available and/or switched on.
- It is displayed whether an unit is a consumer or a generator.
- The priority of the units is displayed.
- The specified power and the actual power of the units is displayed.

(4) Forecast

This area shows the forecast of the current energy consumption value with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

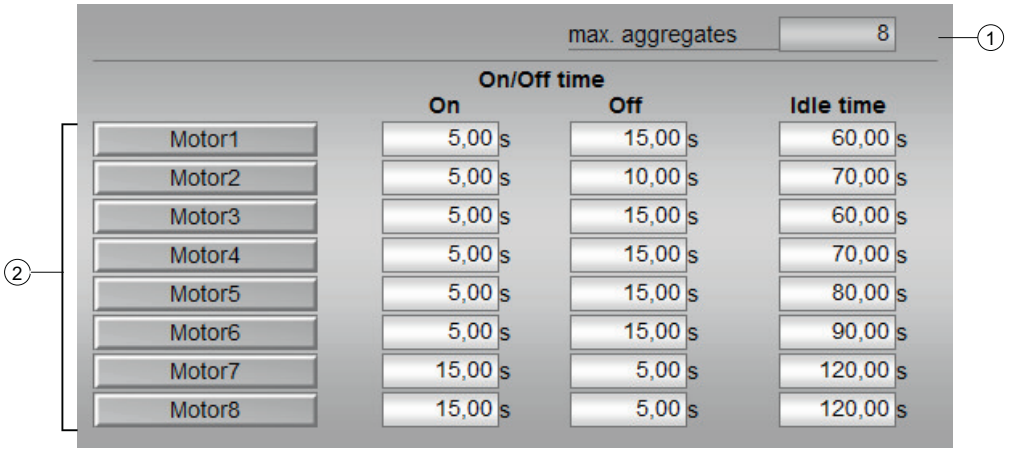
You can change the text for the process value at the "PowerFor" parameter.

(5) Remaining time

Remaining time in the current interval

8.2.8.4 LdMgmt8 time view

LdMdmt8 time view



(1) Number of units

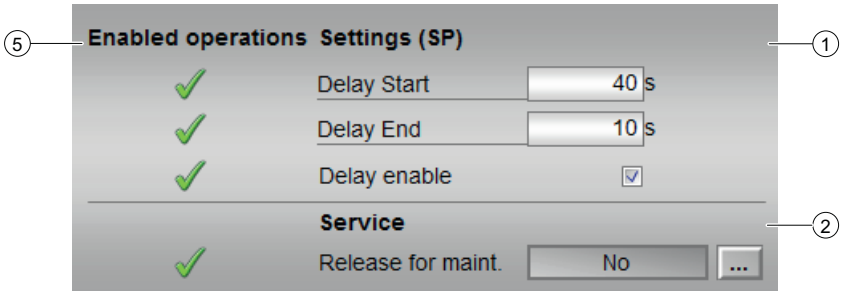
Number of units managed by the LdMgmt8 block

(2) Navigation button for switching to the standard view of the connected unit and delay times

- Use this navigation button to switch to the standard view of the connected unit
- You configure the text in the navigation button with the OS additional text parameter of the "LoadXStarted" I/O. If no text is configured, the standard text "Motor X" is displayed.
- With sufficient authorization, you can adjust the times that the respective unit needs until it actually starts or stops.

8.2.8.5 LdMgmt8 parameter view

LdMdmt8 parameter view



(1) Settings for the reserve time

- Delay start: Reserve time at the beginning of the work area
- Delay end: Reserve time at the end of the work area
- Activate "Delay"

(2) Release for maintenance

You can select the following functions in this area:

- "Release for maintenance"

See the section "Switchover of operating state and operating mode" in the APL manual.
For information on this area, see the following sections in the APL manual:

- Release for maintenance

(3) Enabled operations

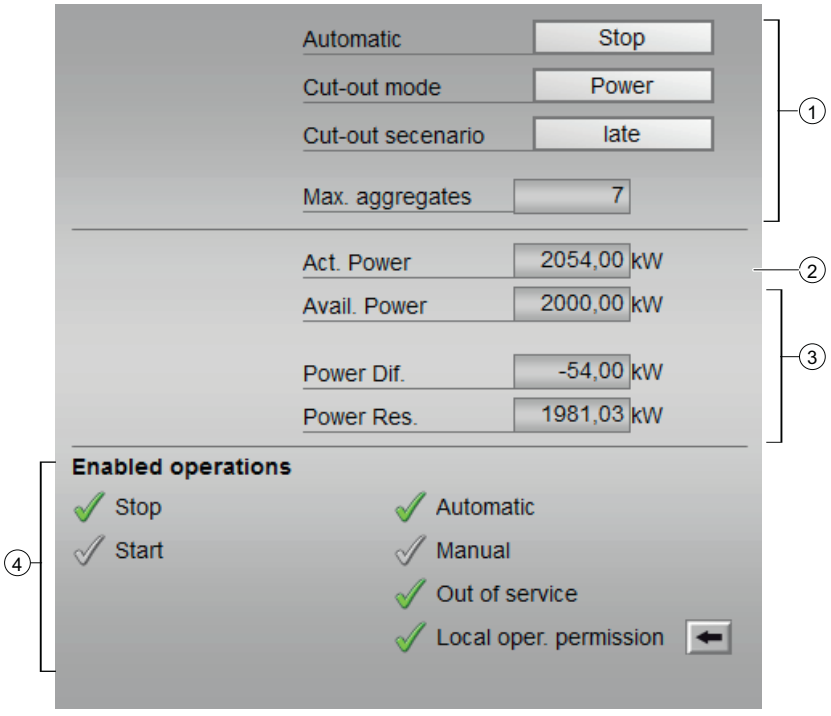
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- Green check mark: The OS operator can control this parameter
- Gray check mark: The OS operator cannot control this parameter at this time due to the process
- Red cross: The OS operator is generally prohibited from accessing this parameter due to the configured AS operator control permissions (OS_Perm or OS1Perm).

8.2.8.6 LdMgmt8 preview

LdMdmt8 preview



(1) Preview

- Automatic: This area shows the command setting for automatic mode.
- Shutdown mode: This area shows the set shutdown mode.
- Shutdown scenario: This area shows the scenario according to which the block is shut down.
- Max units: This area shows the number of units monitored and controlled by the block.

(2) Process values

This area shows the current power.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "PV_Out" parameter.

(3) Preview area

This area shows you a preview for the following values:

- Available power
  If text is configured for this command, it is displayed as additional text and as a button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.
  You can change the text for the process value at the "SP_Out" parameter.
- Power difference
  If text is configured for this command, it is displayed as additional text and as a button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.
  You can change the text for the process value at the "PowerDif" parameter.
- Power reserve
  If text is configured for this command, it is displayed as additional text and as a button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.
  You can change the text for the process value at the "PowerFor" parameter.

(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL manual.

(5) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- Green check mark: The OS operator can control this parameter
- Gray check mark: The OS operator cannot control this parameter at this time due to the process
- Red cross: The OS operator cannot control this parameter due to the configured AS operator permissions (OS_Perm or OS1Perm).

The following enabled operations are shown here:

- "Automatic": You can switch to "automatic mode".
- "Manual": You can switch to "manual mode".
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to change to the standard view of the OpStations/UserM block. Additional information on this topic is available in the section "Operator control permissions" in the APL manual and in the section "Operation of multiple control rooms" in this manual.

- "Stop": You can stop load management.
  If text is configured for this command, it is also displayed in brackets. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.
- "Start": You can start load management.
  If text is configured for this command, it is also displayed in brackets. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

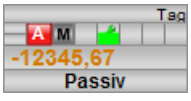
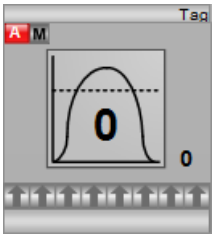
8.2.8.7 LdMgmt8 block icons

LdMgmt8 block icons

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of the alarm and warning limits as well as process control errors
- Operating modes
- Signal status, release for maintenance
- Memo display
- Status display

The block icons from template @TemplateILV8.PDL:

Icons	Selection of the block icon in CFC
	1
	2

You can find additional information on the block icon and the control options in the block icon in the following sections of the APL manual:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

The representation of the unit is different in the icon and image window depending on the type and status of the unit:

Generator:

A generator is displayed with an up arrow.



Consumer:

A consumer is displayed with a down arrow.



Colors:

The status is indicated by colors:

- Green → Object is active:
  - Motor or unit is running
  - LdMgm8 is active but does not make any switching operation
- Gray → Object inactive:
  - Motor or unit is stopped
  - LdMgmt8 is passive
- Yellow → Switching operation active:
  - Switching operation pending at motor or unit
  - LdMgmt8 currently executes switching operation.
- Red → Fault status:
  - LdMgmt8 cannot prevent the limit from being exceeded, because generators can no longer be connected and consumers can no longer be switched off.

Number:

This number represents the currently available aggregates for load management (generator or load).

The available aggregates correspond to the number of possible switching operations (switch off consumption / switch on generator) in order to remain below the power supply company limit.

8.3 CalcPwrL - Acquisition and calculation of energy consumption values

8.3.1 CalcPwrL description

Object name (type + number) and family

Type + number: FB 1193

Family: EngyMgmt

Number of process objects: 2 PO

Area of application for CalcPwrL

The block is for the acquisition and calculation of energy values for units and plant sections

Method of operation

The block shows the current power consumption and calculates the power consumption to be expected (forecast). In addition, the consumption for two time periods can be summed by this block.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB100).

Startup characteristics

Use the Feature Bit 'Setting the startup characteristics' to define the startup characteristics for this block.

The messages are suppressed after startup for the number of cycles set at `RunUpCyc`.

Status word allocation for `Status1` parameter

For a description of the individual parameters, refer to the section CalcPwrL I/Os (Page 314)

Status bit	Parameter
0	`Occupied`
1	`BatchEn`
2	`SimOn`
3	`OosAct`.Value
4	`OosLi`.Value
5	`OnAct`.Value
6	Not used

Status bit	Parameter
7	Interval mode: External trigger
8	Interval mode: Fixed time window
9	Not used
10	SimLiOp.Value
11	Delay of the PV_AH_Lim message
12	Delay of the PV_WH_Lim message
13 - 14	Not used
15	Interval error
16	Interval changed
17	Collection of message delays
18	PowerFor with custom scale and unit
19	Trigger error
20	Interval adaptation
21	Not used
22	PeakMon interconnected and active
23 - 29	Not used
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word allocation for Status2 parameter

For a description of the individual parameters, refer to the section CalcPwrL I/Os (Page 314)

Status bit	Parameter
0	MsgLock
1	PFor_AH_Act.Value
2	PFor_WH_Act.Value
3 - 6	Not used
7	PFor_AH_En
8	PFor_WH_En
9 - 12	Not used
13	PFor_AH_MsgEn
14	PFor_WH_MsgEn
15 - 16	Not used
17	CalcTime changed
18 - 21	Not used
22	GradEn
23 - 27	Not used
28	0 = Falling measured value / 1 = Rising measured value
29 - 30	Not used
31	MS_RelOp

Status word allocation for `Status3` parameter

For a description of the individual parameters, refer to the section CalcPwrL I/Os (Page 314)

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via `EventTsIn`
1	Effective signal 2 of the message block interconnected via `EventTsIn`
2	Effective signal 3 of the message block interconnected via `EventTsIn`
3	Effective signal 4 of the message block interconnected via `EventTsIn`
4	Effective signal 5 of the message block interconnected via `EventTsIn`
5	Effective signal 6 of the message block interconnected via `EventTsIn`
6	Effective signal 7 of the message block interconnected via `EventTsIn`
7	Effective signal 8 of the message block interconnected via `EventTsIn`
8 - 23	Effective signal 8 to 16 of the message block connected via `Event16TsIn`
24-31	Not used

8.3.2 CalcPwrL operating modes

The block can be operated using the following modes:

- On
- Out of service

The next section provides additional block-specific information relating to the general descriptions.

"On"

You can find general information about the "On" mode in the "On" section of the APL manual.

"Out of service"

You can find general information about the "out of service" mode in the "Out of service" section of the APL manual.

8.3.3 CalcPwrL functions

CalcPwrL functions

The functions for this block are listed below.

Time window

The time window defines the time period for the power limit.

The time window can be set externally with a pulse or defined using a parameterizable assignable time (default: 15 minutes for electrical consumption).

Interval characteristics

The intervals can be controlled in two ways:

- Controlled externally by a trigger (e.g. by the power supply company)
- Fixed specified time window

The mode of operation is set in Feature.Bit 9.

CalcPwrL is the central block for setting the interval. Starting from this block, the time interval is forwarded to the downstream CalcPwrL blocks.

Trigger-controlled (Feature.Bit 9 = 1)

If the time window is controlled externally by a trigger, the expected time window (e.g. 15 minutes) still has to be configured.

If a trigger fails to appear at the end of the interval (1), the block starts a new interval and issues a process control message. The duration of the interval depends on the parameter assignment.

If the trigger arrives delayed (2), the block starts a new interval with the trigger signal.

A higher priority thereby results for the trigger signal than for the configured interval time.

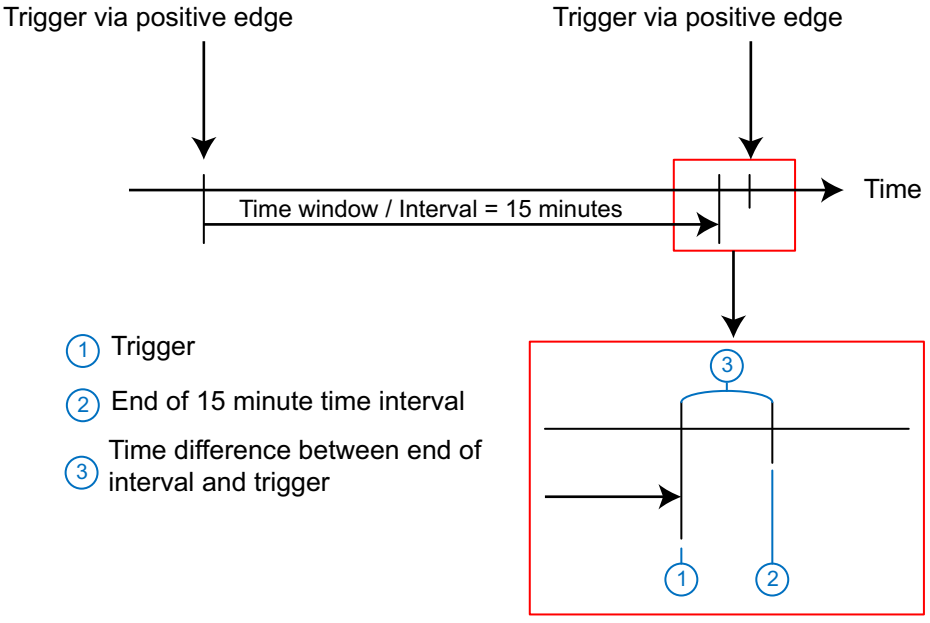


Figure 8-1 Power supply company trigger

Fixed time window (Feature.Bit 9 = 0)

The duration of the interval is set with a parameter for the fixed time window. The start of the interval aligns with the clock time (CPU time) at the full hour. This behavior increases the plausibility in later evaluations because it is made a lot more difficult by asynchronous behavior.

Example:

You specify an interval of 15 minutes. You enter the information at 12:35 PM. Because the interval start aligns with the time, the block does not start "working" immediately after the parameter assignment but rather at 12:45 PM.

Plausibility

The specified time window is always checked for plausibility. The formulas in the example below are used to check whether the time base of the specified time window is plausible for an hour or a day.

If the value is not plausible, an error is output and a text is displayed in the faceplate in the standard view. The block goes into the error state.

The following settings are permitted and checked for plausibility:

- On hourly basis:
  - Y=24/X
    Examples: Y= 24/1='True' or Y= 24/9='False'
- On minute basis
  - Y=60/X
    Examples: Y= 60/15='True' or Y= 60/13='False'

Supply signal

An analog value of the currently consumed power is expected at the input.

If no analog value is available, you can form the required analog counter value from a digital pulse (for example from the pulse output of a quantity counter) with the PulseCon block.

Calculation functions

The block works with the following values:

Available power (power limit)

The available power is the parameter that represents the power limit. This value must not be exceeded in the mean value of the time interval.

The value is always in effect for one time window. If the value is changed, the modified value is applied in the faceplate and takes effect in the next interval. The change is also displayed at the bar graph in the faceplate and as text.

Forecast

The forecast indicates the power consumption expected in the current time window for the entire time window based on the previous power consumption.

The parameter is calculated as follows:

$$ForeCast = Sum\ of\ the\ current\ power\ consumption\ of\ the\ elapsed\ time + \left(\frac{Current\ power}{Intervall\ time} * (Intervall\ time - Elapsed\ time)\right)$$

$$Sum\ of\ the\ current\ power\ consumption\ of\ the\ elapsed\ time$$
$$= Sum\ of\ the\ current\ power\ consumption\ of\ the\ elapsed\ time\ (n^{-1}) + \frac{Current\ power}{Intervall\ time}$$

Here, the forecast corresponds to the currently measured value under the assumption that it is constant.

Energy consumption values (counters)

The measured value `PV`.Value is accumulated in two counter registers, `AcCnt1Out` and `AcCnt2Out`.

The measured value is transferred to the `PV`.Value input.

The `CalcTime` parameter specifies how often the calculation takes place per interval. This means it specifies the integration sampling cycles in seconds. The `CalcTime` value must be greater than the `SampleTime` of the block and cannot be less than the values at the `TimeSlot` and `BasePV` I/Os.

The measured value `PV` is weighted at the `BasePV` parameter. It is essential to ensure that the `CalcTime` and `BasePV` parameters have the same unit (e.g., `BasePV` = 3600 must be specified for a measured value with the unit m³/h and a CalcTime of 1 second).

A deadband can be configured at the `DeadBand` parameter. If `PV` < `DeadBand`, it is not included in the integration.

The count value is calculated with the following formula:

$$AcCnt1Out = AcCnt1Out + PV * FactCnt1 * \frac{Sum\ of\ the\ current\ power\ consumptions\ of\ the\ elapsed\ time}{BasePV}$$

$$AcCnt2Out = AcCnt2Out + PV * FactCnt2 * \frac{Sum\ of\ the\ current\ power\ consumptions\ of\ the\ elapsed\ time}{BasePV}$$

The counter state can be reset via the faceplate or with the interconnectable inputs `RstCnt1Li` / `RstCnt2Li`.

The `Rst1Date` / `Rst2Date` parameters specify the time stamp for the reset operations in the format [yy/mm/dd_hh].

Time base

The two dependent values of the block are standardized to seconds in the calculations. This means an hour is represented as 3600 seconds and 15 minutes as 900 seconds.

Low pass filter

The block smooths the input variable (I/O) after the delay 1. order time delay. This delay time can be configured (SmoothTi connection). The block works according to the following formula:



Where:

- PV_Out = output value
- SmoothTi = delay time
- SampleTime = sampling time
- PV = input value

The formula only applies to SmoothTi > 0. If SmoothTi = 0, the input is passed directly to the output. If the input value is outside the REAL range limits, the calculation is paused. When the input value is outside the range limits again, the calculation is resumed automatically.



Flutter suppression

The block has the flutter suppression function for the input InTrig. Flutter suppression is activated at the "MonitorFlut" input. The time for flutter suppression is configured at the "MonTiFlut" input.

Calling additional blocks

This block includes the standard function Call additional faceplates. Information is available in the section *Calling additional faceplates* in the *APL Function Manual*.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the OS_Perm parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4	1 = Operator can change the time interval
5	Not used
6	1 = Operator can change the simulation value SimPV

Bit	Function
7	1 = Operator can reset the maximum values
8 - 10	Not used
11	1 = Operator can activate the Simulation function
12	1 = Operator can activate the Release for maintenance function
13	1 = Operator can change the limit (`PowerFor`) for the high alarm
14	1 = Operator can change the limit (`PowerFor`) for the high warning
15	Not used
16	1 = Operator can change the limit (`PowerFor`) for the hysteresis
17	1 = Operator can set the trigger monitoring time
18	1 = Operator can activate the trigger monitoring function
19	1 = Operator can set the flutter monitoring time
20	1 = Operator can activate the flutter monitoring function
21	1 = Operator can change the default counter 1 parameter `AcCnt1_Rst`
22	1 = Operator can change the default counter 2 parameter `AcCnt2_Rst`
23	1 = Operator can change the deadband parameter `DeadBand`
24	1 = The operator can enable/disable messages via `PFor_WH_MsgEn`
25	1 = The operator can enable/disable messages via `PFor_WH_MsgEn`
26	Not used
27	1 = Operator can reset accumulated value of counter 1
28	1 = Operator can reset accumulated value of counter 2
29	Not used
30	1 = Operator can change the smoothing time parameter `SmoothTi`
31	Not used

Select unit of measure

This block includes the standard APL function, Select unit of measure.

The default unit is kW (1190), depending on the parameter.

Displaying auxiliary values

This block provides the standard APL function, "Display auxiliary values".

SIMATIC BATCH functionality

This block provides the standard APL function, "SIMATIC BATCH functionality".

Group display `SumMsgAct` for limit monitoring, `CSF` and `ExtMsgx`

The block has the standard APL function, group display for limit monitoring, `CSF` and `ExtMsgx`.

Limit monitoring of the process value

This block has a limit monitoring of the forecast for violation.

Enabling and disabling messages

You can use the `XX_MsgEn` I/Os to individually enable or disable the pending messages of the limits.

All messages are enabled by default.

To disable messages for the high alarm, for example, you set the associated I/O `AH_MsgEn` = 0.

You can disable all messages with the I/O `MsgLock` = 1.

Limit monitoring with hysteresis

This block includes the standard APL function, Limit monitoring with hysteresis. It is performed via the input parameter `PFor_Hyst`.

Generating instance-specific messages

This block includes the standard APL function, "Generating instance-specific messages".

Simulating signals

This block features the standard APL function, "Simulating signals".

If simulating is active, the currently measured power consumption (`PV`) of `SimPV` and `SimPV_Li` is applied.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
9	1 = External trigger / 0 = Fixed time window
16	Forecast with separate display area and custom unit
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"
26	See APL function: "Reaction of the switching points in the "Out of service" operating mode"
28	See APL function: "Deactivate switching points"
29	See APL function: "Signal limit violation"
30	Activate the interconnection to the PeakMon block (`BlkPeakMon`)

Suppressing messages using the `MsgLock` parameter

This block includes the standard APL function, "Suppress messages using the MsgLock parameter".

Release for maintenance

This block has the standard APL function, "Release for maintenance".

Flutter suppression

The block provides the APL standard function Flutter suppression.

8.3.4 CalcPwrL error handling

For troubleshooting all blocks, refer to the "Troubleshooting" section of the APL documentation in the basic information.

The following errors can be displayed for this block:

- Error numbers
- Process control messages

Overview of error numbers

The ErrorNum I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
30	The value `PV` can no longer be represented in the REAL numerical field.
43	TimeFactor < 0 or > 2
44	Invalid `SampleTime`: • `CalcTime` <= `SampleTime` • `IdleTime` <= `SampleTime` • `TimeSlot` <= `SampleTime`
45	Invalid `TimeSlot`: • `CalcTime` >= `TimeSlot` • `IdleTime` >= `TimeSlot` • 60 > `TimeSlot`
46	Invalid `BasePV`: • `CalcTime` >= `BasePV`
47	Invalid `FactCnt1`: • `FactCnt1` <= 0.0

Error number	Meaning of the error number
48	Invalid FactCnt2: • FactCnt2 <= 0.0
72	PeakMon block is not correctly interconnected.

External control system fault (CSF)

This block has the standard function, External control system fault, and gives you the option to connect an external control system fault with the CSF parameter.

8.3.5 CalcPwrL messaging

Message functionality

The following messages can be generated for this block:

- Control system fault
- Process messages
- Instance-specific messages

Control system fault

You can interconnect an external fault (signal) to input parameter CSF. If this CSF = 1, a control system fault is triggered (MsgEvId1, SIG 5).

Process messages

Message instance	Message identifier	Message class	Event
MsgEvID1	SIG 1	Alarm - high	$$BlockComment$$ PowerFor - High alarm limit violated
	SIG 2	Warning - high	$$BlockComment$$ PowerFor - High warning level violated
	SIG 3	AS control system message - fault	$$BlockComment$$ Trigger flutters
	SIG 4	AS control system message - fault	$$BlockComment$$ Trigger not received
	SIG 5	AS control system message - fault	$$BlockComment$$ External fault has occurred

Explanation:

$$BlockComment$$: content of the instance-specific comment

You have the option of connecting an external fault (signal) to input parameter CSF. If this CSF = 1, a control system fault is triggered (MsgEvId1, SIG 5).

Instance-specific messages

You have the option of using up to two instance-specific messages for this block.

Message instance	Message identifier	Message class	Event
MsgEvID1	SIG 6	AS control system message - fault	$$BlockComment$$ External message 1
	SIG 7	AS control system message - fault	$$BlockComment$$ External message 2
	SIG 8	AS control system message - fault	$$BlockComment$$ External message 3

Explanation:

`$$BlockComment$$`: content of the instance-specific comment

Associated values for message instance MsgEvId1

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVal04`
5	`ExtVal05`
6	`ExtVal06`
7	`ExtVal07`
8	`ExtVal08`
9	Reserved
10	Reserved

8.3.6 CalcPwrL I/Os

Input parameters

Parameter	Description	Data type	Default
`AcCnt_Unit`	Unit of AcCnt1 / AcCnt2	INT	1001
`AcCnt1`	Accumulated counter #1	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
`AcCnt2`	Accumulated counter #2	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
`BasePV`	Factor for PV	REAL	3600.0

Parameter	Description	Data type	Default
BlkPeakMon	Connection to PeakMon block	STRUCT	-
BlkPeakMon.BlockID	BlockID PeakMon	INT	0
BlkPeakMon.CalcTime	Sample Time for the Power Forecast calculation [s]	DINT	0
BlkPeakMon.PFor_UnitOut	Power Forecast unit	INT	1190
BlkPeakMon.PowerDif	Power Difference	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.PowerFor	Power forecast	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.PowerRes	Power Rest	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.PV	Process value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.PV_UnitOut	Process Value unit	INT	1190
BlkPeakMon.SP	Setpoint	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkPeakMon.SP_UnitOut	Setpoint unit	INT	1190
BlkPeakMon.Status1	Status1 Word	DWORD	16#00000000
BlkPeakMon.Time_Out	Current time [s]	DINT	0
BlkPeakMon.TimeSlot	Interval Time [s]	DINT	0
CalcTime	Sample Time for the Power Forecast calculation [s]	DINT	1
FactCnt1	Dimension factor for PV (Counter 1)	REAL	1.0
FactCnt2	Dimension factor for PV (Counter 2)	REAL	1.0
GradAct	Gradient active	BOOL	1
IdleTime	Idle time new interval [s]	REAL	5.0
InTrig	Linkable input for external Trigger	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
LagTime	Lag Time Constant [s]	REAL	1.0
MonitorFlut	1=Monitor ON, 0=Monitor OFF: Fluttering monitor	BOOL	1
MonitorTrig	1=Monitor ON, 0=Monitor OFF: Trigger monitor	BOOL	1
MonTiFlut*	Monitoring time for the Fluttering [s]	REAL	3.0

Parameter	Description	Data type	Default
MonTiTrig*	Monitoring time for the Trigger [s]	REAL	3.0
PFor_AH_DC	PowerFor - Alarm Delay Time Coming High [s]	REAL	0.0
PFor_AH_DG	PowerFor - Alarm Delay Time Going High [s]	REAL	0.0
PFor_AH_En	PowerFor - High Alarm Enable	BOOL	1
PFor_AH_Lim*	PowerFor - High Alarm Limit	REAL	95.0
PFor_AH_MsgEn*	Power Forecast Value - High Alarm Message Enable	BOOL	1
PFor_Hyst*	PowerFor - Alarm Hysteresis	REAL	1.0
PFor_OpScale*	PowerFor - Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 0
PFor_Unit*	Unit of Power Forecast Value	INT	1190
PFor_WH_DC	PowerFor - Warning Delay Time Coming High [s]	REAL	0.0
PFor_WH_DG	PowerFor - Warning Delay Time Going High [s]	REAL	0.0
PFor_WH_En	PowerFor - High Warning Enable	BOOL	1
PFor_WH_Lim*	PowerFor - High Warning Limit	REAL	90.0
PFor_WH_MsgEn*	PowerFor - High Warning Message Enable	BOOL	1
Rst1Date	Date of last counter #1 reset [yy/mm/dd_hh]	DWORD	16#00000000
Rst2Date	Date of last counter #2 reset [yy/mm/dd_hh]	DWORD	16#00000000
RstCnt1Li	Linkable input reset counter #1	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
RstCnt1Op *	Operator input reset counter #1	BOOL	0
RstCnt2Li	Linkable input reset counter #2	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
RstCnt2Op *	Operator input reset counter #2	BOOL	0
SmoothTi	Smooth Time Constant [s]	REAL	0.0
TimeSlot*	Operator input for setting time	DINT	900

* Values can be written back to these inputs during execution of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
AcCnt1Out	Accum. counter #1	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
AcCnt1T0_Rst	Accum. counter #1 after reset (x (t))	REAL	0.0
AcCnt1T1_Rst	Accum. counter #1 after reset (x (t - 1))	REAL	0.0
AcCnt1T1Out	Accum. counter #1	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
AcCnt2Out	Accum. counter #2	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
AcCnt2T0_Rst	Accum. counter #1 after reset (x (t))	REAL	0.0
AcCnt2T1_Rst	Accum. counter #1 after reset (x (t - 1))	REAL	0.0
AcCnt2T1Out	Accum. counter #2	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
AcDiCnt1Out	Accum. counter #1 (DINT)	DINT	0
AcDiCnt1T1Out	Accum. counter #1 (DINT)	DINT	0
AcDiCnt2Out	Accum. counter #2 (DINT)	DINT	0
AcDiCnt2T1Out	Accum. counter #2 (DINT)	DINT	0
CalcTime_Out	Sample time for the power forecast calculation [s]	DINT	0
DateCnt	Current date [yy/mm/dd_hh]	DWORD	16#00000000
FlutAct	Fluttering switch active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PFor_AH_Act	PowerFor - Alarm High Active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PFor_AH_Out	PowerFor - High Alarm Limit Output	REAL	0.0
PFor_HysOut	PowerFor - Alarm Hysteresis Output	REAL	0.0
PFor_OpScaleOut	PowerFor - Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 100
PFor_UnitOut	PowerFor unit	INT	1190
PFor_WH_Act	PowerFor - Warning High Active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PFor_WH_Out	PowerFor - High Warning Limit Output	REAL	0.0

Parameter	Description	Data type	Default
PowerFor	Power Forecast Value (analog output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_AvT0 ... PV_AvT1	Process Value Average Interval 0 ... 1 (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_Grad	PV - Gradient Value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_GradNP	PV - Gradient Value minimum Peak Value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_GradPP	PV - Gradient Value maximum Peak Value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_MaxT0 ... PV_MaxT1	Process Value Max. Interval 0 ... 1 (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_MinT0 ... PV_MinT1	Process Value Min. Interval 0 ... 1(Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_OpScaleOut	PV - Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 0
PV_Out	Process Value (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_UnitOut	PV unit	INT	1190
TdAs	AS time of the day	DINT	0
TdAs	AS time of the day	DINT	0
Time_Out	Current time [s]	DINT	0
TimeSlot_Bad	TimeSlot error	BOOL	0
TimeSlot_Out	Current time slot [s]	DINT	0

8.3.7 CalcPwrL block diagram

A block diagram is not provided for this block.

8.3.8 Operator control and monitoring

8.3.8.1 CalcPwrL views

CalcPwrL block views

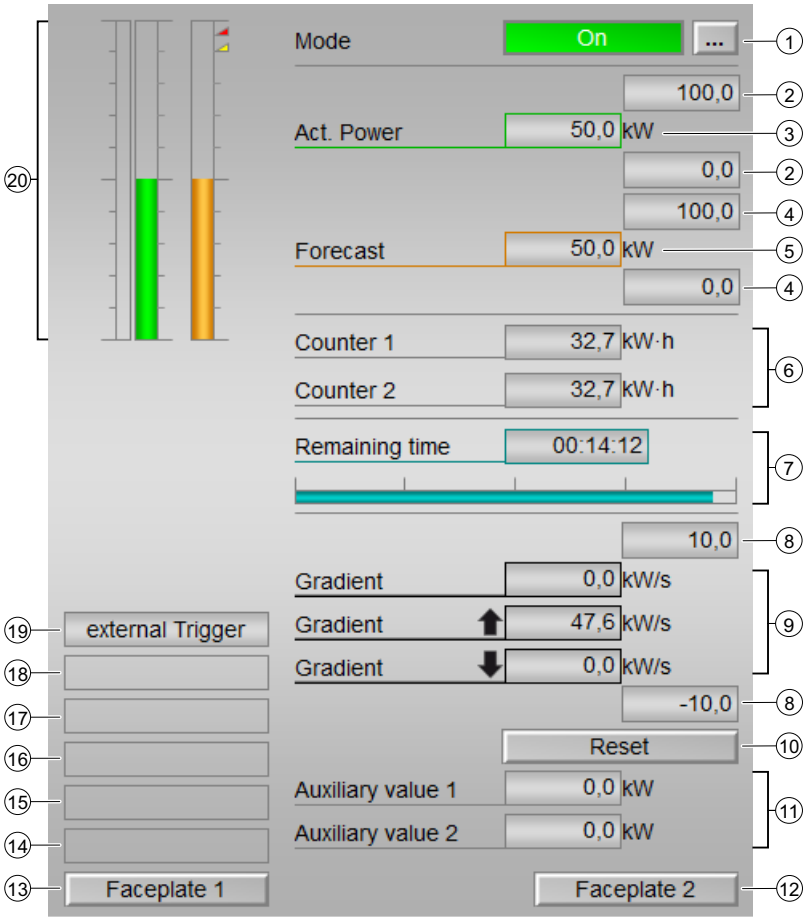
The CalcPwrL block provides the following views:

- CalcPwrL standard view (Page 320)
- Message view
- Limit view of CalcPwrL (Page 323)
- Trend view
- CalcPwrL parameter view (Page 325)
- CalcPwrL preview (Page 327)
- CalcPwrL histogram view (Page 330)
- Memo view
- Batch view
- Block icon for CalcPwrL (Page 331)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

8.3.8.2 CalcPwrL standard view

CalcPwrL standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can

be displayed here:

- On
- Out of service

For more information on changing the operating mode, refer to the "Switching operating states and operating modes" section in the APL manual.

(2) (4) High and low scale range for the process value

These values provide information on the display range for the bar graph of the process value. The scale range is defined in the engineering system.

(3) Current power

This area shows the currently consumed power with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and as a button label for command selection.
Additional information is available in the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "PV_Out" parameter.

(5) Forecast

This area shows the forecast of the current energy consumption value with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "PowerFor" parameter.

(6) Counter

This section shows the state of the two counters of the block.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "AcCnt1Out" and "AcCnt2Out" parameters.

(7) Time window

This area shows the remaining time for the active time window.

(8) High and low scale range for the gradient value

These values provide information on the display range for the bar graph of the gradient.

(9) Display of the gradient

This area shows the current, minimum and maximum gradient value and the rise and fall of the value. This display of the minimum and maximum gradient value functions like a min/max pointer.

(10) Reset the peak values of the gradient

You can use these buttons to reset the maximum or minimum peak value of the gradient.

(11) Display of auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system.

For more information, refer to the section "Displaying auxiliary values" in the APL manual.

(12) Navigation button for switching to the standard view of any faceplate

Use this navigation button to switch to the standard view of the associated PeakMon.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "BlkPeakMon" parameter.

(13) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL manual.

(14) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

(15) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

(16) Display area for block states

This area provides additional information on the operating state of the block:

- "Delay time changed"
- "CalcTime changed"

(17) Display area for block states

This area provides additional information on the operating state of the block:

- "Trigger error"

(18) Display area for block states

This area provides additional information on the operating state of the block:

- "Interval changed"
- "Interval adaptation"
- "Interval error"

(19) Display area for block states

This area provides additional information on the operating state of the block:

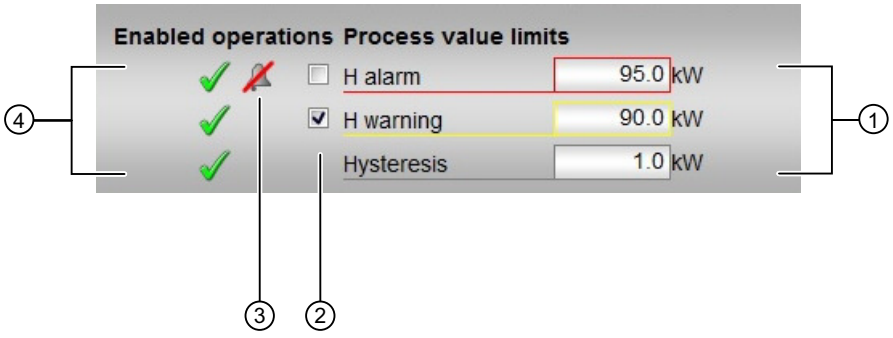
- "External trigger"
- "Fixed interval"

(20) Status display

This area shows the current "process values" in the form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

8.3.8.3 Limit view of CalcPwrL

CalcPwrL limit view



(1) Process value limits

In this area, you can enter the limits for the process value. Refer to the "Changing values" section in the APL manual.

You can change the following limits:

- "H alarm": Alarm high
- "H warning": Warning high
- "Hysteresis"

You can enable/disable messages by setting the check mark.

(2) Suppress messages

You can enable/disable messages by setting the check mark.

(3) Message suppression/delay

Message suppression indicates whether the suppression of the associated message in the AS block is activated with the `xx_MsgEn` parameter. The output of messages is not suppressed when the block is installed (all `xx_MsgEn` parameters are preset to 1). Messages can only be output if limit monitoring of the additional analog value has been enabled.

Alarm delays are also displayed in this position; for more on this, see section Area of application of the alarm delays in the APL documentation.

(4) "Enabled operations"

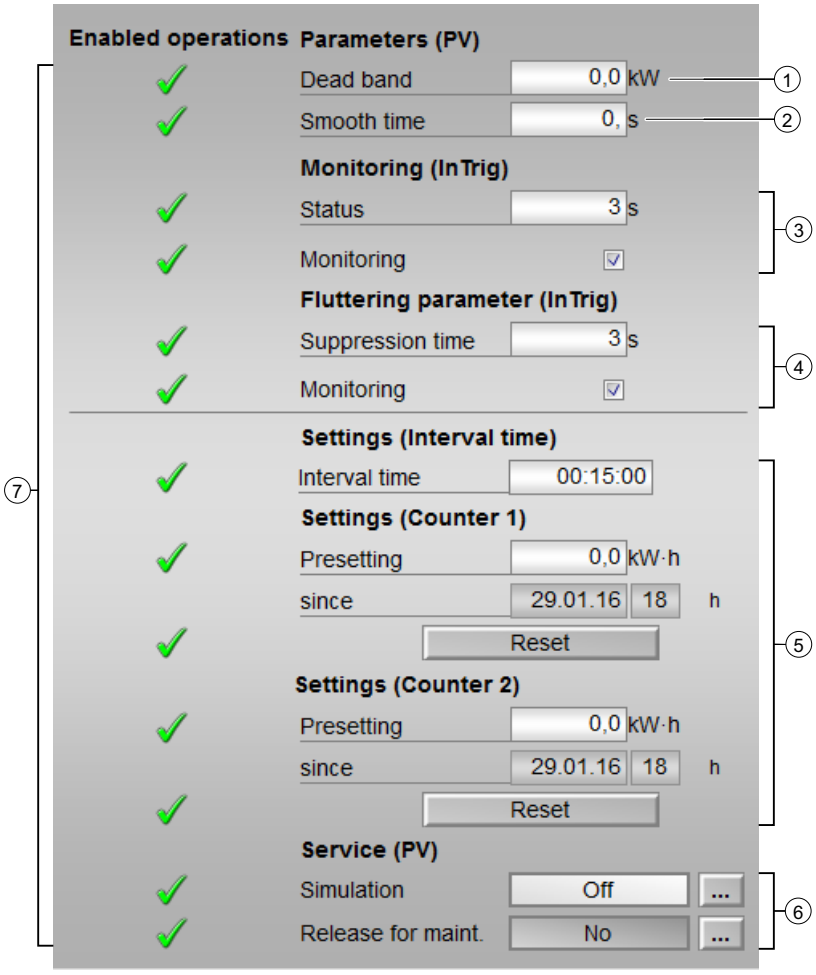
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Symbols for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: The OS operator cannot control this parameter at this time due to the process
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

8.3.8.4 CalcPwrL parameter view

CalcPwrL parameter view



(1) Parameters

In this area, you change parameters and thereby influence the block:

- "Deadband": Deadband width, see description of deadband in the *Advanced Process Library* manual
- "Smoothing time": Time for smoothing the process value

(3) Monitoring (area is only visible when interval is switched by an external trigger "Feature.Bit 9 = 1")

In this area, you change parameters and thereby influence the block:

- "Status": Monitoring time during runtime.
- Activate "Monitoring"

(4) Flutter parameter (area is only visible when interval is switched by an external trigger "Feature.Bit 9 = 1")

In this area, you change parameters and thereby influence the block:

- Suppression time: Enter the time period during which signal flutter is suppressed on the trigger signal.
- Activate "Monitoring"

(5) Settings

This area shows the current setting for the time window of the interval time and the counters.

If the interval time is not specified by an interconnection (e.g. from PeakMon), it can be changed here depending on the operator authorization.

A value can be specified for both counters as default. Use the "Reset" button to set this default as counter value.

(6) Service

You can select the following functions in this area:

- "Simulation"
- "Release for maintenance"

Refer to the "Switching operating states and operating modes" section in the APL manual.

You can find information on this area in the APL manual in sections:

- Simulating signals
- Release for maintenance

(7) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

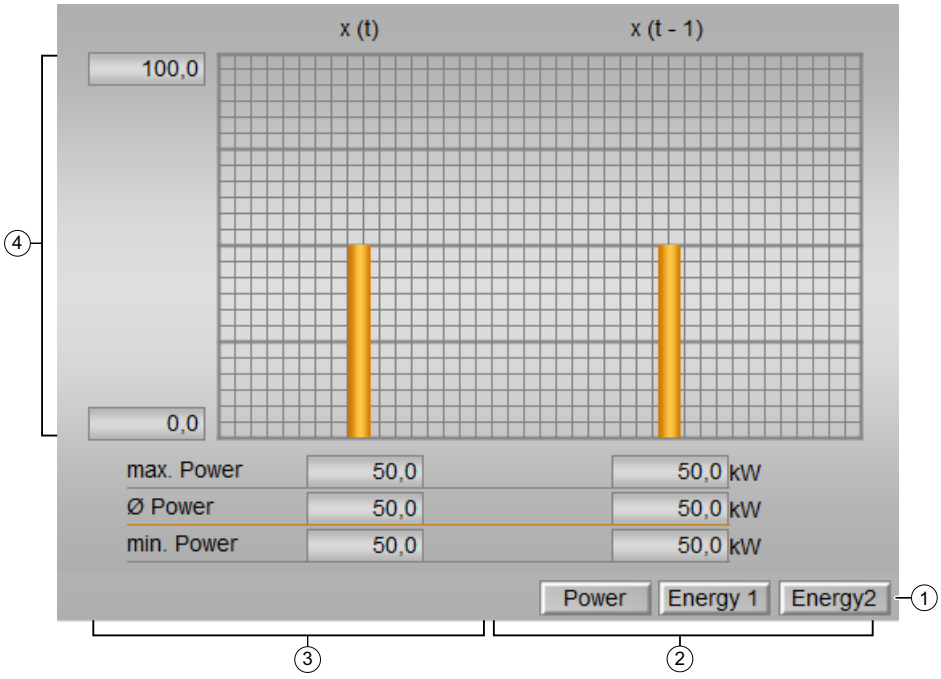
Icons for enabled operations:

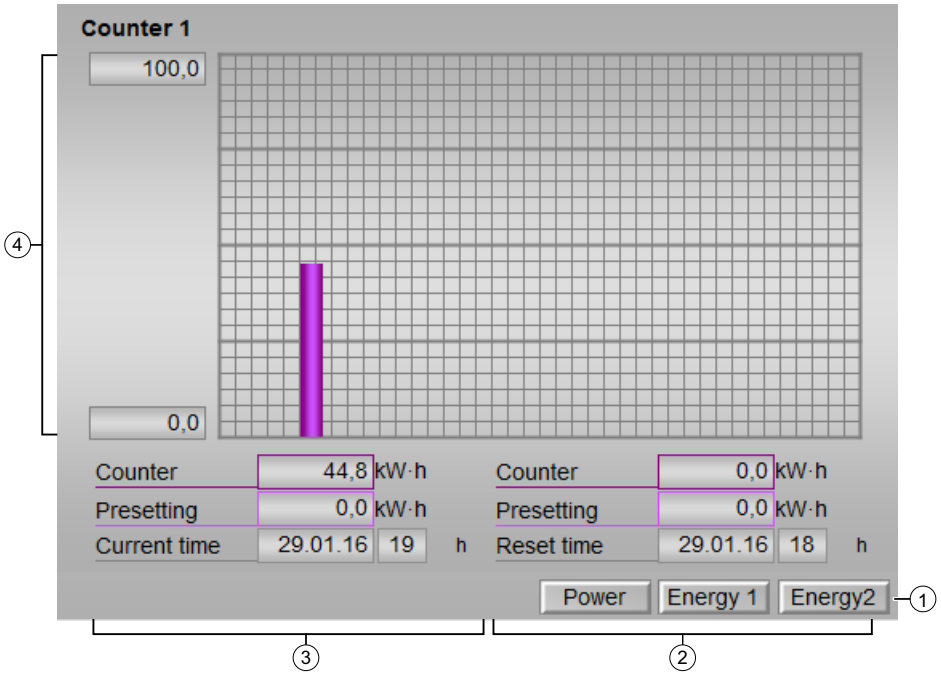
- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: The OS operator cannot control this parameter at this time due to the process
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

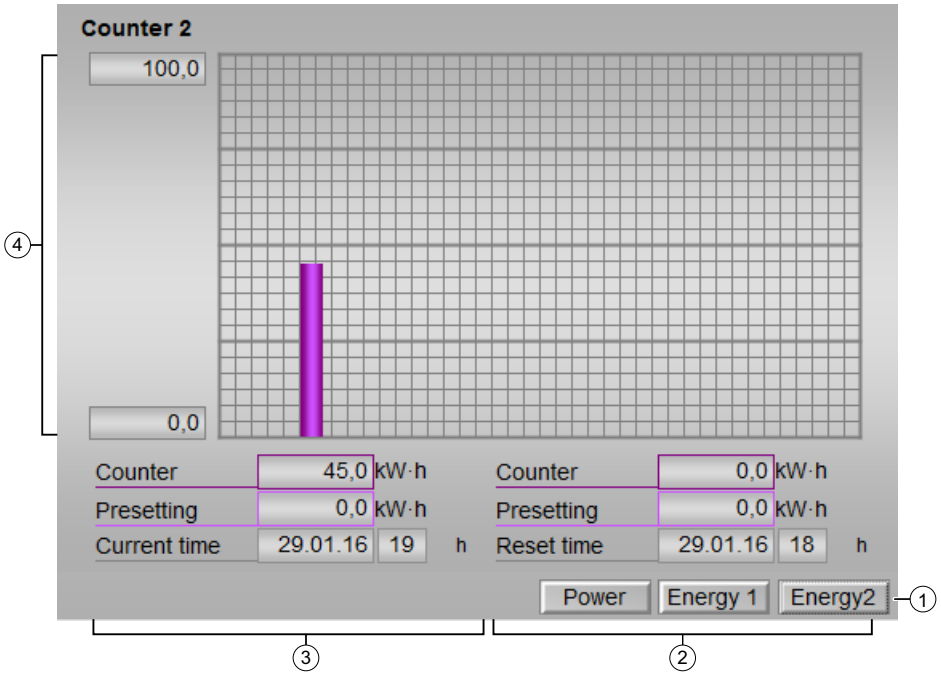
8.3.8.5 CalcPwrL histogram view

CalcPwrL histogram view

The histogram view of CalcPwrL shows the value in the current and preceding interval for the power, the first counter and the second counter.




Counter 1
100,0
0,0
Counter 44,8 kW·h
Presetting 0,0 kW·h
Current time 29.01.16 19 h
Counter 0,0 kW·h
Presetting 0,0 kW·h
Reset time 29.01.16 18 h
Power
Energy 1
Energy2
1
2
3
4


Counter 2
100,0
0,0
Counter 45,0 kW·h
Presetting 0,0 kW·h
Current time 29.01.16 19 h
Counter 0,0 kW·h
Presetting 0,0 kW·h
Reset time 29.01.16 18 h
Power
Energy 1
Energy2
1
2
3
4

(1) Switching the views

The buttons are used to switch the histogram view to the values

- of the power consumption
- of the first counter and
- of the second counter.

(2) Count value at last reset/power value of the last interval with bar graph

Power

The following values are displayed for the preceding time interval:

- Maximum power in the last time interval
- Average power in the last time interval
- Minimum power in the last time interval

The average power is also displayed in a bar graph.

Counter

The following values are displayed for the counter:

- Count value at the time of the last reset
- Default setting for the displayed count value
- Time of the last reset

(3) Current count value/current power value with bar graph

Power

The following values are displayed for the current time interval:

- Maximum power in the current time interval
- Average power in the current time interval
- Minimum power in the current time interval

The average power is also displayed in a bar graph.

Counter

The following values are displayed for the counter:

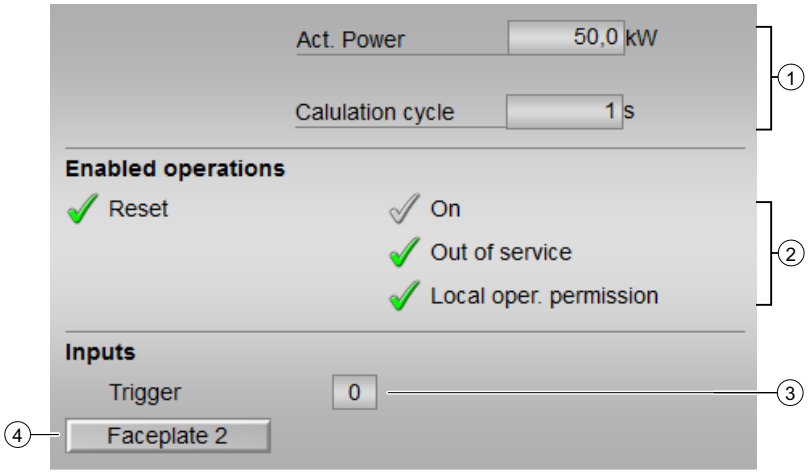
- Count value at the current time
- Default setting for the displayed count value
- Current time

(4) High and low scale range for the process value

These values provide information on the display range for the bar graph of the process value. The scale range is defined in the engineering system.

8.3.8.6 CalcPwrL preview

CalcPwrL preview



(1) Preview area

This area shows you a preview for the following values:

- "Current power": Display of the power in the current calculation cycle
- "Calculation cycle": Current time for the calculation

(2) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- Green check mark: The OS operator can control this parameter
- Gray check mark: The OS operator cannot control this parameter at this time due to the process
- Red cross: The OS operator cannot control this parameter due to the configured AS operator permissions (OS_Perm or OS1Perm).

The following enabled operations are shown here:

- "Reset":
- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to change to the standard view of the OpStations/UserM block. Additional information on this topic is available in the section "Operator control enable" in the APL manual and in the section "Operation of multiple control rooms" in this manual.

(3) Inputs (area is only visible when interval is switched by an external trigger "Feature.Bit 9 = 1")

- Trigger: Status display with signal status of the trigger input

(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

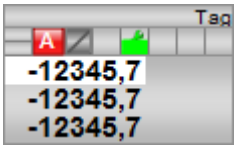
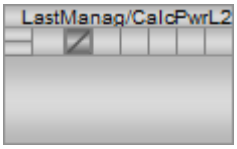
8.3.8.7 CalcPwrL block icons

CalcPwrL block icons

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes
- Signal status, release for maintenance
- Memo display

The block icons from template @TemplateILV8.PDL:

Icons	Selection of the block icon in CFC
	1
	2
	Representation of "Out of Service"

You can find additional information on the block icon and the control options in the block icon in the following sections of the APL manual:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

8.4 PeakMon - Monitoring and forecast of consumption peaks at the plant infeed

8.4.1 PeakMon description

Object name (type + number) and family

Type + number:FB 1190

Family: EngyMgmt

Number of process objects: 20 PO

Area of application for PeakMon

The block is used for monitoring and forecasting consumption values at the plant infeed.

Mode of operation

The block calculates the available power until reaching the set power limit. It also forecasts the expected power consumption, the power difference and the power reserve based on the previous power consumption.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB100).

Startup characteristics

Use the Feature Bit 'Setting the startup characteristics' to define the startup characteristics for this block.

The messages are suppressed after startup for the number of cycles set at `RunUpCyc`.

Status word allocation for `Status1` parameter

You can find a description for each parameter in the section PeakMon I/Os (Page 344).

Status bit	Parameter
0	`Occupied`
1	`BatchEn`
2	`SimOn`
3	`OosAct`.Value
4	`OosLi`.Value
5	`OnAct`.Value
6	Not used

Status bit	Parameter
7	Interval mode: External trigger
8	Interval mode: Fixed time window
9	Not used
10	SimLiOp.Value
11	Delay of the PV_AH_Lim message
12	Delay of the PV_WH_Lim message
13	Delay of the PV_TH_Lim message
14	Not used
15	Interval error
16	Interval changed
17	Collection of message delays
18	PowerFor with custom scale and unit
19	Trigger error
20	Interval adaptation
21-26	Not used
27	SP_ExtAct.Value
28	SP_LoAct.Value
29	SP_HiAct.Value
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word allocation for `Status2` parameter

You can find a description for each parameter in the section PeakMon I/Os (Page 344).

Status bit	Parameter
0	MsgLock
1	PFor_AH_Act.Value
2	PFor_WH_Act.Value
3	PFor_TH_Act.Value
4-6	Not used
7	PFor_AH_En
8	PFor_WH_En
9	PFor_TH_En
10-12	Not used
13	PFor_AH_MsgEn
14	PFor_WH_MsgEn
15	PFor_TH_MsgEn
16-21	Not used
22	GradEn
23-27	Not used
28	0 = Falling measured value / 1 = Rising measured value
29	SP_Out.Value <> SP_OutT0.Value

Status bit	Parameter
30	Not used
31	MS_RelOp

Status word allocation for Status3 parameter

You can find a description for each parameter in the section PeakMon I/Os (Page 344).

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via EventTsIn
1	Effective signal 2 of the message block interconnected via EventTsIn
2	Effective signal 3 of the message block interconnected via EventTsIn
3	Effective signal 4 of the message block interconnected via EventTsIn
4	Effective signal 5 of the message block interconnected via EventTsIn
5	Effective signal 6 of the message block interconnected via EventTsIn
6	Effective signal 7 of the message block interconnected via EventTsIn
7	Effective signal 8 of the message block interconnected via EventTsIn
8 - 23	Effective signal 8 to 16 of the message block connected via Event16TsIn
24-31	Not used

8.4.2 PeakMon operating modes

PeakMon operating modes

The block can be operated using the following modes:

- "On"
- "Out of service"

"On"

You can find general information about the "On" mode in the "On" section of the APL manual.

"Out of service"

You can find general information about the "out of service" mode in the "Out of service" section of the APL manual.

8.4.3 PeakMon functions

PeakMon functions

The functions for this block are listed below.

Time window

The time window defines the time period for the power limit.

The time window can be set externally with a pulse or defined using a parameterizable assignable time (default: 15 minutes for electrical consumption).

Interval characteristics

The intervals can be controlled in two ways:

- Controlled externally by a trigger (e.g. by the power supply company)
- Fixed specified time window

The mode of operation is set in Feature.Bit 9.

PeakMon is the central block for setting the interval. Starting from this block, the time interval is forwarded to the downstream LdMgment8 blocks.

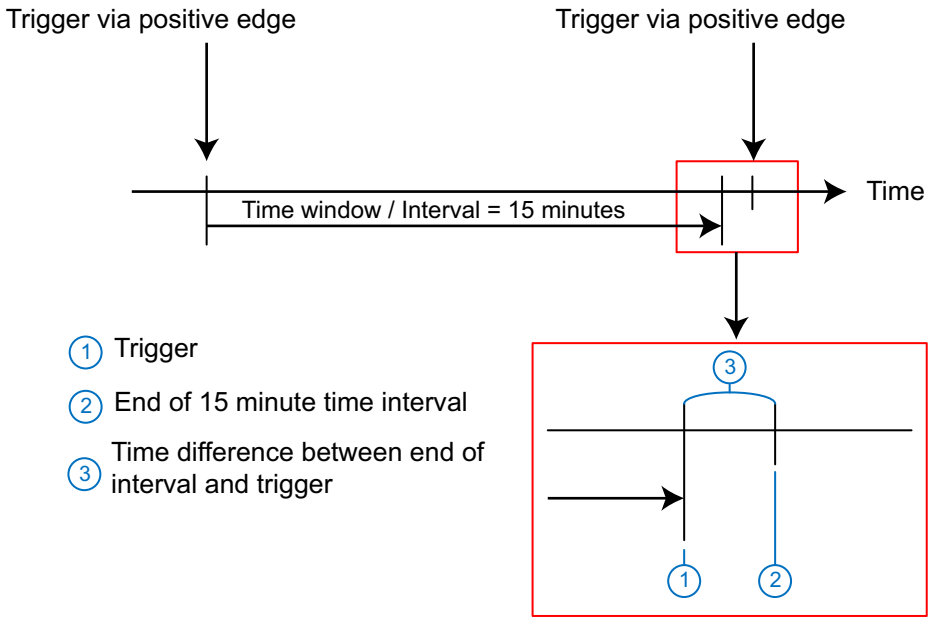
Trigger-controlled (Feature.Bit 9 = 1)

If the time window is controlled externally by a trigger, the expected time window (e.g. 15 minutes) still has to be configured.

If a trigger fails to appear at the end of the interval (1), the block starts a new interval and issues a process control message. The duration of the interval depends on the parameter assignment.

If the trigger arrives delayed (2), the block starts a new interval with the trigger signal.

A higher priority thereby results for the trigger signal than for the configured interval time.



Fixed time window (Feature.Bit 9 = 0)

The duration of the interval is set with a parameter for the fixed time window. The start of the interval aligns with the clock time (CPU time) at the full hour. This behavior increases the plausibility in later evaluations because it is made a lot more difficult by asynchronous behavior.

Example:

You specify an interval of 15 minutes. You enter the information at 12:35 PM. Because the interval start aligns with the time, the block does not start "working" immediately after the parameter assignment but rather at 12:45 PM.

Plausibility

The specified time window is always checked for plausibility. The formulas in the example below are used to check whether the time base of the specified time window is plausible for an hour or a day.

If the value is not plausible, an error is output and a text is displayed in the faceplate in the standard view. The block goes into the error state.

The following settings are permitted and checked for plausibility:

- On hourly basis:
  - Y=24/X
    Examples: Y= 24/1='True' or Y= 24/9='False'
- On minute basis
  - Y=60/X
    Examples: Y= 60/15='True' or Y= 60/13='False'

Supply signal

An analog value of the currently consumed power is expected at the input.

If no analog value is available, you can form the required analog counter value from a digital pulse (for example from the pulse output of a quantity counter) with the PulseCon block.

Calculation functions

The block works with the following values:

Available power (setpoint)

The available power is the parameter that represents the power limit. This value must not be exceeded in the mean value of the time interval.

The value is always in effect for one time window. If the value is changed, the modified value is applied in the faceplate and takes effect in the next interval. The change is also displayed at the bar graph in the faceplate and as text.

Forecast

The forecast indicates the power consumption expected in the current time window for the entire time window based on the previous power consumption.

The parameter is calculated as follows:

$$ForeCast = Sum\ of\ the\ current\ power\ consumption\ of\ the\ elapsed\ time + \left(\frac{Current\ power}{Intervall\ time} * (Intervall\ time - Elapsed\ time)\right)$$

$$Sum\ of\ the\ current\ power\ consumption\ of\ the\ elapsed\ time = Sum\ of\ the\ current\ power\ consumption\ of\ the\ elapsed\ time\ (n^{-1}) + \frac{Current\ power}{Intervall\ time}$$

Here, the forecast corresponds to the currently measured value under the assumption that it is constant.

Power difference

The power difference is the difference between forecast and available power.

The parameter is calculated as follows:

$$Power\ Diff. = Setpoint - ForeCast$$

Because the power difference increases with increasing interval duration (to infinity), it must be possible to set a limit (parameter) for this.

Power reserve

The power reserve indicates how much higher the power still may be to stay within the available power limit.

The parameter is calculated as follows:

$$Power\ Rest = \frac{Avial.\,Power - Sum\ of\ the\ current\ power\ consumption\ of\ the\ elapsed\ time}{Remaining\ Time} * Intervall\ time$$

Time Basis

The two dependent values of the block are standardized to seconds in the calculations. This means an hour is represented as 3600 seconds and 15 minutes as 900 seconds.

Calling additional blocks

This block includes the standard function Call additional faceplates. Information is available in the section *Calling additional faceplates* in the *APL Function Manual*.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator authorizations* in the *APL Function manual*.

The block has the following authorizations for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4	1 = Operator can change the time interval
5	Not used
6	1 = Operator can change the simulation value `SimPV`
7	1 = Operator can reset the maximum values
8	Not used
9	Not used
10	Not used
11	1 = Operator can enable the Simulation function
12	1 = Operator can activate the Release for maintenance function
13	1 = The operator can increase the limit (PowerFor) for the alarm
14	1 = The operator can decrease the limit (PowerFor) for the warning
15	1 = The operator can increase the limit (PowerFor) for the tolerance
16	1 = The operator can change the limit (PowerFor) for the hysteresis
17	1 = Operator can set the trigger monitoring time
18	1 = Operator can activate the Trigger monitoring function
19	1 = Operator can set the flatter monitoring time
20	1 = Operator can activate the Flatter monitoring function
21	Not used
22	Not used
23	1 = Operator can change the DeadBand parameter

Bit	Function
24	1 = The operator can activate/deactivate messages
25	1 = The operator can activate/deactivate messages using `PFor_WH_MsgEn`
26	1 = The operator can activate/deactivate messages using `PFor_TH_MsgEn`
27	1 = Operator can switch to internal
28	1 = Operator can switch to external
29	1 = Operator can perform bumpless switchover
30	1 = Operator can change `SP_Int`
31	Not used

Select unit of measure

This block includes the standard APL function, Select unit of measure.

The default unit is kW (1190), depending on the parameter.

Displaying auxiliary values

This block provides the standard APL function, "Display auxiliary values".

SIMATIC BATCH functionality

This block provides the standard APL function, "SIMATIC BATCH functionality".

Group display SumMsgAct for limit monitoring, CSF and ExtMsgx

The block has the standard APL function, Group display for limit monitoring, CSF and ExtMsgx.

Limit monitoring of the process value

This block has a limit monitoring of the forecast for violation.

Activation and deactivation of messages

You can use the XX_MsgEn I/Os to individually enable or disable the pending messages of the limits.

All messages are activated by default.

To deactivate messages for the high alarm, for example, you set the associated AH_MsgEn = 0 I/O.

You can deactivate all messages via the MsgLock = 1 I/O.

Limit monitoring with hysteresis

This block includes the standard APL function, Limit monitoring with hysteresis. It is performed via the PV_Hyst input parameter.

Generating instance-specific messages

This block includes the standard APL function, "Generating instance-specific messages".

Simulating signals

This block features the standard APL function, "Simulate signals".

If simulating is active, the currently measured power consumption (PV) of SimPV and SimPV_Li I/O is applied.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
9	1 = External trigger / 0 = Fixed time window
16	Forecast with separate display area and custom unit
19	See APL function: "Changes signal status of outputs in OOS"
22	Update acknowledgment and error status of the message call
23	See APL function: "Update acknowledgment and error status of the message call" See APL function: "Limits disabled in faceplate"
25	See APL function: "Enable local operating permission"
26	See APL function: "Suppression of all messages"
28	See APL function: "Deactivate switching points"
29	See APL function: "Signal limit violation"

Suppressing messages using the `MsgLock` parameter

This block includes the standard APL function, "Suppress messages using the MsgLock parameter".

Release for maintenance

This block has the standard APL function, "Release for maintenance".

External/internal setpoint specification

The block provides the standard APL function, Setpoint specification - internal/external.

Flutter suppression

The block provides the APL standard function Flutter suppression.

8.4.4 PeakMon error handling

For troubleshooting all blocks, refer to the "Troubleshooting" section of the APL documentation in the basic information.

The following errors can be displayed for this block:

- Error numbers
- Process control messages

Overview of error numbers

The ErrorNum I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when inserting the block; the block is not processed.
0	There is no error.
30	The value `PV` can no longer be represented in the REAL numerical field.
43	TimeFactor < 0 or > 2
44	Invalid `SampleTime`: • `CalcTime` <= `SampleTime` • `IdleTime` <= `SampleTime` • `TimeSlot` <= `SampleTime`
45	Invalid `TimeSlot`: • `CalcTime` >= `TimeSlot` • `IdleTime` >= `TimeSlot` • 60 > `TimeSlot`
51	Invalid signal for `SP_LiOp` = 1: • `SP_ExtLi` = 1 and `SP_IntLi` = 1

External control system fault (CSF)

This block has the standard function, External control system fault, and gives you the option to connect an external control system fault with the `CSF` parameter.

8.4.5 PeakMon messaging

Messaging

The following messages can be generated for this block:

- Control system fault
- Process messages
- Instance-specific messages

Control system fault

You can interconnect an external fault (signal) to input parameter CSF. If this signal changes to CSF = 1, a control system fault is triggered (MsgEvId1, SIG 6).

Process messages

Message instance	Message identifier	Message class	Event
MsgEvID1	SIG 1	Alarm - high	$$BlockComment$$ PowerFor - High alarm limit violated
	SIG 2	Warning - high	$$BlockComment$$ PowerFor - High warning level violated
	SIG 3	Tolerance - high	$$BlockComment$$ PowerFor - High tolerance limit violated
	SIG 4	AS control system message - fault	$$BlockComment$$ Trigger flutters
	SIG 5	AS control system message - fault	$$BlockComment$$ Trigger not received
	SIG 6	AS control system message - fault	$$BlockComment$$ External fault has occurred

Explanation:

$$BlockComment$$: content of the instance-specific comment

You have the option of connecting an external fault (signal) to input parameter CSF. If this CSF = 1, a control system error is triggered (MsgEvId1, SIG 6).

Instance-specific messages

You have the option of using up to two instance-specific messages for this block.

Message instance	Message identifier	Message class	Event
MsgEvID1	SIG 7	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 8	AS process control message - fault	$$BlockComment$$ External message 2

Explanation:

$$BlockComment$$: content of the instance-specific comment

Associated values for message instance MsgEvId1

Associated value	Block parameters
1	BatchName
2	StepNo

Associated value	Block parameters
3	BatchID
4	ExtVal04
5	ExtVal05
6	ExtVal06
7	ExtVal07
8	ExtVal08
9	Reserved
10	Reserved

8.4.6 PeakMon I/Os

Input parameters

Parameter	Description	Data type	Default
CalcTime	Sample Time for the Power Forecast calculation [s]	DINT	1
GradAct	Gradient active	BOOL	1
IdleTime	Idle time new interval [s]	REAL	5.0
InTrig	Linkable input for external Trigger	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
LagTime	Lag Time Constant [s]	REAL	1.0
MonitorFlut	1=Monitor ON, 0=Monitor OFF: Fluttering monitor	BOOL	1
MonitorTrig	1=Monitor ON, 0=Monitor OFF: Trigger monitor	BOOL	1
MonTiFlut*	Monitoring time for the Fluttering [s]	REAL	3.0
MonTiTrig*	Monitoring time for the Trigger [s]	REAL	3.0
PFor_AH_DC	PowerFor - Alarm Delay Time Coming High [s]	REAL	0.0
PFor_AH_DG	PowerFor - Alarm Delay Time Going High [s]	REAL	0.0
PFor_AH_En	PowerFor - High Alarm Enable	BOOL	1
PFor_AH_Lim*	PowerFor - High Alarm Limit	REAL	95.0
PFor_AH_MsgEn*	Power Forecast Value - High Alarm Message Enable	BOOL	1
PFor_Hyst*	PowerFor - Alarm Hysteresis	REAL	1.0
PFor_OpScale*	PowerFor - Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 0
PFor_TH_DC	PowerFor - Toleranz Delay Time Coming High [s]	REAL	0.0

Parameter	Description	Data type	Default
PFor_TH_DG	PowerFor - Toleranz Delay Time Going High [s]	REAL	0.0
PFor_TH_En	PowerFor - High Toleranz Enable	BOOL	1
PFor_TH_Lim*	PowerFor - High Tolerance Limit	REAL	85.0
PFor_TH_MsgEn*	PowerFor - High Toleranz Message Enable	BOOL	1
PFor_Unit*	Unit of Power Forecast Value	INT	1190
PFor_WH_DC	PowerFor - Warning Delay Time Coming High [s]	REAL	0.0
PFor_WH_DG	PowerFor - Warning Delay Time Going High [s]	REAL	0.0
PFor_WH_En	PowerFor - High Warning Enable	BOOL	1
PFor_WH_Lim*	PowerFor - High Warning Limit	REAL	90.0
PFor_WH_MsgEn*	PowerFor - High Warning Message Enable	BOOL	1
SP_OpScale*	SP- Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 0
SP_Unit*	Engineering units of Input	INT	1190
TimeSlot*	Operator input for setting time	DINT	900

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
BlkLdMgmt	Connection to LdMgmt08 Block	STRUCT	-
BlkLdMgmt.BlockID	BlockID PeakMon	INT	0
BlkLdMgmt.CalcTime	Sample Time for the Power Forecast calculation [s]	DINT	0
BlkLdMgmt.PFor_UnitOut	Power Forecast unit	INT	1190
BlkLdMgmt.PowerDif	Power Difference	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkLdMgmt.PowerFor	Power Forecast	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkLdMgmt.PowerRes	Power Rest	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
BlkLdMgmt.PV	Process Value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkLdMgmt.PV_UnitOut	Process Value unit	INT	1190
BlkLdMgmt.SP	Setpoint	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
BlkLdMgmt.SP_UnitOut	Setpoint unit	INT	1190
BlkLdMgmt.Status1	Status1 Word	DWORD	16#00000000
BlkLdMgmt.Time_Out	Current time [s]	DINT	0
BlkLdMgmt.TimeSlot	Interval Time [s]	DINT	0
CalcTime_Out	Sample Time for the power Forecast calculation [s]	DINT	0
FlutAct	Fluttering Switch Active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PFor_AH_Act	PowerFor - Alarm High Active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PFor_AH_Out	PowerFor - High Alarm Limit Output	REAL	0.0
PFor_HysOut	PowerFor - Alarm Hysteresis Output	REAL	0.0
PFor_OpScaleOut	PowerFor - Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 100
PFor_TH_Act	PowerFor - Tolerance High Active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PFor_TH_Out	PowerFor - High Tolerance Limit Output	REAL	0.0
PFor_UnitOut	PowerFor unit	INT	1190
PFor_WH_Act	PowerFor - Warning High Active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PFor_WH_Out	PowerFor - High Warning Limit Output	REAL	0.0
PowerDif	Power Difference Value (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PowerFor	Power Forecast Value (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
PowerRes	Power Rest Value (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_AvT0 ... PV_AvT5	Process Value Average Interval 1 ... 5 (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_Grad	PV - Gradient Value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_GradNP	PV - Gradient Value minimum Peak Value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_GradPP	PV - Gradient Value maximum Peak Value	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_MaxT0 ... PV_MaxT5	Process Value Max. Interval 1 ... 5 (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_MinT0 ... PV_MinT5	Process Value Min. Interval 1 ... 5 (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_OpScaleOut	PV - Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 0
PV_Out	Process Value (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
PV_UnitOut	PV unit	INT	1190
SP_Bad	Bad status of SP_Out (not 16#80 and not 16#60)	BOOL	0
SP_ExtAct	1: External SP is active, 0: Internal SP is active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
SP_HiAct	Input High Limit Active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
SP_LoAct	Input Low Limit Active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
SP_OpScaleOut	SP- Bar Display Limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 100 • 0
SP_Out	Setpoint Value (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
SP_OutT0 ... SP_OutT5	Setpoint Interval 0 (Analog Output)	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
SP_UnitOut	SP unit	INT	1190
TdAs	AS time of the day	DINT	0
Time_Out	Current time [s]	DINT	0
TimeSlot_Bad	TimeSlot Error	BOOL	0
TimeSlot_Out	Current time slot [s]	DINT	0

8.4.7 PeakMon block diagram

A block diagram is not provided for this block.

8.4.8 Operator control and monitoring

8.4.8.1 PeakMon views

PeakMon views

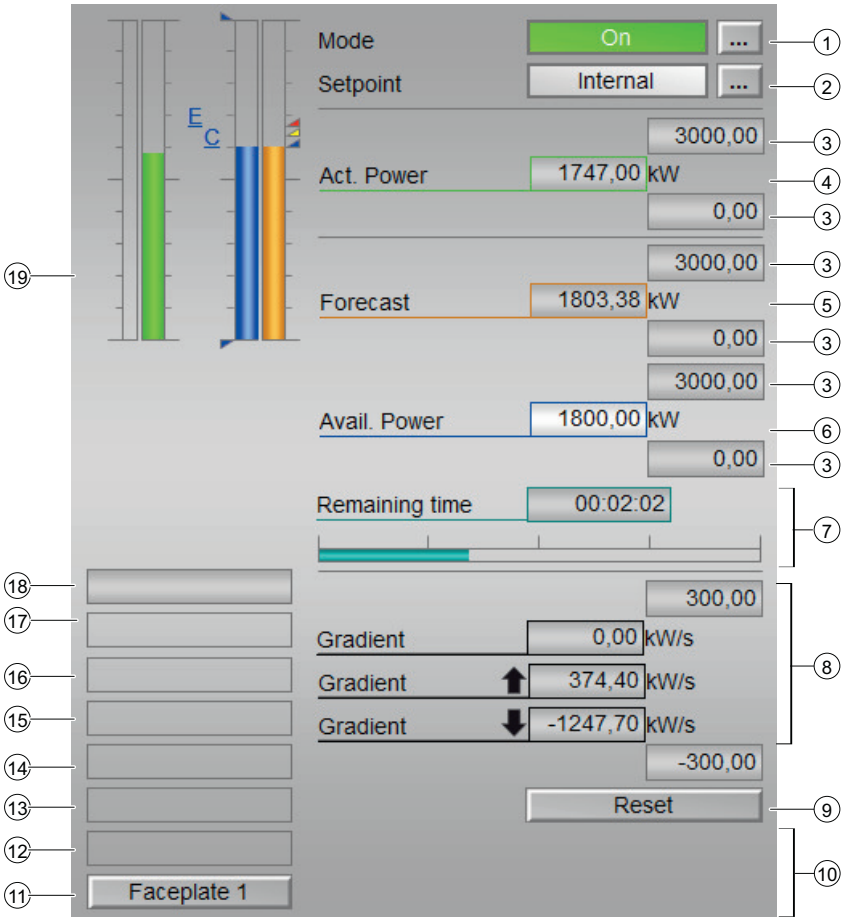
The PeakMon block has the following views:

- PeakMon standard view (Page 349)
- Message view
- PeakMon limit view (Page 354)
- Trend view
- PeakMon parameter view (Page 355)
- PeakMon histogram view (Page 357)
- PeakMon preview (Page 358)
- Memo view
- Batch view
- PeakMon block icons (Page 359)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

8.4.8.2 PeakMon standard view

PeakMon standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

For more information on changing the operating mode, refer to the "Switching operating states and operating modes" section in the APL manual.

(2) Display and switchover of the setpoint specification

This area shows how to specify the setpoint. The setpoint can be specified as follows:

- By the application ("External", CFC/SFC)
- By the user directly in the faceplate ("Internal").

For switchover of the setpoint specification, refer to the section "Switchover of operating state and operating mode" in the APL manual.

(3) High and low scale range for the process value

These values provide information on the display range for the bar graph of the process value. The scale range is defined in the engineering system.

(4) Display of the current power including signal status

This area shows the current consumption value with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the `PV_Out` parameter:

Changing the display of the text for "Current power"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the parameter (`PV_Out`) to what you want to later see in runtime.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding `PV_Out` input.

(5) Display of the forecast including signal status

This area shows the forecast of the current energy consumption value with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the `PowerFor` parameter:

Changing the display of the text for "Forecast"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "Identifier" column, change the default setting for the input parameter (`PowerFor`) to what you later want to see in runtime.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding PowerFor input.

(6) Display of the available power including signal status

This area shows the still available power in the current time window with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection.

For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the `SP_Out` parameter:

Changing the display of the text for "Forecast"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "Identifier" column, change the default setting for the input parameter (`SP_Out`) to what you later want to see in runtime.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding PowerFor input.

(7) Time window

Display of the remaining time for the active time window.

(8) Display of the gradients with high and low scaling range of the gradient value

Display of the gradient

These values provide information on the display range for the bar graph of the gradient. The scale range corresponds to 10% of the scale range for the process value: For example, once you have specified a process value scale range of 0 to 100, the scale range of the gradient will be automatically set to a value between -10 and 10.

The current gradient value is displayed when one of the following monitoring functions is activated:

- Gradient monitoring for positive changes `(GradHUpEn = 1)`
- Gradient monitoring for negative changes `(GradHDnEn = 1)`
- Gradient monitoring `(GradLEn = 1)`

High and low scale range for the gradient value

This area shows the current, minimum and maximum gradient value and the rise and fall of the value. This display of the minimum and maximum gradient value functions like a min/max pointer.

The current gradient value is displayed when one of the following monitoring functions is activated:

- Gradient monitoring for positive changes `(GradHUpEn = 1)`
- Gradient monitoring for negative changes `(GradHDnEn = 1)`
- Gradient monitoring `(GradLEn = 1)`

The maximum peak gradient value is displayed when the gradient monitoring is activated for positive changes `(GradHUpEn = 1)`

The minimum peak gradient value is displayed when the gradient monitoring is activated for negative changes `(GradHDnEn = 1)`

(9) Reset the peak values of the gradient

You can use this button to reset the maximum or minimum peak value of the gradient (`PV_GradPP` and `PV_GradNP` output parameters).

The button is displayed when gradient monitoring is activated for positive `(GradHUpEn = 1)` or negative changes `(GradHDnEn = 1)`.

(10) Display of auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system. You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(11) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(12) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

(13) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

(14) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

(15) Display area for block states

This area provides additional information on the operating state of the block:

- "Delay time"
- "CalcTime changed"

(16) Display area for block states

This area provides additional information on the operating state of the block:

- "Setpoint changes"

(17) Display area for block states

This area provides additional information on the operating state of the block:

- "Trigger error"

(18) Display area for block states

This area provides additional information on the operating state of the block:

- "Interval changed"
- "Interval adaptation"
- "Interval error"

(19) bar graph for the process values, display of the gradients

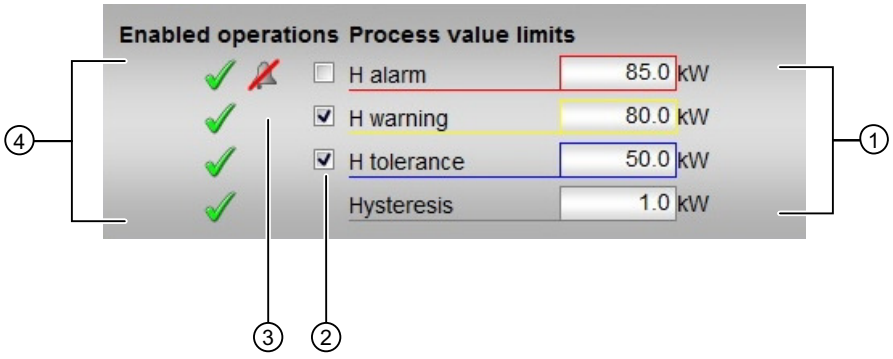
This area shows the current values of the forecast, the available power, the current power and the gradient in the form of a bar graph.

The small colored triangles indicate the specified limits in the respective bar graph.

The visible area in the bar graph depends on the configuration in the engineering system (ES).

8.4.8.3 PeakMon limit view

Limit view of PeakMon



(1) Process value limits

In this area, you can enter the limits for the process value. Refer to the "Changing values" section in the APL manual.

You can change the following limits:

- "H alarm": Alarm high
- "H warning": Warning high
- "H tolerance": Tolerance high
- "Hysteresis"

(2) Suppress messages

You can enable/disable messages by setting the check mark.

(3) Message suppression/delay

Message suppression indicates whether the suppression of the associated message in the AS block is activated with the `xx_MsgEn` parameter. The output of messages is not suppressed when the block is installed (all `xx_MsgEn` parameters are preset to 1). Messages can only be output if limit monitoring of the additional analog value has been enabled.

Alarm delays are also displayed in this position; for more on this, see section Area of application of the alarm delays in the APL documentation.

(4) "Enable operations"

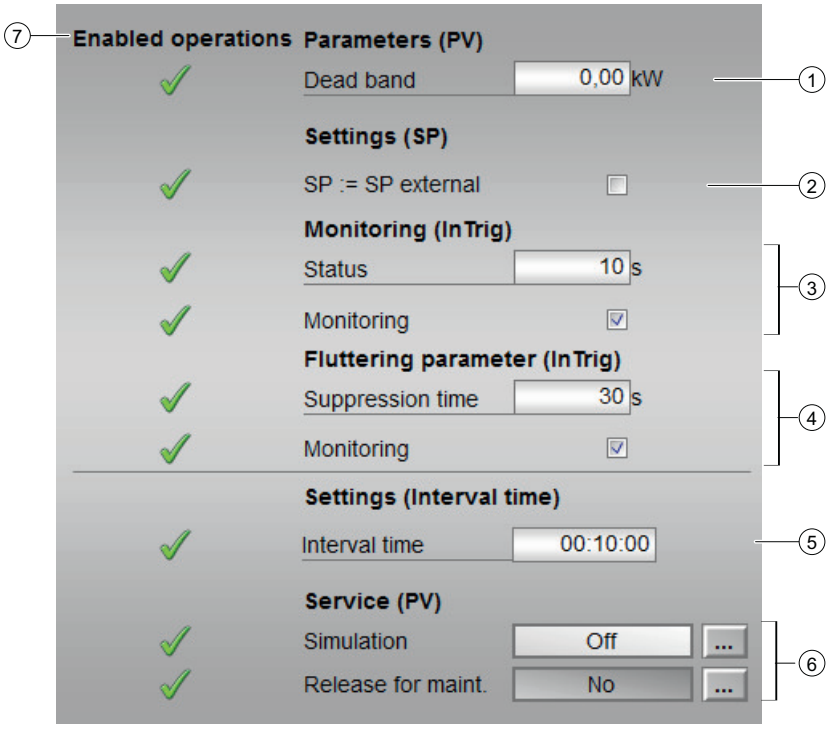
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

8.4.8.4 PeakMon parameter view

PeakMon parameter view



(1) Parameters

You can change the following parameters in this area:

- "Deadband"

You can find additional information on this in the section "Changing values" in the manual of the APL.

(2) SP setting

- "SP := SP extern": ☑ Bumpless switchover of the setpoint for setpoint switchover from "external" to "internal". The internal setpoint is tracked to the external one.

(3) Monitoring (area is only visible when interval is switched by an external trigger "Feature.Bit 9 = 1")

In this area, you change parameters and thereby influence the block:

- "Status": Monitoring time during runtime.
- Activate "Monitoring"

(4) Flutter parameter (area is only visible when interval is switched by an external trigger "Feature.Bit 9 = 1")

In this area, you change parameters and thereby influence the block:

- Suppression time: Enter the time period during which signal flutter is suppressed on the trigger signal.
- Activate "Monitoring"

(5) Setting

The current setting for the time window is displayed in this area.

(6) Service

You can select the following functions in this area:

- "Simulation"
- "Release for maintenance"

Refer to the "Switching operating states and operating modes" section in the APL manual.

You can find information on this area in the section:

- "Simulate signals"
- "Release for maintenance"

in the APL manual.

(7) Enabled operations

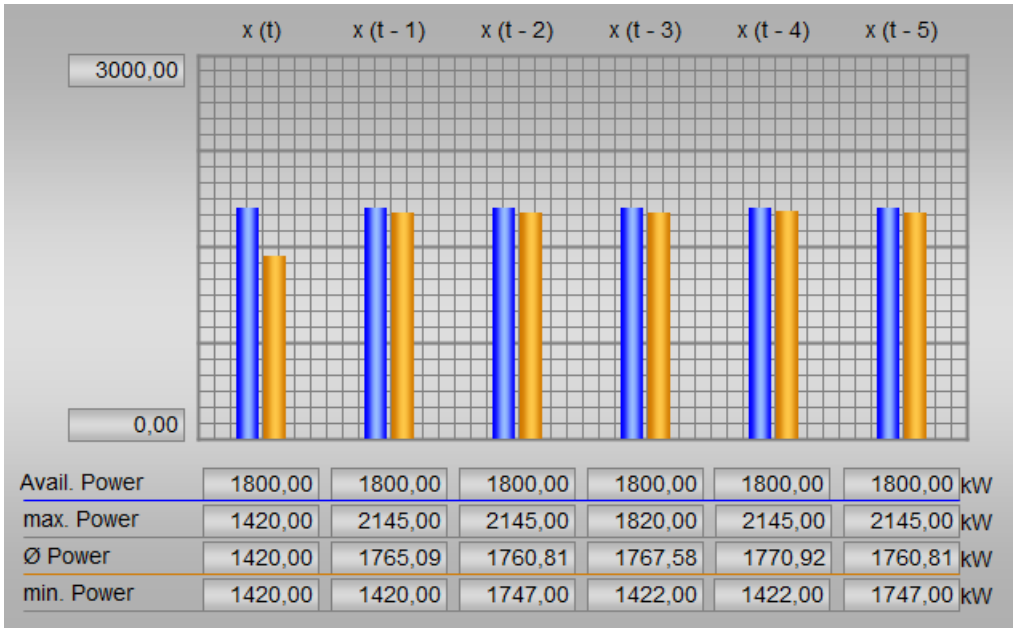
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

8.4.8.5 PeakMon histogram view

PeakMon histogram view



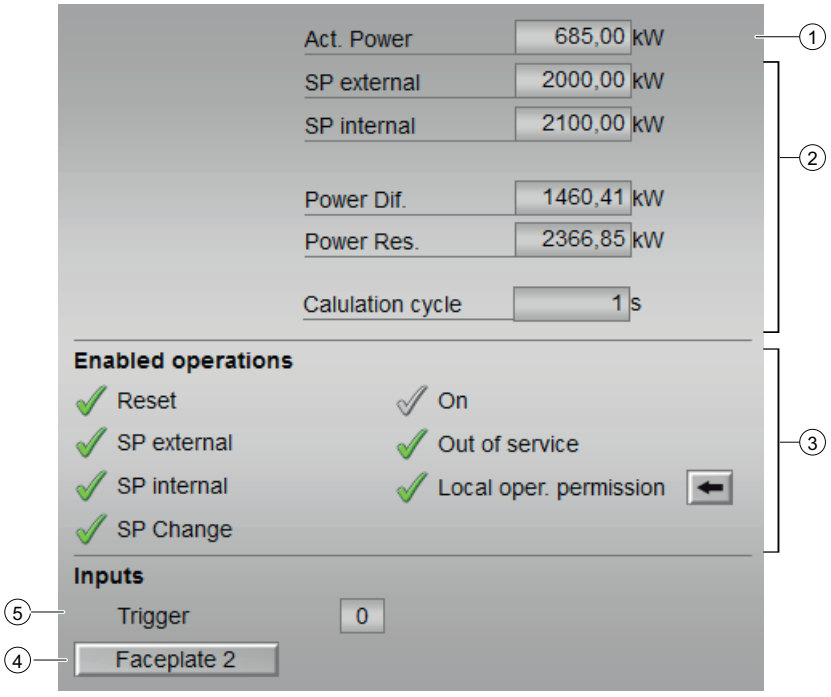
The following values are displayed in the histogram view:

- Average power per time interval
- Maximum power per time interval
- Minimum power per time interval

These three values are displayed for the last 5 time intervals.

8.4.8.6 PeakMon preview

PeakMon preview



(1) Process values

This area displays the real process value (PV).

(2) Preview area

This area shows you a preview for the following values:

- "SP external": currently applicable external setpoint
- "SP internal": currently applicable internal setpoint
- Power difference
- Power reserve
- Calculation cycle

(3) Operator control enable

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: the OS operator cannot control this parameter due to the configured AS operator permissions (OS_Perm or OS1Perm).

The following enabled operations are shown here:

- "Reset": You can reset the peak value of the gradient.
- "SP external": You can use the external setpoint
- "SP internal": You can use the internal setpoint
- "Change SP": You can change the setpoint
- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the <-- button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

(5) Display current control signals

This area shows the most important parameters for this block with the current control:

- "Trigger": Status display with signal status of the trigger input
  This display is only visible if the interval switching is activated by external trigger (Feature.Bit 9 = 1 ").

8.4.8.7 PeakMon block icons

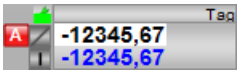
PeakMon block icons

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes
- Internal and external setpoint specification

- Signal status, release for maintenance
- Memo display

The block icons from template @TemplateILV8.PDL:

Icons	Selection of the block icon in CFC
	1
	2

You can find additional information on the block icon and the control options in the block icon in the following sections of the APL manual:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

8.5 PulseCon - Pulse converter

8.5.1 PulseCon description

Object name (type + number) and family

Type + number: FB 1192

Family: EngyMgmt

Number of process objects: 1 PO

Area of application of PulseCon

The block is used for processing of the pulse output of a quantity counter.

Mode of operation

The block uses pulses to calculate the quantity for a defined time period (e.g. 1 pulse = 1m³ or 1 pulse = 0.5 kWh.)

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). The block is also installed automatically in the startup OB (OB100).

NOTICE
Special features when inserting into the processing Because, depending on the application, this block must process pulses with very high frequencies, it is important to observe the correct sequence and processing speed of block and process image partition of the associated signal when installing this block. If these factors are not taken into consideration, pulses can be lost during counting.

8.5.2 PulseCon operating modes

The block has no operating modes.

8.5.3 PulseCon functions

Pulse input

The pulse is defined, for example, by a binary input module at the `Pulse` block input.

Because the pulses are generated in quick succession and can only be pending for a short time period, this block must be called in a fast cycle OB. Because the block is very compact, it can be called in a cycle OB with a very fast processing cycle. (e.g. 20ms)

NOTICE
The connected input module must be processed in the associated process image partition before the block so that the pulse signal is always current.

The value of a pulse is specified with the `PulseValue` input. (e.g., 1 pulse = 1 m^3 or 1 pulse = 1 kWh, etc.).

Calculation / count value formation

Instantaneous value formation:

The block forms an instantaneous value based on the pulses at the `Pulse` input and the parameters `CalcTime` and `PulseValue` and outputs this value at the `PV_Out` output.

The value is calculated with the following formula:

$$PV_Out = \frac{3600}{CalcTime} * PulseValue * PulseCounter$$

The parameter `CalcTiOffset` specifies how long the calculation is executed if no pulse is registered. If no pulse is registered after the time `CalcTiOffset` + `CalcTime`, the calculation is not made until the next pulse.

Count value/consumption value formation per period:

The short-term counter `CountPeriod` is reset at the start of each period. A period is started by a positive edge at the `Reset` input. For the count value to be reliably transferred to the downstream block in a slower cycle, you can delay the reset of the short-term counter with the parameter `DelayTime`.

The `CountReset` output shows the count value of the previous period.

Because this input is a real value, the value can be represented with a high accuracy using decimal places. (e.g. 3.45 kWh).

Note

Especially for smaller pulse values (e.g. `PulseValue` = 0.003), the value can only display a maximum value of about 100,000 due to the accuracy of REAL values.

Count value/consumption value as long-term value:

The block also contains the `CountTotal` output which is a long-term counter that registers each count pulse continuously and **completely**.

This output is designed in DINT format (Double-Integer) and, therefore, cannot reach decimal places but offer significantly higher counter readings. (max. ca. 2,000,000,000).

To prevent counting errors by exceeding the value range of a DINT, the I/O is reset to zero and continues counting from zero when the maximum value set at the `CountMax` I/O (also of the DINT type) is reached.

Providing count value

The count value of the block can be provided with the value configured at the block input `CountValue` by a positive edge at the block input `CountSet`.

8.5.4 PulseCon error handling

For troubleshooting all blocks, refer to the "Troubleshooting" section of the APL documentation in the basic information.

The following errors can be displayed for this block:

- Error numbers

Overview of error numbers

The `ErrorNum` I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
76	Invalid `SamleTime`: • `CalcTime` <= `SampleTime`
77	Invalid `CalcTime`: • `CalcTime` < 1 • `CalcTime` >= 3600 • 3600/`CalcTime` = ..,1515

8.5.5 PulseCon messaging

No message behavior is applicable to this block.

8.5.6 PulseCon I/Os

Input parameters

Parameter	Description	Data type	Default
CalcTime	CalcTime [s]	REAL	10.0
CalcTiOffset	CalcTime Offset [s]	REAL	10.0
CountMax	max. Counter Value (Counter starts with 0)	DINT	1000000000
CountSet	Set Counter to CountValue	BOOL	0
CountValue	Value for Counter	DINT	0
DelayTime	Delay time CountPeriod (Reset)	REAL	1.0
Pulse	Pulse Input	STRUCT • VALUE: BOOL • ST: BYTE	- • 0 • 16#80
PulseValue	Value per Pulse	REAL	1.0
Reset	Reset the calculation of the energy	STRUCT • VALUE: BOOL • ST: BYTE	- • 0 • 16#80
SampleTime	Sampling Time [s]	REAL	0.1

Output parameters

Parameter	Description	Data type	Default
CountPeriod	Engery Counter (current period)	REAL	0.0
CountReset	Engery Counter (after Reset)	REAL	0.0
CountTotal	Engery Counter (total)	DINT	0
ErrorNum	Error Number	INT	-1
PV_Out	Analog Value (current Value)	STRUCT • VALUE: REAL • ST: BYTE	- • 0.0 • 16#80

8.5.7 PulseCon block diagram

The block has no block diagram.

8.5.8 Operator control and monitoring of PulseCon

The block has no views.

Function-based control blocks

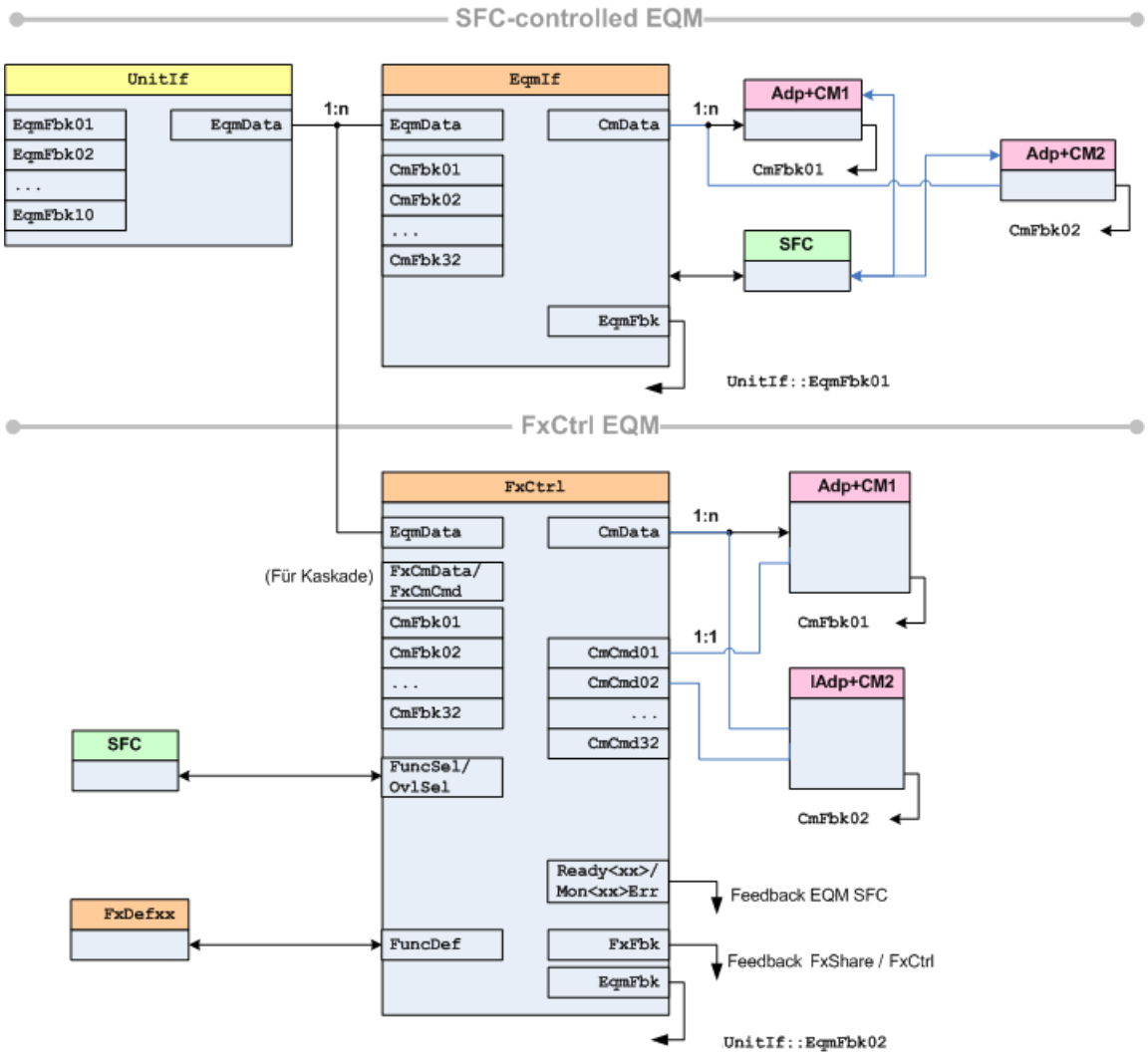
9

9.1 FxCtrl – Function-based control and monitoring block

Area of application and operating principle

This block offers the possibility of a function-based (also step-based) control and monitoring of control modules and signals. In combination with the EQM SFC of the equipment module up to 32 process tags can be controlled through up to 64 programmable function definitions. Programming is carried out through an own online-compatible ES configuration dialog in which the function definitions are displayed in a max. of 64 columns and the process tags or elements in a max. of 32 lines. This means that a simpler and faster change of the controls as well as of the monitoring functions is achieved in comparison to a purely SFC-based solution.

The following figure shows the control schemes both for the "classical" SFC control and for the "extended function-based" control of the control modules of an EQM:



From the point of view of the Industry Library the `FxCtrl` takes up the position of a `EqmIf` block. At a SFC-controlled EQM, controlling of the individual CMs is made directly from the SFC step programming, whereas if the FxCtrl is used, the SFC only provides a function number. This is implemented via the function definitions (control and monitoring masks) configured at the block in CM controls on the one hand and and in step enabling conditions for the SFC on the other hand.

Data flows

The interconnectable inputs and outputs of the FxCtrl can be structured functionally as follows:

- `FuncDef` → Interface for programming the standard and overlay functions. Function definitions that are created with the ES configuration dialog and stored at the block FxDef<xx> are transferred to the FxCtrl through this interface.
- `EqmData / EqmFbk` → Interface to the higher-level level (unit). Here the SIMATIC Batch data, operating modes, resets, simulation operation, enables / locks are exchanged with the FxCtrl .

- `FuncSel / OvlSel / Ready<xx>` → Interface to the controlling SFC instance of the equipment module. Here the function selection is carried out through the SFC instance as well as the feedback of the FxCtrl .
- `CmData / CmCmd<xx>` → Interface to the lower-level elements. This interface is used to activate the selected functions (standard or overlay) and carry out their actions or queries. In addition, the FxCtrl transfers the settings and states between its higher-level level Unit and its lower-level level Process tag.
- `FxFbk / FxCmData / FxCmCmd` → Interface for the master / slave operation. Data exchange between cascading FxCtrl blocks as well as between FxShare and FxCtrl (see the section "FxShare – Function definition block for FxCtrl (Page 412)").

Configuration

- Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).
- The FxCtrl block is instantiated preferably in the EQM-specific CFC together with the SFC type. A block of the type FxDef16 / 32 / 64 is additionally required to store the function definitions (see section "Description of 'FxDef16 / 32 / 64' (Page 410)").
- In addition the interconnection of the control modules and process tags (general "elements) is carried out in the CFC Editor.
- The functions to be used are first defined freely as enumerations in the PCS 7 project and configured at the `FuncSel` or `OvlSel` inputs.
- Next the programming of the function definitions is carried out in a specific configuration dialog that is installed with the PCS 7 Industry Library on the ES (see the section "Configuration of the function definitions (Page 385)"). During the storage all the information at the block parameters in the CFC is stored and also written, if desired, online to the AS.

Interconnection UnitIf - FxCtrl

A 1:n data structure connection `EqmData` with the following elements is required:

Structure parameters

Structure element	Description	Type
`dwUnitID`	UnitID	`DWORD`
`xUnitOcc`	Unit is allocated	`BOOL`
`xUnitOccPulse`	Unit uses pulse	`BOOL`
`xEqmLock`	Lock Equipment Modul	`BOOL`
`xOnOccAuto`	Switch to automatic mode when unit is allocated	`BOOL`
`xManOpEn`	Enable manual mode for operator	`BOOL`
`xCMSimEn`	Enable CM simulation	`BOOL`
`xUnitSimOn`	Unit simulation is active	`BOOL`
`xResetPulse`	Reset Impuls	`BOOL`
`xSwitchToManual`	Manual mode command	`BOOL`

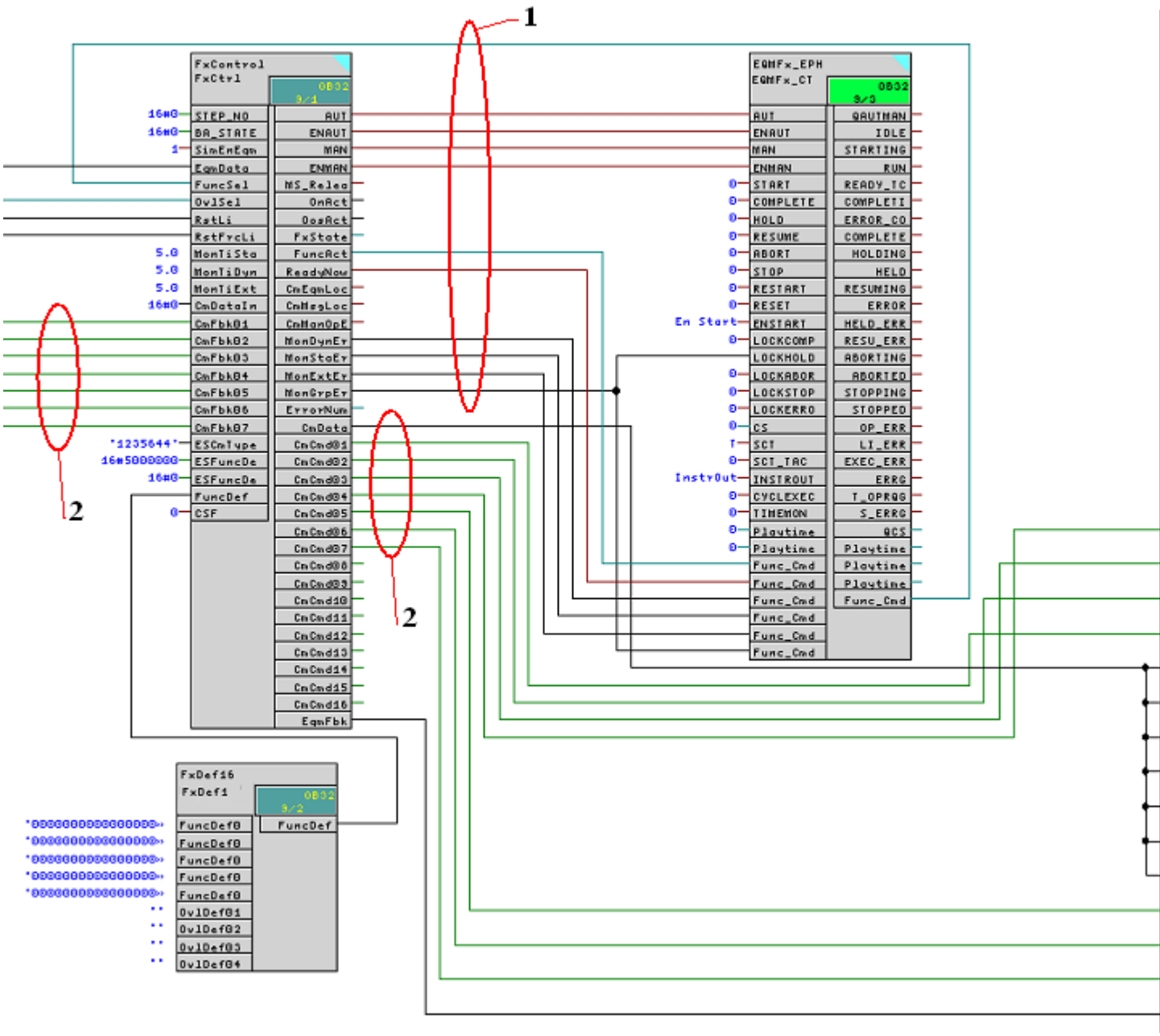
Structure element	Description	Type
xSwitchToAuto	Auto mode command	BOOL
xBatchEnabled	Batch is enabled	BOOL
xOccupyEqm	Equipment module occupies request	BOOL
xMsgLock	Lock messages for all lower-level modules	BOOL
xRes3	ResBo 5.5	BOOL
xRes4	ResBo 5.6	BOOL
xRes5	ResBo 5.7	BOOL
byFbkDelayRatio	Unit-wide delay ratio for the simulation of CM feedback signals	BYTE
dwBatchID	Batch ID	DWORD
sBatchName	Batch Name	STRING[32]

Feedback signal transmission FxCtrl to UnitIf

One data structure connection EqmFbk to Eqm01Fbk ... Eqm12Fbk per EQM (up to 12 EQMs can be interconnected). The data structure contains the feedback information of the respective equipment module that is used for the status display in the UnitIf faceplate.

(The internal bitwise encoding is not relevant for the user.)

Example of the interconnection FxCtrl to EQM SFC and lower-level control modules



1. Interconnection FxCtrl - SFC type
   In this marking section the mandatory connections (`FuncSel, OvlSel, AUT, ENAUT, MAN, ENMAN`) as well as the optional connection `LOCKHOLD` is displayed. An important role is played by the `FuncSel` input through which the FxCtrl receives the function or step number request from EQM SFC. The reached states, step enabling conditions and error events are supplied to the SFC through the output connections `FuncAct, ReadyNow, MonStaErr, MonDynErr, MonGrpErr` . This allows the system to react to the process states in the SFC steps.
2. Interconnection FxCtrl – CM adapter/CM
   - The interconnection of the unit-specific information signals of the lower-level CMs or of their adapter blocks is made on the basis of a 1:n structure connection, meaning that from the FxCtrl a connection of the output `CmData` to each assigned CM is required.
   - The interconnection of the command and feedback signals of the lower-level control modules or their adapter blocks is made with one 1:1 connection each of the type DWORD from the FxCtrl output `CmCmd<xx>` to the adapter input `CmCmd` of the assigned CMs.
   - One 1:1 connection each is required from the adapter output `CMFbk` to the FxCtrl input `CmFbk<xx>` for the feedback signals of the lower-level elements.

   Note

   Additional details for these connections are available in the section Higher-level controlling of the unit / EQM level (Page 70)

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'FxCtrl' (Page 382)".

Status bit	Parameter
0	`Occupied`
1	`BatchEn`
2	`EqmData.UnitSimOn` (from the unit)
3	`OnAct.Value`
4	`OosAct.Value`
5	`CmSimOn` (from the `Control Module`)
6	1 = All the CMs in Auto Mode
7	1 = All the CMs in Manual Mode
8	`MonDynErr`
9	`MonStaErr`
10	`MonExtErr`
11	1 = Monitoring deactivated
12	1 = Block inactive (IDLE State)
13	1 = Active and no error
14	1 = Error delay time is running
15	1 = Error is active
16	1 = Error outgoing

Status bit	Parameter
17	Reserve
18	MS_Release (first free Bit)
19	`OosLi.Value`
20-22	Reserve
23	1 = Block status IDLE
24	1 = Block status activation
25	1 = Block status activated and position monitoring
26	Pulse step enabling condition fulfilled
27	Pulse step enabling condition no longer fulfilled
28	Step enabling condition fulfilled
29	One / several CMs are in 'force mode"
30	Copy from: Feature Bit 4 (display dec. function ID.)
31	One / several CMs are in 'superposed operation'

Status word allocation 'Status2'

Status bit	Parameter
0 ... 7	Copy of output `FuncAct`
8 ... 15	Copy of output `OvlAct`
16 ... 23	Copy of SFC S88 status output - BA_STATE
24	`1 = MsgLock` OR `CmData.xMsgLock`
25	Copy of BA_STATE status bit 'Batch Enabled'
26	Copy of BA_STATE status bit 'Occupied'
27	Copy of BA_STATE status bit 'Mode Auto'
...	Reserve

9.1.1 Operating modes of 'FxCtrl'

This block can be operated using the following modes:

- On
- Out of service

General information on these operating modes is available in the "Operating modes of the blocks" section of the APL documentation.

The following section provides additional block-specific information related to the general descriptions.

Out of service

The block is set to its initial state. All function- and element-based data is deleted.

Note

Change from "Out of service" to "On"

A change of the function selection is required for safety reasons when you return from the "Out of service" mode to "On". The same applies to overlay functions.

9.1.2 Functions of 'FxCtrl'

9.1.2.1 Connection possibilities

The block `FxCtrl` supports up to 32 connectable elements. It is not possible to connect "foreign" blocks directly here, but only those adapter blocks of the Industry Library itself that are mentioned in the following table:

Adapter block	Provides connection for
• `ApMotL`	• APL block - MotL
• `ApMotRevL`	• APL block - MotRevL
• `ApMotSpdCL`	• APL block - MotSpdCL
• `ApPIDConL`	• APL block - PidConL / PidConM
• `ApVlvL`	• APL block - VlvL
• `ApVlvAnL`	• APL block - VlvAnL
• `ApVlvDsL`	• APL block – VlvDsL (double-seat valve)
`ApCmGen`	Further APL blocks or blocks from third parties
`FxSigBin`	Up to four binary signals that can be controlled and monitored individually or bundled.
`FxSigInt`	An integer control value to be monitored (0 ... 64)
`FxShare`	A `FxCtrl` block operating subordinately that controls "shared equipment".
`FxCtrl`	A further FxCtrl block. As a master the block can control further lower-level FxCtrl blocks that can be connected as a process tag (slave).

The `FxCtrl` includes 32 structure interfaces each at its inputs and outputs. In case of a homogeneous use, for example, either a maximum of 32 "control modules" or "integer control

values" or 32 x 4 = 128 binary signals can be processed. Mixed operation is naturally also possible.

Note

Cascaded operation of several blocks BlFxCtrl

In the case of a chaining of for example 2 instances of the block in accordance with the following scheme:

- `FxCtrl (Master) → FxCtrl (Slave)`
- `FxCtrl (Master) → FxShare → FxCtrl (Slave)`

take into account that only function controls and not overlay definitions are supported in the lower-level block (slave). In the process only function controls of the lower-level `FxCtrl` are offered in the overlay design of the higher-level `FxCtrl` - no overlays, even if these should be defined. In cascaded `FxCtrl` operation the lack of the overlay transmission to the Slave does not mean a functional limitation because when the Master returns from overlay operation the previous function control is always assumed, both in the Master and in the Slave. A Slave therefore does not actually require an overlay function.

Connections of ApPID, FxSigBin

A connected block of this type is only visible in the configuration dialog if a signal interconnection exists at at least one of the ApPID/FxlSigBin input parameters `Fbk1 ... Fbk4`. The respectively associated control output `Cmd1CM ... Cmd4CM` can also serve as the signal source.

9.1.2.2 Function-based activation and monitoring

- Selection and activation of the functions in runtime is carried out using the input parameters `FuncSel` and `OvlSel` that are controlled by the SFC instance of the equipment module as a rule. The block supports up to 64 control and up to 4 overlay functions.
- A function consists of a group of elements (connected control modules or their adapter blocks or other blocks from the table above) as well as specifications for their activation and monitoring.
- The functions used per block instance (configured respectively as an enumeration at the block input `FuncSel` and `OvlSel`) are displayed to the operator in faceplate (multiple languages)

The activations and monitorings made available depend on the type of the connected element:

Adapter block	Activations, monitorings
• ApMotL • ApMotRevL • ApMotSpdCL • ApPID • ApVlvL • ApVlvAnL • ApVlvDsL	• Depending on the block type, the following activations / monitorings are possible for this group: Start, Stop Open, Close Fwd, Rev, Stop, These are all the controls available at the associated APL blocks that have "Aut" in their suffix (StartAut, OpenAut, FwdAut, etc.). • The feedback messages are available differentiated (also in name, i.e. the technical feedback for "Start" is not also called "Start" or "Started", but rather "Run": Run, Open, Fwd, ...
ApGenCm	Up to four neutrally named controls and feedbacks each.
FxSigBin	Each of the four binary signals can be activated and / or monitored.
FxSigInt	Activation value (integer) 0 ... 64. The value can be activated and / or monitored.
FxShare	Specification of a function number, waiting for feedback from the same
FxCtrl	(as above)

A detailed representation of the controls / monitoring functions available for the above-mentioned adapter blocks is available in the section '(5) Control state' in section "Faceplate (Page 391)".

The following additional attributes are available for the configuration of a function element:

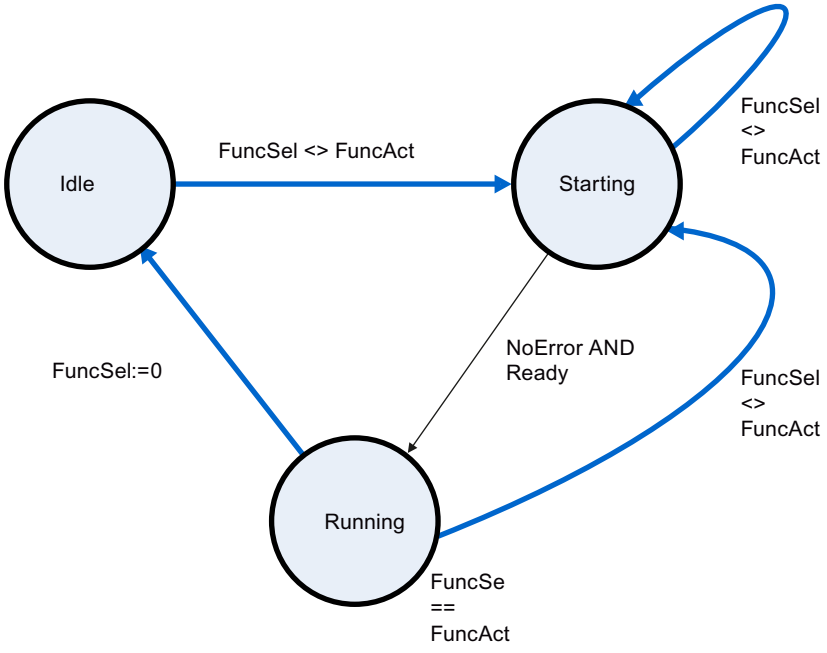
- No activation, only state queries
- Activation is to be carried out conditionally (an additional process condition is to be taken into consideration)
- The signal is included in the error evaluation and is signaled by means of the output MonGrpErr
- The signal is relevant for the step enabling condition (also called end transition) and is signaled through the output ReadNow
- "Simply do nothing"
  This makes cumulating control operation possible.

9.1.2.3 Function states, transitions

`FxCtrl` always assumes one of the following states:

- Idle
  Rest position, no function or overlay request is active (`FuncSel, OvlSel = 0`). All activations and monitorings are shut down.
- Starting
  The period between the request of a function and its successful establishing. A function is considered as successfully established when all the elements to be activated provide a positive feedback and the group signal "next step condition fulfilled" is active.
  In the "Starting" state the control and monitoring information to be executed or activated is read once from the block inputs.
- Running
  The function is considered as successfully established.
  During the "Running" state control and monitoring information is not read any longer. A change at the inputs (through a change in the configuration) does not become active until the next "Starting".

Associated status transition diagram:



These have the following meaning:

`FuncSel`	External function request
`FuncAct`	Function currently being executed
`NoError AND Ready`	Internally formed group state under consideration of the transition-relevant elements. If an error occurs during "Starting" (see below), a status change to "Running" does not take place.

Note

The block itself is passive with regard to the S88 status transitions or sequence of the equipment module. The actual triggering or activation of the control and monitoring functions is always carried out on the basis of the steps in the SFC.

9.1.2.4 Error monitorings

The block `FxCtrl` differentiates between internal and external errors during error monitoring:

Internal error

- Time monitoring of the function request
  - Dynamic monitoring of the elements, meaning whether the commands requested by the function are assumed by the elements
- Feedback monitoring of the function elements
  - Static monitoring, meaning whether the elements 'remain' in the state specified by the function currently active

External errors

- These are explicit element errors such as a CM group error or a binary signal marked as an error.

The error monitorings are state-dependent. In addition, each error group includes its own configurable time delay. The following table shows the correlations:

Parameter	Type of monitoring time	Active state for the monitoring
`MonTiDyn`	Time monitoring of the function request "Dynamic monitoring time" – The time delay of the error detection for an explicitly triggered (expected) state change.	Starting
`MonTiSta`	Feedback monitoring of the function elements "Static monitoring time" – The time delay of the error detection for an unexpected position change.	Running
`MonTiExt`	Time delay of the explicit element error They time delay of the error recognition for an element error.	Starting + Running

The error evaluation of these three groups can be enabled or locked at (interconnectable) block inputs:

- EnMonDynamic
- EnMonStatic
- EnMonExtern

If the parameters are interconnected, they are only displayed in faceplate. Otherwise they can also be changed during the runtime.

Recognition of parameter configuration and control errors

If the interconnection of the lower-level elements as well as the FxCtrl configuration in the ES configuration dialog is not consistent (for example interconnection change after the function configuration), faulty control data can arise for the block execution. In order to exclude internal runtime errors as far as possible, the parameter configuration is subjected to a (cyclic) consistency check. Implausible data are generally not accepted, critical changes to the configuration are recognized as such and rejected. In addition an "AS – control system message – error" is generated. As long as a parameter configuration error is active, no internal data transfer takes place, the output image is 'frozen".

The display is carried out in the standard view of the faceplate of the respective instance. In addition error messages are generated.

The following parameter configuration and control errors are recognized:

No.	Error description / Event text	Remarks
1	Control error: Invalid function identifier	The number range between 0 and 64 is valid for function requests.
2	Control error: Invalid overlay function identifier	The number range between 0 and 4 is valid for overlay function requests.
3	Configuration error: Function definition missing	A FxDefXX block that contains the function definitions was not connected
4	Configuration error: Invalid function definition	The definition of a function element is carried out via a hexadecimal-coded WORD. Valid characters are '0' to '9' as well as 'A' to 'F'. Lower-case characters are not supported. All other characters also result in this error.
5	Configuration error: Invalid overlay function definition	As for No. 4, only for the overlay function
6	Configuration error: Element types inconsistent (ES)	The element types described in CmTypeDef by the ES configuration dialog do not match the actually connected elements
7	Configuration error: Element types inconsistent (Fbk/Cmd)	The element types of the CmCmd/CmFbk structures at a block index differ
8	Configuration error: Interconnections swapped	The CmCmd/CmFbk connections do not lie on the same connection index

No.	Error description / Event text	Remarks
9	Control error: Function is not enabled	A function was selected that was not enabled by the ES configuration dialog (`FuncDefHi/Lo`).
10	ES data transfer timeout	Online writing of a function definition by the ES configuration dialog is time-monitored. (Input `ESTransAct`). The monitoring time is specified at the input parameter `ESTransTm` (default setting: 5.0 sec)

In order to make the error cause more transparent for the user it is attached as a message associated value and converted to readable form using a text library (see the section "Messages of 'FxCtrl' (Page 381)"). Parameter configuration errors are signaled at two outputs:

- `ParamErr [BOOL] Display of parameter configuration error`
- `ParamErrNo [INT] Cause`

9.1.2.5 Overlay functions

The `FxCtrl` block provides the possibility of overlaying the function currently being processed for particular operating states (for example a change of the SFC type to the state HOLD). Thus if `FuncSel > 0` and `OvlSel > 0` simultaneously, the overlay function is activated and the standard function is deactivated. When reversed (`OvlSel = 0`), the previous state of the elements and thus the previous function is established automatically. Further properties of the overlay function are:

- Only one function can be active at any one time.
- While an overlay function is active, requests for a change of the main function are ignored.
- The same processing options are available in an overlay function as in a "normal" function.
- `FxCtrl` supports up to 4 overlay functions. Activation is carried out using the block input `OvlSel`
  - 0 = No overlay
  - 1 to 4 = Activate overlay function 1 to 4.

9.1.2.6 Element-wise forced operation

Every function element can be included or excluded from processing forcibly during the runtime irrespective of its consideration by the function currently active. Activation is carried out by the operator in the faceplate. In this context, however, "forced operation" refers only to the `CmCmd<xx>` output of the `FxCtrl` block. It should not be confused with the "Force" functions of a subsequent APL block for example.

9.1.2.7 Trigger for feedback simulation

The `FxCtrl` also includes the option of activating / deactivating the external simulation in the faceplate. Through the external simulation the type-specific CM adapters can simulate the feedbacks corresponding to the respective control for the APL block (if desired also time-delayed).

9.1.2.8 Passing on the initial state request

With regard to the passing on of the initial state request at unit allocations the `FxCtrl` block behaves like its non-function-based counterpart `EqmIf`:

- Switchover to the automatic operating mode
- Issuing of a reset command
- Deactivation of the external simulation

9.1.2.9 APL standard functions of 'FxCtrl'

Calling additional faceplates

This block includes the function "Call additional faceplates". To this purpose the input parameter `CmFbk01 ... CmFbk32` has the attribute 'k_Jump'. If CFC interconnections are available at these inputs, the respectively interconnected Control Module instances can be opened in the OS faceplate.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in the *"APL documentation" in the section "Configurable functions with the Feature I/O"*. The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	See APL function: "Setting the startup response" Startup without call in OB100 (default = 0)
1	See APL function: "Reaction to the out of service mode" Release for changing to "out of service" via OosLi (default = 0)
2	Display of the CM instance name *) 1 = CFC name (process tag) as per preset value
3	Display of the CM instance name *) 1 = Block name
4	Display of the function numbers in decimal form
...	...
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Activate local operator permission" 1 = "Local" operator permission is active
...	...
*)→ It is also possible to have both names together!	

Operator permissions

This block includes the APL standard function "Operator permissions". The block has the following authorizations for the `OS_Perm` parameter:

Bit	Parameter
0	1= Operator can switch to "On" mode
1	1= Operator can switch to "Out of service" mode
2	1= Operator can reset "memory error"
3	1= Operator can activate the CM feedback simulation
4	1= Operator can deactivate the CM feedback simulation
5	1= Operator can switch the CMs to automatic
6	1= Operator can switch the CMs to manual
7	1= Operator can reset the CMs
8	1= Operator can switch the CMs to force mode
9	1= Operator can enable the monitoring of dynamic errors
10	1= Operator can enable the monitoring of runtime errors
11	1= Operator can enable the monitoring of external errors
12	1= Operator can change the monitoring time for dynamic errors
13	1= Operator can change the monitoring time for runtime errors
14	1= Operator can change the monitoring time for external errors
15	1= Operator can activate the release for maintenance
16	1= Operator can reset the force mode for all CMs
...	Reserve

Overview of error numbers

The following error numbers are output using the connection `ErrorNum` :

Bit	Parameter	Remark
-1	Default value when implementing the block; block will not be processed	
0	There is no error.	
1	Control error: Invalid function identifier	The number range between 0 and 64 is valid for function requests.
2	Control error: Invalid overlay function identifier	The number range between 0 and 4 is valid for overlay function requests.
3	Parameter assignment error: Function definition missing	A `FxDefXX` block that contains the function definitions was not connected
4	Parameter assignment error: Invalid function definition	The definition of a function element is carried out via a hexadecimal-coded WORD. Valid characters are '0' to '9' as well as 'A' to 'F'. Lower-case characters are not supported. All other characters also result in this error.
5	Parameter assignment error: Invalid overlay function definition	As for No. 4, only for the overlay function
6	Configuration error: Element types inconsistent (ES)	The element types described in `ESCmTypeDef`do not match the actually connected elements

Bit	Parameter	Remark
7	Parameter assignment error: Inconsistent element types (Fbk/Cmd)	The element types of the CmCmd/CmFbk structures at a block index differ
8	Parameter assignment error: Inputs/outputs swapped	The CmCmd/CmFbk connections do not lie on the same connection index
9	Control error: Function is not enabled	A function was selected that was not enabled by the ES Wizard (ESFuncDefHi/Lo).
10	ES data transfer timeout	Online writing of a function definition by the ES configuration dialog is time-monitored. (Input ESTransAct). The monitoring time is specified at the input parameter ESTransTm (default setting: 5.0 sec)

9.1.3 Messages of 'FxCtrl'

The block uses an instance of the ALARM_8P block (MsgEvId). The following messages can be generated:

Message instance	Message identifier	Message class	Remark
MsgEvId	SIG 1	AS process control message - fault	$$BlockComment$$ Control error
	SIG 2	AS process control message - fault	$$BlockComment$$ Feedback error
	SIG 3	AS process control message - fault	$$BlockComment$$ External element error
	SIG 4	AS process control message - fault	$$BlockComment$$ @...ILFxCtrlParamErrors@
	SIG 5	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 6	AS process control message - fault	$$BlockComment$$ External message 2
	SIG 7	AS process control message - fault	$$BlockComment$$ External message 3
	SIG 8	AS process control message - fault	$$BlockComment$$ External error has occurred

Explanation:

- $$BlockComment$$ → Content of the instance-specific comment
- @...FxCtrlParamerrors@ → generates an event identifier using the text library FxCtrlParamErrors' (see table "Recognition of parameter configuration and control errors" in the section "Error monitorings (Page 376)").

- You can interconnect an external fault to the input parameter CSF. If this signal changes to `CSF = 1`, a control system error is triggered (MsgEvId, SIG 8).
- In addition up to three external messages SIG 5, SIG 6, SIG 7 can be generated by using the input parameters `ExtMsg1 ... ExtMsg3` .

Associated values for message instance MsgEvId

Associated value	Block parameter
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ErrorNum`
5	`ExtVa105`
6	`ExtVa106`
7	`ExtVa107`
8	`UnitIDOut`

The associated values 5 ... 7 are allocated to the parameters `ExtVa105 ... ExtVa107` and can be used by yourself. See the "Process Control System PCS 7 - Engineering System" manual.

9.1.4 Connections of 'FxCtrl'

Input parameters

Parameter	Description	Type	Default
`STEP_NO`	Step number from EPH (SFC)	`DWORD`	
`BA_STATE`	Batch Status DWORD	`DWORD`	
`CmManModOp`	Switch equipment elements to manual mode	`BOOL`	
`CmAutModOp`	Switch equipment elements to automatic mode	`BOOL`	
`CmSimOnOp`	Activate simulation for equipment elements	`BOOL`	
`SimEnEqm`	Enable equipment simulation	`BOOL`	`True`
`EqmData`	Equipment info structure from UnitIf	`BlUdtEqmData`	
`FuncSel`	Function selection (0..64)	`INT`	
`OvlSel`	Overlay function selection (0..4)	`INT`	
`FmRstOp`	Reset failure memory (operator)	`BOOL`	
`RstFrcOp`	Reset force mode for all elements (operator)	`BOOL`	
`CmRstOp`	Reset CM errors (operator)	`BOOL`	
`FmRstLi`	Reset failure memory (process)	`DigVal`	
`CmRstLi`	Reset CM errors (process)	`DigVal`	
`EnMonSta`	Enable monitoring of unexpected feedback changes	`BOOL`	`True`
`EnMonDyn`	Enable monitoring of expected feedback changes	`BOOL`	`True`

Parameter	Description	Type	Default
EnMonExt	Enable monitoring of external element errors	BOOL	True
MonTiSta	Monitoring time for unexpected feedback changes [s]	REAL	5.0
MonTiDyn	Monitoring time for expected feedback changes [s]	REAL	5.0
MonTiExt	Monitoring time for external element errors [s]	REAL	5.0
FxCmData	Control and status information from higher-ranking FxCtrl / FxShare	BlUdtCmData	
FxCmCmd	Command data from higher-ranking FxCtrl / FxShare	BlUdtCmCmd	
OvwUnitID	1=Overwrite CmData unit ID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
OvwManOpEn	1=Overwrite CmData ManOpEn with input ManOpEn	BOOL	
UnitID	Unit ID (usually from UnitIf block)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIdLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
ManOpEn	Enable switchover to manual mode	BOOL	
MsgLock	Inhibit process messages	DigVal	
MsgEvId	Message event ID	DWORD	16#FFFFFFFF
ExtMsg1 ... ExtMsg3	External message signal 1... 3	DigVal	
ExtVa105 ... ExtVa107	External value 4 ... 7	ANY	
CmFbk01 ... CmFbk32	CM feedback 1 ...32	BlUdtCmFbk	
ESTransAct	... Internal	INT	
ESTransTm	... Internal	REAL	5.0
OSOpCmd	... Internal	UdtOpCmd	
OSFxElOnly	... Internal	DWORD	
FuncDef	Function definitions	BlUdtDefData	
MS_RelOp	Operator input for MS Release, 1: MS release requirement	BOOL	
RunUpCyc	Number of cycles for which this block keeps in start up mode	INT	3
SampleTime	Sampling time [s]	REAL	1.0
UserStatus	User status bits	BYTE	
CSF	Control System Fault message - external error	BOOL	
OnOp	1=On mode: On mode by operator	BOOL	
OosOp	1=Oos mode: Oos mode by operator	BOOL	
OosLi	1=Oos mode: Oos mode switch via link	DigVal	
OS_PERM	Operator Permissions	STRUCT	

Parameter	Description	Type	Default
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

Output parameters

Parameter	Description	Type	Default
AUT	SFC command: Set automatic mode	BOOL	
ENAUT	SFC command: Enable automatic mode	BOOL	
MAN	SFC command: Set manual mode	BOOL	
ENMAN	SFC command: Enable manual mode	BOOL	
MS_Release	1: Maintenance status release	BOOL	
OnAct	On mode is active	DigVal	
OosAct	Out of service is active	DigVal	
UnitIDOut	Unit ID	DWORD	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING[32]	
StepNo	Step number from EPH	DWORD	
FxState	Internal processing state	INT	
FuncAct	Active function ID	INT	
ReadyCame	Active function's ready signal came	BOOL	
ReadyWent	Active function's ready signal went	BOOL	
ReadyNow	Active function is ready at the moment	BOOL	
CmSimEn	Local control module simulation is enabled	BOOL	
CmSimOn	Simulate control modules' feedback	BOOL	
CmEqmLock	Routed signal from unit, can be used in own interlocks	BOOL	
CmMsgLock	Lock messages	BOOL	
CmManOpEn	Enable operator input manual mode	BOOL	
P_Rst	Reset pulse	BOOL	
MonDynErr	Feedback error due to an expected change in control output	DigVal	
MonStaErr	Feedback error due to an unexpected feedback change	DigVal	
MonExtErr	External element error	DigVal	
MonGrpErr	Runtime function group error (dynamic, static or element error)	DigVal	
ErrorNum	Parameterization error number	INT	
ErrAddInfo	Additional error information	INT	
CmModLiOp	Mode switch selector	DigVal	
CmAutModLi	Switch to AUTO mode	DigVal	
CmManModLi	Switch to MANUAL mode	DigVal	

Parameter	Description	Type	Default
OSCmInUse	... Internal	DWORD	
OSCmSigLev	... Internal	STRING	
OSUpdate	... Internal	INT	
OSCmCnt	... Internal	INT	
OSRelink	... Internal	DWORD	16#80000000
OSCmRtd01 ... OSCmRtd32	... Internal	DWORD	
CmData	Control Module Information structure	BlUdtCmData	
CmCmd01 ... CmCmd32	Command data 1 ... 32	DWORD	
EqmFbk	This module's feedback for UnitIf	BlUdtEqmFbk	
MsgErr	ALARM_8P: Messaging error	BOOL	
MsgStat	ALARM_8P: Status output	WORD	
MsgAckn	ALARM_8P: ACK_STATE output	WORD	
OS_PermOut	Operator permissions (OS info)	DWORD	16#FFFFFFFF
OS_PermLog	Operator permissions (OS info)	DWORD	16#FFFFFFFF
OS_PermLnk	Operator permissions (OS info)	DWORD	
OpSt_Out	Enabled operator stations	DWORD	
Status1	Status1 data	DWORD	
Status2	Status2 data	DWORD	
ST_WORST	Worst signal status	BYTE	16#80
FxFbk	Feedback data for EqmIf / FxCtrl / FxShare	BlUdtCmFbk	

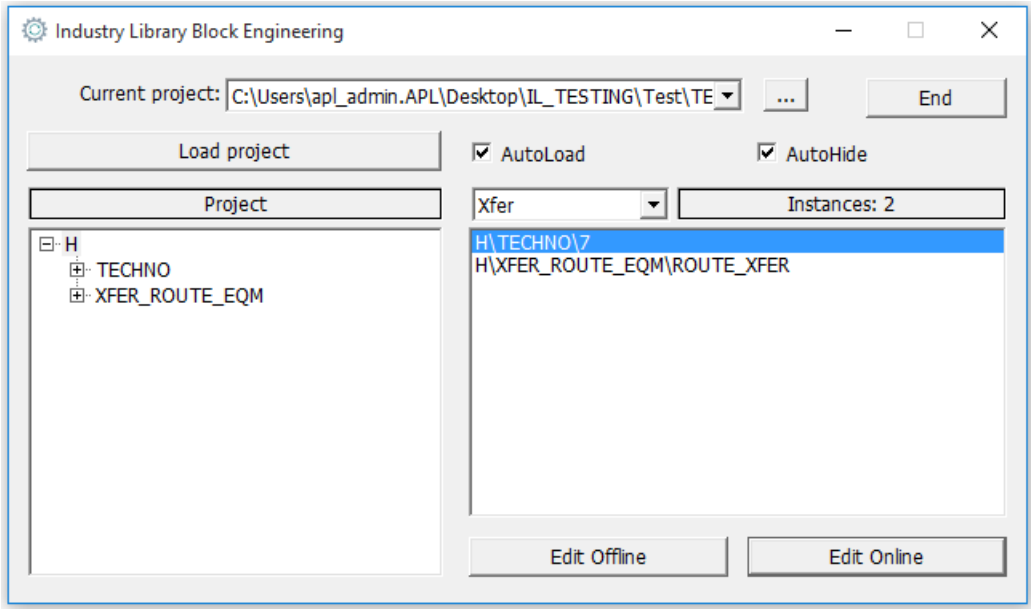
9.1.5 Configuration of the function definitions

Configuration

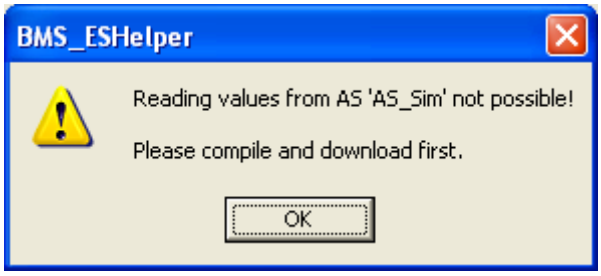
Configuration is carried out in the following steps:

1. Instantiating and interconnecting on the described blocks and adapters in the CFC Editor.
2. Parameter assignment to the enumerations for the inputs:
   - `FuncSel` (default 'FxFuncDef1'),
   - `OvlSel` (default 'FxOvlDef1'),
3. Calling up the ES configuration dialog using the SIMATIC Manager menu "Options → SIMATIC Industry Library block configuration" → the dialog for reading in the ES project and selecting the block type (FxCtrl, Xfer) as well as the desired block instance is displayed first.

9.1.5.1 Configuration dialog "Project overview screen"



(1)	Selecting and downloading the ES project	• "Current Project" – Select the path for the ES project • "Load Project" button – For reading in the project data The project has to be loaded again after configuration changes in the FxCtrl instances and enumerations (CFC Editor or SIMATIC Manager) • "Auto Load" check box – The project last selected is also loaded whenever the application is started – After a new project has been selected, it is loaded immediately • "Auto Hide" check box – After the editing window (Edit Offline, Edit Online) has been loaded, the "Project overview screen" is minimized automatically so that, for example, it cannot cover a CFC in the background. • "Exit" button – Exits the application
(2)	Select block types and instances	• List box for selecting the block type (FxCtrl, Xfer) Only instances of the respective type are displayed • The CFC folders and charts with clock instances are displayed in tree view under "Project" • The charts with instances for the selected folder are listed in tree view under "Instance(s) x/y"

(3)	Start the editor dialog	The editing window for the selected instance can be opened either in Offline or Online mode: • Offline – can always be opened for instantiated blocks with basic wiring • Online – can only be opened under the following preconditions: – Block instances have been compiles and loaded to AS – Connection to AS is functioning and CFC test mode is possible Note: Due to CFC restrictions the command "Compile and download AS changes" first has to be carried out in the CFC editor after offline changes before the editor can be started in Online mode. A message box is displayed if the preconditions for Online mode are unfulfilled.
		

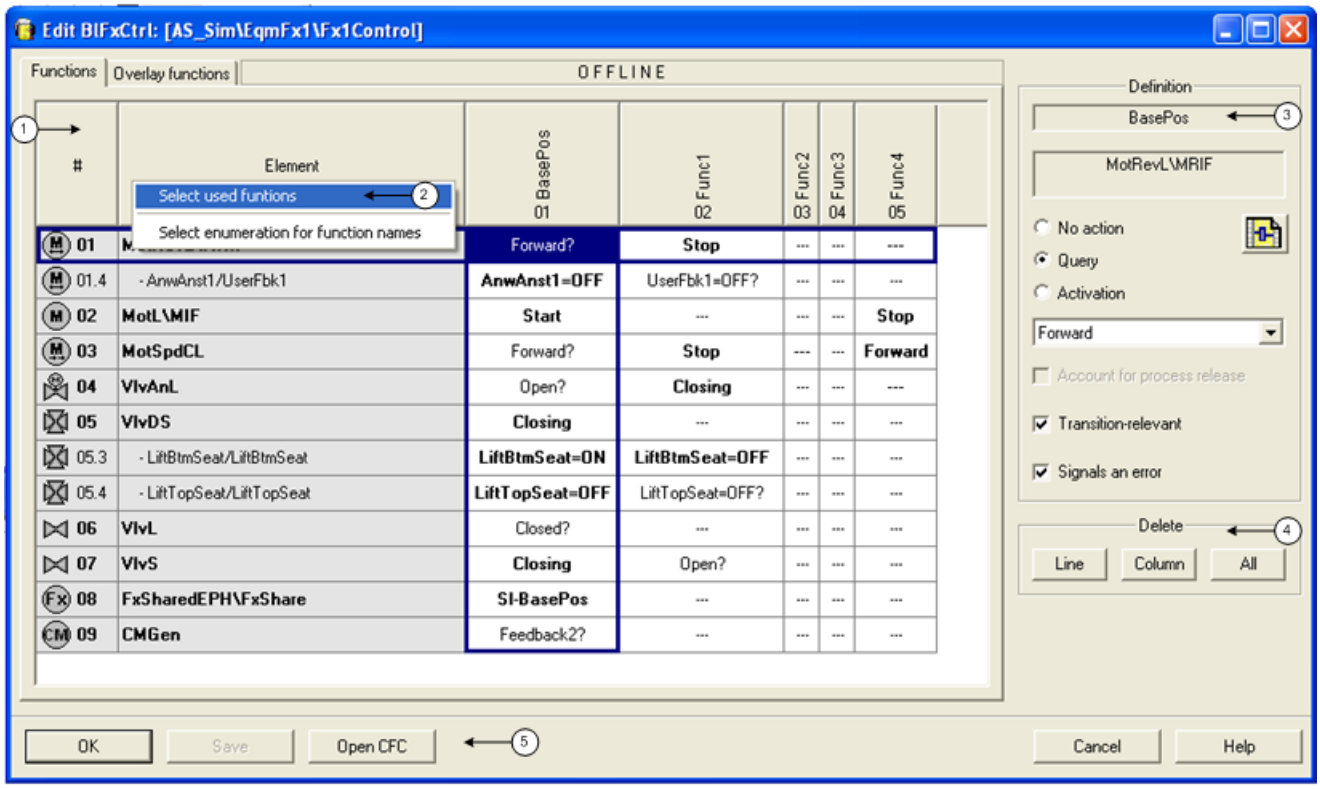
9.1.5.2 Configuration dialog "Editing window"

Introduction

Configuration of the block instance takes place in the editing window described below. Tabs are available respectively for the configuration of the

- Function definition
- Overlay function

NOTICE
Changes in online mode In the case of configuration changes in online mode and subsequent saving, the current block instance is written directly to the AS after a repeated confirmation prompt. Before saving you have to ensure that the element controls cannot induce damage in the system for all the functions.



Explanation of the screen shot	
(1)	**Table with functions and elements**
	• The elements interconnected to this block instance are displayed with the element type, number and name (derived from the `CmFbk01 ... CmFbk32` connections) in the lines from top to bottom. • Element designations in bold font are the main element whereas indented texts in normal font reference the subelements belonging to the main element such as secondary functions or user-specific command extensions (derived from the configured enumerations of the respective element). • A maximum of 64 function definitions or a maximum of 4 overlay definitions are displayed after the elements in the columns from left to right. The function / overlay names are derived from the correspondingly configured enumerations (see above). • The tooltip text displays the instance of the block FxDefxx in the header line and the block instance of the CM adapter or FxShare block in the element lines.

Explanation of the screen shot	
(2)	**Selection of the enumerations and used functions**
	The following specifications have to be made in the shortcut menu (right-hand mouse button) of the function line (this sequence should be observed during the first callup): • Selection of enumeration for function names The enumeration valid for the function name is to be selected from the list of available enumerations (is usually carried out once at the beginning of configuration) • Selection of functions used In the window then opened the active functions are specified by moving the function into the corresponding column "Hide" or "Display"
(3)	**Function definition**
	Configuration of a function definition is carried out by means of a property field that always references the cell currently selected in the matrix. A function definition can contain the following options depending on the element type (options that are not available are grayed out): • Option 'No action' – The element is to keep its last state • Option 'Query' meaning no activation • Option 'Activation' • The type-dependent query and activation commands are selected in the list box (for possible commands refer to the table in the section ("Faceplate (Page 391)", section "Standard view of the FxCtrl"). In the case of "cascaded" blocks FxCtrl or lower-level "shared elements" / FxShare block the slave function definitions appear here (overlay definitions of the slave are not supported). • 'Account for process release' check box • 'Transition-relevant' check box, meaning is the feedback relevant for the function result? • 'Signals an error' check box, meaning is the feedback relevant for the error evaluation? • When in the configuration dialog the 'Help' button is used to open the documentation for the configuration dialog.
(4)	**Clear function**
	The following clearing functions are available here: • Line – All the function definitions are set to "No action" for the selected element • Column – All the elements are set to "No action" for the selected function definitions • All – All the function definitions of this block instance are set to "No action"

Explanation of the screen shot	
(5)	**Command buttons**
	• OK Editor is exited – Storing of changes is carried out implicitly in Offline mode – In Online mode a query may be displayed whether changes are to be rejected • Save – In Offline mode storing is carried out offline – In Online mode storing is carried out online and offline
	Note: Due to CFC restrictions "Compile and download AS changes" first has to be carried out in the CFC editor after offline changes (for example through changes in the process object view or a second instance of the configuration dialog) before storing in Online mode is possible. A corresponding message box is displayed if the preconditions for storing are not fulfilled.
	• Open CFC – The CFC editor can be opened by using the button. Compiling / Downloading can, for example, be carried out here. • Abort – Exit the editor If changes in the configuration have been carried out, a prompt is displayed whether these are to be rejected • Help – Opens the documentation window (CHM file) with the subsection "Configuration of the function definitions".

Note

1:n interconnection between FxDef16 / 32 / 64 and FxCtrl block

- If the function definition storage block of the type FxDef16 / 32 / 64 is connected with several FCtrl blocks ("multi-instance configuration" at equipment modules with identical function definition), an additional information is displayed during storing that several FxCtrl instances are changed by this procedure.
- This configuration of a "multi-instance" is displayed in the title bar of the editor window next to the block instance.

9.1.6 Operator control and monitoring

9.1.6.1 Symbol



Tag name
Status bar
Block instance name
Active function (from enumeration)

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	Common operating mode of lower-level elements	
		Automatic
		Manual
		Auto / Manual mixed
2	Operating mode	
		On
		Out of service
3		Error monitoring on / unfulfilled
		Error monitoring on / fulfilled
4		Step enabling condition unfulfilled
		Step enabling condition fulfilled
5		One / several CMs are in 'overlayed operation'
6		One / several CMs are in 'force mode'
7		Unit not released
		Unit released
8		Unit allocated
9		Notice is available

9.1.6.2 Faceplate

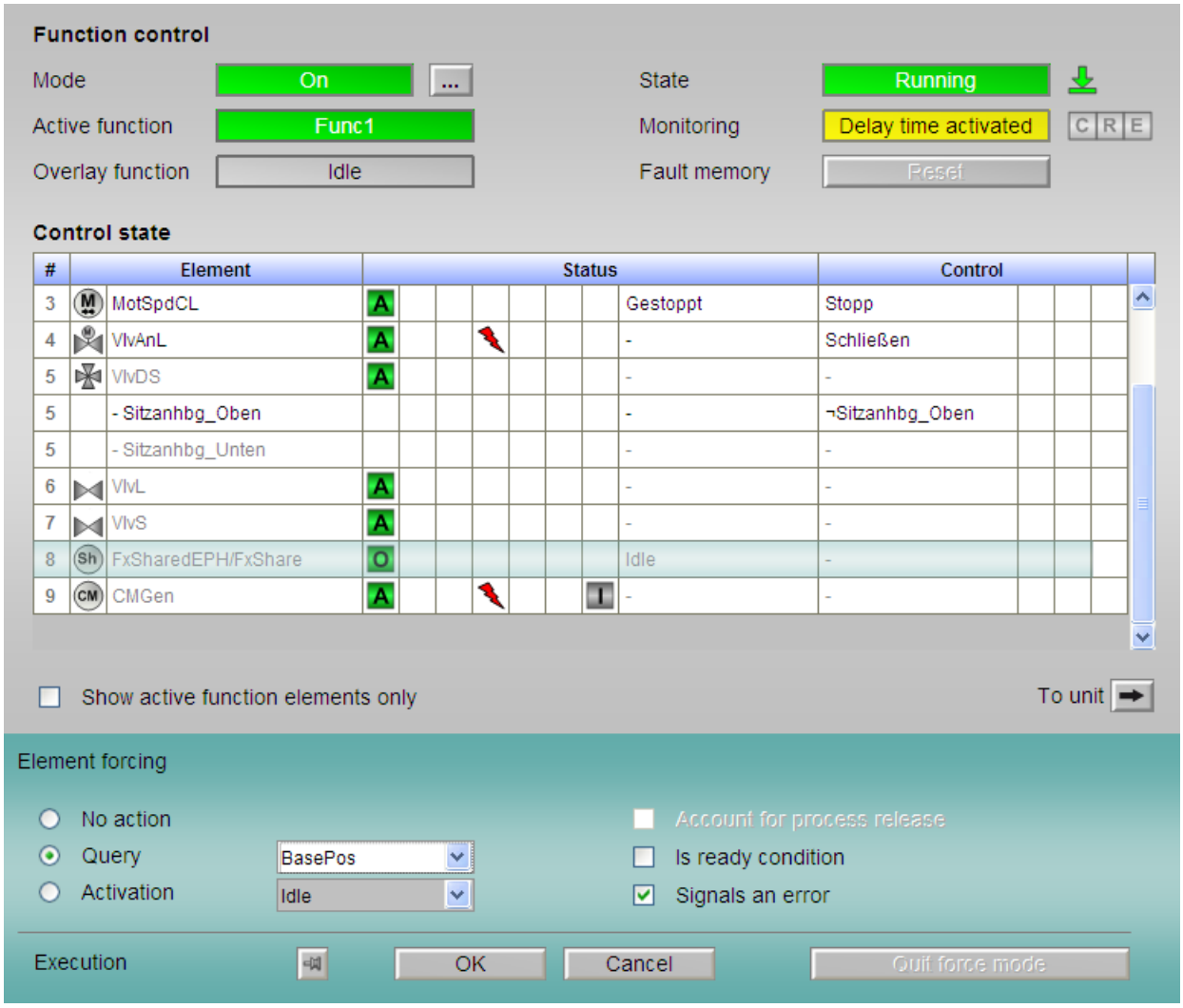
Standard view of the FxCtrl

The block FxCtrl provides the following views:

	Standard view	See below.
	Message view	Standard message view of the Advanced Process Library
	Parameter view	See below.
	Preview	See below.
	External sequential control system	See below.

	Memo view	Standard memo view of the Advanced Process Library
	Batch view	Standard batch view of the Advanced Process Library

Standard view of the FxCtrl.



Explanation of the screen shot	
(1)	**State**
	The field contains information about the block status • "Idle" • "Activation" new function / control to the CMs has been output • "Activated" new function / control to the CMs has been carried out / position monitoring active

<table>
<tr><th colspan="2">Explanation of the screen shot</th></tr>
<tr><td>(2)</td><td>Monitoring</td></tr>
<tr><td></td><td>The field contains information about the block monitoring status
• Deactivated
• Inactive (IDLE)
• Active (no error)
• Wait time running
• Error
– Details of the monitoring are displayed to the right with the following symbols:
– The display symbols are to be understood as group displays of the lower-level elements</td></tr>
<tr><td></td><td>
<table>
<tr><td></td><td>Runtime error inactive</td><td></td><td>Runtime error active</td></tr>
<tr><td></td><td>Static error inactive</td><td></td><td>Static error active
• Evaluation is carried out at function definitions with the attribute "error-relevant"</td></tr>
<tr><td></td><td>External error inactive</td><td></td><td>External error active</td></tr>
</table>
</td></tr>
<tr><td>(3)</td><td>Resetting of the error log</td></tr>
<tr><td></td><td>In case of a corresponding permission this button can be used to send a reset request to all the lower-level modules through an additional operating dialog.</td></tr>
<tr><td>(4)</td><td>Function control</td></tr>
<tr><td></td><td>The fields contain information about the active control.
• Operating mode ("On" / "Out of service")
The button can be used to change the operating mode though a further operating dialog if corresponding permission is available.
• Active function → input parameter FuncAct
• Overlay → input parameter OvlAct</td></tr>
</table>

Explanation of the screen shot

(5) **Control state**

- This section contains a table with all the connected elements with their states.
- The Element column displays optionally the CFC names of the *process tags*, those of the CM block instances or both together (selection by means of Feature Bits).
- The state column contains the following information:

Operating mode	
	Manual
	Automatic
	Local
	On
	Out of service
	Interlock
	Simulation
	Monitoring error
	Forced operation (APL basic function "Forcing of operating states")
	Out of service
The text column contains the feedback status of the element (see table)	

- The function column contains the currently controlled function definition of this element in accordance with the following table.
  Further attributes of the control are represented by the following symbols:

	Conditional activation (process release) / unfulfilled
	Conditional activation (process release) / fulfilled
	Transition-relevant / unfulfilled
	Transition-relevant / fulfilled
	Error monitoring on / unfulfilled
	Error monitoring on / fulfilled

Clicking a CM line calls the corresponding faceplate.

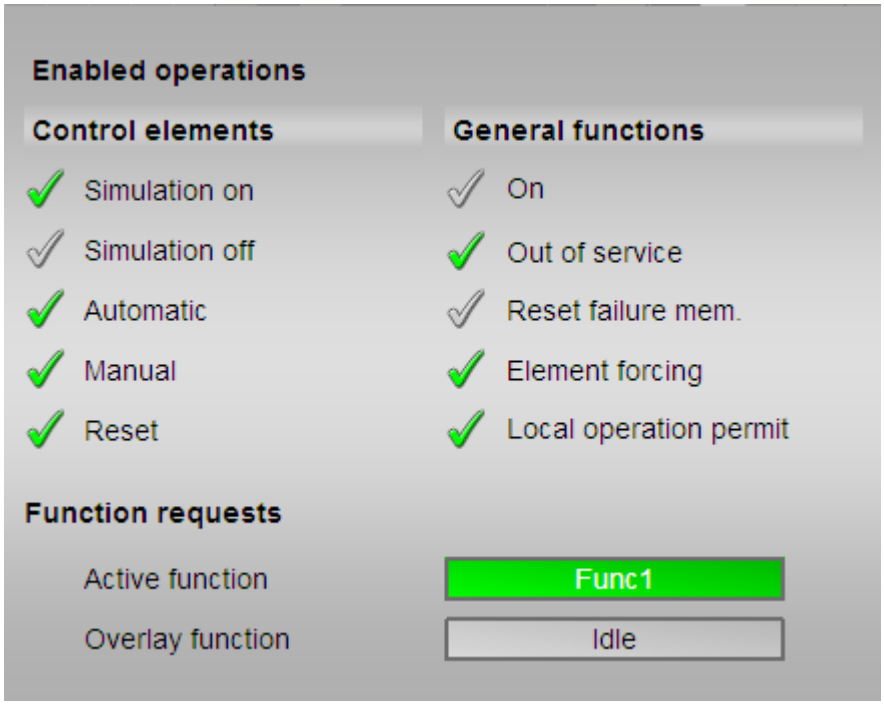
The resolution of the activation and feedback values is displayed in the following table (specified in the type-specific enumeration):

Element type	Activation signal	Activation value	Feedback
`MotL`	Start Stop	1 2	Running Stopped

Explanation of the screen shot				
		AnwAnst1 AnwAnst2	3 4	AnwRM1 AnwRM2
	`MotRevL`	Forward Reverse Stop AnwAnst1	1 2 3 4	Forward Reverse Stopped AnwRM1
	`MotSpdCL`	Forward Reverse Stop AnwAnst1	1 2 3 4	Forward Reverse Stopped AnwRM1
	`VlvL`	Open Close AnwAnst AnwAnst2	1 2 3 4	Opened Closed AnwRM1 AnwRM2
	`VlvAnL`	Open Close AnwAnst AnwAnst2	1 2 3 4	Opened Closed AnwRM1 AnwRM2
	`VlvDS`	Open Close Sitzanhebg_Oben Sitzanhebg_Unten	1 2 3 4	Opened Closed Sitzanhebg_Oben Sitzanhebg_Unten
	Note: • The bits not required for internal controls are made available as "user commands" (outputs `UsrCmd1Out` / `UsrCmd2Out`) or "user feedbacks" (inputs `UsrFbk1` / `UsrFbk2`). • The type-specific designations of the commands and feedbacks are specified through an enumeration at the respective adapter (see description of the adapter blocks).			
(6)	**Show active function elements only**			
	This check box can be used to hide elements that are not used in the current function (light-gray font in the element texts).			
(7)	**Calling the unit faceplate**			
	This button is used to call the higher-level UnitIf faceplate.			

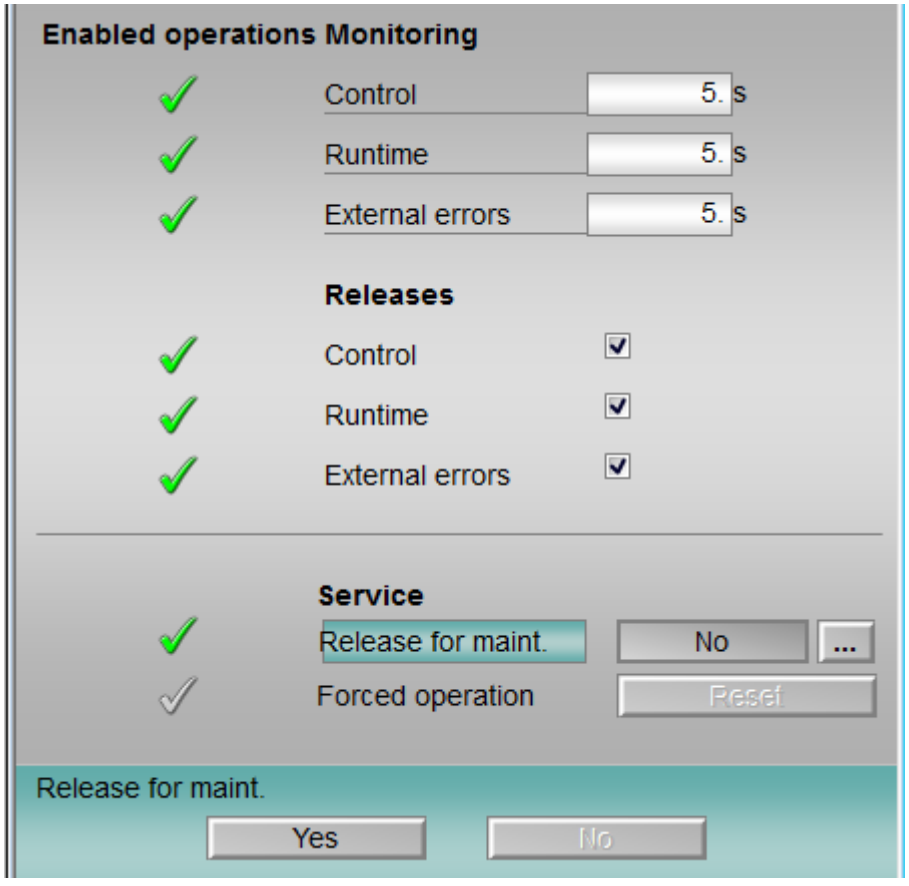
Explanation of the screen shot	
(8) below	**Forced operation**
	An additional dialog can be opened by right-clicking on the element line for each connected element. This allows the activation of the forced operation for this element with the options and check boxes described below. Note: Forced operation here means: The control that can be set manually in the additional dialog for this element overwrites the currently requested element control from the configured function definition: • Option 'Activation' • Option 'Query' meaning no activation • Option 'No action' meaning that the element is to keep its last state • The type-dependent query and activation commands are selected in the respective list box. In the case of "cascaded" blocks FxCtrl or lower-level "shared elements" / FxShare block their function definitions appear here. • 'Account for process release' check box • 'Transition-relevant' check box, meaning is the feedback relevant for the function result? • 'Transition-relevant' check box, meaning is the feedback relevant for the function result? The selected settings are accepted with the OK button and the control for this element is written at the block input `CmCmd<xx>` . Forced operation per element is retained until it is terminated explicitly even if other functions are actuated by the automatic program in the meantime. This is visualized by the yellow marking of the element line. Central termination of the forced operation for all the lower-level elements of this block instance is possible in the parameter view

Preview view of the FxCtrl



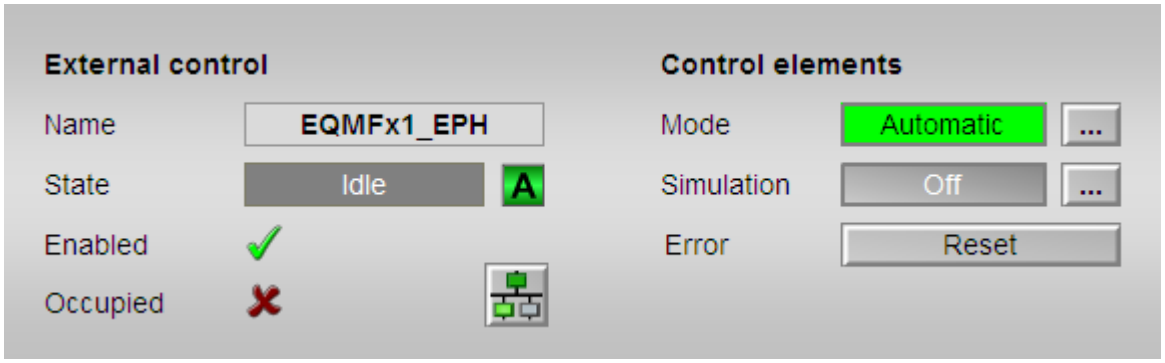
Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the OS_PERM parameter of this block. The operator control enables shown in the screenshot are possible here. Meaning of the symbols: • Green check mark: – The OS operator can control this parameter • Gray check mark: – The OS operator cannot control this parameter at this time due to the process • Red cross: – The OS operator cannot control this parameter due to the configured operator permissions
(2)	**Function requests**
	The required function or overlay definitions are displayed in this area. These can differ from the definition currently active (see standard view): • Active function → input parameter `FuncSel` • Overlay → input parameter `OvlSel`

Parameter view of FxCtrl



Explanation of the screen shot	
(1)	**Monitoring times**
	The delay times for the error monitoring can be configured here: • (Operator permission = "Higher-order operator process controls") – Control monitoring time until feedback (for valves → open, close, etc.) – Runtime monitoring time when controlled state is left – External error – error feedbacks from the CMs The current operator control enables of the block instance are displayed in the "Operator control enable" section. They depend on the parameter `OS_Perm` of the block.
(2)	**Releases**
	The releases for the error monitoring can be configured here: • (Operator permission = "Higher-order operator process controls") – Control – Runtime – External errors The current operator control enables of the block instance are displayed in the "Operator control enable" section. They depend on the parameter `OS_Perm` of the block.
(3)	**Service**
	• Release for maintenance "No" / "Yes" – Prerequisite for issuing the release for maintenance is that the block is in one of the operating modes "On" or "Out of service". You set the release for maintenance (operator control permission "Higher-order operator process control" required) in the parameter view using the input parameter MS_RelOp = 1. A release for maintenance is then made available via the interconnectable output parameter MS_Release = 1 for further processing. The issuing of a release for maintenance does not have any influence on the function of the block. An operation message is generated. • Forced operation resetting – The button can be used to revoke the forced operation for all the lower-level elements (operator permission "Higher-order operator process control" required).

External sequential control system of FxCtrl



Explanation of the screen shot	
(1)	**Displaying and switching the operating mode**
	This section shows the currently valid operating mode for the EQM as well as the lower-level CMs. The following operating modes can be shown here: • Automatic mode → lower-level CMs are in automatic mode • Manual mode → lower-level CMs are in manual mode • (Mixed) → lower-level CMs have different operating modes The button can be used to switch the EQM incl. the lower-level CMs to the operating mode 'Automatic' or 'Manual operation'.
(2)	**Switching the unit simulation on / off**
	This section shows the state of the unit simulation. The following states can be shown and executed here: • On • Off
(3)	**Resetting the modules**
	In case of a corresponding permission this button can be used to send a reset request to all the lower-level modules through an additional operating dialog.
(4)	**External sequential control system**
	This section displays the current state of the EQM SFC as well as whether it is enabled for SIMATIC BATCH and a batch is allocated to it. The ICON button can be used to call the associated SFC Visualization.

9.2 FxSigBin – Binary signal routing block for FxCtrl

9.2.1 Description of 'FxSigBin'

Area of application and operating principle

This function block acts as an adapter for up to four binary signals (control and feedback signal each) for an upstream FxCtrl block. Independent of one another, these binary signals can be either controlled/monitored or only monitored (can be set with Feature bit).

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38).

Interconnection FxCtrl - FxSigBin

The interconnection is carried out through the following connections also required at other adapter blocks:

- 1:n structure connection input `CmData` → unit-specific information signals
- 1:1 connection input `CmCmd` → command structure of FxCtrl
- 1:1 connection output `CmFbk` → feedback structure of FxCtrl

Further information about the interconnection is available in the section FxCtrl – Function-based control and monitoring block (Page 365) / section Interconnection FxCtrl to EQM SFC and lower-level control modules.

Interconnection FxSigBin to the binary elements

Signal groups for control, feedback monitoring, releases for up to 4 elements are available (see Functions of 'FxSigBin' (Page 401)).

Status word allocation 'Status1' status bit	Parameter
0	`BatchOcc`
1	`BatchEn`
2	`False`
3	Higher-level simulation active
4	`False`
...	Reserve

9.2.2 Functions of 'FxSigBin'

The function of the block is explained on the basis of the most important input and output parameters:

- Cmd1CM ... Cmd4CM → Output of the up to 4 independent command bits of the FxCtrl function configuration
- Cmd1Rel ... Cmd4Rel → Process release signals for the command bits (default = enable if not interconnected). These signals are interlinked in FxCtrl with the configured function enable signals.
- Fbk1 ... Fbk4 → Input of feedback or monitoring states to the FxCtrl
  - If these are pure monitoring signals for external errors, the respective Feature-Bits (4 to 7) must be set. They can no longer be used as a combination of control and feedback signals at the same time.
  - If there are no feedback signals from the process, these can be updated using Feature Bits 8 ... 11 of the control.

  Note

  Only the binary elements connected to these inputs are visualized in the faceplate of the FxCtrl block. Free inputs (signal status = 16#FF) are not displayed.

- CmdDesc / FbkDesc → serves to configure the activation / feedback descriptions". These are displayed in the configuration dialog as well as in the faceplate. An instance-specific enumeration with the respective activation commands or the feedbacks of the connected module can be configured here. By default the enumerations CmdSigB / FbkSigB are configured.

Further functions of FxSigBin

In this block the following overwrite functions are implemented with regard to the input selections:

- OvwCommands
  1 = The inputs Cmd1 ... Cmd4 are used instead of the control command CmCmd of the FxCtrl block
- OvwUnitID
  1 = The input UnitID is used instead of the unit ID from CmData of the EqmIf block
- OvwBatchInfo
  1 = The inputs BatchEnLi, BatchOccLi, BatchIdLi, BatchNameLi, StepNoLi are used instead of the batch data from CmData of the EqmIf block

Calling additional faceplates

This block does not itself dispose of this function. However, it makes the inputs SelFpSi1 ... SelFpSi4 for the higher-level FxCtrl block available. These input parameters have the attribute "'k_Jump". In the case of existing CFC interconnections at these inputs the binary elements interconnected here can be opened in the OS faceplate of the FxCtrl , in as far as these have a faceplate (however, the element line only appears at an interconnected Fbk1 ... Fbk4 signal!).

Forming the signal status for blocks

This block includes the APL standard function "Form and output signal status for blocks".

The worst signal status for the block is formed from the following parameters:

`Cmd1 … Cmd4` (depending on the switchover signal `OvwCommands`)

`Fbk1 … Fbk4`

`Cmd1Rel … Cmd4Rel`

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the Feature parameter is available in the *"APL documentation" in the section "Configurable functions with the Feature I/O"*. The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	Reserve
1	Reserve
2	Display of the instance name 1 = CFC name (process tag)
3	Display of the instance name 1 = Block name
4	1 = Fbk1 signals an external error
5	1 = Fbk2 signals an external error
6	1 = Fbk3 signals an external error
7	1 = Fbk4 signals an external error
8	1 = Fbk1 follows Cmd1 if not interconnected
9	1 = Fbk2 follows Cmd2 if not interconnected
10	1 = Fbk3 follows Cmd3 if not interconnected
11	1 = Fbk4 follows Cmd4 if not interconnected
…	…

Operator permissions

This block includes the APL standard function "Operator permissions". The block has the following permissions for the `OS_Perm` parameter:

Status bit	Parameter
0	1= Operator can change delay time for Sig1
1	1= Operator can change delay time for Sig2
2	1= Operator can change delay time for Sig3
3	1= Operator can change delay time for Sig4
…	Reserve

9.2.3 Connections of 'FxSigBin'

Input parameters

Parameter	Description	Type	Default
CmdDesc	Command descriptions; to be parameterized with type specific enumeration 'CmdSigB'	INT	
FbkDesc	Feedback descriptions; to be parameterized with type specific enumeration 'FbkSigIB'	INT	
Cmd1 … Cmd4	Monitor or release signal	DigValFF	
Fbk1 … Fbk4	Command feedback if present (will be simulated if not connected)	DigValFF	
Cmd1Rel … Cmd4Rel	Process release (implicitly true if not connected)	DigValFF	
RstLi	Linkable reset signal	DigValFF	
Cmd1_DC … Cmd4_DC	Command delay	AnaValFF	
CmData	Control and status information for this CM	BlUdtCmData	
CmCmd	Command data from FxCtrl	DWORD	
OvwCommands	1=Overwrite CmCmd commands with inputs Cmd1 … Cmd4	BOOL	True
OvwUnitID	1=Overwrite CmData unit ID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
UnitID	Unit ID (usually from UnitIf block)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIDLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
SampleTime	Sampling time [s]	REAL	1.0
SelFpSi1 … SelFpSi4	Select faceplate for signal <x> (used by FxCtrl)	BOOL	
OS_Perm	Operator permissions	STRUCT_DWORD	
Feature	Feature collection	STRUCT_DWORD	

Output parameters

Parameter	Description	Type	Default
Cmd1CM ... Cmd4CM	Activation signal	DigVal	
ModLiOpCM	1=Link/Auto 0=Manual: Input to auto/manual commands	DigVal	
AutModLiCM	1=Auto mode: auto mode switch via link	DigVal	
ManModLiCM	1=Manual mode: manual mode switch via link	DigVal	
RstLiCM	Reset signal	DigVal	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING [32]	
StepNo	Batch Stepp number	DWORD	
UnitID_CM	Connect to CM's ExtVa104 for unit message filtering	DWORD	
EqmLock	EQM lock signal from CmData	BOOL	
P_RST	Reset pulse	BOOL	
SimOnParent	Higher ranking simulation is on	BOOL	
OS_PermOut	Operator permission: output 1 for OS	DWORD	
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
Status1	Status 1	DWORD	16#FFFFFFFF
St_Worst	Worst signal status	BYTE	
CmFbk	Feedback data for EqmIf / FxCtrl	BlUdtCmFbk	16#80

9.3 FxSigInt – INTEGER - Signal routing block for FxCtrl

9.3.1 Description of 'FxSigInt'

Area of application and operating principle

This function block acts as an adapter for controlling and / or monitoring a control module by means of an integer control value with a range of values 0 ... 64 for an upstream FxCtrl block.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38).

Interconnection FxCtrl - FxSigInt

The interconnection is carried out through the following connections also required at other adapter blocks:

- 1:n structure connection input `CmData` → unit-specific information signals
- 1:1 connection input `CmCmd` → command structure of FxCtrl
- 1:1 connection output `CmFbk` → feedback structure of FxCtrl

Further information about the interconnection is available in the section "FxCtrl – Function-based control and monitoring block (Page 365)" / section Interconnection FxCtrl to EQM SFC and lower-level control modules.

Interconnection FxSigInt to the control module / element

One connection each for controlling, feedback monitoring, enabling is available (see Functions of 'FxSigInt' (Page 406)).

Status word allocation 'Status1'

Status bit	Parameter
0	`BatchOcc`
1	`BatchEn`
2	`False`
3	Higher-level simulation active
4	`False`
...	Reserve

9.3.2 Functions of 'FxSigInt'

Functions of FxSigInt

The function of the block is explained on the basis of the most important input and output parameters.

- `CmdCM` Output of the command byte of the FxCtrl function configuration
- `FbkCM` Input of the feedback at the FxCtrl
- `CmdRestPos` Rest position command
  This command is output when the process enable signal `CmdRel` is missing or the FxCtrl block changes to the function "Inactive".
- `NoFbkCM` 1 = No feedback signal available
- `CmdRel` Process enable signal for the command byte
  (default = Enable if not interconnected). This signal is interlinked in FxCtrl with the configured function enable signal.
- `CmdDesc / FbkDesc` serves to configure the "activation / feedback descriptions". These are displayed in the configuration dialog as well as in the faceplate. An instance-specific enumeration with the respective activation commands or with the feedbacks of the connected module can be configured here. By default the enumerations CmdSigI / FbkSigI are configured.

Further functions of FxSigInt

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwCommands`
  1 = The INT input `Cmd` is used instead of the `CmCmd` control command of the FxCtrl block
- `OvwUnitID`
  1 = The input `UnitID`is used instead of the unit ID from `CmData` of the EqmIf block
- `OvwBatchInfo`
  1 = The inputs `BatchEnLi, BatchOccLi, BatchIdLi, BatchNameLi, StepNoLi` are used instead of the batch data from `CmData` of the EqmIf block

Call additional faceplates

This block does not itself dispose of this function. However, it makes the input `SelFpSig` with the attribute ‚Blk_Jump' available for the higher-level FxCtrl block. If a CFC interconnection is available at this input, the respectively interconnected FxCtrl instance can be opened in the OS faceplate.

Forming the signal status for blocks

This block includes the APL standard function "Form and output signal status for blocks".

The worst signal status for the block is formed from the following parameters:

- Cmd (depending on the switchover signal OvwCommands)
- Fbk
- CmdRel

Configurable reactions using the Feature parameter

An overview of all the reactions that are provided by the Feature parameter is available in the "*APL documentation" in the section "Configurable functions with the Feature I/O"*. The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	Reserve
1	Reserve
2	Display of the instance name 1 = CFC name (process tag)
3	Display of the instance name 1 = Block name
...	...

Operator permissions

The block has the following permissions for the OS_Perm parameter:

Status bit	Parameter
0	1= Operator can change delay time for Cmd
...	Reserve

9.3.3 Connection of 'FxSigInt'

Input parameters

Parameter	Description	Type	Default
CmdDesc	Command descriptions; to be parameterized with type specific enumeration 'CmdSigB'	INT	
FbkDesc	Feedback descriptions; to be parameterized with type specific enumeration 'FbkSigIB'	INT	
Cmd	Monitor or release value (0-64)	INT	
CmdRel	Process release (implicitly true if not connected)	DigValFF	
FbkCM	Command feedback	INT	

Parameter	Description	Type	Default
NoFbkCM	1=No feedback present	BOOL	
RstLi	Linkable reset signal	DigValFF	
Cmd_DC	Command delay	AnaValFF	
CmData	Control and status information for this CM	BlUdtCmData	
CmCmd	Command data from FxCtrl	DWORD	
OvwCommands	1=Overwrite CmCmd commands with inputs OpenAut/ CloseAut	BOOL	True
OvwUnitID	1=Overwrite CmData unit ID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
UnitID	Unit ID (usually from UnitIf block)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIDLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
SampleTime	Sampling time [s]	REAL	1.0
SelFpSig	Select faceplate for signal (used by FxCtrl)	BOOL	
OS_Perm	Operator permissions	STRUCT_DWORD	
Feature	Feature collection	STRUCT_DWORD	

Output parameters

Parameter	Description	Type	Default
CmdCM	Activation signal	DigVal	
ModLiOpCM	1=Link/Auto 0=Manual: Input to auto/manual commands	DigVal	
AutModLiCM	1=Auto mode: auto mode switch via link	DigVal	
ManModLiCM	1=Manual mode: manual mode switch via link	DigVal	
RstLiCM	Reset signal	DigVal	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING [32]	
StepNo	Batch Step number	DWORD	
UnitID_CM	Connect to CM's ExtVa104 for unit message filtering	DWORD	
EqmLock	EQM lock signal from CmData	BOOL	
P_RST	Reset pulse	BOOL	
SimOnParent	Higher ranking simulation is on	BOOL	
OS_PermOut	Operator permission: output 1 for OS	DWORD	
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
Status1	Status 1	DWORD	16#FFFFFFFF

Parameter	Description	Type	Default
St_Worst	Worst signal status	BYTE	
CmFbk	Feedback data for EqmIf/FxCtrl	BlUdtCmFbk	16#80

9.4 FxDef16 / 32 / 64 – Function definition block for FxCtrl

9.4.1 Description of 'FxDef16 / 32 / 64'

Area of application and operating principle

This function block serves to store the function and CM type definitions and makes these available through a structure connection to the downstream FxCtrl block. For such applications where equipment modules are structured with the same CMs or MOFs respectively since they have an identical technical function and can therefore use exactly the same function definitions, this can be realized by means of a 1:n interconnection from one FxDef to several FxCtrl blocks. This block offers the advantage of rapid configuration as well as a lower consumption of resources than individual assignment and configuration.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).

Interconnection

The interconnection is realized through a 1:1 connection `FuncDef` (1:n at several equal EQMs) between FxDef<xx> and FxCtrl block.

See also

FxCtrl – Function-based control and monitoring block (Page 365)

Functions of 'FxDef16 / 32 / 64' (Page 410)

9.4.2 Functions of 'FxDef16 / 32 / 64'

The block contains 16 / 32 / 64 STRING parameters for function definitions as well as 4 STRING parameters for overlay definitions in the input parameter strip. These are configured using the special configuration dialog of the block FxCtrl . The internal structure of the STRING parameters is not relevant for the user. Simple 'copy / paste' operations between the function / overlay definitions of an instance or between block instances can also be carried out in the process object view.

9.4.3 Connections of 'FxDef16 / 32 / 64'

Input parameters

Parameter	Description	Type	Default
FuncDef01 ... <FuncDef16> <FuncDef32> <FuncDef64>	Internal	STRING[192]	
OvlDef01 ... OvlDef04	Internal	STRING[192]	
CmTypeDef	Internal	STRING[64]	
FuncDefHi	Internal	DWORD	
FuncDefLo	Internal	DWORD	

Output parameters

Parameter	Description	Type	Default
FuncDef	Functiondefinition datastructure to be connected to 'FxCtrl'	BlUdtDefData	

9.5 FxShare – Function definition block for FxCtrl

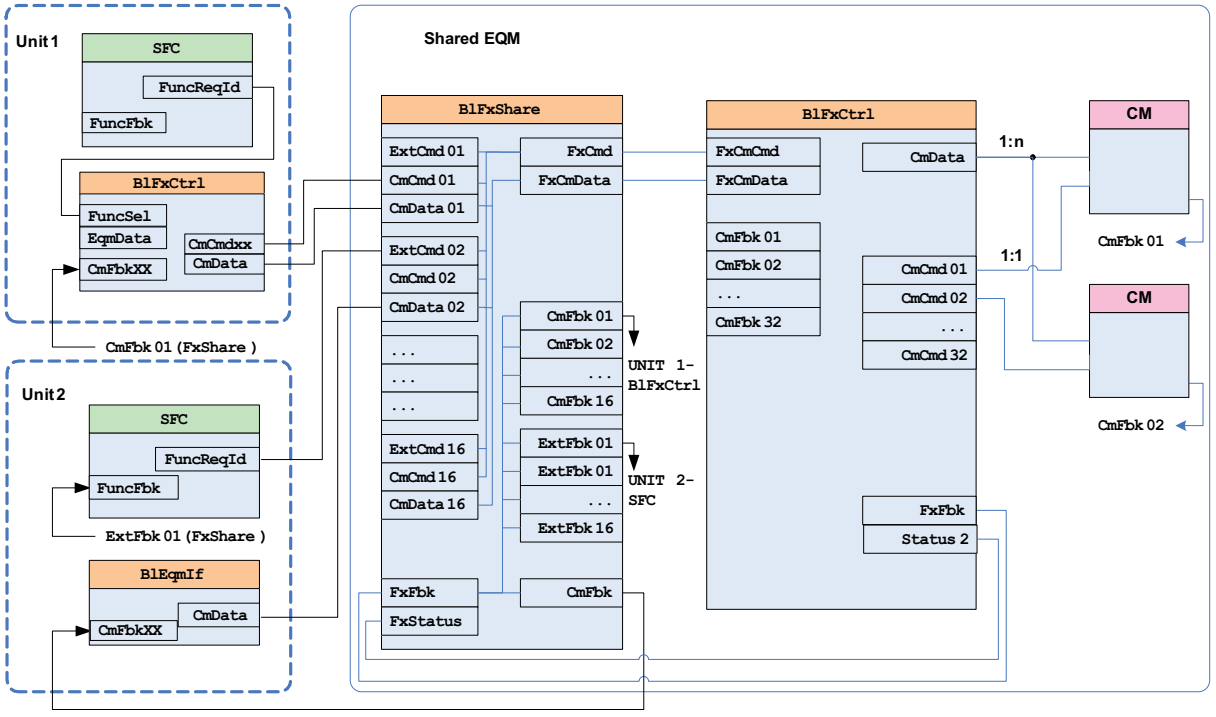
9.5.1 Description of 'FxShare'

Area of application and operating principle

This block offers the possibility for up to 16 units of accessing an equipment module using the FxCtrl block in the context of function-based controlling and monitoring.

The block is controlled optionally by higher-level FxCtrl blocks or EQM SFCs and coordinates multiple accesses to a common equipment module that is controlled in turn by a FxCtrl . The representation of the pending as well as the respectively active request is carried out in a faceplate.

The following figure shows the control scheme for the access of two UNITs to the control modules of an EQMs. UNIT 1 uses controlling through a FxCtrl block in the EQM and UNIT 2 uses the classical SFC control.



Data flows

As a component of the common equipment module the FxShare block provides the following interfaces (interconnection black = higher-level, blue = lower level):

- CmData<xx>
  - Interface to the higher-level level (unit) independently of the type of EQM control system. Here the SIMATIC Batch data, operating modes, resets, simulation operation, enables / locks are exchanged with the FxCtrl .
- ExtCmd<xx> / CmFbk
  - Command and feedback interface to the higher-level EQMs with SFC control. Here controlling is carried out through the specification of a function number of the type INT (0 to 64) that has to be interconnected to one of the inputs ExtCmd1 ... 16 .
- CmCmd<xx> / CmFbk<xx>
  - Command and feedback interface to the higher-level EQMs with FxCtrl control. Here controlling is carried out through the data structure connection CmCmd01 ... 16. In addition to the function number this contains further type information and enabling signals.
- FxCmData / FxCmd / FxFbk / FxStatus
  - Interface to the lower-level Shared EQM-FxCtrl block. The function specified by the unit is activated and its feedbacks evaluated through this interface.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).

The FxShare block is required for each shared equipment module and is instantiated together with the FxCtrl block of the shared EQM in the CFC.

In addition the interconnection to the lower-level FxCtrl block as well as to the higher-level EQM interface blocks (depending on the EQM type, see above) is carried out in the CFC editor.

The FxShare block itself does not need a configuration dialog. The function configuration of both the higher-level and the lower-level FxCtrl instances is carried out in a specific configuration dialog that is installed with the PCS 7 Industry Library on the ES (see the section "Configuration of the function definitions (Page 385)").

In the higher-level FxCtrl configuration window the FxShare block is displayed as an element line with an own type, whereby the selectable control commands are determined by the functions already configured of the lower-level FxCtrl block.

It is therefore advisable to first carry out the configuration of the Shared EQM-FxCtrl block and afterwards the configuration of the higher-level / UNIT -specific FxCtrl instances.

Note

Cascaded operation of several blocks FxCtrl

In the case of a chaining of for example 2 instances of the block in accordance with the following scheme:

- `FxCtrl (Master) → FxCtrl (Slave)`
- `FxCtrl (Master) → FxShare → FxCtrl (Slave)`

take into account that only function controls and not overlay definitions are supported in the lower-level block (slave). In the process only function controls of the lower-level `FxCtrl` are offered in the overlay design of the higher-level `FxCtrl` - no overlays, even if these should be defined. In cascaded `FxCtrl` operation the lack of the overlay transmission to the Slave does not mean a functional limitation because when the Master returns from overlay operation the previous function control is always assumed, both in the Master and in the Slave. A Slave therefore does not actually require an overlay function.

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'FxShare' (Page 417)".

Status bit	Parameter
0	Occupied
1	BatchEn
2	1= FxShare is in the operating mode "On"
3	1= FxShare is in the "Out of service" operating mode
...	Reserve
8	1= Lower-level FxCtrl is in the operating mode "On"
9	1= Lower-level FxCtrl is in the "Out of service" operating mode
10	1= Lower-level FxCtrl has a "group error"
11	1= At least 1 element in the lower-level FxCtrl is in "forced operation"
12	`OosLi.Value`

9.5.2 Functions of 'FxShare'

9.5.2.1 Connection possibilities

At the input side the FxShare block supports up to 16 function requests (units / EQMs). At the output side an equipment module is supported in the form of a FxCtrl block that controls the assigned elements (control modules).

Function requests to the FxShare can be made by two methods as shown above:

- By a FxCtrl block
  - Relevant connections in this case are the data structures
  - CmData01 ... CmData16 - Info structure of FxCtrl
  - CmCmd01 ... CmCmd16 of FxCtrl for the function specification
  - CmFbk01 ... CmFbk16 at FxCtrl for the feedback of function numbers, flags, group errors)
  - In the faceplate view of the higher-level FxCtrl block the shared equipment module is displayed as an element line with an allocation feedback, the "Function" activation, the activation value in the form of the function enumeration (is configured at Shared-FxCtrl) as well as the function currently being executed at the lower-level Ctrl .
  - Since only blocks of the Industry Library are used in this variation and their correlations are clear, the function designations of the lower-level FxCtrl are available directly while configuring the higher-level FxCtrl.
- By a standard EQM SFC with EqmIf block
  - Relevant connections here are:
    CmData01 ... CmData16 - Info structure of EqmIf
    ExtCmd01 ... ExtCmd16 of EQM SFC for the function specification
    ExtFbk01 ... ExtFbk01 to the EQM-SFC for the feedback of function numbers, "Allocation OK" flags, group errors)
  - In the faceplate view of the EqmIf block the shared equipment module is displayed as an element line with an allocation feedback
  - The connections for connecting to the EQM SFC have the attribute "S7_contact".

Note

Mixed usage of the function requirements

The up to 16 function requests / feedbacks can only be used once with reference to the connection index. This means for an index "x" either the connection pair CmCmd<x>/ CmFbk<x> or alternatively ExtCmd<x> / ExtFbk<x> may be connected.

9.5.2.2 Allocation request and control of the EQM

The request of the shared equipment module is initiated through the specification of a valid function number at one of the inputs CmCmd<xx> bzw. ExtCmd<xx> . If there are several requests in the same cycle, the numerically lowest connection number 'wins'.

The unit-related information and operating mode specifications are read by the CmData<xx> structure inputs.

Both the command or function identifier as well as the unit data of the respectively active request are passed through to the outputs. Controlling remains at this unit until the function identifier Inactive (=0) is requested.

With a successful allocation the FxShare takes over establishing the initial state (automatic mode, resetting, deactivation of the external simulation) by activating the corresponding bits in

the CmData structure. These automatisms can be enabled or locked analog to the UnitIf block by means of Feature bits.

9.5.2.3 APL standard functions of 'FxShare'

Calling additional faceplates

This block includes the function "Call additional faceplates". To this purpose the input parameters CmData01 ... CmData16 as well as 'CmFbk ' have the attribute k_Jump. If CFC interconnections are available at these inputs, the higher-level FxCtrl or EqmIf blocks as well as the lower-level FxCtrl instance can be opened in the OS faceplate.

Configurable reactions using the Feature parameter

An overview of all the reactions that are provided by the Feature parameter is available in the *"APL documentation" in the section "Configurable functions with the Feature I/O"*. The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	
1	See APL function: "Behavior for 'Out of service' mode" Release for switchover to "Out of service" via OosLi (default = 0)
2	Display of the CM instance name *) 1 = CFC name (process tag) as per preset value
3	Display of the CM instance name *) 1 = Block name
4	1 = At unit allocation: Switch all the lower-level elements (CMs, SFCs) to AUTO mode
5	1 = At unit allocation: Deactivate the unit simulation
6	1 = At unit allocation: Reset all the lower-level CMs
...	...
24	See APL function: "Enable local operator permission" 1 = "Local" operator permission is active
...	...

Operator permissions

This block includes the APL standard function "Operator permissions". The block has the following authorizations for the OS_Perm parameter:

Bit	Parameter
0	1 = Operator can switch to "On" mode
...	Reserve

Overview of error numbers

The following error numbers are output using the connection `ErrorNum` :

No.	Description of errors	Remarks
-1	Default value when implementing the block; block will not be processed	
0	There is no error.	
1	Control error: external Invalid function identifier	The number range between 0 and 64 is valid for function requests.
2	Parameter assignment error: Duplicate allocation at command input parameter	The connections `CmCmd<xx>` and `ExtCmd<xx>` with the same index are allocated simultaneously

The associated connection index (1 to 16) of the error identified through `ErrNum` is output additionally through the connection `ErrAddInfo` .

9.5.3 Connections of 'FxShare'

Input parameters

Parameter	Description	Type	Default
`MsgLock`	Lock CMs' messages (default if not owned)	`BOOL`	
`ManOpEn`	Enable CMs' manual mode switch for operator (default if not owned)	`BOOL`	True
`SimEnCM`	Enable CMs' higher ranking simulation (default if not owned)	`BOOL`	True
`SimOnCM`	Activate CMs' higher ranking simulation (default if not owned)	`BOOL`	

`CmData01`
...
`CmData16`
	Control and status information	`BlUdtCmData`	

`CmCmd01`
...
`CmCmd16`
	Command data from FxCtrl	`BlUdtCmCmd`	

`ExtCmd01`
...
`ExtCmd16`
	Function request from SFC	`INT`	`"EXT_CMD_IDLE"`

`ExtOvl01`
...
`ExtOvl16`
	Overlay function request from SFC	`INT`	`"EXT_CMD_IDLE"`

`MS_RelOp`	Operator input for MS Release, 1: MS release requirement	`BOOL`	
`OnOp`	1=On mode: On mode by operator	`BOOL`	
`OosOp`	1=Oos mode: Oos mode by operator	`BOOL`	

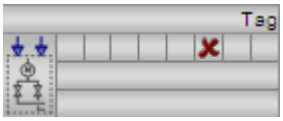
Parameter	Description	Type	Default
OosLi	1=Oos mode: Oos mode switch via link	DigVal	
OS_PERM	Operator Permissions	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	
FxFbk	Feedback from FxCtrl	BlUdtCmFbk	
FxStatus2	Status2f rom FxCtrl	DWORD	

Output parameters

Parameter	Description	Type	Default
MS_Release	1: Maintenance status release	BOOL	
OnAct	On mode is active	DigVal	
OosAct	Out of service is active	DigVal	
OvlSel	Overlay function for FxCtrl	INT	
FxCmData	CmData for FxCtrl	BlUdtCmData	
FxCmd	CmCmd for FxCtrl	BlUdtCmCmd	
CurOwner	Current shared equipment's owner	INT	
CmFbk01 ... CmFbk16	Feedback for controlling FxCtrl	BlUdtCmFbk	
ExtFbk01 ... ExtFbk16	Assignment feedback for controlling SFC	BlUdtFxShExtFbk	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING[32]	
StepNo	Step number from EPH	DWORD	
OSElRtd01 ... OSElRtd16	Internal	WORD	
OSElInUse	Internal	WORD	
CmFbk	This module's feedback for UnitIf	BlUdtCmFbk	
OS_PermOut	Operator permissions (OS info)	DWORD	16#FFFFFFFF
OS_PermLog	Operator permissions (OS info)	DWORD	16#FFFFFFFF
OpSt_Out	Enabled operator stations	DWORD	
Status1	Status1 data	DWORD	
ST_WORST	Worst signal status	BYTE	16#80
ErrorNum	Error number	INT	
ErrAddInfo	Additional error information	INT	

9.5.4 Operator control and monitoring

9.5.4.1 Symbol

	Tag name Status bar Block instance name Active function (from enumeration)

Since the faceplate is generally called form other faceplates (EqmIf, FxCtrl), this symbol is only required optionally.

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	FxShare block operating mode	
		On
		Out of service
2	Empty	
3	Lower-level FxCtrl block operating mode	
		On
		Out of service
4		Lower-level FxCtrl block has a group error
5		One / several CMs are in 'force mode"
6		Unit not released
		Unit released
7		Unit allocated
8		Notice is available

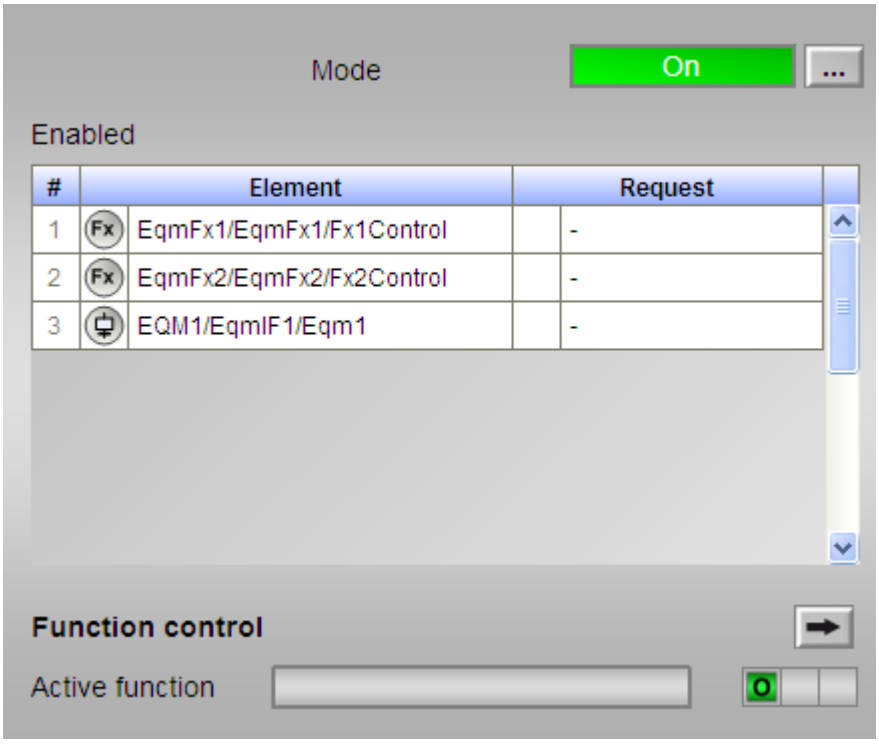
9.5.4.2 Faceplate

Views of the FxShare block

The block FxShare provides the following views:

	Standard view	See below.
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

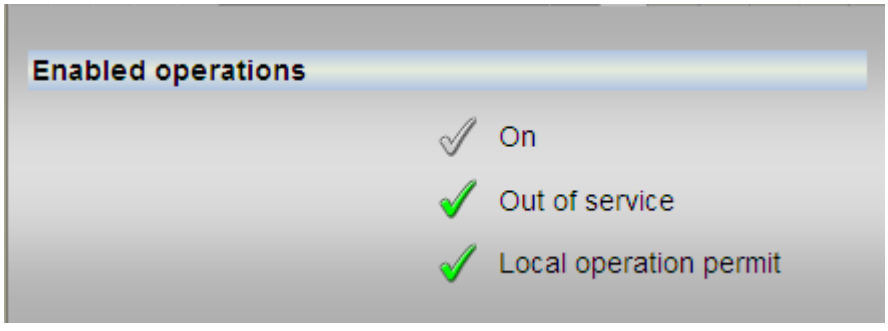
Standard view of FxShare



Explanation of the screen shot	
(1)	**Displaying and switching the operating mode**
	This area provides information on the currently valid operating mode. The following operating modes can be shown and executed here: • On • Out of service

Explanation of the screen shot		
(2)	**Function control**	
	This sections shows you the status of the lower level equipment module (Blis sections shows you the status of the lower level equipment module (FxCtrl block): Active function → requested function or overlay definition	
	• The 3 fields at the outer right contain information about the monitoring mode:	
		On
		Out of service
		Group error
		Forced operation for at least one element
	The → button can be used to call the Bl button can be used to call the FxCtrl faceplate of the shared equipment module.	
(3)	**Allocation state**	
	This table lists the higher-level elements with the request status. • The Element column displays optionally the CFC name of the *process tag*, the block instance name 'FxCtrl' or both together (selection by means of Feature Bits). • The Request column contains the currently controlled function definition of this element in accordance with the configuration of the functional or overlay function. Further attributes of the control are represented by the following symbols:	
		Function request fulfilled
		Function request active / unfulfilled

Preview of FxShare



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the `OS_PERM` parameter of this block. The operator control enables shown in the screenshot are possible here. • Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions

HVAC blocks 10

10.1 CalcWatP - Calculation of thermal power and emitted energy

10.1.1 CalcWatP description

Object name (type + number) and family

Type + number: FB 1144

Family: HVAC

Number of process objects: 1 PO

Operating principle of CalcWatP

The CalcWatP block uses the volume flow rate and the water supply and return temperatures to determine the current emitted thermal power and the total emitted thermal energy of the water.

As input parameters, the block requires the current volume flow rate `VolFlowRat` [m³/h], the supply and return temperatures `TempFor`, `TempRet` in [°C], and information concerning whether the volume is measured in the supply (`VolMeasFor` = TRUE) or in the return (`VolMeasFor` = FALSE).

The `FactPower` and `FactEnergy` inputs can be used to change the unit of the `MinPower`, `Power_Out` or `Energy_Out`, `Energy2Out` and `StartEnergy` parameters. By default, the power is indicated in kW and the energy in kWh. (`FactPower` = 1, `FactEnergy` = 1). If `FactPower` is set to a value of 10000, for example, the resulting power unit will be `FactPower` * kW = 1000 * kW = 1 MW. If a factor has a value of 0, the value 1 is set in the program.

On the basis of the supply and return temperatures, a thermal coefficient [kWh/(m³/h/°C)] is determined from a table that is stored within the block. The table provides the thermal coefficients for supply and return temperatures ranging from 10°C to 140°C in steps of 10°C. The block calculates intermediate values by performing linear interpolation. Values that fall outside this range are extrapolated linearly.

The current thermal power is provided at the `Power_Out` output in the unit [`FactPower` * kW]. The total emitted thermal energy is provided at the `Energy_Out` and `Energy2Out` output in the unit [`FactEnergy` * kWh]. If the amount of thermal power is below the value of `MinPower`, a thermal power of zero is used. The current thermal power is always output as a positive value. In case of heating (supply temperature > return temperature) the `Heating` output is set to TRUE, in case of cooling it is set to FALSE.

A positive edge at the `RstEnergyLi` or `RstEnergyOp` input resets the value of `Energy_Out` to zero. A positive edge at the `RstEnergy2Li` or `RstEnergy2Op` input resets the value of `Energy2Out` to zero. In the event of a CPU restart (in OB100), the value of `StartEnergy` (Start Thermal Energy) is used as the start value for calculating the thermal energy `Energy_Out` and `Energy2Out`.

Calculation of the thermal power:

- `Power_Out` = ABS(`VolFlowRat` * KT * (`TempFor – TempRet`))

Calculation of the emitted thermal energy (iteratively at the i-th block call after `RstEnergryLi`/`RstEnergy2Li` or `RstEnergyOp`/`RstEnergy2Op`, or a CPU restart):

- `Energy_Out`[i] = `Energy_Out`[i-1] + `Power_Out` * (`SampleTime`/3600.0)
- `Energy2Out`[i] = `Energy2Out`[i-1] + `Power_Out` * (`SampleTime`/3600.0)

Startup characteristics

Following startup, the messages are suppressed for the number of cycles assigned in the `RunUpCyc` parameter. A startup can be simulated via the `Restart` input.

Time response

The block must be called by means of a cyclic interrupt OB. The sampling time of the block is entered in the SampleTime parameter.

Called blocks

FC369	SelST16
SFC6	RD_SINFO
SFB35	ALARM_8P

Calling OBs

In the same OB with and after the block whose measured value is to be monitored. Also in OB100.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section CalcWatP I/Os (Page 428).

Status bit	Parameter
0	Occupied
1	BatchEn
2	SimAct
3	OosAct.Value
4	OosLi.Value
5	Not used
6	OnAct.Value
7	Not used
8	VolMeasFor
9	NOT VolMeasFor
10	Heat

Status bit	Parameter
11	Do NOT heat
12 - 29	Not used
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section CalcWatP I/Os (Page 428).

Status bit	Parameter
0	MsgLock
1	TempForCSF
2	TempRetCSF
3	VolFlowCSF
4 - 30	Not used
31	MS_RelOp

10.1.2 CalcWatP operating modes

The block does not have any operating modes.

10.1.3 CalcWatP functions

Quality code

The .ST structure elements of the parameters for the measured values `VolFlowRat`, `TempFor` and `TempRet`plus the external message signals contain the quality code.

The following quality code data is evaluated:

- Quality Code = 16#60: Simulation on driver block active
- Quality Code = 16#80: Valid value
- Quality Code <> 16#60 oder <> 16#80: Invalid value

Release for maintenance

This block features the standard function "Release for maintenance".

Simulating signals

This block features the standard function "Simulate signals".

Interconnection of limits

The `Power_Out` limit can be interconnected. If it is not interconnected (`Power_Out.ST = 16#FF`), the operator can change the corresponding limit.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4 - 6	Not used
7	1 = Operator can reset accumulated values (Energy 1)
8	1 = Operator can reset accumulated values (Energy 2)
9 - 10	Not used
11	1 = Operator can enable the Simulation function
12	1 = Operator can activate the Release for maintenance function
13 - 15	Not used
16	1 = Operator can change the power factor
17	1 = Operator can change the energy factor
18	1 = Operator can change the minimum power
19 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

The following modes of behavior are available for this block at the relevant bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"

10.1.4 CalcWatP error handling

The input for control system fault `CSF` can be interconnected with an external error output. The error is not evaluated. A message is generated at the ALARM_8P block.

The current value is always used. If the last valid value or a substitute value is to be used, this must be set at the driver block.

10.1.5 CalcWatP messaging

Message behavior

The CalcWatP block uses the ALARM8_P block for generating messages. Message triggers are:

- The `CSF` signals of the measured values linked to the quality code (`CSF_Out`) and the freely available inputs `ExtMsg1` to `ExtMsg4`, which are obtained as control system faults by means of interconnection.

The messages can be suppressed centrally via the faceplate or via `MsgLock` (program).

The free alarm inputs can be interconnected via the parameters `ExtMsg1` to `ExtMsg4`.

The associated values (`ExtVaxy`) of the message block can be assigned freely.

`MsgSup` is set if the `RunUpCyc` cycles have not finished running since the restart, `MsgLock = TRUE` or `MsgStat = 21`.

Process messages

Message instance	Message ID	Message class	Event
MsgEvId	SIG 1	AS control system fault	$$BlockComment$$ External error has occurred
	SIG 2	AS control system fault	$$BlockComment$$ External error has occurred
	SIG 3	AS control system fault	$$BlockComment$$ External error has occurred
	SIG 4	<deactivated>	$$BlockComment$$ Simulation active
	SIG 5	AS control system fault	$$BlockComment$$ External message 1
	SIG 6	AS control system fault	$$BlockComment$$ External message 2
	SIG 7	AS control system fault	$$BlockComment$$ External message 3
	SIG 8	AS control system fault	$$BlockComment$$ External message 4

Associated values for the message instance MsgEvId

Associated value	Block parameters
1	BatchName
2	StepNo
3	BatchID
4	ExtVal04
5	ExtVal05
6	ExtVal06
7	ExtVal07
8	ExtVal08
9	ExtVal09
10	ExtVal10

10.1.6 CalcWatP I/Os

Input parameters

Parameter	Description	Data type	Default
Energy_Unit	Unit of Energy_Out	INT	1179
ExtVal04 ... ExtVal10	External value 4 to 10	ANY	
FactEnergy	Factor for units of energy	REAL	1.0
FactPower	Factor for units of power	REAL	1.0
MinPower	Minimum power for QTH_POW [F_POWER*kW]	REAL	0.0
MsgEvId	Message ID	DWORD	16#0000000F
Power_OpScale	Power trend display limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • 50000.0 • 0.0
Power_Unit	Unit of Power_Out	INT	1190
Restart	Manual restart	BOOL	0
RstEnergy2Li	Linkable Input Reset Energy2Out	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
RstEnergy2Op	Operator Input Reset Energy2Out	BOOL	inactive
RstEnergyLi	Linkable Input Reset Energy_Out	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
RstEnergyOp	Operator Input Reset Energy_Out	BOOL	inactive
SimTempFor	Simulation TempFor	REAL	0.0

Parameter	Description	Data type	Default
SimTempFor_Li	Linkable simulation value TempFor	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
SimTempRet	Simulation TempRet	REAL	0.0
SimTempRet_Li	Linkable simulation value TempRet	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
SimVolFlowRat	Simulation VolFlowRat	REAL	0.0
SimVolFlowRat_Li	Linkable simulation value VolFlowRat	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
StartEnergy	Start Value for Energy_Out when restarting the CPU	REAL	0.0
TempFor	forerun temperature [°C]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
TempFor_Unit	Unit of TempFor	INT	1001
TempForCSF	forerun temperature (TempFor) faulty	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
TempRet	return temperature [°C]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
TempRet_Unit	Unit of TempRet	INT	1001
TempRetCSF	return temperature (TempRet) faulty	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
VolFlowCSF	Volume flow rate (VolFlowRat) faulty	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
VolFlowRat	Volume flow rate [m^3/h]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
VolFlowRat_Unit	Unit of VolFlowRat	INT	1349
VolMeasFor	Location of volume measurement (1=forerun)	BOOL	forerun

Output parameters

Parameter	Description	Data type	Default
Energy2Out	Thermal energy 2 [FactEnergy *kWh]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
Energy_Out	Thermal energy 1 [FactEnergy *kWh]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
ExtMsg1Out … ExtMsg4Out	Message input 1 ... 4 - output	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
GrErr	Group error	BOOL	0
Heating	0=cooling mode 1=heating mode	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
MsgAckn	Message acknowledge	WORD	16#0000
MsgErr	Message error	BOOL	0
MsgStat	Message status	WORD	16#0000
MsgSup	Message suppressed	BOOL	0
Power_Out	Thermal Power [FactPower *kW]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
SimAct	1=Simulation active	BOOL	0
TempFor_Out	forerun temperature [°C]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
TempForCSF_Out	forerun temperature (TempFor) faulty	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
TempRet_Out	return temperature [°C]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
TempRetCSF_Out	return temperature (TempRet) faulty	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
VolFlowCSF_Out	Volume flow rate (VolFlowRat) faulty	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
VolFlowRat_Out	Volume flow rate [m³/h]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80

10.1.7 CalcWatP block diagram

The block has no block diagram.

10.1.8 Operator control and monitoring

10.1.8.1 CalcWatP views

CalcWatP block views

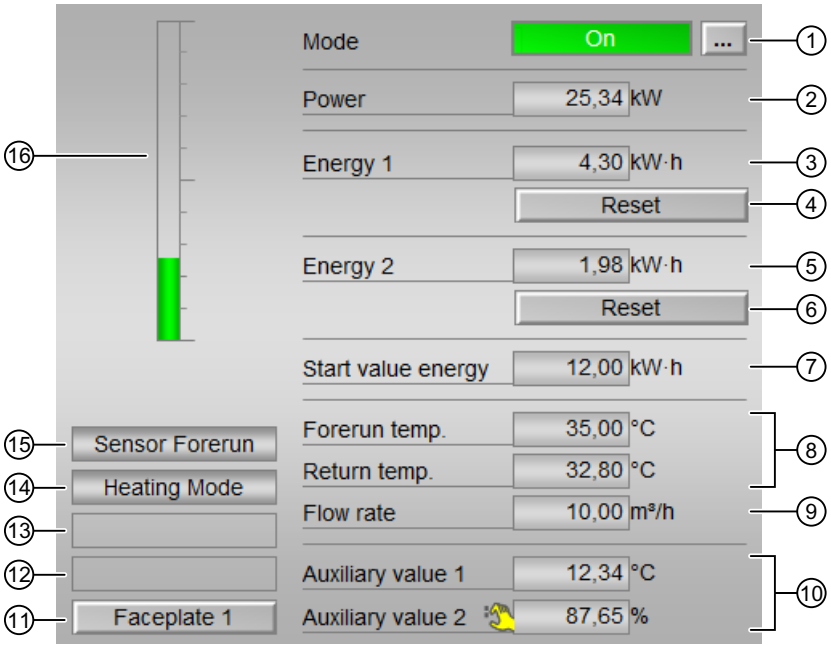
The block provides the following views:

- CalcWatP standard view (Page 432)
- Message view
- Trend view
- CalcWatP parameter view (Page 435)
- CalcWatP preview (Page 436)
- Memo view
- Batch view
- CalcWatP block icons (Page 437)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

10.1.8.2 CalcWatP standard view

CalcWatP standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

For more information on changing the operating mode, refer to the "Switching operating states and operating modes" section in the APL manual.

(2) Display of the current power including signal status

This area shows the current flow rate with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "XX" parameter.

(3) Display of the start value of the energy including signal status

This area shows the current flow rate with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "VolFlowRat" parameter.

(4) Display of the first counter of the thermal energy

This area shows the current count with signal status.

You can change the text for the counter at the "XX" parameter.

(5) Reset the first counter

Use these buttons to reset the thermal energy value of the first counter.

(6) Display of the second counter of the thermal energy

This area shows the current count with signal status.

You can change the text for the counter at the "XX" parameter.

(7) Reset the second counter

Use these buttons to reset the thermal energy value of the second counter.

(8) Display of the pre-processing and post-processing temperatures including signal status

This area shows the current pre-processing and post-processing temperature with the associated signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "TempFor_Out" and "TempRet_Out" parameters.

(9) Display of the flow rate including signal status

This area shows the current flow rate with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the "VolFlowRat" parameter.

(10) Display of auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system.

For more information, refer to the section "Displaying auxiliary values" in the APL manual.

(11) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section 'Opening additional faceplates' in the APL manual

(12) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

Additional information is available in the 'Maintenance enable' section in the APL manual.

(13) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

For more information, refer to the 'Simulating signals' section in the APL manual

(14) Display area for block states

This area provides additional information on the operating state of the block:

- "Cooling mode"
- "Heating mode"

(15) Display area for block states

This area provides additional information on the operating state of the block:

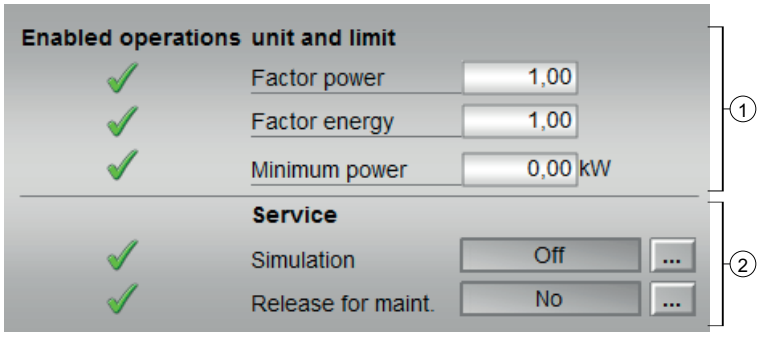
- Sensor Forerun
- Sensor Follow-up

(16) Bar graph display

This area shows the current power value in the form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

10.1.8.3 CalcWatP parameter view

CalcWatP parameter view



(1) Factor and limit

In this area, you change parameters and thereby influence the block. See the 'Changing values' section in the APL manual for more on this.

You can influence the following parameters:

- "Factor for power"
- "Factor for energy"
- "Minimum power"

(2) Service

You can select the following functions in this area:

- "Simulation"
- "Release for maintenance"

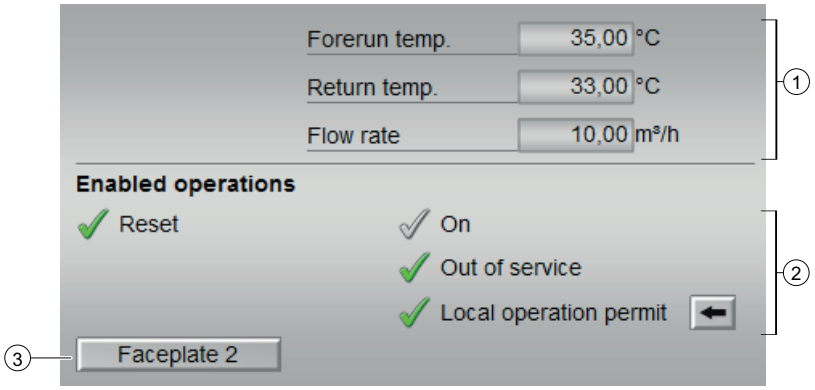
Refer to the "Switchover of operating state and operating mode" section in the APL manual.

You can find information on this area in the APL manual in the section:

- Simulating signals
- Release for maintenance

10.1.8.4 CalcWatP preview

CalcWatP preview



(1) Preview area

The process values for:

- Forward temperature
- Follow-up temperature
- Flow rate

are displayed in this area.

(2) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- Reset: You can reset the peak value of the gradient.
- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(3) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

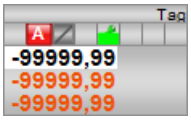
10.1.8.5 CalcWatP block icons

CalcWatP block icons

A variety of block icons are available with the following functions:

- Process tag type
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes
- Signal status, release for maintenance
- Memo display
- Process values

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	
	3	
	4	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

10.2 HxFct - Calculation of enthalpy, absolute humidity, and saturated humidity according to Mollier

10.2.1 HxFct description

Object name (type + number) and family

Type + number: FB 1147

Family: HVAC

Number of process objects: 1 PO

Area of application

The HxFct block uses the relative humidity, the temperature and the atmospheric pressure to determine the absolute humidity, the enthalpy, and the saturated humidity.

As input parameters, the block requires the relative humidity `RelHum` [%], the temperature `Temp` [°C] and the atmospheric pressure `AtmPress` [mbar] [hPa].

The determined enthalpy is provided in the unit [kJ/kg] at the `Enthalpy_Out` output. The absolute humidity is provided in the unit [g/kg] at the `AbsHum_Out` output. The saturated humidity is provided in the unit [g/kg] at the `SatHum_Out` output.

If one of the inputs for the relative humidity or the temperature fails, the relevant input (`RelHumCSF` or `TempCSF`) has to be set. If one of the values is faulty, the last valid value is used and the `GrErr` output is set.

Startup characteristics

Following startup, the messages are suppressed for the number of cycles assigned in the `RunUpCyc` parameter. A startup can be simulated via the `Restart` input.

Time response

The block must be called by means of a cyclic interrupt OB. The sampling time of the block is entered in the `SampleTime` parameter.

Called blocks

FC369	SelST16
SFC6	RD_SINFO
SFB35	ALARM_8P

Calling OBs

In the same OB with and after the block whose measured value is to be monitored. Also in OB100.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section HxFct I/Os (Page 444).

Status bit	Parameter
0	Allocated
1	`BatchEn`
2	`SimAct`
3	`OosAct.Value`
4	`OosLi.Value`
5	Not used
6	`OnAct.Value`
7	Not used
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word assignment for the parameter `Status2`

For a description of the individual parameters, refer to the section HxFct I/Os (Page 444).

Status bit	Parameter
0	`MsgLock`
1	`Tempts`
2	`RelHumCSF`
3 - 30	Not used
31	`MS_RelOp`

10.2.2 HxFct operating modes

The block does not have any operating modes.

10.2.3 HxFct functions

Wet bulb temperature

The block uses the relative humidity [%] and temperature [°C] to calculate the wet bulb temperature in the physical unit [°C] or [°F]. The wet bulb temperature is provided at the `TempWeBu_Out` output.

$$t_{FK} = -5{,}809 + 0{,}058 * \varphi + 0{,}697 * t_{Luft} + 0{,}003 * \varphi * t_{Luft}$$

Figure 10-1 Formula_Wet_Bulb

t_{FK}= Wet bulb temperature (°C)

φ= Relative humidity (%)

t_{Air}= Air temperature (°C)

Quality code

The .ST structure elements of the parameters for the measured values `Temp` and `RelHum` plus the external message signals contain the quality code.

The following quality code data is evaluated:

- Quality Code = 16#60: Simulation on driver block active
- Quality Code = 16#80: Valid value
- Quality Code <> 16#60 oder <> 16#80: Invalid value

Release for maintenance

This block features the standard function "Release for maintenance".

Simulating signals

This block features the standard function "Simulate signals".

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode

Bit	Function
4 - 6	Not used
7	1 = Operator can reset accumulated values
8 - 10	Not used
11	1 = Operator can enable the Simulation function
12	1 = Operator can activate the Release for maintenance function
13 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

The following modes of behavior are available for this block at the relevant bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
19	See APL function: "Changes signal status of outputs in OOS"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"

10.2.4 HxFct error handling

The input for control system fault `CSF` can be interconnected with an external error output. The error is not evaluated. A message is generated at the ALARM_8P block.

The current value is always used. If the last valid value or a substitute value is to be used, this must be set at the driver block.

10.2.5 HxFct messaging

Message behavior

The HxFct block uses the ALARM8_P block for generating messages. Message triggers are:

- The CSF signals of the measured values linked to the quality code (`CSF_Out`) and the freely available inputs `ExtMsg1` to `ExtMsg5`, which are obtained as control system faults by means of interconnection.

The messages can be suppressed centrally via the faceplate or via `MsgLock` (program).

The free alarm inputs can be interconnected via the parameters `ExtMsg1` to `ExtMsg5`.

The associated values (ExtVaxy) of the message block can be assigned freely.

`MsgSup` is set if the `RunUpCyc` cycles have not finished running since the restart, `MsgLock = TRUE` or `MsgStat = 21`.

Process messages

Message instance	Message ID	Message class	Event
MsgEvId1	SIG 1	AS process control message - fault	$$BlockComment$$ External error for temperature has occurred
	SIG 2	AS process control message - fault	$$BlockComment$$ External error for relative humidity has occurred
	SIG 3	< No message >	$$BlockComment$$ Simulation active
	SIG 4	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 5	AS process control message - fault	$$BlockComment$$ External message 2
	SIG 6	AS process control message - fault	$$BlockComment$$ External message 3
	SIG 7	AS process control message - fault	$$BlockComment$$ External message 4
	SIG 8	AS process control message - fault	$$BlockComment$$ External message 5

Associated values for the message instance MsgEvId

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVal04`
5	`ExtVal05`
6	`ExtVal06`
7	`ExtVal07`
8	`ExtVal08`
9	`ExtVal09`
10	`ExtVal10`

10.2.6 HxFct I/Os

Input parameters

Parameter	Description	Data type	Default
AbsHum_OpScale	AbsHum trend display limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • HIGH: 65.0 • LOW: 0.0
AbsHum_Unit	Unit of AbsHum_Out	INT	0
AtmPress	Atmospheric pressure for calculating saturation humidity [mbar] [hPa]	REAL	1013.0
AtmPress_Unit	Unit of atmospheric pressure	INT	1.136
C1NegTemp	C1 constant in case of negative temperature	REAL	175.043
C1PosTemp	C1 constant in case of positive temperature	REAL	224.433
C2NegTemp	C2 constant in case of negative temperature	REAL	241.175
C2PosTemp	C2 constant in case of positive temperature	REAL	272.186
C3	C3 constant	REAL	611.213
Enthalpy_OpScale	Enthalpy trend display limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • HIGH: 200.0 • LOW: 0.0
Enthalpy_Unit	Unit of Enthalpy_Out	INT	1208
MsgEvId	Message ID	DWORD	16#0000000E
RelHum	Relative humidity [%]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
RelHum_OpScale	RelHum trend display limits for OS	STRUCT • HIGH: REAL • LOW: REAL	 • HIGH: 100.0 • LOW: 0.0
RelHum_Unit	Unit of RelHum	INT	1342
RelHumCSF	Relative humidity (RelHum) faulty	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Restart	Manual restart	BOOL	0
SatHum_OpScale	SatHum trend display limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • HIGH: 65.0 • LOW: 0.0
SatHum_Unit	Unit of SatHum_Out	INT	0
SimRelHum	Simulation RelHum	REAL	0.0

Parameter	Description	Data type	Default
SimRelHum_Li	Linkable simulation value RelHum	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
SimTemp	Simulation Temp	REAL	0.0
SimTemp_Li	Linkable simulation value Temp	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
Temp	Temperature [°C]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
Temp_OpScale	Temp trend display limits for OS	STRUCT • HIGH: REAL • LOW: REAL	- • HIGH: 50.0 • LOW: 0.0
Temp_Unit	Unit of Temp	INT	1001
TempCSF	Temperature (Temp) faulty	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
AbsHum_Out	Absolute humidity [g/kg]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
Enthalpy_Out	Enthalpy [kJ/kg]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
GrErr	Group error	BOOL	0
MsgAckn	Message acknowledge	WORD	16#0000
MsgErr	Message error	BOOL	0
MsgStat	Message status	WORD	16#0000
MsgSup	Message suppressed	BOOL	0
RelHum_Out	Relative humidity [%]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
RelHumCSF_Out	Relative humidity (RelHum) faulty	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
SatHum_Out	Humidity at saturation [g/kg]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
SimAct	1=Simulation active	BOOL	0
Temp_Out	Temperature [°C]	STRUCT • Value: REAL • ST: BYTE	- • 0 • 16#80
TempCSF_Out	Temperature (Temp) faulty	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
TempWeBu_Out	Wet bulb temperature [°C]	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80

10.2.7 HxFct block diagram

The block has no block diagram.

10.2.8 Operator control and monitoring

10.2.8.1 HxFct views

HxFct block views

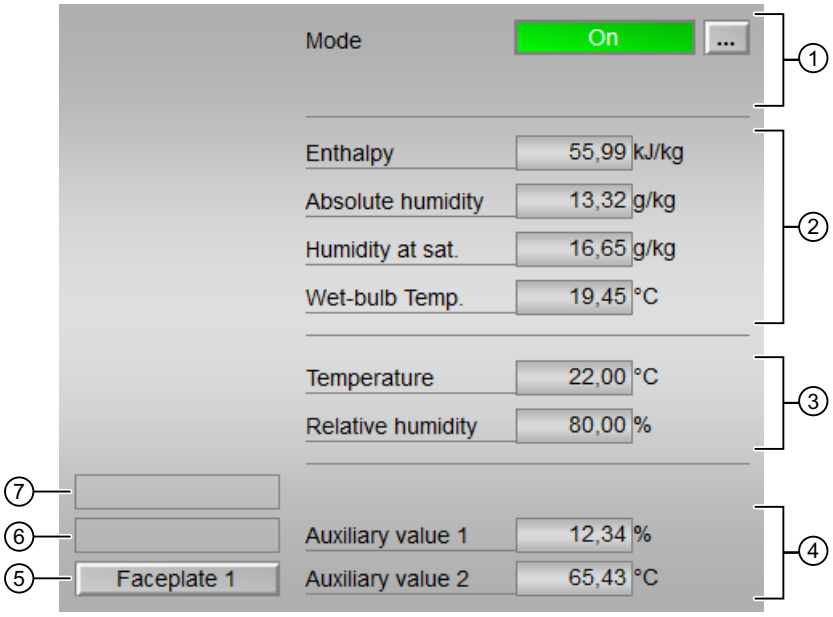
The block provides the following views:

- HxFct standard view (Page 447)
- Message view
- Trend view
- HxFct parameter view (Page 448)
- HxFct preview (Page 449)
- Memo view
- Batch view
- HxFct block icons (Page 450)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

10.2.8.2 HxFct standard view

HxFct standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(2) Determined values

- Enthalpy in kJ/kg
- Absolute humidity in g/kg
- Saturated humidity in g/kg
- Wet bulb temperature in °C

(3) Display of the process values including signal status

This area shows the current process values with the corresponding signal status.

- Temperature in °C
- Relative humidity in %

(4) Display of auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system. You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(5) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(6) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

(7) Display area for block states

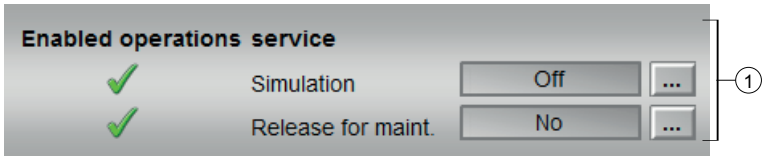
This area provides additional information on the operating state of the block:

- "Simulation"

You can find additional information on this in section "Simulating signals" in the manual of the APL.

10.2.8.3 HxFct parameter view

HxFct parameter view



(1) Service and operator control enable

You can select the following functions in this area:

- "Simulation"
- "Release for maintenance"

Refer to the "Switchover of operating state and operating mode" section in the APL manual.

You can find information on this area in the APL manual in the section:

- Simulating signals
- Release for maintenance

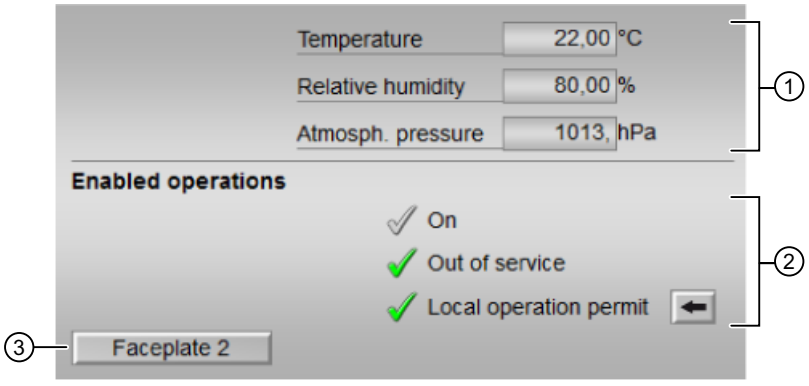
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator is generally prohibited from accessing this parameter due to the configured AS operator control permissions (OS_Perm or OS1Perm).

10.2.8.4 HxFct preview

HxFct preview



(1) Preview area

The process values for:

- Temperature
- Relative humidity
- Atmospheric pressure

are displayed in this area.

(2) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- Reset: You can reset the peak value of the gradient.
- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(3) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

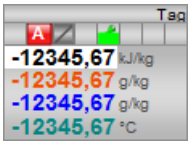
10.2.8.5 HxFct block icons

HxFct block icons

A variety of block icons are available with the following functions:

- Process tag type
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes
- Signal status, release for maintenance
- Memo display
- Process values

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	
	3	
	4	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

10.3 OptiOT - Optimization of operating time depending on outdoor temperature

10.3.1 OptiOT description

Object name (type + number) and family

Type + number: FB 1150

Family: HVAC

Number of process objects: 0 PO

Startup characteristics

Following startup, block processing is started after the number of cycles assigned in the `RunUpCyc` parameter. A startup can be simulated via the `Restart` input.

Called blocks

- FC369 (SelST16)
- SFC6 (RD_SINFO)

Calling OBs

In the same OB and after the block whose switching points are to be optimized and which includes the TimeSwitch block. Additionally in OB100 (see Setting the startup characteristics).

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section OptiOT I/Os (Page 458).

Status bit	Parameter
0 - 1	Not used
2	`SimAct`
3	`OosAct`
4	`OosLi`
5	Not used
6	`OnAct`
7	Not used
8	HeatMode (`Mode = 0`)
9	CoolMode (`Mode = 1`)
10	`LeadOnAct` (`SwitchOn = true` due to lead time)
11	`LeadOffAct` (`SwitchOn = false` due to lead time)
12	`SimLiOp`

Status bit	Parameter
13	Heating (`SwitchOn = true` in heat mode)
14	Cooling (`SwitchOn = true` in cool mode)
15	Off (`SwitchOn = false`)
16	Not used
17	`OptAct` (Optimization on)
18	`ParaErrTS` (TimeSwitch parameterization error)
19	`ParaErrLeadOn` (Lead time active parameterization error)
20	`ParaErrLeadOff` (Lead time inactive parameterization error)
21 - 29	Not used
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

10.3.2 OptiOT operating modes

Heat mode (MODE = 0)

In heat mode, the `OutTempMin` input can be used to set the initial temperature at which the maximum lead time (`LeadTimeMaxOn` input) is calculated for switch-on. The end temperature is set via the switch-on operating point (`OpPointOn` input) and no lead time is calculated for this. The lead time is calculated on a linear basis between these temperatures.
The maximum lead time for switch-off (`LeadTimeMaxOff` input) is applied at the `OutTempMax` initial temperature. The end temperature is determined by the switch-off operating point, `OpPointOff`, and no lead time is calculated for this. The lead time is calculated on a linear basis between these temperatures.

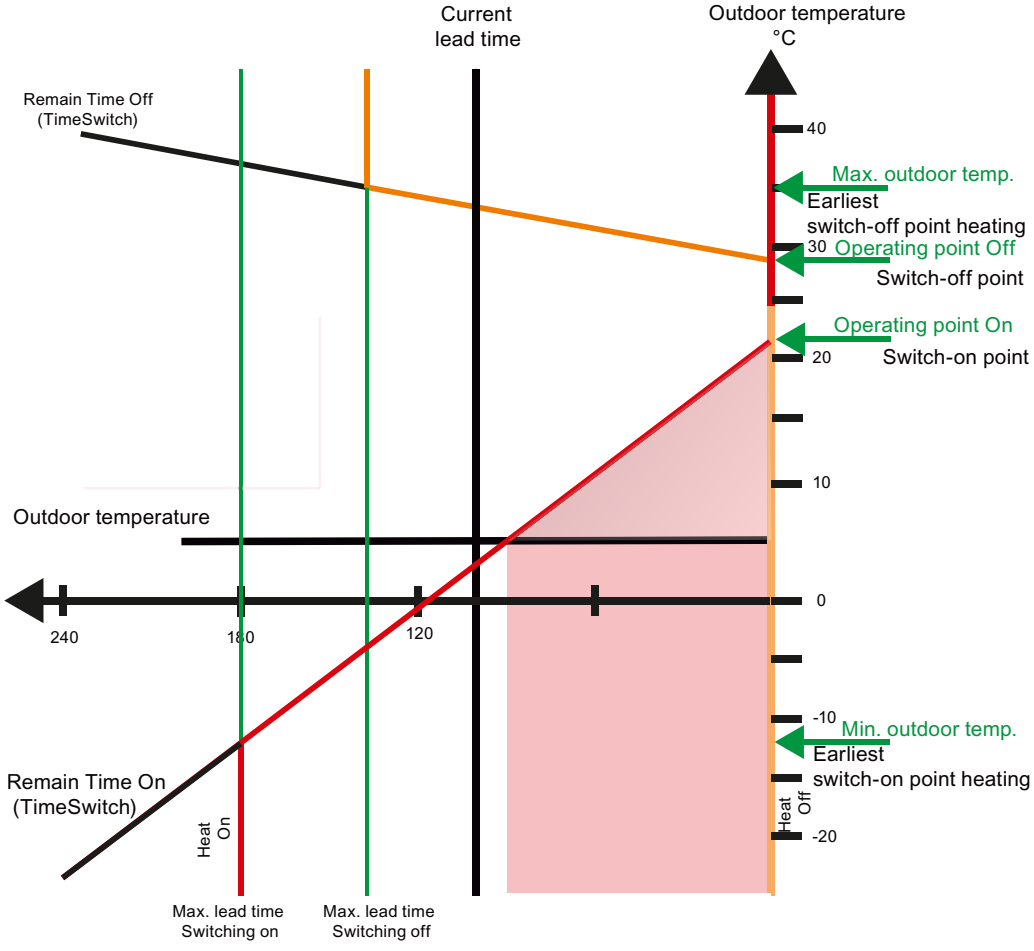


Figure 10-2 OptiOT: Heat

Cool mode (MODE = 1)

In cool mode, the OutTempMax input can be used to set the initial temperature, at which the maximum lead time (LeadTimeMaxOn input) is calculated for switch-on. The end temperature is set via the switch-on operating point (OpPointOn input) and no lead time is calculated for this. The lead time is calculated on a linear basis between these temperatures.
The maximum lead time for switch-off (LeadTimeMaxOff input) is applied at the OutTempMin initial temperature. The end temperature is determined by the switch-off operating point, OpPointOff, and no lead time is calculated for this. The lead time is calculated on a linear basis between these temperatures.

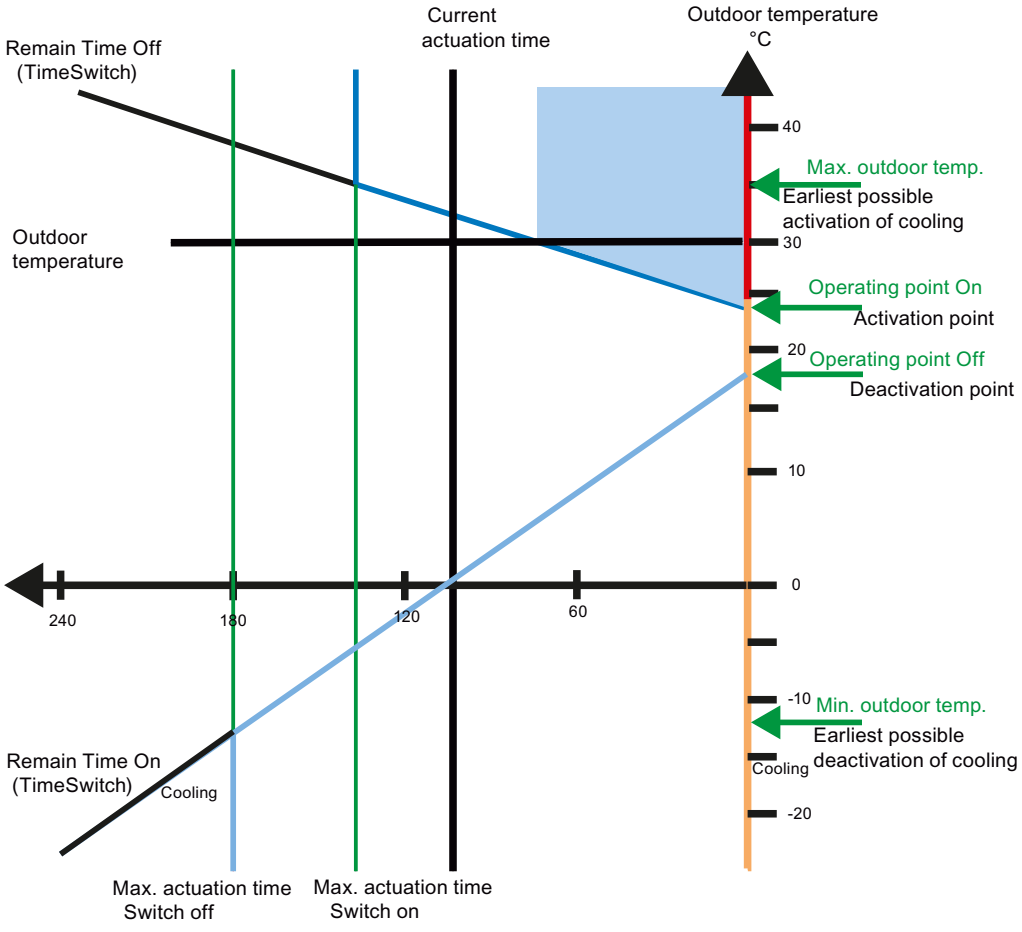


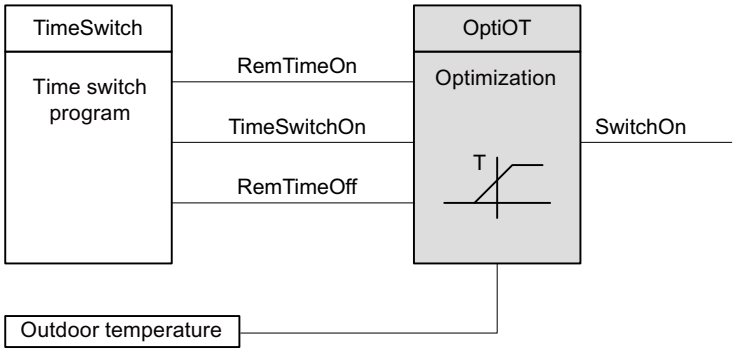
Figure 10-3 OptiOT: Cooling

The SwitchOn output issues the detecting switching signal for the block.
The optimization calculated for the next switching time in [s] is issued at the LeadTime output. RemTimeNxSw issues the time (without optimization) in [s] until the next switching time (on or off) from the TimeSwitch block. RemTimeSwitch includes the optimized time in [s] until the next switching time (RemTimeSwitch = RemTimeNxSw – LeadTime).
LeadTimeOn is the optimized time in [s] for the next switch-on process; LeadTimeOff is the optimized time in [s] for the next switch-off process (at the current outdoor temperature, OutTemp).

10.3.3 OptiOT functions

The OptiOT block optimizes the utilization period depending on the `OutTemp` outdoor temperature. Using this block, it is possible to switch heating or cooling systems on and off earlier (depending on the outdoor temperature) in order to achieve energy-optimized system operations. The block is used to optimize utilization periods and save primary energy as a result. The prerequisite for operating this block is the weekly time-switching program, TimeSwitch, which is also included in the Industry Library. The `On`, `NxTiOn`, and `NxTiOff` outputs of the TimeSwitch block must be connected to the `TimeSwitchOn`, `RemTimeOn`, and `RemTimeOff` inputs of the OptiOT block. Of these, `NxTiOn` indicates the remaining time in [s] until the next switch-on time, `NxTiOff` indicates the remaining time in [s] until the next switch-off time, and `On` indicates the resulting current switching status. In addition, the block also requires the outdoor temperature as an input parameter.

OptiOT_Function:



The block use must be parameterized at the `Mode` block input:

- `Mode = 0`: The block is used to optimize a heating system
- `Mode = 1`: The block is used to optimize a cooling system

Switching time optimization can be deactivated (`LiOp`, `OptOnOp`, `OptOffOp`, `OptOnLi`, `OptOffLi`). When optimization is deactivated, the signal (`On`) for the TimeSwitch block loops through the `SwitchOn` output.

Quality code

The .ST structure elements of the parameters for the measured values `OutTemp`, `RemTimeOn`, `RemTimeOff` and `TimeSwitchOn` contain the quality code.
The following quality code data is evaluated:
Quality Code = 16#60: Simulation on driver block active (
Quality Code = 16#80): Valid value
Quality Code <> 16#60 oder <> 16#80: Invalid value

Simulating signals

This block features the standard function "Simulate signals". It can be used to simulate the outdoor temperature (`OutTemp_Out`) (`SimLiOp`, `SimOutTemp`, `SimOutTemp_Li`).

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4	1 = Operator can enable optimization
5	1 = Operator can disable optimization
6 - 10	Not used
11	1 = Operator can activate the simulation function
12	Not used
13	1 = Operator can change the maximum derivative time (On)
14	1 = Operator can change the maximum derivative time (Off)
15	1 = Operator can change the minimum outside temperature
16	1 = Operator can change the maximum outside temperature
17	1 = Operator can change the operating point (On)
18	1 = Operator can change the operating point (Off)
19 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
24	See APL function: "Enable local operating permission"

10.3.4 OptiOT error handling

The `ParaErr` output issues parameterization errors. The following parameterization errors are identified:

1. Parameterization error in the TimeSwitch block (`Status1.Bit18` = true)
   `RemTimeOn` and `RemTimeOff = 0` or `RemTimeOn = RemTimeOff`
2. Parameterization error in the maximum lead time for switch-on (`Status1.Bit19` = true)
   The next switching time parameterized by the TimeSwitch block should set the `SwitchOn` = false output. The optimized startup time would be before the switch-off time (`RemTimeOff`) as a result of the parameterized maximum lead time for switch-on (`LeadTimeMaxOn`), the current outdoor temperature (`OutTemp`), and the remaining time until the next switch-on (`RemTimeOn`). In this case, the optimized switch-on time is ignored and the switch-on time for the TimeSwitch block (`RemTimeOn`) is used instead.
3. Parameterization error in the maximum lead time for switch-off (`Status1.Bit20` = true)
   The next switching time parameterized by the TimeSwitch block should set the `SwitchOn` = true output. The optimized switch-off time would be before the startup time (`RemTimeOn`) as a result of the parameterized maximum lead time for switch-off (`LeadTimeMaxOff`), the current outdoor temperature (`OutTemp`), and the remaining time until the next switch-off (`RemTimeOff`). In this case, the optimized switch-off time is ignored and the switch-off time for the TimeSwitch block (`RemTimeOff`) is used instead.

10.3.5 OptiOP messaging

Message behavior

The block does not have any messages.

10.3.6 OptiOT I/Os

Input parameters

Parameter	Description	Data type	Default
`LeadTimeMaxOff *`	Maximum switch off lead time [s]	DINT	3600
`LeadTimeMaxOn *`	Maximum switch on lead time [s]	DINT	3600
`LiOp`	1 = Link/Auto, 0 = Manual: Switch on/off optimization	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
`Mode`	Mode. 0 = Heating, 1 = Cooling	INT	0
`OpPointOff *`	Operating point temperature for switching off	REAL	20.0
`OpPointOn *`	Operating point temperature for switching on	REAL	20.0

Parameter	Description	Data type	Default
OptOffLi	Optimization off, via interconnection	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OptOffOp *	Optimization off, via operator	BOOL	0
OptOnLi	Optimization on, via interconnection	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OptOnOp *	Optimization on, via operator	BOOL	0
OutTemp	Outdoor temperature	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
OutTempMax *	Maximum outdoor temperature of optimization	REAL	18.0
OutTempMin *	Minimum outdoor temperature of optimization	REAL	-15.0
RemTimeOff	Remaining time to switch off [s]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
RemTimeOn	Remaining time to switch on [s]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
Restart	Manual restart	BOOL	0
SimOutTemp *	Simulation OutTemp	REAL	0.0
SimOutTemp_Li	Linkable simulation value OutTemp	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
TempUnit *	Unit of temperature	INT	1001
TimeSwitchOn	Time switch has switched on	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
LeadTime	Calculated lead time [s]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
LeadTimeOff	Calculated lead time from next switch to off [s]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
LeadTimeOn	Calculated lead time from next switch to on [s]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
OptAct	1 = Optimization activated	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
OutTemp_Out	Outdoor temperature	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
ParaErr	1 = Parameterization error	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
RemTimeNxSw	Remaining time to next switch from TimeSwitch [s]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
RemTimeSwitch	Remaining time to switching point [s]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
SimAct	1=Simulation active	BOOL	0
SwitchOn	0 = Timer has switched off, 1 = Timer has switched on	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80

10.3.7 OptiOT block diagram

The block has no block diagram.

10.3.8 Operator control and monitoring

10.3.8.1 OptiOT views

OptiOT block views

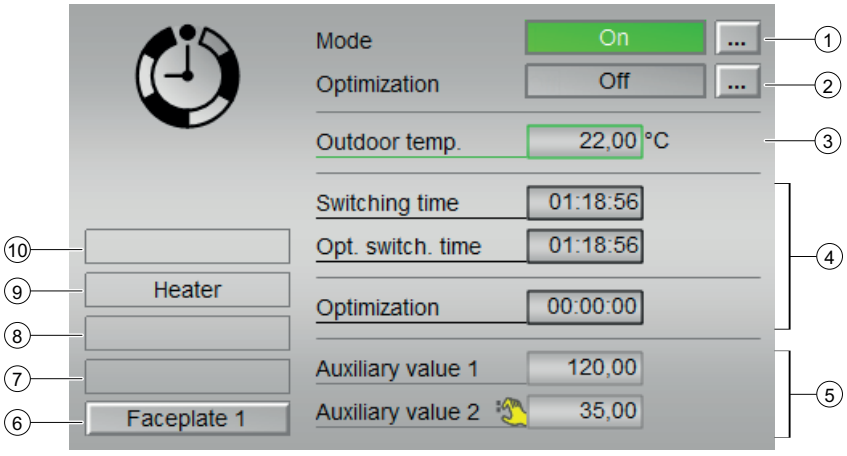
The OptiOT block provides the following views:

- OptiOT standard view (Page 461)
- Trend view
- OptiOT parameter view (Page 463)
- OptiOT preview (Page 464)
- Memo view
- OptiOT block icons (Page 466)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

10.3.8.2 OptiOT standard view

OptiOT standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(2) Display and switchover of the optimization

This area shows the currently valid operating state. The following operating states can be displayed:

- Off
- On

You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(3) Display of the outdoor temperature including signal status

This area shows the current process value of the outside temperature.

(4) Switching time and optimization

This area shows the switching times calculated by the block and the optimization.

- Switching time
- Opt switching time
- Optimization

(5) Display of auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system. You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(6) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(7) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

You can find additional information on this in section "Simulating signals" in the manual of the APL.

(8) Display area for block states

This area provides additional information on the operating state of the block:

- "Derivative on"
- "Derivative off"

(9) Display area for block states

This area provides additional information on the operating state of the block:

- Heater
- Cooling

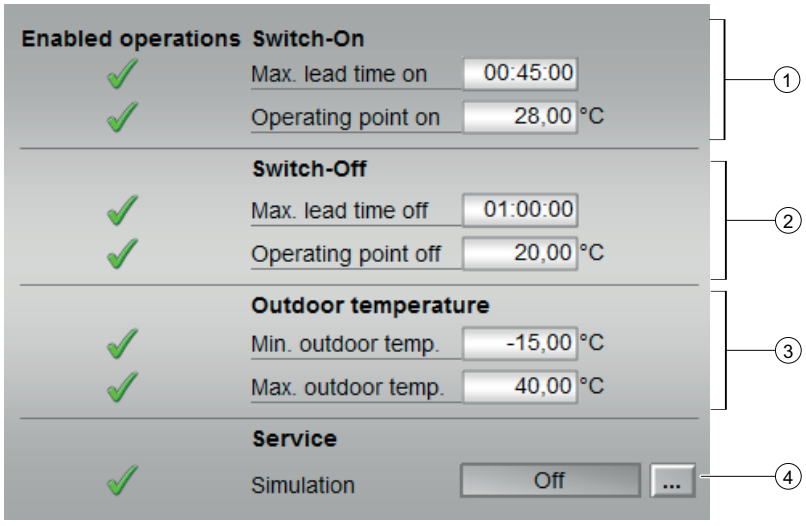
(10) Display area for block states

This area provides additional information on the operating state of the block:

- Error TimeSwitch
- Error derivative on
- Error derivative off

10.3.8.3 OptiOT parameter view

OptiOT parameter view



(1) Switch on

- Maximum time for Derivative on
- Operating point On

(2) Disable

- Maximum time for Derivative off
- Operating point Off

(3) Outdoor temperature

- Minimum outdoor temperature
- Maximum outdoor temperature

(4) Service

You can select the following functions in this area:

- "Simulation"

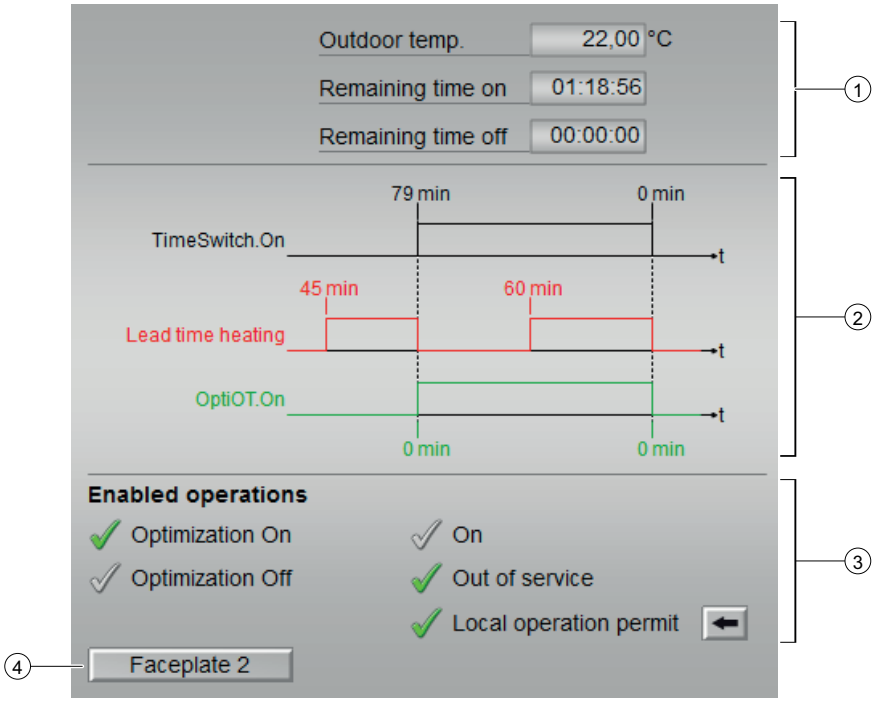
Refer to the "Switchover of operating state and operating mode" section in the APL manual.

You can find information on this area in the APL manual in the section:

- Simulating signals

10.3.8.4 OptiOT preview

OptiOT preview



(1) Preview area

The process values for:

- Outdoor temperature
- Remaining time for startup
- Remaining time for shutdown

are displayed in this area.

(2) Graphical representation

Graphical representation of the values displayed in the standard view under (4) "Switching time and optimization".

(3) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- "Optimization on": You can enable the optimization.
- "Optimization off": You can disable the optimization.
- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

10.3.8.5 OptiOT block icons

OptiOT block icons

A variety of block icons are available with the following functions:

- Process tag type
- Operating modes
- Memo display

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

10.4 ConvCF - Conversion of unit of temperature from °C to °F or from °F to °C

10.4.1 ConvCF description

Object name (type + number) and family

Type + number: FC 609

Family: HVAC

Number of process objects: 0 PO

Startup characteristics

The block has no start-up characteristics.

Time response

The block has no time response.

Called blocks

FC369	SelST16

Calling OBs

In the same OB with and after the block whose measured value is to be monitored.

10.4.2 ConvCF operating modes

The block does not have any operating modes.

10.4.3 ConvCF functions

The ConvCF block converts the unit of temperature from °C to °F, or vice versa.

The block requires the Temperatur in °C or in °F at the Temp input. The conversion direction can be set at the `SelTemp` input parameter by selecting TRUE or FALSE.

The converted temperature is output in the relevant unit at the `Temp_Out` output.

Quality code

The .ST structure elements of the Temp measured value and the selector signal contain the quality code.

The following quality code data is evaluated:

- Quality Code = 16#60: Simulation on driver block active
- Quality Code = 16#80: Valid value
- Quality Code <> 16#60 oder <> 16#80: Invalid value

10.4.4 ConvCF error handling

There is no error handling for this block.

10.4.5 ConvCF messaging

Message behavior

The block does not have any messages.

10.4.6 ConvCF I/Os

Input parameters

Parameter	Description	Data type	Default
SelTemp	Selection of conversion : 0=°C ->°F, 1=°F->°C	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
Temp	Temperature value	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
Temp_Out	Conversion of temperature value	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80

10.4.7 ConvCF block diagram

The block has no block diagram.

10.4.8 Operator control and monitoring of ConvCF

The block does not have any views.

10.5 ConvAbRe - Conversion of humidity from absolute to relative or relative to absolute

10.5.1 ConvAbRe description

Object name (type + number) and family

Type + number: FC 610

Family: HVAC

Number of process objects: 0 PO

Functions

The ConvAbRe block converts absolute humidity to relative humidity, or vice versa.

The block requires the temperature in °C or in °F at the Temp input and the atmospheric humidity in g/kg or in % at the Hum input. The conversion direction can be set at the SelHum input parameter by selecting TRUE or FALSE.

The converted humidity is output in the relevant unit at the Hum_Out output.

Startup characteristics

The block has no start-up characteristics.

Time response

The block has no time response.

Called blocks

FC369	SelST16

Calling OBs

In the same OB with and after the block whose measured value is to be monitored.

10.5.2 ConvAbRe operating modes

The block does not have any operating modes.

10.5.3 ConvAbRe functions

Quality code

The .ST structure elements of the parameters for the measured values Temp and Hum plus the selector signal contain the quality code.

The following quality code data is evaluated:

- Quality Code = 16#60: Simulation on driver block active
- Quality Code = 16#80: Valid value
- Quality Code <> 16#60 oder <> 16#80: Invalid value

10.5.4 ConvAbRe error handling

There is no error handling for this block.

10.5.5 ConvAbRe messaging

Message behavior

The block does not have any messages.

10.5.6 ConvAbRe I/Os

Input parameters

Parameter	Description	Data type	Default
Hum	Humidity value in [g/kg or %]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
SelHum	Selection of conversion : 0=g/kg ->%, 1= %->g/kg	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Temp	Air temperature value in [°C]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
Hum_Out	Conversion of humidity value in [g/kg or %]	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80

10.5.7 ConvAbRe block diagram

The block has no block diagram.

10.5.8 Operator control and monitoring of ConvAbRe

The block does not have any views.

10.6 CalcTHX - Calculation of setpoints for temperature and humidity controller of a ventilation system controlled by an Hx diagram

10.6.1 CalcTHX description

Object name (type + number) and family

Type + number: FB 1159

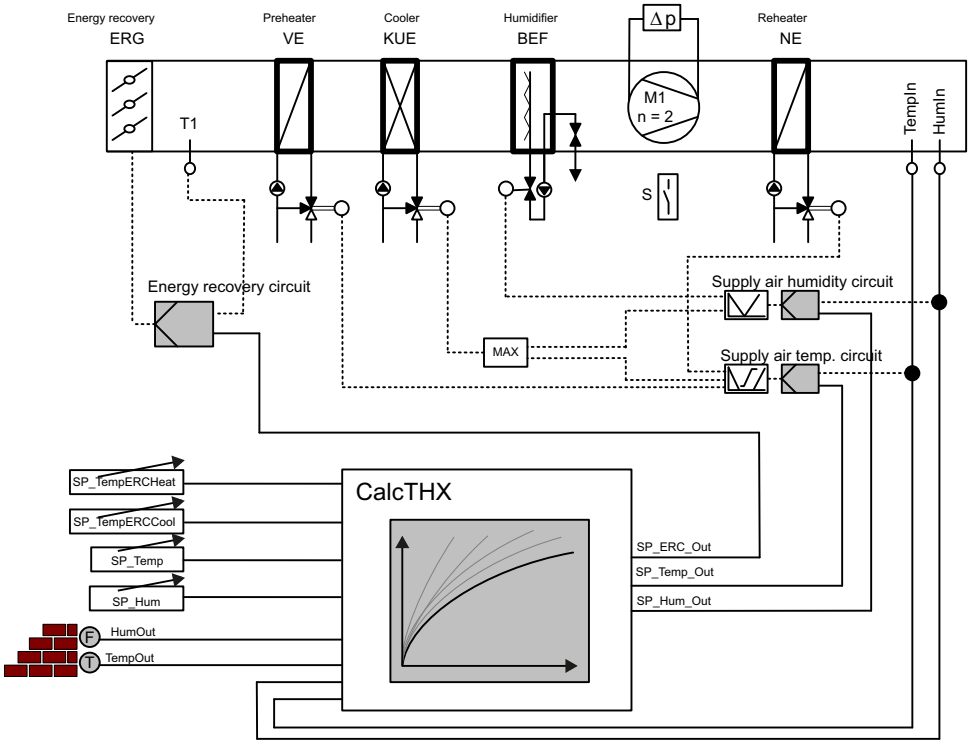
Family: HVAC

Number of process objects: 1 PO

Area of application of CalcTHX

The block is used for calculation of setpoints for temperature and humidity control of a ventilation system (see plant diagram) in the physical unit [°C] or [°F] which is controlled by an Hx diagram and is equipped with a humidifier or washer.

Plant diagram (ventilation system with washer, without master controller):

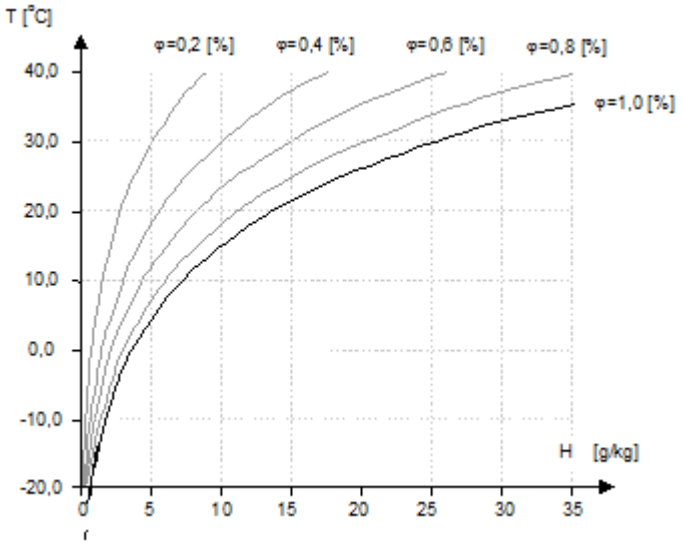


Functional principles

The block calculates the setpoint in the physical unit [°C] or [°F] for the upstream temperature controller for energy recovery (ERC circuit) from the specified setpoints for the supply air temperature and the supply air humidity and the values of the outdoor temperature and the outdoor humidity. To control this energy recovery circuit, the ventilation system needs the temperature value behind the ERC register.

The block represents the status of the ventilation system and the status of the supply air in the Hx diagram.

Hx diagram:



The block can distinguish between four different states of the used ventilation system:

- Heating and humidifying
- Cooling and humidifying
- Heating and cooling without humidifying / dehumidifying
- Heating and cooling with dehumidifying

The block determines the enables for the technological blocks required for heating, cooling, humidifying, dehumidifying depending on the status of the outdoor air. These enables can be used to either reverse the direction of operation or to lock or enable individual blocks.

The block has I/Os to a higher-level adaptive optimization of an ERC circuit.

The block has a selector switch `Feature.Bit6` which determines if the I/O values are relative or absolute humidity values. When the two humidity values (absolute / relative) are switched, the display of the physical units in the faceplate is also switched. The physical unit for absolute humidity values is kJ/kg; relative humidity values are indicated in %.

The block also features switching between °C and °F. The switchover is made with `Feature.Bit5`. The units are displayed according to the parameter assignment of the feature bit.

Startup characteristics

Following startup, the messages are suppressed for the number of cycles assigned in the `RunUpCyc` parameter. A startup can be simulated via the `Restart` input.

Time response

The block must be called by means of a cyclic interrupt OB. The sampling time of the block is entered in the `SampleTime` parameter.

Called blocks

- SFC6 (RD_SINFO)
- SFB35 (ALARM_8P)
- FC369 (SelST16)

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section CalcTHX I/Os (Page 487).

Status bit	Parameter
0	`Occupied`
1	`BatchEn`
2	`SimAct`
3	`OosAct`.Value
4	`OosLi`.Value
5	Not used
6	`OnAct`.Value
7	Not used
8	`EnableERC`.Value
9	`EffectERC`.Value
10	`EnablePreheat`.Value
11	`EnableCooler`.Value
12	`SimLiOp`
13	`EnableHum`
14	`EnableReheat`
15	Relative
16	Absolute
17 - 29	Not used
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word assignment for the parameter `Status2`

For a description of the individual parameters, refer to the section CalcTHX I/Os (Page 487).

Status bit	Parameter
0	`MsgLock`
1	`TempOutCSF`
2	`HumOutCSF`
3	`TempInCSF`
4	`HumInCSF`
5 - 30	Not used
31	`MS_RelOp`

10.6.2 CalcTHX operating modes

The block can be operated using the following modes:

- On
- Out of service

The next section provides additional block-specific information relating to the general descriptions.

"On"

You can find general information about the "On" mode in the On chapter of the APL manual.

"Out of service"

You can find general information about the "out of service" mode in the Out of service chapter of the APL manual.

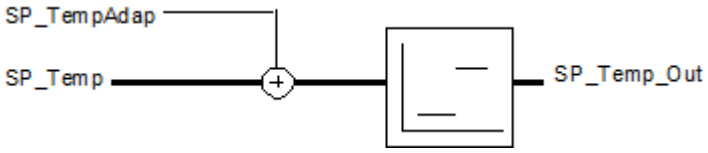
10.6.3 CalcTHX functions

The functions for this block are listed below.

I/O of an external ERC optimization

The block can influence the effective temperature setpoint for the supply air channel `SP_Temp_Out` by means of an external adaptive optimization. The adaptation value `SP_TempAdap` is added to the set setpoint `SP_Temp` and then limited to the defined high limit and low limit (`AdapHighLimit` and `AdapLowLimit`).

Connection diagram:

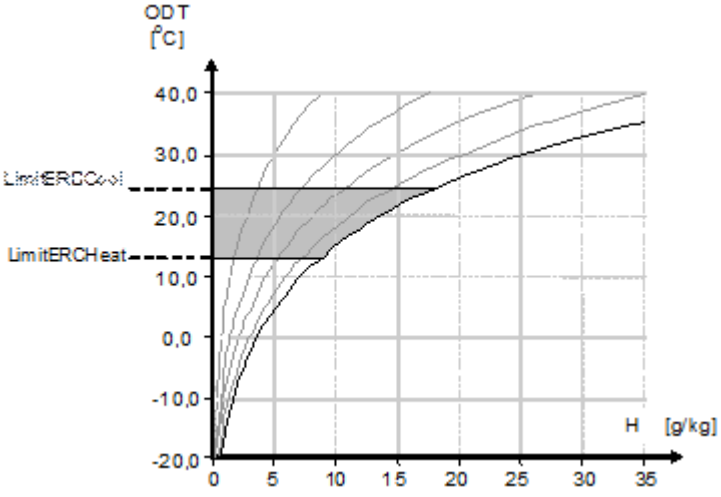


Enabling the ERC circuit and its direction of operation

The control of the ERC circuit only makes sense if the difference between outdoor temperature and supply air temperature exceeds a specific absolute value. The parameters of the high enable limit `LimitERCCool` and the low enable limit `LimitERCHeat` of the ERC circuit can be freely assigned. If the outdoor temperature is above the high enable limit `LimitERCCool`, the `EffectERC` output is set which signals that the ERC circuit operates as cooling final controlling element. Once the high enable limit `LimitERCHeat` has been reached, the `EffectERC` output is reset which signals that the ERC circuit operates as heating final controlling element.

The two enable limits are identified in the faceplate in the Hx diagram by the two gray horizontal lines.

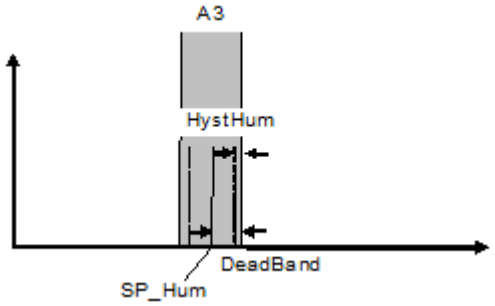
Enable limits:



Deadband and hysteresis of the humidity values

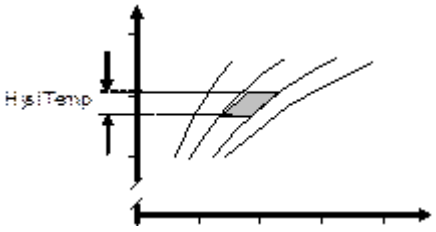
The switchover between humidifying, dehumidifying and neither humidifying nor dehumidifying takes place by separate, configurable internal deadband and hysteresis values.

Deadband:



Hysteresis of the temperature value

The temperature setpoint also offers a hysteresis value. Together with the two hysteresis values of the humidity, it forms a tolerance field of the supply air which looks as follows:



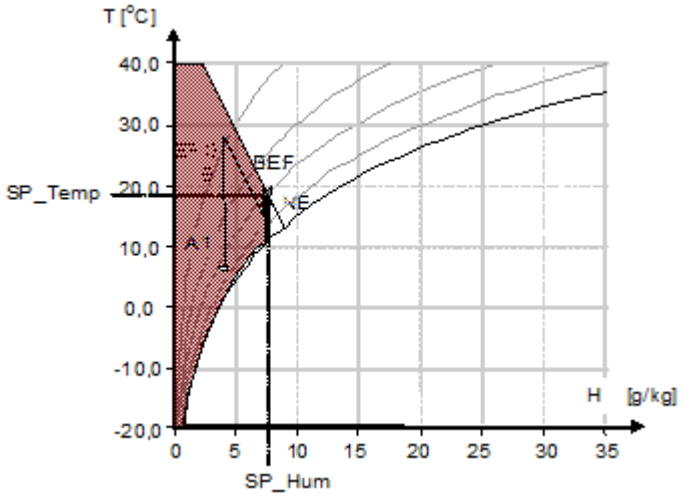
Ramp in setpoint of ERC circuit

The switchover between the different setpoints of the ERC circuit is floating with the help of a ramp function whose ramp gradient, LimitRamp, can be set at the interface. The ramp is reset due to two different events:

- During startup of the block
- Upon user request `SP_ERC_RmpOn` = False

The ramp is virtually deactivated with the setting `SP_ERC_RmpOn` = False. It works with a ramp gradient of 100% per sampling cycle.

Heating and humidifying



ERC circuit:

Setpoint:	`SP_TempERCHeat`, if `TempOut` < `LimitERCHeat`
	`SP_Temp`, if `TempOut` > `LimitERCHeat`
	`SP_Temp`, if the controller is not enabled

Sequence:	Heating only
Enable:	If `TempOut` < `LimitERCHeat`

Supply air temperature circuit:

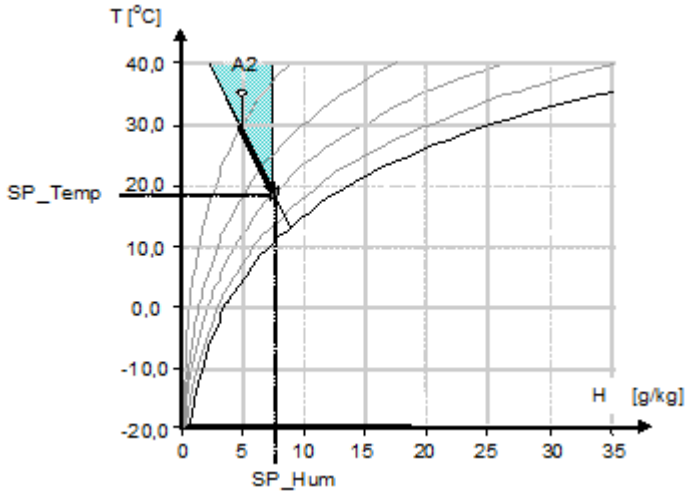
Setpoint:	`SP_Temp`
Sequence:	Heating only via preheater and reheater

Supply air humidity circuit:

Setpoint:	`SP_Hum`
Sequence:	Humidifier only

Enables	
ERC	TRUE (Heating)
PH	TRUE
COO	FALSE
HUM	TRUE
RH	TRUE

Cooling and humidifying



ERC circuit:

Setpoint:	`SP_TempERCCool`, if the controller is enabled `SP_Temp`, if the controller is not enabled
Sequence:	Cooling
Enable:	If `TempOut` > `LimitERCCool`

Supply air temperature circuit:

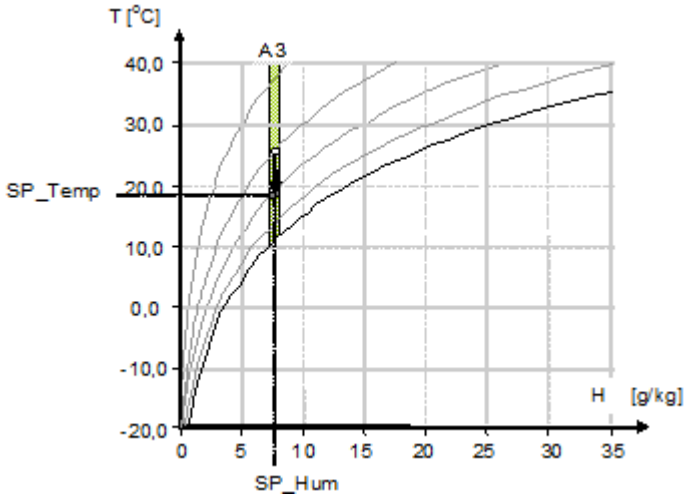
Setpoint:	`SP_Temp`
Sequence:	Heating via preheater, reheater and cooling

Supply air humidity circuit:

Setpoint:	`SP_Hum`
Sequence:	Humidifier only

Enables	
ERC	TRUE (Cooling)
PH	FALSE
COO	TRUE
HUM	TRUE
RH	TRUE

Heating and cooling without humidifying



ERC circuit:

Setpoint:	`SP_TempERCCool`, if `TempOut` > `LimitERCCool`
	`SP_TempERCHeat`, if `TempOut` < `LimitERCHeat`
	`SP_Temp`, if the controller is not enabled

Sequence:	Cooling and heating
Enable:	If `LimitERCHeat` > `TempOut` > `LimitERCCool`

Supply air temperature circuit:

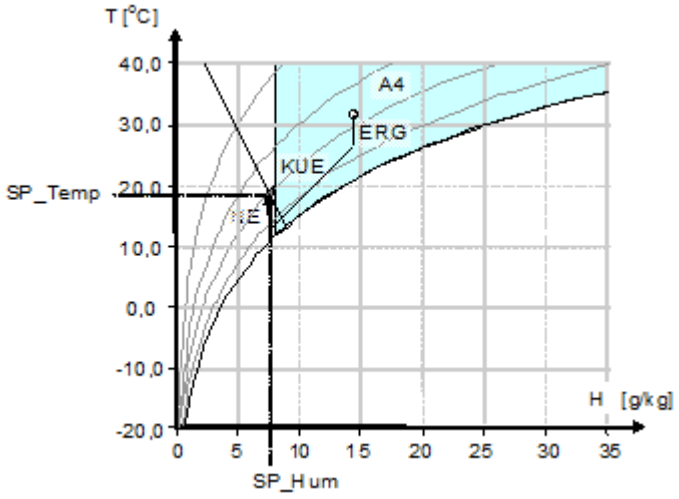
Setpoint:	`SP_Temp`
Sequence:	Heating via preheater, reheater and cooling

Supply air humidity circuit:

Setpoint:	`SP_Hum`
Sequence:	Cooling and humidifying

Enables	
ERC	TRUE
PH	TRUE
COO	TRUE
HUM	FALSE
RH	TRUE

Heating and cooling with humidifying



ERC circuit:

Setpoint:	`SP_TempERCCool`, if `TempOut` > `LimitERCCool` `SP_Temp`, if the controller is not enabled
Sequence:	Cooling
Enable:	If `TempOut` > `LimitERCCool`

Supply air temperature circuit:

Setpoint:	`SP_Temp`
Sequence:	Heating via preheater, reheater and cooling

Supply air humidity circuit:

Setpoint:	`SP_Hum`
Sequence:	Humidifier only

Enables	
ERC	TRUE (Cooling)
PH	FALSE
COO	TRUE
HUM	FALSE
RH	TRUE

Note

The ERC setpoints and limits have a physical correlation. The limits should be outside or identical to the setpoints. Recommended: `SP_TempERCHeat >= LimitERCHeat` and `SP_TempERCCool <= LimitERCCool`. These recommendations are not run as plausibility checks in the block, however, to not subject the user to limitations in the parameter assignment. They are merely recommendations.

Wet bulb temperature

The block uses the relative humidity [%] and temperature [°C] to calculate the wet bulb temperature in the physical unit [°C] or [°F]. The wet bulb temperature is provided at the `TempWeBu_Out` output.

$$t_{FK} = -5{,}809 + 0{,}058 * \varphi + 0{,}697 * t_{Luft} + 0{,}003 * \varphi * t_{Luft}$$

Figure 10-4 Formula_Wet_Bulb

t_{FK}= Wet bulb temperature (°C)

φ= Relative humidity (%)

t_{Air}= Air temperature (°C)

Opening additional faceplates

In the same OB with and after the block whose measured value is to be monitored. Additionally in OB100 (see Setting the startup characteristics).

Quality code

The .ST structure elements of the parameters for the measured values `Temp` and `RelHum` plus the external message signals contain the quality code.

The following quality code data is evaluated:

Quality code = 16#60:	Simulation on driver block active
Quality code = 16#80:	Valid value
Quality code <> 16#60 or <> 16#80:	Invalid value

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4 - 10	Not used
11	1 = Operator can enable the Simulation function
12	1 = Operator can activate the Release for maintenance function
13 - 15	Not used
16	1 = Operator can change the temperature setpoint
17	1 = Operator can change the humidity setpoint
18	1 = Operator can change the limit temperature for ERC heating
19	1 = Operator can change the limit temperature for ERC cooling
20	1 = Operator can change the humidity deadband
21	1 = Operator can change the ERC system limits
22	1 = Operator can change the humidity hysteresis
23	1 = Operator can change the temperature hysteresis
24	1 = Operator can change the setpoint for ERC heating
25	1 = Operator can change the setpoint for ERC cooling
26	1 = Operator can change the limit for high adaptation
27	1 = Operator can change the limit for low adaptation
28 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
5	0 = Processing of temperature values in °C; 1 = Processing of temperature values in °F
6	0 = Operating mode relative humidity; 1 = Operating mode absolute humidity
19	See APL function: "Changes signal status of outputs in OOS"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"

Release for maintenance

This block features the standard function "Release for maintenance".

Simulating signals

This block features the standard function "Simulate signals".

10.6.4 CalcTHX error handling

The input for control system fault `CSF` can be interconnected with an external error output. The error is not evaluated. A message is generated at the ALARM_8P block.

10.6.5 CalcTHX messaging

Message behavior

The following messages can be generated for this block:

- Process messages
- Instance-specific messages

The messages can be suppressed centrally via the faceplate or via `MsgLock` (program).

The free alarm inputs can be interconnected via the parameters `ExtMsg1` to `ExtMsg3`.

The associated values (`ExtValxy`) of the message block can be assigned freely.

MsgSup is set if the `RunUpCyc` cycles have not finished running since the restart, `MsgLock` = TRUE or `MsgStat` = 21.

Messages

You can use up to three instance-specific messages with this block.

Message instance	Message ID	Message class	Event
MsgEvId 1	SIG 1	AS process control message - fault	$$BlockComment$$ External error for outdoor temperature has occurred
	SIG 2	AS process control message - fault	$$BlockComment$$ External error for outdoor humidity has occurred
	SIG 3	AS process control message - fault	$$BlockComment$$ External error for supply air temperature has occurred
	SIG 4	AS process control message - fault	$$BlockComment$$ External error for supply air humidity has occurred
	SIG 5	< No message >	$$BlockComment$$ Time-driven step enabling not possible
	SIG 6	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 7	AS process control message - fault	$$BlockComment$$ External message 2
	SIG 8	AS process control message - fault	$$BlockComment$$ External message 3

Explanation:

$$BlockComment$$: content of the instance-specific comment

The message "Time-driven step enabling not possible" is sent if the time for time-driven step enabling has expired but the block is internal or external step enabling mode.

The message is not active by default.

Associated values for the message instance **MsgEvId**

Associated value	Block parameters
1	BatchName
2	StepNo
3	BatchID
4	ExtVa104
5	ExtVa105
6	ExtVa106
7	ExtVa107
8	ExtVa108

Associated value	Block parameters
9	ExtVa109
10	ExtVa110

The associated values 4 ... 10 are allocated to the parameters ExtVa104 ... ExtVa110, which you are free to use. You can find additional information on this in the "Process Control System PCS 7 - Engineering System" manual.

10.6.6 CalcTHX I/Os

Input parameters

Parameter	Description	Data type	Default
AdapHighLimit *	High limit of the permissible setpoint of the supply air temperature after the adaption increase in [°C] or [°F]	REAL	23.0
AdapLowLimit *	Low limit of the permissible setpoint of the supply air temperature after the adaption increase in [°C] or [°F]	REAL	13.0
AtmPress *	Atmospheric pressure for the calculation of the saturated humidity	REAL	1013.0
AtmPress_Unit *	Unit of measure for the atmospheric pressure	INT	1136
C1NegTemp	C1 constant for the calculation of the saturated pressure in case of a negative temperature (preset: 22.4433)	REAL	224.433
C1PosTemp	C1 constant for the calculation of the saturated pressure in case of a positive temperature (preset: 17.5043)	REAL	175.043
C2NegTemp	C2 constant for the calculation of the saturated pressure in case of a negative temperature (preset: 272.186)	REAL	272.186
C2PosTemp	C2 constant for the calculation of the saturated pressure in case of a positive temperature (preset: 241.175)	REAL	241.175
C3	C3 Constant for the calculation of the saturated pressure (preset: 611.213)	REAL	611.213
DeadBandHum *	Deadband of the absolute humidity in [g/kg]	REAL	1.0
HumHiLim *	High limit for humidity parameter	REAL	40.0
HumIn *	Measured supply air humidity	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
HumInCSF	External error (supply air humidity channel error)	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80

Parameter	Description	Data type	Default
HumLoLim *	Low limit for humidity parameter	REAL	0.0
HumOut *	Measured outside humidity	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
HumOutCSF	External error (outside humidity channel error)	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
HystHum *	Hysteresis for the humidity in [g/kg]	REAL	0.1
HystTemp *	Hysteresis for the temperature in [°C] or [°F]	REAL	0.3
LimitERCCool *	Limit for enabling the ERG cooling system in [°C] or [°F]	REAL	23.0
LimitERCHeat *	Limit for enabling the ERG heating system in [°C] or [°F]	REAL	14.0
LimitRamp *	Rate of change of the setpoint in [%/s]	REAL	2.0e-2
SimHumIn *	Supply air humidity that is used at SimOn = 1	REAL	0.0
SimHumIn_Li	Interconnectable supply air humidity that is used at SimOnLi = 1	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
SimHumOut *	Outside humidity that is used at SimOn = 1	REAL	0.0
SimHumOut_Li	Interconnectable outside humidity that is used at SimOnLi = 1	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
SimTempIn *	Supply air temperature that is used at SimOn = 1	REAL	0.0
SimTempIn_Li	Interconnectable supply air temperature that is used at SimOnLi = 1	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
SimTempOut *	Outside temperature that is used at SimOn = 1	REAL	0.0
SimTempOut_Li	Interconnectable outside temperature that is used at SimOnLi = 1	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
SP_ERC_RmpOn	1 = Activate ramp for SP_ERC_Out to target setpoint SP_TempERCHeat / SP_TempERCCool	BOOL	1
SP_Hum *	Setpoint supply air humidity in [g/kg]	REAL	0.0
SP_Temp *	Setpoint supply air temperature in [°C] or [°F]	REAL	15.0
SP_TempAdap *	Setpoint increase through adaption in [°C] or [°F]	REAL	0.0

Parameter	Description	Data type	Default
SP_TempERCCool *	Setpoint of the ERG cooling circuit in [°C] or [°F]	REAL	22.0
SP_TempERCHeat *	Setpoint of the ERG heating circuit in [°C] or [°F]	REAL	14.0
TempHiLim *	High limit for temperature parameter	REAL	40.0
TempIn *	Measured supply air temperature	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
TempInCSF	External error (supply temperature channel error)	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
TempLoLim *	Low limit for temperature parameter	REAL	0.0
TempOut *	Measured outside temperature	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
TempOutCSF	External error (outside temperature channel error)	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
DisplayX_PV1	Determined display value in Hx diagram for the measured outside humidity	REAL	0.0
DisplayX_PV2	Determined display value in Hx diagram for the measured supply air humidity	REAL	0.0
DisplayX_SP	Determined display value in Hx diagram for the setpoint of the supply air humidity	REAL	0.0
DisplayX2	Determined display value in Hx diagram for the x-value	REAL	0.0
DisplayY_PV1	Determined display value in Hx diagram for the measured outside temperature	REAL	0.0
DisplayY_PV2	Determined display value in Hx diagram for the measured supply air temperature	REAL	0.0
DisplayY_SP	Determined display value in Hx diagram for the setpoint of the supply air temperature	REAL	0.0
DisplayY1	Determined display value in Hx diagram for the y-value 1	REAL	0.0
DisplayY2	Determined display value in Hx diagram for the y-value 2	REAL	0.0

Parameter	Description	Data type	Default
DisplayY3	Determined display value in Hx diagram for the y-value 3	REAL	0.0
EffectERC	0 = ERG system for heating activated, 1 = ERG system for cooling activated	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
EnableCooler	1 = Cooler enabled	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
EnableERC	1 = ERG system enabled	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
EnableHum	1 = Humidifier enabled	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
EnablePreheat	1 = Preheater enabled	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
EnableReheat	1 = Reheater enabled	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
HumIn_Out	Measured supply air humidity	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
HumIn_Unit	Unit of measure for the supply humidity	INT	0
HumInCSF_Out	External error of the supply air humidity	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
HumOut_Out	Measured outside air humidity	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
HumOut_Unit	Unit of measure for outside humidity	INT	0
HumOutCSF_Out	External error of the outside humidity	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
HxAreaA1 ... HxAreaA4	1 = Field A1 ... A4 in Hx diagram for heating and humidifying active	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
ProcInF_Act	Editing of the temperatures in °F active	BOOL	0
SP_ERC_Out	Determined setpoint for the ERG system	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80

Parameter	Description	Data type	Default
SP_ERC_Unit	Unit of measure of the setpoint for the ERG system	INT	1001
SP_Hum_Out	Determined setpoint for the supply humidity (low limit: 0, high limit if Feature.Bit6 = 0: 100%, high limit if Feature.Bit6 = 1: 40 g/kg)	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
SP_Hum_Unit	Unit of measure of the setpoint for the supply air humidity	INT	0
SP_Temp_Out	Determined setpoint for the supply air temperature (low limit: AdapLowLimit, high limit: AdapHighLimit)	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
SP_Temp_Unit	Unit of measure of the setpoint for the supply air temperature	INT	1001
StatusHxArea	Status for the Hx diagram	INT	0
TempIn_Out	Measured supply air temperature	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
TempIn_Unit	Unit of measure for the supply air temperature	INT	1001
TempInCSF_Out	External error of the supply air temperature	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
TempOut_Out	Measured outside air temperature	STRUCT • Value: REAL • ST: BYTE	 • 0.0 • 16#80
TempOut_Unit	Unit of measure for the outside temperature	INT	1001
TempOutCSF_Out	External error of the outside temperature	STRUCT • Value: BOOL • ST: BYTE	 • 0 • 16#80
TempWeBu_Out	Wet bulb temperature [°C]	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

10.6.7 CalcTHX block diagram

A block diagram is not provided for this block.

10.6.8 Operator control and monitoring

10.6.8.1 CalcTHX views

CalcTHX block views

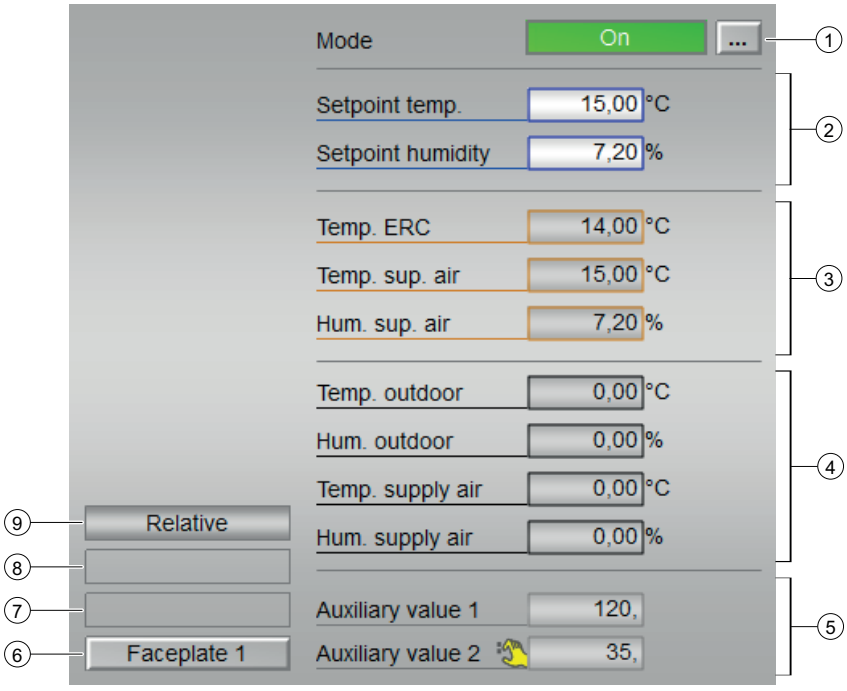
The CalcTHX block provides the following views:

- CalcTHX standard view (Page 493)
- Message view
- Limit view of CalcTHX (Page 495)
- CalcTHX diagram view (Page 496)
- CalcTHX parameter view (Page 497)
- CalcTHX preview (Page 499)
- Memo view
- Batch view
- CalcTHX block icons (Page 501)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

10.6.8.2 CalcTHX standard view

CalcTHX standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(2) Display and change of the setpoints

This area shows the current setpoints:

- Setpoint temperature in °C
- Setpoint humidity in %

Refer to the section "Changing values" in the APL manual for information about changing the setpoint.

(3) Calculated values

This area shows the currently calculated values of the block algorithm.

- Temperature ERC in °C
- Supply air temperature in °C
- Supply air humidity in %

(4) Display of the process values including signal status

This area shows the current process values with the corresponding signal status.

- Outdoor air temperature in °C
- Outdoor air humidity in %
- Supply air temperature in °C
- Supply air humidity in %

(5) Display of auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system. You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(6) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(7) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

You can find more information about this in the "Release for maintenance" section of the APL manual.

(8) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

You can find additional information on this in section "Simulating signals" in the manual of the APL.

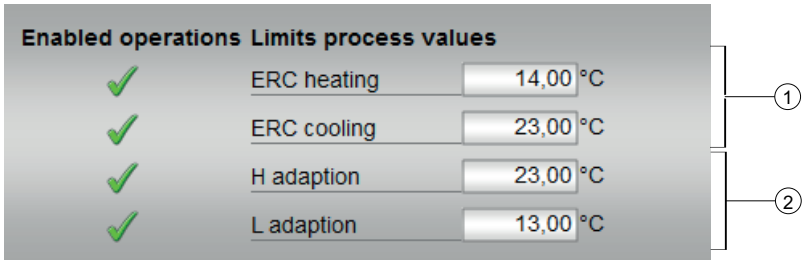
(9) Display area for block states

This area provides additional information on the operating state of the block:

- Relative
- Absolute

10.6.8.3 Limit view of CalcTHX

Limit view of CalcTHX



(1) Limits process values - ERC

You can enter the limits in this area. Refer to the "Changing values" section in the APL manual.

- ERC heating
  Specifies the value of the maximum outdoor temperature at which the ERC is still enabled as heating final controlling element. The parameter has an effect on (T_EN_HRH).
- ERC cooling
  Specifies the value of the minimum outdoor temperature at which the ERC is still enabled as cooling final controlling element. (ERC heating must be less than ERC cooling). The parameter has an effect on (T_EN_HRC).

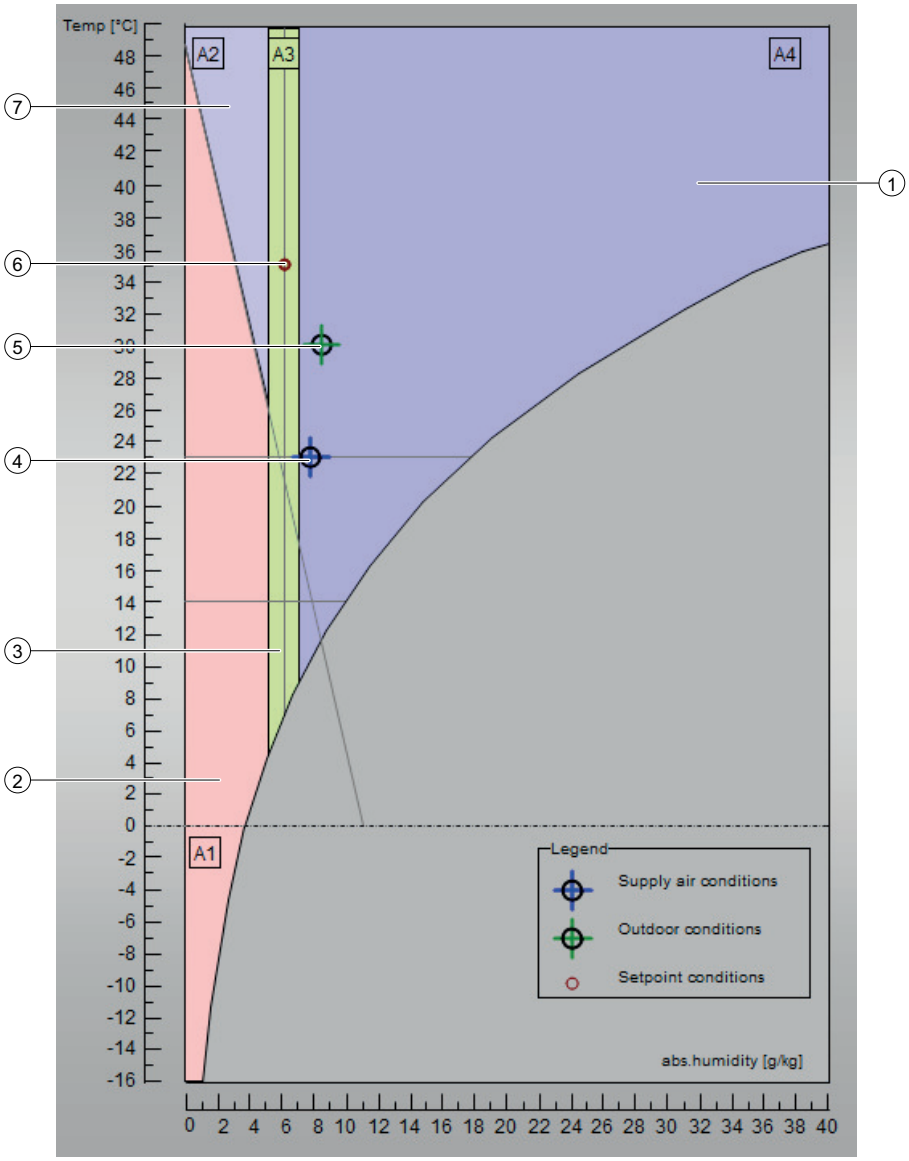
(2) Limits - Adaption

You can enter the limits in this area. Refer to the "Changing values" section in the APL manual.

- H adaption
  Specifies the value of the maximum value permitted for adaption.
- L adaption
  Specifies the value of the minimum value permitted for adaption.

10.6.8.4 CalcTHX diagram view

CalcTHX chart view



(1) Area A4

In this area, the function described in the section "Functions of CalcTHX (Page 476)" under "Heating and cooling with humidifying" applies.

(2) Area A1

In this area, the function described in the section "Functions of CalcTHX (Page 476)" under "Heating and cooling" applies

(3) Area A3

In this area, the function described in the section "Functions of CalcTHX (Page 476)" under "Heating and cooling without humidifying" applies

(4) Status of the supply air

This sign indicates the current status of the supply air.

(5) Status of the outside air

This sign indicates the current status of the outside air.

(6) Target status of the supply air

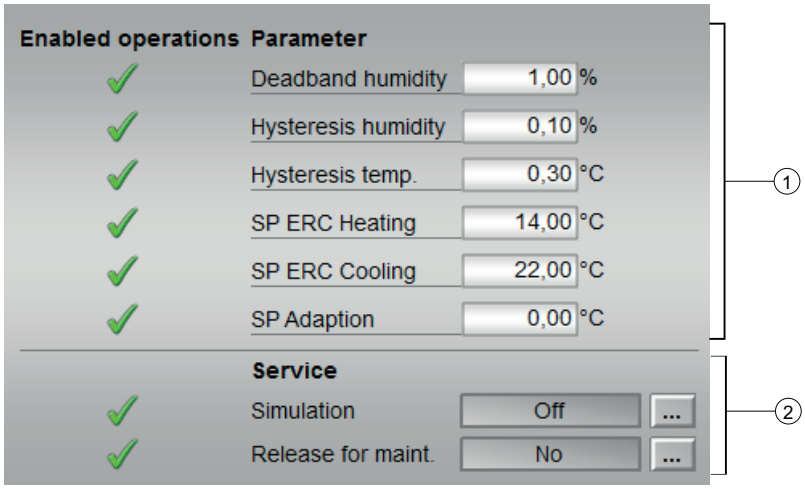
This sign indicates the setpoint of the supply air.

(7) Area A2

In this area, the function described in the section "Functions of CalcTHX (Page 476)" under "Cooling and humidifying" applies

10.6.8.5 CalcTHX parameter view

Parameter view of CalcTHX



(1) Parameter

In this area, you change parameters and thereby influence the calculation of the block. Refer to the "Changing values" section in the APL manual.

- Deadband humidity: Specifies the deadband value of the permitted supply air humidity
- Hysteresis humidity: Specifies the hysteresis value of the permitted supply air humidity values
- Hysteresis temp.: Specifies the hysteresis value of the permitted supply air temperature values
- SP ERC Heating: Specifies the setpoint for the EVR circuit that becomes active with enabled ERC and its heating case by means of the described ramp function at output (SP_ERC_Out). The parameter has an effect on (SP_TempERCHeat).
- SP ERC Cooling: Specifies the setpoint for the EVR circuit that becomes active with enabled ERC and its cooling case by means of the described ramp function at output (SP_ERC_Out). The parameter has an effect on (SP_TempERCCool).
- SP Adaption

(2) Service

You can select the following functions in this area:

- "Simulation"
- "Release for maintenance"

Refer to the "Switchover of operating state and operating mode" section in the APL manual.

You can find information on this area in the APL manual in the section:

- Simulating signals
- Release for maintenance

Enabled operations

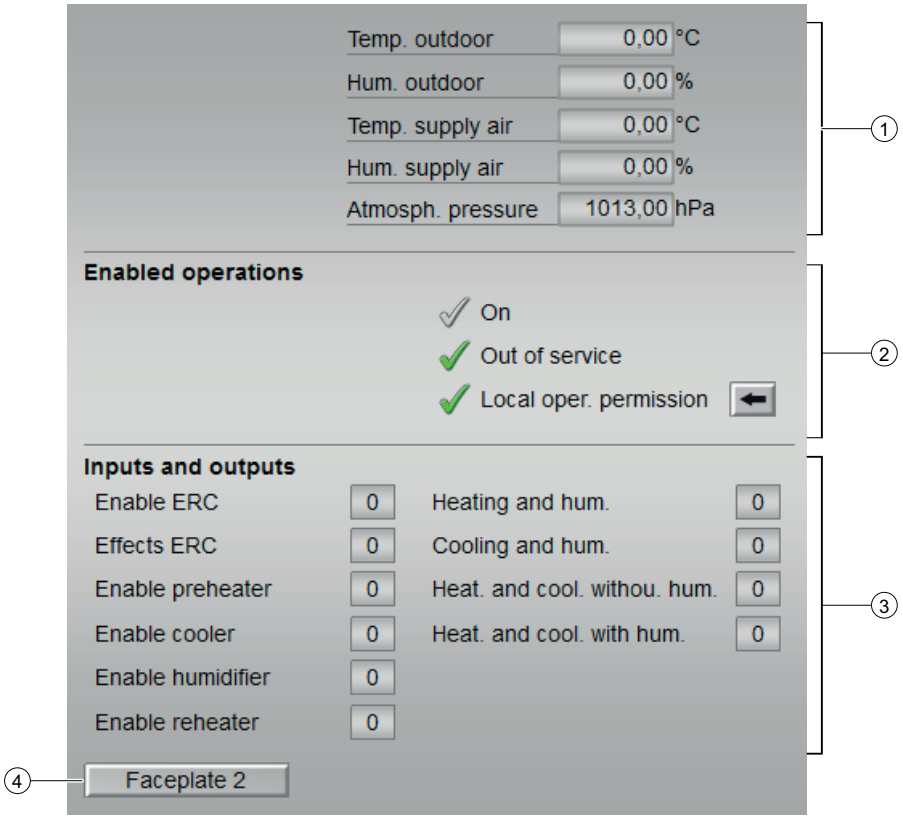
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- Green check mark: The OS operator can control this parameter
- Gray check mark: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- Red X: The OS operator is generally prohibited from accessing this parameter due to the configured AS operator control permissions (OS_Perm or OS1Perm).

10.6.8.6 CalcTHX preview

CalcTHX preview



(1) Preview area

The process values for:

- Outdoor air temperature in °C
- Outdoor air humidity in %
- Supply air temperature in °C
- Supply air humidity in %
- Atmospheric pressure in hPa

are displayed in this area.

(2) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(3) Inputs and outputs

This area shows the most important parameters for this block with the current control:

- Enable ERC
- ERC acts as cooler
- Enable preheater
- Enable cooler
- Enable humidifier
- Enable reheater
- Heating and humidifying
- Cooling and humidifying
- Heating and cooling without humidifying
- Heating and cooling with humidifying

(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

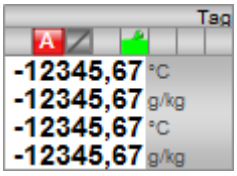
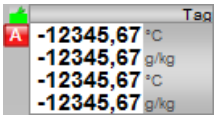
10.6.8.7 CalcTHX block icons

CalcTHX block icons

A variety of block icons are available with the following functions:

- Process tag type
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes
- Signal status, release for maintenance
- Memo display
- Process values

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

Communication blocks

11

11.1 Application areas for the communication blocks

The Industry Library provides communication blocks for several application areas:

- Communication between S7-400 (single) and S7-300
- Communication between S7-400 (single) and S7-1500
- Communication between S7-400H (redundant) and S7-300
- Communication between S7-400H (redundant) and S7-1500
- Communication between S7-400H (redundant) and S7-400 (single)

Note

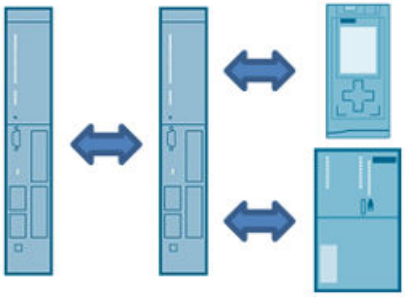
The blocks for use in S7-300 and S7-1500 are identical in their function. The blocks are supplied for CPU S7-300 for use in the SIMATIC Manager and for CPU S7-1500 for use in the TIA Portal.

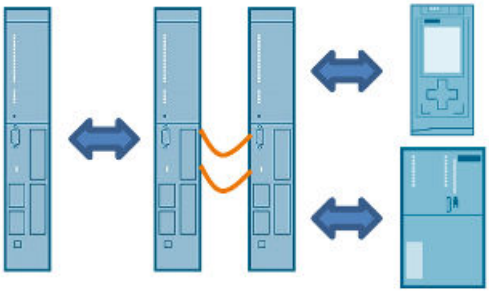
Note

Communication between redundant (S7-400H) and non-redundant (S7-300 / S7-1500 / S7-400) stations

When communicating via the H blocks, the communication is only guaranteed for simple failure (fault in Rack0 or Rack1) on the H system. If multiple failures occurs (fault in Rack0 and Rack1, e.g. CPU in Rack0 is disturbed and CP in Rack1 is disrupted), communication is no longer possible.

Used blocks

Application	S7-400H	S7-400	S7-300	S7-1500
 Single communication	-	Snd_DigVal[1]	S7RcvDig[2]	S7RcvDig[2]
	-	Rcv_DigVal[1]	S7SndDig[2]	S7SndDig[2]
	-	Snd_AnaVal[1]	S7RcvAna[2]	S7RcvAna[2]
	-	Rcv_AnaVal[1]	S7SndAna[2]	S7SndAna[2]

Application	S7-400H	S7-400	S7-300	S7-1500
 H communication	RcvH_DigVal[3]	SndH_DigVal[3]	S7RcvHDig[2]	S7RcvHDig[2]
	SndH_DigVal[3]	RcvH_DigVal[3]	S7SndHDig[2]	S7SndHDig[2]
	RcvH_AnaVal[3]	SndH_AnaVal[3]	S7RcvHAna[2]	S7RcvHAna[2]
	SndH_AnaVal[3]	RcvH_AnaVal[3]	S7SndHAna[2]	S7SndHAna[2]
	ASSendH[3]	ASRcvH[3]	-	-
	ASRcvH[3]	ASSendH[3]	-	-

1 Block is part of the APL

2 Block is part of the IL S7

3 Block is part of the IL PCS 7

Communication between S7-400 (single) and S7-300

To realize coordinated communication between a non-redundant S7-400 and an S7-300, the following blocks are used:

The blocks Snd_DigVal, Snd_AnaVal, Rcv_DigVal and Rcv_AnaVal of the APL are used in the S7-400.

The IL provides the following blocks for S7-300:

- S7SndDig: Change-triggered transmission of up to 128 analog structures.
- S7SndAna: Change-triggered transmission of up to 32 analog structures.
- S7RcvDig: Reception of up to 128 digital structures.
- S7RcvAna: Reception of up to 32 analog structures.

Both stations are configured in SIMATIC Manager and CFC.

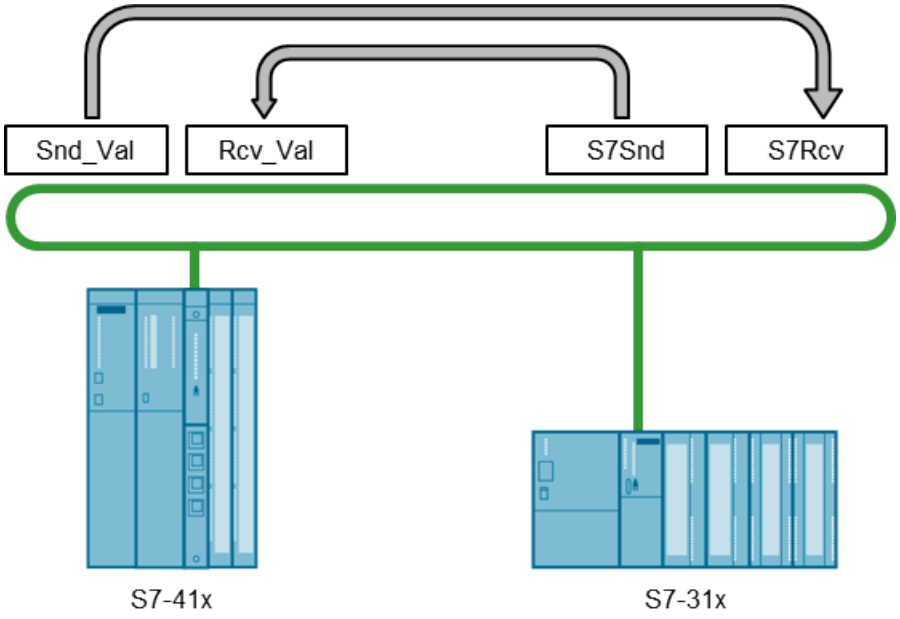


Figure 11-1 Communication between S7-400 (single) and S7-300

Communication between S7-400 (single) and S7-1500

To realize coordinated communication between a non-redundant S7-400 and an S7-1500, the following blocks are used:

The blocks Snd_DigVal, Snd_AnaVal, Rcv_DigVal and Rcv_AnaVal of the APL are used in the S7-400.

The IL provides the following blocks for S7-1500:

- S7SndDig: Change-triggered transmission of up to 128 analog structures.
- S7SndAna: Change-triggered transmission of up to 32 analog structures.
- S7RcvDig: Reception of up to 128 digital structures
- S7RcvAna: Reception of up to 32 analog structures.

S7-400 is configured in the SIMATIC Manager and CFC. S7-1500 is configured in the TIA Portal. The blocks are optimized in the TIA Portal for the use of LAD, FBD or STL.

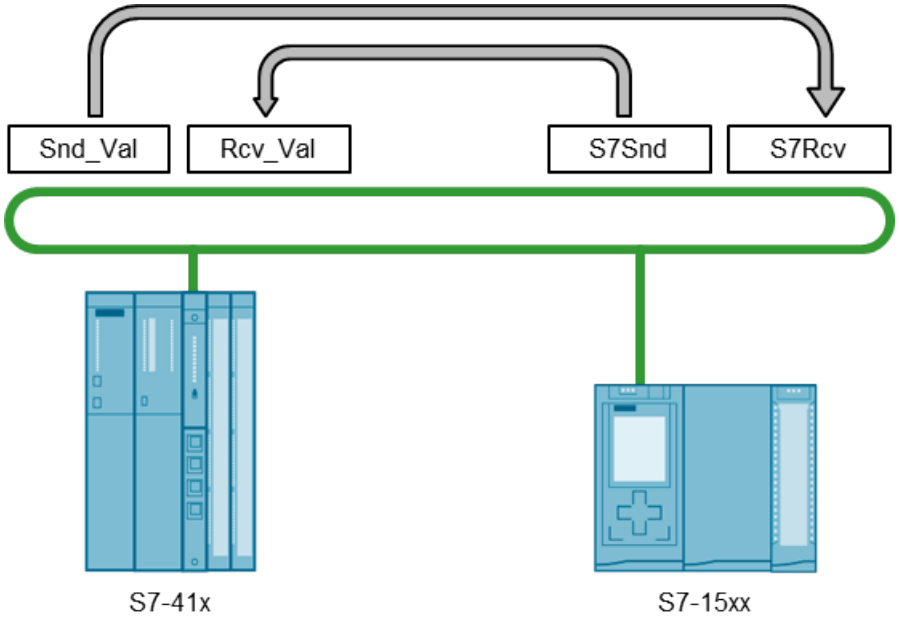


Figure 11-2 Communication between S7-400 (single) and S7-1500

Communication between S7-400H (redundant) and S7-300

To realize coordinated communication between a redundant S7-400H and an S7-300, the following blocks are used:

The IL provides the following blocks for S7-400H:

- SndH_DigVal: Change-triggered transmission of up to 128 analog structures.
- SndH_AnaVal: Change-triggered transmission of up to 32 analog structures.
- RcvH_DigVal: Reception of up to 128 digital structures
- RcvH_AnaVal: Reception of up to 32 analog structures.

The IL provides the following blocks for S7-300:

- S7SndHDig: Change-triggered transmission of up to 128 analog structures.
- S7SndHAna: Change-triggered transmission of up to 32 analog structures.
- S7RcvHDig: Reception of up to 128 digital structures
- S7RcvHAna: Reception of up to 32 analog structures.

Both stations are configured in SIMATIC Manager and CFC.

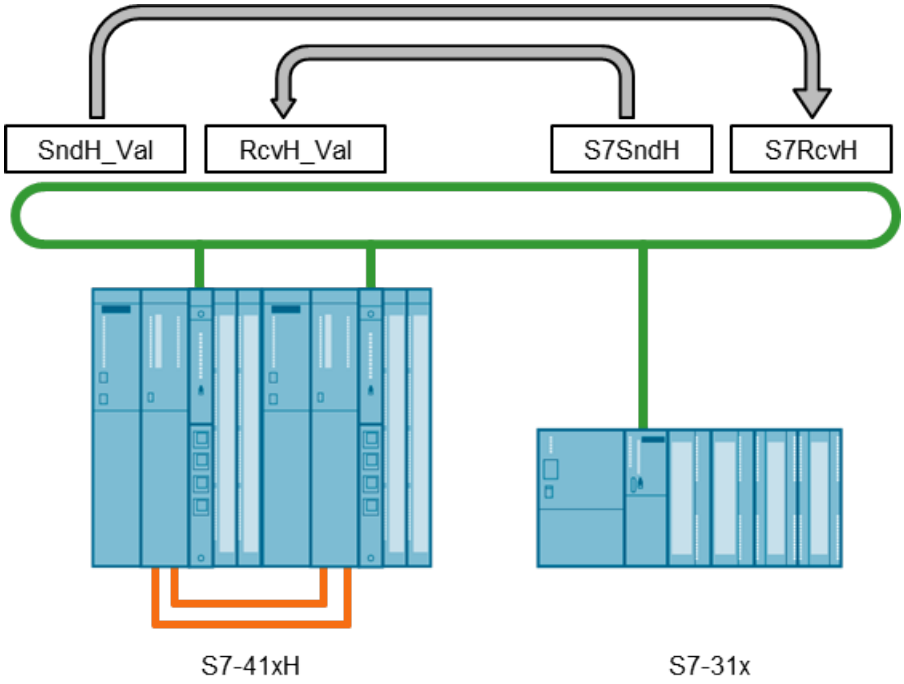


Figure 11-3 Communication between S7-400H (redundant) and S7-300

Communication between S7-400H (redundant) and S7-1500

To realize coordinated communication between a redundant S7-400H and an S7-1500, the following blocks are used:

The IL provides the following blocks for S7-400H:

- SndH_DigVal: Change-triggered transmission of up to 128 analog structures.
- SndH_AnaVal: Change-triggered transmission of up to 32 analog structures.
- RcvH_DigVal: Reception of up to 128 digital structures
- RcvH_AnaVal: Reception of up to 32 analog structures.

The IL provides the following blocks for S7-1500:

- S7SndHDig: Change-triggered transmission of up to 128 analog structures.
- S7SndHAna: Change-triggered transmission of up to 32 analog structures.
- S7RcvHDig: Reception of up to 128 digital structures
- S7RcvHAna: Reception of up to 32 analog structures.

NOTICE

S7-400 is configured in the SIMATIC Manager and CFC. S7-1500 is configured in the TIA Portal. The blocks are optimized in the TIA Portal for the use of LAD, FBD or STL.

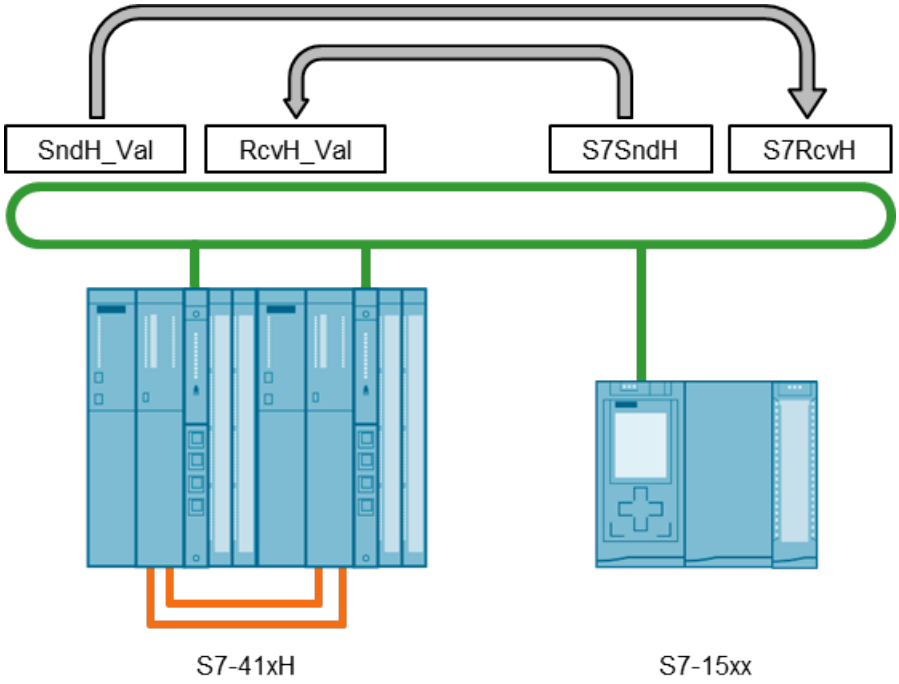


Figure 11-4 Communication between S7-400H (redundant) and S7-1500

Communication between S7-400H (redundant) and S7-400 (single)

To realize coordinated communication between a redundant and a non-redundant S7-400, the following blocks are used:

- SndH_DigVal: Change-triggered transmission of up to 128 analog structures.
- SndH_AnaVal: Change-triggered transmission of up to 32 analog structures.
- RcvH_DigVal: Reception of up to 128 digital structures
- RcvH_AnaVal: Reception of up to 32 analog structures.

The `RedCPU` I/O is used in this application to define in which of the two communication partners the block is used. (`RedCPU`: 1 = Red. Controller S7-41xH, 0 = Single controller S7-41x)

Alternatively, the existing communication blocks continue to be maintained for existing plants:

- ASSendH: Send up to 30 real and 30 binary values with QualityCode
- ASRcvH: Reception of up to 30 real and 30 binary values with QualityCode

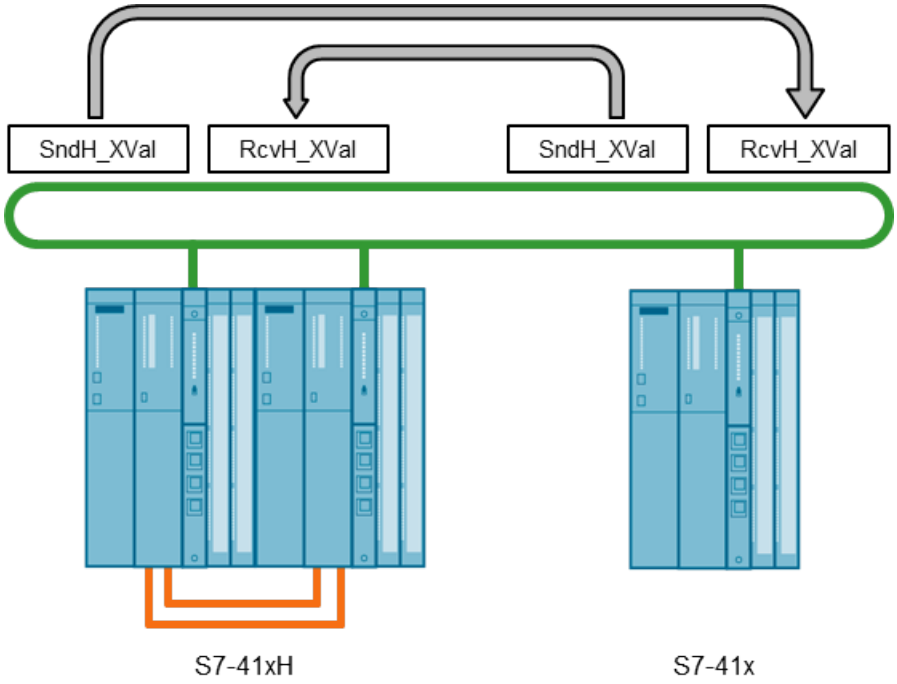


Figure 11-5 Communication between S7-400H (redundant) and S7-400 (single)

Consumption of connection resources when using the communication blocks

The basis for the data transfer is a physical connection between two controllers via MPI, PROFIBUS or Industrial Ethernet. Based on this connection, the S7 connections are configured on both sides in PCS 7, STEP 7 or TIA Portal, depending on the CPU type. It is important to ensure that a connection resource of the CPU is used for each S7 connection.

The following table shows an overview of the available connection resources of the respective CPU types.

CPU type S7-	Connection resources (minus PG and OP connections 2 / 10 \ [S7-1500])			
	Total	MPI	DP	IE/PN
412-x	32	32	16	-
412-2 PN	48	32	16	48
414-x	32	32	16	-
414-3 PN/DP 414F-3 PN/DP	64	32	16	64
416-2 416F-2 416-3	64	44	32	-
416-3 PN/DP 416F3 PN/DP	96	44	32	96
417	120	44	32	-
410-5H	120	120	120	120
31x-y PN/DP	12 - 32	n	n	n
151x	64 – 192	n	n	n

For more details, refer to the manuals of the CPU types.

The data limit for the data exchange is 64 KB per block pair. The transmission of the data via a communication processor (CP) of the S7-300 is an exception with 32 KB. A maximum of eight jobs can be executed in parallel for each S7 connection.

11.2 Communication between S7-41x and S7-31x / S7-15xx

11.2.1 S7SndDig - Send digital values

11.2.1.1 S7SndDig description

Object name (type + number) and family

Type + number: FB 330

Family: COMM

Number of process objects: 0 PO

Application area for S7SndDig

The "S7SndDig" block represents a simple user interface to SFB 12 "BSEND". The "S7SndDig" block sends 128 digital structures via an MPI, PROFIBUS or Industrial Ethernet connection to another CPU. This CPU calls the function block "Rcv_DigVal" (FB 1892) of the PCS 7 Advanced Process Library to receive the data.

How it works

Sending is initiated by calling the block with the value 1 at the `SendEn` control input. During the transmission of data, `SendAct` is set to 1.

If the job is completed without error, `SendDone` = 1 is set. `SendErr` is set to 1 is an error occurs and new job is automatically triggered with the current data until all data have been successfully sent.

If the input `SendEn` = 0, incomplete transmission is canceled. No further sending takes place afterward (`SendAct` = 0).

The `ID` parameter is the connection number from the configuration of the S7 connection in "NetPro" (PCS 7 / STEP 7) or "Devices and Networks" (TIA Portal).

The `R_ID` parameter is any number for each S7 connection (`ID`) that is unique and identical at the corresponding send and receive blocks.

You can re-configure the addressing parameters `ID` and `R_ID` in runtime. The new parameters take effect with each new job once the previous job is completed. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Note

Error code or incorrect data at the data block

Only one Rcv block may be assigned to each Snd block. The combination of `ID` (connection number) and `R_ID` (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor.

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter `Status`

This block does not have the `Status` parameter

11.2.1.2 S7SndDig operating modes

This block has no operating mode.

11.2.1.3 Function of S7SndDig

The block does not have any configurable characteristics.

11.2.1.4 S7SndDig error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 12 "BSEND".

The `SendErr(ERROR)` and `SendStat(STATUS)` outputs show the specific error information for the lower-level SFB 12 "BSEND".

You can find the error information for the ERROR and STATUS outputs of the SFB 12 in the manual "System Software for S7-300/400 - System and Standard Functions".

If an error occurs, a new job is automatically triggered with the current data until all data have been successfully transferred.

11.2.1.5 S7SndDig messaging

No message behavior is applicable to this block.

11.2.1.6 S7SndDig I/Os

Input parameters

Parameter	Description	Type	Initial value
EN	1 = Called block is executed	BOOL	1
ID	ID of physical connection	WORD	16#0000
R_ID	ID of message frame connection	DWORD	16#00000000
SndEn	1 = Activate continuous sending	BOOL	1
PV_In00 ... PV_In127	Input_0 ... Input_127	BOOL	1

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
SndAct	1 = Command is active	BOOL	0
SndDone	1 = Command is executed	BOOL	0
SndErr	1 = Command is executed with error	BOOL	0
SndStat	Type of error	WORD	16#0000

11.2.1.7 S7SndDig block diagram

A block diagram is not provided for this block.

11.2.2 S7SndAna - Send analog values

11.2.2.1 S7SndAna description

Object name (type + number) and family

Type + number: FB 332

Family: COMM

Number of process objects: 0 PO

Application area for S7SndAna

The "S7SndAna" block represents a simple user interface to SFB 12 "BSEND". The "S7SndAna" block sends 32 analog structures via MPI, PROFIBUS or Industrial Ethernet connection to another CPU. This CPU calls the function block "Rcv_AnaVal" (FB 1894) of the PCS 7 Advanced Process Library to receive the data.

How it works

Sending is initiated by calling the block with the value 1 at the `SendEn` control input. The data are sent according to the settings of `CylMin`, `CylMax` or `PV_In_HysXX`. During the transmission of data, `SendAct` is set to 1.

If the job is completed without error, `SendDone` = 1 is set. `SendErr` is set to 1 is an error occurs and new job is automatically triggered with the current data until all data have been successfully sent.

If the input `SendEn` = 0, incomplete transmission is canceled. No further sending takes place afterward (`SendAct` = 0).

You can use `CylMIN` to set the number of cycles to wait before triggering the transfer of the current input data, regardless of any changes to one or more values.

You can use `CylMAX` to specify how many cycles after the last valid data transfer to wait before sending the current input data. Sending is executed even when no value has changed or when the changes of a REAL value are within the hysteresis `PV_In_HysXX`.

The default value of the hysteresis is `PV_In_HysXX` = 0. If sending is not to be performed after each value change, configure the `PV_In_HysXX` input accordingly. This involves an absolute value.

When `CylMax` is reached, transfer is forced after the corresponding number of cycles.

The `ID` parameter is the connection number from the configuration of the S7 connection in "NetPro" (PCS 7 / STEP 7) or "Devices and Networks" (TIA Portal).

The `R_ID` parameter is any number for each S7 connection (`ID`) that is unique and identical at the corresponding send and receive blocks.

You can re-configure the addressing parameters `ID` and `R_ID` in runtime. The new parameters take effect with each new job once the previous job is completed. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Note

Error code or incorrect data at the data block

Only one Rcv block may be assigned to each Snd block. The combination of `ID` (connection number) and `R_ID` (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor.

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter `Status`

This block does not have the `Status` parameter

11.2.2.2 S7SndAna operating modes

This block has no operating mode.

11.2.2.3 Function of S7SndAna

The block does not have any configurable characteristics.

11.2.2.4 S7SndAna error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 12 "BSEND".

The `SendErr(ERROR)` and `SendStat(STATUS)` outputs show the specific error information for the lower-level SFB 12 "BSEND".

You can find the error information for the ERROR and STATUS outputs of the SFB 12 in the manual "System Software for S7-300/400 - System and Standard Functions".

If an error occurs, a new job is automatically triggered with the current data until all data have been successfully transferred.

11.2.2.5 S7SndAna messaging

No message behavior is applicable to this block.

11.2.2.6 S7SndAna I/Os

Input parameters

Parameter	Description	Type	Initial value
EN	1 = Called block is executed	BOOL	1
ID	ID of physical connection	WORD	16#0000
R_ID	ID of message frame connection	DWORD	16#00000000
SndEn	1 = Activate continuous sending	BOOL	1
CylMin	Minimum waiting cycle	INT	1
CylMax	Maximum waiting cycle	INT	10
PV_In_Hys00 ... PV_In_Hys31	Hysteresis for PV_In_00 ... Hysteresis for PV_In_31	REAL	0.0
PV_In00 ... PV_In31	Input_00 ... Input_31	REAL	0.0

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
SndAct	1 = Command is active	BOOL	0
SndDone	1 = Command is executed	BOOL	0
SndErr	1 = Command is executed with error	BOOL	0
SndStat	Type of error	WORD	16#00000

11.2.2.7 S7SndAna block diagram

A block diagram is not provided for this block.

11.2.3 S7RcvDig - Receive digital values

11.2.3.1 S7RcvDig description

Object name (type + number) and family

Type + number: FB 331

Family: COMM

Number of process objects: 0 PO

Application area for S7RcvDig

The "S7RcvDig" block represents a simple user interface to SFB 13 "BRCV". The block receives 128 digital structures via MPI, PROFIBUS or Industrial Ethernet connection from another CPU. This CPU must call the function block "Snd_DigVal" (FB 1891) of the PCS 7 Advanced Process Library to send the data.

In STEP 7, a uniform connection must be configured for both communication partners and transferred to the AS.

How it works

Data are entered in the data block asynchronously to user-program execution. After "S7RcvDig" has been called, instance DB data may not be processed as long as the job is in progress (`RcvNewData` = 0). If the job is completed without errors, the `RcvNewData` output is set to 1 for one cycle. In the next cycle, the receive enable is automatically sent by the FB to the operating system of the CPU.

The receive enable can take effect prior to the arrival of the first receive job. In this case, the receive enable is stored by the operating system.

The `ID` parameter is the connection number from the configuration of the S7 connection in "NetPro" (PCS 7 / STEP 7) or "Devices and Networks" (TIA Portal).

The `R_ID` parameter is any number for each S7 connection (`ID`) that is unique and identical at the corresponding send and receive blocks.

You can re-configure the addressing parameters `ID` and `R_ID` in runtime. The new parameters take effect with each new job once the previous job is completed. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Call the block pair consisting of the send (S7RcvDig) and receive block (Snd_DigVal) in each program cycle (cyclically or also via timeout alarm).

Note

Error code or incorrect data at the data block

Only one Snd block may be assigned to each Rcv block. The combination of `ID` (connection number) and `R_ID` (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

If you want to simulate the values, enable simulation via the `SimOn` input. In this case, the values from the inputs `SimPVX` are written to the outputs `PV_Outx`. As long as simulation is running, the data sent are not applied, since data reception is switched off.

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor.

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter `Status`

This block does not have the `Status` parameter

11.2.3.2 S7RcvDig operating modes

This block has no operating mode.

11.2.3.3 Function of S7RcvDig

Configurable reactions using the `Feature` parameter

The following function is available at the `Feature` I/O for this block:

- Bit 29: Output substitute values if raw value is invalid

11.2.3.4 S7RcvDig error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 13 "BRCV".

The SendErr (ERROR) and SendStat (STATUS) outputs show the specific error information for the lower-level SFB 13 "BRCV".

You can find additional information in the manual System Software for S7-300/400 - System and Standard Functions. There you can find a description for the RcvXErr (ERR) and RcvXStat (STAT) outputs of SFB 13.

If Bit29 in the Feature structure is set, and no new data is received after the expiration of RcvMonCyc (number of cycles), the substitute value is output and the RcvMonErr output is set to 1. While RcvMonCyc cycles are running, the last received values are applied to the output.

If Bit29 in the Feature structure is not set and an error occurs, the last valid values are always applied to the output.

11.2.3.5 S7RcvDig messaging

No message behavior is applicable to this block.

11.2.3.6 S7RcvDig I/Os

Input parameters

Parameter	Description	Type	Initial value
EN	1 = Called block is executed	BOOL	1
ID	ID of physical connection	WORD	16#0000
R_ID	ID of message frame connection	DWORD	16#00000000
RcvMonCyc	Number of cycles for receiving monitoring error	INT	3
SimOn	1= simulation active	BOOL	0
SimPV0 ... SimPV127	Simulation value	BOOL	0
SubsPV1 ... SubsPV127	Substitute value	BOOL	0
Feature	I/O for additional functions (Page 518)	STRUCT	-
	• Reserved	• Bit0 ... Bit28: BOOL	• 0
	• 1 = substitute value	• Bit29: BOOL	• 0
	• Reserved	• Bit30 ... Bit31: BOOL	• 0

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
PV_Out0 ... PVOut127	Output_0 ... Output_1277	BOOL	0

Parameter	Description	Type	Initial value
SimAct	1= simulation active	BOOL	0
RcvMonErr	1 = no data received	BOOL	0
RcvNewData	1 = no data received	BOOL	0
RcvErr	1 = Command is executed with error	BOOL	0
RcvStat	Type of error	WORD	16#0000

11.2.3.7 S7RcvDig block diagram

A block diagram is not provided for this block.

11.2.4 S7RcvAna - Receive analog values

11.2.4.1 S7RcvAna description

Object name (type + number) and family

Type + number: FB 333

Family: COMM

Number of process objects: 0 PO

Application area for S7RcvAna

The "S7RcvAna" block represents a simple user interface to SFB 12 "BRCV". The block sends 32 analog structures via MPI, PROFIBUS or Industrial Ethernet connection to another CPU. This CPU must call the function block "Snd_AnaVal" (FB 1893) of the PCS 7 Advanced Process Library to send the data.

In STEP 7, a uniform connection must be configured for both communication partners and transferred to the AS.

How it works

Data are entered in the data block asynchronously to user-program execution. After "S7RcvAna" has been called, instance DB data may not be processed as long as the job is in progress (RcvNewData = 0). If the job is completed without errors, the RcvNewData output is set to 1 for one cycle. In the next cycle, the receive enable is automatically sent by the FB to the operating system of the CPU.

The receive enable can take effect prior to the arrival of the first receive job. In this case, the receive enable is stored by the operating system.

The ID parameter is the connection number from the configuration of the S7 connection in "NetPro" (PCS 7 / STEP 7) or "Devices and Networks" (TIA Portal).

The R_ID parameter is any number for each S7 connection (ID) that is unique and identical at the corresponding send and receive blocks.

You can re-configure the addressing parameters ID and R_ID in runtime. The new parameters take effect with each new job once the previous job is completed. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Call the block pair consisting of the send (S7RcvAna) and receive block (Snd_AnaVal) in each program cycle (cyclically or also via timeout alarm).

Note

Error code or incorrect data at the data block

Only one Snd block may be assigned to each Rcv block. The combination of ID (connection number) and R_ID (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

If you want to simulate the values, enable simulation via the SimOn input. In this case, the values from the inputs SimPVX are written to the outputs PV_Outx. As long as simulation is running, the data sent are not applied, since data reception is switched off.

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor.

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter Status

This block does not have the Status parameter

11.2.4.2 S7RcvAna operating modes

This block has no operating mode.

11.2.4.3 Function of S7RcvAna

Responses that can be configured with the "Feature" parameter

The following function is available at the `Feature` I/O for this block:

- Bit 29: Output substitute values if raw value is invalid

11.2.4.4 S7RcvAna error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 13 "BRCV".

The `SendErr` (`ERROR`) and `SendStat` (`STATUS`) outputs show the specific error information for the lower-level SFB 13 "BRCV".

You can find additional information in the manual System Software for S7-300/400 - System and Standard Functions. There you can find a description for the `RcvXErr` (ERR) and `RcvXStat` (STAT) outputs of SFB 13.

If Bit29 in the `Feature` structure is set, and no new data is received after the expiration of `RcvMonCyc` (number of cycles), the substitute value is output and the `RcvMonErr` output is set to 1. While `RcvMonCyc` cycles are running, the last received values are applied to the output.

If Bit29 in the `Feature` structure is not set and an error occurs, the last valid values are always applied to the output.

11.2.4.5 S7RcvAna messaging

No message behavior is applicable to this block.

11.2.4.6 S7RcvAna I/Os

Input parameters

Parameter	Description	Type	Initial value
`EN`	1 = Called block is executed	BOOL	1
`ID`	ID of physical connection	WORD	16#0000
`R_ID`	ID of message frame connection	DWORD	16#00000000
`RcvMonCyc`	Number of cycles for receiving monitoring error	INT	3
`SimOn`	1 = Simulation active	BOOL	0
`SimPV0...SimPV31`	Simulation value	REAL	0.0
`SubsPV0...SubsPV31`	Substitute value	REAL	0.0
`Feature`	I/O for additional functions (Page 522) • Reserved • 1 = substitute value • Reserved	STRUCT • Bit0 ... Bit28: BOOL • Bit29: BOOL • Bit30 ... 31: BOOL	- • 0 • 0 • 0

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
PV_Out0 ... PV_Out31	Output_0 ... Output_31	REAL	0.0
SimAct	1 = Simulation active	REAL	0.0
RcvMonErr	1 = no data received	BOOL	0
RcvNewData	1 = new data received	BOOL	0
RcvErr	1 = Command is executed with error	BOOL	0
RcvStat	Type of error	WORD	16#0000

11.2.4.7 S7RcvAna block diagram

A block diagram is not provided for this block.

11.3 Communication between S7-41xH and S7-31x / S7-15xx

11.3.1 SndH_DigVal - Send digital values

11.3.1.1 SndH_DigVal description

Object name (type + number) and family

Type + number: FB 1170

Family: COMM

Number of process objects: 0 PO

Application area for SndH_DigVal

The "SndH_DigVal" block represents a simple user interface to SFB 12 "BSEND". The "SndH_DigVal" block sends 128 digital structures via an MPI, PROFIBUS or Industrial Ethernet connection to another CPU. This CPU calls the function block "RcvH_DigVal" (FB 1171) or "S7RcvHDig" (FB 334) of the PCS 7 Industry Library to receive the data.

The blocks are used to create a connection between a redundant and non-redundant automation system.

How it works

Sending is initiated by calling the block with the value 1 at the `SendEn` control input. During the transmission of data, `SendXAct` is set to 1.

If the job is completed without error, `SendXDone` = 1 is set. `SendXErr` is set to 1 is an error occurs and new job is automatically triggered with the current data until all data have been successfully sent.

If the input `SendEn` = 0, incomplete transmission is canceled. No further sending takes place afterward (`SendXAct` = 0).

The `IDX` parameter is the connection number from the configuration of the S7 connection in "NetPro".

The `R_IDX` parameter is any number for each S7 connection (`IDX`) that is unique and identical at the corresponding send and receive blocks.

The addressing parameters `IDX` and `R_IDX` are only evaluated for the first call of the block (the actual parameters or the pre-defined values from the instance). With the first call, the communication relationship (connection) to the remote partner is defined until the next restart

(warm start) or cold start. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Note

Error code or incorrect data at the data block

Only one Rcv block may be assigned to each Snd block. The combination of `ID` (connection number) and `R_ID` (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

The block detects the current status of the redundant system using the SFC 51 "DSYSST" and passes the status to the receiver. The call for SFC 51 is activated and deactivated a the `RdSysStEn` I/O.

When `RdSysStEn` = 1, the following data are transferred:

- `RdSysStEnOut`: RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active
- `RedCPUOut`: Controller: 1 = Red. Controller, 0 = Single controller
- `Snd1Mstr`: 1 = CPU rack 0 is master (red. CPU)
- `Snd1Run`: 1 = CPU rack 0 in RUN (red. CPU)
- `Snd2Mstr`: 1 = CPU rack 1 is master (red. CPU)
- `Snd2Run`: 1 = CPU rack 1 in RUN (red. CPU)
- `SyncLink`: 1 = Synchronization possible (red. CPU)
- `ERR` : 1 = Internal error active (red. CPU)
- `ErrCode`: 1 = Internal error active (red. CPU)

The `RedCPU` I/O is needed when the block is used for communication between S7-400H (redundant) and S7-400 (single). In this case, when the call is made in the S7-400H the `RedCPU` I/O = 1, in the S7-400 communication partner the `RedCPU` I/O = 0.

The I/O `RedCPU` is always 1 for communication between S7-400H (redundant) and S7-300/ S7-1500.

Configuration

The block is installed in the CFC editor in a cyclic interrupt OB (OB30 to OB38).

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter `Status`

This block does not have the `Status` parameter

11.3.1.2 SndH_DigVal operating modes

This block has no operating mode.

11.3.1.3 Function of SndH_DigVal

The block does not have any configurable characteristics.

11.3.1.4 SndH_DigVal error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 12 "BSEND".

The `SendErr(ERROR)` and `SendStat(STATUS)` outputs show the specific error information for the lower-level SFB 12 "BSEND".

You can find the error information for the `ERROR` and `STATUS` outputs of the SFB 12 in the manual "System Software for S7-300/400 - System and Standard Functions".

If an error occurs, a new job is automatically triggered with the current data until all data have been successfully transferred.

11.3.1.5 SndH_DigVal messaging

No message behavior is applicable to this block.

11.3.1.6 SndH_DigVal I/Os

Input parameters

Parameter	Description	Type	Initial value
`EN`	1 = Called block is executed	BOOL	1
`ID1`	ID of physical connection 1	WORD	16#0000
`R_ID1`	Message frame connection ID 1	DWORD	16#00000000
`ID2`	ID of physical connection 2	WORD	16#0000
`R_ID2`	Message frame connection ID 2	DWORD	16#00000000
`RdSysStEn`	RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active	BOOL	1
`RedCPU`	Controller: 1 = Red. Controller, 0 = Single controller	BOOL	1

Parameter	Description	Type	Initial value
SndEn	1 = Activate continuous sending	BOOL	1
PV_In00 ... PV_In127	Input_0 ... Input_127	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
RdSysStEnOut	RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active	BOOL	0
RedCPUOut	Controller: 1 = Red. Controller, 0 = Single controller	BOOL	0
Snd1Mstr	1 = CPU rack 0 is master (red. CPU)	BOOL	0
Snd1Run	1 = CPU rack 0 in RUN (red. CPU)	BOOL	0
Snd1Act	1 = Command is active for connection 1	BOOL	0
Snd1Done	1 = Command is executed for connection 1	BOOL	0
Snd1Err	1= Command is executed with error for connection 1	BOOL	0
Snd1Stat	Type of error for connection 1	WORD	16#0000
Snd2Mstr	1 = CPU rack 1 is master (red. CPU)	BOOL	0
Snd2Run	1 = CPU rack 1 in RUN (red. CPU)	BOOL	0
Snd2Act	1 = Command is active for connection 2	BOOL	0
Snd2Done	1 = Command is executed for connection 2	BOOL	0
Snd2Err	1= Command is executed with error for connection 2	BOOL	0
Snd2Stat	Type of error for connection 2	WORD	16#0000
SyncLink	1 = Synchronization possible (red. CPU)	BOOL	0
ERR	1 = Internal error active (red. CPU)	BOOL	0
ErrCode	Error numbers of RDSYSST (red. CPU)	WORD	16#0000

11.3.1.7 SndH_DigVal block diagram

A block diagram is not provided for this block.

11.3.2 SndH_AnaVal - Send analog values

11.3.2.1 SndH_AnaVal description

Object name (type + number) and family

Type + number: FB 1172

Family: COMM

Number of process objects: 0 PO

Application area for SndH_AnaVal

The "SndH_AnaVal" block represents a simple user interface to SFB 12 "BSEND". The "SndH_AnaVal" block sends 32 analog structures via MPI, PROFIBUS or Industrial Ethernet connection to another CPU. This CPU calls the function block "RcvH_AnaVal" (FB 1173) or "S7RcvHAna"" (FB 337) of the PCS 7 Industry Library to receive the data.

The blocks are used to create a connection between a redundant and non-redundant automation system.

How it works

Sending is initiated by calling the block with the value 1 at the `SendEn` control input. The data are sent according to the settings of `CylMin`, `CylMax` or `PV_In_HysXX`. During the transmission of data, `SendXAct` is set to 1.

If the job is completed without error, `SendXDone` = 1 is set. `SendXErr` is set to 1 is an error occurs and new job is automatically triggered with the current data until all data have been successfully sent.

If the input `SendEn` = 0, incomplete transmission is canceled. No further sending takes place afterward (`SendXAct` = 0).

You can use `CylMIN` to set the number of cycles to wait before triggering the transfer of the current input data, regardless of any changes to one or more values.

You can use `CylMAX` to specify how many cycles after the last valid data transfer to wait before sending the current input data. Sending is executed even when no value has changed or when the changes of a REAL value are within the hysteresis `PV_In_HysXX`.

The default value of the hysteresis is `PV_In_HysXX` = 0. If sending is not to be performed after each value change, configure the `PV_In_HysXX` input accordingly. This involves an absolute value.

When `CylMax` is reached, transfer is forced after the corresponding number of cycles.

The `IDX` parameter is the connection number from the configuration of the S7 connection in "NetPro".

The `R_IDX` parameter is any number for each S7 connection (`IDX`) that is unique and identical at the corresponding send and receive blocks.

The addressing parameters IDX and R_IDX are only evaluated for the first call of the block (the actual parameters or the pre-defined values from the instance). With the first call, the communication relationship (connection) to the remote partner is defined until the next restart (warm start) or cold start. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Note

Error code or incorrect data at the data block

Only one Rcv block may be assigned to each Snd block. The combination of ID (connection number) and R_ID (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

The block detects the current status of the redundant system using the SFC 51 "DSYSST" and passes the status to the receiver. The call for SFC 51 is activated and deactivated a the RdSysStEn I/O.

When RdSysStEn = 1, the following data are transferred:

- RdSysStEnOut: RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active
- RedCPUOut: Controller: 1 = Red. Controller, 0 = Single controller
- Snd1Mstr: 1 = CPU rack 0 is master (red. CPU)
- Snd1Run: 1 = CPU rack 0 in RUN (red. CPU)
- Snd2Mstr: 1 = CPU rack 1 is master (red. CPU)
- Snd2Run: 1 = CPU rack 1 in RUN (red. CPU)
- SyncLink: 1 = Synchronization possible (red. CPU)
- ERR : 1 = Internal error active (red. CPU)
- ErrCode: 1 = Internal error active (red. CPU)

The RedCPU I/O is needed when the block is used for communication between S7-400H (redundant) and S7-400 (single). In this case, when the call is made in the S7-400H the RedCPU I/O = 1, in the S7-400 communication partner the RedCPU I/O = 0.

The I/O RedCPU is always 1 for communication between S7-400H (redundant) and S7-300/ S7-1500.

Configuration

The block is installed in the CFC editor in a cyclic interrupt OB (OB30 to OB38).

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter `Status`

This block does not have the `Status` parameter

11.3.2.2 SndH_AnaVal operating modes

This block has no operating mode.

11.3.2.3 Function of SndH_AnaVal

The block does not have any configurable characteristics.

11.3.2.4 SndH_AnaVal error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 12 "BSEND".

The `SendErr(ERROR)` and `SendStat(STATUS)` outputs show the specific error information for the lower-level SFB 12 "BSEND".

You can find the error information for the `ERROR` and `STATUS` outputs of the SFB 12 in the manual "System Software for S7-300/400 - System and Standard Functions".

If an error occurs, a new job is automatically triggered with the current data until all data have been successfully transferred.

11.3.2.5 SndH_AnaVal messaging

No message behavior is applicable to this block.

11.3.2.6 SndH_AnaVal I/Os

Input parameters

Parameter	Description	Type	Initial value
`EN`	1 = Called block is executed	BOOL	1
`ID1`	ID of physical connection 1	WORD	16#0000
`R_ID1`	Message frame connection ID 1	DWORD	16#00000000
`ID2`	ID of physical connection 2	WORD	16#0000
`R_ID2`	Message frame connection ID 2	DWORD	16#00000000

Parameter	Description	Type	Initial value
RdSysStEn	RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active	BOOL	1
RedCPU	Controller: 1 = Red. Controller, 0 = Single controller	BOOL	1
SndEn	1 = Activate continuous sending	BOOL	1
CylMin	Minimum waiting cycle	INT	1
CylMax	Maximum waiting cycle	INT	10
PV_In_Hys00 ... PV_In_Hys31	Hysteresis for PV_In_00 ... Hysteresis for PV_In_31	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
PV_In00 ... PV_In31	Input_00 ... Input_31	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
RdSysStEnOut	RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active	BOOL	0
RedCPUOut	Controller: 1 = Red. Controller, 0 = Single controller	BOOL	0
Snd1Mstr	1 = CPU rack 0 is master (red. CPU)	BOOL	0
Snd1Run	1 = CPU rack 0 in RUN (red. CPU)	BOOL	0
Snd1Act	1 = Command is active for connection 1	BOOL	0
Snd1Done	1 = Command is executed for connection 1	BOOL	0
Snd1Err	1= Command is executed with error for connection 1	BOOL	0
Snd1Stat	Type of error for connection 1	WORD	16#00000
Snd2Mstr	1 = CPU rack 1 is master (red. CPU)	BOOL	0
Snd2Run	1 = CPU rack 1 in RUN (red. CPU)	BOOL	0
Snd2Act	1 = Command is active for connection 2	BOOL	0
Snd2Done	1 = Command is executed for connection 2	BOOL	0
Snd2Err	1= Command is executed with error for connection 2	BOOL	0
Snd2Stat	Type of error for connection 2	WORD	16#00000
SyncLink	1 = Synchronization possible (red. CPU)	BOOL	0
ERR	1 = Internal error active (red. CPU)	BOOL	0
ErrCode	Error numbers of RDSYSST (red. CPU)	WORD	16#0000

11.3.2.7 SndH_AnaVal block diagram

A block diagram is not provided for this block.

11.3.3 RcvH_DigVal - Receive digital values

11.3.3.1 RcvH_DigVal description

Object name (type + number) and family

Type + number: FB 1171

Family: COMM

Number of process objects: 0 PO

Application area for RcvH_DigVal

The "RcvH_DigVal" block represents a simple user interface to SFB 13 "BRCV". The block receives 128 digital structures via MPI, PROFIBUS or Industrial Ethernet connection from another CPU. This CPU must call the function block "SndH_DigVal" (FB 1170) or "S7SndHDig" (FB 334) of the PCS 7 Industry Library to send the data.

In STEP 7, a uniform connection must be configured for both communication partners and transferred to the AS.

The blocks are used to create a connection between a redundant and non-redundant automation system.

How it works

Data are entered in the data block asynchronously to user-program execution. After "RcvH_DigVal" has been called, instance DB data may not be processed as long as the job is in progress (`RcvXNewData` = 0). If the job is completed without errors, the `RcvXNewData` output is set to 1 for one cycle. In the next cycle, the receive enable is automatically sent by the FB to the operating system of the CPU.

The receive enable can take effect prior to the arrival of the first receive job. In this case, the receive enable is stored by the operating system.

The `IDX` parameter is the connection number from the configuration of the S7 connection in "NetPro".

The `R_IDX` parameter is any number for each S7 connection (`IDX`) that is unique and identical at the corresponding send and receive blocks.

The addressing parameters `IDX` and `R_IDX` are only evaluated for the first call of the block (the actual parameters or the pre-defined values from the instance). With the first call, the communication relationship (connection) to the remote partner is defined until the next restart (warm start) or cold start. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Call the block pair consisting of the send (RcvH_DigVal) and receive block (SndH_DigVal / S7SndHDig) in each program cycle (cyclically or also via timeout alarm).

Note

Error code or incorrect data at the data block

Only one Snd block may be assigned to each Rcv block. The combination of `ID` (connection number) and `R_ID` (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

If you want to simulate the values, enable simulation via the `SimOn` input. In this case, the values from the inputs `SimPVX` are written to the outputs `PV_OutX`. As long as simulation is running, the data sent are not applied, since data reception is switched off.

Configuration

The block is installed in the CFC editor in a cyclic interrupt OB (OB30 to OB38).

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter `Status`

This block does not have the `Status` parameter

11.3.3.2 RcvH_DigVal operating modes

This block has no operating mode.

11.3.3.3 Function of RcvH_DigVal

Configurable reactions using the `Feature` parameter

The following function is available at the `Feature` I/O for this block:

- Bit 29: Output substitute values if raw value is invalid

11.3.3.4 RcvH_DigVal error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 13 "BRCV".

The `SendErr` (`ERROR`) and `SendStat` (`STATUS`) outputs show the specific error information for the lower-level SFB 13 "BRCV".

You can find additional information in the manual System Software for S7-300/400 - System and Standard Functions. There you can find a description for the `RcvXErr` (ERR) and `RcvXStat` (STAT) outputs of SFB 13.

If Bit29 in the `Feature` structure is set, and no new data is received after the expiration of `RcvMonCyc` (number of cycles), the substitute value is output and the `RcvMonErr` output is set to 1. While `RcvMonCyc` cycles are running, the last received values are applied to the output.

If Bit29 in the `Feature` structure is not set and an error occurs, the last valid values are always applied to the output.

11.3.3.5 RcvH_DigVal messaging

No message behavior is applicable to this block.

11.3.3.6 RcvH_DigVal I/Os

Input parameters

Parameter	Description	Type	Initial value
`EN`	1 = Called block is executed	BOOL	1
`ID1`	ID of physical connection 1	WORD	16#0000
`R_ID1`	Message frame connection ID 1	DWORD	16#00000000
`ID2`	ID of physical connection 2	WORD	16#0000
`R_ID2`	Message frame connection ID 2	DWORD	16#00000000
`RcvMonCyc`	Number of cycles for receiving monitoring error	INT	3
`SimOn`	1= simulation active	BOOL	0
`SimPV0 ... SimPV127`	Simulation value	BOOL	0
`SubsPV1 ... SubsPV127`	Substitute value	BOOL	0
`Feature`	I/O for additional functions (Page 533)	STRUCT	-
	• Reserved	• Bit0 ... Bit28: BOOL	• 0
	• 1 = substitute value	• Bit29: BOOL	• 0
	• Reserved	• Bit30 ... Bit31: BOOL	• 0

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
PV_Out0 ... PVOut127	Output_0 ... Output_1277	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
SimAct	1= simulation active	BOOL	0
RdSysStEnOut	RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active	BOOL	0
RedCPUOut	Controller: 1 = Red. Controller, 0 = Single controller	BOOL	0
Rcv1Mstr	1 = CPU rack 0 is master (red. CPU)	BOOL	0
Rcv1Run	1 = CPU rack 0 in RUN (red. CPU)	BOOL	0
Rcv1MonErr	1= No data received for connection 1	BOOL	0
Rcv1NewData	1= No data received for connection 1	BOOL	0
Rcv1Err	1= Command is executed with error for connection 1	BOOL	0
Rcv1Stat	Type of error for connection 1	WORD	16#0000
Rcv2Mstr	1 = CPU rack 1 is master (red. CPU)	BOOL	0
Rcv2Run	1 = CPU rack 1 in RUN (red. CPU)	BOOL	0
Rcv2MonErr	1= No data received for connection 2	BOOL	0
Rcv2NewData	1= No data received for connection 2	BOOL	0
Rcv2Err	1= Command is executed with error for connection 2	BOOL	0
Rcv2Stat	Type of error for connection 2	WORD	16#0000
SyncLink	1 = Synchronization possible (red. CPU)	BOOL	0
ERR	1 = Internal error active (red. CPU)	BOOL	0

11.3.3.7 RcvH_DigVal block diagram

A block diagram is not provided for this block.

11.3.4 RcvH_AnaVal - Receive analog values

11.3.4.1 RcvH_AnaVal description

Object name (type + number) and family

Type + number: FB 1173

Family: COMM

Number of process objects: 0 PO

Application area for RcvH_AnaVal

The "RcvH_AnaVal" block represents a simple user interface to SFB 12 "BRCV". The block sends 32 analog structures via MPI, PROFIBUS or Industrial Ethernet connection to another CPU. This CPU must call the function block "SndH_AnaVal" (FB 1172) or "S7SndHAna" (FB 336) of the PCS 7 Industry Library to send the data.

In STEP 7, a uniform connection must be configured for both communication partners and transferred to the AS.

The blocks are used to create a connection between a redundant and non-redundant automation system.

How it works

Data are entered in the data block asynchronously to user-program execution. After "RcvH_AnaVal" has been called, instance DB data may not be processed as long as the job is in progress (`RcvXNewData` = 0). If the job is completed without errors, the `RcvXNewData` output is set to 1 for one cycle. In the next cycle, the receive enable is automatically sent by the FB to the operating system of the CPU.

The receive enable can take effect prior to the arrival of the first receive job. In this case, the receive enable is stored by the operating system.

The `IDX` parameter is the connection number from the configuration of the S7 connection in "NetPro".

The `R_IDX` parameter is any number for each S7 connection (`IDX`) that is unique and identical at the corresponding send and receive blocks.

The addressing parameters `IDX` and `R_IDX` are only evaluated for the first call of the block (the actual parameters or the pre-defined values from the instance). With the first call, the communication relationship (connection) to the remote partner is defined until the next restart (warm start) or cold start. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Call the block pair consisting of the send (RcvH_AnaVal) and receive block (SndH_AnaVal / S7SndHAna) in each program cycle (cyclically or also via timeout alarm).

Note

Error code or incorrect data at the data block

Only one Snd block may be assigned to each Rcv block. The combination of `ID` (connection number) and `R_ID` (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

If you want to simulate the values, enable simulation via the `SimOn` input. In this case, the values from the inputs `SimPVX` are written to the outputs `PV_OutX`. As long as simulation is running, the data sent are not applied, since data reception is switched off.

Configuration

The block is installed in the CFC editor in a cyclic interrupt OB (OB30 to OB38).

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter Status

This block does not have the Status parameter

11.3.4.2 RcvH_AnaVal operating modes

This block has no operating mode.

11.3.4.3 Function of RcvH_AnaVal

Responses that can be configured with the "Feature" parameter

The following function is available at the Feature I/O for this block:

- Bit 29: Output substitute values if raw value is invalid

11.3.4.4 RcvH_AnaVal error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 13 "BRCV".

The SendErr (ERROR) and SendStat (STATUS) outputs show the specific error information for the lower-level SFB 13 "BRCV".

You can find additional information in the manual System Software for S7-300/400 - System and Standard Functions. There you can find a description for the RcvXErr (ERR) and RcvXStat (STAT) outputs of SFB 13.

If Bit29 in the Feature structure is set, and no new data is received after the expiration of RcvMonCyc (number of cycles), the substitute value is output and the RcvMonErr output is set to 1. While RcvMonCyc cycles are running, the last received values are applied to the output.

If Bit29 in the Feature structure is not set and an error occurs, the last valid values are always applied to the output.

11.3.4.5 RcvH_AnaVal messaging

No message behavior is applicable to this block.

11.3.4.6 RcvH_AnaVal I/Os

Input parameters

Parameter	Description	Type	Initial value
EN	1 = Called block is executed	BOOL	1
ID1	ID of physical connection 1	WORD	16#0000
R_ID1	Message frame connection ID 1	DWORD	16#00000000
ID2	ID of physical connection 2	WORD	16#0000
R_ID2	Message frame connection ID 2	DWORD	16#00000000
RcvMonCyc	Number of cycles for receiving monitoring error	INT	3
SimOn	1 = Simulation active	BOOL	0
SimPV0...SimPV31	Simulation value	REAL	0.0
SubsPV0...SubsPV31	Substitute value	REAL	0.0
Feature	I/O for additional functions (Page 537) • Reserved • 1 = substitute value • Reserved	STRUCT • Bit0 ... Bit28: BOOL • Bit29: BOOL • Bit30 ... 31: BOOL	- • 0 • 0 • 0

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
PV_Out0 ... PV_Out31	Output_0 ... Output_31	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
SimAct	1 = Simulation active	REAL	0.0
RdSysStEnOut	RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active	BOOL	0
RedCPUOut	Controller: 1 = Red. Controller, 0 = Single controller	BOOL	0
Rcv1Mstr	1 = CPU rack 0 is master (red. CPU)	BOOL	0
Rcv1Run	1 = CPU rack 0 in RUN (red. CPU)	BOOL	0
Rcv1MonErr	1= No data received for connection 1	BOOL	0
Rcv1NewData	1= No data received for connection 1	BOOL	0
Rcv1Err	1= Command is executed with error for connection 1	BOOL	0
Rcv1Stat	Type of error for connection 1	WORD	16#0000
Rcv2Mstr	1 = CPU rack 1 is master (red. CPU)	BOOL	0
Rcv2Run	1 = CPU rack 1 in RUN (red. CPU)	BOOL	0
Rcv2MonErr	1= No data received for connection 2	BOOL	0

Parameter	Description	Type	Initial value
Rcv2NewData	1= No data received for connection 2	BOOL	0
Rcv2Err	1= Command is executed with error for connection 2	BOOL	0
Rcv2Stat	Type of error for connection 2	WORD	16#0000
SyncLink	1 = Synchronization possible (red. CPU)	BOOL	0
ERR	1 = Internal error active (red. CPU)	BOOL	0

11.3.4.7 RcvH_AnaVal block diagram

A block diagram is not provided for this block.

11.3.5 S7SndHDig - Send digital values

11.3.5.1 S7SndHDig description

Object name (type + number) and family

Type + number: FB 334

Family: COMM

Number of process objects: 0 PO

Application area for S7SndHDig

The "S7SndHDig" block represents a simple user interface to SFB 12 "BSEND". The "S7SndHDig" block sends 128 digital structures via an MPI, PROFIBUS or Industrial Ethernet connection to another CPU. This CPU calls the function block "RcvH_DigVal" (FB 1171) of the PCS 7 Industry Library to receive the data.

The blocks are used to create a connection between a redundant and non-redundant automation system.

How it works

Sending is initiated by calling the block with the value 1 at the SendEn control input. During the transmission of data, SendXAct is set to 1.

If the job is completed without error, SendXDone = 1 is set. SendXErr is set to 1 is an error occurs and new job is automatically triggered with the current data until all data have been successfully sent.

If the input SendEn = 0, incomplete transmission is canceled. No further sending takes place afterward (SendXAct = 0).

The IDX parameter is the connection number from the configuration of the S7 connection in "NetPro" (PCS 7 / STEP 7) or "Devices and Networks" (TIA Portal).

The R_IDX parameter is any number for each S7 connection (IDX) that is unique and identical at the corresponding send and receive blocks.

You can re-configure the addressing parameters IDX and R_IDX in runtime. The new parameters take effect with each new job once the previous job is completed. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Note

Error code or incorrect data at the data block

Only one Rcv block may be assigned to each Snd block. The combination of ID (connection number) and R_ID (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor.

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter Status

This block does not have the Status parameter

11.3.5.2 S7SndHDig operating modes

This block has no operating mode.

11.3.5.3 Function of S7SndHDig

The block does not have any configurable characteristics.

11.3.5.4 S7SndHDig error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 12 "BSEND".

The SendErr(ERROR) and SendStat(STATUS) outputs show the specific error information for the lower-level SFB 12 "BSEND".

You can find the error information for the ERROR and STATUS outputs of the SFB 12 in the manual "System Software for S7-300/400 - System and Standard Functions".

If an error occurs, a new job is automatically triggered with the current data until all data have been successfully transferred.

11.3.5.5 S7SndHDig messaging

No message behavior is applicable to this block.

11.3.5.6 S7SndHDig I/Os

Input parameters

Parameter	Description	Type	Initial value
EN	1 = Called block is executed	BOOL	1
ID1	ID of physical connection 1	WORD	16#0000
R_ID1	Message frame connection ID 1	DWORD	16#00000000
ID2	ID of physical connection 2	WORD	16#0000
R_ID2	Message frame connection ID 2	DWORD	16#00000000
SndEn	1 = Activate continuous sending	BOOL	1
PV_In00 ... PV_In127	Input_0 ... Input_127	BOOL	1

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
Snd1Act	1 = Command is active for connection 1	BOOL	0
Snd1Done	1 = Command is executed for connection 1	BOOL	0
Snd1Err	1= Command is executed with error for connection 1	BOOL	0
Snd1Stat	Type of error for connection 1	WORD	16#0000
Snd2Act	1 = Command is active for connection 2	BOOL	0
Snd2Done	1 = Command is executed for connection 2	BOOL	0
Snd2Err	1= Command is executed with error for connection 2	BOOL	0
Snd2Stat	Type of error for connection 2	WORD	16#0000

11.3.5.7 S7SndHDig block diagram

A block diagram is not provided for this block.

11.3.6 S7SndHAna - Send analog values

11.3.6.1 S7SndHAna description

Object name (type + number) and family

Type + number: FB 336

Family: COMM

Number of process objects: 0 PO

Application area for S7SndHAna

The "S7SndHAna" block represents a simple user interface to SFB 12 "BSEND". The "S7SndHAna" block sends 32 analog structures via MPI, PROFIBUS or Industrial Ethernet connection to another CPU. This CPU calls the function block "RcvH_AnaVal" (FB 1173) of the PCS 7 Industry Library to receive the data.

The blocks are used to create a connection between a redundant and non-redundant automation system.

How it works

Sending is initiated by calling the block with the value 1 at the `SendEn` control input. The data are sent according to the settings of `CylMin`, `CylMax` or `PV_In_HysXX`. During the transmission of data, `SendXAct` is set to 1.

If the job is completed without error, `SendXDone` = 1 is set. `SendXErr` is set to 1 is an error occurs and new job is automatically triggered with the current data until all data have been successfully sent.

If the input `SendEn` = 0, incomplete transmission is canceled. No further sending takes place afterward (`SendXAct` = 0).

You can use `CylMIN` to set the number of cycles to wait before triggering the transfer of the current input data, regardless of any changes to one or more values.

You can use `CylMAX` to specify how many cycles after the last valid data transfer to wait before sending the current input data. Sending is executed even when no value has changed or when the changes of a REAL value are within the hysteresis `PV_In_HysXX`.

The default value of the hysteresis is `PV_In_HysXX` = 0. If sending is not to be performed after each value change, configure the `PV_In_HysXX` input accordingly. This involves an absolute value.

When `CylMax` is reached, transfer is forced after the corresponding number of cycles.

The IDX parameter is the connection number from the configuration of the S7 connection in "NetPro" (PCS 7 / STEP 7) or "Devices and Networks" (TIA Portal).

The R_IDX parameter is any number for each S7 connection (IDX) that is unique and identical at the corresponding send and receive blocks.

You can re-configure the addressing parameters IDX and R_IDX in runtime. The new parameters take effect with each new job once the previous job is completed. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Note

Error code or incorrect data at the data block

Only one Rcv block may be assigned to each Snd block. The combination of ID (connection number) and R_ID (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor.

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter Status

This block does not have the Status parameter

11.3.6.2 S7SndHAna operating modes

This block has no operating mode.

11.3.6.3 Function of S7SndHAna

The block does not have any configurable characteristics.

11.3.6.4 S7SndHAna error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 12 "BSEND".

The `SendErr(ERROR)` and `SendStat(STATUS)` outputs show the specific error information for the lower-level SFB 12 "BSEND".

You can find the error information for the `ERROR` and `STATUS` outputs of the SFB 12 in the manual "System Software for S7-300/400 - System and Standard Functions".

If an error occurs, a new job is automatically triggered with the current data until all data have been successfully transferred.

11.3.6.5 S7SndHAna messaging

No message behavior is applicable to this block.

11.3.6.6 S7SndHAna I/Os

Input parameters

Parameter	Description	Type	Initial value
`EN`	1 = Called block is executed	BOOL	1
`ID1`	ID of physical connection 1	WORD	16#0000
`R_ID1`	Message frame connection ID 1	DWORD	16#00000000
`ID2`	ID of physical connection 2	WORD	16#0000
`R_ID2`	Message frame connection ID 2	DWORD	16#00000000
`SndEn`	1 = Activate continuous sending	BOOL	1
`CylMin`	Minimum waiting cycle	INT	1
`CylMax`	Maximum waiting cycle	INT	10
`PV_In_Hys00 ... PV_In_Hys31`	Hysteresis for PV_In_00 ... Hysteresis for PV_In_31	REAL	0.0
`PV_In00 ... PV_In31`	Input_00 ... Input_31	REAL	0.0

Output parameters

Parameter	Description	Type	Initial value
`ENO`	1 = Block algorithm completed without errors	BOOL	0
`Snd1Act`	1 = Command is active for connection 1	BOOL	0
`Snd1Done`	1 = Command is executed for connection 1	BOOL	0
`Snd1Err`	1= Command is executed with error for connection 1	BOOL	0
`Snd1Stat`	Type of error for connection 1	WORD	16#00000

Parameter	Description	Type	Initial value
Snd2Act	1 = Command is active for connection 2	BOOL	0
Snd2Done	1 = Command is executed for connection 2	BOOL	0
Snd2Err	1= Command is executed with error for connection 2	BOOL	0
Snd2Stat	Type of error for connection 2	WORD	16#00000

11.3.6.7 S7SndHAna block diagram

A block diagram is not provided for this block.

11.3.7 S7RcvHDig - Receive digital values

11.3.7.1 S7RcvHDig description

Object name (type + number) and family

Type + number: FB 335

Family: COMM

Number of process objects: 0 PO

Application area for S7RcvHDig

The "S7RcvHDig" block represents a simple user interface to SFB 13 "BRCV". The block receives 128 digital structures via MPI, PROFIBUS or Industrial Ethernet connection from another CPU. This CPU must call the function block "SndH_DigVal" (FB 1170) of the PCS 7 Advanced Process Library to send the data.

In STEP 7, a uniform connection must be configured for both communication partners and transferred to the AS.

The blocks are used to create a connection between a redundant and non-redundant automation system.

How it works

Data are entered in the data block asynchronously to user-program execution. After "S7RcvHDig" has been called, instance DB data may not be processed as long as the job is in progress (RcvXNewData = 0). If the job is completed without errors, the RcvXNewData output is set to 1 for one cycle. In the next cycle, the receive enable is automatically sent by the FB to the operating system of the CPU.

The receive enable can take effect prior to the arrival of the first receive job. In this case, the receive enable is stored by the operating system.

The IDX parameter is the connection number from the configuration of the S7 connection in "NetPro" (PCS 7 / STEP 7) or "Devices and Networks" (TIA Portal).

The R_IDX parameter is any number for each S7 connection (IDX) that is unique and identical at the corresponding send and receive blocks.

You can re-configure the addressing parameters IDX and R_IDX in runtime. The new parameters take effect with each new job once the previous job is completed. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Call the block pair consisting of the send (S7RcvHDig) and receive block (SndH_DigVal) in each program cycle (cyclically or also via timeout alarm).

Note

Error code or incorrect data at the data block

Only one Snd block may be assigned to each Rcv block. The combination of ID (connection number) and R_ID (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

If you want to simulate the values, enable simulation via the SimOn input. In this case, the values from the inputs SimPVX are written to the outputs PV_OutX. As long as simulation is running, the data sent are not applied, since data reception is switched off.

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor.

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter Status

This block does not have the Status parameter

11.3.7.2 S7RcvHDig operating modes

This block has no operating mode.

11.3.7.3 Function of S7RcvHDig

Configurable reactions using the `Feature` parameter

The following function is available at the `Feature` I/O for this block:

- Bit 29: Output substitute values if raw value is invalid

11.3.7.4 S7RcvHDig error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 13 "BRCV".

The `SendErr` (`ERROR`) and `SendStat` (`STATUS`) outputs show the specific error information for the lower-level SFB 13 "BRCV".

You can find additional information in the manual System Software for S7-300/400 - System and Standard Functions. There you can find a description for the `RcvXErr` (ERR) and `RcvXStat` (STAT) outputs of SFB 13.

If Bit29 in the `Feature` structure is set, and no new data is received after the expiration of `RcvMonCyc` (number of cycles), the substitute value is output and the `RcvMonErr` output is set to 1. While `RcvMonCyc` cycles are running, the last received values are applied to the output.

If Bit29 in the `Feature` structure is not set and an error occurs, the last valid values are always applied to the output.

11.3.7.5 S7RcvHDig messaging

No message behavior is applicable to this block.

11.3.7.6 S7RcvHDig I/Os

Input parameters

Parameter	Description	Type	Initial value
`EN`	1 = Called block is executed	BOOL	1
`ID1`	ID of physical connection 1	WORD	16#0000
`R_ID1`	Message frame connection ID 1	DWORD	16#00000000
`ID2`	ID of physical connection 2	WORD	16#0000
`R_ID2`	Message frame connection ID 2	DWORD	16#00000000
`RcvMonCyc`	Number of cycles for receiving monitoring error	INT	3
`SimOn`	1= simulation active	BOOL	0
`SimPV0 ... SimPV127`	Simulation value	BOOL	0

Parameter	Description	Type	Initial value
SubsPV1 ... SubsPV127	Substitute value	BOOL	0
Feature	I/O for additional functions (Page 547)	STRUCT	-
	• Reserved	• Bit0 ... Bit28: BOOL	• 0
	• 1 = substitute value	• Bit29: BOOL	• 0
	• Reserved	• Bit30 ... Bit31: BOOL	• 0

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
PV_Out0 ... PVOut127	Output_0 ... Output_1277	BOOL	0
SimAct	1= simulation active	BOOL	0
RdSysStEnOut	RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active	BOOL	0
RedCPUOut	Controller: 1 = Red. Controller, 0 = Single controller	BOOL	0
Rcv1Mstr	1 = CPU rack 0 is master (red. CPU)	BOOL	0
Rcv1Run	1 = CPU rack 0 in RUN (red. CPU)	BOOL	0
Rcv1MonErr	1= No data received for connection 1	BOOL	0
Rcv1NewData	1= No data received for connection 1	BOOL	0
Rcv1Err	1= Command is executed with error for connection 1	BOOL	0
Rcv1Stat	Type of error for connection 1	WORD	16#0000
Rcv2Mstr	1 = CPU rack 1 is master (red. CPU)	BOOL	0
Rcv2Run	1 = CPU rack 1 in RUN (red. CPU)	BOOL	0
Rcv2MonErr	1= No data received for connection 2	BOOL	0
Rcv2NewData	1= No data received for connection 2	BOOL	0
Rcv2Err	1= Command is executed with error for connection 2	BOOL	0
Rcv2Stat	Type of error for connection 2	WORD	16#0000
SyncLink	1 = Synchronization possible (red. CPU)	BOOL	0
ERR	1 = Internal error active (red. CPU)	BOOL	0

11.3.7.7 S7RcvHDig block diagram

A block diagram is not provided for this block.

11.3.8 S7RcvHAna - Receive analog values

11.3.8.1 S7RcvHAna description

Object name (type + number) and family

Type + number: FB 337

Family: COMM

Number of process objects: 0 PO

Application area for S7RcvHAna

The "S7RcvHAna" block represents a simple user interface to SFB 12 "BRCV". The block sends 32 analog structures via MPI, PROFIBUS or Industrial Ethernet connection to another CPU. This CPU must call the function block "SndH_AnaVal" (FB 1170) of the PCS 7 Advanced Process Library to send the data.

In STEP 7, a uniform connection must be configured for both communication partners and transferred to the AS.

The blocks are used to create a connection between a redundant and non-redundant automation system.

How it works

Data are entered in the data block asynchronously to user-program execution. After "S7RcvHAna" has been called, instance DB data may not be processed as long as the job is in progress (`RcvXNewData` = 0). If the job is completed without errors, the `RcvXNewData` output is set to 1 for one cycle. In the next cycle, the receive enable is automatically sent by the FB to the operating system of the CPU.

The receive enable can take effect prior to the arrival of the first receive job. In this case, the receive enable is stored by the operating system.

The `IDX` parameter is the connection number from the configuration of the S7 connection in "NetPro" (PCS 7 / STEP 7) or "Devices and Networks" (TIA Portal).

The `R_IDX` parameter is any number for each S7 connection (`IDX`) that is unique and identical at the corresponding send and receive blocks.

You can re-configure the addressing parameters `IDX` and `R_IDX` in runtime. The new parameters take effect with each new job once the previous job is completed. You can find additional information on these parameters in the system documentation for BSEND/BRCV.

Call the block pair consisting of the send (S7RcvHAna) and receive block (SndH_AnaVal) in each program cycle (cyclically or also via timeout alarm).

Note

Error code or incorrect data at the data block

Only one Snd block may be assigned to each Rcv block. The combination of `ID` (connection number) and `R_ID` (identification number) must be unique within the project for every block pair.

Ensure that this assignment of the block pairs is correct, otherwise data will be sent to the wrong blocks.

If you want to simulate the values, enable simulation via the `SimOn` input. In this case, the values from the inputs `SimPVX` are written to the outputs `PV_OutX`. As long as simulation is running, the data sent are not applied, since data reception is switched off.

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor.

Startup characteristics

The block has no startup characteristics.

Time behavior

The block does not have any timing behavior.

Status word assignment for the parameter `Status`

This block does not have the `Status` parameter

11.3.8.2 S7RcvHAna operating modes

This block has no operating mode.

11.3.8.3 Function of S7RcvHAna

Responses that can be configured with the "Feature" parameter

The following function is available at the `Feature` I/O for this block:

- Bit 29: Output substitute values if raw value is invalid

11.3.8.4 S7RcvHAna error handling

Troubleshooting the block is restricted to the error information of the lower-level SFB 13 "BRCV".

The SendErr (ERROR) and SendStat (STATUS) outputs show the specific error information for the lower-level SFB 13 "BRCV".

You can find additional information in the manual System Software for S7-300/400 - System and Standard Functions. There you can find a description for the RcvXErr (ERR) and RcvXStat (STAT) outputs of SFB 13.

If Bit29 in the Feature structure is set, and no new data is received after the expiration of RcvMonCyc (number of cycles), the substitute value is output and the RcvMonErr output is set to 1. While RcvMonCyc cycles are running, the last received values are applied to the output.

If Bit29 in the Feature structure is not set and an error occurs, the last valid values are always applied to the output.

11.3.8.5 S7RcvHAna messaging

No message behavior is applicable to this block.

11.3.8.6 S7RcvHAna I/Os

Input parameters

Parameter	Description	Type	Initial value
EN	1 = Called block is executed	BOOL	1
ID1	ID of physical connection 1	WORD	16#0000
R_ID1	Message frame connection ID 1	DWORD	16#00000000
ID2	ID of physical connection 2	WORD	16#0000
R_ID2	Message frame connection ID 2	DWORD	16#00000000
RcvMonCyc	Number of cycles for receiving monitoring error	INT	3
SimOn	1 = Simulation active	BOOL	0
SimPV0...SimPV31	Simulation value	REAL	0.0
SubsPV0...SubsPV31	Substitute value	REAL	0.0
Feature	I/O for additional functions (Page 550) • Reserved • 1 = substitute value • Reserved	STRUCT • Bit0 ... Bit28: BOOL • Bit29: BOOL • Bit30 ... 31: BOOL	- • 0 • 0 • 0

Output parameters

Parameter	Description	Type	Initial value
ENO	1 = Block algorithm completed without errors	BOOL	0
PV_Out0 ... PV_Out31	Output_0 ... Output_31	REAL	0.0
SimAct	1 = Simulation active	REAL	0.0
RdSysStEnOut	RdSysSt: 1 = RdSysSt active, 0 = RdSysSt not active	BOOL	0
RedCPUOut	Controller: 1 = Red. Controller, 0 = Single controller	BOOL	0
Rcv1Mstr	1 = CPU rack 0 is master (red. CPU)	BOOL	0
Rcv1Run	1 = CPU rack 0 in RUN (red. CPU)	BOOL	0
Rcv1MonErr	1= No data received for connection 1	BOOL	0
Rcv1NewData	1= No data received for connection 1	BOOL	0
Rcv1Err	1= Command is executed with error for connection 1	BOOL	0
Rcv1Stat	Type of error for connection 1	WORD	16#0000
Rcv2Mstr	1 = CPU rack 1 is master (red. CPU)	BOOL	0
Rcv2Run	1 = CPU rack 1 in RUN (red. CPU)	BOOL	0
Rcv2MonErr	1= No data received for connection 2	BOOL	0
Rcv2NewData	1= No data received for connection 2	BOOL	0
Rcv2Err	1= Command is executed with error for connection 2	BOOL	0
Rcv2Stat	Type of error for connection 2	WORD	16#0000
SyncLink	1 = Synchronization possible (red. CPU)	BOOL	0
ERR	1 = Internal error active (red. CPU)	BOOL	0

11.3.8.7 S7RcvHAna block diagram

A block diagram is not provided for this block.

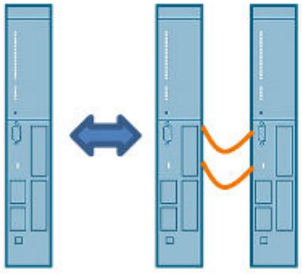
11.4 Communication between S7-41xH and S7-41x

11.4.1 SndH_DigVal, SndH_AnaVal, RcvH_DigVal, RcvH_AnaVal

Communication between S7-41xH and S7-41x

For a description of the blocks SndH_DigVal (Page 524), SndH_AnaVal (Page 528), RcvH_DigVal (Page 532) and RcvH_AnaVal (Page 535), refer to section Communication between S7-41xH and S7-31x / S7-15xx (Page 524).

Used blocks

Application	Block in S7-400	Block in S7-400H
	SndH_DigVal*	RcvH_DigVal
	SndH_AnaVal*	RcvH_AnaVal
	RcvH_DigVal	SndH_DigVal**
	RcvH_AnaVal	SndH_AnaVal**

* When call is made in S7-400, I/O `RedCPU` = 0

** When call is made in S7-400H, I/O `RedCPU` = 1

11.4.2 ASSendH - H-system communication send block

11.4.2.1 ASSendH description

Object name (type + number) and family

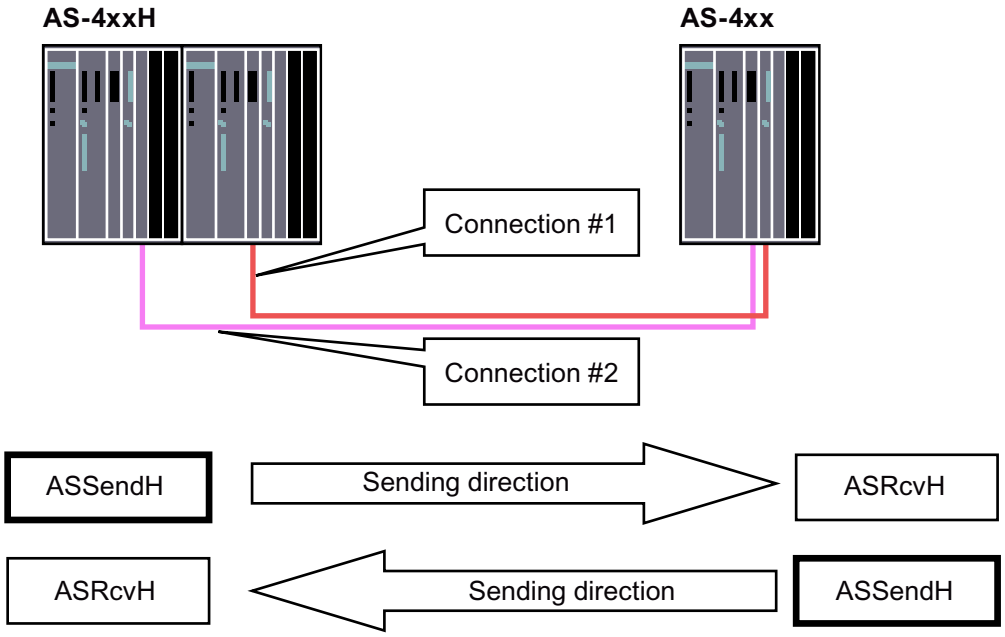
Type + number: FB 1116

Family: Comm

Number of process objects: 0 PO

Functional principles

The block coordinates the sending of telegrams between a redundant and a non-redundant automation station via S7 communication (BSEND). A maximum of 30 REAL values and 30 binary values are sent. Each value also has a binary quality code which specifies whether the measured value is correct.



Startup characteristics

The `RunUpCyc` parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Time response

The block must be called by means of a cyclic interrupt OB. The sampling time of the block is entered in the `SampleTime` parameter.

Called blocks

SFC6	RD_SINFO
SFB35	ALARM_8P
FB1151	IL_BS

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

11.4.2.2 ASSendH operating modes

The block does not have any operating modes.

11.4.2.3 ASSendH functions

Monitoring the send process

The two connections to the receiving partner are monitored separately. The block usually sends data via both connections simultaneously.

If an error is detected, a group message is sent to the OS. Following a total failure of both connections, the auxiliary value (quality flag) supplied for every value is also activated. A message is not generated until the SupprTime (suppression time) has elapsed. This parameter is adjustable.

The send error is reset when at least one telegram containing valid data has been successfully sent. If SupprTime < SampleTime, the error message is generated immediately.

11.4.2.4 ASSendH error handling

The block does not have any error handling.

11.4.2.5 ASSendH messaging

All messages can be disabled with MsgLock. When all messages are disabled (MsgLock = TRUE) or if the number of calls due after a restart (RunUpCyc) have not yet been performed, the MsgSup output is set to TRUE and no message is sent.

Messages

Message instance	Message ID	Message class	Event
MsgEvId	SIG 1	AS control system fault	$$BlockComment$$ Communication failure conn. 1
	SIG 2	AS control system fault	$$BlockComment$$ Communication failure conn. 2
	SIG 3	AS control system fault	$$BlockComment$$ Complete loss of communication
	SIG 4	-	$$BlockComment$$ External message 1
	SIG 5	-	$$BlockComment$$ External message 2
	SIG 6	-	$$BlockComment$$ External message 3
	SIG 7	-	$$BlockComment$$ External message 4
	SIG 8	-	$$BlockComment$$ External message 5

Associated values for the message instance MsgEvId

Associated value	Block parameters
1	Status1
2	Status2
3	ExtVa103
4	ExtVa104
5	ExtVa105
6	ExtVa106
7	ExtVa107
8	ExtVa108
9	ExtVa109
10	ExtVa110

The associated values (ExtVa1x) of the message block can be assigned freely.

11.4.2.6 ASSendH I/Os

Input parameters

Parameter	Description	Data type	Default
Bool01 … Bool30		STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
FirstVar	Start of send data	BOOL	1
ID_1	ID for connection #1	WORD	16#0000
ID_2	ID for connection #2	WORD	16#0000
LastVar	End of send data	BOOL	1
MsgAckEn	Evaluate Acknowledge State	BOOL	0
MsgEvId	Message ID	DWORD	16#000000D5
R_ID_1	R_ID for connection #1	DWORD	16#00000000
R_ID_2	R_ID for connection #2	DWORD	16#00000000
Real01 … Real30		STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
ResHist1	Reset the history data comm. #1	BOOL	0
ResHist2	Reset the history data comm. #2	BOOL	0
Restart	Manual restart	BOOL	0
SendEn	1:Send, 0:Don't send	BOOL	1
SendMode	0:Send once (pos.edge), 1:Send cyclic, 2:Send every n. cycle 255	BYTE	16#01
SupprTime	Suppression time [s]	REAL	10.0

Output parameters

Parameter	Description	Data type	Default
Err1Out ... Err2Out	1:Error 0:no Error (static) conn. #1 ... #2	BOOL	0
ErrCnt1 ... ErrCnt2	Counter for Errors on comm. #1 ... #2	DINT	0
Hist1St1 ... Hist1St2		WORD	16#0000
Hist1Ti1 ... Hist1Ti2		DATE_AND_TIME	1990-01-01-0:00:00
Hist2St1 ... Hist2St2		WORD	16#0000
Hist2Ti1 ... Hist2Ti2		DATE_AND_TIME	1990-01-01-0:00:00
Hist3St1 ... Hist3St2		WORD	16#0000
Hist3Ti1 ... Hist3Ti2		DATE_AND_TIME	1990-01-01-0:00:00

Parameter	Description	Data type	Default
Hist4St1 ... Hist4St2		WORD	16#0000
Hist4Ti1 ... Hist4Ti2		DATE_AND_TIME	1990-01-01-0:00:00
HistLSt1 ... HistLSt2		WORD	16#0000
HistLTi1 ... HistLTi2		DATE_AND_TIME	1990-01-01-0:00:00
IntCnt1 ... IntCnt2	Integration of send date via comm. #1 ... #2	DINT	0
Length1 ... Length2	Length of the send area conn. #1 ... #2	INT	0
MsgAckn	Message: ACK_STATE output	WORD	16#0000
MsgErr	1=Message error	BOOL	0
MsgStat	Message: STATUS output	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
Ready1... Ready2	0:Busy 1:New trigger possible conn. #1 ... #2	BOOL	0

11.4.2.7 ASSendH block diagram

The block has no block diagram.

11.4.2.8 Operator control and monitoring of ASSendH

The block does not have any views.

11.4.3 ASRcvH - H-system communication receive block

11.4.3.1 ASRcvH description

Object name (type + number) and family

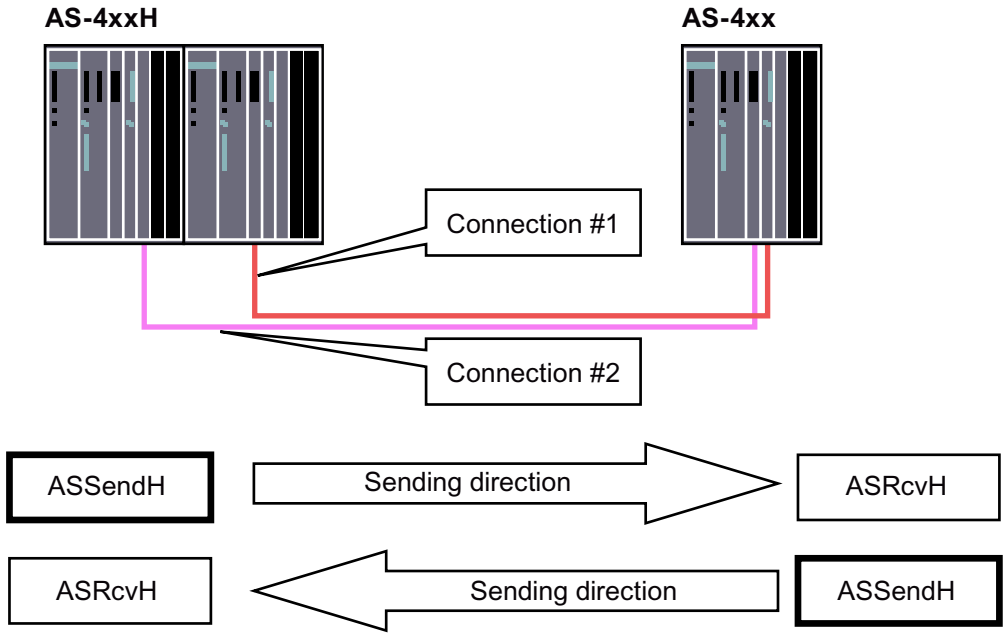
Type + number: FB 1117

Family: Comm

Number of process objects: 0 PO

Functional principles

The block coordinates the receipt of telegrams between a redundant and a non-redundant automation station via S7 communication (BRECV). A maximum of 30 REAL values and 30 binary values are sent. Each value also has a binary quality code which specifies whether the measured value is correct.



Startup characteristics

The `RunUpCyc` parameter can be used to set for how long (number of cycles) messages are to be suppressed.

`Restart = TRUE` can be used to simulate a restart.

Time response

The block must be called by means of a cyclic interrupt OB. The sampling time of the block is entered in the `SampleTime` parameter.

Called blocks

SFC6	RD_SINFO
SFB35	ALARM_8P
FB1152	IL_BR

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

11.4.3.2 ASRcvH operating modes

The block does not have any operating modes.

11.4.3.3 ASRcvH functions

Monitoring the receive process

The two connections to the sending partner are monitored separately.

If the process is functioning without errors, the receive data is transferred from the first connection; if one connection fails, the data is transferred from whichever of the two connections still exists.

If an error is detected, a group message is sent to the OS. Following a total failure of both connections, the auxiliary value (quality flag) supplied for every value is also activated. A message is not generated until the `SupprTime` (suppression time) has elapsed. This parameter is adjustable.

The send error is reset when at least one telegram containing valid data has been successfully sent. If `SupprTime` < `SampleTime`, the error message is generated immediately.

11.4.3.4 ASRcvH error handling

The block does not have any error handling.

11.4.3.5 ASRcvH messaging

All messages can be disabled with `MsgLock`. When all messages are disabled (`MsgLock = TRUE`) or if the number of calls due after a restart (`RunUpCyc`) have not yet been performed, the `MsgSup = TRUE` output is set and no message is sent.

Messages

Message instance	Message ID	Message class	Event
MsgEvId	SIG 1	AS control system fault	$$BlockComment$$ Communication failure conn. 1
	SIG 2	AS control system fault	$$BlockComment$$ Communication failure conn. 2
	SIG 3	AS control system fault	$$BlockComment$$ Complete loss of communication
	SIG 4	-	$$BlockComment$$ External message 1
	SIG 5	-	$$BlockComment$$ External message 2

Message instance	Message ID	Message class	Event
	SIG 6	-	$$BlockComment$$ External message 3
	SIG 7	-	$$BlockComment$$ External message 4
	SIG 8	-	$$BlockComment$$ External message 5

Associated values for the message instance MsgEvId

Associated value	Block parameters
1	Status1
2	Status2
3	ExtVa103
4	ExtVa104
5	ExtVa105
6	ExtVa106
7	ExtVa107
8	ExtVa108
9	ExtVa109
10	ExtVa110

The associated values (ExtVa1x) of the message block can be assigned freely.

11.4.3.6 ASRcvH I/Os

Input parameters

Parameter	Description	Data type	Default
Bool01_1In … Bool030_1In		STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Bool01_2In … Bool030_2In		STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Dummy		WORD	16#0000
FirstVar1 ... FirstVar2	Start of receive data conn. #1 ... #2	BOOL	1
ID_1 ... ID_2	ID for connection #1 ... #2	WORD	16#0000
LastVar1 ... LastVar2	End of receive data	BOOL	1
MsgAckEn	Evaluate Acknowledge State	BOOL	0

Parameter	Description	Data type	Default
MsgEvId	Message ID	DWORD	16#000000D4
R_ID_1 ... R_ID_2	R_ID for connection #1 ... #2	DWORD	16#00000000
Real01_1In … Real30_1In		STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
Real01_2In … Real30_2In		STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
ResHist1 ... ResHist2	Reset the history data comm. #1 ... #2	BOOL	0
Restart	Manual restart	BOOL	0
SupprTime	Suppression time [s]	REAL	10.0

Output parameters

Parameter	Description	Data type	Default
Bool01 … Bool30		STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Err1Out ... Err2Out	1:Error 0:no Error (static)	BOOL	0
ErrCnt1 ... ErrCnt2	Counter for Errors conn. #1 ... #2	DINT	0
Hist1Ti1 ... Hist4Ti1		DATE_AND_TIME	1990-01-01-0:00:00
Hist1St1 ... Hist4St1		WORD	16#0000
Hist1Ti2 ... Hist4Ti2		DATE_AND_TIME	1990-01-01-0:00:00
Hist1St2 ... Hist4St2		WORD	16#0000
HistLSt1 ... HistLSt2		WORD	16#0000
HistLTi1 ... HistLTi2		DATE_AND_TIME	1990-01-01-0:00:00
IntCnt1 ... IntCnt2	Integration of receive date conn. #1 ... #2	DINT	0
Length1 ... Length2	Length of the receive area	INT	0
MsgAckn	Message: ACK_STATE output	WORD	16#0000
MsgErr	1=Message error	BOOL	0
MsgStat	Message: STATUS output	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0

Parameter	Description	Data type	Default
Real01 … Real30		STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
TrigReady1 ... TrigReady2	0:Busy 1:new trigger possible	BOOL	0

11.4.3.7 ASRcvH block diagram

The block has no block diagram.

11.4.3.8 Operator control and monitoring of ASRcvH

The block does not have any views.

11.5 TCP communication blocks

11.5.1 TRcv_AnaVal - Receive analog values including the quality code

11.5.1.1 TRcv_AnaVal Description

Description

Receives up to 32 analog structures as a message frame from another controller.

11.5.1.2 TRcv_AnaVal Configuration

Parameterizable behavior

Specifies whether a substitute error is output in the event of a communication error.

Bit assignment in the "Feature" parameter

Number	Value	Effect	Standard setting
29	0	Deactivated. The last valid values are output at the output.	x
	1	Activated. The last valid values are output during a parameterizable number of cycles in `RcvMonCyc`. The parameterized substitute values are then output. The quality code of the output values is set to "manipulated value (for example, substitute value, simulation, last valid value)".	-

11.5.1.3 TRcv_AnaVal Functions

TRcv_AnaVal Controlling the simulation

Description

Activates the simulation at the block.

Relevant input parameters

Switch on simulation" command:

- `SimOn=1`

Relevant output parameters

Simulation is active:

- `SimAct=1`

TRcv_AnaVal Parameterizing substitute values

Description

Specifies the substitute values:

- Simulation is active
- Error handling

Parameterizable behavior

For more information on Parameterizable behavior, refer to Feature.Bit[29] | Output substitute value in the event of communication errors (Page 564).

Relevant input parameters

Parameterizing simulation values:

- `SimPV[0..31]`

Parameterizing substitute values in the event of communication errors:

- `SubsPV[1..31]`
- `Feature.Bit[29]=1`

Relevant output parameters

-

TRcv_AnaVal Parameterizing mode of operation

Description

Parameterizes the mode of operation.

Relevant input parameters

Parameterizing ID of physical connection:

- `ID`

Parameterize ID of message frame connection:

- `R_ID`

Relevant output parameters

New data received:

- `RcvNewData=1`

Received process values:

- `PV_Out[0..31]`

TRcv_AnaVal Parameterizing error handling

Description

Configures troubleshooting.

Parameterizable behavior

Refer to Feature.Bit[29] | Output substitute value in the event of communication errors (Page 564)

Relevant input parameters

Number of cycles until the substitute value is output.

- `RcvMonCyc`

Relevant output parameters

No data received:

- `RcvMonErr=1`

Data transfer completed with error:

- `RcvErr=1`

Type of fault:

- RcvStat

11.5.1.4 TRcv_AnaVal Error handling

Error handling

For more information on Error handling, refer to Parameterizing error handling (Page 566).

11.5.1.5 TRcv_AnaVal I/Os

Input parameters

Parameter	Description	Type	Start value
EN	1 = Called block is processed	BOOL	1
Feature	IO for additional functions Reserved 1 = Substitute value Reserved	STRUCT • Bit 0..28: BOOL • Bit 29: BOOL • Bit 30..31: BOOL	- • 0 • 0 • 0
ID	ID of physical connection	WORD	16#0000
R_ID	ID of message frame connection	DWORD	16#00000000
RcvMonCyc	Number of cycles until the substitute value is output in the event of a communication error	INT	3
SimOn	1 = Simulation is active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
SimPV[0..31]	Simulation value	REAL	0.0
SubsPV[0..31]	Substitute value	REAL	0.0

Output parameters

Parameter	Description	Type	Start value
ENO	1 = Block algorithm has been executed without errors	BOOL	0
PV_Out[0..31]	Output_0 .. Output_31	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
RcvErr	1 = Command is executed with error	BOOL	0
RcvMonErr	1 = No data received	BOOL	0

Parameter	Description	Type	Start value
RcvNewData	1 = New data received	BOOL	0
RcvStat	Type of fault	WORD	16#0000
SimAct	1 = Simulation is active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80

11.5.2 TRcv_DigVal - Receive boolean values including the quality code

11.5.2.1 TRcv_DigVal Description

Receives up to 128 digital structures as a message frame from another controller.

11.5.2.2 TRcv_DigVal Configuration

Description

Specifies whether a substitute error is output in the event of a communication error.

Bit assignment in the "Feature" parameter

Number	Value	Effect	Standard setting
29	0	Deactivated. The last valid values are output at the output.	x
	1	Activated. The last valid values are output during a parameterizable number of cycles in RcvMonCyc. The parameterized substitute values are then output. The quality code of the output values is set to "manipulated value (for example, substitute value, simulation, last valid value)".	-

11.5.2.3 TRcv_DigVal Functions

TRcv_DigVal Controlling the simulation

Description

Activates the simulation at the block.

Relevant input parameters

"Switch on simulation" command:

- `SimOn=1`

Relevant output parameters

Simulation is active:

- `SimAct=1`

TRcv_DigVal Parameterizing substitute values

Description

Specifies the substitute values:

- Simulation is active

Parameterizable behavior

Feature.Bit[29] | Output substitute value in the event of communication errors (Page 568)

Relevant input parameters

Parameterizing simulation values:

- `SimPV[0..127]`

Parameterizing substitute values:

- `SubsPV[1..127]`

Relevant output parameters

-

See also

TRcv_DigVal Description (Page 568)

TRcv_DigVal Parameterizing mode of operation

Description

Parameterizes the mode of operation.

Relevant input parameters

Parameterizing ID of physical connection:

- `ID`

Parameterize ID of message frame connection:

- `R_ID`

Relevant output parameters

New data received:

- `RcvNewData=1`

Received process values:

- `PV_Out[0..127]`

TRcv_DigVal Parameterizing error handling

Description

Configures troubleshooting.

Parameterizable behavior

Feature.Bit[29] | Output substitute value in the event of communication errors (Page 568)

Relevant input parameters

Number of cycles until the substitute value is output.

- `RcvMonCyc`

Relevant output parameters

No data received:

- `RcvMonErr=1`

Data transfer completed with error:

- `RcvErr=1`

Type of fault:

- `RcvStat`

11.5.2.4 TRcv_DigVal Error handling

For more information on error handling refer TRcv_DigVal Parameterizing error handling (Page 570)

11.5.2.5 TRcv_DigVal I/Os

Input parameters

Parameter	Description	Type	Start value
EN	1 = Called block is processed	BOOL	1
Feature	IO for additional functions Reserved 1 = Substitute value Reserved	STRUCT • Bit 0..28: BOOL • Bit 29: BOOL • Bit 30..31: BOOL	- • 0 • 0 • 0
ID	ID of physical connection	WORD	16#0000
R_ID	ID of message frame connection	DWORD	16#00000000
RcvMonCyc	Number of cycles until the substitute value is output in the event of a communication error	INT	3
SimOn	1 = Simulation is active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
SimPV[0..127]	Simulation value	BOOL	0
SubsPV[0..127]	Substitute value	BOOL	0

Output parameters

Parameter	Description	Type	Start value
ENO	1 = Block algorithm has been executed without errors	BOOL	0
PV_Out[0..127]	Output_00 .. Output_127	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
RcvErr	1 = Command is executed with error	BOOL	0
RcvMonErr	1 = No data received	BOOL	0
RcvNewData	1 = New data received	BOOL	0
RcvStat	Type of fault	WORD	16#0000
SimAct	1 = Simulation is active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80

11.5.3 TSnd_AnaVal - Send analog values including the quality code

11.5.3.1 TSnd_AnaVal Description

Transmits up to 32 analog structures as a message frame to another controller.

11.5.3.2 TSnd_AnaVal Functions

TSnd_AnaVal Parameterizing mode of operation

Description

Parameterizes the mode of operation.

Relevant input parameters

Parameterizing ID of physical connection:

- `ID`

Parameterize ID of message frame connection:

- `R_ID`

Activate continuous sending

- `SndEn=1`

Parameterizing the number of cycles until suppression of transmission:

- `CycMin`

Parameterizing the number of cycles until resumption of transmission:

- `CycMax`

Relevant output parameters

Data are sent:

- `SndAct=1`

Data transfer completed without error:

- `SndDone=1`

TSnd_AnaVal Parameterizing a process value

Description

Parameterizes the process value.

Relevant input parameters

Parameterizing a process value:

- `PV_In[0..31]`

Parameterizing EDC hysteresis

- `PV_In_Hys[00..31]`

Relevant output parameters

-

TSnd_AnaVal Parameterizing error handling

Description

Indicates whether an error information is active at least one input or output parameter.

Relevant input parameters

-

Relevant output parameters

Data transfer completed with error:

- `SndErr=1`

Type of fault:

- `SndStat`

11.5.3.3 TSnd_AnaVal Error handling

Description

In the event of a transmission error sending is repeated until transmission is successful, see TSnd_AnaVal Parameterizing error handling (Page 573).

11.5.3.4 TSnd_AnaVal I/Os

Input parameters

Parameter	Description	Type	Start value
`CycMax`	Maximum waiting cycle	INT	10
`CycMin`	Minimum waiting cycle	INT	1
`EN`	1 = Called block is processed	BOOL	1
`ID`	ID of physical connection	WORD	16#0000
`PV_In_Hys[00..31]`	EDC hysteresis PV_In0 .. EDC hysteresis PV_In31.	REAL	0.0
`PV_In[0..31]`	Input_0 .. Input_31	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80

Parameter	Description	Type	Start value
R_ID	ID of message frame connection	DWORD	16#00000000
SndEn	1 = Activate continuous sending	BOOL	1

Output parameters

Parameter	Description	Type	Start value
ENO	1 = Block algorithm has been executed without errors	BOOL	0
SndAct	1 = Command is active	BOOL	0
SndDone	1 = Command has been executed	BOOL	0
SndErr	1 = Command is executed with error	BOOL	0
SndStat	Type of fault	WORD	16#0000

11.5.4 TSnd_DigVal - Send boolean values including the quality code

11.5.4.1 TSnd_DigVal Description

Description

Transmits up to 128 digital structures as a message frame to another controller.

11.5.4.2 TSnd_DigVal Functions

TSnd_DigVal Parameterizing mode of operation

Description

Parameterizes the mode of operation.

Relevant input parameters

Parameterizing ID of physical connection:

- ID

Parameterize ID of message frame connection:

- R_ID

Activate continuous sending

- SndEn=1

Relevant output parameters

Data are sent:

- SndAct=1

Data transfer completed without error:

- SndDone=1

TSnd_DigVal Parameterizing process value

Description

Parameterizes the process value.

Relevant input parameters

Parameterizing a process value:

- PV_In[0..127]

Relevant output parameters

-

TSnd_DigVal Parameterizing error handling

Description

Indicates whether an error information is active at least one input or output parameter.

Relevant input parameters

-

Relevant output parameters

Data transfer completed with error:

- SndErr=1

Type of fault:

- SndStat

11.5.4.3 TSnd_DigVal Error handling

Description

In the event of a transmission error sending is repeated until transmission is successful.

For more information on Error handling, refer Parameterizing error handling (Page 575).

11.5.4.4 TSnd_DigVal IOs

Parameter	Description	Type	Start value
EN	1 = Called block is processed	BOOL	1
ID	ID of physical connection	WORD	16#0000
PV_In[0..127]	Input_0 .. Input_127	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
R_ID	ID of message frame connection	DWORD	16#00000000
SndEn	1 =Activate continuous sending	BOOL	1

Output parameters

Parameter	Description	Type	Start value
ENO	1 = Block algorithm has been executed without errors	BOOL	0
SndAct	1 =Command is active	BOOL	0
SndDone	1 =Command has been executed	BOOL	0
SndErr	1 =Command is executed with error	BOOL	0
SndStat	Type of fault	WORD	16#0000

11.5.5 TCPCon - TCP connection via Open TCP/IP communication

11.5.5.1 TCPCon Description

Description

Connects the blocks for sending and receiving.

11.5.5.2 TCPCon Functions

Controlling communication

Description

Switches the communications on and off.

Relevant input parameters

Establish connection:

- `Connect=TRUE`

Terminate connection:

- `Connect=FALSE`

Parameterizing connection:

- active: `Active=TRUE`
- not active: `Active=FALSE`

Relevant output parameters

The connection is established.

- `Connected=TRUE`

Connection problem:

- `Error=TRUE`

Status of the connection pending:

- `Status`

Number of entries in the Send buffer:

- `SendBuffer`

Parameterizing mode of operation

Description

Connects the blocks for sending and receiving.

Relevant input parameters

Parameterizing IP of the connection:

- `ID`

Select connection IO for connection A:

- X5: `Interface=0`
- X8: `Interface=1`
- External communications processor: `Interface=2`

Select rack for connection A:

- Rack0: `Rack=0`
- Rack1: `Rack=1`

Parameterizing slot number of the external communication processor:

- `Slot`

Parameterizing the IP address of the partner CPU:

- `IPAddress`

Parameterizing connection type:

- TCP: `ConnectionType=0`
- ISO on TCP: `ConnectionType=1`

Port number for the connection (for TCP only):

- `Port`

Parameterizing local TSAP-ID (only for ISO on TCP):

- `LocalTSAP`

Parameterizing Remote TSAP-ID (only for ISO on TCP):

- `RemoteTSAP`

Relevant output parameters

Output for connections to blocks for sending and receiving:

- `toSendRcv`

11.5.5.3 TCPCon I/Os

Input parameters

Parameter	Description	Type	Start value
`Active`	Connection is active: TRUE = Connection is active FALSE = Connection is not active	BOOL	TRUE
`Connect`	Switch the communication on/off: TRUE = Establish connection: FALSE = Terminate connection	BOOL	FALSE

Parameter	Description	Type	Start value
ConnectionType	Connection type: 0 = TCP 1 = ISO on TCP	BYTE	16#00
ID	Unique ID for each connection	WORD	16#0003
Interface	IO selection for connection A: 0 = X5 1 = X8 2 = External communications processor	BYTE	16#00
IPAddress	IP address of the partner CPU. Format: Four parts of the IP address written as a hexadecimal number. Example: 192.168.0.10 >> 16#C0A8000A	DWORD	16#00000000
LocalTSAP	Local TSAP-ID (only for ISO on TCP)	WORD	16#1003
Port	Port number for the connection (for TCP only)	INT	502
Rack	Rack selection for connection A: 0 = Rack0 1 = Rack1	BYTE	16#00
RemoteTSAP	Remote TSAP-ID (only for ISO on TCP)	WORD	16#1003
Slot	Slot number of external communications processor	BYTE	16#05

Output parameters

Parameter	Description	Type	Start value
Connected	TRUE = The connection is established.	BOOL	FALSE
Error	TRUE = Connection problem	BOOL	FALSE
SendBuffer	Number of entries in the Send buffer maximum 20	INT	0
Status	Status of the connection 0 = No problem	WORD	16#0000
toSendRcv	Output for connections to blocks for sending and receiving:	Struct_ANY	

Logic blocks 12

12.1 SelStr - Select one of two STRING values

12.1.1 SelStr description

Object name (type + number) and family

Type + number: FB 1106

Family: Logic

Number of process objects: 0 PO

Called blocks

UDT1190	UDT_BoolSt

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

See also

SelStr I/Os (Page 582)

12.1.2 SelStr operating modes

The block does not have any operating modes.

12.1.3 SelStr functions

The block selects one of two STRING values (In1 or In2) in accordance with an input (`Sel_In2`) and outputs it at the `Out08`, `Out16`, `Out32`, and `Out254` outputs. The `Out08`, `Out16`, and `Out32` outputs refer to the first 8, 16, and 32 characters respectively. `Sel_In2` is passed through at the In2Selected output.

12.1.4 SelStr error handling

The block does not have any error handling.

12.1.5 SelStr messaging

The block has no message behavior.

12.1.6 SelStr I/Os

Input parameters

Parameter	Description	Data type	Default
`In1`	Input 1	STRING	''
`In2`	Input 2	STRING	''
`Sel_In2`	Selector: 0=In1, 1=In2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
`In2Selected`	0=In1 selected, 1=In2 selected	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`Out08`	Output value	STRING[8]	''
`Out16`	Output value	STRING[16]	''
`Out254`	Output value	STRING	''
`Out32`	Output value	STRING[32]	''

12.1.7 SelStr block diagram

The block has no block diagram.

12.1.8 Operator control and monitoring of SelStr

Operator control and monitoring

The block does not have any views.

12.2 SelR - Select one of two REAL values

12.2.1 SelR description

Object name (type + number) and family

Type + number: FC 601

Family: Logic

Number of process objects: 0 PO

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

12.2.2 SelR operating modes

The block does not have any operating modes.

12.2.3 SelR functions

The block selects one of two REAL values (In1 or In2) in accordance with an input (Sel_In2) and outputs it at the Out output. Sel_In2 is passed through at the In2Selected output.

12.2.4 SelR error handling

The block does not have any error handling.

12.2.5 SelR messaging

The block has no message behavior.

12.2.6 SelR I/Os

Input parameters

Parameter	Description	Data type	Default
In1	Input 1	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
In2	Input 2	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
Sel_In2	Selector: 0=In1, 1=In2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
In2Selected	0=In1 selected, 1=In2 selected	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Out	Output value	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80

12.2.7 SelR block diagram

The block has no block diagram.

12.2.8 Operator control and monitoring of SelR

The block does not have any views.

12.3 SelB - Select one of two BOOL values

12.3.1 SelB description

Object name (type + number) and family

Type + number: FC 602

Family: Logic

Number of process objects: 0 PO

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

12.3.2 SelB operating modes

The block does not have any operating modes.

12.3.3 SelB functions

The block selects one of two Boolean values (`In1` or `In2`) in accordance with an input (`Sel_In2`) and outputs it at the `Out` output. `Sel_In2` is passed through at the `In2Selected` output.

12.3.4 SelB error handling

The block does not have any error handling.

12.3.5 SelB messaging

The block has no message behavior.

12.3.6 SelB I/Os

Input parameters

Parameter	Description	Data type	Default
In1	Input 1	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
In2	Input 2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Sel_In2	Selector: 0=In1, 1=In2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
In2Selected	0=In1 selected, 1=In2 selected	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Out	Output value	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

12.3.7 SelB block diagram

The block has no block diagram.

12.3.8 Operator control and monitoring of SelB

The block does not have any views.

12.4 SelI - Select one of two INTEGER values

12.4.1 SelI description

Object name (type + number) and family

Type + number: FC 603

Family: Logic

Number of process objects: 0 PO

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

Called blocks

The block calls no further blocks.

12.4.2 SelI operating modes

The block does not have any operating modes.

12.4.3 SelI functions

The block selects one of two INT values (`In1` or `In2`) in accordance with an input (`Sel_In2`) and outputs it at the `Out` output. `Sel_In2` is passed through at the `In2Selected` output.

12.4.4 SelI error handling

Error handling

The block does not have any error handling.

12.4.5 SelI messaging

The block has no message behavior.

12.4.6 SelI I/Os

Input parameters

Parameter	Description	Data type	Default
In1	Input 1	INT	0
In2	Input 2	INT	0
Sel_In2	Selector: 0=In1, 1=In2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
In2Selected	0=In1 selected, 1=In2 selected	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Out	Output value	INT	0

12.4.7 SelI block diagram

The block has no block diagram.

12.4.8 Operator control and monitoring of SelI

The block does not have any views.

12.5 SelByt - Select one of two BYTE values

12.5.1 SelByt description

Object name (type + number) and family

Type + number: FC 604

Family: Logic

Number of process objects: 0 PO

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

12.5.2 SelByt operating modes

The block does not have any operating modes.

12.5.3 SelByt functions

The block selects one of two BYTE values (`In1` or `In2`) in accordance with an input (`Sel_In2`) and outputs it at the `Out` output. `Sel_In2` is passed through at the `In2Selected` output.

12.5.4 SelByt error handling

The block does not have any error handling.

12.5.5 SelByt messaging

The block has no message behavior.

12.5.6 SelByt I/Os

Input parameters

Parameter	Description	Data type	Default
In1	Input 1	BYTE	16#00
In2	Input 2	BYTE	16#00
Sel_In2	Selector: 0=In1, 1=In2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
In2Selected	0=In1 selected, 1=In2 selected	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Out	Output value	BYTE	16#00

12.5.7 SelByt block diagram

The block has no block diagram.

12.5.8 Operator control and monitoring of SelByt

The block does not have any views.

12.6 SelW - Select one of two WORD values

12.6.1 SelW description

Object name (type + number) and family

Type + number: FC 605

Family: Logic

Number of process objects: 0 PO

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

12.6.2 SelW operating modes

The block does not have any operating modes.

12.6.3 SelW functions

The block selects one of two WORD values (`In1` or `In2`) in accordance with an input (`Sel_In2`) and outputs it to the `Out` output. `Sel_In2` is passed through at the `In2Selected` output.

12.6.4 SelW error handling

The block does not have any error handling.

12.6.5 SelW messaging

The block has no message behavior.

12.6.6 SelW I/Os

Input parameters

Parameter	Description	Data type	Default
In1	Input 1	WORD	16#0000
In2	Input 2	WORD	16#0000
Sel_In2	Selector: 0=In1, 1=In2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
In2Selected	0=In1 selected, 1=In2 selected	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Out	Output value	WORD	16#0000

12.6.7 SelW block diagram

The block has no block diagram.

12.6.8 Operator control and monitoring of SelW

The block does not have any views.

12.7 SelDW - Select one of two DOUBLE WORD values

12.7.1 SelDW description

Object name (type + number) and family

Type + number: FC 606

Family: Logic

Number of process objects: 0 PO

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

12.7.2 SelDW operating modes

The block does not have any operating modes.

12.7.3 SelDW functions

The block selects one of two DWORD values (`In1` or `In2`) in accordance with an input (`Sel_In2`) and outputs it at the `Out` output. `Sel_In2` is passed through at the `In2Selected` output.

12.7.4 SelDW error handling

The block does not have any error handling.

12.7.5 SelDW messaging

The block has no message behavior.

12.7.6 SelDW I/Os

Input parameters

Parameter	Description	Data type	Default
In1	Input 1		16#00000000
In2	Input 2		16#00000000
Sel_In2	Selector: 0=In1, 1=In2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
In2Selected	0=In1 selected, 1=In2 selected	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Out	Output value		16#00000000

12.7.7 SelDW block diagram

The block has no block diagram.

12.7.8 Operator control and monitoring of SelDW

The block does not have any views.

12.8 SelDI - Select one of two DOUBLE INTEGER values

12.8.1 SelDI description

Object name (type + number) and family

Type + number: FC 607

Family: Logic

Number of process objects: 0 PO

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

12.8.2 SelDI operating modes

The block does not have any operating modes.

12.8.3 SelDI functions

The block selects one of two DINT values (`In1` or `In2`) in accordance with an input (`Sel_In2`) and outputs it at the `Out` output. `Sel_In2` is passed through at the `In2Selected` output.

12.8.4 SelDI error handling

The block does not have any error handling.

12.8.5 SelDI messaging

The block has no message behavior.

12.8.6 SelDI I/Os

Input parameters

Parameter	Description	Data type	Default
In1	Input 1	DINT	0
In2	Input 2	DINT	0
Sel_In2	Selector: 0=In1, 1=In2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
In2Selected	0=In1 selected, 1=In2 selected	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Out	Output value	DINT	0

12.8.7 SelDI block diagram

The block has no block diagram.

12.8.8 Operator control and monitoring of SelDI

The block does not have any views.

12.9 SelC - Select one of two CHAR values

12.9.1 SelC description

Object name (type + number) and family

Type + number: FC 608

Family: Logic

Number of process objects: 0 PO

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

12.9.2 SelC operating modes

The block does not have any operating modes.

12.9.3 SelC functions

The block selects one of two CHAR values (`In1` or `In2`) in accordance with an input (`Sel_In2`) and outputs it at the `Out` output. `Sel_In2` is passed through at the `In2Selected` output.

12.9.4 SelC error handling

The block does not have any error handling.

12.9.5 SelC messaging

The block has no message behavior.

12.9.6 SelC I/Os

Input parameters

Parameter	Description	Data type	Default
In1	Input 1	BYTE	16#00
In2	Input 2	BYTE	16#00
Sel_In2	Selector: 0=In1, 1=In2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
In2Selected	0=In1 selected, 1=In2 selected	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Out	Output value	BYTE	16#00

12.9.7 SelC block diagram

The block has no block diagram.

12.9.8 Operator control and monitoring of SelC

The block does not have any views.

12.10 VoteAnL - Monitoring block for up to 8 analog process values

12.10.1 VoteAnL description

Object name (type + number) and family

Type + number: FB 1192

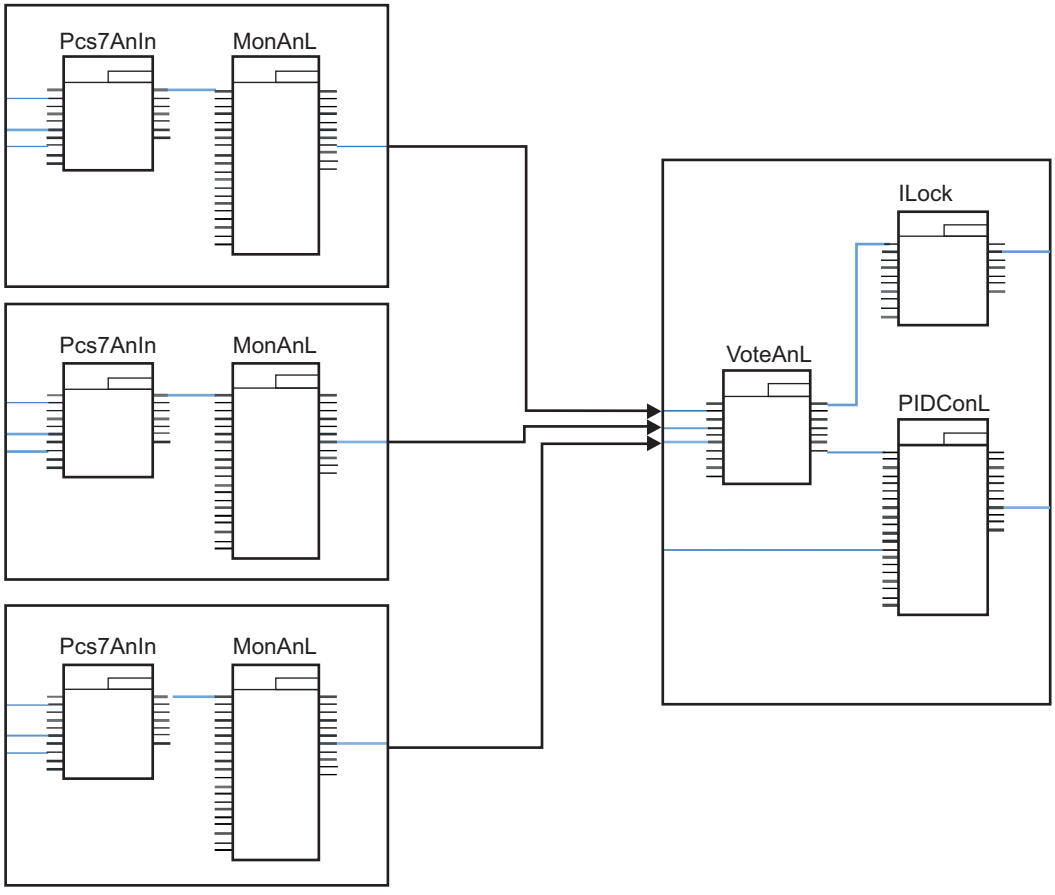
Family: LogicAn

Number of process objects: 10 PO

Area of application of VoteAnL

The block is used to increase the availability of analog process values. Up to 8 process values for signal status, limits and deviation are checked. Checking multiple process values increases availability by preventing failures caused by individual errors.

Up to 8 process values are interconnected at the function block, are shown in the figure.



Startup characteristics

Use the startup characteristics Feature bit 0 "Set startup characteristics" to define the startup characteristics for this block.

The messages are suppressed after startup for the number of cycles set at `RunUpCyc`.

The output parameters PV_AH_Out, PV_WH_Out, PV_Hys, PV_DevHysOut, PV_Dev_Out, PV_AL_Out and PV_WL_Out are specified by the corresponding input parameters PV_AH_Lim, PV_WH_Lim, PV_Hyst, PV_DevHyst, PV_Dev_Lim, PV_AL_Lim and PV_WL_Lim.

Time response

The block must be called via a cyclic interrupt OB. The sampling time of the block is set in the `SampleTime` parameter.

Called blocks

FC369	SelST16
SFC6	RD_SINFO
SFB35	ALARM_8P

Calling OBs

In the same OB with and after the block whose measured value is to be monitored. Also in OB100.

Status word allocation for the parameter `Status1`

You can find a description for each parameter in the section VoteAnL I/Os (Page 620)

Status bit	Parameter
0	Allocated
1	`BatchEn`
2	`SimOn`
3	`OosAct`.Value
4	`OosLi`.Value
5	Not used
6	`OnAct`.Value
7 - 9	Not used
10	`SimLiOp`.Value
11	Delay of the `PVx_AH_Lim` message
12	Delay of the `PVx_WH_Lim` message
13	Delay of the `PVx_Dev_Lim` message
14	`Feature.Bit15`
15	Delay of the `PVx_WL_Lim` message
16	Delay of the `PVx_AL_Lim` message

Status bit	Parameter
17	Collection of message delays
18 - 22	Not used
23	Delay of the `Out`.Value
24	Result of logic operation with inclusion of the output bypass
25	`Out_Byp_Out`.Value
26	Result of logic operation without inclusion of the output bypass
27	Pre-trip warning
28	Not used
29	All interconnected process value inputs (`PVx`) in simulation and bypass for `Out` are not active.
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word allocation for the parameter `Status2`

You can find a description for each parameter in the section VoteAnL I/Os (Page 620)

Status bit	Parameter
0	`MsgLock`
1 - 3	Not used
4	1 = Deviation voting active
5	1 = Limit voting active
6	1 = Signal status voting active
7	`PV_AH_En`
8	`PV_WH_En`
9	`PV_Dev_En`
10	Not used
11	`PV_WL_En`
12	`PV_AL_En`
13	`PV_AH_MsgEn`
14	`PV_WH_MsgEn`
15	`PV_Dev_MsgEn`
16	Not used
17	`PV_WL_MsgEn`
18	`PV_AL_MsgEn`
19 - 22	Not used
23	`PV1_AH_Act`.Value
24	`PV2_AH_Act`.Value
25	`PV3_AH_Act`.Value
26	`PV4_AH_Act`.Value
27	`PV5_AH_Act`.Value
28	`PV6_AH_Act`.Value
29	`PV7_AH_Act`.Value

Status bit	Parameter
30	PV8_AH_Act.Value
31	MS_RelOp

Status word allocation for the parameter Status3

You can find a description for each parameter in the section VoteAnL I/Os (Page 620)

Status bit	Parameter
0	PV1_WH_Act.Value
1	PV2_WH_Act.Value
2	PV3_WH_Act.Value
3	PV4_WH_Act.Value
4	PV5_WH_Act.Value
5	PV6_WH_Act.Value
6	PV7_WH_Act.Value
7	PV8_WH_Act.Value
8	PV1_Dev_Act.Value
9	PV2_Dev_Act.Value
10	PV3_Dev_Act.Value
11	PV4_Dev_Act.Value
12	PV5_Dev_Act.Value
13	PV6_Dev_Act.Value
14	PV7_Dev_Act.Value
15	PV8_Dev_Act.Value
16	PV1_WL_Act.Value
17	PV2_WL_Act.Value
18	PV3_WL_Act.Value
19	PV4_WL_Act.Value
20	PV5_WL_Act.Value
21	PV6_WL_Act.Value
22	PV7_WL_Act.Value
23	PV7_WL_Act.Value
24	PV1_AL_Act.Value
25	PV2_AL_Act.Value
26	PV3_AL_Act.Value
27	PV4_AL_Act.Value
28	PV5_AL_Act.Value
29	PV6_AL_Act.Value
30	PV7_AL_Act.Value
31	PV8_AL_Act.Value

Status word allocation for the parameter `Status4`

You can find a description for each parameter in the section VoteAnL I/Os (Page 620)

Status bit	Parameter
0	`PV1` interconnected
1	`PV2` interconnected
2	`PV3` interconnected
3	`PV4` interconnected
4	`PV5` interconnected
5	`PV6` interconnected
6	`PV7` interconnected
7	`PV8` interconnected
8	`PV1` QC Bad
9	`PV2` QC Bad
10	`PV3` QC Bad
11	`PV4` QC Bad
12	`PV5` QC Bad
13	`PV6` QC Bad
14	`PV7` QC Bad
15	`PV8` QC Bad
16	`PV1_Byp_Out`.Value
17	`PV2_Byp_Out`.Value
18	`PV3_Byp_Out`.Value
19	`PV4_Byp_Out`.Value
20	`PV5_Byp_Out`.Value
21	`PV6_Byp_Out`.Value
22	`PV7_Byp_Out`.Value
23	`PV8_Byp_Out`.Value
24	`PV1` Degraded
25	`PV2` Degraded
26	`PV3` Degraded
27	`PV4` Degraded
28	`PV5` Degraded
29	`PV6` Degraded
30	`PV7` Degraded
31	`PV8` Degraded

Note

The degradation of the `Status4.Bit24 - Status4.Bit31` is indicated only if there is an error in the block (for example: bad status of inputs `PV1 - PV8`) with `Feature.Bit13` = 1.

Status word allocation for the parameter Status5

You can find a description for each parameter in the section VoteAnL I/Os (Page 620)

Status bit	Parameter
0	FirstIn PV1
1	FirstIn PV2
2	FirstIn PV3
3	FirstIn PV4
4	FirstIn PV5
5	FirstIn PV6
6	FirstIn PV7
7	FirstIn PV8
8 - 31	Not used

Status word allocation for the parameter Status6

You can find a description for each parameter in the section VoteAnL I/Os (Page 620)

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via EventTsIn1
1	Effective signal 2 of the message block interconnected via EventTsIn1
2	Effective signal 3 of the message block interconnected via EventTsIn1
3	Effective signal 4 of the message block interconnected via EventTsIn1
4	Effective signal 5 of the message block interconnected via EventTsIn1
5	Effective signal 6 of the message block interconnected via EventTsIn1
6	Effective signal 7 of the message block interconnected via EventTsIn1
7	Effective signal 8 of the message block interconnected via EventTsIn1
8 - 16	Effective signal 8 to 16 of the message block connected via Event16TsIn1
16 - 31	Not used

Status word allocation for the parameter Status7

You can find a description for each parameter in the section VoteAnL I/Os (Page 620)

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via EventTsIn2
1	Effective signal 2 of the message block interconnected via EventTsIn2
2	Effective signal 3 of the message block interconnected via EventTsIn2
3	Effective signal 4 of the message block interconnected via EventTsIn2
4	Effective signal 5 of the message block interconnected via EventTsIn2
5	Effective signal 6 of the message block interconnected via EventTsIn2
6	Effective signal 7 of the message block interconnected via EventTsIn2
7	Effective signal 8 of the message block interconnected via EventTsIn2
8 - 16	Effective signal 8 to 16 of the message block connected via Event16TsIn2
16 - 31	Not used

12.10.2 VoteAnL operating modes

The block can be operated using the following modes:

- On
- Out of service

General information operating modes is available in the section *Operating modes of the blocks* in the *APL Function manual*.

12.10.3 VoteAnL functions

Handling non-connected inputs

Inputs which are not interconnected are not evaluated. They are not displayed in the faceplate either.

Alarm delays with two time values

With the APL function "Separate delay times for each alarm" (Feature bit 8), separate input parameters can be set for the alarm delay at the high and low limits.

Feature bit **8 = 0**

This block provides the standard function, Alarm delay for two time values per limit pair.

Feature bit **8 = 1**

This block provides the standard function, Alarm delay for two time values for each individual limit.

Limit and deviation monitoring of the process value

This block provides the standard APL function, Limit monitoring of process values. Deviation monitoring is provided in addition. The mean process value of the connected process values is used as reference value in this case. A deviation occurs when:

- PVx >= median value + PV_Dev_Out or
- PVx <= median value - PV_Dev_Out

Group display `SumMsgAct` for limit monitoring, trip, pre-trip and `ExtMsgx`

The block provides the standard APL function Group display for limit monitoring as well as the functions Group display for deviation monitoring, Trip, Pre-trip and ExtMsgx.

Suppressing messages using the `MsgLock` parameter

This block includes the standard APL function, "Suppress messages using the MsgLock parameter".

Providing PV limit at the output

The following input parameters are also available at the output for making additional interconnections to other blocks:

• PV_AH_Out	:= PV_AH_Lim
• PV_WH_Out	:= PV_WH_Lim
• PV_Dev_Out	:= PV_Dev_Lim
• PV_WL_Out	:= PV_WL_Lim
• PV_AL_Out	:= PV_AL_Lim
• PV_HysOut	:= PV_HysOut
• PV_DevHysOut	:= PV_DevHysOut

Bypass

Note

In the VoteAnL block, exclusion (bypassing) the process value monitoring means that process value monitoring is excluded from the logic of the block, in other words, this signal is ignored in the logic operation.

This function can only be executed in the faceplate with "high-level operating permission".

You can disable input signals that are temporarily not to be used for the calculation in the block by setting the corresponding I/O.

- Regardless of whether or not `BypLix` is connected, the operator can exclude the bypass signal:
  `PVx_Byp` = 1 (by the operator via faceplate)
- `PVx_BypLi.ST` <> 16#FF (input is connected) and a change to the `PVx_BypLi` value is detected:
  `PVx_BypLi.Value` = 1 (of I/O on CFC)

The I/O is shown in the faceplate by the following symbol:



Depending on the configuration of the Feature bit "Separate evaluation of excluded and simulated interlock signals" (`Feature`.Bit 2), the "Bypass" output is formed and the hidden bypass signal is set at the `Out` output.

Feature bit =0:

If one of the input values is excluded (`PV1_Byp`.. `PV8_Byp` or `PV1_BypLi` .. `PV8_BypLi`), the `BypAct` output is set. The hidden bypass signal at the `Out` output is reset.

- `BypAct`.Value =
  `PV1_Byp` OR `PV1_BypLi` (when `PV1_BypLi` is connected and the value has changed)
  OR `PVx_Byp` OR `PVx_BypLi` (when `PVx_BypLi` is connected and the value has changed)
- `Out`.Bit1 = 0

An excluded input value that is switch-relevant after the exclusion sets the status of Out to Simulation.

Feature bit =1:

If one of the input values is excluded (PV1_Byp.. PV8_Byp or PV1_BypLi .. PV8_BypLi) or the hidden bypass signal is set at one of the process value inputs (PV1.. PV8), the BypAct output is set. The bypass signal at the PV1PV8 input can be read from the upstream block at the BypAct output. The hidden bypass signal at the Out output is set to the value of the BypAct .

- BypAct.Value =
  PV1_Byp OR PV1_BypLi (when PV1_BypLi is connected and the value has changed)
  OR PVx_Byp OR PVx_BypLi (when PVx_BypLi is connected and the value has changed)
  OR PV1 = 16#60 OR PVx = 16#60
  OR PV1.Bit1 OR PVx.Bit1 (hidden lines)
- Out.Bit1 = BypAct.Value

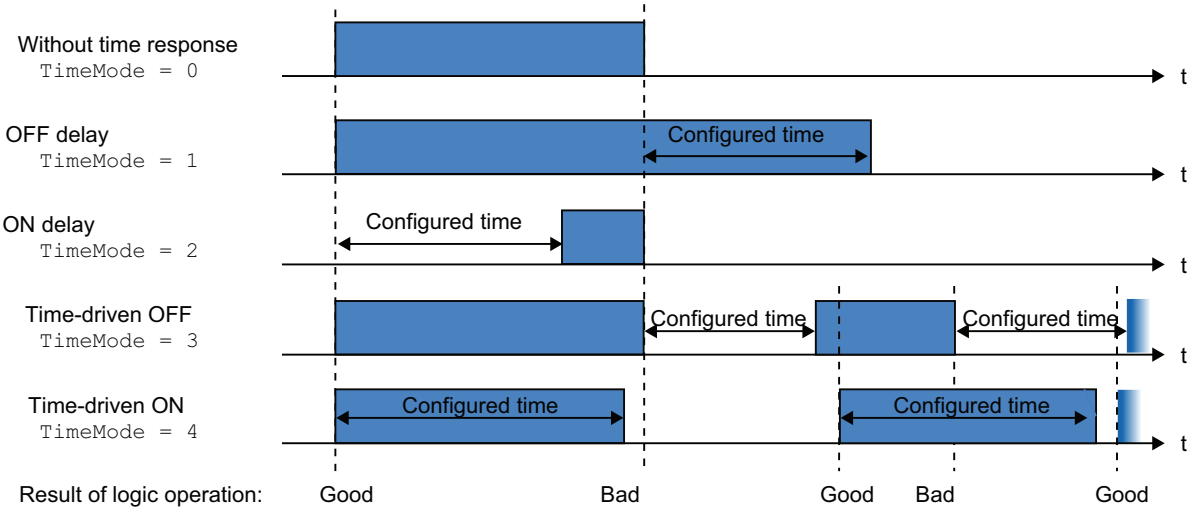
An excluded interlock input has no influence on the status of Out.

Time behavior

A time response for the output value can be configured for the block. The TimeMode I/O can be used to configure the following time responses:

- TimeMode =0: Without time response (default)
- TimeMode =1: OFF delay (changing the result of logic operation from good to bad is delayed by VoteTime)
- TimeMode =2: ON delay (changing the result of logic operation from bad to good is delayed by VoteTime)
- TimeMode =3: Time-driven OFF (the result of logic operation is bad for the time VoteTime)
- TimeMode =4: Time-driven ON (the result of logic operation is good for the time VoteTime)

The time for the response is specified at the VoteTime I/O.



WARNING

Changes to the `TimeMode` connection

If the time response at the `TimeMode` connection is changed during runtime, this may have an effect on the signal state of the block output. The time function has an effect on the output once after the change regardless of the input signals.

Calling additional blocks

This block includes the standard function Call additional faceplates. Information is available in the section *Calling additional faceplates* in the *APL Function Manual*.

First-in signal detection with the block

This block provides the standard APL function, first-in signal detection. Note that a signal status change only has an effect on the initial signal detection when the `Feature` Bit 31 (Enable first-in signal detection) and the `FirstInEn` input are set.

The response to deactivation via the `FirstInEn` input can be influenced by `Feature` Bit 21 (first-in signal acquisition response to deactivation).

Note

This function can only be executed in the faceplate with "process control" operating permission.

Forming PV_Out for blocks

The block forms the `PV_Out` using the `PV_Mode` parameter. The following parameter settings are possible:

- `PV_Mode` := 1:
  `PV_Out`.Value := median `PV_Median`.Value / `PV_Out`.ST := median `PV_Median`.ST
- `PV_Mode` := 2:
  `PV_Out`.Value := median `PV_Mean`.Value / `PV_Out`.ST := median `PV_Mean`.ST
- `PV_Mode` := 3
  `PV_Out`.Value := median `PV_Max`.Value / `PV_Out`.ST := median `PV_Max`.ST
- `PV_Mode` := 4:
  `PV_Out`.Value := median `PV_Min`.Value / `PV_Out`.ST := median `PV_Min`.ST

`PV_Median` is the median process value, `PV_Mean` is the mean process value, `PV_Max` is the highest process value and `PV_Min` is the lowest process value of all connected process values, whereby only process values without the signal status 16#00 and 16#28 are used for determining the output variables and the signal status.

Forming the signal status for blocks

This block provides the standard APL function, Form and output the signal status for technological blocks.

The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `PV_Out`.ST (if trip is enabled)
- `Out`.ST (if trip is enabled)

Release for maintenance

This block has the standard APL function, "Release for maintenance".

Simulating signals

This block features the standard APL function, "Simulating signals".

You can simulate the following values:

- Process value (`SimPV`, `SimPV_Li`)

Separate unit and scaling range for each process value

Using the "Separate unit device and scaling range for each process value" feature (`Feature`.Bit 9), you can assign a separate unit of measure and a separate scaling range to each process value. The default setting is 0.

Feature Bit =0:

The unit and the scaling range are specified for all process values with `PV_Unit` and `PV_OpScale`.

Feature Bit =1:

Each process value has its own unit and scaling range. These are specified at `PVX_Unit` and `PVX_OpScale`.

Limit voting

Depending on the configuration of the feature "Limit voting (`PVx` bad at alarm)" (`Feature`.Bit 10), a signal is evaluated as bad due to a violated limit:

Feature bit =0:

The process value is evaluated independent of the configured limits and further processed.

Feature bit =1:

The process value is evaluated as bad with a limit violation and further processed.

Deviation voting

Depending on the configuration of the feature "Deviation voting (`PVx` bad at deviation)" (`Feature`.Bit 11), a signal is evaluated as bad due to exceeding a deviation from other signals:

Feature bit =0:

The process value is evaluated independent of the deviation and further processed.
When a configure deviation is reached, a "Warning - high" type message is generated (MsgEvId1 SIG4).

Feature bit =1:

The process value is evaluated as bad upon exceeding a configured deviation and further processed.
When a configure deviation is reached, an "Alarm - high" type message is generated (MsgEvId1 SIG3).

Signal status voting

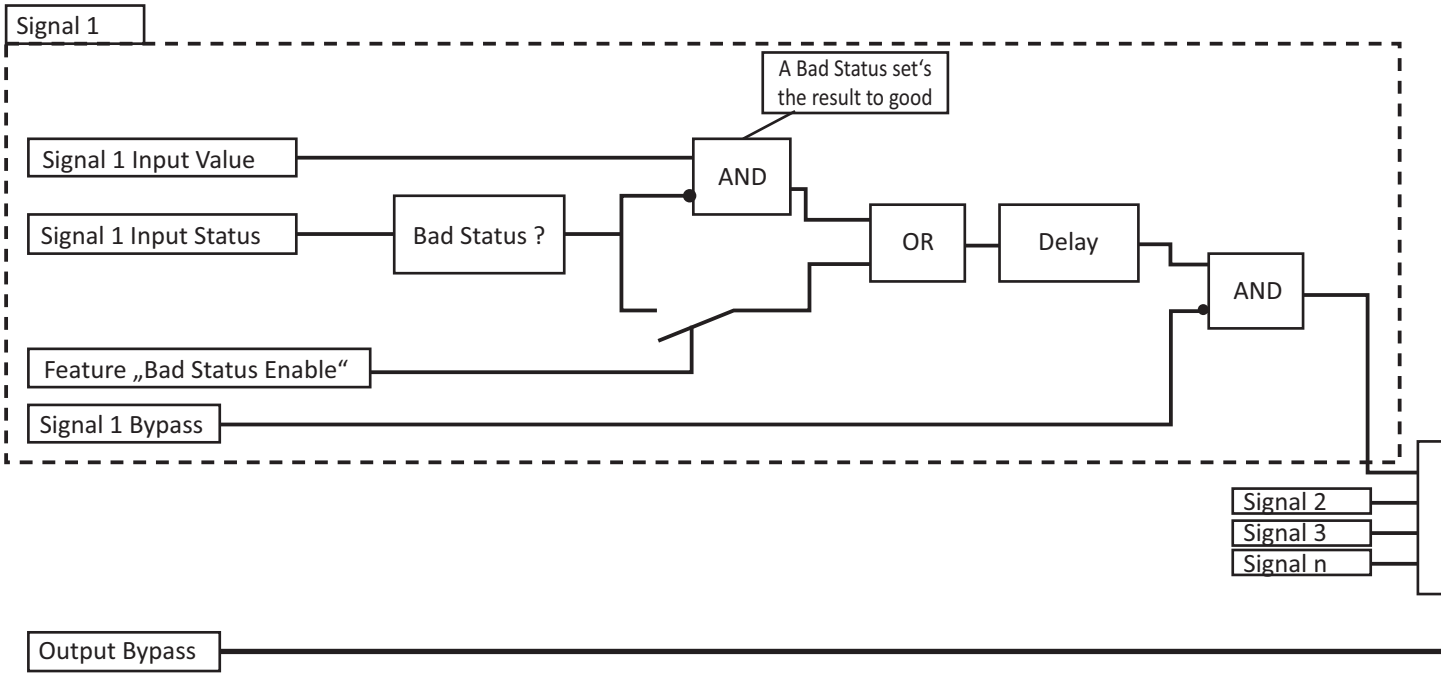
Depending on the configuration of the feature "Signal status voting (`PVx` bad at bad signal status)" (`Feature`.Bit 12), a signal is considered bad due to a bad signal status:

Feature bit =0:

The process value is evaluated independent of the signal status and further processed.

Feature bit =1:

The process value is evaluated immediately as bad with bad signal status and further processed.



Degraded Voting (degraded voting)

The input signals of a block can be "excluded from the voting" based on a bad signal status or based on a bypass, or can be evaluated as "bad/good". The behavior depends on the setting of the "Degraded Voting" feature (`Feature`.Bit 13)

Feature Bit =0:

Degraded voting is not active.

Input signals of the block with bad signal status are evaluated as "bad" (`Feature`.Bit 12 =1) or continue to be processed normally (`Feature`.Bit 12 =0). Bypassed input signals are evaluated as "good".

Feature Bit =1:

Degraded Voting is active.

Input signals of the block with poor signal status (depending on `Feature`.Bit 12) and bypassed input signals are excluded from voting. With `Feature`.Bit 12 =1, the input signals with bad signal status are "discarded". With `Feature`.Bit 12 =0, the bad signal status is ignored.

An input is ultimately "Good", "Bad", "Discarded" or "Bypassed". This result enters into the logic function. Discarded or bypassed signals result in a change (reduction) of the logic function. For example, a 2oo3 (2 out of 3) logic reduces to a 2oo2 logic for a bypassed or discarded signal. Another discarded/bypassed signal results in 1oo1 voting. Discarded or bypassed signals have the same effect here – except when all the input signals have been bypassed.

Function 2003			*Result*	
Input signal 1	Input signal 2	Input signal 3	Reduced Function	Logic Result
Good	Good	Good	2oo3	Good
Good	Good	Bad	2oo3	Good
Good	Bad	Bad	2oo3	Bad
Bad	Bad	Bad	2oo3	Bad
Good	Good	Bypassed	2oo2	Good
Good	Bypassed	Bypassed	1oo1	Good
Bypassed	Bypassed	Bypassed	0oo0	Good
Good	Good	Discarded	2oo2	Good
Good	Discarded	Discarded	1oo1	Good
Discarded	Discarded	Discarded	0oo0	Bad
Discarded	Discarded	Bypassed	0oo0	Bad
Discarded	Bypassed	Bypassed	0oo0	Bad

Figure 12-1 Result of the logic function using the example of a 2oo3 function based on differently evaluated input signals

Signal value for tripping

Depending on the parameter assignment of the "Signal value for tripping" feature (`Feature`.Bit 14), the output is 1 or 0 if the logic is fulfilled.

Feature Bit =0:

Deenergize-to-trip (DTT) ⇒ Logic fulfilled: `Out`.Value = 0

Feature Bit =1:

Energize-to-trip (ETT) ⇒ Logic fulfilled: `Out`.Value = 1

Automatic reset

Depending on the parameter assignment of the "Automatic reset" feature (`Feature`.Bit 15), the output is automatically reset after tripping or must be reset by the operator.

Feature Bit =0:

Automatic reset not active The output value becomes good only when the logic is no longer fulfilled and the operator has reset the output.

Feature Bit =1:

Automatic reset active. The output value becomes good as soon as the logic is no longer fulfilled.

Early trigger warning

Depending on the configuration of the Feature "Trigger prewarning active" (`Feature`.Bit 16), a message is generated if only a bad value for triggering the block is missing.

Feature Bit =0:

A message is triggered as a warning.

Feature Bit =1:

No message is triggered.

A bypass for all `PVx`

Depending on the parameter assignment of the feature "A bypass for all `PVx`" (`Feature`.Bit 17) it is possible to only allow the bridging of an input signal.

Feature Bit =0:

There is no limit to setting the bypass.

Feature Bit =1:

A bypass can only be set for one input signal.

Low / high median

Use the `Feature`.Bit "Low/high median" (`Feature`.Bit 18) to specify which value is selected as the mean value when there is an even number of values. Only plausible values are included in the calculation of the mean value.

Feature Bit = 0:

The lower median is selected as the mean value.

Feature Bit = 1:

The higher median is selected as the mean value.

The table below shows the effects of `Feature` bit 18 with an even number of values:

Feature.Bit 18	PV1	PV2	PV3	PV4	Mean value
0	20.0	**21.0**	22.0	23.0	21.0
1	20.0	21.0	**22.0**	23.0	22.0

The table below shows the effects of `Feature` bit 18 with an odd number of values:

Feature.Bit 18	PV1	PV2	PV3	PV4	PV5	Mean value
0	20.0	21.0	**22.0**	23.0	26.0	22.0
1	20.0	21.0	**22.0**	23.0	22.0	22.0

Select unit of measure

This block includes the standard APL function, Select unit of measure.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4 - 5	Not used
6	1 = Operator can change the simulation value `SimPV`
7 - 10	Not used
11	1 = Operator can activate the Simulation function
12	1 = Operator can activate the Release for maintenance function
13	1 = Operator can change the limit (`PV_AH_Lim`) for the high alarm
14	1 = Operator can change the limit (`PV_WH_Lim`) for the high warning
15	1 = Operator can change the limit (`PV_DevHyst`) for the hysteresis of the deviation

Bit	Function
16	1 = Operator can change the limit (PV_Hyst) for the hysteresis
17	1 = Operator can change the limit (PV_AL_Lim) for the low alarm
18	1 = Operator can change the limit (PV_WL_Lim) for the low warning
19	1 = Operator can change the limit (PV_Dev_Lim) for the deviation
20 - 23	Not used
24	1 = The operator can enable/disable messages via PV_AH_MsgEn
25	1 = The operator can enable/disable messages via PV_WH_MsgEn
26	1 = The operator can enable/disable messages via PV_Dev_MsgEn
27	1 = The operator can enable/disable messages via PV_WL_MsgEn
28	1 = The operator can enable/disable messages via PV_AL_MsgEn
29 - 31	Not used

The block has the following permissions for the OS1Perm parameter:

Bit	Function
0	1 = Operator can exclude values (bypass)
1	1 = Operator can reset the exclusion of input values
2	1 = Operator can reset the recording of the first signal
3	1 = Operator can switch PV_Mode = 1 (PV_Out = PV_Median)
4	1 = Operator can switch PV_Mode = 2 (PV_Out = PV_Mean)
5	1 = Operator can switch PV_Mode = 3 (PV_Out = PV_Max)
6	1 = Operator can switch PV_Mode = 4 (PV_Out = PV_Min)
7 - 8	Not used
9	1 = Operator can reset the output value
10 - 12	Not used
13	1 = Operators can change VoteTime
14 - 17	Not used
18	1 = Operator can set or reset the exclusion of process value PV1
19	1 = Operator can set or reset the exclusion of process value PV2
20	1 = Operator can set or reset the exclusion of process value PV3
21	1 = Operator can set or reset the exclusion of process value PV4
22	1 = Operator can set or reset the exclusion of process value PV5
23	1 = Operator can set or reset the exclusion of process value PV6
24	1 = Operator can set or reset the exclusion of process value PV7
25	1 = Operator can set or reset the exclusion of process value PV8
26	1 = Operator can set or reset the exclusion of output value Out
27 - 31	Not allocated

Note

If you interconnect a parameter that is also listed in OS_Perm / OS1Perm as a parameter, you have to reset the corresponding OS_Perm / OS1Perm bit.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
2	See APL function: "Separate evaluation of excluded and simulated interlock signals"
5	Activate OS1Perm Bits 0 = OS1Perm Bits 18...25 disabled (evaluation only in faceplate) 1 = OS1Perm Bits 18...25 enabled (evaluation only in faceplate)
6	Activate OS1Perm Bits 0 = OS1Perm Bit 26 disabled (evaluation only in faceplate) 1 = OS1Perm Bit 26 enabled (evaluation only in faceplate)
8	See APL function: "Separate delay times for each alarm"
9	Separate unit and scaling range for each process value
10	Limit voting (`PVx` bad for alarm)
11	Deviation voting (`PVx` bad at deviation)
12	Signal status voting (`PVx` bad for bad signal status)
13	Degraded voting
14	Signal value for tripping
15	Automatic reset
16	Early trigger warning active
17	A bypass for all `PVx`
18	Low / high median
19	See APL function: "Changes Signal status of outputs in OOS"
21	See APL function: "First-in detection response to deactivation"
22	See APL function: "Update acknowledgment and error status of the message call"
23	See APL function: "Limits disabled in faceplate"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"
26	See APL function: "Reaction of the switching points in the "Out of service" operating mode"
28	See APL function: "Deactivate switching points"
29	See APL function: "Signal limit violation"
31	See APL function: "Enable first-in detection"

Configurable reactions using the Feature2 parameter

An overview of all the reactions that are provided by the Feature parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0 - 31	Not used

Generating instance-specific messages

This block includes the standard APL function, "Generating instance-specific messages".

Specifying the display area for process and setpoint values as well as operations

This block features the standard APL function "Display and operator input area for process values and setpoints".

Time stamp

This block receives a time stamp value via the EventTSIn input parameter. For more information on this, refer to the section "Functions of EventTs" in the APL documentation.

SIMATIC BATCH functionality

This block provides the standard APL function, "SIMATIC BATCH functionality".

12.10.4 VoteAnL error handling

VoteAnL error handling

For information about error handling for all blocks, see also "Error handling" in the Basics of APL section of the APL documentation.

The following errors can be displayed for this block:

- Error numbers

Overview of error numbers

The ErrorNum I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
30	The value PVx can no longer be represented in the REAL numerical field.
90	Parameter assignment error

12.10.5 VoteAnL messaging

Message response

The following messages can be generated for this block:

- Process messages
- Instance-specific messages

Process messages

Message instance	Message identifier	Message class	Event
MsgEvId1	SIG 1	Alarm - high	$$BlockComment$$ PVx - High alarm limit violated
	SIG 2	Warning - high	$$BlockComment$$ PVx - High warning limit violated
	SIG 3	Alarm – high	$$BlockComment$$ PVx - Deviation limit violated
	SIG 4	Warning – high	$$BlockComment$$ PVx - Deviation limit violated
	SIG 5	Warning – low	$$BlockComment$$ PVx - Low warning limit violated
	SIG 6	Alarm – low	$$BlockComment$$ PVx - Low alarm limit violated
	SIG 7	Alarm - high	$$BlockComment$$ Trip is enabled
	SIG 8	Warning - high	$$BlockComment$$ Trip pending

Explanation:

$$BlockComment$$: Content of the instance-specific comment

Instance-specific messages

You can use up to three instance-specific messages with this block.

Message instance	Message identifier	Message class	Event
MsgEvId2	SIG 1	AS control system message - fault	$$BlockComment$$ External message 1
	SIG 2	AS control system message - fault	$$BlockComment$$ External message 2
	SIG 3	AS control system message - fault	$$BlockComment$$ External message 3

Explanation:

$$BlockComment$$: Content of the instance-specific comment

Associated values for message instance `MsgEvId1`

Associated value	Block parameters
1	BatchName
2	StepNo
3	BatchID
4	ExtVa104
5	ExtVa105
6	ExtVa106
7	ExtVa107
8	ExtVa108
9	Reserved
10	Reserved

The associated values 4 ... 8 are allocated to the `ExtVa104` ... `ExtVa108` parameters and are available for use. You can find additional information in the "Process Control System PCS 7 - Engineering System" manual.

Associated values for message instance `MsgEvId2`

Associated value	Block parameters
1	BatchName
2	StepNo
3	BatchID
4	ExtVa204
5	ExtVa205
6	ExtVa206
7	ExtVa207
8	ExtVa208
9	Reserved
10	Reserved

The associated values 4 ... 8 are allocated to the `ExtVa204` ... `ExtVa208` parameters and are available for use. You can find additional information in the "Process Control System PCS 7 - Engineering System" manual.

12.10.6 VoteAnL I/Os

Input parameters

Parameter	Description	Type	Default
EventTsIn1	For interconnecting data between a technology block and the message blocks EventTs, Event16Ts. The input parameter EventTsIn1 is used to interconnect with the output parameter EventTsOut of the EventTs, Event16Ts block. When this interconnection is configured, the messages of the EventTs, Event16TS block are displayed and can also be acknowledged on the OS in the message view of the technological block.	ANY	-
EventTsIn2	For interconnecting data between a technology block and the message blocks EventTs, Event16Ts. The input parameter EventTsIn1 is used to interconnect with the output parameter EventTsOut of the EventTs, Event16Ts block. When this interconnection is configured, the messages of the EventTs, Event16TS block are displayed and can also be acknowledged on the OS in the message view of the technological block.	ANY	-
Feature	I/O for additional functions (Page 606)	STRUCT • Bit 0: BOOL • ... • Bit 31: BOOL	- • 0 • 0 • 0
FirstInEn	First-in detection	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OS_Perm	I/O for operator permissions (Page 606)	STRUCT • Bit 0: BOOL • ... • Bit 31: BOOL	- • 1 • 1 • 1
OS1Perm	I/O for operator permissions (Page 606)	STRUCT • Bit 0: BOOL • ... • Bit 31: BOOL	- • 1 • 1 • 1
Out_Byp	Bypass by operator	BOOL	0
PV_A_DC	Delay time for incoming alarms for overshoot/ undershoot or only for undershot process values [s]	REAL	0.0
PV_A_DG	Delay time for outgoing alarms for overshoot/ undershoot or only for undershot process values [s]	REAL	0.0

Parameter	Description	Type	Default
PV_AH_DC	Delay time for incoming alarms for overshot process values [s]	REAL	0.0
PV_AH_DG	Delay time for outgoing alarms for overshot process values [s]	REAL	0.0
PV_AH_En	1 = Enable alarm limit (high)	BOOL	1
PV_AL_En	1 = Enable alarm limit (low)	BOOL	1
PV_Dev_DC	Delay time for incoming deviation message for overshoot/undershoot or only for overshot process values [s]	REAL	0.0
PV_Dev_DG	Delay time for outgoing deviation message for overshoot/undershoot or only for undershot process values [s]	REAL	0.0
PV_Dev_En*	1 = Activate deviation limits	BOOL	1
PV_Dev_Lim*	Limit deviation	REAL	10.0
PV_Dev_MsgEn	1 = Activate message for deviation	BOOL	1
PV_DevHyst*	Hysteresis for deviation	REAL	0.0
PV_Mode*	Output value at PV_Out: 1 = PV_Median 2 = PV_Mean 3 = PV_Max 4 = PV_Min	INT	1
PV_W_DC	Delay time for incoming warnings for overshoot/ undershoot or only for undershot process values [s]	REAL	0.0
PV_W_DG	Delay time for outgoing warnings for overshoot/ undershoot or only for undershot process values [s]	REAL	0.0
PV_WH_DC	Delay time for incoming warnings for overshot process values [s]	REAL	0.0
PV_WH_DG	Delay time for outgoing warnings for overshot process values [s]	REAL	0.0
PV_WH_En	1 = Activate warning limit (high)	BOOL	1
PV_WL_En	1 = Activate warning limit (low)	BOOL	1
PV1 .. PV8*	Process value 1 .. Process value 8	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
PV1_Byp .. PV8_Byp*	1 =Bypass PV1 .. PV8 via operator	BOOL	0
PV1_OpScale .. PV8_OpScale*	Limit for scale in PV1 .. PV8 bar graph of faceplate	STRUCT • High: REAL • Low: REAL	- • 100.0 • 0.0
PV1_Unit .. PV8_Unit *	Unit of measure for process value 1.. Process value 8	INT	1001
RstBypOp*	1 = Reset PV1_Byp. PV8_Byp / Out_Byp by operator	BOOL	0
RstFirstOp*	1 = Reset FirstEn by operator	BOOL	0

Parameter	Description	Type	Default
TimeMode*	Dynamic response: 0: No time response 1: On delay 2: Off delay 3: Time-controlled OFF 4: Time-driven ON	INT	0
VoteMode*	Number of "Bad" or "Error" values that lead to an error at the output. E.g. if voting mode 1oo2 is configured, for two inputs if any one of the input has an error or bad condition the output will also result in an error.	INT	1
VoteTime*	Time of ON/OFF delay or evaluation [s]	REAL	0.0

* Values can be written back to these inputs during execution of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
BypAct	1 = Bypass active for PV1..PV8	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
FirstIn	Bit-coded number of the first signal that has resulted in a change at the output	DWORD	16#00000000
OS1PermLog	Display of OS1Perm with settings changed by the block algorithm	DWORD	16#FFFFFFFF
OS1PermOut	Display of OS1Perm	DWORD	16#FFFFFFFF
Out	Output	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Out_Byp_Out	1 = Bypass active for Out	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV	Process value output with simulation	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
PV_AH_Out	PV - Alarm limit (high) output	REAL	0.0
PV_AL_Out	PV - Alarm limit (low) output	REAL	0.0
PV_Dev_Out	PVx - deviation output	REAL	0.0
PV_DevHysOut	Hysteresis for PV deviation, output	REAL	0.0
PV_HysOut	Hysteresis for PV alarm, output	REAL	0.0
PV_Max	Highest value over all PV inputs	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF

Parameter	Description	Data type	Default
PV_Mean	Mean value over all PV inputs	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
PV_Median	Mean value over all PV inputs	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
PV_Min	Lowest value over all PV inputs	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
PV_OpScaleOut	Limit for scale in PV bar graph of faceplate	STRUCT • High: REAL • Low: REAL	- • 100.0 • 0.0
PV_Out	Process value output	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
PV_UnitOut	Unit of measure for process value	INT	1190
PV_WH_Out	PVx - warning limit (low) output	REAL	0.0
PV_WL_Out	PVx - warning limit (low) output	REAL	0.0
PV1_AH_Act .. PV8_AH_Act	1 = PV1 .. PV8 alarm (high) active. You can change the response for this parameter with Feature Bit 28 (deactivate switching points) and with Feature Bit 29 (signal limit violation).	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV1_AL_Act .. PV8_AL_Act	1 = PV1 .. PV8 alarm (low) active. You can change the response for this parameter with Feature Bit 28 (deactivate switching points) and with Feature Bit 29 (signal limit violation).	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV1_Bad .. PV8_Bad	1 = PV1 .. PV8 bad	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV1_Byp_Out .. PV8_Byp_Out	1 = Bypass active PV1 .. PV8	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV1_Dev_Act .. PV8_Dev_Act	1 = PV1 .. PV8 deviation active. You can change the response for this parameter with Feature Bit 28 (deactivate switching points) and with Feature Bit 29 (signal limit violation).	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
PV1_Out .. PV8_Out	Process value output	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
PV1_WH_Act .. PV8_WH_Act	1 = PV1 .. PV8 warning (low) active. You can change the response for this parameter with Feature Bit 28 (deactivate switching points) and with Feature Bit 29 (signal limit violation).	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV1_WL_Act .. PV8_WL_Act	1 = PV1 .. PV8 warning (low) active. You can change the response for this parameter with Feature Bit 28 (deactivate switching points) and with Feature Bit 29 (signal limit violation).	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Status1	Status word 1	DWORD	16#00000000
Status2	Status word 2	DWORD	16#00000000
Status3	Status word 3	DWORD	16#00000000
Status4	Status word 4	DWORD	16#00000000
Status5	Status word 5	DWORD	16#00000000
Status6	Status word 6	DWORD	16#00000000
Status7	Status word 7	DWORD	16#00000000
VoteTime_Out	Remaining delay time [s]	REAL	0.0

12.10.7 VoteAnL block diagram

VoteAnL block diagram

A block diagram is not provided for this block.

12.10.8 Operator control and monitoring

12.10.8.1 VoteAnL views

VoteAnL block views

The VoteAnL block provides the following views:

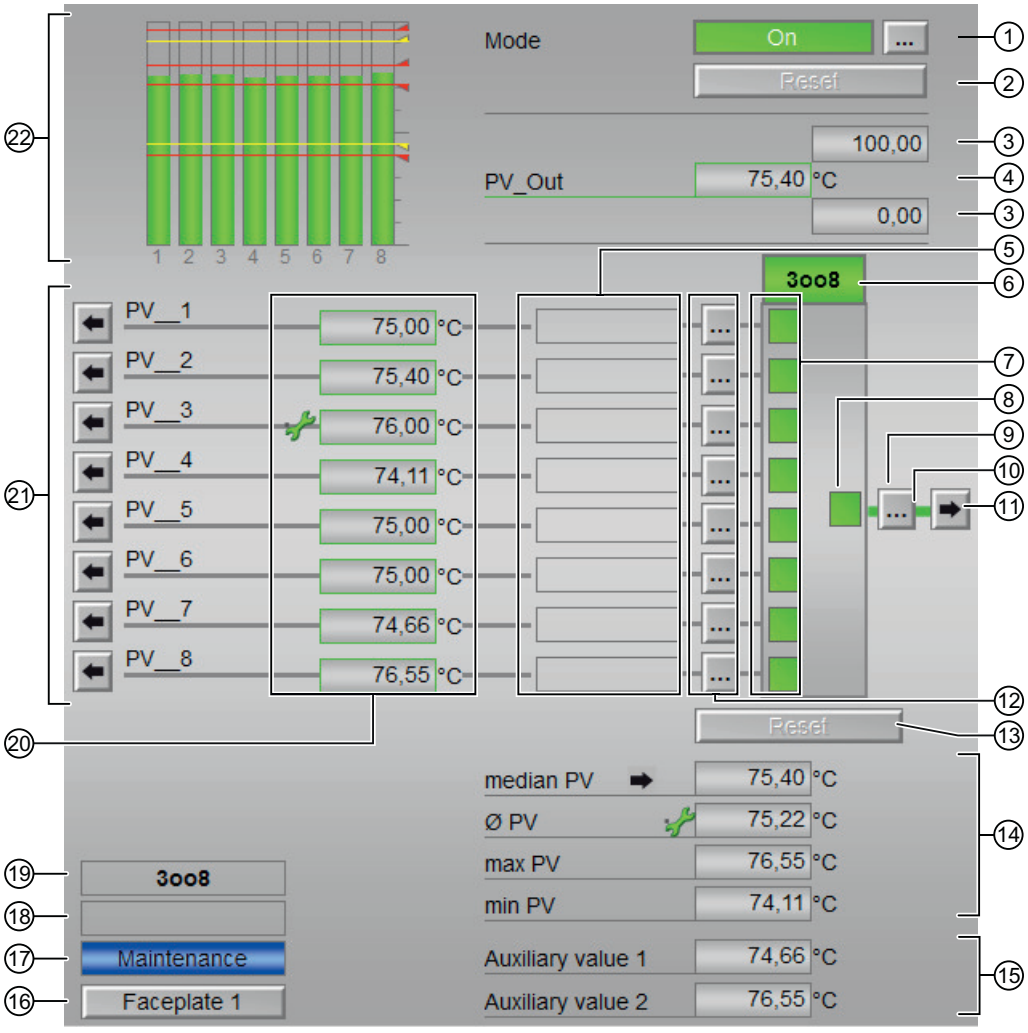
- VoteAnL standard view (Page 625)
- Message view

- Limit view of VoteAnL (Page 632)
- Trend view
- VoteAnL parameter view (Page 633)
- VoteAnL preview (Page 635)
- Memo view
- Batch view
- VoteAnL block icon (Page 636)

You can find general information on the faceplate and block icon in the "Structure of the faceplate" and "Block icon structure" sections of the APL function manual.

12.10.8.2 VoteAnL standard view

VoteAnL standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

For more information on switching the operating mode, refer to the section "Switching operating states and operating modes" in the APL documentation.

(2) Reset the output value

You can reset the output value with the button. The button is only operable if a reset is required.

The display of this button depends on the configuration in the engineering system (ES).

(3) High and low scale range for the process value

These values provide information on the display range for the bar graph of the process value. The scale range is defined in the engineering system.

(4) Display of the process value including signal status

This area shows the current process value with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. You can find additional information on this in the "Labeling buttons and texts" section in the APL documentation.

You can change the text for the process value at the `PV_Out` parameter:

Changing the display of the text for "Process value"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the parameter (`PV_Out`) to what you want to later see in runtime.
- In the "Identifier" column, change the default setting for the input parameter (`PV_Out`) to what you later want to see in runtime.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding `PV_Out` input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (`PV_Out`), this text is displayed even then the "Identifier" text field of the input parameter (`PV_Out`) is not empty. If the "OS additional text" field of the (`PV_Out`) input parameter is empty, the "Identifier" text of the (`PV_Out`) input parameter is displayed.

(5) Display of the process value status

These fields show the status information associated with the process values (21):

Color	Status
	The process value PVx is error-free and will be further processed as good.
warning	The process value PVx has violated a high or low warning limit and will be further processed as good.
alarm	The process value PVx has violated a high or low alarm limit and will be further processed as bad.
Deviation	The process value PVx has violated a high or low deviation limit and will be further processed as bad.
Bad value	The process value PVx has a bad signal status (16#00 or 16#28) and will be further processed as bad.

(6) Status of the output signal of the block

This area (6) shows the status of the output signal of the block (priority from high to low). You can configure the logic in the engineering system (ES).

Color	Logic (voting mode 1oo1, 1oo2, 1oo3, 2oo3 etc.)
Gray	Block is not used
Blue	Excluded (bypass)
Red	Good state (Out active)
Yellow	Pre-trip warning
Yellow	Simulated
Green	Good state (Out OK)

(7) Display of the status for further processing

The symbol shows the status for further processing of the input values:

Color	Output status
	The process value PVx will be further processed as good.
	The process value PVx will be further processed as bad.
B	The process value PVx will be excluded from further processing.

(7) "First-in" status display

The following symbol is displayed next to an input value, if this input value has caused the last output signal change from 1 to 0 (good state to locked):



You can reset the first-in signal with the button (12).

Note

This function can only be executed in the faceplate with "high-level operating permission".

For more information, refer to the section Recording the first signal for interlock blocks in the APL documentation.

For more information, refer to the section Switching operating states and operating modes in the APL documentation.

(8) Display of the output signal

The icon displays the status of the output signal after the execution of the logic:

Color	Output status
	The output signal is further processed as Good.
	The output signal is further processed as Bad.
	The output signal is excluded from further processing.

(9) Exclusion of output signals

You can use the (9) button to exclude the output signal from the logic and place it in the good state. Depending on the previous settings, you can "Set" or "Reset" this property.

If the input value has been excluded, the following symbol appears in the field (8):



For more information, refer to the section Switching operating states and operating modes in the APL documentation.

Note

This function can only be executed in the faceplate with "high-level operating permission".

Note

Operator authorizations via `OS1Perm` depend on the setting of `Feature` bit 5.

You can find additional information in the section Enabling `OS1Perm` bits in the APL documentation.

(10) Status of the block output

The line color indicates the status of the block output:

Color	Output status
Gray	Block is not used (out of service)
Blue	Excluded (bypass)
Yellow	Pre-trip warning or simulated
Red	Output is enabled
Green	Output is disabled

(11) Open the faceplate of the output value

When you press the highlighted button, you can open the faceplate associated with the output value. The function of this button depends on the configuration in the engineering system (ES). For more information, refer also to the section "Opening additional faceplates" in the APL documentation.

(12) Exclusion of input signals

You can use the button (12) to exclude input values from processing. Depending on the previous settings, you can "Set" or "Reset" this property.

If the input value has been excluded, the following symbol appears in the field (6):



For more information, refer to the section Switching operating states and operating modes in the APL documentation.

Note

This function can only be executed in the faceplate with "high-level operating permission".

Note

Operator authorizations via `OS1Perm` depend on the setting of `Feature` bit 5.

You can find additional information in the section Enabling `OS1Perm` bits in the APL documentation.

(13) "Reset" the settings for further processing

When you press the button (13), you can "Reset" all input values:

- "Reset exclusions": the exclusions of the input values are reset.
- "Reset FirstIn": First-in detection / status display (7) is reset.

For more information, refer to the section Recording the first signal for interlock blocks in the APL documentation.

(14) Display of the calculated output values with signal status

This area shows the calculated output values with signal status. The following values are displayed:

- `PV_Median` (mean process value)
- `PV_Mean` (mean process value)
- `PV_Max` (max. process value)
- `PV_Min` (min. process value)

One of these output values is set at the process value output `PV_Mode` via the `PV_Out` parameter. This value is marked by the following symbol:



Changing the display of the text:

You can change the displays in the CFC in the object properties of the VoteAnL block:

- Navigate to the I/Os (object properties).
- In the "Identifier" column, change the default setting for the input parameter (`PV_Median`, `PV_Mean`, `PV_Max`, `PV_Min`) to what you want to see in the faceplate.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding PVx input.

(15) Display for auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system. You can find more information on this in the section Displaying auxiliary values in the APL documentation.

(16) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

(17) Display area for block states

This area provides additional information on the operating state of the block:

- "Trip state"

(18) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

(19) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

(20) Display of process values with signal status

This area shows the interconnected process values (PVx). Set a unit of measure (PVx_Unit) for each process value in the engineering system (ES).

You can change the text for the process value at the PV_Out parameter:

Changing the display of the text for "Process value"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the parameter (PVX_Out) to what you want to later see in runtime.
- In the "Identifier" column, change the default setting for the input parameter (PVX_Out) to what you later want to see in runtime.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding PVX_Out input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (PVX_Out), this text is displayed even then the "Identifier" text field of the input parameter (PVX_Out) is not empty. If the "OS additional text" field of the (PVX_Out) input parameter is empty, the "Identifier" text of the (PVX_Out) input parameter is displayed.

(21) Call faceplate of the process values

When you press the button (21), you can open the associated faceplate for each process value. The function of this button depends on the configuration in the engineering system (ES). For more information, refer also to the section "Opening additional faceplates" in the APL documentation.

(22) Bar graph for "Process values"

This area shows the current "Process value" in the form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

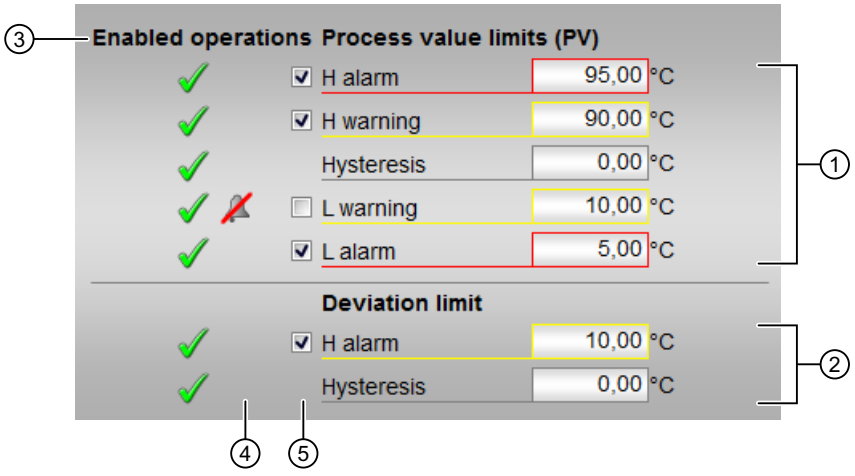
12.10.8.3 Limit view of VoteAnL

Limit view of VoteAnL

Several values are set in this view by default:

- Process value limits
- Deviation limit

The toolbar of the faceplate and the block icon indicate when the limits are reached or violated.



(1) Process value limits

In this area, you can enter the limits for the process value. For more information, refer to the section Changing values in the APL documentation.

You can change the following limits:

- "H alarm": Alarm high
- "H warning": Warning high
- "Hysteresis"
- "L warning": Warning low
- "L alarm": Alarm low

(2) Deviation limit

You can enter the deviation limit in this area. For more information, refer to the section Changing values in the APL documentation.

You can change the following limits:

- "H alarm" or "H warning": Deviation limit (depending on the configuration of `Feature` bit11 "Deviation voting (PVx bad at deviation)"
- "Hysteresis": Hysteresis of the deviation limit

(3) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: The OS operator cannot control this parameter at this time due to the process
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

(4) Message suppression/delay

Message suppression indicates whether the suppression of the associated message in the AS block is activated with the `xx_MsgEn` parameter. The output of messages is not suppressed when the block is installed (all `xx_MsgEn` parameters are preset to 1). Messages can only be output if limit monitoring of the additional analog value has been enabled.

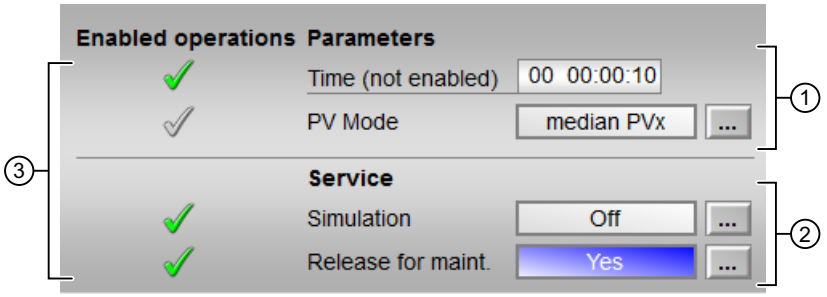
Alarm delays are also displayed in this position; for more on this, see section Area of application of the alarm delays in the APL documentation.

(5) Suppress messages

You can enable/disable messages by setting the check mark.

12.10.8.4 VoteAnL parameter view

VoteAnL parameter view



(1) Parameters

You can change the following parameters in this area:

- "Time response":
  The time for the time response configured at the block can be set here.
  The display depends on the configuration of the time response.
  You can influence the following parameters:
  - Time (not enabled): A delay is not configured for the output; parameter cannot be manipulated.
  - On-delay: On-delay is configured for the output. This can be controlled with corresponding operator authorization.
  - Off-delay: Off-delay is configured for the output. This can be controlled with corresponding operator authorization.
  - Duration: The output is active time controlled. The minimum time for which the output is active can be controlled with the appropriate operator permission.
- "PV Mode"

You can find more information on this in the section Changing values in the APL documentation.

(2) Service

You can select the following functions in this area:

- "Simulation"
- "Release for maintenance"

For more information, refer to the section Switching operating modes in the APL documentation.

You can find information on this area in the section:

- Simulating signals in the APL documentation
- Release for maintenance in the APL documentation

(3) Enabled operations

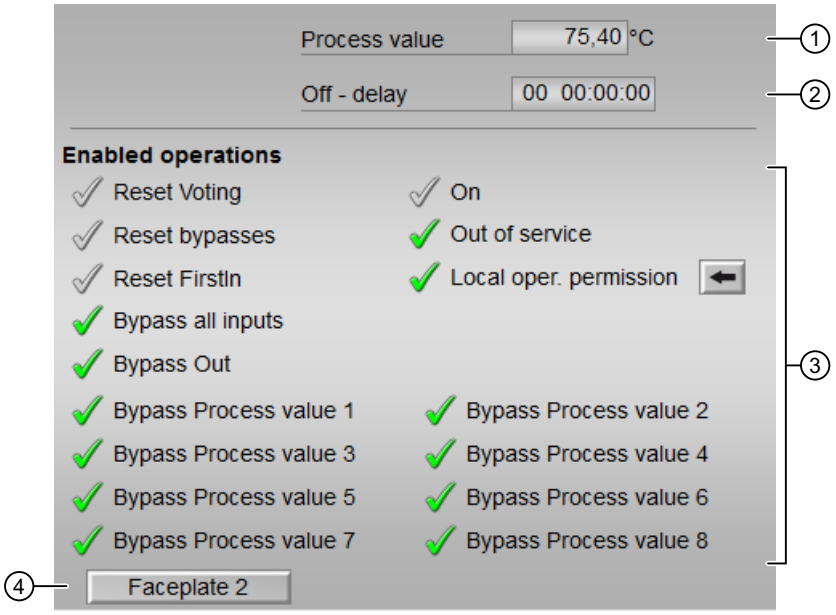
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

12.10.8.5 VoteAnL preview

VoteAnL preview



(1) Process value

This area shows the real process value (PV).

(2) Time response

The current time for the time response configured for the block is displayed here.

The display depends on the configuration.

You can display the following times here depending on the configuration:

- Time (not enabled): No delay is configured for the output.
- On-delay: On-delay is configured for the output.
- Off-delay: Off-delay is configured for the output.
- Duration: The output is active time controlled.

(3) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- Reset voting: You can reset the trip.
- Reset exclusion: You can reset all the excluded process values.
- Reset first in: You can reset first-in signal detection
- Exclude all inputs: You can set/reset the exclusion for all process values
- Exclude output: You can set/reset the exclusion for the output
- Exclude process value 1 .. 8: You can set/reset the exclusion for the respective process value
- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to change to the standard view of the OpStations block. You can find more information on this in the section "Operator control permissions" in the APL documentation.

(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

12.10.8.6 VoteAnL block icon

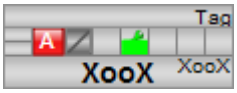
VoteAnL block icons

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of alarm, warning, and deviation limits as well as trip and pre-trip errors
- Operating modes
- Signal status, release for maintenance
- Memo display

- Process value
- Voting status

The block icons from the template @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

12.11 VoteDiL - Monitoring block for up to 8 digital process values

12.11.1 VoteDiL description

Object name (type + number) and family

Type + number: FB 1130

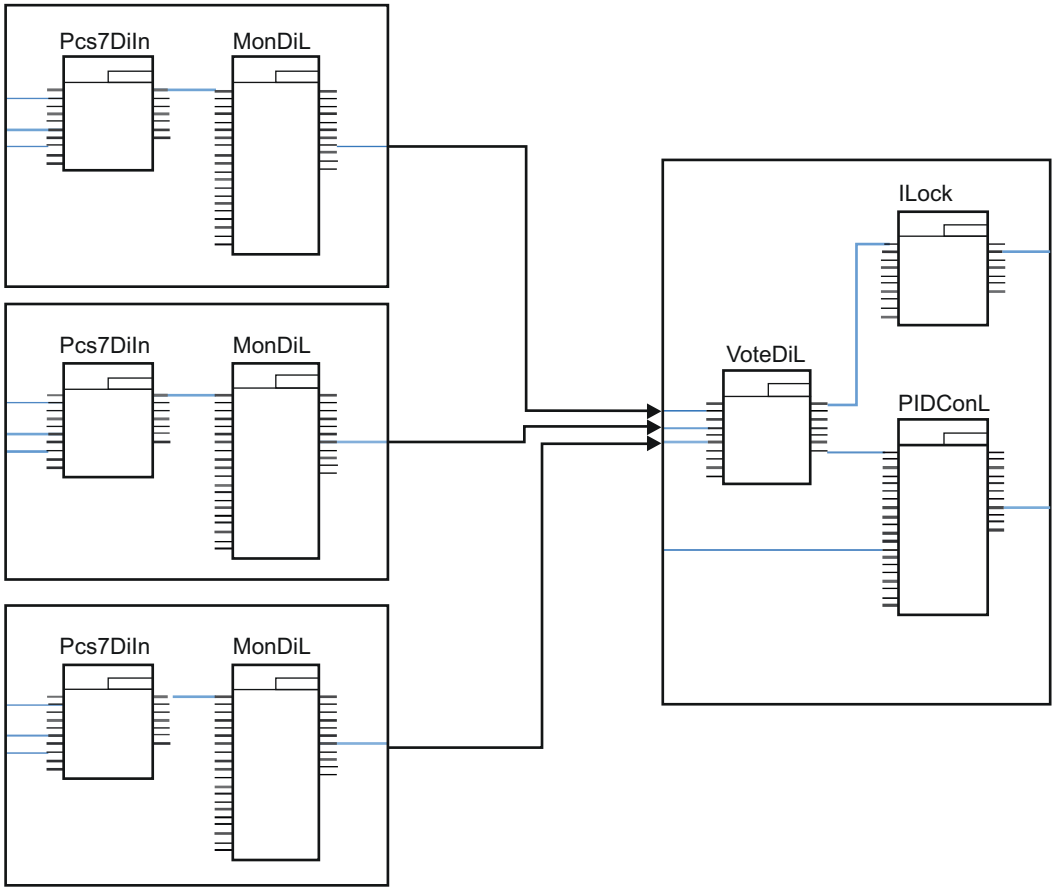
Family: LogicDi

Number of process objects: 1 PO

Area of application of VoteDiL

The block is used to increase the availability of digital process values. Up to 8 process values for signal status and value are checked. The testing of multiple process values increases the availability since individual failures lead to an error.

Up to 8 process values are interconnected at the function block, are shown in the figure.



Startup characteristics

Use the startup characteristics Feature bit 0 "Set startup characteristics" to define the startup characteristics for this block.

The messages are suppressed after startup for the number of cycles set at `RunUpCyc`.

Time response

The block must be called via a cyclic interrupt OB. The sampling time of the block is set in the `SampleTime` parameter.

Called blocks

FC369	SelST16
SFC6	RD_SINFO
SFB35	ALARM_8P

Calling OBs

In the same OB with and after the block whose measured value is to be monitored. Also in OB100.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	Allocated
1	`BatchEn`
2	Not used
3	`OosAct`.Value
4	`OosLi`.Value
5	Not used
6	`OnAct`.Value
7 - 10	Not used
11	`Feature`.Bit2
12 - 13	Not used
14	`Feature`.Bit15
15 - 16	Not used
17	Collection of message delays
18 - 22	Not used
23	Time response active
24	Result of logic operation with inclusion of the output bypass
25	`Out_Byp_Out`.Value
26	Result of logic operation without inclusion of the output bypass

Status bit	Parameter
27	Pre-trip warning
28	Not used
29	All interconnected process value inputs (`PVx`) in simulation and bypass for `Out` are not active.
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word assignment for the parameter `Status2`

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	`MsgLock.Value`
1 - 5	Not used
6	1 = Signal status voting active
7 - 30	Not used
31	`MS_RelOp`

Status word assignment for the parameter `Status3`

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	`PV1`.Value
1	`PV2`.Value
2	`PV3`.Value
3	`PV4`.Value
4	`PV5`.Value
5	`PV6`.Value
6	`PV7`.Value
7	`PV8`.Value
8 - 15	Not used
16	`PV1` QC Bad
17	`PV2` QC Bad
18	`PV3` QC Bad
19	`PV4` QC Bad
20	`PV5` QC Bad
21	`PV6` QC Bad
22	`PV7` QC Bad
23	`PV8` QC Bad
24 - 31	Not used

Status word assignment for the parameter Status4

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	PV1 not interconnected
1	PV2 not interconnected
2	PV3 not interconnected
3	PV4 not interconnected
4	PV5 not interconnected
5	PV6 not interconnected
6	PV7 not interconnected
7	PV8 not interconnected
8 - 15	Not used
16	NOT(PV_1.Value XOR PV1_Inv)
17	NOT(PV_2.Value XOR PV2_Inv)
18	NOT(PV_3.Value XOR PV3_Inv)
19	NOT(PV_4.Value XOR PV4_Inv)
20	NOT(PV_5.Value XOR PV5_Inv)
21	NOT(PV_6.Value XOR PV6_Inv)
22	NOT(PV_7.Value XOR PV7_Inv)
23	NOT(PV_8.Value XOR PV8_Inv)
24 - 31	Not used

Status word assignment for the parameter Status5

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	PV1_Inv
1	PV2_Inv
2	PV3_Inv
3	PV4_Inv
4	PV5_Inv
5	PV6_Inv
6	PV7_Inv
7	PV8_Inv
8 - 15	Not used
16	PV1 Degraded
17	PV2 Degraded
18	PV3 Degraded
19	PV4 Degraded
20	PV5 Degraded
21	PV6 Degraded
22	PV7 Degraded

Status bit	Parameter
23	PV8 Degraded
24 - 31	Not used

Status word assignment for the parameter Status6

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	AV1 not interconnected
1	AV2 not interconnected
2	AV3 not interconnected
3	AV4 not interconnected
4	AV5 not interconnected
5	AV6 not interconnected
6	AV7 not interconnected
7	AV8 not interconnected
8 - 31	Not used

Status word assignment for the parameter Status7

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	PV1_Byp
1	PV2_Byp
2	PV3_Byp
3	PV4_Byp
4	PV5_Byp
5	PV6_Byp
6	PV7_Byp
7	PV8_Byp
8 - 15	Not used
16	PV1 Hidden Bypass
17	PV2 Hidden Bypass
18	PV3 Hidden Bypass
19	PV4 Hidden Bypass
20	PV5 Hidden Bypass
21	PV6 Hidden Bypass
22	PV7 Hidden Bypass
23	PV8 Hidden Bypass
24 - 31	Not used

Status word assignment for the parameter Status8

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	FirstIn PV1
1	FirstIn PV2
2	FirstIn PV3
3	FirstIn PV4
4	FirstIn PV5
5	FirstIn PV6
6	FirstIn PV7
7	FirstIn PV8
8 - 15	Not used
16	PV1_BypLi
17	PV2_BypLi
18	PV3_BypLi
19	PV4_BypLi
20	PV5_BypLi
21	PV6_BypLi
22	PV7_BypLi
23	PV8_BypLi
24 - 31	Not used

Status word assignment for the parameter Status9

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via EventTsIn1
1	Effective signal 2 of the message block interconnected via EventTsIn1
2	Effective signal 3 of the message block interconnected via EventTsIn1
3	Effective signal 4 of the message block interconnected via EventTsIn1
4	Effective signal 5 of the message block interconnected via EventTsIn1
5	Effective signal 6 of the message block interconnected via EventTsIn1
6	Effective signal 7 of the message block interconnected via EventTsIn1
7	Effective signal 8 of the message block interconnected via EventTsIn1
8 - 16	Effective signal 8 to 16 of the message block connected via Event16TsIn1
16 - 31	Not used

Status word assignment for the parameter Status10

For a description of the individual parameters, refer to the section VoteDiL I/Os (Page 654)

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via EventTsIn2
1	Effective signal 2 of the message block interconnected via EventTsIn2
2	Effective signal 3 of the message block interconnected via EventTsIn2
3	Effective signal 4 of the message block interconnected via EventTsIn2
4	Effective signal 5 of the message block interconnected via EventTsIn2
5	Effective signal 6 of the message block interconnected via EventTsIn2
6	Effective signal 7 of the message block interconnected via EventTsIn2
7	Effective signal 8 of the message block interconnected via EventTsIn2
8 - 16	Effective signal 8 to 16 of the message block connected via Event16TsIn2
16 - 31	Not used

12.11.2 VoteDiL operating modes

The block can be operated using the following modes:

- On
- Out of service

General information operating modes is available in the section *Operating modes of the blocks* in the *APL Function manual*.

12.11.3 VoteDiL functions

Handling non-connected inputs

Inputs which are not interconnected are not evaluated. They are not displayed in the faceplate either.

Group display SumMsgAct for limit monitoring, trip, pre-trip and ExtMsgx

The block provides the standard APL function Group display for limit monitoring as well as the functions Group display for deviation monitoring, Trip, Pre-trip and ExtMsgx.

Suppressing messages using the MsgLock parameter

This block includes the standard APL function, "Suppress messages using the MsgLock parameter".

Bypass

Note

In the VoteDiL block, exclusion (bypassing) the process value monitoring means that process value monitoring is excluded from the logic of the block, in other words, this signal is ignored in the logic operation.

This function can only be executed in the faceplate with "high-level operating permission".

You can disable input signals that are temporarily not to be used for the calculation in the block by setting the corresponding I/O.

- Regardless of whether or not `BypLix` is connected, the operator can exclude the bypass signal:
  `PVx_Byp` = 1 (by the operator via faceplate)
- `PVx_BypLi.ST` <> 16#FF (input is connected) and a change to the `PVx_BypLi` value is detected:
  `PVx_BypLi.Value` = 1 (of I/O on CFC)

The I/O is shown in the faceplate by the following symbol:



Depending on the configuration of the Feature bit "Separate evaluation of excluded and simulated interlock signals" (`Feature`.Bit 2), the "Bypass" output is formed and the hidden bypass signal is set at the `Out` output.

Feature bit =0:

If one of the input values is excluded (`PV1_Byp`.. `PV8_Byp` or `PV1_BypLi` .. `PV8_BypLi`), the `BypAct` output is set. The hidden bypass signal at the `Out` output is reset.

- `BypAct`.Value =
  `PV1_Byp` OR `PV1_BypLi` (when `PV1_BypLi` is connected and the value has changed)
  OR `PVx_Byp` OR `PVx_BypLi` (when `PVx_BypLi` is connected and the value has changed)
- `Out`.Bit1 = 0

An excluded input value that is switch-relevant after the exclusion sets the status of `Out` to Simulation.

Feature bit =1:

If one of the input values is excluded (`PV1_Byp`.. `PV8_Byp` or `PV1_BypLi` .. `PV8_BypLi`) or the hidden bypass signal is set at one of the process value inputs (`PV1`.. `PV8`), the `BypAct` output is set. The bypass signal at the `PV1PV8` input can be read from the upstream block at the `BypAct` output. The hidden bypass signal at the `Out` output is set to the value of the `BypAct` .

- `BypAct`.Value =
  `PV1_Byp` OR `PV1_BypLi` (when `PV1_BypLi` is connected and the value has changed)
  OR `PVx_Byp` OR `PVx_BypLi` (when `PVx_BypLi` is connected and the value has changed)
  OR `PV1` = 16#60 OR `PVx` = 16#60
  OR `PV1`.Bit1 OR `PVx`.Bit1 (hidden lines)
- `Out`.Bit1 = `BypAct`.Value

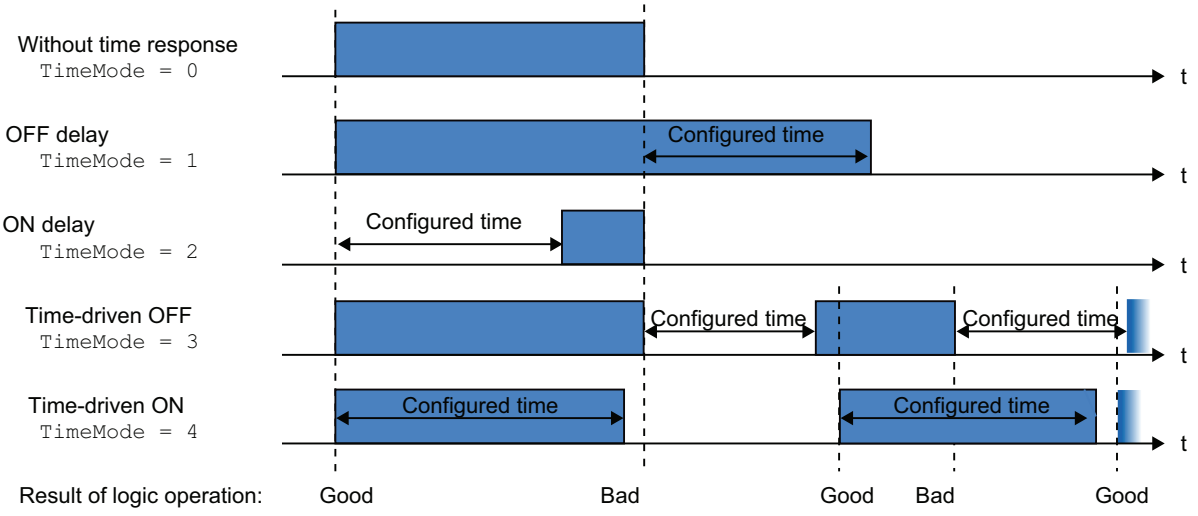
An excluded interlock input has no influence on the status of `Out`.

Time behavior

A time response for the output value can be configured for the block. The `TimeMode` I/O can be used to configure the following time responses:

- `TimeMode` =0: Without time response (default)
- `TimeMode` =1: OFF delay (changing the result of logic operation from good to bad is delayed by `VoteTime`)
- `TimeMode` =2: ON delay (changing the result of logic operation from bad to good is delayed by `VoteTime`)
- `TimeMode` =3: Time-driven OFF (the result of logic operation is bad for the time `VoteTime`)
- `TimeMode` =4: Time-driven ON (the result of logic operation is good for the time `VoteTime`)

The time for the response is specified at the `VoteTime` I/O.



WARNING

Changes to the `TimeMode` connection

If the time response at the `TimeMode` connection is changed during runtime, this may have an effect on the signal state of the block output. The time function has an effect on the output once after the change regardless of the input signals.

Calling additional blocks

This block includes the standard function Call additional faceplates. Information is available in the section *Calling additional faceplates* in the *APL Function Manual*.

First-in signal detection with the block

This block provides the standard APL function, first-in signal detection. Note that a signal status change only has an effect on the initial signal detection when the `Feature` Bit 31 (Enable first-in signal detection) and the `FirstInEn` input are set.

The response to deactivation via the `FirstInEn` input can be influenced by `Feature` Bit 21 (first-in signal acquisition response to deactivation).

Note

This function can only be executed in the faceplate with "process control" operating permission.

Forming the signal status for blocks

This block provides the standard APL function, Form and output the signal status for technological blocks.

The worst signal status `ST_Worst` for the block is formed from the following parameters:

- `Out`.ST (if trip is enabled)

Release for maintenance

This block has the standard APL function, "Release for maintenance".

Signal status voting

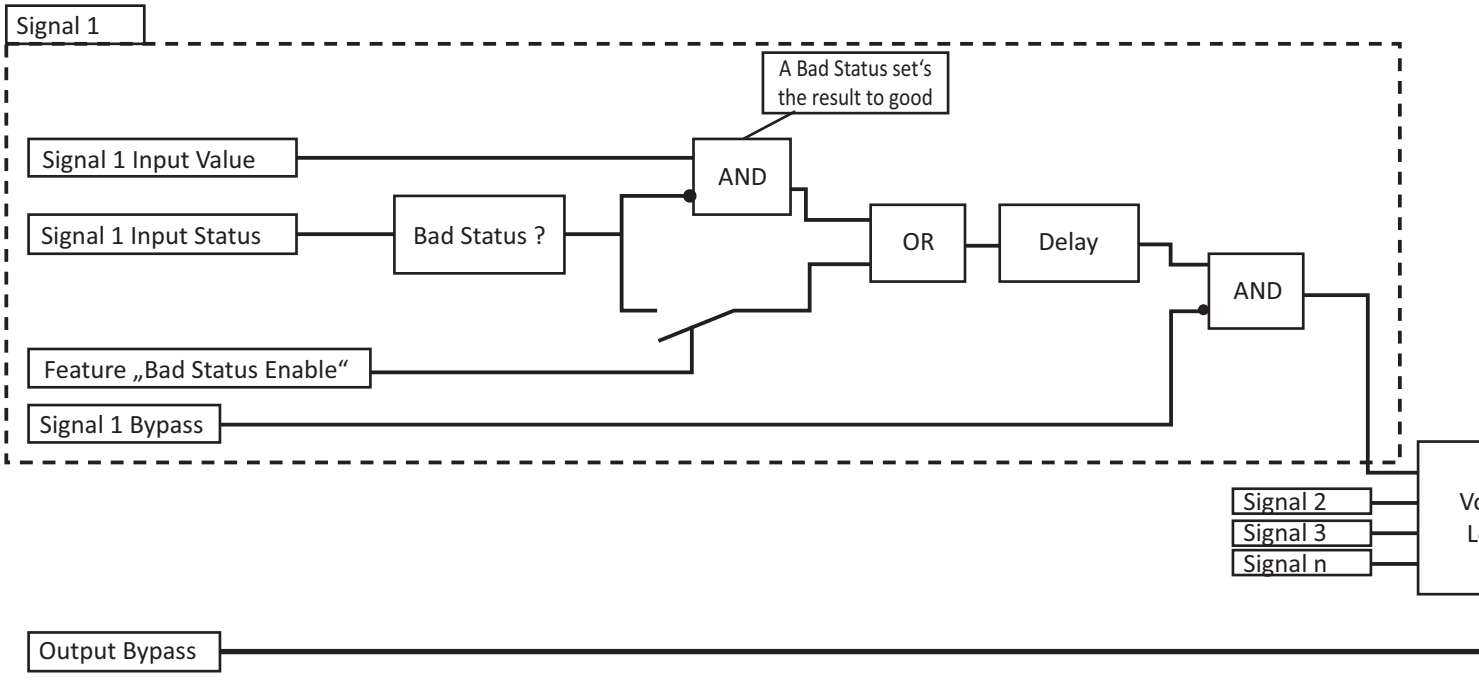
Depending on the configuration of the feature "Signal status voting (`PVx` bad at bad signal status)" (`Feature`.Bit 12), a signal is considered bad due to a bad signal status:

Feature Bit =0:

The process value is evaluated independent of the signal status and further processed.

Feature Bit =1:

The process value is evaluated immediately as bad with bad signal status and further processed.



Degraded Voting (degraded voting)

The input signals of a block can be "excluded from the voting" based on a bad signal status or based on a bypass, or can be evaluated as "bad/good". The behavior depends on the setting of the "Degraded Voting" feature (`Feature`.Bit 13)

Feature Bit =0:

Degraded voting is not active.

Input signals of the block with bad signal status are evaluated as "bad" (`Feature`.Bit 12 =1) or continue to be processed normally (`Feature`.Bit 12 =0). Bypassed input signals are evaluated as "good".

Feature Bit =1:

Degraded Voting is active.

Input signals of the block with poor signal status (depending on `Feature`.Bit 12) and bypassed input signals are excluded from voting. With `Feature`.Bit 12 =1, the input signals with bad signal status are "discarded". With `Feature`.Bit 12 =0, the bad signal status is ignored.

An input is ultimately "Good", "Bad", "Discarded" or "Bypassed". This result enters into the logic function. Discarded or bypassed signals result in a change (reduction) of the logic function. For example, a 2oo3 (2 out of 3) logic reduces to a 2oo2 logic for a bypassed or discarded signal. Another discarded/bypassed signal results in 1oo1 voting. Discarded or bypassed signals have the same effect here – except when all the input signals have been bypassed.

Function 2003			*Result*	
Input signal 1	Input signal 2	Input signal 3	Reduced Function	Logic Result
Good	Good	Good	2oo3	Good
Good	Good	Bad	2oo3	Good
Good	Bad	Bad	2oo3	Bad
Bad	Bad	Bad	2oo3	Bad
Good	Good	Bypassed	2oo2	Good
Good	Bypassed	Bypassed	1oo1	Good
Bypassed	Bypassed	Bypassed	0oo0	Good
Good	Good	Discarded	2oo2	Good
Good	Discarded	Discarded	1oo1	Good
Discarded	Discarded	Discarded	0oo0	Bad
Discarded	Discarded	Bypassed	0oo0	Bad
Discarded	Bypassed	Bypassed	0oo0	Bad

Figure 12-2 Result of the logic function using the example of a 2oo3 function based on differently evaluated input signals

Signal value for tripping

Depending on the parameter assignment of the "Signal value for tripping" feature (`Feature`.Bit 14), the output is 1 or 0 if the logic is fulfilled.

Feature Bit =0:

Deenergize-to-trip (DTT) ⇒ Logic fulfilled: `Out`.Value = 0

Feature Bit =1:

Energize-to-trip (ETT) ⇒ Logic fulfilled: `Out`.Value = 1

Automatic reset

Depending on the parameter assignment of the "Automatic reset" feature (`Feature`.Bit 15), the output is automatically reset after tripping or must be reset by the operator.

Feature Bit =0:

Automatic reset not active The output value becomes good only when the logic is no longer fulfilled and the operator has reset the output.

Feature Bit =1:

Automatic reset active. The output value becomes good as soon as the logic is no longer fulfilled.

Early trigger warning

Depending on the configuration of the Feature "Trigger prewarning active" (`Feature`.Bit 16), a message is generated if only a bad value for triggering the block is missing.

Feature Bit =0:

A message is triggered as a warning.

Feature Bit =1:

No message is triggered.

A bypass for all `PVx`

Depending on the parameter assignment of the feature "A bypass for all `PVx`" (`Feature`.Bit 17) it is possible to only allow the bridging of an input signal.

Feature Bit =0:

There is no limit to setting the bypass.

Feature Bit =1:

A bypass can only be set for one input signal.

Select unit of measure

This block includes the standard APL function, Select unit of measure.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4 - 11	Not used
12	1 = Operator can activate the Release for maintenance function
13 - 31	Not used

The block has the following permissions for the OS1Perm parameter:

Bit	Function
0	1 = Operator can exclude values (bypass)
1	1 = Operator can reset the exclusion of input values
2	1 = Operator can reset the recording of the first signal
3 - 8	Not used
9	1 = Operator can reset the output value
10 - 12	Not used
13	1 = Operators can change VoteTime
14 - 17	Not used
18	1 = Operator can set or reset the exclusion of process value PV1
19	1 = Operator can set or reset the exclusion of process value PV2
20	1 = Operator can set or reset the exclusion of process value PV3
21	1 = Operator can set or reset the exclusion of process value PV4
22	1 = Operator can set or reset the exclusion of process value PV5
23	1 = Operator can set or reset the exclusion of process value PV6
24	1 = Operator can set or reset the exclusion of process value PV7
25	1 = Operator can set or reset the exclusion of process value PV8
26	1 = Operator can set or reset the exclusion of output value Out
27 - 31	Not used

Note

If you interconnect a parameter that is also listed in OS_Perm / OS1Perm as a parameter, you have to reset the corresponding OS_Perm / OS1Perm bit.

Configurable reactions using the Feature parameter

An overview of all the reactions that are provided by the Feature parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
2	See APL function: "Separate evaluation of excluded and simulated interlock signals"
5	Activate OS1Perm Bits 0 = OS1Perm Bits 18...25 disabled (evaluation only in faceplate) 1 = OS1Perm Bits 18...25 enabled (evaluation only in faceplate)
6	Activate OS1Perm Bits 0 = OS1Perm Bit 26 disabled (evaluation only in faceplate) 1 = OS1Perm Bit 26 enabled (evaluation only in faceplate)
12	Signal status voting (PVx bad for bad signal status)

Bit	Function
13	Degraded voting
14	Signal value for tripping
15	Automatic reset
16	Early trigger warning active
17	A bypass for all `PVx`
19	See APL function: "Changes signal status of outputs in OOS"
21	See APL function: "First-in detection response to deactivation"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"
26	See APL function: "Reaction of the switching points in the "Out of service" operating mode"
29	See APL function: "Signal limit violation"
31	See APL function: "Enable first-in detection"

Configurable reactions using the `Feature2` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0 - 31	Not used

Generating instance-specific messages

This block includes the standard APL function, "Generating instance-specific messages".

Specifying the display area for process and setpoint values as well as operations

This block features the standard APL function "Display and operator input area for process values and setpoints".

Time stamp

This block receives a time stamp value via the EventTSIn input parameter. For more information on this, refer to the section "Functions of EventTs" in the APL documentation.

SIMATIC BATCH functionality

This block provides the standard APL function, "SIMATIC BATCH functionality".

12.11.4 VoteDiL error handling

VoteDiL error handling

For information about troubleshooting for all blocks, see also "Troubleshooting" in the Basics of APL section of the APL documentation.

The following errors can be displayed for this block:

- Error numbers

Overview of error numbers

The `ErrorNum` I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
90	Parameter assignment error

12.11.5 VoteDiL messaging

Message response

The following messages can be generated for this block:

- Process messages
- Instance-specific messages

Messages

You can use up to three instance-specific messages with this block.

Message instance	Message identifier	Message class	Event
`MsgEvId1`	SIG 1	Alarm - high	$$BlockComment$$ Trip is enabled
	SIG 2	Warning - high	$$BlockComment$$ Trip pending
	SIG 3	AS control system message - fault	$$BlockComment$$ External message 1
	SIG 4	AS control system message - fault	$$BlockComment$$ External message 2
	SIG 5	AS control system message - fault	$$BlockComment$$ External message 3

Explanation:

$$BlockComment$$: Content of the instance-specific comment

Associated values for message instance `MsgEvId1`

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVa104`
5	`ExtVa105`
6	`ExtVa106`
7	`ExtVa107`
8	`ExtVa108`
9	Reserved
10	Reserved

The associated values 4 ... 8 are allocated to the `ExtVa104` ... `ExtVa108` parameters and are available for use. You can find additional information in the "Process Control System PCS 7 - Engineering System" manual.

12.11.6 VoteDiL I/Os

Input parameters

Parameter	Description	Type	Default
`EventTsIn1`	For interconnecting data between a technology block and the message blocks EventTs, Event16Ts. The input parameter EventTsIn1 is used to interconnect with the output parameter EventTsOut of the EventTs, Event16Ts block. When this interconnection is configured, the messages of the EventTs, Event16TS block are displayed and can also be acknowledged on the OS in the message view of the technological block.	ANY	-
`EventTsIn2`	For interconnecting data between a technology block and the message blocks EventTs, Event16Ts. Input parameter EventTsIn2 is used to interconnect output parameter EventTsOut of the EventTs, Event16Ts block. When this interconnection is configured, the messages of the EventTs, Event16TS block are displayed and can also be acknowledged on the OS in the message view of the technological block.	ANY	-

Parameter	Description	Type	Default
Feature	I/O for additional functions	STRUCT • Bit 0: BOOL • ... • Bit 31: BOOL	- • 0 • 0 • 0
Feature1	I/O for additional functions	STRUCT • Bit 0: BOOL • ... • Bit 31: BOOL	- • 0 • 0 • 0
FirstInEn	First-in signal detection	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
MsgEvId1	Message number (assigned automatically)	DWORD	16#00000000
MsgLock	1 = Suppress process messages. For more information, refer to the "Suppressing messages by means of parameter MsgLock" section.	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OnOp*	1 = "On" mode via operator	BOOL	0
OosLi	1 = "Out of service", via interconnection or SFC (0-1 edge transition)	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OosOp*	1 = "Out of service", via OS operator input	BOOL	0
OpSt_In	Input parameter for local operating permission, to be interconnected with the Out output parameter of the upstream OpStations block	DWORD	16#00000000
OS_Perm	I/O for operator permissions	STRUCT • Bit 0: BOOL • ... • Bit 31: BOOL	- • 1 • 1 • 1
OS1Perm	I/O for operator permissions	STRUCT • Bit 0: BOOL • ... • Bit 31: BOOL	- • 1 • 1 • 1
PV1 .. PV8	Process value 1 .. Process value 8	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
PV1_Inv .. PV8_Inv	Invert process value 1 ... process value 8	BOOL	0
PV1_Byp .. PV8_Byp*	1 =Bypass PV1 .. PV8 via operator	BOOL	0
PV1_BypLi .. PV8_BypLi	1 =Bypass PV1 .. PV8 via interconnection	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF

Parameter	Description	Type	Default
AV1 .. AV8	Associated value 1 ... Associated value 8	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
AV1_Unit .. AV8_Unit	Unit of AV1 .. AV8	INT	0
RstOp*	1 = Reset trip	BOOL	0
RstLi	1 = Reset via interconnection	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
RstFirstOp*	1 = Reset FirstEn	BOOL	0
RstBypOp*	1 = Reset PV1_Byp. PV8_Byp / Out_Byp	BOOL	0
RunUpCyc	Number of cycles in startup; messages are suppressed during these cycles	INT	3
SampleTime	Sampling time [s] (assigned automatically)	REAL	0.1
SelFp1	1 = Call a block saved in this parameter as an additional faceplate in the standard view	ANY	
SelFp2	1 = Call a block saved in this parameter as an additional faceplate in the preview	ANY	
TimeMode	Dynamic response: 0: No time response 1: On delay 2: Off delay 3: Time-controlled OFF 4: Time-driven ON	INT	0
Out_Byp*	Bypass by operator	BOOL	0
UA1unit	Unit of measure for analog auxiliary value 1	INT	0
UA2unit	Unit of measure for analog auxiliary value 2	INT	0
UserAna1	Analog auxiliary value 1	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
UserAna2	Analog auxiliary value 2	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
UserStatus	Freely assignable bits for use in PCS 7 OS	BYTE	16#00
VoteMode	Number of "Bad" or “Error” values that lead to an error at the output. E.g. if voting mode 1oo2 is configured, for two inputs if any one of the input has an error or bad condition the output will also result in an error.	INT	1
VoteTime*	Time of ON/OFF delay or evaluation [s]	REAL	0.0

***** Values can be written back to these inputs during execution of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
BypAct	1 = Bypass active PV1..PV8	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
ErrorNum	Output of pending error number. For error numbers that can be output by this block, see Troubleshooting VoteAnL.	INT	-1
FirstIn	Bit-coded number of the first signal that has resulted in a change at the output	DWORD	16#00000000
MS_Release	Release for maintenance: 1 = Enable for OS operator	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
MsgAckn1	Message - acknowledgment status 1 (ACK_STATE output of the first ALARM_8P)	WORD	16#0000
MsgErr1	1 = Message error 1 (output ERROR of first ALARM_8P)	BOOL	0
MsgStat1	Message status 1 (STATUS output of the first ALARM_8P)	WORD	16#0000
OnAct	1 = "On" mode enabled	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OosAct	1 = Block is "Out of service"	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
OpSt_Out	Value of the input parameter OpSt_In, for further interconnection with other blocks. Bit 31 of this parameter is used by Feature bit 24	DWORD	16#00000000
OS_PermLog	Display of OS_Perm with settings changed by the block algorithm	DWORD	16#FFFFFFFF
OS_PermOut	Display of OS_Perm	DWORD	16#FFFFFFFF
OS1PermLog	Display of OS1Perm with settings changed by the block algorithm	DWORD	16#FFFFFFFF
OS1PermOut	Display of OS1Perm	DWORD	16#FFFFFFFF
Out	Output	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Out_Byp_Out	1 = Bypass active Out	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
PV1_A_Act .. PV8_A_Act	1 = PV1 .. PV8 alarm active. You can change the response for this parameter with Feature Bit 29 (signal limit violation).	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV1_Degrad .. PV8_Degrad	PV1 ..PV8 degraded	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV1_Byp_Out .. PV8_Byp_Out	1 = Bypass active PV1 .. PV8	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV1_Bad_Out .. PV8_Bad_Out	1 = PV1 .. PV8 bad	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
PV1_Out .. PV8_Out	Output process value 1 .. Process value 8	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
ST_Worst	Worst signal status	BYTE	16#80
Status1	Status word 1	DWORD	16#00000000
Status2	Status word 2	DWORD	16#00000000
Status3	Status word 3	DWORD	16#00000000
Status4	Status word 4	DWORD	16#00000000
Status5	Status word 5	DWORD	16#00000000
Status6	Status word 6	DWORD	16#00000000
Status7	Status word 7	DWORD	16#00000000
Status8	Status word 8	DWORD	16#00000000
Status9	Status word 9	DWORD	16#00000000
Status10	Status word 10	DWORD	16#00000000
SumMsgAct	1 = Group message is active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
VoteTime_Out	Remaining delay time [s]	REAL	0.0

12.11.7 VoteDiL block diagram

VoteDiL block diagram

A block diagram is not provided for this block.

12.11.8 Operator control and monitoring

12.11.8.1 VoteDiL views

VoteDiL block views

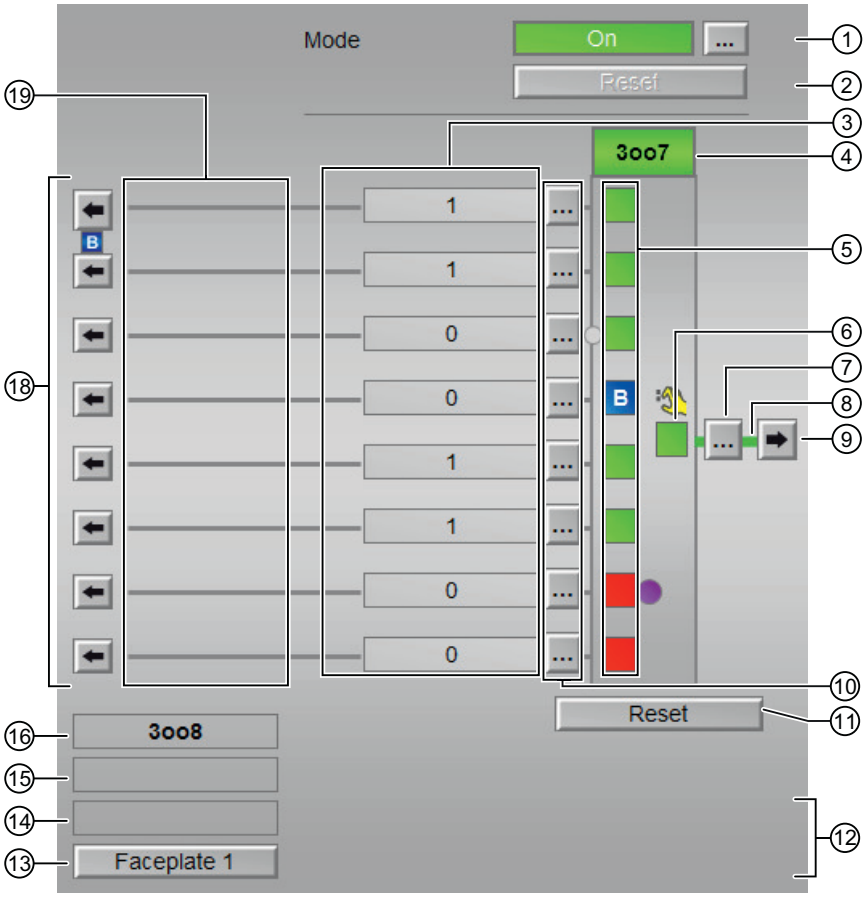
The VoteDiL block provides the following views:

- VoteDiL standard view (Page 660)
- Message view
- Trend view
- VoteDiL parameter view (Page 665)
- VoteDiL preview (Page 667)
- Memo view
- Batch view
- Block icon for VoteDiL (Page 668)

You can find general information on the faceplate and block icon in the "Structure of the faceplate" and "Block icon structure" sections of the APL function manual.

12.11.8.2 VoteDiL standard view

VoteDiL standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

For more information on switching the operating mode, refer to the section "Switching operating states and operating modes" in the APL documentation.

(2) Reset the output value

You can reset the output value with the button. The button is only operable if a reset is required.

The display of this button depends on the configuration in the engineering system (ES).

(3) Display of input values (BOOL) with signal status (in front of the field)

These fields show the value and corresponding interlock information with a signal status:

- 1 = "Good" state
- 0 = "Locked"

Changing the display

You can change the displays for 0 and 1 in the CFC in the object properties of the block:

- Navigate to the I/Os (object properties).
- In the Text 0 and Text 1 columns, change the default for the input parameter (PVc) to what you later want to see in runtime.

These fields show the status information associated with the process values:

Color	Status
	The process value PVx is error-free and will be further processed accordingly.
	The process value PVx has a bad signal status (16#00 or 16#28) and will be further processed as bad.

(4) Status of the output signal of the block

This area (6) shows the status of the output signal of the block (priority from high to low). You can configure the logic in the engineering system (ES).

Color	Logic (voting mode 1oo1, 1oo2, 1oo3, 2oo3 etc.)
Gray	Block is not used
Blue	Excluded (bypass)
Red	Bad state (Out active)
Yellow	Pre-trip warning
Yellow	Simulated
Green	Good state (Out OK)

(5) Display of the status for further processing

The symbol shows the status for further processing of the input values:

Color	Output status
	The process value PVx will be further processed as good.
	The process value PVx will be further processed as bad.
	The process value PVx will be excluded from further processing.

(5) "First-in" status display

The following symbol is displayed next to an input value, if this input value has caused the last output signal change from 1 to 0 (good state to locked):



You can reset the first-in signal with the button (11).

Note

This function can only be executed in the faceplate with "high-level operating permission".

For more information, refer to the section Recording the first signal for interlock blocks in the APL documentation.

For more information, refer to the section Switching operating states and operating modes in the APL documentation.

(6) Display of the output signal

The icon displays the status of the output signal after the execution of the logic:

Color	Output status
	The output signal is further processed as Good.
	The output signal is further processed as Bad.
	The output signal is excluded from further processing.

(7) Exclusion of output signals

You can use the (7) button to exclude the output signal from the logic and place it in the good state. Depending on the previous settings, you can "Set" or "Reset" this property.

If the input value has been excluded, the following symbol appears in the field (6):



For more information, refer to the section Switching operating states and operating modes in the APL documentation.

Note

This function can only be executed in the faceplate with "high-level operating permission".

Note

Operator authorizations via `OS1Perm` depend on the setting of `Feature` bit 5.

You can find additional information in the section Enabling `OS1Perm` bits in the APL documentation.

(8) Status of the block output

The line color indicates the status of the block output:

Color	Output status
Gray	Block is not used (out of service)
Blue	Excluded (bypass)
Yellow	Pre-trip warning or simulated
Red	Output is enabled
Green	Output is disabled

(9) Call faceplate of the output value

When you press the highlighted button, you can open the faceplate associated with the output value. The function of this button depends on the configuration in the engineering system (ES). For more information, refer also to the section "Opening additional faceplates" in the APL documentation.

(10) Exclusion of input signals

You can use the button (10) to exclude input values from processing. Depending on the previous settings, you can "Set" or "Reset" this property.

If the input value has been excluded, the following symbol appears in the field (5):



For more information, refer to the section Switching operating states and operating modes in the APL documentation.

Note

This function can only be executed in the faceplate with "high-level operating permission".

Note

Operator authorizations via `OS1Perm` depend on the setting of `Feature` bit 5.

You can find additional information in the section Enabling `OS1Perm` bits in the APL documentation.

(11) "Reset" the settings for further processing

When you press the button (11), you can "Reset" all input values:

- "Reset exclusions": the exclusions of the input values are reset.
- "Reset FirstIn": First-in detection / status display (5) is reset.

For more information, refer to the section Recording the first signal for interlock blocks in the APL documentation.

(12) Display for auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system. You can find more information on this in the section Displaying auxiliary values in the APL documentation.

(13) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

(14) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

(15) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

(16) Display area for block states

This area provides additional information on the operating state of the block:

- "Trip state"

(18) Call faceplate of the process values

When you press the button (), you can open the faceplate associated with each input value. The function of this button depends on the configuration in the engineering system (ES). Refer the Opening additional faceplates section of the APL documentation.

(19) Display of analog input values

The interconnected analog input values (AVx) are displayed in this area. You set a unit of measure (AVx_Unit) for each input in the engineering system (ES).

Changing the display of the text:

You can change the displays in the CFC in the object properties of the Interlock block:

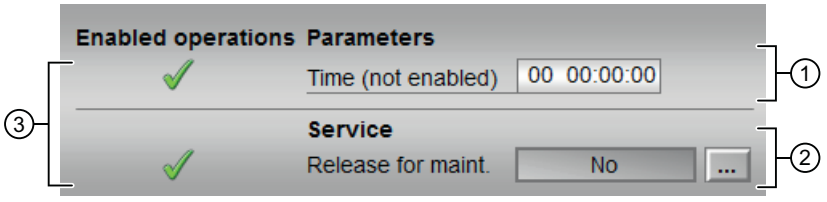
- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the input parameter (PVx) to what you want to see during runtime later.
- In the "Identifier" column, change the default setting for the input parameter (AVx) to what you want to see during runtime later.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding AVx input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (PVx), this text is displayed even then the "Identifier" text field of the input parameter (AVx) is not empty. If the "OS additional text" field of the (PVx) input parameter is empty, the "Identifier" text of the (AVx) input parameter is displayed.

12.11.8.3 VoteDiL parameter view

VoteDiL parameter view



(1) Time response

The time for the time response configured at the block can be set here.

The display depends on the configuration of the time response.

You can influence the following parameters:

- Time (not enabled): A delay is not configured for the output; parameter cannot be manipulated.
- On-delay: On-delay is configured for the output. This can be controlled with corresponding operator authorization.

- Off-delay: Off-delay is configured for the output. This can be controlled with corresponding operator authorization.
- Duration: The output is active time controlled. The minimum time for which the output is active can be controlled with the appropriate operator permission.

(2) Service

You can select the following functions in this area:

- "Release for maintenance"

For more information, refer to the section Switching operating modes in the APL documentation.

You can find information on this area in the section:

- Simulating signals in the APL documentation
- Release for maintenance in the APL documentation

(3) Enabled operations

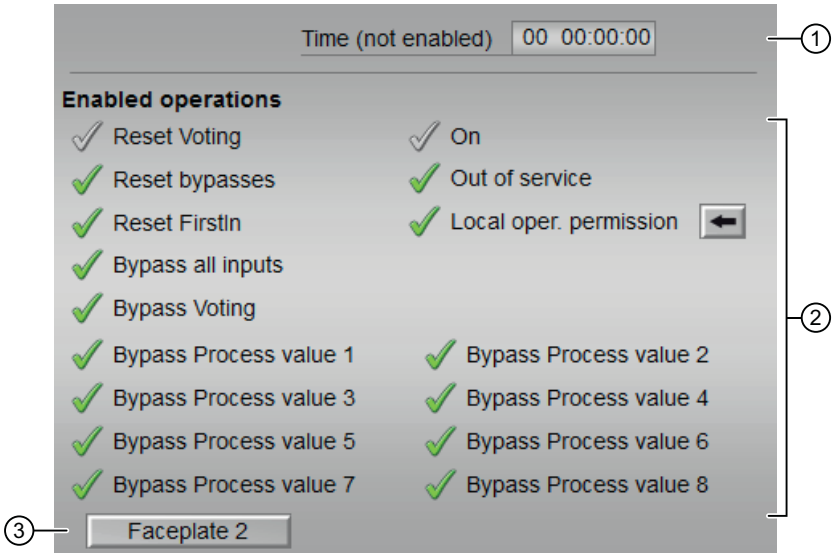
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

12.11.8.4 VoteDiL preview

VoteDiL preview



(1) Time response

The current time for the time response configured for the block is displayed here.

The display depends on the configuration.

You can display the following times here depending on the configuration:

- Time (not enabled): No delay is configured for the output.
- On-delay: On-delay is configured for the output.
- Off-delay: Off-delay is configured for the output.
- Duration: The output is active time controlled.

(2) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- Reset: You can reset the trip.
- Reset: You can reset all the excluded process values.
- Reset: You can reset first-in signal detection
- Set/reset: You can set/reset the exclusion of process values
- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to change to the standard view of the OpStations block. You can find more information on this in the section "Operator control permissions" in the APL documentation.

(3) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

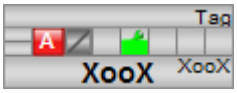
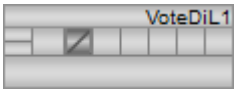
12.11.8.5 VoteDiL block icon

VoteDiL block icons

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of alarm, warning, and deviation limits as well as trip and pre-trip errors
- Operating modes
- Signal status, release for maintenance
- Memo display
- Process value
- Voting status

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	-	Block main icon in the "Out of service" operating mode.

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

Mathematical blocks 13

13.1 AccuS - Accumulates measured values with a specific factor

13.1.1 AccuS description

Object name (type + number) and family

Type + number: FB 1111

Family: Math

Number of process objects: 0 PO

Area of application

The block is used to integrate a measured value in two separate registers. A distinction can be made between a counting pulse or a continuous value during accumulation. The measured value can be multiplied with a specific factor (e.g. specific heat) for the purpose of accumulation.

Time response

The block must be called by means of a cyclic interrupt OB. The sampling time of the block is entered in the SampleTime parameter.

Startup characteristics

During the startup, the counter values are saved using internal variables. A startup can be simulated manually via the Restart input.

Called blocks

FC1	AD_DT_TM
FC369	SelST16
SFC1	READ_CLK
SFC6	RD_SINFO
SFC51	RDSYSST

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section AccuS I/Os (Page 674).

Status bit	Parameter
0	Allocated
1	BatchEn
2	Not used
3	OosAct.Value
4	OosLi. value
5	Not used
6	OnAct. value
7	Not used
8 - 30	Not used
31	MS_Release value

13.1.2 AccuS operating modes

A distinction is made between two different operating modes at the `PulseAct` parameter:

`PulseAct = 0`:

- Measured value as a continuous value
- The `PV` input specifies the process value as a continuous quantity value. The process value is recorded in the `AccuTime` sampling time, weighted using the `BasePV` parameter, and then accumulated.

`PulseAct = 1`:

- Measured value as pulse input
- The process value is transferred to the `Pulse` parameter as a pulse, weighted using the `BasePulse` parameter, and then accumulated.

13.1.3 AccuS functions

Integrate

The measured value (`PulseAct=0:=PV; PulseAct=1`:=Pulse) is accumulated in two counter registers (`AcCnt1Out` and `AcCnt2Out`).

The following parameters must be configured for this:

- PulseAct = 0:
  The measured value is transferred to the PV input. The AccuTime parameter specifies the sampling cycles of the integration in seconds.
  The measured value PV is weighted at the BasePV parameter. It is essential to ensure that the AccuTime and BasePV parameters have the same unit (e.g., BasePV = 3600 must be specified for a measured value with the unit m³/h and an AccuTime of 1 second).
  You can configure a deadband at the DeadBand parameter. When PV < DeadBand, the deadband is not included in the integration. In this case, the BasePulse parameter is not relevant. The measured value is calculated with the following formula:

$$AcCntXOut = AcCntXOut + PV * SpecFact * \frac{AccuTime}{BasePV}$$

- PulseAct = 1:
  The measured value is transferred to the Pulse input as a pulse.
  The pulse is weighted at the BasePulse parameter. In this case, the BasePV, AccuTime parameter is not relevant.

Reset counter:

The counter states are reset via the faceplate and via the interconnectable inputs RstCnt1Li/RstCnt1Li.

The Rst1Date/Rst2Date parameters each specify the time stamp for the reset operations in the format [yy/mm/dd_hh].

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the OS_Perm parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4 - 6	Not used
7	1 = Operator can reset accumulated values
8 - 11	Not used
12	1 = Operator can activate the Release for maintenance function
13 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
24	See APL function: "Enable local operating permission"

13.1.4 AccuS error handling

The block does not have any error handling.

13.1.5 AccuS messaging

The block has no message behavior.

13.1.6 AccuS I/Os

Input parameters

Parameter	Description	Data type	Default
`AcCnt_Unit`	Unit of AcCnt1 / AcCnt2	INT	1001
`AcCnt1`	Accumulated counter #1	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
`AcCnt2`	Accumulated counter #2	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
`AccuTime`	Sample time for accumulation [s]	REAL	1.0
`BasePulse`	Value/Pulse	REAL	0.0
`BasePV`	Factor for PV	REAL	0.0
`Pulse`	Pulse input	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`PulseAct`	Mode of integrating 0:Level;1:Impulse	BOOL	0
`Restart`	Manual restart	BOOL	0

Parameter	Description	Data type	Default
Rst1Date	Date of last counter #1 reset [yy/mm/dd_hh]	DWORD	16#00000000
Rst2Date	Date of last counter #2 reset [yy/mm/dd_hh]	DWORD	16#00000000
RstCnt1Li	Linkable input reset counter #1	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
RstCnt1Op *	Operator input reset counter #1	BOOL	0
RstCnt2Li	Linkable input reset counter #2	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
RstCnt2Op *	Operator input reset counter #2	BOOL	0
SpecFact *	Specific factor	REAL	1.0

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
AcCnt1Out	Accum. counter #1	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
AcCnt2Out	Accum. counter #2	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
AcDiCnt1Out	Accum. counter #1 (DINT)	DINT	0
AcDiCnt2Out	Accum. counter #2 (DINT)	DINT	0
ErrBasePV	Error : Parameter BasePV = 0	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
ErrSpecFact	Error : Parameter SpecFact <= 0	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MS_Release	1: MS release	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
O_MS_Ext	Reserved	DWORD	16#00000000

Parameter	Description	Data type	Default
OnAct	On mode is active	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
OosAct	Out of service is active	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

13.1.7 AccuS block diagram

The block has no block diagram.

13.1.8 Operator control and monitoring

13.1.8.1 AccuS views

AccuS block views

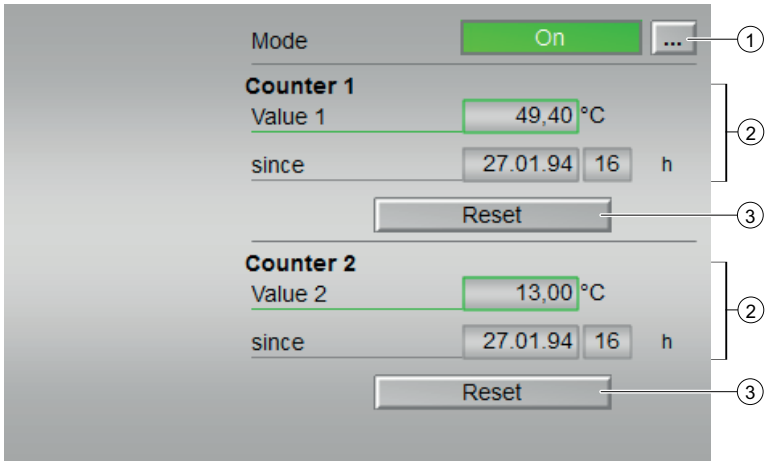
The AccuS block provides the following views:

- AccuS standard view (Page 677)
- AccuS parameter view (Page 678)
- AccuS preview (Page 679)
- Memo view
- Batch view
- AccuS block icons (Page 679)

You can find general information on the faceplate and block icon in the "Structure of the faceplate" and "Block icon structure" sections of the APL function manual.

13.1.8.2 AccuS standard view

AccuS standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

For information on the operating mode, refer to the section "Operating modes for the blocks without 'Manual' and 'Automatic operation'" in the APL manual. You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(2) Counter

This area is used to display the integrated measured value in the respective tab.

Time stamp of the last reset of the associated tab.

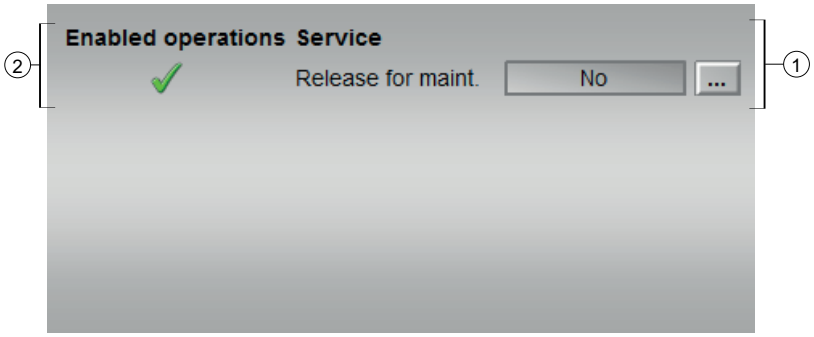
This area shows the time of the last reset process for the displayed process value.

(3) Reset

Operation of the input `RstOp`. This button sets the respective tab back to zero.

13.1.8.3 AccuS parameter view

AccuS parameter view



(1) Service

You can select the following functions in this area:

- "Release for maintenance"

For more information, refer to the section Switching operating modes in the APL documentation.

You can find information on this area in the section:

- Release for maintenance

in the APL documentation

(2) Enabled operations

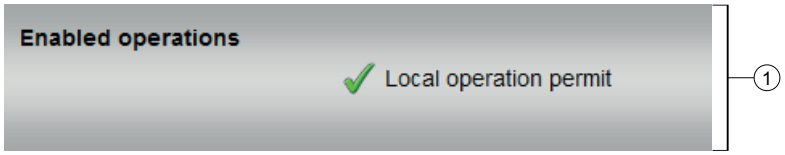
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

13.1.8.4 AccuS preview

AccuS preview



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- "Local operating permission": Use the ← button to change to the standard view of the OpStations block. You can find more information on this in the section "Operator control permissions" in the APL documentation.

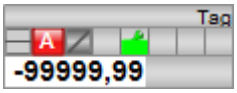
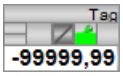
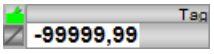
13.1.8.5 AccuS block icons

AccuS block icons

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Alarm and warning limits overshoot/undershoot
- Operating modes
- Signal status, release for maintenance
- Memo display
- Process value

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	
	3	
	4	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

Motor and valve blocks

14

14.1 VlvDiv - Diverter valve

14.1.1 VlvDiv description

Object name (type + number) and family

Type + number: FB 1127

Family: Drives

Number of process objects: 1 PO

Area of application of VlvDiv

The block is used for the following tasks:

- Control and monitoring of a 3-way valve (`VlvType` = 3 parameter)
- Control and monitoring of a 5/4-way valve (`VlvType` = 4 parameter)
- Control and monitoring of a 9/8-way valve (`VlvType` = 8 parameter)

Note

The `VlvDiv` parameter is limited to the range 2 to 8 by the block.

Functional principles

The diverter valve is controlled by a control signal. The valve always travels by exactly one position. A control reset pulse is output when the next position is reached. The number of available routes is specified by the VlvType input parameter. After leaving the highest position (`VlvType`), the request returns to position 1.

The control is monitored by means of the position signals "`Position1`" to "`Position8`" (feedbacks).

Various inputs are available for control purposes. The following sections provide more detailed information on configuration, operating principles, visualization and operation.

Configuration

Install the block in a cyclic interrupt OB (OB30 to OB38) in the CFC Editor. The block is also installed automatically in the startup OB (OB100).

Startup characteristics

Use the Feature Bit 'Setting the startup characteristics' to define the startup characteristics for this block.

Following startup, the messages are suppressed for the number of cycles assigned in the `RunUpCyc` parameter.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section VlvDiv I/Os (Page 697).

Status bit	Parameter
0	`Occupied`
1	`BatchEn`
2	`SimOn`
3	`OosAct`.Value
4	`OosLi`.Value
5	`AutoAct`.Value
6	`LocalAct`.Value
7	0: Open padlock in block icon 1: Closed padlock in block icon
8	Feedback error without control change
9	Feedback error due to control change
10	Undefined position with deactivated feedback monitoring (e.g. two unexpected feedbacks)
11	Bypass in Simulation/Local mode active
12	Invalid signal status
13	Mode switchover error
14	1 = `Intlock` is active
15	1 = `Permit` is active
16	1 = `Protect` is active
17	`CtrlChnST` interconnected
18	`CtrlRstChnST` interconnected
19 - 23	Not used
24	Automatic preview `Pos1`
25	Automatic preview `Pos2`
26	Automatic preview `Pos3`
27	Automatic preview `Pos4`
28	Automatic preview `Pos5`
29	Automatic preview `Pos6`
30	Automatic preview `Pos7`
31	Automatic preview `Pos8`

Status word assignment for the parameter `Status2`

For a description of the individual parameters, refer to the section VlvDiv I/Os (Page 697).

Statusbit	Parameter
0	1 = Message suppression active`MsgLock`
1	Not used
2	1 = Display for interlocks in block icon
3	`WarnAct`.Value
4	External error generated by `FaultExt` or external `CSF` control system fault when Feature bit 18 is set See also the section "Activate error state with external control system fault CSF" in the APL documentation.
5	Error in diverter valve for status display
6	Message condition for panel block
7	Not used
8	`CtrlCorrOut`.Value
9	1 = Valve opens
10	1 = Position 1 is used (`VlvType` >= 1)
11	1 = Position 2 is used (`VlvType` >= 2)
12	1 = Position 3 is used (`VlvType` >= 3)
13	1 = Position 4 is used (`VlvType` >= 4)
14	1 = Position 5 is used (`VlvType` >= 5)
15	1 = Position 6 is used (`VlvType` >= 6)
16	1 = Position 7 is used (`VlvType` >= 7)
17	1 = Position 8 is used (`VlvType` = 8)
18	Reset request in automatic
19	1 = Input signals have no impact on "Local" mode when `LocalSetting` = 2 and `LocalSetting` = 4
20	`Ctrl`.Value
21	`P_CtrlRst`.Value
22	`FbkP1Out`.Value
23	`FbkP2Out`.Value
24	`FbkP3Out`.Value
25	`FbkP4Out`.Value
26	`FbkP5Out`.Value
27	`FbkP6Out`.Value
28	`FbkP7Out`.Value
29	`FbkP8Out`.Value
30	1 = Bypass information from the preceding function block
31	`MS_RelOp`

Status word assignment for the parameter `Status3`

For a description of the individual parameters, refer to the section VlvDiv I/Os (Page 697).

Status bit	Parameter
0	1 = "Interlock" button is enabled
1	1 = "Permission" button is enabled
2	1 = "Protection" button is enabled
3	`Pos1Out`
4	Monitoring error in Position 1
5	`Pos2Out`
6	Monitoring error in Position 2
7	Pos3Out
8	Monitoring error in Position 3
9	`Pos4Out`
10	Monitoring error in Position 4
11	`Pos5Out`
12	Monitoring error in Position 5
13	`Pos6Out`
14	Monitoring error in Position 6
15	`Pos7Out`
16	Monitoring error in Position 7
17	`Pos8Out`
18	Monitoring error in Position 8
19 - 22	Not used
23	Monitoring time deviation
24	`UserAna1` interconnected
25	`UserAna2` interconnected
26	Show automatic preview in the standard view
27	Not used
28	`GrpErr`.Value
29	`RdyToStart`.Value
30 - 31	Not used

Status word assignment for the parameter `Status4`

For a description of the individual parameters, refer to the section VlvDiv I/Os (Page 697).

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via `EventTsIn`
1	Effective signal 2 of the message block interconnected via `EventTsIn`
2	Effective signal 3 of the message block interconnected via `EventTsIn`
3	Effective signal 4 of the message block interconnected via `EventTsIn`
4	Effective signal 5 of the message block interconnected via `EventTsIn`
5	Effective signal 6 of the message block interconnected via `EventTsIn`

Status bit	Parameter
6	Effective signal 7 of the message block interconnected via EventTsIn
7	Effective signal 8 of the message block interconnected via EventTsIn
8 - 31	Not used

14.1.2 VlvDiv operating modes

The block can be operated using the following modes:

- Local mode
- Automatic mode
- Manual mode
- Out of service

The next section provides additional block-specific information relating to the general descriptions.

Local mode

You can find general information on the local mode, changing modes and bumpless changeover in the APL documentation, section "Local mode".

In "local mode" you can:

- move the diverter valve to position 1 (Pos1Local = 1)
- move the diverter valve to position 2 (Pos2Local = 1)
- move the diverter valve to position 3 (Pos3Local = 1)
- move the diverter valve to position 4 (Pos4Local = 1)
- move the diverter valve to position 5 (Pos5Local = 1)
- move the diverter valve to position 6 (Pos6Local = 1)
- move the diverter valve to position 7 (Pos7Local = 1)
- move the diverter valve to position 8 (Pos8Local = 1).

The availability of the individual commands depends on the number of existing routes (VlvType parameter). Position commands outside the configured range (Pos7Local and Pos8Local with VlvType = 6), are not evaluated by the block and are always overwritten with 0.

When you set the block to "Local mode", the control is effected either by the "local" signals (Pos1Local, Pos2Local, Pos3Local, Pos4Local, Pos5Local, Pos6Local, Pos7Local or Pos8Local input parameters) or by the feedback signals (FbkP1, FbkP2, FbkP3, FbkP4, FbkP5, FbkP6, FbkP7 or FbkP8 input parameters; if a position cannot be assigned, the last valid position is assumed). You can configure the LocalSetting input parameter (0,1,2,3,4) accordingly.

"Manual" mode

You will find general information on "Manual" mode, changing modes, and bumpless switchover in the APL documentation. Please see the section titled "Manual and automatic mode for motors, valves and dosers".

In "Manual mode" you can:

- move the diverter valve to position 1 (`Pos1Man` = 1)
- move the diverter valve to position 2 (`Pos2Man` = 1)
- move the diverter valve to position 3 (`Pos3Man` = 1)
- move the diverter valve to position 4 (`Pos4Man` = 1)
- move the diverter valve to position 5 (`Pos5Man` = 1)
- move the diverter valve to position 6 (`Pos6Man` = 1)
- move the diverter valve to position 7 (`Pos7Man` = 1)
- move the diverter valve to position 8 (`Pos8Man` = 1).

The availability of the individual commands depends on the number of existing routes (`VlvType` parameter). Position commands outside the configured range (`Pos7Man` and `Pos8Man` with `VlvType = 6`), are not evaluated by the block and are always overwritten with 0.

"Out of service"

You can find general information about the "Out of service" mode in the "Out of service" chapter of the APL documentation.

The switch can only be set to "Out of service" if it is not controlled. During a control until it reaches the target position, the `OS_Perm.Bit3` is reset.

The switch remains in the current position after switchover to "Out of service".

Automatic mode

You will find general information on "Automatic" mode, changing modes, and bumpless switchover in the APL documentation. Please see the section titled "Manual and automatic mode for motors, valves and dosers".

In "Automatic mode" you can:

- move the diverter valve to position 1 (`Pos1Aut` = 1)
- move the diverter valve to position 2 (`Pos2Aut` = 1)
- move the diverter valve to position 3 (`Pos3Aut` = 1)
- move the diverter valve to position 4 (`Pos4Aut` = 1)
- move the diverter valve to position 5 (`Pos5Aut` = 1)
- move the diverter valve to position 6 (`Pos6Aut` = 1)
- move the diverter valve to position 7 (`Pos7Aut` = 1)
- move the diverter valve to position 8 (`Pos8Aut` = 1).

The availability of the individual commands depends on the number of existing routes (`VlvType` parameter). Position commands outside the configured range (`Pos7Aut` and `Pos8Aut` with `VlvType = 6`), are not evaluated by the block and are always overwritten with 0.

14.1.3 VlvDiv functions

VlvDiv functions

The functions for this block are listed below.

Calling further faceplates

This block features the standard function "Opening additional faceplates". Information is available in the section *Opening additional faceplates* in the *APL Function manual*.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can switch to automatic mode
1	1 = Operator can switch to manual mode
2	1 = Operator can switch to local mode
3	1 = Operator can switch to Out of service mode
4	1 = Operator can move the diverter valve to position 1
5	1 = Operator can move the diverter valve to position 2
6	1 = Operator can move the diverter valve to position 3
7	1 = Operator can move the diverter valve to position 4
8	1 = Operator can move the diverter valve to position 5
9	1 = Operator can move the diverter valve to position 6
10	1 = Operator can move the diverter valve to position 7
11	1 = Operator can move the diverter valve to position 8
12	1 = Operator can operate the 'Control' maintenance function
13	1 = Operator can operate the 'Reset' maintenance function
14	1 = Operator can reset the diverter valve
15	1 = Operator can define the monitoring time for startup
16	1 = Operator can define the monitoring time for the run time
17	1 = Operator can activate the monitoring time function (bits 8 & 9)
18	1 = Operator can change the pulse duration for the control reset pulse
19	1 = Operator can enable the Simulation function
20	1 = Operator can activate the Release for maintenance function

Bit	Function
21	1 = Operator can operate the 'Correction signal' maintenance function
22-31	Not used

Interlocks

This block provides the following interlocks:

- Activation enable
- Interlock without reset ("Interlock")
- Interlock with reset ("Protection")

For more information see the section "Interlocks" in the APL documentation.

Disabling interlocks

This block features the standard APL function "Disabling interlocks".

Resetting the block in case of interlocks

This block features the standard APL function "Resetting the block in case of interlocks or errors".

Outputting a signal for start readiness

This block features the standard APL function "Outputting a signal for start readiness".

Forming the group status for interlocks

This block provides the standard APL function "Forming the group status for interlock information".

Forming the signal status for blocks

This block features the standard APL function "Forming and outputting the signal status for technological blocks".

The worst signal status ST_Worst for the block is formed from the following parameters:

- `FbkP1Out.ST`
- `FbkP2Out.ST`
- `FbkP3Out.ST`
- `FbkP4Out.ST`
- `FbkP5Out.ST`
- `FbkP6Out.ST`
- `FbkP7Out.ST`
- `FbkP8Out.ST`

- Ctrl.ST
- LocalLi.ST
- Pos1Local.ST
- Pos2Local.ST
- Pos3Local.ST
- Pos4Local.ST
- Pos5Local.ST
- Pos6Local.ST
- Pos7Local.ST
- Pos8Local.ST
- Pos1Aut.ST (only if Feature2.Bit10 =1)
- Pos2Aut.ST (only if Feature2.Bit10 =1)
- Pos3Aut.ST (only if Feature2.Bit10 =1)
- Pos4Aut.ST (only if Feature2.Bit10 =1)
- Pos5Aut.ST (only if Feature2.Bit10 =1)
- Pos6Aut.ST (only if Feature2.Bit10 =1)
- Pos7Aut.ST (only if Feature2.Bit10 =1)
- Pos8Aut.ST (only if Feature2.Bit10 =1)

Consider bad quality of the automatic commands or the external values

When the Feature bit "Consider bad quality of the automatic commands or the external values" (Feature2.Bit10) is set to 1 and one of the following parameters has a bad signal status (16#00 or 16#28), the block switches to "Automatic" mode in the last valid position:

- Pos1Aut.ST
- Pos2Aut.ST
- Pos3Aut.ST
- Pos4Aut.ST
- Pos5Aut.ST
- Pos6Aut.ST
- Pos7Aut.ST
- Pos8Aut.ST

Monitoring the feedbacks

This block features the standard APL function "Monitoring the feedbacks".

The behavior of the monitoring is set by switching 'Monitoring' on/off and by setting the monitoring times MonTiDynamic and MonTiStatic.

Behavior with Monitoring on

Static monitoring (valve is in a valid position)
The following errors are monitored when the valve is in a valid position:

- More than two feedbacks result in an immediate error message (static error)
- Feedback change without control results in an error message (static error) as soon as a new, valid position is reached.
  If a new, valid position is not reached, the error message (static error) is generated after expiration of the monitoring time `MonTiStatic`.
  If the valve is in local mode 2/4, an error message is not generated but the valve detects a position command and waits for the new position.

Dynamic monitoring (valve is in motion)

If the valve is in motion, which means a new position is assumed or the valve block expects a new position (local mode 2/4), the following errors are monitored:

- The dynamic monitoring time `MonTiDynamic` is started when the motion starts. If the expected position is not reached within the time (for example, position 2 when the valve was in position 1), an error message (dynamic error) is generated.
- An error message (static error) is output immediately when an unexpected position is reached.
- More than two feedbacks result in an immediate error message (static error).
- Two feedbacks immediately result in an error message (static error), if the combination is not permitted (if the valve moves from position 1 to position 2, feedbacks 1 and 2 are permitted but not any other combinations).

Behavior with Monitoring off

If monitoring is switched off, no error messages are generated and errors must not be reset even in automatic mode. In case of unexpected or incorrect feedbacks, the block changes to the 'Undefined position' status. The block does not accept any new position commands in this status. The block changes back to the 'Valid position' status as soon as a valid position has been reached.

Static monitoring (valve is in a valid position)
The following errors are monitored when the valve is in a valid position:

- More than two feedbacks result in the status 'Undefined position'.
- Feedback change without control immediately results in the status 'Undefined position'
  If the valve is in local mode 2/4, the valve does not assume the 'Undefined position' status but recognizes a position command and waits for the new position.

Dynamic monitoring (valve is in motion)

If the valve is in motion, which means a new position is assumed or the valve block expects a new position (local mode 2/4), the following errors are monitored:

- The status 'Undefined position' is assumed immediately when an unexpected position is reached.
- More than two feedbacks result in the status 'Undefined position'.
- Two feedbacks immediately result in the status 'Undefined position', if the combination is not permitted (if the valve moves from position 1 to position 2, feedbacks 1 and 2 are permitted but not any other combinations).

Suppress messages using the MsgLock parameter

This block includes the standard APL function "Suppressing messages using the MsgLock parameter".

Release for maintenance

This block features the standard APL function "Release for maintenance".

Specifying warning times for control functions

This block features the standard APL function "Specifying warning times for control functions at motors and valves".

The warning signal is output before the diverter valve is moved to a new position. Warning signals can be generated in the following modes:

- Manual mode (`WarnTiMan` input parameter)
- Automatic mode (`WarnTiAut` input parameter)

You specify the warning times in seconds using the input parameters `WarnTiMan` and `WarnTiAut`. If, for example, a new route is controlled, this is displayed at the output parameter with `WarnAct = 1`. The diverter valve then moves after the set warning time has expired and `WarnAct` returns to 0.

A corresponding warning is not output if the values specified for the warning times (`WarnTiMan` or `WarnTiAut`) are lower than the SampleTime parameter.

Simulating signals

This block features the standard APL function "Simulating signals".

If the block is in simulation, the feedbacks, the control function and the control reset pulse are simulated in the block. The outputs `Ctrl` and `P_CtrRst` cannot be activated.

Selecting a unit of measure

This block provides the standard function 'Selecting a unit of measure'.

Idle state

This block features the standard APL function "Neutral position for motors, valves and controllers".

The neutral position always represents the deenergized state. This means for the diverter valve that it rests and that there is no control, regardless of the current position.

The neutral position is assumed when:

- The runtime monitoring was addressed (see 'Specifying startup characteristics')
- One of the interlock conditions is active (see 'Interlocks')

Generate instance-specific messages

This block features the standard APL function "Generating instance-specific messages".

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
2	See APL function: "Reset the commands for switching over the mode"
3	See APL function: "Activate resetting of commands for the control settings"
4	See APL function: "Set switch or button mode"
5	Set control reset pulse for feedback error
6	0 = turning CW 1 = turning CCW
9	See APL function: "Reset via input signals in the event of interlocks (protection) or errors"
10	See APL function: "Exit 'Local' mode"
11	Enable run time for feedback signals in simulation mode
17	See APL function: "Bumpless switchover to 'Automatic' mode for valves, motors, and dosers"
18	Valves, motors or dosers display external error when CSF is active
19	See APL function: "Changes signal status of outputs in OOS"
21	See APL function: "Bumpless switchover to automatic mode for operator only"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"
27	See APL function: "Interlock display with LocalSetting 2 or 4"
30	See APL function: "Set the reset depending on the operating mode or the LiOp parameter"
31	See APL function: "Activate reset of protection/fault in manual mode"

In button mode (Bit 4 = 0), automatic commands are saved in "Automatic mode", which means `Pos1Aut`, `Pos2Aut`, `Pos3Aut`, `Pos4Aut`, `Pos5Aut`, `Pos6Aut`, `Pos7Aut`, `Pos8Aut` can be reset to 0 after switching to the selected position. In manual and local modes, however, the automatic commands are not saved, and in the absence of automatic commands the position is tracked.

In switching mode (Bit 4 = 1), all positions are selected with static signals via the inputs `Pos1Aut`, `Pos2Aut`, `Pos3Aut`, `Pos4Aut`, `Pos5Aut`, `Pos6Aut`, `Pos7Aut` and `Pos8Aut`.

Configurable reactions using the `Feature2` parameter

Bit	Function
2	See APL function: "Separate evaluation of excluded and simulated interlock signals"
5	See APL function: "Evaluation of the signal status of the interlock signals"
10	See APL function: "Consider bad quality of the automatic commands or the external values"

Displaying auxiliary values

This block features the standard APL function "Displaying auxiliary values".

Time stamping

This block obtains a time stamp value via the EventTSIn input parameter. For more information see the section "Functions of EventTs" in the APL documentation.

SIMATIC BATCH functionality

This block provides the standard APL function "SIMATIC BATCH functionality".

Labeling of buttons

This block features the standard APL function "Labeling of buttons and text".

Instance-specific texts can be configured for the following parameters:

- `Pos1Man`
- `Pos2Man`
- `Pos3Man`
- `Pos4Man`
- `Pos5Man`
- `Pos6Man`
- `Pos7Man`
- `Pos8Man`

Position of the diverter valve

When requesting a new route or a new position (`PosxMan`, `PosxAut`, `PosxLocal`), the `Ctrl` output is addressed until the next valid position is reached. Next the `Ctrl` output is reset and the control reset pulse `P_CtrlRst` is set for the duration `PulseWidthCtrlRst`.

The described procedure is repeated until the target position has been reached. The output and the control reset pulse is set and reset for any position that has to be reached or has been reached.

Manual control of switch (maintenance function)

The `Ctrl` output can be set directly with the 'Control' maintenance function. The requirements are that the value is not in motion and that the `P_CtrlRst` output is not set.

Manual reset of switch (maintenance function)

The `P_CtrlRst` output can be set directly with the 'Reset' maintenance function. The requirements are that the valve is not in motion and that the `Ctrl` output is not set.
Unlike with normal operation, an output pulse is not generated but the output is actually set or reset.

'Correction signal' output which can be operated manually (maintenance function)

The `CtrlCorrOut` output can be set and reset directly with the 'Correction signal' maintenance function. The requirement is that the interconnectable input `CtrlCorr_En` is set.

The function does not have a direct impact on the function of the valve block. It is used to implement application-specific functions in CFC and to make it operable from the OS for maintenance purposes.

14.1.4 VlvDiv error handling

For information about error handling for all blocks, please also refer to the basic information in the "Error handling" section of the APL documentation.

The following errors can be displayed for this block:

- Error numbers
- Mode switchover error
- Invalid input signals

Overview of error numbers

The ErrorNum I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when inserting the block; the block is not processed
0	There is no error.
41	The value for the `LocalSetting` I/O is not within the approved limit from 0 to 4.
42	`LocalSetting` = 0 or `LocalSetting` = 3 or `LocalSetting` = 4 and `LocalLi` = 1

Error number	Meaning of the error number
51	For ModLiOp = 1: • AutModLi = 1 and ManModLi = 1 If "Local" mode is active: • More than one input Pos1Local ... Pos8Local = 1 When "Automatic" mode is active: • More than one input Pos1Aut ... Pos8Aut = 1
52	LocalAct = 1 and LocalSetting = 2 or 4 and SimOn = 1

Mode switchover error

This error can be output by the block, see the "Error handling" section of the APL documentation.

Invalid input signals

This error can be output by the block, see the "Error handling" section of the APL documentation.

External error (FaultExt), external control system fault (CSF)

This block can connect an external error with the FaultExt parameter or an external control system fault with the CSF parameter.

Group error

This block features the standard APL function "Outputting group errors".

The following parameters are taken into consideration when generating the group error:

- CSF
- MonDynErr
- MonStaErr
- MonPosErr
- FaultExt

14.1.5 VlvDiv messaging

Message behavior

The following messages can be generated for this block:

- Control system fault
- Instance-specific messages

Control system fault

The following messages can be generated for this block:

Message instance	Message ID	Message class	Event
MsgEvID1	SIG 1	AS process control message - fault	$$BlockComment$$ Feedback error
	SIG 2	AS process control message - fault	$$BlockComment$$ External error has occurred

Explanation:

$$BlockComment$$: content of the instance-specific comment

You can interconnect an external fault (signal) to the input parameter `CSF`. If this signal changes to `CSF` = 1, a control system fault is triggered (MsgEvId1, SIG 2).

Instance-specific messages

You can use up to three instance-specific messages with this block.

Message instance	Message ID	Message class	Event
MsgEvID1	SIG 3	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 4	AS process control message - fault	$$BlockComment$$ External message 2
	SIG 5	AS process control message - fault	$$BlockComment$$ External message 3

Explanation:

`$$BlockComment$$`: content of the instance-specific comment

Associated values for the message instance MsgEvId1

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVa105`
5	`ExtVa106`
6	`ExtVa107`
7	`ExtVa108`
8	reserved
9	reserved
10	reserved

14.1.6 VlvDiv I/Os

Input parameters

Parameter	Description	Data type	Default
CtrlCorr_En	1=Correction signal is enabled	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
CtrlCorrMan *	1=Correction signal in manual mode.	BOOL	0
CtrlMan *	1=Control command in manual mode.	BOOL	0
CtrlRstMan *	1=Control reset command in manual mode.	BOOL	0
FbkP1 ... FbkP8	1=Feedback state for position 1 ... 8	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Feature	I/O for additional functions (Page 687)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 0 • .. • 0
Feature2	I/O for additional functions (Page 687)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 0 • .. • 0
OS_Perm	I/O for operator permissions (Page 687)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 1 • .. • 1
Pos1Aut ... Pos8Aut	1= Position 1 ... 8 command in auto mode.	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Pos1Local ... Pos8Local	1=Position 1 ... 8 command in local mode.	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Pos1Man ... Pos8Man*	1= Position 1 ... 8 command in manual mode.	BOOL	0
PulseWidthCtrlRst *	Control reset output pulse width [s]	REAL	3.0
VlvType *	Count of available positions	INT	4

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
Ctrl	Control output	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
CtrlAnaOut	PosXOut analog	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
CtrlCorrOut	Correction signal	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
FbkAnaOut	Analog value feedback Position	INT	0
FbkP1Out ... FbkP8Out	Feedback Position 1 ... 8 (original or simulated)	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
LastValidPos	Analog value for last valid position	INT	1
P_Ctrl	Pulsive control output	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
P_CtrlRst	Control reset output	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Pos1 ... Pos8	1=Pos1 ... 8 is reached	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Pos1Out ... Pos8Out	Position 1 ... 8 is active	BOOL	0

14.1.7 VlvDiv block diagram

A block diagram is not provided for this block.

14.1.8 Operator control and monitoring

14.1.8.1 VlvDiv views

VlvDiv block views

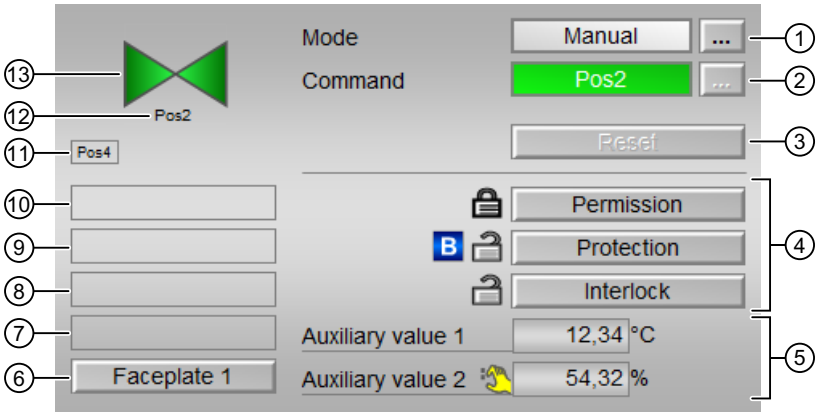
The VlvDiv block provides the following views:

- VlvDiv standard view (Page 699)
- Message view
- Trend view
- VlvDiv parameter view (Page 702)
- VlvDiv preview (Page 704)
- Memo view
- Batch view
- VlvDiv block icons (Page 708)

You can find general information on the faceplate and block icon in the "Structure of the faceplate" and "Block icon structure" sections of the APL function manual.

14.1.8.2 VlvDiv standard view

VlvDiv standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- Manual mode
- Automatic mode

- Local mode
- Out of service

You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(2) Open and close the valve

This area shows you the operating state set for the valve. The following states can be shown and executed here:

- "Pos1" to "Pos8"

You can learn about the switchover of the state in the section "Switchover of operating state and operating mode" in the APL manual.

The availability of the individual commands depends on the number of existing routes (VlvType parameter). With VlvType = 4, for example, only positions 1 to 4 can be selected.

If texts are configured for these commands, they are displayed as state text and as button labels during command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

(3) Reset the block

Click "Reset" in the event of interlocks or errors. You can find more information on this in the section "Resetting the block in case of interlocks or errors" in the manual of the APL.

(4) Operating range for the interlock functions of the block

This display is only visible when the corresponding block input is interconnected.

You can use this button to control the interlock functions of the block. You can find more information on this in the section "Interlock functions" in the manual of the APL.

The following is displayed in addition to the buttons:

- Interlock status (see section "Group status for interlock information" in the manual of the APL), e.g.:



- Signal status (see section "Forming and outputting the signal status for technological blocks" in the APL manual)



- Bypass information (see section "Forming the group status for interlock information" in the manual of the APL):



(5) Display of auxiliary values

This display is only visible when the corresponding block input is interconnected.

You can use this area to display two auxiliary values that have been configured in the engineering system (ES). You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(6) Navigation button for switching to the standard view of any faceplate

This display is only visible when the corresponding block input is interconnected.

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(7) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

You can find additional information on this in the section "Release for maintenance" in the manual of the APL.

(8) Display area for block states

This area provides additional information on the operating state of the block:

- "Time delay"

You can find more information on this in the section "Display of delay times" in the manual of the APL.

(9) Display area for block states

This area provides additional information on the operating state of the block:

- "End position error"
- "Control error"
- "Undefined position"
- "Invalid signal"
- "Switchover error"
- "External error"

You can find more information on this in the sections "Monitoring the feedbacks" and "Error handling" (subsections "Invalid input signals" as well as "Mode switchover error" and "Motor protection function") in the manual of the APL

(10) Display area for block states

This area provides additional information on the operating state of the block:

- "Request 0/1": A reset to "automatic mode" is expected.

(11) Automatic preview

This display is only visible in "Manual mode", in "Local mode", or with a reset request in "Automatic mode", when the current output signals are not identical to the control in "Automatic mode".

The display shows what state the valve would assume if you switched from "manual" or "local" mode to "automatic mode", or performed a reset to "automatic mode".

(12) Status display of the valve

This area shows the active position feedback (black font). If no position feedback is active, the last valid position feedback (gray font) is displayed.

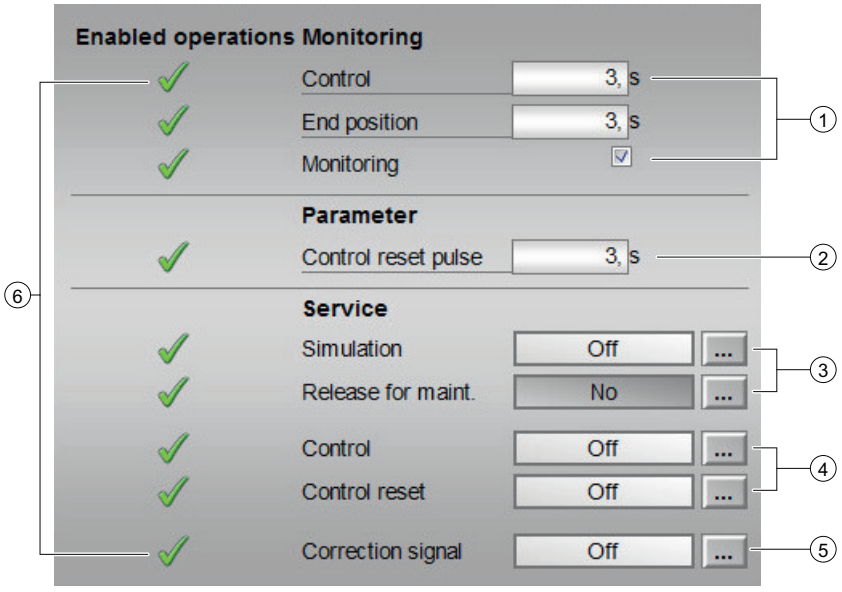
(13) Status display of the valve

The current status of the valve is graphically displayed here.

You can find more information about this in the section VlvDiv block icons (Page 708)

14.1.8.3 VlvDiv parameter view

VlvDiv parameter view



(1) Monitoring

In this area, you change parameters and thereby influence the valve. For more on this, see section "Changing values" in the APL manual.

You can influence the following parameters:

- "Control": Monitoring time during position change of the diverter valve (dynamic)
- "Runtime": Monitoring time during standstill of the diverter valve (static)

Enable monitoring

You can enable monitoring by selecting the check box (☑)

You can find more information about this in the section "Monitoring the feedbacks" in the manual of the APL.

(2) Parameters

In this area, you change parameters and thereby influence the diverter valve. For more on this, see section "Changing values" in the APL manual.

You can influence the following parameters:

- "Control reset pulse": Time for duration of the control reset pulse

(3) Service

You can select the following functions in this area:

- "Simulation"
- "Release for maintenance" (with display for a maintenance demanded)

You can find more information on this in the section "Switching operating states and operating modes" in the manual of the APL.

You can find information on this area in the APL manual section:

- "Simulating signals"
- "Release for maintenance"

(4) Maintenance functions, control / reset

You can select the following functions in this area:

- "Control" (direct control of the Ctrl output signal)
- "Control reset" (direct control of the P_CtrlRst output signal)

(5) Maintenance function, correction signal

You can select the following functions in this area:

- "Correction signal" (direct control of the CtrlCorrOut output signal)

(6) Enabled operations

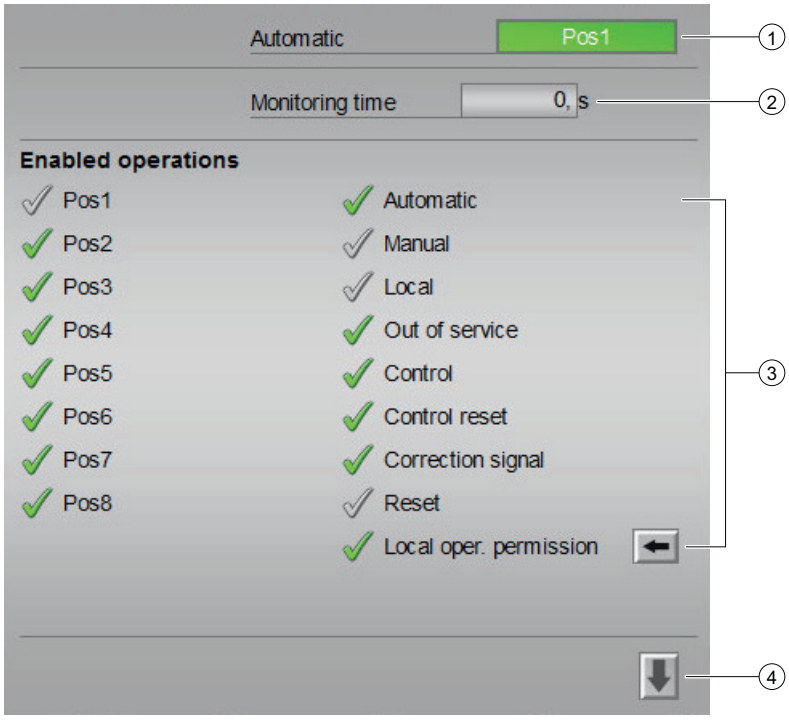
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

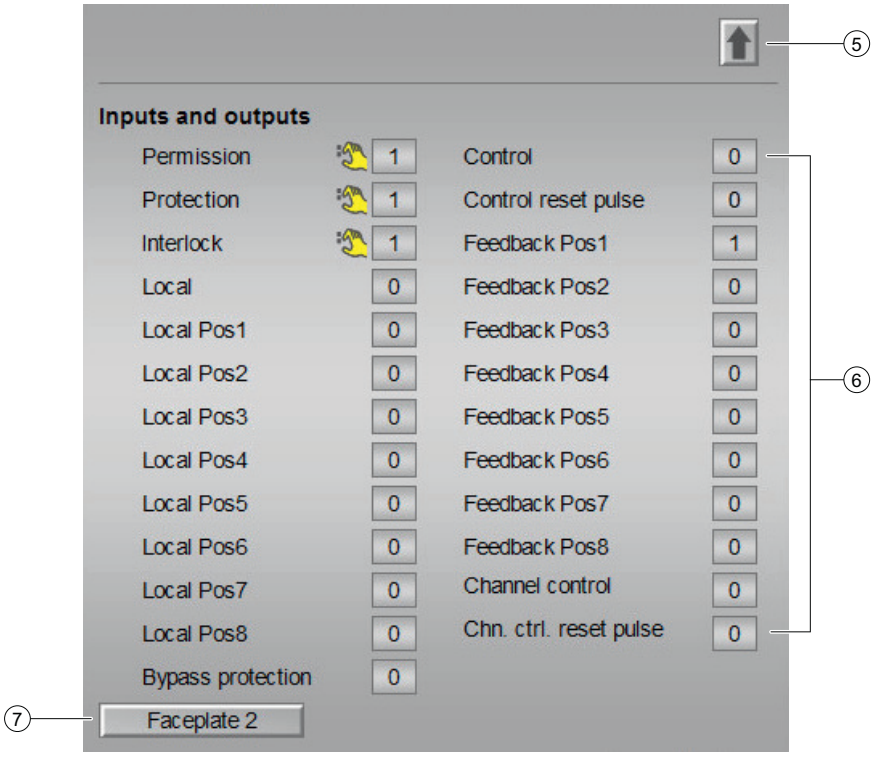
- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

14.1.8.4 VlvDiv preview

VlvDiv preview



The preview has an upper half and a lower half. You can navigate between the two halves with the arrow keys.



(1) Automatic preview

This area shows you the block status after it has switched from "Manual" or "Local" mode to "Automatic" mode.

If the block is in automatic mode, the current block state is displayed.

This area also shows the worst signal status of the following automatic commands:

- `Pos1Aut`
- `Pos2Aut`
- `Pos3Aut`
- `Pos4Aut`
- `Pos5Aut`
- `Pos6Aut`
- `Pos7Aut`
- `Pos8Aut`

(2) Preview area

- "Monitoring time": Display of the current monitoring time

This display is only visible when monitoring is activated.

(3) Operator control enable

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: the OS operator cannot control this parameter due to the configured AS operator permissions (OS_Perm or OS1Perm).

The following enabled operations are shown here:

- "Pos1" ... "Pos8": You can move the diverter valve to "Position 1" to "Position 8"
  If a text is copied for one of the commands, it is also displayed in parentheses. For more information, refer to the section "Labeling of buttons and text" in the APL documentation. Depending on the VlvType parameter, only the used positions are displayed. With VlvType = 4, for example, only positions 1 to 4 are visible.
- "Automatic": You can switch to "automatic mode".
- "Manual": You can switch to "manual mode".
- "Local": You can switch to "local mode".
- "Out of service": You can switch to "Out of service" operating mode.
- "Control": You can control the maintenance function "Control".
- "Control reset": You can control the maintenance function "Control reset".
- "Correction signal": You can control the correction signal.
- "Reset": You can reset the diverter valve in case of errors.
- "Local operating permission": Use the <-- button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(4) Arrow key

Arrow key for changing to the lower section of the preview.

(5) Arrow key

Arrow key for changing to the upper section of the preview.

(6) Display current control signals

This area shows the most important parameters for this block with the current control:

- "Permission":
  This display is only visible when the corresponding block input is interconnected.
  - 0 =No OS activation enable for the valve
  - 1 = Enable for "starting"/"stopping" from the idle state
- "Protection":
  This display is only visible when the corresponding block input is interconnected.
  - 0 = Protective interlocking is in effect, you must reset the block once the interlocking condition has disappeared
  - 1 = Good state
- "Interlock":
  This display is only visible when the corresponding block input is interconnected.
  - 0 = Interlocking without reset is active; you can operate the block without reset once the interlocking condition has disappeared
  - 1 = Good state
- "Local": 1 = Control signal for "Local Mode" (LocalLi) is active
- "Local Pos1" ... "Local Pos8": 1 = Block was moved to "Position 1" ... "Position 8" in "Local mode"

Depending on the VlvType parameter, the local signals are only displayed for the used positions. With VlvType = 4, for example, only signals for the positions 1 to 4 are visible.

- "Bypass protection": 0 = Exclusion disabled, 1 = Exclusion of interlock in "Local mode" and in simulation
- "Control": 1 = Control signal for diverter valve
- "Control reset pulse": 1 = Control reset signal (pulse) for diverter valve
- "Feedback Pos1" ... "Feedback Pos8": 1 = Diverter valve is in "Position 1" ... "Position 1"
- "Channel control"
- "Channel reset pulse"

Depending on the VlvType parameter, the feedback is only displayed for the used positions. With VlvType = 4, for example, only the signals for positions 1 to 4 are visible. This feedback signals are only visible when the corresponding block input is connected.

(7) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

14.1.8.5 VlvDiv block icons

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of alarm, warning, and tolerance limits as well as the control system fault
- Operating modes
- Signal status, release for maintenance
- Interlocks
- Memo display
- Valve status display

The block icons from template @TemplateILV8.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	Main block icon in the full display
	2	Icon 2 for symbolic status display of the diverter value
	3	Icon 3 for symbolic display of a position feedback
	-	Block main icon in "Out of service" mode (example of block icon type 1), without process tag number (APLShowTag = false)

Additional information on the block icon and the control options in the block icon is available in the following sections of the APL documentation:

- Configuring the block icons
- Block icon structure
- Operation via the block icon

Icon 1:

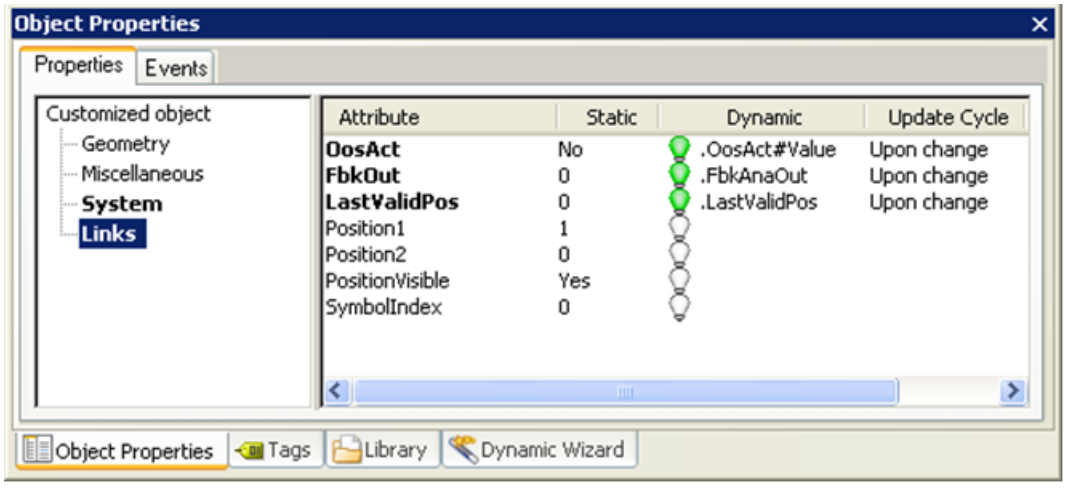
The main icon is placed by default during automatic creation of the screen objects in the process picture. It includes all standard information, such as group display and tag name and is required for each instance.

Icon 2:

The icon 2 can also be used from the @TemplateILV8.pdl screen to represent the status information of the diverter valve as well as the valve status display.

Icon 3:

The icon 3 also gives you the option to adapt the individual position feedbacks graphically to your needs. You can place an icon 3 for each position feedback from the screen @TemplateILV8.pdl in the process picture and connect it with the process tag.
You still have to configure the position to be displayed in the object properties of the icon. The following parameters are available at the icon for configuration:



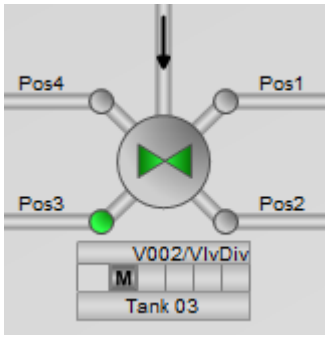
OosAct:	Shows if the block is Out of service
FbkOut:	Includes the current position feedback
LastValidPos:	Includes the last valid position feedback
Position1:	Number to be configured of the position feedback for which this object is used
Position2:	Number of the second position feedback for diverter valves without separate inlet nozzle, otherwise always 0
PositionVisible:	YES = Value at "Position1" is displayed at icon in runtime
SymbolIndex:	Is described by scripts and displays the respective status

Diverter valves with separate inlet (e.g., 5/4 diverter valve):

Each outlet nozzle has its own position feedback in this case. The set route is always from the inlet nozzle to the outlet nozzle.

To display the correct position feedback, the "Position1" icon parameter must be configured with the corresponding position number. The "Position2" icon parameter must be configured with "0".

5/4 diverter valve:



9/8 diverter valve:

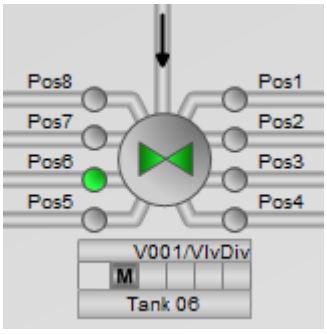


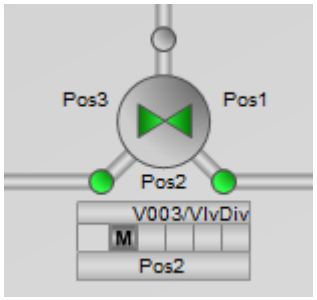
Table 14-1 Configuring for 5/4 diverter valve:

Additional icon 2 top right **Active at Position 1**	**Additional icon 2 bottom right** **Active at Position 2**	**Additional icon 2 bottom left** **Active at Position 3**	**Additional icon 2 top left** **Active at Position 4**
"Position1" = 1 "Position2" = 0	"Position1" = 2 "Position2" = 0	"Position1" = 3 "Position2" = 0	"Position1" = 4 "Position2" = 0

Diverter valve without separate inlet (e.g., 3 diverter valve):

The diverter valve has no separate inlet in this particular case. The valve always connects exactly two nozzles for each set route. To represent a connection of the active nozzles, the additional icons at the "Position1" and "Position2" parameters must be configured with the position numbers of the two possible feedbacks.

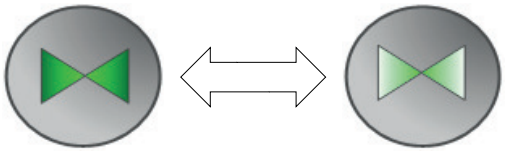
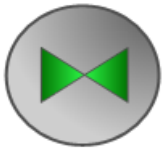
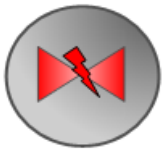
Position feedback 2 is active in the example below. The configuration of the three additional icons must look as follows:



Additional icon 2 top **active for position 1 and 3**	**Additional icon 2 right** **active for position 1 and 2**	**Additional icon 2 left** **active for position 2 and 3**
"Position1" = 1 "Position2" = 3	"Position1" = 1 "Position2" = 2	"Position1" = 2 "Position2" = 3

Valve status display

The following valve states are shown here:

Icon	Meaning
	Diverter valve moves to new position
	Diverter valve stands in defined position (one feedback active)
	Error at diverter valve (monitoring error)
	Diverter valve out of service

14.2 VlvDsL - Double seat valve

14.2.1 VlvDsL description

Object name (type + number) and family

Type + number: FB 1103

Family: Drives

Number of process objects: 1 PO

Functional principles

The standard version of a double seat valve usually has the basic setting "normally closed" to guarantee safe isolation of the two pipe systems in a zero current or depressurized state. In this state, the double seat valve is equivalent to two crossed, separate pipes.

When the valve is open, the medium flows freely between the upper and lower valve bodies, i.e. between the two pipe systems.

The valve closes and the double seat (with double seal) forms a separation chamber (leakage room) between the two pipe systems. In this way, a CIP (= cleaning process) can take place safely and with the complete separation of the two media, without risk of intermixture with production.

In addition, the double seat valve has two separately executed seat liftings, so that it is possible to clean a valve body, including the safety chamber. A pulsed control method is frequently used for the seat lifting.

Operation & monitoring, as well as control/monitoring of the seat liftings, are independent of "normal" valve functions. However, the two seat liftings can be interlocked one against the other.

Startup characteristics

Use the Feature Bit 'Setting the startup characteristics' to define the startup characteristics for this block.

Following startup, the messages are suppressed for the number of cycles assigned in the `RunUpCyc` parameter.

Called blocks

FC369	SEL ST16
SFB35	ALARM_8P
SFC6	RD_SINFO
SFC20	BLK MOV

Calling OBs

The cyclic interrupt OB in which you install the blocks OB30 to OB38. Also in OB100.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section VlvDsL I/Os (Page 724).

Status bit	Parameter
0	Allocated
1	`BatchEn`
2	`SimOn`
3	`OosAct.Value`
4	`OosLi.Value`
5	`AutAct.Value`
6	`LocalAct.Value`
7	0 = Open padlock in the block icon 1 = Closed padlock in the block icon
8	"Open"/"Closed" command (1 = "Open")
9	`FbkOpenOut.Value`
10	`FbkCloseOut.Value`
11	Feedback error without control change
12	Feedback error due to control change
13	`BypProt`
14	Invalid signal status
15	Mode switchover error
16	1 = Intlock is active
17	1 = Permit is active
18	1 = Protect is active
19	`OpenForce.Value`
20	`CloseForce.Value`
21	`Force`
22	Automatic preview (1 = "Open")
23	Smooth switchover in "Automatic mode"
24	`SafePos`
25	`UserAna1.ST` interconnected
26	`UserAna2.ST` interconnected
27	`Occupied2`
28	`BatchEn2`
29	`FbkSLTop.Value`
30	`FbkSLBtm.Value`
31	External error generated by `FaultExt` or external `CSF` control system fault when `Featurebit18` is set

Status word assignment for the parameter `Status2`

For a description of the individual parameters, refer to the section VlvDsL I/Os (Page 724).

Status bit	Parameter
0	`MsgLock.Value`
1	Not used
2	1 = Display for interlocks in block icon
3	1 = Top seat lifting is in use
4	1 = Bottom seat lifting is in use
5 - 18	Not used
19	1 = Input signals have no impact on "Local" mode when `LocalSetting` = 2 and `LocalSetting` = 4
20	1 = Valve open
21	1 = Valve closed
22	1 = Valve opens
23	1 = Valve closes
24	`SLTop.Value`
25	`SLBtm.Value`
26	`SLTopForce.Value`
27	`SLBtmForce.Value`
28	`SLPuPaMode`
29	Signal status for the output signal `Ctrl` of the control
30	Bypass information from preceding function block
31	`MS_RelOp`

Status word assignment for the parameter `Status3`

For a description of the individual parameters, refer to the section VlvDsL I/Os (Page 724).

Status bit	Parameter
0 - 7	Effective signal 1 to 8 of the message block connected via `EventTsIn`
8	"Interlock" button is enabled
9	"Permission" button is enabled
10	"Protection" button is enabled
11	1 = Permit Bypass active
12	1 = Interlock Bypass active
13	1 = Protect Bypass active
14	1 = Feature Bit Interlock Monitoring active
15	1 = Monitoring time active
16	`xOpSLTopMan`
17	`xOpSLBtmMan`
18	1 = Input parameter Top seat lifting `SLTop` is interconnected
19	1 = Input parameter Bottom seat lifting `SLBottom` is interconnected
20	1 = Input parameter `FbkClose` is interconnected

Status bit	Parameter
21	1 = Input parameter `FbkOpen` is interconnected
22	1 = Input parameter `FbkSLTop` is interconnected
23	1 = Input parameter `FbkSLBtm` is interconnected
24 - 25	Not used
26	Show automatic preview in the standard view
27	Not used
28	`GrpErr.Value`
29	`RdyToStart.Value`

14.2.2 VlvDsL operating modes

The block can be operated using the following modes:

- "Local" mode
- "Automatic" mode
- "Manual" mode
- "Out of service"

The block can be operated with all standard operating modes.

The next section provides additional block-specific information relating to the general descriptions.

"Local" mode

You will find general information on "Local" mode, changing modes, and bumpless switchover in the APL documentation. Please see the section titled "Local mode".

Valve actions you can control in "Local" mode:

- Open (`OpenLocal = 1`)
- Close (`CloseLocal = 1`)
- Activate upper seat lifting (`SLTopLocal = 1`)
- Activate lower seat lifting (`SLBtmLocal = 1`)

A block operated in "Local" mode is controlled either by local signals or by feedback signals (input parameters `FbkOpen` and `FbkClose`; if no position can be assigned, the last valid position is accepted). You can configure the `LocalSetting` input parameter accordingly.

"Automatic" mode

You will find general information on "Automatic" mode, changing modes, and bumpless switchover in the APL documentation. Please see the section titled "Manual and automatic mode for motors, valves and dosers".

Valve actions you can control in "Automatic" mode:

- Open (`OpenAut = 1`)
- Close (`CloseAut = 1`)
- Activate upper seat lifting (`SLTopAut = 1`)
- Activate lower seat lifting (`SLBtmAut = 1`)

"Manual" mode

You will find general information on "Manual" mode, changing modes, and bumpless switchover in the APL documentation. Please see the section titled "Manual and automatic mode for motors, valves and dosers".

Valve actions you can control in "Manual" mode:

- Open (`OpenMan = 1`)
- Close (`CloseMan = 1`)
- Activate upper seat lifting (`SLTopMan = 1`)
- Activate lower seat lifting (`SLBtmMan = 1`)

14.2.3 VlvDsL functions

The functions for this block are listed below.

- The seat liftings function independently of any existing interlocks (`Permit, Intlock, Protect`).
- Seat lifting pulses can be generated internally or specified externally.
  - `SLPuPaMode`:
    1=External pulse/pause mode
    0=Internal
- The pulses can be configured separately for both seat liftings:
  `SLTopPulse, SLTopPause, SLBtmPulse, SLBtmPause`
- The open/close/control signals may be configured with a delay in "Automatic" mode (`OpnAutDelay`, `ClsAutDelay`).
- During seat cleaning, the valve feedback signal can sometimes get lost in the case of certain valve types. In these cases, the monitoring error can be suppressed. The suppression is configurable by means of a "feature bit".

Opening additional faceplates

This block features the standard APL function "Opening additional faceplates".

Operator authorizations

This block features the standard APL function "Operator permissions". The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	1= Operator can switch to "Auto" mode
1	1= Operator can switch to "Manual" mode
2	1= Operator can switch to "Local" mode
3	1 = Operator can switch to "Out of service" mode
4	1= Operator can open the valve
5	1= Operator can close the valve
6	1= Operator can reset the valve
7	1 = Operator can change the monitoring time for startup
8	1 = Operator can change the monitoring time for run
9	1 = Operator can activate the monitoring function (bits 7 & 8)
10	1 = Operator can activate the "Simulation" function
11	1 = Operator can activate the "Release for maintenance" function
12	1 = Operator can activate TOP seat lifting
13	1 = Operator can activate BOTTOM seat lifting
14	1 = Operator can change the pulse/pause times for TOP seat lifting
15	1 = Operator can change the pulse/pause times for BOTTOM seat lifting
16	1 = Operator can change the OPEN delay time
17	1 = Operator can change the CLOSE delay time

Interlocks

This block provides the following interlocks:

- Activation enable
- Interlock without reset (interlock)
- Interlock with reset (protection)

You will find further information on this subject in the "Interlocks" section of the "APL documentation".

Disabling interlocks

This block features the standard APL function "Disabling interlocks".

Resetting the block in case of interlocks

This block features the standard APL function "Reset the block in case of interlocks or errors".

Group error

This block features the standard APL function "Output group errors".

The following parameters are taken into consideration when generating the group error:

- CSF
- MonDynErr
- MonStaErr
- FaultExt

Outputting a signal for start readiness

This block features the standard APL function "Outputting a signal for start readiness".

Forming the group status for interlocks

This block features the standard APL function "Forming the group status for interlock information".

Forming the signal status for blocks

This block features the APL standard function "Forming and outputting signal status for blocks".

The worst signal status for the block is generated from the following parameters:

- FbkOpenOut.ST
- FbkCloseOut.ST
- FbkSLTopOut.ST
- FbkSLBtmOut.ST
- Permit.ST
- Intlock.ST
- Protect.ST
- CtrlChnST.ST
- OpenAut.ST (only if Feature2.Bit10 =1)
- CloseAut.ST (only if Feature2.Bit10 =1)
- SLTopAut.ST
- SLBtmAut.ST

Consider bad quality of the automatic commands or the external values

When the Feature bit "Consider bad quality of the automatic commands or the external values" (Feature2.Bit10) is set to 1 and one of the following parameters has a bad signal status (16#00 or 16#28), the block switches to "Automatic" mode in the last valid position:

- OpenAut.ST
- CloseAut.ST

Force operating states

This block features the standard APL function "Force operating states".

- The `OpenForce`, `CloseForce` inputs force the block to open or close.
- The `SLTopForce`, `SLBtmForce` inputs force the seat lifting (top, bottom) to open or close.

Feedback messages

Monitoring the feedbacks

This block features the standard APL function "Monitoring the feedbacks".

- Monitoring of the startup characteristics is set at the `MonTiDynamic` parameter.
- The `MonTiStatic` parameter monitors compliance with the position.

Deactivate feedback

This block features the standard APL function "Deactivate feedback for valves".

- Feedback monitoring can be deactivated separately for each feedback with `NoFbkOpen`, `NoFbkClose`, `NoFbkSLTop`, or `NoFbkSLBtm` as required.

Suppress messages using the MsgLock parameter

This block features the standard APL function "Suppress messages using the MsgLock parameter".

Release for maintenance

This block features the standard APL function "Release for maintenance".

Specifying warning times for control functions

This block features the standard APL function "Specifying warning times for control functions at motors and valves". The warning signal is output before the valve moves away from the safe position. No signal is output in respect of movement toward the safe position.

You can generate warning signals when, for example, valves open. Warning signals can be generated in the following modes:

- Manual mode (`WarnTiMan` input parameter)
- Automatic mode (`WarnTiAut` input parameter)
- Local mode (input parameter `WarnTiMan`) only if `Feature2`.Bit8 = 1

You specify the warning times in seconds using the input parameters `WarnTiMan` and `WarnTiAut`. If, for example, a valve opens, this is displayed at the output parameter with `WarnAct = 1`. The valve then opens after the set warning time has expired and `WarnAct` then returns to 0.

A corresponding warning is not output if the values specified for the warning times (`WarnTiMan` or `WarnTiAut`) are lower than the `SampleTime` parameter.

Simulating signals

This block features the standard APL function "Simulating signals".

Selecting a unit of measure

This block features the standard APL function "Selecting a unit of measure" (`UA1unit`, `UA2unit`).

Idle state

This block features the standard APL function "Neutral position for motors, valves and controllers". The neutral position (de-energized state) is set using the SafePos parameter.

- `SafePos = 0`
  When `Ctrl = 0` the valve drive closes
  When `Ctrl = 1` the valve drive opens (de-energized state is "closed")
- `SafePos = 1`
  When `Ctrl = 0` the valve drive opens
  When `Ctrl = 1` the valve drive closes (de-energized state is "Open")

Generate instance-specific messages

This block features the standard APL function "Generate instance-specific messages".

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
2	See APL function: "Reset the commands for switching over the mode"
3	See APL function: "Activate resetting of commands for the control settings"
4	See APL function: "Set switch or button mode"
5	Characteristics of the "Valve seat lifting" function 0 = Either TOP or BOTTOM seat lifting possible 1 = TOP/BOTTOM seat lifting possible at the same time
6	Monitoring during "Valve seat lifting" 1 = Position monitoring switched off during seat lifting
9	See APL function: "Reset via input signals in the event of interlocks (protection) or errors"
10	See APL function: "Exit 'Local' mode"
17	See APL function: "Bumpless switchover to 'Automatic' mode for valves, motors, and dosers"
19	See APL function: "Changes signal status of outputs in OOS"

Bit	Function
20	See APL function: "Deactivation of calculation of the pulse control in LocalSetting 2 and 4"
21	See APL function: "Bumpless switchover to automatic mode for operator only"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"
27	See APL function: "Interlock display with LocalSetting 2 or 4"
30	See APL function: "Set the reset depending on the operating mode or the LiOp parameter"
31	See APL function: "Activate reset of protection/fault in manual mode"

Configurable reactions using the `Feature2` parameter

Bit	Function
2	See APL function: "Separate evaluation of excluded and simulated interlock signals"
3	See APL function: "Control priority in the event of an invalid input command"
4	See APL function: "Set switch or button mode for local commands"
5	See APL function: "Evaluation of the signal status of the interlock signals"
8	See APL function: "Force operating states in "Local" mode"
10	See APL function: "Consider bad quality of the automatic commands or the external values"

14.2.4 VlvDsL error handling

For information about error handling for all blocks, please also refer to the basic information in the "Error handling" section of the APL documentation.

The following errors can be displayed for this block:

- Error numbers
- Mode switchover error
- Invalid input signals

Overview of error numbers

The `ErrorNum` I/O can be used to output the following error numbers:

Error number	Meaning
-1	Predefined value when inserting the block; the block is not processed
0	There is no error.
41	The value for the LocalSetting I/O is not within the approved limit from 0 to 4.

Error number	Meaning

42	An error has occurred while `LocalLi = 1` and one of the following parameters is active:
	`LocalSetting = 0`
	`LocalSetting = 3`
	`LocalSetting = 4`

51	`AutModLi = 1` and `ManModLi = 1`
	`OpenLocal = 1` and `CloseLocal = 1`
	`OpenAut = 1` and `CloseAut = 1`
	`OpenForce = 1` and `CloseForce = 1`

52	An error has occurred in the following cases:
	`LocalAct = 1` and `LocalSetting = 2` and `SimOn = 1`
	`LocalAct = 1` and `LocalSetting = 4` and `SimOn = 1`

Mode switchover error

This error can be output by the block, see the "Error handling" section of the APL documentation.

Invalid input signals

This error can be output by the block, see the "Error handling" section of the APL documentation.

For the following invalid input signals, the control output can be stopped or switched to the neutral position. This depends on the APL function "Control priority for invalid input command".

Operating mode	Invalid input signals	Control reaction with Feature2 bit 3 =1
`Localsetting` = 1 or `Localsetting` = 3	Pushbutton operation for local mode (`Feature2` Bit 4 =0): `OpenLocal` =1 and `CloseLocal` =1 `SLBtmLocal` =1 and `SLTopLocal` =1	Valve is set to its neutral position.
Local: `Localsetting` = 1 or`Localsetting` = 3 and Force	`OpenForce` =1 and `CloseForce` =1 `SLBtmForce` =1 and `SLTopForce` =1	
Forcing and no "Local mode"	`OpenForce` =1 and `CloseForce` =1 `SLBtmForce` =1 and `SLTopForce` =1	
"Automatic mode" and no forcing	Pushbutton operation (`Feature` Bit 4 =0): `OpenAut` =1 and `CloseAut` =1 `SLBtmAut` =1 and `SLTopAut` =1	
"Manual mode" and no forcing	`OpenMan` =1 and `CloseMan` =1 `SLBtmMan` =1 and `SLTopMan` =1	

14.2.5 VlvDsL messaging

Message behavior

The following messages can be generated for this block:

- Control system fault
- Instance-specific messages

Two independent routes or batches can be active via this valve. The 'VlvDS' block has two message blocks for the double batch reference. The second message is only generated if the corresponding batch occupation signal is present (Occupied2).

Control system fault

Message instance	Message ID	Message class	Event
MsgEvId1 MsgEvId2	SIG 1	AS control system message – fault	$$BlockComment$$ Feedback error
	SIG 2	AS control system message – fault	$$BlockComment$$ External error occurred

$$BlockComment$$: content of the instance-specific comment

You can interconnect an external fault (signal) to the input parameter `CSF`. If this signal changes to `CSF = 1`, a control system fault is triggered (MsgEvId1, SIG 2).

Instance-specific messages

Message instance	Message ID	Message class	Event
MsgEvId1 MsgEvId2	SIG 3	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 4	AS process control message - fault	$$BlockComment$$ External message 2
	SIG 5	AS process control message - fault	$$BlockComment$$ External message 3

Associated values for message instances MsgEvId1 and MsgEvId2

Associated value	Block parameters
1	`BatchName / BatchName2`
2	`StepNo / StepNo2`
3	`BatchID / BatchID2`
4	ExtVa104
5	ExtVa105

Associated value	Block parameters
6	ExtVa106
7	ExtVa107
8	ExtVa108
9	Reserved
10	Reserved

The associated values 4 ... 8 are allocated to the parameters `ExtVa104 ... ExtVa108`, which you are free to use. For more information, refer to the manual titled "PCS 7 Process Control System - Engineering System".

14.2.6 VlvDsL I/Os

Input parameters

Parameter	Description	Type	Default
`AutModOp *`	1=Auto mode: auto mode by operator	BOOL	0
`BatchEn2`	Enable remote operation of controller by batch recipe (second batch)	BOOL	0
`BatchID *`	Current batch ID (number)	DWORD	16#00000000
`BatchID2 *`	Current batch ID (second batch)	DWORD	16#00000000
`BatchName *`	Current batch name	STRING[32]	''
`BatchName2 *`	Current batch name (second batch)	STRING[32]	''
`BypProt *`	Bypass protection in simulation / local modes	BOOL	0
`CloseAut`	1=Close: Close command in auto mode	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`CloseForce`	1=Close: forced close command	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`CloseLocal *`	1=Close local: close command	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`CloseMan *`	1=Close: Close command in manual mode	BOOL	0
`ClsAutDelay *`	Close command delay in auto mode	REAL	0.0
`CtrlChnST`	Output channel state of Ctrl	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`ExtVal04 ... ExtVal08`	External value 4 ... External value 8	ANY	

Parameter	Description	Type	Default
FbkClose	1=Close feedback	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
FbkOpen	1=Open feedback	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
FbkSLBtm	1=Bottom seat lifting feedback	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
FbkSLTop	1=Top seat lifting feedback	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Feature	I/O for additional functions (Page 716)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 0 • .. • 0
Feature2	I/O for additional functions (Page 716)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 0 • .. • 0
MonSafePos	1=On: monitor error, set valve to safe position	BOOL	1
NoFbkClose	1=Close feedback is not present	BOOL	0
NoFbkOpen	1=Open feedback is not present	BOOL	0
NoFbkSLBtm	1=Bottom seat lift feedback is not present	BOOL	1
NoFbkSLTop	1=Top seat lift feedback is not present	BOOL	1
Occupied2	Occupied by batch, enables second message (second batch)	BOOL	0
OpenAut	1=Open: Open command in auto mode	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
OpenForce	1=Open: forced open command	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
OpenLocal *	1=Open local: open command	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
OpenMan *	1=Open: Open command in manual mode	BOOL	0
OpnAutDelay *	Open command delay in auto mode	REAL	0.0

Parameter	Description	Type	Default
OS_Perm	I/O for operator permissions (Page 716)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 1 • .. • 1
SelFp1	Select faceplate 1	ANY	0
SelFp2	Select faceplate 2	ANY	0
SLBtmAut	1=Lift bottom seat in auto mode	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SLBtmForce	1=Force bottom seat lifting	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SLBtmLocal *	1=Lift bottom seat locally	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SLBtmMan *	1=Lift bottom seat in manual mode	BOOL	0
SLBtmPause *	Pause duration for bottom seat lifting when in internal pulse/pause mode [s]	REAL	180.0
SLBtmPulse *	Pulse duration for bottom seat lifting when in internal pulse/pause mode [s]	REAL	10.0
SLPuPaMode	1=External pulse/pause mode, 0=Internal	BOOL	Internal
SLTopAut	1=Lift top seat in auto mode	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SLTopForce	1=Force top seat lifting	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SLTopLocal *	1=Lift top seat locally	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SLTopMan *	1=Lift top seat in manual mode	BOOL	0
SLTopPause *	Pause duration for top seat lifting when in internal pulse/pause mode [s]	REAL	180.0
SLTopPulse *	Pulse duration for top seat lifting when in internal pulse/pause mode [s]	REAL	10.0
StepNo2 *	Batch step number (second batch)	DWORD	16#00000000

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
Ctrl	Control output (dependent on SafePos)	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
FbkCloseOut	1 = Valve is closed	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
FbkOpenOut	1=Valve is opened	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
FbkSLBtmOut	1 = Bottom seat is being lifted	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
FbkSLTopOut	1 = Top seat is being lifted	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
P_Close	Pulsive close control (independent of SafePos)	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
P_Open	Pulsive open control (independent of SafePos)	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SLBtm	Control output for bottom seat lifting	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SLTop	Control output for top seat lifting	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

14.2.7 VlvDsL block diagram

The block has no block diagram.

14.2.8 Operator control and monitoring

14.2.8.1 VlvDsL views

VlvDsL block views

The VlvDsL block provides the following views:

- VlvDsL standard view (Page 729)
- Message view
- Parameter view (Standard - parameter view for motors and valves in Advanced Process Library. In addition, delay times for opening and closing the valves can be operated (only in "Manual" mode) and monitored.
- VlvDsL seat lifting view (Page 732)
- Trend view
- VlvDsL preview (Page 734)
- Memo view
- Batch view 1
- Batch view 2
- VlvDsL block icons (Page 736)

The block provides two batch views:

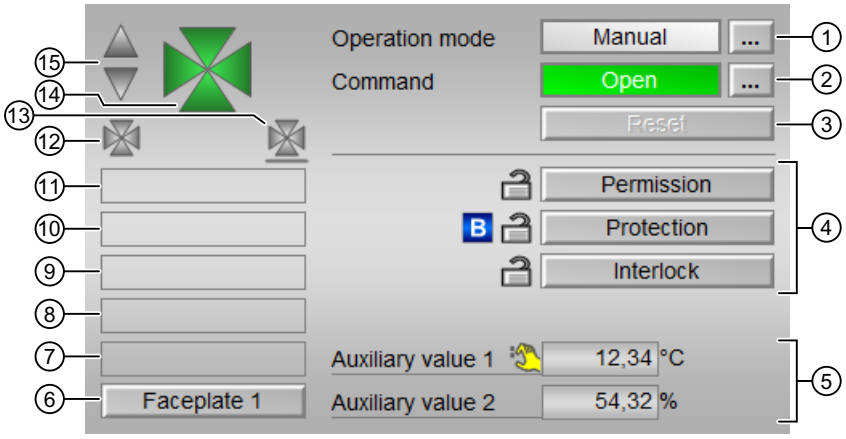
- The standard batch view
- A special block batch view for double batch references

Both views have the same behavior. Furthermore, the block issues a unique operator message for both batches.

You can find general information on the faceplate and block icon in the "Structure of the faceplate" and "Block icon structure" sections of the APL function manual.

14.2.8.2 VlvDsL standard view

VlvDsL standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- Manual mode
- Automatic mode
- Local mode
- Out of service

You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(2) Open and close the valve

This area shows you the operating state set for the valve. The following states can be shown and executed here:

- "Open"
- "Close"

You can learn about the switchover of the state in the section "Switchover of operating state and operating mode" in the APL manual.

If texts are configured for these commands, they are displayed as state text and as button labels during command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

(3) Reset the block

Click "Reset" in the event of interlocks or errors. You can find more information on this in the section "Resetting the block in case of interlocks or errors" in the manual of the APL.

(4) Operating range for the interlock functions of the block

This display is only visible when the corresponding block input is interconnected.

You can use this button to control the interlock functions of the block. You can find more information on this in the section "Interlock functions" in the manual of the APL.

The following is displayed in addition to the buttons:

- Interlock status (see section "Group status for interlock information" in the manual of the APL), e.g.:



- Signal status (see section "Forming and outputting the signal status for technological blocks" in the APL manual)



- Bypass information (see section "Forming the group status for interlock information" in the manual of the APL):



(5) Display of auxiliary values

This display is only visible when the corresponding block input is interconnected.

You can use this area to display two auxiliary values that have been configured in the engineering system (ES). You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(6) Navigation button for switching to the standard view of any faceplate

This display is only visible when the corresponding block input is interconnected.

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(7) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

You can find additional information on this in the section "Release for maintenance" in the manual of the APL.

(8) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

You can find additional information on this in section "Simulating signals" in the manual of the APL.

(9) Display area for block states

This area provides additional information on the operating state of the block:

- "End position error"
- "Control error"
- "Undefined position"
- "Invalid signal"
- "Switchover error"
- "External error"

You can find more information on this in the sections "Monitoring the feedbacks" and "Error handling" (subsections "Invalid input signals" as well as "Mode switchover error" and "Motor protection function") in the manual of the APL

(10) Display area for block states

This area provides additional information on the operating state of the block:

- "Forced up"
- "Forced down"

(11) Display area for block states

This area provides additional information on the operating state of the block:

- "Forced open"
- "Forced close"

(12) Automatic preview

This display is only visible in "Manual mode", in "Local mode", or with a reset request in "Automatic mode", when the current output signals are not identical to the control in "Automatic mode".

The display shows what state the valve would assume if you switched from "manual" or "local" mode to "automatic mode", or performed a reset to "automatic mode".

(13) Idle state of the valve

The neutral position of the valve is shown here.

If the idle state of the valve is "Closed" (SafePos = 0), a gray valve is shown.

If the idle state of the valve is "Open" (SafePos = 0), a green valve is shown.

(14) Status display of the valve

This area shows if the valve is open or closed:

- Green: Valve is open
- Gray: Valve is closed
- Red: Fault at valve

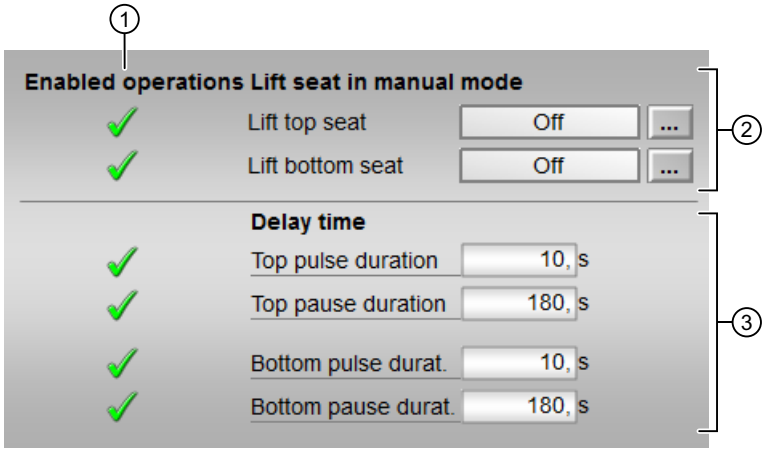
(15) Status display of the seat lifting

This area shows whether the seat lifting (top or bottom) is active or inactive:

- Green: Seat lifting active
- Gray: Seat lifting inactive

14.2.8.3 VlvDsL seat lifting view

VlvDsL seat lifting view



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

(2) Seatlifting in manual mode

In this area, you can change parameters and thereby influence the valve.

You can influence the following parameters:

- "Lift top seat": Valve is opened in manual operation in top seat lifting
- "Lift bottom seat": Valve is opened in manual operation in bottom seat lifting

(2) Delay

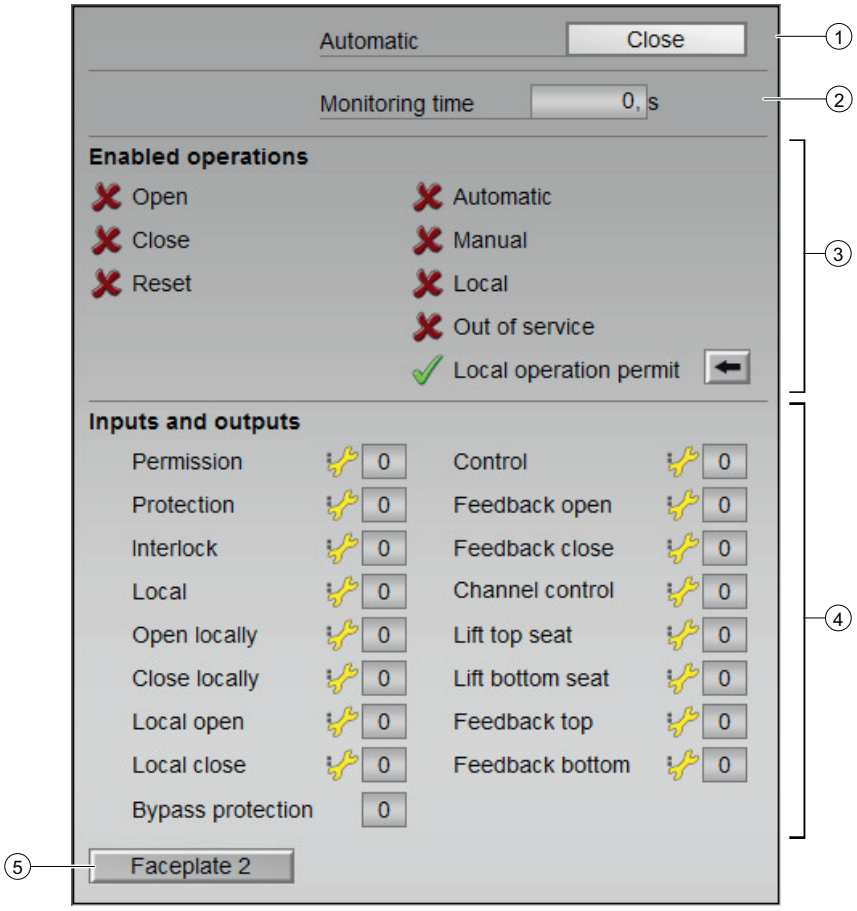
In this area, you can change parameters and thereby influence the valve.

You can influence the following parameters:

- "Top pulse duration": Pulse duration for top control
- "Top pause duration": Pause duration for top control
- "Bottom pulse duration": Pulse duration for bottom control
- "Bottom pulse duration": Pause duration for bottom control

14.2.8.4 VlvDsL preview

VlvDsL preview



(1) Automatic preview

This area shows you the block status after it has switched from "Manual" or "Local" mode to "Automatic" mode.

If the block is in automatic mode, the current block state is displayed.

This area also shows the worst signal status of the following automatic commands:

- `OpenAut`
- `CloseAut`

(2) Preview area

- "Monitoring time": Display of the current monitoring time

This display is only visible when monitoring is activated.

(3) Operator control enable

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: the OS operator cannot control this parameter due to the configured AS operator permissions (OS_Perm or OS1Perm).

The following enabled operations are shown here:

- "Open": You can open the value.
  If text is configured for this command, it is also displayed in brackets. For more information, refer to the section "Labeling of buttons and text".
- "Close": You can close the value.
  If text is configured for this command, it is also displayed in brackets. For more information, refer to the section "Labeling of buttons and text".
- "Reset": You can reset the valve if interlocks or errors occur.
- "Automatic": You can switch to "automatic mode".
- "Manual": You can switch to "manual mode".
- "Local": You can switch to "local mode".
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the <-- button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(4) Display current control signals

This area shows the most important parameters for this block with the current control:

- "Permission": This display is only visible when the corresponding block input is interconnected.
  - 0 = Valve activation not enabled on OS
  - 1 = Enable for "opening" / "closing" from the neutral position
- "Protection": This display is only visible when the corresponding block input is interconnected.
  - 0 = Protective interlocking in effect; once the interlocking condition has cleared, you have to reset the block
  - 1 = Good state

- "Interlock": This display is only visible when the corresponding block input is interconnected.
  - 0 = Interlocking without reset in effect; you can operate the block without reset once the interlocking condition has cleared
  - 1 = Good state
- "Local": 1 = Control signal for "Local mode" (LocalLi) is enabled
- "Local open": 1 = Open valve in "Local mode"
- "Local close": 1 = Close valve in "Local mode"
- "Local mode up": 1 = Raise valve in "Local mode"
- "Local mode down": 1 = Lower valve in "Local mode"
- "Bypass protection":
  - 0 = Disable exclusion
  - 1 = Exclusion of interlock in "Local mode" and in simulation
- "Control": Display for valve control:
  - 0 = Valve will be closed
  - 1 = Valve will be open
- "Feedback open": 1 = Valve is open
- "Feedback close": 1 = Valve is closed
- "Channel Control": Control signal of the output channel block
- "Top seat lifting": Display for valve control
- "Bottom seat lifting": Display for valve control
- "Feedback top": 1 = Valve is at top
- "Feedback bottom": 1 = Valve is at bottom

(5) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

14.2.8.5 VlvDsL block icons

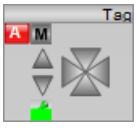
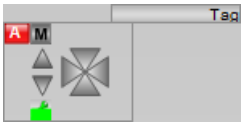
Properties of the VlvDsL block icon

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of alarm, warning, and tolerance limits as well as the control system fault
- Operating modes

- Signal status, release for maintenance
- Forcing states
- Display for bypassing interlocks
- Interlocks
- Memo display
- Valve status display

The block icons from template @TemplateILV8.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	
	3	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

Valve status display

The following valve states are shown here:

Icon	Meaning
	Valve open
	Error at valve
	Valve is opening
	Valve closed
	Valve is closing

Panel blocks 15

15.1 PCalcWatP - Panel communication block for CalcWatP

15.1.1 PCalcWatP description

Object name (type + number) and family

Type + number: FB 1146

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PCalcWatP" must be connected to any given "CalcWatP" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PCalcWatP" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block "CalcWatP", the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

RESTART = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC

SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (OB32). Also in OB100.

Status word assignment for the parameters OSStatAI and OSStatAI_Out

For a description of the individual parameters, refer to the section PCalcWatP I/Os (Page 741).

Status bit	Parameter
0	VolFlowCSF
1	TempForCSF
2	TempRetCSF
3 - 6	Not used
7	SimAct
8 - 9	Not used
10	ExtMsg1
11	ExtMsg2
12	ExtMsg3
13	ExtMsg4
14 - 31	Not used

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PCalcWatP I/Os (Page 741).

Status bit	Parameter
0 - 2	Not used
3	PanelAct
4 - 5	Not used
6	SimAct
7	Error
8 - 21	Not used
22	OosAct
23 - 31	Not used

15.1.2 PCalcWatP operating modes

The block does not have any operating modes.

15.1.3 PCalcWatP functions

The "PCalcWatP" block acts as the interface between the "CalcWatP" technological block from the Industry Library (IL) and the operator panel.

"PCalcWatP" is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.1.4 PCalcWatP error handling

The block does not have any error handling.

15.1.5 PCalcWatP messaging

Message behavior

The messages correspond to those of the "CalcWatP" block from the Industry Library (IL).

15.1.6 PCalcWatP I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV8	Auxiliary value message value 1 ... 8	STRUCT • Value:REAL • ST: BYTE	- • 0.0 • 16#80
BlockConnector	Connection to CalcWatP block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01 ... MsgEvId08	Message ID 1 ... 8	DWORD	16#60000001 ... 16#60000008
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_Energy	Status thermal energy 1	REAL	0.0
Op_Energy2	Status thermal energy 2	REAL	0.0

Parameter	Description	Data type	Default
Op_Heating	Status 1=Heating mode	BOOL	0
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Power	Status thermal power	REAL	0.0
Op_TempFor	Status forerun temperature	REAL	0.0
Op_TempRet	Status return temperature	REAL	0.0
Op_Visibility	Objects visibility	DWORD	16#00000000
Op_VolFlowRat	Status volume flow rate	REAL	0.0
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch permission	DWORD	16#00000001

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
GrErr	Status: 1=Error	STRUCT	
GrErr.ST	Signal status	BYTE	16#80
GrErr.Value	Value	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAck01 ... MsgAck08	Message: ACK_STATE Output 1 ... 8	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat08	Message: STATUS Output 1 ... 8	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.1.7 PCalcWatP block diagram

The block has no block diagram.

15.1.8 Operator control and monitoring

15.1.8.1 PCalcWatP WinCC views

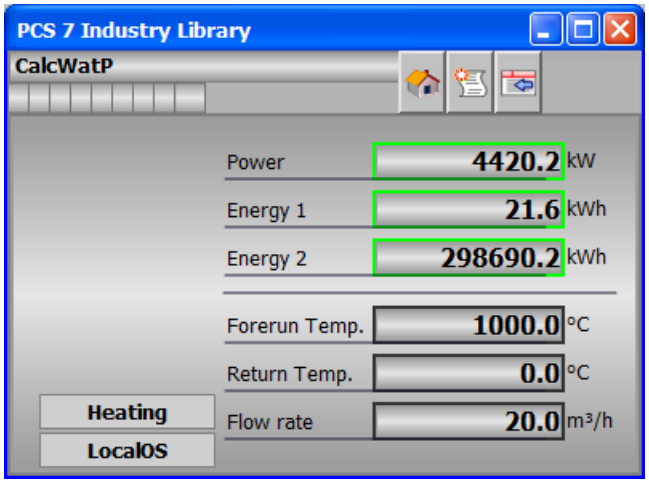
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54).

15.1.8.2 PCalcWatP WinCC Comfort views

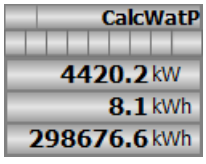
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.1.8.3 PCalcWatP WinCC Comfort



15.1.8.4 PCalcWatP WinCC Comfort block icons



15.2 PHxFct - Panel communication block for HxFct

15.2.1 PHxFct description

Object name (type + number) and family

Type + number: FB 1149

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PHxFct" must be connected to any given "HxFct" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PHxFct" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block HxFct, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

RESTART = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19_	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (OB32). Also in OB100.

Status word assignment for the parameters OSStatAl and OSStatAl_Out

For a description of the individual parameters, refer to the section PHxFct I/Os (Page 746).

Status bit	Parameter
0	RelHumCSF
1	TempCSF
2 - 6	Not used
7	SimAct
8 - 9	Not used
10	ExtMsg1
11	ExtMsg2
12	ExtMsg3
13	ExtMsg4
14	ExtMsg5
15 - 31	Not used

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PHxFct I/Os (Page 746).

Status bit	Parameter
0 - 2	Not used
3	PanelAct
4 - 5	Not used
6	SimAct
7	Error
8 - 21	Not used
22	OosAct
23 - 31	Not used

15.2.2 PHxFct operating modes

The block does not have any operating modes.

15.2.3 PHxFct functions

The "PHxFct" block acts as the interface between the "HxFct" technological block from the Industry Library (IL) and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.2.4 PHxFct error handling

The block does not have any error handling.

15.2.5 PHxFct messaging

Message behavior

The messages correspond to those of the "HxFct" block from the Industry Library (IL).

15.2.6 PHxFct I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV8	Auxiliary value message value 1 ... 8	STRUCT • Value:REAL • ST: BYTE	- • 0.0 • 16#80
BlockConnector	Connection to HxFct block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01	Message ID 1	DWORD	16#60000009
MsgEvId02... MsgEvId07	Message ID 2 ... 7	DWORD	16#6000000A ... 16#6000009F
MsgEvId08	Message ID 8	DWORD	16#60000010
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_AbsHum	Status absolute humidity	REAL	0.0
Op_Enthalpy	Status enthalpy	REAL	0.0
Op_Mode	Operator mode	WORD	16#0000

Parameter	Description	Data type	Default
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_RelHum	Relative humidity status	REAL	0.0
Op_SatHum	Humidity at saturation status	REAL	0.0
Op_Temp	Status temperature	REAL	0.0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch permission	DWORD	

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
GrErr	Status: 1=Error	STRUCT	
GrErr.ST	Signal status	BYTE	16#80
GrErr.Value	Value	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAck01 … MsgAck08	Message: ACK_STATE Output 1 ... 8	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 … MsgStat08	Message: STATUS Output 1 ... 8	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.2.7 PHxFct block diagram

The block has no block diagram.

15.2.8 Operator control and monitoring

15.2.8.1 PHxFct WinCC views

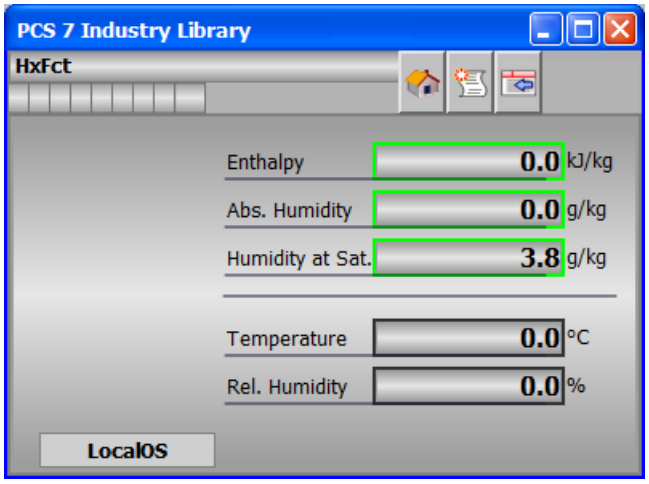
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54).

15.2.8.2 PHxFct WinCC Comfort views

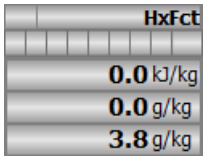
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.2.8.3 PHxFct WinCC Comfort standard view



15.2.8.4 PHxFct WinCC Comfort block icons



15.3 PMotL - Panel communication block for MotL

15.3.1 PMotL description

Object name (type + number) and family

Type + number: FB 1107

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PMotL" must be connected to any given "MotL" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PMotL" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block MotL, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature.Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAI and OSStatAI_Out

For a description of the individual parameters, refer to the section PMotL I/Os (Page 751).

Status bit	Parameter
Bit0	External error has occurred (`CSF`)
Bit1	Motor feedback error
Bit2	-
Bit3	Motor protection triggered
Bit4	-
Bit5	External message 1
Bit6	External message 2
Bit7	External message 3
Bit18 ... Bit31	-

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PMotL I/Os (Page 751).

Status bit	Parameter
Bit0	`LocalAct`.Value
Bit1	`ManAct`.Value
Bit2	-
Bit3	Panel active
Bit4	`AutoAct`.Value
Bit5	`SimOn`.Value
Bit6	-
Bit7	Motor error
Bit8	Motor is running
Bit9	Motor is stopped
Bit10	-
Bit11	-
Bit12	Motor starting
Bit13	Motor is stopped
Bit14	`StartForce`.Value
Bit15	`StopForce`.Value
Bit16	-
Bit17	-
Bit18	Show fault profile
Bit19 ... Bit21	-

Status bit	Parameter
Bit22	OosAct.Value
Bit23 ... Bit31	-

15.3.2 PMotL operating modes

The block does not have any operating modes.

15.3.3 PMotL functions

The "PMotL" block acts as the interface between the "MotL" technological block from the APL and the operator panel. It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.3.4 PMotL error handling

The block does not have any error handling.

15.3.5 PMotL messaging

Message behavior

The messages correspond to those of the "MotL" block from the APL.

15.3.6 PMotL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV6	Auxiliary value message value 1 ... 13	STRUCT • Value:REAL • ST: BYTE	- • 0.0 • 16#80
BlockConnector	Connection to MotL block		
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1

Parameter	Description	Data type	Default
EnAux	Enable measured value	BOOL	0
MsgEvI01	Message ID_1	DWORD	16#6000000F
MsgEvId02 ... MsgEvId06	Message ID_2 ... ID_6	DWORD	16#60000010 ... 16#60000014
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Reset	Operator input error reset	BOOL	0
Op_StartMan	Operator input manual start	BOOL	0
Op_StopMan	Operator input manual stop	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit Reporting Procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch permission	DWORD	16#00000001

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn06	Message: ACK_STATE Output 1 ... Output 6	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat06	Message: STATUS Output 1 ... MsgStat06	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000

Parameter	Description	Data type	Default
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.3.7 PMotL block diagram

The block has no block diagram.

15.3.8 Operator control and monitoring

15.3.8.1 PMotL WinCC views

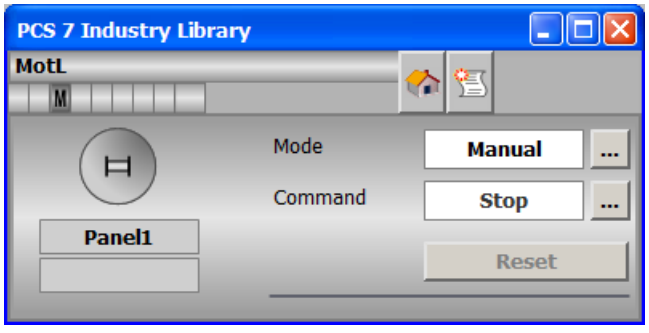
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.3.8.2 PMotL WinCC Comfort views

The block provides the following views:

- Standard view
- Message view
- Associated value view

15.3.8.3 PMotL WinCC Comfort standard view



15.3.8.4 PMotL WinCC Comfort block icons



15.4 PMotRevL - Panel communication block for MotRevL

15.4.1 PMotRevL description

Object name (type + number) and family

Type + number: FB 1108

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PMotRevL" must be connected to any given "MotRevL" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PMotRevL" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block MotRevL, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAI and OSStatAI_Out

For a description of the individual parameters, refer to the section PMotRevL I/Os (Page 757).

Status bit	Parameter
Bit0	External error has occurred (CSF)
Bit1	Motor feedback error
Bit2	-
Bit3	Motor protection triggered
Bit4	-
Bit5 - 7	External message 1 ... 3
Bit8 ... Bit15	-

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PMotRevL I/Os (Page 757).

Status bit	Parameter
Bit0	LocalAct.Value
Bit1	ManAct.Value
Bit2	-
Bit3	Panel active
Bit4	AutoAct.Value
Bit5	SimOn.Value
Bit6	-
Bit7	Motor error
Bit8	Motor running in "Forward" mode
Bit9	Motor running in "Reverse" mode
Bit10	Motor is stopped
Bit11	Motor starting in "Forward" mode
Bit12	Motor starting in "Reverse" mode
Bit13	Motor is stopped
Bit14 ... Bit17	-
Bit18	Show fault profile
Bit19	FwdForce.Value
Bit20	RevForce.Value
Bit21	StopForce.Value
Bit22	OosAct.Value
Bit23 ... Bit31	-

15.4.2 PMotRevL operating modes

The block does not have any operating modes.

15.4.3 PMotRevL functions

The "PMotL" block acts as the interface between the "MotL" technological block from the APL and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.4.4 PMotRevL error handling

The block does not have any error handling.

15.4.5 PMotRevL messaging

Message behavior

The messages correspond to those of the "MotRevL" block from the APL.

15.4.6 PMotRevL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 … Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 … AV6	Auxiliary value message value 1	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to MotRevL block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01 … MsgEvId06	Message ID_1 ... ID_6	DWORD	16#60000001 ... 16#60000006

Parameter	Description	Data type	Default
Op_FwdMan	Status operator panel 1 = Start: Forward start command in manual mode	BOOL	0
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Reset	Operator input error reset	BOOL	0
Op_RevMan	Status operator panel 1 = Start: Reverse start command in manual mode	BOOL	0
Op_StopMan	Status operator panel 1 = Stop: Stop command in manual mode	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit Reporting Procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm *	Switch permission	DWORD	

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn06	Message: ACK_STATE Output 1 ... Output 6	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat06	Message: STATUS Output 1 ... MsgStat06	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	LL

15.4.7 PMotRevL block diagram

The block has no block diagram.

15.4.8 Operator control and monitoring

15.4.8.1 PMotRevL WinCC views

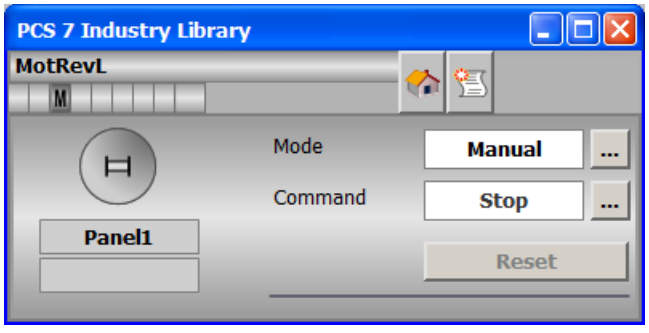
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54).

15.4.8.2 PMotRevL WinCC Comfort views

The block provides the following views:

- Standard view
- Message view
- Associated value view

15.4.8.3 PMotRevL WinCC Comfort standard view



15.4.8.4 PMotRevL WinCC Comfort block icons



15.5 PMotSpdCL - Panel communication block for MotSpdCL

15.5.1 PMotSpdCL description

Object name (type + number) and family

Type + number: FB 1110

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PMotSpdCL" must be connected to any given "MotSpdCL" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PMotSpdCL" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block MotSpdCL, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAI and OSStatAI_Out

For a description of the individual parameters, refer to the section PMotSpdCL I/Os (Page 762).

Status bit	Parameter
Bit0	External error has occurred (CSF)
Bit1	Motor feedback error
Bit2	Not used
Bit3	Motor protection triggered
Bit4	Rbk – High warning limit violated
Bit5	Rbk – Low warning limit violated
Bit6 ... Bit8	External message 1 ... 3
Bit9 - 31	Not used

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PMotSpdCL I/Os (Page 762).

Status bit	Parameter
Bit0	LocalAct.Value
Bit1	ManAct.Value
Bit2	Not used
Bit3	Panel Active
Bit4	AutoAct.Value
Bit5	SimOn.Value
Bit6	Not used
Bit7	Motor error
Bit8	Motor running in "Forward" mode
Bit9	Motor running in "Reverse" mode
Bit10	Motor is stopped
Bit11	Motor starting in "Forward" mode
Bit12	Motor starting in "Reverse" mode
Bit13	Motor is stopped
Bit14 - 17	Not used
Bit18	Show fault profile
Bit19	FwdForce.Value
Bit20	RevForce.Value
Bit21	StopForce.Value
Bit22	OosAct.Value
Bit23	SP_Ext
Bit24 - 31	Not used

15.5.2 PMotSpdCL operating modes

The block does not have any operating modes.

15.5.3 PMotSpdCL functions

The "PMotSpdCL" block acts as the interface between the "MotSpdCL" technological block from the APL and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.5.4 PMotSpdCL error handling

Error handling

The block does not have any error handling.

15.5.5 PMotSpdCL messaging

Message behavior

The messages correspond to those of the "MotSpdCL" block from the APL.

15.5.6 PMotSpdCL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV8	Auxiliary value message value 1 ... 8	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to MotSpdCL block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0

Parameter	Description	Data type	Default
MsgEvId01 ... MsgEvId03	Message ID_1 ... 3	DWORD	16#60000007 ... 16#60000009
MsgEvId04 ... MsgEvId08	Message ID_4 ... 8	DWORD	16#6000000A ... 16#6000000E
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_FwdMan	Status operator panel 1 = Start: Forward start command in manual mode	BOOL	0
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Reset	Operator input error reset	BOOL	0
Op_RevMan	Status operator panel 1 = Start: Reverse start command in manual mode	BOOL	0
Op_SP	Status operator panel external setpoint	REAL	0.0
Op_SP_HL	Operator input external setpoint high limit	REAL	0.0
Op_SP_LL	Operator input external setpoint low limit	REAL	0.0
Op_StopMan	Status operator panel 1 = Stop: Stop command in manual mode	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
SwitchPerm *	Switch permission	DWORD	16#00000001

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn08	Message: ACK_STATE Output 1 ... 8	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat08	Message: STATUS Output 1 ... 8	WORD	16#0000

Parameter	Description	Data type	Default
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.5.7 PMotSpdCL block diagram

The block has no block diagram.

15.5.8 Operator control and monitoring

15.5.8.1 PMotSpdCL WinCC views

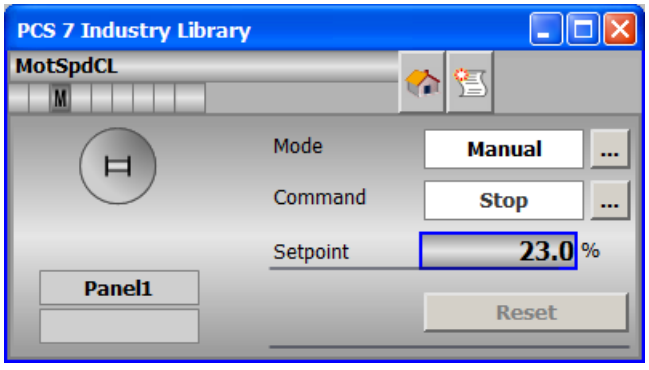
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.5.8.2 PMotSpdCL WinCC Comfort views

The block provides the following views:

- Standard view
- Message view
- Associated value view

15.5.8.3 PMotSpdCL WinCC Comfort standard view



15.5.8.4 PMotSpdCL WinCC Comfort block icons



15.6 PMotSpdL - Panel communication block for MotSpdL

15.6.1 PMotSpdL description

Object name (type + number) and family

Type + number: FB 1109

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PMotSpdL" must be connected to any given "MotSpdL" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PMotSpdL" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block MotSpdL, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word allocation for OSStatAl and OSStatAl_Out parameters

For a description of the individual parameters, refer to the section PMotSpdL I/Os (Page 768).

Status bit	Parameter
Bit0	External error has occurred (CSF)
Bit1	Motor feedback error
Bit2	Not used
Bit3	Motor protection triggered
Bit4	Not used
Bit5	External message 1
Bit6	External message 2
Bit7	External message 3
Bit8 - 31	Not used

Status word allocation for OSStat and OSStat_Out parameters

For a description of the individual parameters, refer to the section PMotSpdL I/Os (Page 768).

Status bit	Parameter
Bit0	LocalAct.Value
Bit1	ManAct.Value
Bit2	Not used
Bit3	Panel active
Bit4	AutAct active
Bit5	SimOn.Value
Bit6	Not used
Bit7	Motor error
Bit8	Motor running at speed 1
Bit9	Motor running at speed 2
Bit10	Motor is stopped
Bit11	Motor starting with speed 1
Bit12	Motor starting with speed 2
Bit13	Motor is stopped
Bit14 - 17	Not used
Bit18	Show fault profile
Bit19	Spd1Force.Value
Bit20	Spd2Force.Value
Bit21	StopForce.Value
Bit22	OosAct.Value
Bit23 - 31	Not used

15.6.2 PMotSpdL operating modes

The block does not have any operating modes.

15.6.3 PMotSpdL functions

The "PMotSpdL" block acts as the interface between the "MotSpdL" technological block from the APL and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.6.4 PMotSpdL error handling

The block does not have any error handling.

15.6.5 PMotSpdL messaging

Message behavior

The messages correspond to those of the "MotSpdL" block from the APL.

15.6.6 PMotSpdL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 … Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 – AV6	Auxiliary value message value 1 -6	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to MotSpdL block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01 … MsgEvId06	Message ID_1 ... ID_6	DWORD	16#60000001 ... 16#60000006

Parameter	Description	Data type	Default
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Reset	Operator input error reset	BOOL	0
Op_Spd1Man	Status operator panel 1 = Start: Speed1 start command in manual mode	BOOL	0
Op_Spd2Man	Status operator panel 1 = Start: Speed2 start command in manual mode	BOOL	0
Op_StopMan	Status operator panel 1 = Stop: Stop command in manual mode	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm *	Switch permission	DWORD	16#00000001

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn06	Message: ACK_STATE Output 1 ... 6	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat06	Message: STATUS Output 1 ... 6	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000

Parameter	Description	Data type	Default
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.6.7 PMotSpdL block diagram

The block has no block diagram.

15.6.8 Operator control and monitoring

15.6.8.1 PMotSpdL WinCC views

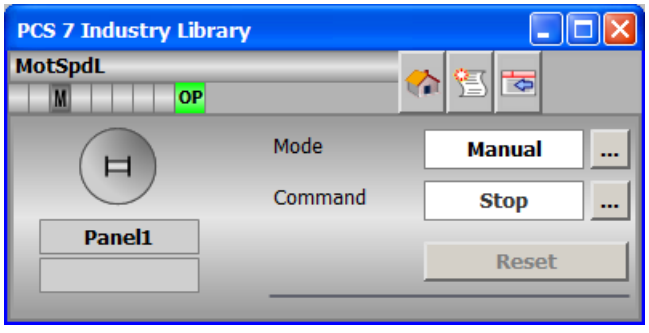
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.6.8.2 PMotSpdLWinCC Comfort views

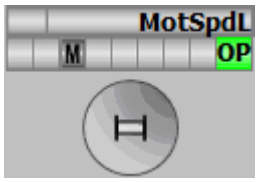
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.6.8.3 PMotSpdLWinCC Comfort standard view



15.6.8.4 PMotSpdLWinCC Comfort block icons



15.7 PMonAnL - Panel communication block for MonAnL

15.7.1 PMonAnL description

Object name (type + number) and family

Type + number: FB 1120

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PMonAnL" must be connected to any given "MonAnL" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PMonAnL" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block MonAnL, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAI and OSStatAI_Out

For a description of the individual parameters, refer to the section PMonAnL I/Os (Page 774).

Status bit	Parameter
Bit0	External error has occurred (CSF)
Bit1	PV - High alarm limit violated
Bit2	PV - Low alarm limit violated
Bit3	PV - High tolerance limit violated
Bit4	PV - Low tolerance limit violated
Bit5	PV - High warning limit violated
Bit6	PV - Low warning limit violated
Bit7	Limit (high) positive gradient
Bit8	Limit (low) negative gradient
Bit9	Limit (low) absolute gradient
Bit10	External message 1
Bit11	External message 2
Bit12 - 31	Not used

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PMonAnL I/Os (Page 774).

Status bit	Parameter
Bit0 - 2	Not used
Bit3	Panel active
Bit4 - 5	Not used
Bit6	SimAct
Bit7	Fault
Bit8 - 21	Not used
Bit22	OosAct
Bit 23 - 31	Not used

15.7.2 PMonAnL operating modes

The block does not have any operating modes.

15.7.3 PMonAnL functions

The "PMonAnL" block acts as the interface between the "MonAnL" technological block from the APL and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.7.4 PMonAnL error handling

The block does not have any error handling.

15.7.5 PMonAnL messaging

Message behavior

The messages correspond to those of the "MonAnL" block from the APL.

15.7.6 PMonAnL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV13	Auxiliary value message value 1 ... 13	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to MonAnL block	ANY	
CmpID	Area code	DWORD	16#00000001
EnAux	Enable measured value	BOOL	0
MsgEvId01	Message ID_1	DWORD	16#60000009
MsgEvId02 ... MsgEvId07	Message ID_2 ... 7	DWORD	16#6000000A ... 16#6000000F
MsgEvId08 ... MsgEvId13	Message ID_2 ... 7	DWORD	16#60000010 ... 16#60000015
Op_Mode	RESERVED Not used yet.	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_PV	Feedback analog input	REAL	0.0

Parameter	Description	Data type	Default
Op_PV_AH_Lim	Reserved	REAL	95.0
Op_PV_AL_Lim	Reserved	REAL	5.0
Op_PV_Grad	Reserved	REAL	0.0
Op_PV_WH_Lim	Reserved	REAL	90.0
Op_PV_WL_Lim	Reserved	REAL	10.0
Op_ScaleHigh	PV - Bar display limits for OS (High)	REAL	100.0
Op_ScaleLow	PV - Bar display limits for OS (Low)	REAL	0.0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUpCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch permission	DWORD	

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	Enable measured value	BOOL	0
GrErr	Status: 1=Error	STRUCT • Value:BOOL • ST: BYTE	 • 0 • 16#80
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn13	Message: ACK_STATE Output 1 ... 13	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat13	Message: STATUS Output 1 ... 13	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	LL

15.7.7 PMonAnL block diagram

The block has no block diagram.

15.7.8 Operator control and monitoring

15.7.8.1 PMonAnL WinCC views

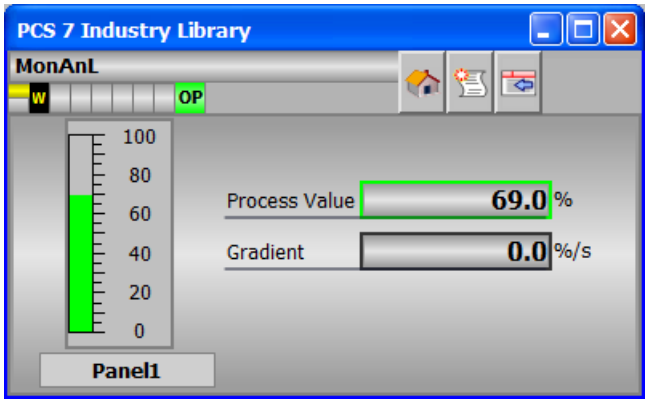
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.7.8.2 PMonAnL WinCC Comfort views

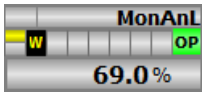
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.7.8.3 PMonAnL WinCC Comfort standard view



15.7.8.4 PMonAnL WinCC Comfort block icons



15.8 PMonAn08 - Panel communication block for MonAn08

15.8.1 PMonAn08 description

Object name (type + number) and family

Type + number: FB 1118

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PMonAn08" must be connected to any given "MonAn08" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock =0, "PMonAn08" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block MonAn08, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAl and OSStatAl_Out

For a description of the individual parameters, refer to the section PMonAn08 I/Os (Page 779).

Status bit	Parameter
Bit0	External error has occurred (CSF)
Bit1 - Bit 8	Limit 1 – 8
Bit9	Simulation active
Bit10 - 15	External message 1 - 6
Bit16 - Bit31	-

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PMonAn08 I/Os (Page 779).

Status bit	Parameter
Bit0 - Bit2	-
Bit3	Panel active
Bit3 - Bit4	-
Bit6	SimAct
Bit7	Fault
Bit7 - Bit21	-
Bit22	OosAct
Bit23 - Bit31	-

15.8.2 PMonAn08 operating modes

The block does not have any operating modes.

15.8.3 PMonAn08 functions

The "PMonAn08" block acts as the interface between the "MonAn08" technological block from the Industry Library (IL) and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.8.4 PMonAn08 error handling

The block does not have any error handling.

15.8.5 PMonAn08 messaging

Message behavior

The messages correspond to those of the "MonAn08" block from the Industry Library (IL).

15.8.6 PMonAn08 I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... value4	REAL	0.0
AV1 ... AV16	Auxiliary value message value 1 ... 16	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to MonAn08 block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01 ... MsgEvId04	Message ID_1 ... ID_4	DWORD	16#60000016 ... 16#60000019
MsgEvId05 ... MsgEvId10	Message ID_5 ... ID_10	DWORD	16#6000001A ... 16#6000001F
MsgEvId11 ... MsgEvId16	Message ID_11 ... ID_16	DWORD	16#60000020 ... 16#60000025
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_PV	Status process value	REAL	0.0
Op_PV_Lim1 ... Op_PV_Lim8	Reserved	REAL	0.0
Op_ScaleHigh	PV - Bar display limits for OS (High)	REAL	100.0

Parameter	Description	Data type	Default
Op_ScaleLow	PV - Bar display limits for OS (Low)	REAL	0.0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch permission	DWORD	

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
GrErr	Status: 1=Error	STRUCT	
GrErr.ST	Signal status	BYTE	16#80
GrErr.Value	Value	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn16	Message: ACK_STATE Output 1 ... 16	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat16	Message: STATUS Output 1 ... 16	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	LL

15.8.7 PMonAn08 block diagram

The block has no block diagram.

15.8.8 Operator control and monitoring

15.8.8.1 PMonAn08 WinCC views

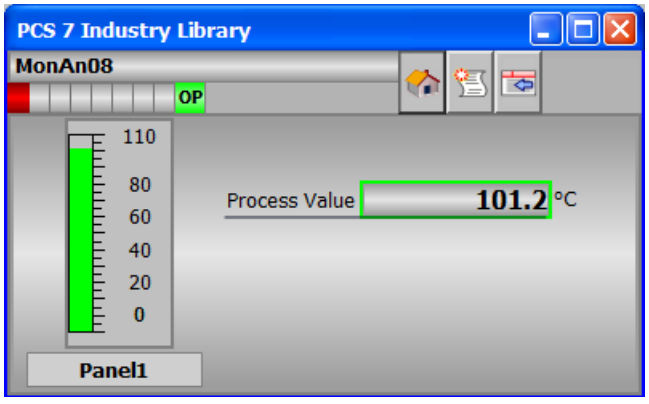
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.8.8.2 PMonAn08 WinCC Comfort views

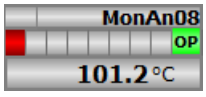
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.8.8.3 PMonAn08 WinCC Comfort standard view



15.8.8.4 PMonAn08 WinCC Comfort block icons



15.9 PMonAnDi - Panel communication block for MonAnDi

15.9.1 PMonAnDi description

Object name (type + number) and family

Type + number: FB 1119

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PMonAnDi" must be connected to any given "MonAnDi" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PMonAnDi" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block MonAnDi, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAI and OSStatAI_Out

For a description of the individual parameters, refer to the section PMonAnDi I/Os (Page 784).

Status bit	Description
Bit0	External error has occurred (CSF)
Bit1	PV - High alarm limit violated
Bit2	PV - Low alarm limit violated
Bit3 - Bit4	-
Bit5	PV - High warning limit violated
Bit6	PV - Low warning limit violated
Bit7	Simulation active
Bit8 - Bit9	-
Bit10	External message 1
Bit11	External message 2
Bit12	External message 3
Bit13	External message 4
Bit14	External message 5
Bit15	External message 6
Bit15 - Bit31	-

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PMonAnDi I/Os (Page 784).

Status bit	Description
Bit0 - Bit2	-
Bit3	Panel active
Bit4 - Bit5	-
Bit6	SimAct
Bit7	Fault
Bit8 - Bit21	-
Bit22	OosAct
Bit22 - Bit22	-

15.9.2 PMonAnDi operating modes

The block does not have any operating modes.

15.9.3 PMonAnDi functions

The "PMonAnDi" block acts as the interface between the "MonAnDi" technological block from the Industry Library (IL) and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.9.4 PMonAnDi error handling

Error handling

The block does not have any error handling.

15.9.5 PMonAnDi messaging

Message behavior

The messages correspond to those of the "MonAnDi" block from the Industry Library (IL).

15.9.6 PMonAnDi I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV8	Auxiliary value message value 1 ... 8	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to MonAnDi block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01 ... MsgEvId08	Message ID_1 ... ID_8	DWORD	16#60000001 ... 16#60000008
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_Mode	Operator mode	WORD	16#0000

Parameter	Description	Data type	Default
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_PV	Status process value	REAL	0.0
Op_PV_AH	Limit error AH	BOOL	0
Op_PV_AH_Lim	Reserved	REAL	95.0
Op_PV_AL	Limit error AL	BOOL	0
Op_PV_AL_Lim	Reserved	REAL	5.0
Op_PV_WH	Limit error WH	BOOL	0
Op_PV_WH_Lim	Reserved	REAL	90.0
Op_PV_WL	Limit error WL	BOOL	0
Op_PV_WL_Lim	Reserved	REAL	10.0
Op_ScaleHigh	PV - Bar display limits for OS (High)	REAL	100.0
Op_ScaleLow	PV - Bar display limits for OS (Low)	REAL	0.0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	16#00000004
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm *	Switch permission	DWORD	16#00000001

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
GrErr	Status: 1=Error	STRUCT	
GrErr.ST	Signal status	BYTE	16#80
GrErr.Value	Value	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn08	Message: ACK_STATE Output 1 ... 8	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
MsgStat01 ... MsgStat08	Message: STATUS Output 1 ... 8	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.9.7 PMonAnDi block diagram

The block has no block diagram.

15.9.8 Operator control and monitoring

15.9.8.1 PMonAnDi WinCC views

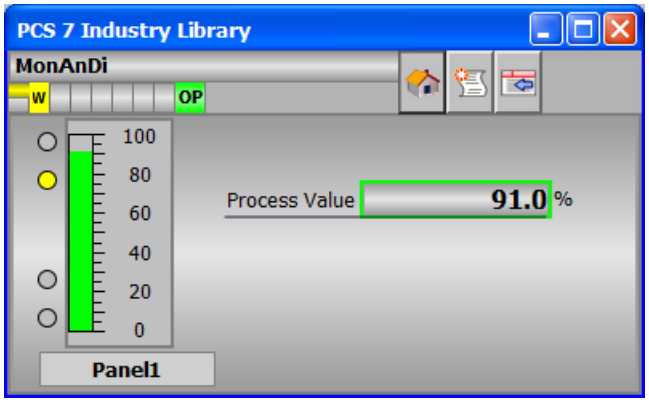
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.9.8.2 PMonAnDi WinCC Comfort views

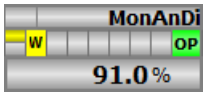
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.9.8.3 PMonAnDi WinCC Comfort standard view



15.9.8.4 PMonAnDi WinCC Comfort block icons



15.10 PMonDiL - Panel communication block for MonDiL

15.10.1 PMonDiL description

Object name (type + number) and family

Type + number: FB 1121

Family: Panel

Number of process objects: 1 PO

Configuring instructions

The "BlockConnector" input of "PMonDiL" must be connected to any given "MonDiL" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PMonDiL" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block MonDiL, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAl and OSStatAl_Out

For a description of the individual parameters, refer to the section PMonDiL I/Os (Page 790).

Status bit	Description
Bit0	External error has occurred (CSF)
Bit1	Out – Binary value set
Bit2	Flutter limits violated
Bit3 ... Bit8	-
Bit9	External message 1
Bit10	External message 2
Bit11	External message 3
Bit12 ... Bit31	-

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PMonDiL I/Os (Page 790).

Status bit	Description
Bit0 ... Bit2	-
Bit3	Panel active
Bit4 ... Bit5	-
Bit6	SimAct
Bit7	Fault
Bit8 ... Bit21	-
Bit22	OosAct
Bit23 ... Bit31	-

15.10.2 PMonDiL operating modes

The block does not have any operating modes.

15.10.3 PMonDiL functions

The "PMonDiL" block acts as the interface between the "MonDiL" technological block from the APL and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.10.4 PMonDiL error handling

The block does not have any error handling.

15.10.5 PMonDiL messaging

Message behavior

The messages correspond to those of the "MonDiL" block from the APL.

15.10.6 PMonDiL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV6	Auxiliary value message value 1 ... 6	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to MonDiL block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01	Message ID_1	DWORD	16#60000009
MsgEvId02 ... MsgEvId06	Message ID_2 ... ID_6	DWORD	16#6000000A ... 16#6000000E
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Q	Status process value	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0

Parameter	Description	Data type	Default
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch permission	DWORD	

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
GrErr	Status: 1=Error	STRUCT	
GrErr.ST	Signal status	BYTE	16#80
GrErr.Value	Value	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn06	Message: ACK_STATE Output 1 ... Output 6	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT	
MsgLock_Out.ST	Signal status	BYTE	16#80
MsgLock_Out.Value	Value	BOOL	0
MsgStat01 ... MsgStat06	Message: STATUS Output 1 ... Output 6	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	LL

15.10.7 PMonDiL block diagram

The block has no block diagram.

15.10.8 Operator control and monitoring

15.10.8.1 PMonDiL WinCC views

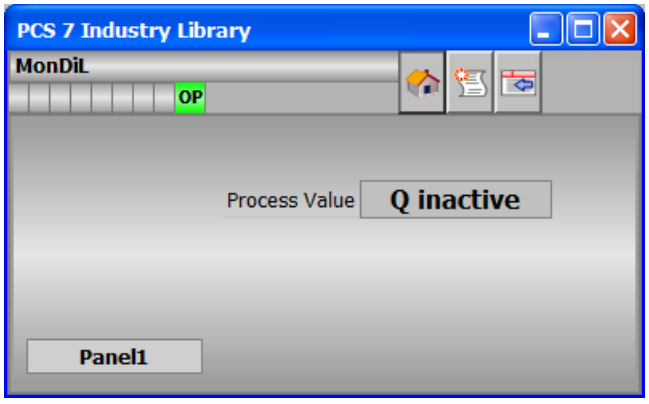
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.10.8.2 PMonDiL WinCC Comfort views

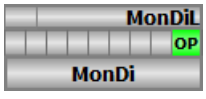
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.10.8.3 PMonDiL WinCC Comfort standard view



15.10.8.4 PMonDiL WinCC Comfort block icons



15.11 PMonDi08 - Panel communication block for MonDi08

15.11.1 PMonDi08 description

Object name (type + number) and family

Type + number: FB 1122

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PMonDi08" must be connected to any given "MonDi08" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PMonDi08" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block MonDi08, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAl and OSStatAl_Out

For a description of the individual parameters, refer to the section PMonDi08 I/Os (Page 795).

Status bit	Description
Bit0	-
Bit1 ... Bit8	Signal change signal 1 ... 8 has occurred
Bit9 ... Bit31	-

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PMonDi08 I/Os (Page 795).

Parameter	Description
Bit0 ... Bit2	-
Bit3	Panel active
Bit4 ... Bit6	-
Bit7	Fault
Bit8 ... Bit21	-
Bit22	OosAct
Bit23 ... Bit31	-

15.11.2 PMonDi08 operating modes

The block does not have any operating modes.

15.11.3 PMonDi08 functions

The "PMonDi08" block acts as the interface between the "MonDi08" technological block from the Industry Library (IL) and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.11.4 PMonDi08 error handling

The block does not have any error handling.

15.11.5 PMonDi08 messaging

Message behavior

The messages correspond to those of the "MonDi08" block from the APL.

15.11.6 PMonDi08 I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV8	Auxiliary value message value 1 ... 8	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to MonDi08 block	ANY	
CmpID	Area code	DWORD	16#00000001
EnAux	Enable measured value	BOOL	0
EN	-	BOOL	1
MsgEvId01 ... MsgEvId06	Message ID_1 ... ID_6	DWORD	16#60000001 ... 16#60000006
MsgEvId07 ... MsgEvId08	Message ID_7 ... ID_8	DWORD	16#60000013 ... 16#60000014
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Q1 ... Op_Q8	Status digital input 1 value ... 8 value	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm *	Switch permission	DWORD	16#00000001

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
GrErr	Status: 1=Error	STRUCT	
GrErr.ST	Signal status	BYTE	16#80
GrErr.Value	Value	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn08	Message: ACK_STATE Output 1 ... 8	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat08	Message: STATUS Output 1 ... 8	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.11.7 PMonDi08 block diagram

The block has no block diagram.

15.11.8 Operator control and monitoring

15.11.8.1 PMonDi08 WinCC views

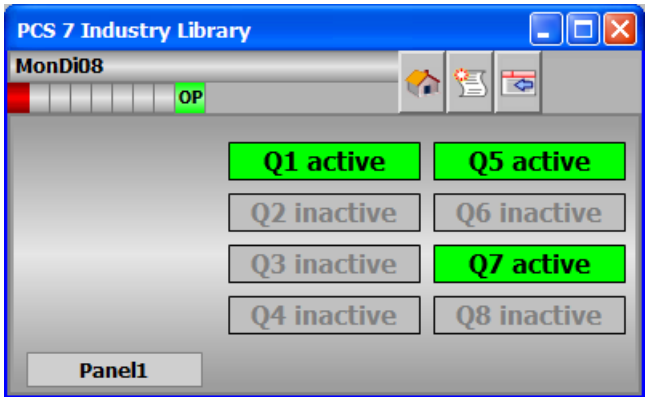
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.11.8.2 PMonDi08 WinCC Comfort views

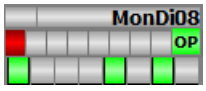
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.11.8.3 PMonDi08 WinCC Comfort standard view



15.11.8.4 PMonDi08 WinCC Comfort block icons



15.12 POpAnL - Panel communication block for OpAnL

15.12.1 POpAnL description

Object name (type + number) and family

Type + number: FB 1123

Family: Panel

Configuration

The "BlockConnector" input of "POpAnL" must be connected to any given "OpAnL" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

When the MsgLock_Out output is connected to the MsgLock input of the OpAnL technological block, the messages of the technological block are locked to avoid duplicate messages at the OS operator station. If control system faults (CSF) and external messages (ExtMsgx) are to be suppressed as well, feature bit 25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

If MsgLock = 0, "POpAnL" sends the messages to the operator panel and to the OS operator station.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAl and OSStatAl_Out

For a description of the individual parameters, refer to the section POpAnL I/Os (Page 800).

Status bit	Description
Bit0 - 7	Not used
Bit8	External message 1
Bit9	External message 2
Bit10	External message 3
Bit11	External message 4
Bit12 - 31	Not used

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section POpAnL I/Os (Page 800).

Status bit	Description
Bit0	Not used
Bit1	OnAct
Bit2	Not used
Bit3	Panel active
Bit4 - 6	Not used
Bit7	Error
Bit8 - 21	Not used
Bit22	OosAct
Bit23	SP_Ext
Bit24 - 31	Not used

15.12.2 POpAnL operating modes

The block does not have any operating modes.

15.12.3 POpAnL functions

The "POpAnL" block acts as the interface between the "OpAnL" technological block from the APL and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.12.4 POpAnL error handling

The block does not have any error handling.

15.12.5 POpAnL messaging

The messages correspond to those of the "OpAnL" block from the APL.

15.12.6 POpAnL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 … Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 … AV4	Auxiliary value message value 1 ... 4	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to OpAnL block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01	Message ID_1	DWORD	16#6000000F
MsgEvId02 … MsgEvId04	Message ID_2 ... 4	DWORD	16#60000010 ... 16#60000012
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_Mode	RESERVED Not used yet.	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_PV	Feedback analog input	REAL	0.0
Op_SP	Operator panel setpoint	REAL	0.0
Op_SP_HiLim	High limit U	REAL	100.0
Op_SP_LoLim	Low limit U	REAL	0.0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10

Parameter	Description	Data type	Default
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch permission	DWORD	16#00000001

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ExtMsg1_Out ... ExtMsg4_Out	External message 1 ... 4 Out	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
ENO	1 = Block algorithm completed without errors	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn04	Message: ACK_STATE Output 1 ... 4	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat04	Message: STATUS Output 1 ... 4	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.12.7 POpAnL block diagram

The block has no block diagram.

15.12.8 Operator control and monitoring

15.12.8.1 POpAnL WinCC views

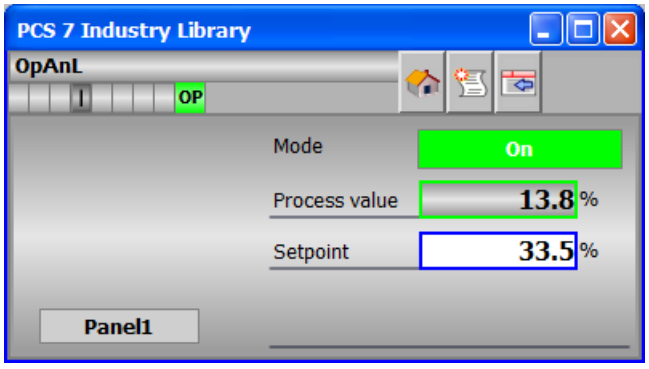
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.12.8.2 POpAnL WinCC Comfort views

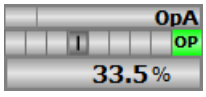
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.12.8.3 POpAnL WinCC Comfort standard view



15.12.8.4 POpAnL WinCC Comfort block icons



15.13 POpD - Panel communication block for OpDi01 / OpTrig

15.13.1 POpD description

Object name (type + number) and family

Type + number: FB 1131

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "POpD" must be connected to any given output of the OpDi01 or OpTrig.

The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

You must also configure which block type is connected with the "BlockType" input. The following values are possible:

- 1: OpDi01
- 2: OpTrig

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set when the block is to be processed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC24	TEST_DB

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section POpD I/Os (Page 805).

Status bit	Description
Bit0	Not used
Bit1	OnAct
Bit2	Not used
Bit3	Panel active
Bit4 - 21	Not used
Bit22	OosAct
Bit23 - 31	Not used

15.13.2 POpD operating modes

The block does not have any operating modes.

15.13.3 POpD functions

The "POpD" block acts as the interface between the "OpDi01" and "OpTrig" technological blocks from the APL and the operator panel.

15.13.4 POpD error handling

The block does not have any error handling.

15.13.5 POpD messaging

Message behavior

The block has no message behavior.

15.13.6 POpD I/Os

Input parameters

Parameter	Description	Data type	Default
Aux1 … Aux4	Auxiliary value1 ... 4	REAL	0.0
BlockConnector	Connection to OpDL block	ANY	
BlockType	BlockType (1 = OpDi01, 2 = OpTrig)	INT	1
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
Op_FbkIn	Feedback input	BOOL	0
Op_Mode	RESERVED Not used yet.	WORD	16#0000
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Q	Status 1=Operator Panel output 1: • 0=inactive • 1=Active	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch permission	DWORD	16#00000001

Output parameters

Parameter	Description	Data type	Default
BlockType_Err	Unknown BlockType number	BOOL	1
DB_ConnErr	Data block connection error	BOOL	1
ENO		BOOL	0
IDBNo	Number instance DB	INT	0
OSStat_Out	Status WinCC	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.13.7 POpD block diagram

The block has no block diagram.

15.13.8 Operator control and monitoring

15.13.8.1 POpD WinCC views

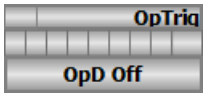
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.13.8.2 Pulse command (OpTrig) views from POpD WinCC Comfort

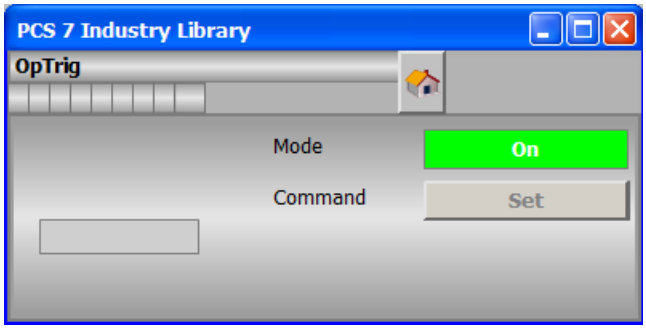
The block provides the following views:

- Standard view
- Associated value view

15.13.8.3 Block icons - Pulse command (OpTrig) from POpD WinCC Comfort



15.13.8.4 Standard view - Pulse command (OpTrig) from POpD WinCC Comfort

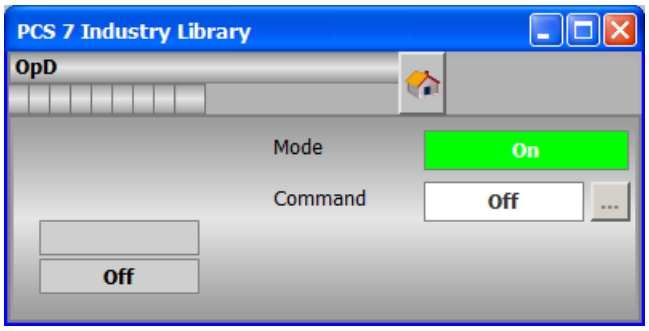


15.13.8.5 Static command (OpD) views from POpD WinCC Comfort

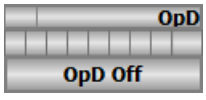
The block provides the following views:

- Standard view
- Associated value view

15.13.8.6 Standard view - Static command (OpD) from POpD WinCC Comfort



15.13.8.7 Block icons- Static command (OpD) from POpD WinCC Comfort



15.14 PPIDL - Panel communication block for PIDL

15.14.1 PPIDL description

Object name (type + number) and family

Type + number: FB 1138

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PPIDL" must be connected to any given output of the "PIDConL", PIDConR" or "PIDStepL".

The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the "Blockconnector" input.

You must also configure which block type is connected with the "BlockType" input. The following types are possible:

- 1: PIDConR
- 2: PIDConL
- 3: PIDStepL

If the parameter MsgLock = 0, "PPIDL" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAI and OSStatAI_Out

For a description of the individual parameters, refer to the section PPIDL I/Os (Page 811).

Status bit	Description
Bit0	External error has occurred (CSF)
Bit1	PV - High alarm limit
Bit2	PV - High warning limit
Bit3	PV - High tolerance limit
Bit4	PV - Low tolerance limit
Bit5	PV - Low warning limit
Bit6	PV - Low alarm limit
Bit7	ER - High alarm limit
Bit8	ER - Low alarm limit
Bit9	Rbk - High warning limit
Bit10	Rbk - Low warning limit
Bit11	External message 1
Bit12	External message 2
Bit13	External message 3
Bit14	External message 4
Bit15 - 31	Not used

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PPIDL I/Os (Page 811).

Status bit	Description
Bit0	Not used
Bit1	ManAct.Value
Bit2	Not used
Bit3	Panel active
Bit4	Auto active-
Bit5	Not used
Bit6	SimOn

Status bit	Description
Bit7	Error (CSF.Value or ExtMsg1.Value)
Bit8	FbkOpened.Value
Bit9	FbkClosed.Value
Bit10 - 11	Not used
Bit12	Control + (NegGain = 0)
Bit13	Control – (NegGain = 1)
Bit14	MV_ForOn
Bit15 - 16	Not used
Bit17	SP_LiOp.Value
Bit18 - 19	Not used
Bit20	AdvCoActMV
Bit21	AdvCoActSP
Bit22	OosAct.Value
Bit23	SP_Ext
Bit24 - 31	Not used

15.14.2 PPIDL operating modes

The block does not have any operating modes.

15.14.3 PPIDL functions

The "PPIDL" block acts as the interface between the "PIDConL", "PIDConR", and "PIDStepL" technological blocks from the APL and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.14.4 PPIDL error handling

The block does not have any error handling.

15.14.5 PPIDL messaging

Message behavior

The messages correspond to those of the "PIDConL", "PIDConR", or "PIDStepL" blocks from the APL.

15.14.6 PPIDL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 … Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 … AV15	Auxiliary value message value 1 ... 15	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to PIDConL, PIDConR or PIDStepL block	ANY	
BlockType	BlockType (1 = PIDConR, 2 = PIDConL, 3 = PIDStepL)	INT	0
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01 … MsgEvId09	Message ID_1 ... 9	DWORD	16#60000001 ... 16#60000009
MsgEvId10 … MsgEvId14	Message ID_10 ... 14	DWORD	16#6000000A ... 16#6000000E
MsgEvId15	Message ID_15	DWORD	16#60000013
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_MV	Operator panel manipulated variable	REAL	0.0
Op_MVHL	Limit (high) for manipulated variable	REAL	100.0
Op_MVLL	Limit (low) for manipulated variable	REAL	0.0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_PV	Process value	REAL	0.0
Op_Rbk	Readback value of the actuator position	REAL	0.0
Op_ScMVHL	OpScale value (high) for manipulated variable	REAL	100.0
Op_ScMVLL	OpScale value (low) for manipulated variable	REAL	0.0
Op_ScPVHL	OpScale value (high) for process value	REAL	100.0
Op_ScPVLL	OpScale value (low) for process value	REAL	0.0
Op_SP	Operator panel setpoint status	REAL	0.0
Op_SP_Ext	Switch to external setpoint	BOOL	0
Op_SP_Int	Switch to internal setpoint	BOOL	0
Op_SPHL	Limit (high) for setpoint	REAL	100.0

Parameter	Description	Data type	Default
Op_SPLL	Limit (low) for setpoint	REAL	0.0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm *	Switch permission	DWORD	

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
BlockType_Err	Unknown BlockType number	BOOL	1
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn15	Message: ACK_STATE Output 1 ... 15	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat15	Message: STATUS Output 1 ... 15	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.14.7 PPIDL block diagram

The block has no block diagram.

15.14.8 Operator control and monitoring

15.14.8.1 PPIDL WinCC views

The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54).

15.14.8.2 PPIDL WinCC Comfort views

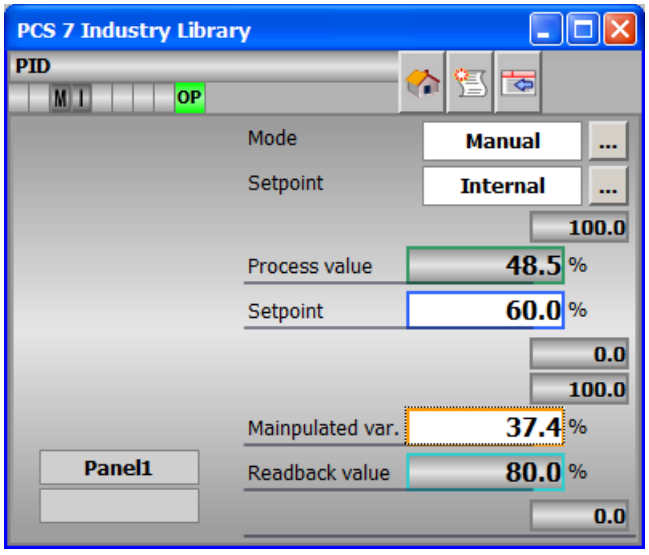
The block provides the following views:

- Standard view
- Message view
- Associated value view

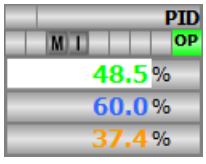
See also

Panel integration (Page 44)

15.14.8.3 PPIDL WinCC Comfort standard view



15.14.8.4 PPIDL WinCC Comfort block icons



15.15 PVlvL - Panel communication block for VlvL

15.15.1 PVlvL description

Object name (type + number) and family

Type + number: FB 1136

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PVlvL" must be connected to any given "VlvL" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PVlvL" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block VlvL, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAI and OSStatAI_Out

For a description of the individual parameters, refer to the section PVlvL I/Os (Page 816).

Parameter	Description
Bit0	External error has occurred (CSF)
Bit1	Feedback error
Bit2 - 4	Not used
Bit5	External message 1
Bit6	External message 2
Bit7	External message 3
Bit8 - 31	Not used

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PVlvL I/Os (Page 816).

Parameter	Description
Bit0	LocalAct.Value
Bit1	ManAct.Value
Bit2	Not used
Bit3	Panel active
Bit4	AutoAct.Value
Bit5	SimOn.Value
Bit6	Not used
Bit7	Fault
Bit8	Valve open
Bit9	Valve closed
Bit10	Not used
Bit11	Not used
Bit12	Valve opens
Bit13	Valve closes
Bit14	OpenForce.Value
Bit15	CloseForce.Value
Bit16	Not used
Bit17	OpenClose.Value
Bit18	Show fault profile
Bit19 - 21	Not used
Bit22	OosAct.Value
Bit23 - 31	Not used

15.15.2 PVlvL operating modes

The block does not have any operating modes.

15.15.3 PVlvL functions

The "PVlvL" block acts as the interface between the "VlvL" technological block from the APL and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.15.4 PVlvL error handling

The block does not have any error handling.

15.15.5 PVlvL messaging

Message behavior

The messages correspond to those of the "VlvL" block from the APL.

15.15.6 PVlvL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV5	Auxiliary value message value 1 ... 5	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to VlvL block	ANY	
CmpID	Area code	DWORD	16#00000001
EnAux	Enable measured value	BOOL	0
EN	-	BOOL	1
MsgEvId01	Message ID_1	DWORD	16#60000009

Parameter	Description	Data type	Default
MsgEvId02 ... MsgEvId05	Message ID_2 ... ID_5	DWORD	16#6000000A ... 16#6000000D
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_CloseMan	Operator input manual close	BOOL	0
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_OpenMan	Operator input manual open	BOOL	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Reset	Operator input error reset	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch permission	DWORD	16#00000001

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn05	Message: ACK_STATE Output 1 ... 5	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat05	Message: STATUS Output 1 ... 5	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.15.7 PVlvL block diagram

The block has no block diagram.

15.15.8 Operator control and monitoring

15.15.8.1 PVlvL WinCC views

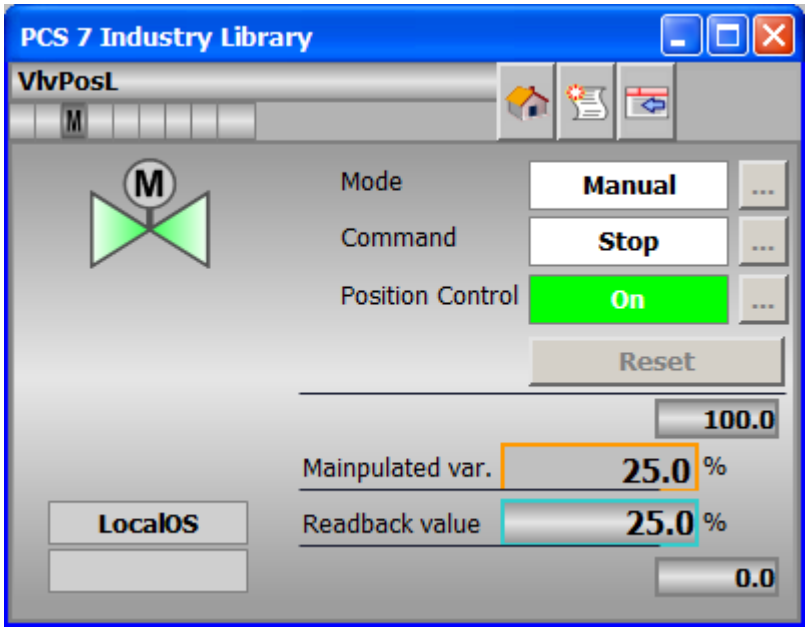
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54)

15.15.8.2 PVlvL WinCC Comfort views

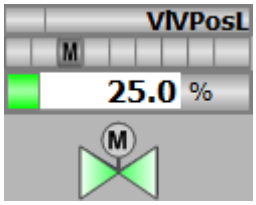
The block provides the following views:

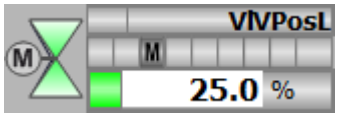
- Standard view
- Message view
- Associated value view

15.15.8.3 PVlvL WinCC Comfort standard view



15.15.8.4 PVlvL WinCC Comfort block icons





15.16 PVlvMotL - Panel communication block for VlvMotL

15.16.1 PVlvMotL description

Object name (type + number) and family

Type + number: FB 1143

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PVlvMotL" must be connected to any given "VlvMotL" output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output cannot be used for the interconnection to the Blockconnector input.

If the parameter MsgLock = 0, "PVlvMotL" sends the messages to the operator panel and the OS operator station.

If the MsgLock_Out output is connected with the MsgLock input of the technological block VlvMotL, the messages at the technological block are blocked to prevent duplication of the messages on the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAl and OSStatAl_Out

For a description of the individual parameters, refer to the section PVlvMotL I/Os (Page 822).

Status bit	Description
Bit0	External error has occurred (CSF)
Bit1	Motor feedback error
Bit2	Motor protection triggered
Bit3	Valve feedback error
Bit4	Not used
Bit5	External message 1
Bit6	External message 2
Bit7	External message 3
Bit8 - 31	Not used

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PVlvMotL I/Os (Page 822).

Status bit	Description
Bit0	LocalAct.Value
Bit1	ManAct.Value
Bit2	Not used
Bit3	Panel active
Bit4	AutoAct.Value
Bit5	SimOn.Value
Bit6	Not used
Bit7	Fault
Bit8	Valve open
Bit9	Valve closed
Bit10	Motor stopped
Bit11	Not used
Bit12	Valve opens
Bit13	Valve closes
Bit14	OpenForce.Value
Bit15	CloseForce.Value
Bit16	StopForce_Value
Bit17	"Open" command
Bit18	Show fault profile
Bit19	"Close" command
Bit20	"RapidStop" command

Status bit	Description
Bit21	"Stop" command
Bit22	OosAct_Value
Bit23 - 31	Not used

15.16.2 PVlvMotL operating modes

The block does not have any operating modes.

15.16.3 PVlvMotL functions

The "PVlvMotL" block acts as the interface between the "VlvMotL" technological block from the APL and the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.16.4 PVlvMotL error handling

The block does not have any error handling.

15.16.5 PVlvMotL messaging

Message behavior

The messages correspond to those of the "VlvMotL" block from the APL.

15.16.6 PVlvMotL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 … Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 … AV7	Auxiliary value message value 1 ... 7	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
BlockConnector	Connection to VlvMotL block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01 ... MsgEvId07	Message ID_1 ... 7	DWORD	16#60000001 ... 16#60000007
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_CloseMan	Operator input manual close	BOOL	0
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_OpenMan	Operator input manual open	BOOL	0
Op_PermLog	Operator permission	DWORD	16#00000000
Op_Reset	Operator input error reset	BOOL	0
Op_StopMan	Operator input manual stop	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm *	Switch permission	DWORD	16#00000001

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn07	Message: ACK_STATE Output 1 ... 7	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
MsgStat01 ... MsgStat07	Message: STATUS Output 1 ... 7	WORD	16#0000
MsgErr	1=Message error	BOOL	0
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch permission	DWORD	16#00000001

15.16.7 PVlvMotL block diagram

The block has no block diagram.

15.16.8 Operator control and monitoring

15.16.8.1 PVlvMotL WinCC views

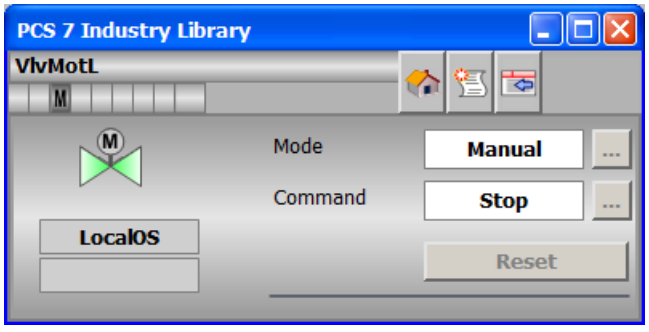
The views of the block in WinCC are available at Operator Control and Monitoring in WinCC (Page 54).

15.16.8.2 PVlvMotL WinCC Comfort views

The block provides the following views:

- Standard view
- Message view
- Associated value view

15.16.8.3 PVlvMotL WinCC Comfort standard view



15.16.8.4 PVlvMotL WinCC Comfort block icons





15.17 PVlvPosL - Communication block for VlvPosL

15.17.1 PVlvPosL description

Object name (type + number) and family

Type + number: FB 1137

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PVlvPosL" must be connected to any VlvPosL output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output must not be used for interconnection with the BlockConnector input.

If parameter MsgLock = 0, "PVlvPosL" sends the messages to the operator panel and to the OS operator station.

When the MsgLock_Out output is connected to the MsgLock input of the VlvPosL technological block, the messages of the technological block are locked to avoid duplicate messages at the OS operator station. If you want control system faults (CSF) and external messages (ExtMsgx) to be suppressed as well, you must also set feature bit 25 ("1 = Suppress all messages if MsgLock = 1") at the technological block.

The IDBNo output must point to an address in the PanelConDB data block. This location in the data block is required for connecting the panel.

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

In the same OB that accompanies and follows the block whose measured value is to be monitored. Also in OB100.

Status word assignment for OSStatAl and OSStatAl_Out parameters

For a description of the individual parameters, refer to the section PVlvPosL I/Os (Page 829)

Parameter	Description
Bit0	External error occurred (CSF)
Bit1	Motor feedback error
Bit2	Motor protection triggered
Bit3	Valve feedback error
Bit4	Not used
Bit5	Not used
Bit6	Not used
Bit7	ER - High alarm limit violated
Bit8	ER - Low alarm limit violated
Bit9	Rbk - High warning limit violated
Bit10	Rbk - Low warning limit violated
Bit11	Not used
Bit12	External message 1
Bit13	External message 2
Bit14	External message 3
Bit15	External message 4
Bit16 - 31	Not used

Status word assignment for OSStat and OSStat_Out parameters

For a description of the individual parameters, refer to the section PVlvPosL I/Os (Page 829)

Parameter	Description
Bit0	LocalAct.Value
Bit1	ManAct.Value
Bit2	Not used
Bit3	Panel active
Bit4	AutoAct.Value
Bit5	SimOn.Value
Bit6	Not used
Bit7	Fault
Bit8	Valve open
Bit9	Valve closed
Bit10	Motor stopped
Bit11	Not used

Parameter	Description
Bit12	Valve opens
Bit13	Valve closes
Bit14	`OpenForce`.Value
Bit15	`CloseForce`.Value
Bit16	`StopForce`.Value
Bit17	Open Command
Bit18	Show fault profile
Bit19	Close Command
Bit20	RapidStop Command
Bit21	Stop Command
Bit22	`OosAct`.Value
Bit23	`MV_ForOn`.Value
Bit24	`PosAct`.Value
Bit25	NOT `PosAct`.Value
Bit26 - 31	Not used

15.17.2 Operating modes of PVlvPosL

The block has no operating modes.

15.17.3 PVlvPosL functions

The "PVlvPosL" block acts as the interface between the "VlvPosL" technological block from the APL to the operator panel.

It is a message-capable block that generates the same Alarm_DQ messages as the associated technological block.

15.17.4 PVlvPosL error handling

The block has no error handling.

15.17.5 PVlvPosL messaging

Message functionality

The messages correspond to those of the "VlvPosL" block from the APL.

15.17.6 PVlvPosL I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS Number for message filter (max. 999)	INT	1
Aux1 … Aux4	Auxilliary Value1 ... 4	REAL	0.0
AV1 … AV12	Auxiliary Value Message Value 1 ... 12	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to VlvPosL block	ANY	
CmpID	area code	DWORD	16#00000000
EN	-	BOOL	1
EnAux	Enable Measured Value	BOOL	0
MsgEvId01	Message ID_1	DWORD	16#00000008
MsgEvId02 … MsgEvId03	Message ID_2 ... 3	DWORD	16#0000000E ... 16#0000000F
MsgEvId04 … MsgEvId12	Message ID_4 ... 12	DWORD	16#000000010 ... 16#000000018
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_CloseMan	Operator Input Manual Close	BOOL	0
Op_Mode	Operator Mode	BOOL	0
Op_MsgFilter	Message filter for panel	BOOL	0
Op_MV	Manipulated var.	REAL	0.0
Op_MVHL	Limit (high) for Manipulated var.	REAL	100.0
Op_MVLL	Limit (low) for Manipulated var.	REAL	0.0
Op_OpenMan	Operator Input Manual Open	BOOL	0
Op_PermLog	Operator Permission	DWORD	16#00000000
Op_PosOnMan	Operator Input Manual PosCon	BOOL	0
Op_Rbk	Actuator Position	REAL	0.0
Op_Reset	Operator Input Error Reset	BOOL	0
Op_ScMVHL	OpScale value (high) for Manipulated var.	REAL	100.0
Op_ScMVLL	OpScale value (low) for Manipulated var.	REAL	0.0
Op_StopMan	Operator Input Manual Stop	BOOL	0
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit Reporting Procedure	DWORD	16#00000000
PanelPerm	Panel Permission	DWORD	Bedienpult1
Restart	Manual Restart	BOOL	0

Parameter	Description	Data type	Default
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
SwitchPerm	Switch Permission	DWORD	16#00000001

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Datablock connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn12	Message: ACK_STATE Output 1 ... 12	BOOL	0
MsgErr	1=Message Error	BOOL	0
MsgLock_Out	Message Lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat12	Message: STATUS Output 1 ... 12	WORD	16#0000
MsgSup	1=Message Suppression Active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit Reporting Procedure	DWORD	16#00000000
PanelAct	Panel active	BOOL	0
SwitchPerm_Out	Switch Permission	DWORD	16#00000001

15.17.7 PVlvPosL block diagram

The block has no block diagram.

15.17.8 Operator control and monitoring

15.17.8.1 PVlvPosL WinCC views

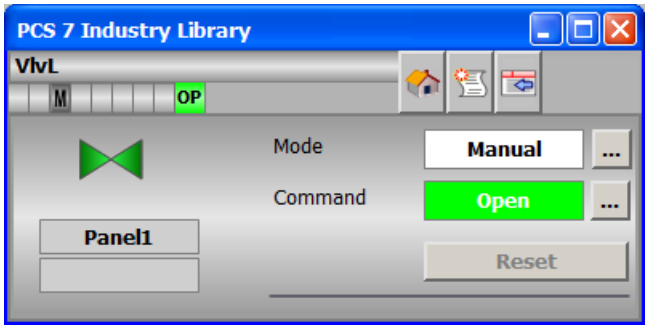
You can find the views of the block in WinCC under Panel block views (Page 54)

15.17.8.2 PVlvPosL WinCC Comfort views

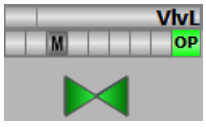
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.17.8.3 PVlvPosL WinCC Comfort standard view



15.17.8.4 PVlvPosL WinCC Comfort block icons



15.18 PUsrM - Panel communication block for UsrM

15.18.1 PUsrM description

Object name (type + number) and family

Type + number: FB 1132

Family: Panel

Number of process objects: 1 PO

Configuration

The "BlockConnector" input of "PUsrM" must be connected to a UsrM output. The relevant data for the panel block (for example, status information or messages) are then automatically processed via the panel block and displayed accordingly.

Exception:

The ENO output must not be used for interconnection with the Block Connector input.

If parameter MsgLock = 0, "PUsrM" sends the messages to the operator panel and to the OS operator station.

When the MsgLock_Out output is connected to the MsgLock input of the UsrM technological block, the messages of the technological block are locked to avoid duplicate messages at the OS operator station. If control system messages (CSF) and external messages (ExtMsgx) are suppressed as well, Feature Bit25 ("1 = Suppress all messages if MsgLock = 1") must also be set at the technological block.

The IDBNo output has to point to an address of the PanelConDB data block. This location in the data block is required for connecting the panel.

The operation can be enabled for PUsrM on the panel using the "@ILPanelStation" operator station ID. The "PUsrM" block is enabled when the comparison between the PanelPerm parameter and "@ILPanelStation" is not equal to "0".

Startup characteristics

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Restart = TRUE can be used to simulate a restart.

Called blocks

SFC6	RD_SINFO
SFC19	ALARM_SC
SFC24	TEST_DB
SFC107	ALARM_DQ

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameters OSStatAI and OSStatAI_Out

For a description of the individual parameters, refer to the section PUsrM I/Os (Page 834).

Status bit	Description
Bit0	Keyswitch activated
Bit1	Parameter assignment error
Bit2	Operating level changed
Bit3	Backup level 1 selected
Bit4	Backup level 2 selected
Bit5	Next available level activated
Bit6	No operating level available
Bit7	External error
Bit8 - 31	Not used

Status word assignment for the parameters OSStat and OSStat_Out

For a description of the individual parameters, refer to the section PUsrM I/Os (Page 834).

Status bit	Description
Bit0	Dev01Act
Bit1	Dev02Act
Bit2	Dev03Act
Bit3	Dev04Act
Bit4	Dev05Act
Bit5	Dev06Act
Bit6	Dev07Act
Bit7	Dev08Act
Bit8	Dev01ActOut
Bit9	Dev02ActOut
Bit10	Dev03ActOut
Bit11	Dev04ActOut
Bit12	Dev05ActOut
Bit13	Dev06ActOut
Bit14	Dev07ActOut
Bit15	Dev08ActOut
Bit16	Keyswitch activated
Bit17	EnAux
Bit18	Parameter assignment error or external error
Bit19	Process errors
Bit20 - 31	Not used

15.18.2 PUsrM operating modes

The block does not have any operating modes.

15.18.3 PUsrM functions

The "PUsrM" block acts as the interface between the "UsrM" technological block from the APL to the operator panel.

It is a message-capable block and generates the same Alarm_DQ messages as the associated technological block.

15.18.4 PUsrM error handling

The block does not have any error handling.

15.18.5 PUsrM messaging

Message behavior

The messages correspond to those of the "UsrM" block of the APL.

15.18.6 PUsrM I/Os

Input parameters

Parameter	Description	Data type	Default
AS_Nr	AS number for message filter (max. 999)	INT	1
Aux1 ... Aux4	Auxiliary value1 ... 4	REAL	0.0
AV1 ... AV8	Auxiliary value message value 1 ... 7	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
BlockConnector	Connection to UsrM block	ANY	
CmpID	Area code	DWORD	16#00000001
EN	-	BOOL	1
EnAux	Enable measured value	BOOL	0
MsgEvId01 ... MsgEvId08	Message ID_1 ... 8	DWORD	16#60000001 ... 16#60000008

Parameter	Description	Data type	Default
MsgLock	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Op_MaxLevel	Maximum number of input devices	INT	0
Op_Mode	Operator mode	WORD	16#0000
Op_MsgFilter	Message filter for panel	DINT	0
Op_Out	Output enabled operator station	DWORD	16#00000000
Op_ SwitchPerm	Switch permission	INT	1
Op_Visibility	Objects visibility	DWORD	16#00000000
OSStat	Status WinCC	DWORD	16#00000000
OSStatAl	Bit reporting procedure	DWORD	16#00000000
PanelPerm	Panel permission	DWORD	Bedienpult1
Restart	Manual restart	BOOL	0
RunUPCyc	Number of cycles for which this block keeps in start up mode	INT	10
SelFp1/2	Select Faceplate 1/2	BOOL	
Op_PermLog *	Operator permission	DWORD	16#00000000

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
DB_ConnErr	Data block connection error	BOOL	1
ENO	1 = Block algorithm completed without errors	BOOL	0
GrErr		STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
IDBNo	Number instance DB	INT	0
MsgAckn01 ... MsgAckn08	Message: ACK_STATE Output 1 ... 8	BOOL	0
MsgErr	1=Message error	BOOL	0
MsgLock_Out	Message lock	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgStat01 ... MsgStat08	Message: STATUS Output 1 ... 8	WORD	16#0000
MsgSup	1=Message suppression active	BOOL	0
OSStat_Out	Status WinCC	DWORD	16#00000000
OSStatAl_Out	Bit reporting procedure	DWORD	16#00000000

Parameter	Description	Data type	Default
SwitchPerm_Out	Switch permission	DWORD	16#00000001
GrErr	1=Error	BOOL	0

15.18.7 PUsrM

The block has no block diagram.

15.18.8 Operator control and monitoring

15.18.8.1 PUsrM WinCC views

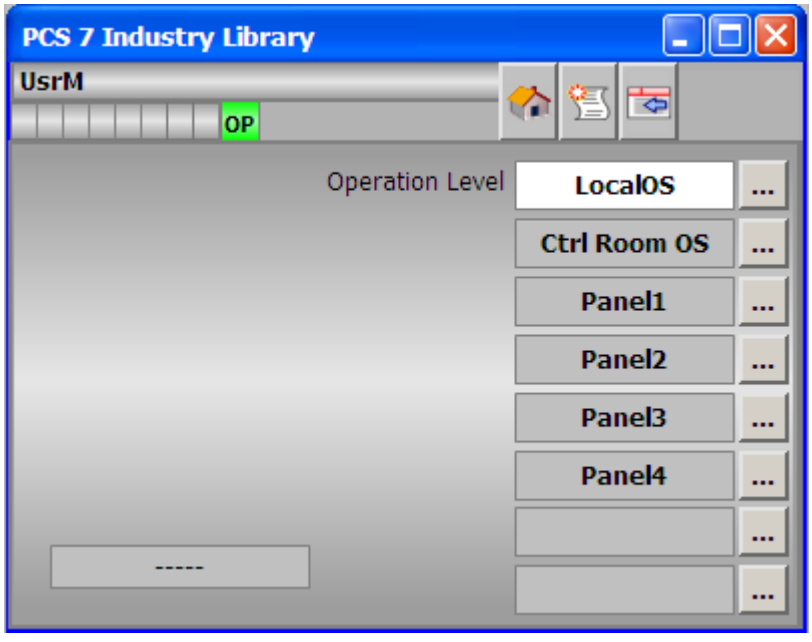
The views of the block in WinCC are available at Panel block views (Page 54).

15.18.8.2 PUsrM WinCC Comfort views

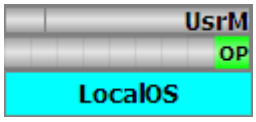
The block provides the following views:

- Standard view
- Message view
- Associated value view

15.18.8.3 PUsrM WinCC Comfort standard view



15.18.8.4 PUsrM WinCC Comfort block icons



Controller blocks 16

16.1 Higher-level controlling of the unit / EQM level

16.1.1 Structure data connections EQM <-> Monitoring function

Signal transmission from the EQM level to the control modules

The data connection of an equipment module with the lower-level CMs is carried out on the basis of a 1:n relationship, meaning that a connection of the EQM interface block (EqmIf, FxCtrl) from the output "CMData" to every assigned CM is required.

This contains the following individual information:

- Batch data
- Unit identifier
- Mode switch / enabling signals

Elements of the data structure `CmData`

Structure element	Description	Type
dwUnitID	UnitID	`DWORD`
dwBatchID	Batch ID	`DWORD`
sBatchName	Batch name	`STRING[32]`
dwBatchStepNo	Batch step number	`DWORD`
xBatchOcc	Batch occupied	`BOOL`
xBatchEn	Batch enabled	`BOOL`
xSimEnCM	Simulation is enabled	`BOOL`
xSimOnParent	Unit or equipment simulation is active	`BOOL`
xEqmLock	Lock equipment module	`BOOL`
xManOpEn	Operator may switch to MANUAL mode	`BOOL`
xResetPls	Reset signal (pulse)	`BOOL`
xModLiOp	Mode switch initiator	`BOOL`
xAutModLi	Switch to AUTO mode	`BOOL`
xManModLi	Switch to MANUAL mode	`BOOL`
xMsgLock	Lock all Messages	`BOOL`
xRes1		`BOOL`
xRes2		`BOOL`

Structure element	Description	Type
xRes3		BOOL
xRes4		BOOL
xRes5		BOOL
byFbkDelayRatio	Feedback delay ratio (in percent)	BYTE

Note

This information can also be switched groupwise to external inputs via various "Overwrite" signals in order to allow the CM-individual control:

The following table lists the individual signals per category as well as the associated switchover signals:

Category	Data	Switchover signal
Unit identifier	UnitID	OvwUnitID
Batch data	BatchIDBatchNameStepNoBatchOccBatchEn	OvwBatchInfo
Manual switching operation	ManOpEn	OvwManOpEn
Mode switch	ModLiOpAutModLi	OvwModSel
Simulation switchover	SimEnExtSimOnExt	OvwSimInfo

The following signals cannot be switched over:

- EQM lock
- Reset pulse

16.2 PolygonExt - Polygon line with up to 8 interpolation points

16.2.1 PolygonExt description

Object name (type + number) and family

Type + number: FB 1155

Family: Control

Number of process objects: 1 PO

Area of application

The block calls the Polygon function block from the Advanced Process Library (APL). There are two methods of using the PolygonExt function block:

- Configuring the interpolation points via the faceplate
- Specification of the interpolation points via interconnection

Operation via faceplate (Feature.Bit1 = 0)

The interpolation points `In1..8`, `Out1..8`, and `Num` (input for specifying the number of interpolation points) can be operated via a faceplate. Operation has to be acknowledged positively via the faceplate (`OkOp`) so that the entered values can be applied to the output.

If an operation is acknowledged negatively (`CancelOp`), the operator inputs are adjusted to the outputs so that the operation is rejected. To provide you with a graphical impression of the entered values, the resulting trend is displayed dashed next to the currently set trend. Only after the new values have been accepted is the temporary trend (dashed) converted to the static representation.

Note

If `Feature.Bit1 = 0` but an input that should actually be operated is nevertheless interlinked, none of the interpolation points `In1..8`, `Out1..8`, and `Num` can be operated via the faceplate.

Operation via interconnection (`Feature.Bit1 = 1`)

Alternatively, interpolation points `In1..8`, `Out1..8`, and `Num` can also be interconnected. For this purpose, `Feature.Bit1` has to be set. In this case, the values are applied immediately. If the values are faulty, a message is generated and the last valid value is output at the `Out` output (standard APL polygon function block behavior).

Startup characteristics

The operation values are copied to the active process values during startup.

The `RunUpCyc` parameter can be used to set for how long (number of cycles) messages are to be suppressed.

`Restart = TRUE` can be used to simulate a restart.

Called blocks

FB1881	Polygon
FC369	SelST16
SFC6	RD_SINFO
SFB35	ALARM_8P

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32). Also in OB100.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section PolygonExt I/Os (Page 846).

Status bit	Parameter
0	Belegt
1	`BatchEn`
2 - 31	Not used

Status word assignment for the parameter `Status2`

For a description of the individual parameters, refer to the section PolygonExt I/Os (Page 846).

Status bit	Parameter
0	MsgLock = 1
1 - 2	Not used
3	`OpAct`: Operation via faceplate active (NOT `Feature.Bit1` AND NOT `Status2.Bit4`)
4	`ProErr`: Configuration error. In linked operation: Not all the In(i) and Out(i) values that are required are connected. In the operator mode: Input to be operated is connected.
3 - 31	Not used

16.2.2 PolygonExt operating modes

The block can be operated using the following modes:

- On
- Out of service

General information on the operating modes is available in the section *Operating modes of the blocks* in the *APL Function manual*.

16.2.3 PolygonExt functions

The functions for this block are listed below.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4 - 11	Not used
12	1 = Operator can activate the Release for maintenance function
13	1 = Operator can enter number
14	1 = Operator can enter In1
15	1 = Operator can enter In2
16	1 = Operator can enter In3
17	1 = Operator can enter In4
18	1 = Operator can enter In5
19	1 = Operator can enter In6
20	1 = Operator can enter In7
21	1 = Operator can enter In8
22	1 = Operator can enter Out1
23	1 = Operator can enter Out2
24	1 = Operator can enter Out3
25	1 = Operator can enter Out4
26	1 = Operator can enter Out5
27	1 = Operator can enter Out6
28	1 = Operator can enter Out7
29	1 = Operator can enter Out8
30 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	1 = `InX`, `OutX`, and `Num` interconnected

Bit	Function
2	See APL function: "Reaction to the out of service operating mode"
5	See APL function: "Limit output Out"
6	See APL function: "Substitute value switch in the event of an error"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"
26	See APL function: "Reaction of the switching points in the "Out of service" operating mode"
28	See APL function: "Deactivate switching points"
29	See APL function: "Signal limit violation"

Configurable reactions using the `Feature2` parameter

Bit	Function
0 - 31	Not used

16.2.4 PolygonExt error handling

The ParamFail error output is set if:

- `In`(i) <= `In`(i+1) for i = 1,2 ... `Num`-1

The `ErrOut` error output is set if:

- `ParamFail = TRUE` or
- Number of interpolation points `Num` < 2 or N > 8

When the input or the interconnection of `Num` is faulty, `Num` is restricted to the limits 2/8 and a corresponding message is issued.

When the input or the connection of `In`(i) is faulty, a corresponding message is issued.

16.2.5 PolygonExt messaging

The `PolygonExt` block uses the `ALARM_8P` block for generating messages.

Process messages

Message instance	Message ID	Message class	Event
MsgEvId	SIG 1	AS control system fault	$$BlockComment$$ Number of interpolation points limited
	SIG 2	AS control system fault	$$BlockComment$$ X-values of interpolation points not ascending
	SIG 3	AS control system fault	$$BlockComment$$ External message 1
	SIG 4	AS control system fault	$$BlockComment$$ External message 2
	SIG 5	AS control system fault	$$BlockComment$$ External message 3
	SIG 6	AS control system fault	$$BlockComment$$ External message 4
	SIG 7	AS control system fault	$$BlockComment$$ External message 5
	SIG 8	AS control system fault	$$BlockComment$$ External message 6

Associated values for the message instance MsgEvId

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVa104`
5	`ExtVa105`
6	`ExtVa106`
7	`ExtVa107`
8	`ExtVa108`
9	`ExtVa109`
10	`ExtVa110`

The messages can be suppressed centrally via the faceplate or via `MsgLock` (program).

The free alarm inputs can be interconnected via the parameters `ExtMsg1` to `ExtMsg6`.

The associated values (`ExtVax`) of the message block can be assigned freely.

`MsgSup` is set if the `RunUpCyc` cycles have not finished running since the restart, `MsgLock = TRUE` or `MsgStat = 21`.

16.2.6 PolygonExt I/Os

Input parameters

Parameter	Description	Data type	Default
CancelOp *	Operator input CANCEL	BOOL	0
In *	Analog input value	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
In1 … In8 *	Input 1 ... 8	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#FF
In_Unit *	Unit of analog input value	INT	1001
MsgEvId	Message ID	DWORD	16#000000CF
Num *	Number of interpolation points 2 >= Num.Value <= 8	INT	8
OkOp *	Operator input OK	BOOL	0
Out1 … Out8 *	Interpolation point 1 ... 8	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#FF
Out_Ext	Reserved	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
Restart	Manual restart	BOOL	0
SubV_In	Substitute value	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#FF

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
Change *	1=Parameters changed	BOOL	0
ErrOut *	Parameterization failure (polygon or number of points error)	BOOL	0
GEIn1 … GEIn7	In is greater or equal In1 (and less than In2) ... In is greater or equal In7 (and less than In8)	BOOL	0
GEIn8	In is greater or equal In8	BOOL	0

Parameter	Description	Data type	Default
In1Out ... In8Out	Input 1 output ... Input 8 output	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
LIn1	In is less than In1	BOOL	0
MsgAckn	Message acknowledge	WORD	16#0000
MsgErr	Message error	BOOL	0
MsgStat	Message status	WORD	16#0000
MsgSup *	Message suppressed	BOOL	0
NumOut *	Number of interpolation points 2 >= Num.Value <= 8	INT	8
Out	Output	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
Out1Out ... Out8Out	Interpolation point 1 ... 8 output	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
Out_Unit *	Unit of analog output value	INT	1001
ParamFail *	Parameterization failure (polygon error)	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

16.2.7 PolygonExt block diagram

The block has no block diagram.

16.2.8 Operator control and monitoring

16.2.8.1 PolygonExt views

PolygonExt block views

The PolygonExt block provides the following views:

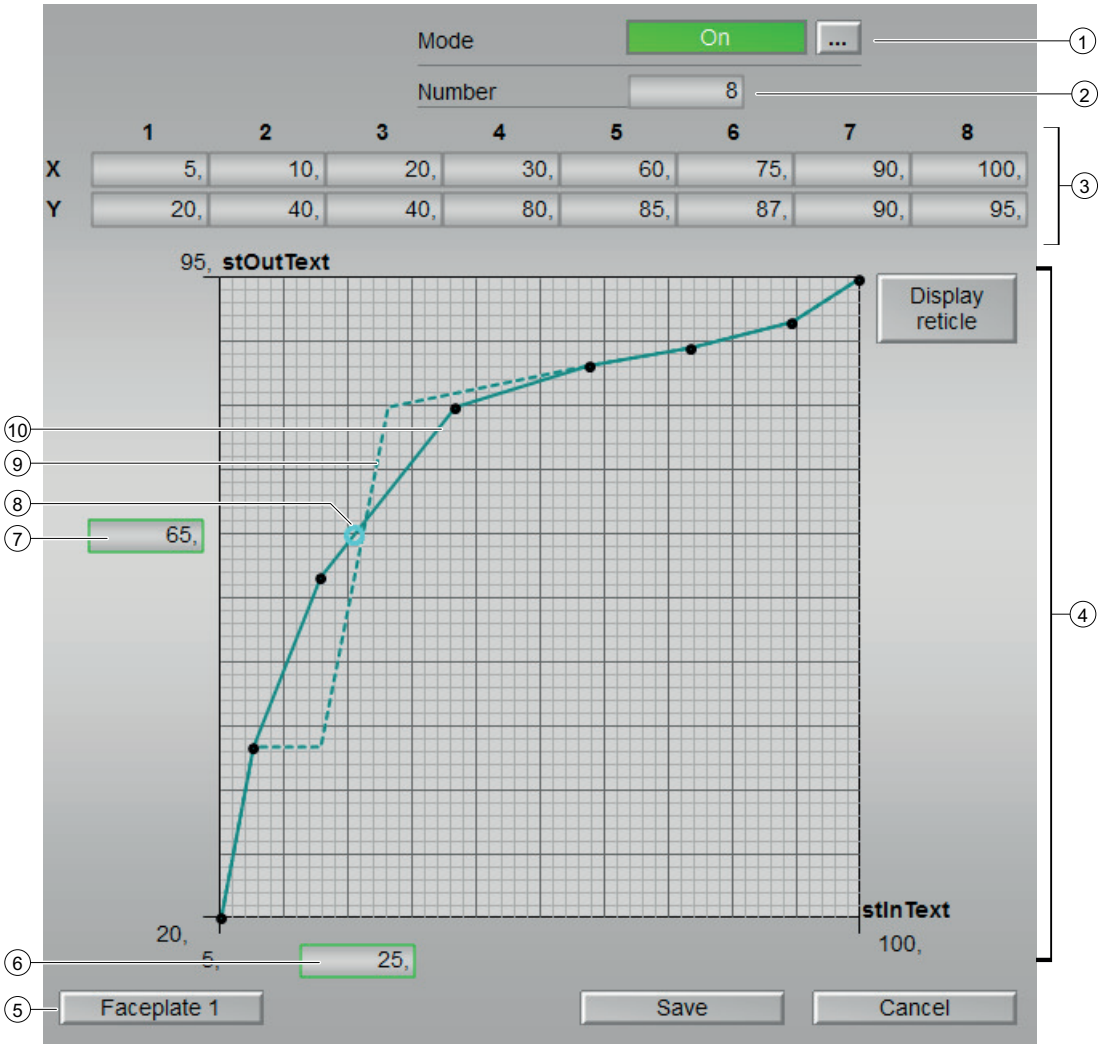
- PolygonExt standard view (Page 848)
- Message view
- PolygonExt parameter view (Page 850)
- PolygonExt preview (Page 851)
- Memo view

- Batch view
- PolygonExt block icon (Page 852)

You can find general information on the faceplate and block icon in the "Structure of the faceplate" and "Block icon structure" sections of the APL function manual.

16.2.8.2 PolygonExt standard view

PolygonExt standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

For more information on switching the operating mode, refer to the section "Switching operating states and operating modes" in the APL documentation.

(2) Display and change of the number of curve points

This area shows the current number of curve points.

Refer to the section "Changing values" in the APL manual for information about changing the value.

(3) Display and setting of curve points

Each field in the area represents a data point. You can reach all data points with the arrow keys left and right. If a data point is changed, the changed trend progression is displayed in the area (4) as a dashed line. However, this change only takes effect when the trend is transferred to the AS.

(4) Display of the current trend

The current trend progression is displayed in this area and the changed trend progression may be displayed (dashed line). The trend is loaded when you open the standard view but it is not continuously updated. In other words, if the trend were to be changed from a different OS, for example, these changes would not be visible until you open the standard view again.
By clicking the area, you set the display of the curve points (3) to the clicked area. The area is highlighted in the trend view.

(5) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

(6) Display of the input value "In"

The display shows the current input value. The position of the display depends on the current value.

(7) Display of the output value "Out"

The display shows the current output value.

(8) Current output value 'Out'

The blue circle represents the current output value. The position of the display depends on the current value.

(9) Trend not changed, values not yet applied

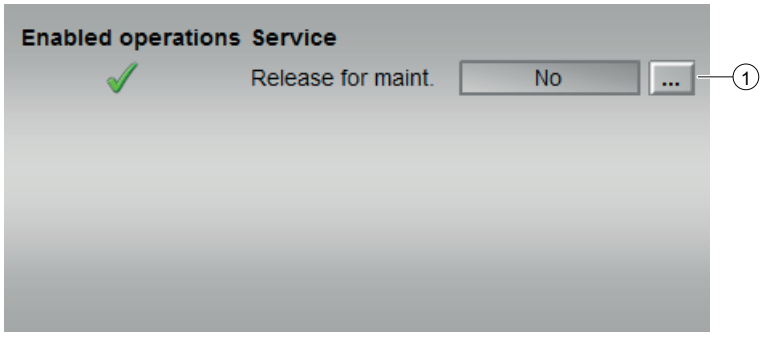
In this area, the changed trend progression not yet applied is shown as a dashed line.

(10) Current trend

The current trend is displayed as a line in this area.

16.2.8.3 PolygonExt parameter view

PolygonExt parameter view



(1) Service

You can select the following functions in this area:

- "Release for maintenance"

For more information, refer to the section Switching operating modes in the APL documentation.

You can find information on this area in the section:

- Release for maintenance

in the APL documentation

Enabled operations

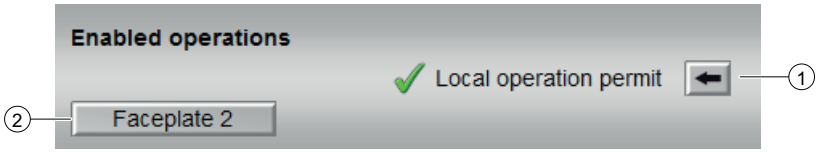
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

16.2.8.4 PolygonExt preview

PolygonExt preview



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- "Local operating permission": Use the ← button to change to the standard view of the OpStations block. You can find more information on this in the section "Operator control permissions" in the APL documentation.

(2) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

16.2.8.5 PolygonExt block icon

PolygonExt block icons

A variety of block icons are available with the following functions:

- Process tag type
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes
- Signal status, release for maintenance
- Memo display
- Process values

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

16.3 PolyCurve - Setpoint generator with variable number of curve points

16.3.1 PolyCurve description

Object name (type + number) and family

Type + number: FB 1156

Family: Control

Number of process objects: 1 PO

Functional principles

The PolyCurve block outputs a matching output value `Out` for an input value `In` defined by means of a curve. The curve is defined by a variable number of value pairs consisting of input and output values. The areas between the defined points are interpolated in a linear fashion.

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor. The block is also installed automatically in the startup OB (OB100).

Data block for setpoint storage

A data block can be used to extend the block with further setpoints.

The structure of the data block is as follows:

- First, a STRUCT is created in the data block. This structure then contains the following data:
  - One data block corresponds to 32 setpoints (value pairs of the input value and the output value). In the first declaration line of the structure (BlMax (INT)), enter the desired number of data blocks in the data block.
  - This is followed by a declaration line for the ID (DINT) of the record. This value is assigned by the block.
  - The ARRAY is next, in which the setpoints are later stored. The data area of the ARRAY is calculated as follows:
    Each setpoint consists of 2 REAL values (input value and output value). 2 REAL values per setpoint x 32 setpoints = 64 REAL values per data block.
    N = Number of data blocks.
    (64*n)-1
    Example calculation:
    Your curve has 64 setpoints. You have specified that your curve consists of 2 data blocks for BlMax. (1 data block = 32 setpoints).
    (64*2)-1=127
    ⇒ You therefore create a ARRAY of REAL with the range 0 .. 127.

A data block may contain several of these structures. A separate structure is required for each block with setpoint storage in the data block for every PolyCurve.

Table 16-1 Structure of the STRUCT in the data block for setpoint storage:

Name	Type	Comment
:		
BlMax	INT	Number of data blocks per curve (**n**)
ID	DINT	ID of the curve data
Daten	ARRAY of REAL	Data area: ARRAY of real values (size according to number of data blocks: (64*__n__)-1)
:		

Note

The actual values have to be set in addition to the initial values. These can be set via a tag table.

Adresse	Name	Typ	Anfangswert	Kommentar
0.0		STRUCT		
+0.0	PolyCurve1	STRUCT		
+0.0	NoOfBlk	INT	2	Number of Datasets
+2.0	ID	DINT	L#0	ID of curve
+6.0	Data	ARRAY[0..127]		
*4.0		REAL		
=518.0		END_STRUCT		
=518.0		END_STRUCT		

Figure 16-1 Data block for storing data records

Startup characteristics

Use the startup characteristics Feature bit 0 "Set startup characteristics" to define the startup characteristics for this block.

Depending on the setting of the Feature bits, the `DBPointer` is initialized during startup and the output values are set to the start value.

If `Feature.Bit2 = 0` (no data block), the input values are copied from `WDatInXX / WDatOutXX` to `RDatInXX /RDatOutXX` . (If no data block is used to save the curve, the data are saved in `RDatInXX/RDatOutXX`.)

The `RunUpCyc` parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Called blocks

- SFB35 (ALARM_8P)
- SFC6 (RD_SINFO)
- SFC20 (BLKMOV)

- SFC24 (TEST_DB)
- SFC260 (ChkREAL)
- SFC369 (SelST16)

Status word assignment for the parameter Status1

For a description of the individual parameters, refer to the section PolyCurve I/Os (Page 861).

Status bit	Parameter
0	Allocated
1	BatchEn
2	Not used
3	OosAct.Value
4	OosLi.Value
5	OnAct.Value
6 - 31	Not used

Status word assignment for the parameter Status2

For a description of the individual parameters, refer to the section PolyCurve I/Os (Page 861).

Status bit	Parameter
0	MsgLock = 1
1 - 30	Not used
31	MS_RelOp

Status word assignment for the parameter Status3

For a description of the individual parameters, refer to the section PolyCurve I/Os (Page 861).

Status bit	Parameter
0 - 31	Not used

Status word assignment for the parameter Status4

For a description of the individual parameters, refer to the section PolyCurve I/Os (Page 861).

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via EventTsIn
1	Effective signal 2 of the message block interconnected via EventTsIn
2	Effective signal 3 of the message block interconnected via EventTsIn
3	Effective signal 4 of the message block interconnected via EventTsIn
4	Effective signal 5 of the message block interconnected via EventTsIn
5	Effective signal 6 of the message block interconnected via EventTsIn
6	Effective signal 7 of the message block interconnected via EventTsIn

Status bit	Parameter
7	Effective signal 8 of the message block interconnected via `EventTsIn`
8 - 31	Not used

16.3.2 PolyCurve operating modes

The block can be operated using the following modes:

- On
- Out of service

General information on operating modes is available in the APL documentation.

16.3.3 PolyCurve functions

The functions for this block are listed below.

Reading of curve points

Each curve point can be modified in the standard view of the faceplate. The curve is automatically read and displayed when you open the standard view.

The OS read interface of the PolyCurve block is busy during the reading operation.

If another instance of the parameter window is opened during this period, the reading operation for this new instance does not commence until the reading operation for the first instance has been completed.

If the reading instance of the parameter window is closed before the reading operation has been completed, the AS block releases the interface automatically after the monitoring time (`RWTime`, preset to 3 seconds) has expired.

Writing of curve points

Curve points are entered in two steps:

- First, the curve points are entered (the new points are displayed as a dashed line; the trend in the AS is displayed as a solid line).
- Then the curve points with "Write to AS" are written to the AS. The writing interface is occupied during a writing operation (in the same way that the reading interface is occupied during a reading operation).

Note

If the window is closed during the writing operation, the writing operation will be terminated. In this case, the data will not be written completely.

You can also import the curve points by using the import function. The import file must be structured as follows:

- It must take the form of a text file with the extension .csv.
- In every row there has to be an input value (REAL) and an output value (REAL), separated by a semicolon. You can use a comma or a point as a decimal marker for the setpoint.
- The number of rows has to be at least equal to the number of curve points. If the file contains more points, only the number of points of the trend are imported.

The easiest method is to export a curve, modify it, and re-import it.

Using data blocks to save data records (`Feature`.Bit 2)

This block has the function "Use of data blocks to store data records".

The points for defining the curve be saved in the block itself (1 curve with 32 setpoints) or in a separate data block.

If the points are saved in a data block, it is possible to save almost any number of data points (depending on the size and definition of the data block) and this is only limited by the maximum possible size of a data block in the AS.
From a practical perspective, however, a curve should not consist of more than 10 data blocks, because the interfaces of the block can only transfer one data block at the same time (referring to the OS: one data block per second). This means that it will take at least 10 seconds for the data to be transferred if you wish to display a curve consisting of 10 data blocks on the OS.

The function is activated via `Feature`.Bit 2.

After activating the feature, you have to connect the `DBPointer` I/O to the data area of the relevant data block.

The structure of the data block is described in the "PolyCurve description (Page 853)" under the heading "Data block for setpoint storage".

Note

Interconnection of the data block

If only one structure is defined in the data block, the `DBPointer` I/O can be connected directly to the data block. If several structures are defined in the data blocks, the `DBPointer` I/O must be interconnected to the corresponding structure within the data block.

The interconnection succeeded in the CFC via the "Interconnection to operand..." command.

Limitation of the output value (`Feature`.Bit 5)

If the "Limitation of output value" function is activated, the `Out` output is limited to the range set in `Out_OpScale`. The limitation is generally only effective in conjunction with the "Linear extrapolation outside the defined range" function.

Substitute value switch in the event of an error (`Feature`.Bit 6)

If the function "Substitute value in case of error" is activated, the defined substitute value is output at the `Out` output in case of an incorrect calculation. A calculation error can occur if the function "Linear extrapolation outside defined range" is activated and the first or second or the next to last and the last defined point of a curve, for example, have the same input value (division by 0) and the function "Limitation of output value" is not activated.

Linear extrapolation outside defined range (`Feature`.Bit 7)

If the feature "Linear extrapolation outside defined range" is not activated, the block outputs the output value of the last defined point when the defined input range is exceeded and the value of the first defined point when the defined input range is undershot.

If the function is activated, the gradient is determined from the first and second or last and next to last point and the curve is continued in a linear fashion.

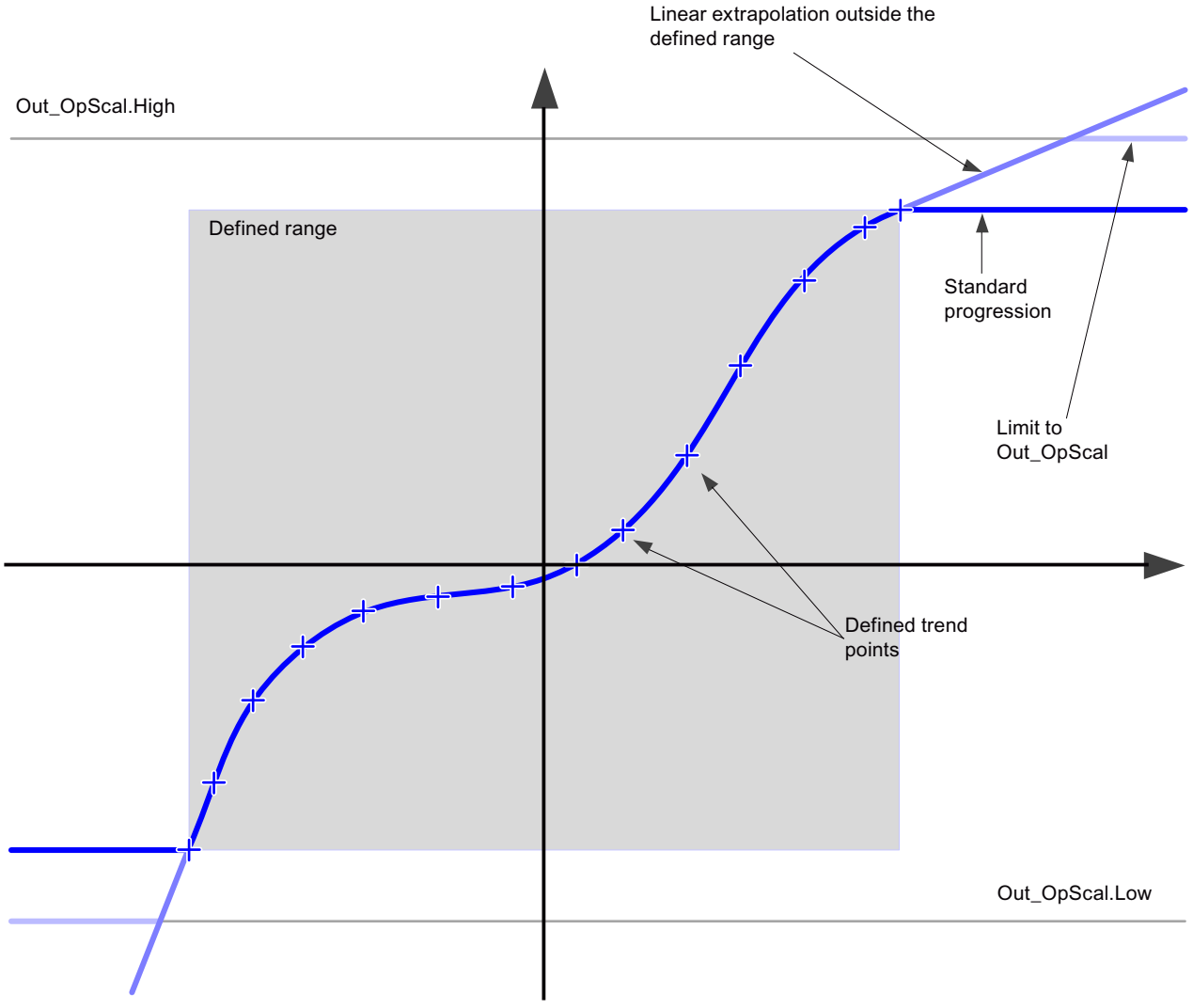


Figure 16-2 Representation of the effect of Feature bits on the curve:

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "OOS" mode
4 - 6	Not used
7	1 = Operator can import a curve
8	1 = Operator can export a curve
9	1 = Operator can modify a curve
10	1 = Operator can copy a curve to the AS
11	1 = Operator can refresh chart
12	1 = Operator can activate the Maintenance Release function
13 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
2	Use of data block for storing curve data
5	See APL function: "Limit output Out"
6	See APL function: "Substitute value switch in the event of an error"
7	Linear extrapolation outside defined range
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"

Configurable reactions using the `Feature2` parameter

Bit	Function
0 - 31	Not used

16.3.4 PolyCurve error handling

Since the block does not handle process values, it only recognizes two errors:

- `MsgErr`
  Generated by ALARM_8P and indicates that a message error has occurred.
- `ErrorNum`
  Displays the current error number. The following numbers are currently possible:

Overview of error numbers

The `ErrorNum` I/O can be used to output various error numbers:

Error number	Meaning of the error number
00	No error
30	Corrupt In value (#Nan or #Inf)
41	Data block error
49	Calculation error

16.3.5 PolyCurve messaging

Message behavior

The PolyCurve block uses the ALARM_8P block for generating messages.

The messages can be suppressed centrally via the faceplate or via `MsgLock` (program).

The free alarm inputs can be interconnected via the parameters `ExtMsg1` to `ExtMsg3`.

The associated values (`ExtVaXXX`) of the message block can be assigned freely.

`MsgSup` is set if the `RunUpCyc` cycles have not finished running since the restart, `MsgLock = TRUE` or `MsgStat = 21`.

Process messages

Message instance	Message ID	Message class	Event
MsgEvId	SIG 1	AS control system fault	$$BlockComment$$ External error
	SIG 2	AS control system fault	$$BlockComment$$ External message 1
	SIG 3	AS control system fault	$$BlockComment$$ External message 2
	SIG 4	AS control system fault	$$BlockComment$$ External message 3
	SIG 5 - 8	-	-

Associated values for the message instance `MsgEvId`

Associated value	Block parameters
1	BatchName
2	StepNo
3	BatchID
4	ExtVa104
5	ExtVa105
6	ExtVa106
7	ExtVa107
8	ExtVa108

16.3.6 PolyCurve I/Os

Input parameters

Parameter	Description	Data type	Default
DBPointer		ANY	
In	In	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
In_OpScale	In - bar display limits for OS	STRUCT: • High: REAL • Low: REAL	- • 100.0 • 0.0
In_Unit	Unit of In	INT	1001
MsgEvId	Message event ID	DWORD	16#00000228
Out_OpScale	Output limits for OS and Out.Value	STRUCT: • High: REAL • Low: REAL	- • 100.0 • 0.0
Out_Unit	Unit of Out	INT	1001
RDBlNo	Read data block number	INT	0
RWTime	OS data read/write watchdog time [s]	REAL	5.0
SubV_In	Substitute value	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
WDatIn00 … 31	OS write data point	REAL	0.0 ... 310.0
WDatOu00 … 31	OS write data point	REAL	0.0
WDBlNo	Write data block number	INT	0

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
MaxBlNo	Number of blocks per data record	INT	1
MsgAckn	ALARM_8P: ACK_STATE output	WORD	16#0000
MsgErr	1=Messaging error occurs	BOOL	0
MsgStat	ALARM_8P: Status output	WORD	16#0000
Out	Output	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
RDatID	ID of read curve (data record)	DINT	0
RDatIn00 ... 31	OS read data point	REAL	0.0
RDatOu00 ... 31	OS read data point	REAL	0.0
RDBlNoOut	Block number of current write data	INT	0
WDBlNoOut	Block number of current write data	INT	0

16.3.7 PolyCurve block diagram

A block diagram is not provided for this block.

16.3.8 Operator control and monitoring

16.3.8.1 PolyCurve views

PolyCurve block views

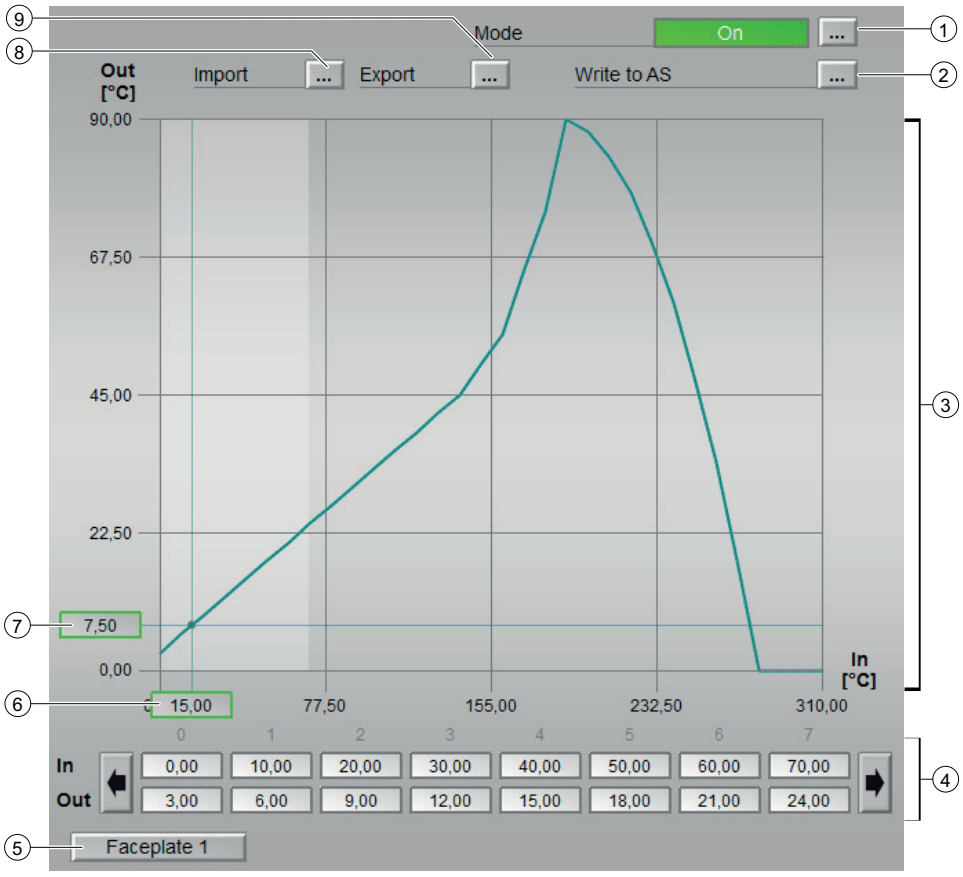
The PolyCurve block provides the following views:

- PolyCurve standard view (Page 863)
- Message view
- PolyCurve parameter view (Page 865)
- PolyCurve preview (Page 866)
- Memo view
- Batch view
- PolyCurve block icons (Page 867)

You can find general information on the faceplate and block icon in the "Structure of the faceplate" and "Block icon structure" sections of the APL function manual.

16.3.8.2 PolyCurve standard view

PolyCurve standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

For more information on switching the operating mode, refer to the section "Switching operating states and operating modes" in the APL documentation.

(2) Write trend in the AS

The trend shown in the area (3) can be changed via the fields (4). The modified trend is a dashed line. You trigger loading of the modified trend to the AS with the 'Write to AS' function.

(3) Display of the current trend

The current trend progression is displayed in this area and the changed trend progression may be displayed (dashed line). The trend is loaded when you open the standard view but it is not continuously updated. In other words, if the trend were to be changed from a different OS, for example, these changes would not be visible until you open the standard view again. Clicking the area sets the display of the curve points (4) to the clicked area. The area is highlighted in the trend view.

(4) Display and setting of curve points

Each field in the area represents a data point. You can reach all data points with the arrow keys left and right. If a data point is changed, the changed trend progression is displayed in the area (3) as a dashed line. But this change only becomes effective when you write the trend to the AS (2).

(5) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

(6) Display of the input value "In"

The display shows the current input value. The position of the display depends on the current value. The current intersection of the input and output values is displayed with a point on the displayed trend.

(7) Display of the output value "Out"

The display shows the current output value. The position of the display depends on the current value. The current intersection of the input and output values is displayed with a point on the displayed trend.

(8) Import trend

Use the 'Import trend' function to import the trend from a .csv file. The "IL File Dialog" is used for this purpose.

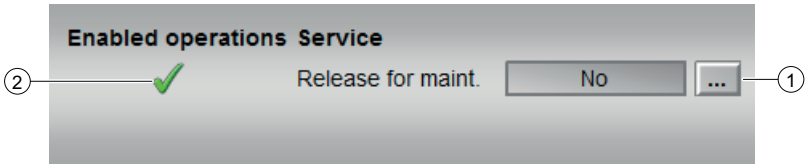
You can find more detailed information on this in the section for "File dialog (Page 56)".
The imported trend is then displayed in the area (3) as a dashed line and can be written to the AS block with the function (2) "Write trend to AS".

(9) Export trend

Use the 'Export trend' function to export the current trend to a .csv file. If there is a current trend and a modified trend (dashed line), the current trend is exported.

16.3.8.3 PolyCurve parameter view

PolyCurve parameter view



(1) Service

You can select the following functions in this area:

- "Release for maintenance"

For more information, refer to the section Switching operating modes in the APL documentation.

You can find information on this area in the section:

- Release for maintenance

in the APL documentation

(2) Operator control enable

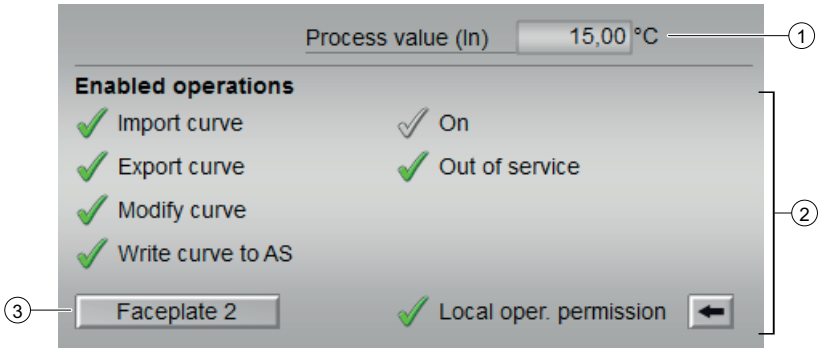
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

16.3.8.4 PolyCurve preview

PolyCurve preview



(1) Preview area

This area displays the real process value (In).

(2) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- "Import trend": You may import trends
- "Export trend": You may export trends
- "Change trend": You can change the trend
- "Write trend to AS": You may write the trend to AS
- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to change to the standard view of the OpStations block. You can find more information on this in the section "Operator control permissions" in the APL documentation.

(3) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

16.3.8.5 PolyCurve block icons

PolyCurve block icons

A variety of block icons are available with the following functions:

- Process tag type
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes
- Signal status, release for maintenance
- Memo display
- Process values

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

16.4 SPCurve - Setpoint scheduler with a variable number of setpoints

16.4.1 SPCurve description

Object name (type + number) and family

Type + number: FB 1141

Family: Control

Number of process objects: 1 PO

Operating principle of SPCurve

The SPCurve block outputs a setpoint along a curve as a function of a time. The curve is defined by a variable number of setpoints (pairs of values from time and value).

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor. The block is also installed automatically in the startup OB (OB100).

Data block for setpoint storage

A data block can be used to extend the block with further setpoints.

The structure of the data block is as follows:

- A curve corresponds to a data record. In the first declaration line, specify the number of data records (curves) as the start value for SeMax1 (INT) in the data block.
- A data block of a data record corresponds to 32 setpoints (value pairs of value and time). In the second declaration line, specify the desired number of data blocks per data record (curve) as the start value for BIMax1 (INT) in the data block.
- For each data record, there is now a declaration line for the ID (DINT) of the records. This value is assigned by the block.
- In next line contains a declaration line for each data block with the data type of the provided UDT "SPDatablock". The setpoints are stored later.
  Example calculation:
  Your curve has 64 setpoints. You want to create different 3 curves.
  ⇒ You therefore create 6 declaration lines. (64 setpoints = 2 data blocks, 3 curves with 2 data blocks each = 6 data blocks)

The data block DB26 "SPCurveDat" is supplied as a template in the library. UDT 8 "SPDatablock" is also in the library.

Table 16-2 Structure of the data block for setpoint storage:

Designation	Data type	Comment
SeMax1	INT	Number of curves or data records (m)
BlMax1	INT	Number of data blocks per curve (n)
Dat_1_ID	DINT	ID of data record 1
Dat_2_ID	DINT	ID of data record 2
Dat_m_ID	DINT	ID of data record m
Dat_1_1	SPDatablock	1st data block of the 1st data record.
Dat_1_2	SPDatablock	2nd data block of the 1st data record.
Dat_1_n	SPDatablock	nth data block of the 1st data record.
Dat_2_1	SPDatablock	1st data block of the 2nd data record.
Dat_2_2	SPDatablock	2nd data block of the 2nd data record.
Dat_2_n	SPDatablock	nth data block of the 2nd data record.
Dat_m_1	SPDatablock	1st data block of the mth data record.
Dat_m_1	SPDatablock	2nd data block of the mth data record.
Dat_m_n	SPDatablock	nth data block of the mth data record.

Startup characteristics

Use the startup characteristics Feature bit 0 "Set startup characteristics" to define the startup characteristics for this block.

Depending on the setting of the Feature bits, the DBPointer is initialized during startup and the output values are set to the initial value.

If Feature Bit2 = 0 (no data block), the input values are copied from WDatVaXX_Op/ WDatTiXX_Op to RDatVaXX_Op/RDatTiXX_Op. (If no data block is used to save the trend, the data are saved in RDatVaXX_Op/RDatTiXX_Op.) The data from WDatVaXX_Op/WDatTiXX_Op is synchronized with this during operation so that the data can be saved using "parameter readback".)

The RunUpCyc parameter can be used to set for how long (number of cycles) messages are to be suppressed.

Called blocks

FB13	TIME_BEG
SFB35	ALARM_8P
SFC6	RD_SINFO
SFC20	BLKMOV

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section SPCurve I/Os (Page 878).

Status bit	Parameter
0	Allocated
1	`BatchEn`
2	Not used
3	`OosAct.Value`
4	`OosLi.Value`
5	`AutAct.Value`
6	Not used
7	`ManAct.Value`
8	`SP_Trk.Value`
9	`ForceOff.Value`
10	Not used
11	`Run.Value`
12 - 13	Not used
14	Invalid control signal
15	Switchover failed
16	Not used
17	1 = Step trend
18	1 = AS time used
19	1 = Display Time to next setpoint
20	1 = `Input PV` is not interconnected (RbkOut.ST = 16#FF)
21	1= `SimOn`
22 - 28	Not used
29	`MS_RelOp`
30 - 31	Not used

Status word assignment for the parameter `Status2`

For a description of the individual parameters, refer to the section SPCurve I/Os (Page 878).

Status bit	Parameter
0	`MsgLock = 1`
1 - 30	Not used
31	`MS_RelOp`

Status word assignment for the parameter `Status3`

For a description of the individual parameters, refer to the section SPCurve I/Os (Page 878).

Status bit	Parameter
0 - 25	Not used
26	Show automatic preview in the standard view

Status bit	Parameter
27	Not used
28	GrpErr.Value
29	RdyToStart.Value
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word assignment for the parameter Status4

For a description of the individual parameters, refer to the section SPCurve I/Os (Page 878).

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via EventTsIn
1	Effective signal 2 of the message block interconnected via EventTsIn
2	Effective signal 3 of the message block interconnected via EventTsIn
3	Effective signal 4 of the message block interconnected via EventTsIn
4	Effective signal 5 of the message block interconnected via EventTsIn
5	Effective signal 6 of the message block interconnected via EventTsIn
6	Effective signal 7 of the message block interconnected via EventTsIn
7	Effective signal 8 of the message block interconnected via EventTsIn
8...31	Not used

16.4.2 SPCurve operating modes

The block can be operated using the following modes:

- Automatic mode
- Manual mode
- Out of service

The next section provides additional block-specific information relating to the general descriptions.

"Automatic" mode

General information on "Automatic mode" and for switching between operating modes is available in the section *Operating modes of the blocks* in the *APL Function manual*.

In "Automatic mode" you can:

- "Start" (StartAut = 1)
- "Stop" (StopAut = 1)
- "Reset time" (TiResAut = 1)

"Manual" mode

General information on "Manual mode" and for switching between operating modes is available in the section *Operating modes of the blocks* in the *APL Function manual*.

In "Manual mode" you can:

- "Run" (`StartMan` = 1)
- "Stop" (`StopMan` = 1)
- "Reset time" (`TiResMan` = 1)

Run/Stop

When the status is "Run", the setpoint calculated on the basis of the time value from the setpoint curve is output.
When the status is "Stop", the last setpoint is retained, but can be changed by the operator or by a higher-level automatic system. The setpoint is restricted to the limits "SP_OpScale.High" and "SP_OpScale.Low".
If the block time is used ("Block time/AS time" feature), the current time of the block can be changed or reset by an operator or a higher-level automatic system.

"Out of service"

General information on "Out of service" mode is available in the section *Operating modes of the blocks* in the *APL Function manual*.

16.4.3 SPCurve functions

The functions for this block are listed below.

Display of the process value

The process value is only displayed in the faceplate, if it has been interconnected at the AS block. The block detects the interconnection automatically based on the quality code. If the quality code is "16#FF", the I/O is not interconnected; in every other case, the I/O is interconnected.

The OS contains corresponding block icons both with and without display of the process value. The block icon is not selected automatically, but has to be specified during the configuration phase.

Using data blocks to save data records (`Feature`.Bit 2)

This block has the function "Use of data blocks to store data records".

The points for defining the setpoint curve can be saved in the block itself (1 curve with 32 setpoints) or in a separate data block.

If the points are saved in a data block, it is possible to save almost any number of data points (depending on the size and definition of the data block) and this is only limited by the maximum possible size of a data block in the AS.

From a practical perspective, however, a curve should not consist of more than 10 data blocks, because the interfaces of the block can only transfer one data block at a time (in relation to the OS, one data block per second). This means that it will take at least 10 seconds for the data to be transferred if you wish to display a curve consisting of 10 data blocks on the OS.

The function is activated via `Feature`.Bit 2.

Copy your data block or the template DB26 "SPCurveDat" and the UDT8 "SPDatablock" from the library into the project and connect the `DBPointer` I/O to the data block.

The structure of the data block is described in the "SPCurve description (Page 868)" under the heading "Data block for setpoint storage".

Note

Interconnection of the data block

The `DBPointer` I/O is connected directly to the data block.

The interconnection succeeded in the CFC via the "Interconnection to Operand..." command.

Block time or AS time (`Feature`.Bit 3)

The SPCurve block can either use the time calculated from a start time (block time) or the AS time as the basis for determining the current setpoint from the curve. The function is activated via `Feature`.Bit 3.

If the block time is used, the operator can set the time to a certain value while the block status is set to "Stop".

Linear curve or step curve (`Feature`.Bit 4)

The SPCurve block can extrapolate the setpoint values between two data points linearly (linear curve) or can keep outputting a setpoint value until the next setpoint is reached (step curve). The function is activated via `Feature`.Bit 4.

Displaying the next setpoint (`Feature`.Bit 5)

If the linear curve is used, the information about when the next setpoint is due and the nature of its value is not relevant. However, if the step curve is used, this information may be of relevance.

If the "View of next setpoint" function is activated, the value of the next setpoint and the time until the next setpoint are displayed in the standard view of the faceplate. The function is activated via `Feature`.Bit 5.

Changing the active trend

The changing of the active trend can be initiated by an operator or by the higher-level automatic system. For this to happen, the block status must be "Stop" and there must be more than one trend available.
When changing or starting the active trend, the block checks that the time sequence for all setpoints is correct and makes any necessary corrections. To do this, it follows the procedure below:

1. The time value of the first setpoint is set to 0.
2. The next point is checked. If the time value is smaller than 0, it is set to 0.
3. The time value of the checked setpoint is stored.
4. The time value of the next point is compared with the stored time value. If the next time value is smaller than the stored value, then it is set to the stored value.
5. After that, the procedure is repeated from step 3 onwards until all points of the active trend have been checked.

The setpoints of the individual setpoints are not checked or changed. The output setpoint is restricted to the limits "`SP_OpScale.High`" and "`SP_OpScale.Low`".

Reading and writing of setpoints (AS)

The AS (higher-level automatic system) can read and write setpoints. For the setpoints to be read and written, the block features both read and write interfaces in the AS and OS directions.

The block can simultaneously read (on the AS and OS interface) and write. Either the AS or the OS is able to write setpoints. You can switch between the two using the `DatExtLi` control input. If `DatExtLi = 1`, the AS writes, otherwise the OS

Setpoints can only be read and written by the AS on a block-by-block basis.

Reading of setpoints/data blocks (AS)

Setpoints / data blocks are read via the "external" data interface as follows:

- `RDSeNoEx` is set to the desired trend number.
  - During the next cycle, the block makes the required data available at the `RDatTi00_Li`... `RDatTi31_Li/RDatVa00_Li`...`RDatVa31_Li` outputs and indicates that this has been done by setting `RDBlNoExOut = RDBlNoEx` and `RDSeNoExOut = RDSeNoEx`. The reading interface data is always valid if `RDSeNoExOut` and `RDBlNoExOut` are set to a value other than 0.
  - For example:Suppose `DBPointer` is connected to DB26. From DB26, the first trend and the second structure with the 32 value pairs is written to the outputs. Set `RDSeNoEx` =1 and `RDB1NoEx` =1 and `Feature.Bit2` =1. The second part from trend 1 is written to the outputs. In the delivered sample DB26 (2 trends 3 structures) the following combinations are possible: `RDSeNoEx`, `RDBlNoEx` (1,1-1,2-1,3) (2,1-2,2-2,3).
- `RDBlNoEx` is set to the desired data block number.

Writing of setpoints/data blocks (AS)

The process for writing setpoints / data blocks is similar to the process for reading them. Writing is possible when the block status is "Stop" or when the trend being written to is inactive. The writing operation is performed as follows:

- `DatExtLi` is set to "true".
- `WDBlNoEx` is set to the desired data block number.
- `WDSeNoEx` is set to the desired trend number.
  - The block takes the data from the `WDatTi00_Li…WdatTi31/WDatVa00_Li… WdatVa31_Li` interface and writes it to the data block or stores the data. The block indicates that the data has been transferred successfully by setting `WDBlNoExOut = WDBlNoEx` and `WDSeNoExOut = WDSeNoEx` .

Reading of setpoints (OS)

Each setpoint can be changed in the parameter view of the faceplate.

A setpoint can only be changed when the block status is "Stop" or when the trend to be changed is not active.

The trend is automatically read and displayed when you open the parameter view.

The OS read interface of the SPCurve block is busy during the reading operation. If another instance of the parameter window is opened during this period, the reading operation for this new instance does not commence until the reading operation for the first instance has been completed. If the reading instance of the parameter window is closed before the reading operation has been completed, the AS block releases the interface automatically after the monitoring time (`RWTime`, preset to 3 seconds) has expired.

Writing of setpoints (OS)

Setpoints are entered in two steps:

1. First, the setpoints are entered (the new points are displayed as a dashed line; the trend in the AS is displayed as a solid line).
2. Then, the setpoints are written to the AS using "Change trend". The writing interface is occupied during a writing operation (in the same way that the reading interface is occupied during a reading operation).

Note

If the parameter window is closed during the writing operation, the writing operation will be terminated. In this case, the data will not be written completely.

You can also import the setpoints by using the import function. The import file must be structured as follows:

- It must take the form of a text file with the extension .csv.
- In every row there has to be a time value (double integer) and a setpoint (REAL), separated by a semicolon. You can use a comma or a point as a decimal marker for the setpoint.
- The number of rows has to be at least equal to the number of setpoints. If the file contains more points, only the number of setpoints of the trend are imported.

The easiest method is to export a curve, modify it, and re-import it.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Reserved
1	1 = Operator can switch to "Manual" mode
2	1 = Operator can switch to "Automatic" mode
3	1 = Operator can switch to "Out of service" mode
4	1 = Operator can switch to "Start"
5	1 = Operator can switch to "Stop"
6	1 = Operator can reset time
7	1 = Operator can set time
8	1 = Operator can set active trend
9	1 = Operator can change trend data
10	1 = Operator can set setpoint
11	1 = Operator can set "Simulation"
12	1 = Operator can activate the Release for maintenance function
13	1 = Operator can set `SimPV`
14 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
2	1 = Use of data blocks to save data records
3	1 = AS time is used to calculate setpoint
4	1 = Step trend

Bit	Function
5	1 = Display time to next setpoint
6	See APL function: "Set switch or button mode"
17	See APL function: "Bumpless switchover to 'Automatic' mode for valves, motors, and dosers"
21	See APL function: "First-in detection response to deactivation"
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"

16.4.4 SPCurve error handling

The following errors are displayed for this block:

- `MsgErr`
  Generated by ALARM_8P and indicates that a message error has occurred.
- `ErrorNum`
  Displays the current error number.

Overview of error numbers

The following numbers are currently possible:

Error number	Meaning of the error number
00	No error
41	Data block error The data block error is displayed in the block if the `Feature.Bit2` is set to 1 and no data block or an incorrectly configured data block is connected to the block.
51	Faulty control from higher-level automatic system (e.g. Run and Stop signal at the same time)

16.4.5 SPCurve messaging

Message behavior

The SPCurve block uses the ALARM_8P block for generating messages.

The messages can be suppressed centrally via the faceplate or via `MsgLock` (program).

The free alarm inputs can be interconnected via the parameters `ExtMsg1` to `ExtMsg3`.

The associated values (`ExtVaXXX`) of the message block can be assigned freely.

`MsgSup` is set if the `RunUpCyc` cycles have not finished running since the restart, `MsgLock = TRUE` or `MsgStat = 21`.

Process messages

Message instance	Message ID	Message class	Event
MsgEvId	SIG 1	AS control system fault	$$BlockComment$$ External error
	SIG 2	AS control system fault	$$BlockComment$$ External message 1
	SIG 3	AS control system fault	$$BlockComment$$ External message 2
	SIG 4	AS control system fault	$$BlockComment$$ External message 3
	SIG 5	-	-
	SIG 6	-	-
	SIG 7	-	-
	SIG 8	-	-

Associated values for the message instance MsgEvId

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVa104`
5	`ExtVa105`
6	`ExtVa106`
7	`ExtVa107`
8	`ExtVa108`

16.4.6 SPCurve I/Os

Input parameters

Parameter	Description	Data type	Default
`AutModLi`	1=Auto Mode: Auto Mode by Linked or SFC	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`AutModOp`	1=Auto Mode: Auto Mode by Operator	BOOL	0
`BatchEn`	Enable Remote Operation of Controller by Batch Recipe	BOOL	0
`BatchID`	Batch ID	DWORD	16#00000000
`BatchName`	Name of the Batch recipe	STRING[32]	''

Parameter	Description	Data type	Default
CSF		STRUCT • Value:BOOL • ST: BYTE	 • 0 • 16#80
DatExtLi	Linkable input to select external data interface	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
DBPointer		ANY	
EN		BOOL	1
EventTsIn	For interconnecting data between a technology block and the message blocks EventTs, Event16Ts.The EventTsIn input parameter serves to interconnect the EventTsOut output parameterof the EventTs, Event16Ts block. When this interconnection is configured, the messages of the EventTs, Event16Ts block are displayed on the OS in the alarm view of the technology block and can also be acknowledged there.	ANY	
ForceStop	Linkable input for forced stop	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MsgEvId	Message event ID	DWORD	16#000000D0
RDBlNo	Read data block number	INT	0
RDBlNoEx	External read data block number	INT	0
RDSeNo	Read data record number	INT	0
RDSeNoEx	External read data record number	INT	0
RWTime	OS data read/write watchdog time [s]	REAL	5.0
SP_OpScale	SP limits for OS and SP(out)	STRUCT • High: REAL • Low: REAL	- • 100.0 • 0.0
SP_SeAut	Linkable input to select SP trend (data record)	INT	1
SP_SeMan	Operator input to select SP trend (data record)	INT	1
SP_Unit	Unit of SP	INT	1001
SPMan	Operator input for setting current SP	REAL	0.0
SPTrk	Linkable input for setting current SP	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SPTrkVal	Linkable input for setting current SP	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
StartAut	1=Start: Start command in auto mode	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
StartMan	1=Start: Start command in auto mode	BOOL	0
TiAut	Linkable input for setting current time value	DINT	0
TiMan	Operator input for setting current time	DINT	0
TiResAut	Linkable input to reset time (set time to 0)	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
TiResMan	Operator input to reset time (set time to 0)	BOOL	0
TiSetAut	Linkable input for setting current time value	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
WDatTi00_Li ... WDatTi31_Li	External write data setpoint time	DINT	0
WDatTi00_Op ... WDatTi31_Op	OS write data setpoint time	DINT	0 ... 310 (in increments of ten)
WDatVa00_Li ... WDatVa31_Li	External write data setpoint value	REAL	0.0
WDatVa00_Op ... WDatVa31_Op	OS write data setpoint value	REAL	0.0
WDBlNo	Write data block number	INT	0
WDBlNoEx	External write data block number	INT	0
WDSeNo	Write data record number	INT	0
WDSeNoEx	External write data record number	INT	0

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
CurrPointNum	Number of current setpoint	INT	0
DatExtAct	• 1: External setpoint is active • 0: Internal setpoint is active	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
MaxBlNo	Number of blocks per data record	INT	1
MaxSeNo	Number of data records	INT	1
MsgAckn	ALARM_8P: ACK_STATE output	WORD	16#0000
MsgErr	1=Messaging error occurs	BOOL	0
MsgStat	ALARM_8P: Status output	WORD	16#0000

Parameter	Description	Data type	Default
RDatID_Li	ID of read curve (data record)	DINT	0
RDatID_Op	ID of read curve (data record)	DINT	0
RDatTi00_Li ... RDatTi31_Li	External read data setpoint time	DINT	0
RDatTi00_Op ... RDatTi31_Op	OS read data setpoint time	DINT	0
RDatVa00_Li ... RDatVa31_Li	External read data setpoint value	REAL	0.0
RDatVa00_Op ... RDatVa31_Op	OS read data setpoint value	REAL	0.0
RDBlNoExOut	Block number of current read data	INT	0
RDBlNoOut	Block number of current read data	INT	0
RDSeNoExOut	Set number of current read data	INT	0
RDSeNoOut	Set number of current read data	INT	0
Run	1 = Run: SPCurve is running	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SP_Next	Next setpoint	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SP_Prep	Prepared setpoint	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
SP_SeOut	Number of current SP trend (data record)	INT	0
TimeCurr	Current time	DINT	0
TimeHigh	Next setpoint time	DINT	0
TimeLow	Last setpoint time	DINT	0
TimeMax	Last setpoint time	DINT	0
TimeNext	Time left to next setpoint time	DINT	0
WDBlNoExOut	Block number of current write data	INT	0
WDBlNoOut	Block number of current write data	INT	0
WDSeNoExOut	Set number of current write data	INT	0
WDSeNoOut	Set number of current write data	INT	0

16.4.7 SPCurve block diagram

The block has no block diagram.

16.4.8 Operator control and monitoring

16.4.8.1 SPCurve views

SPCurve block views

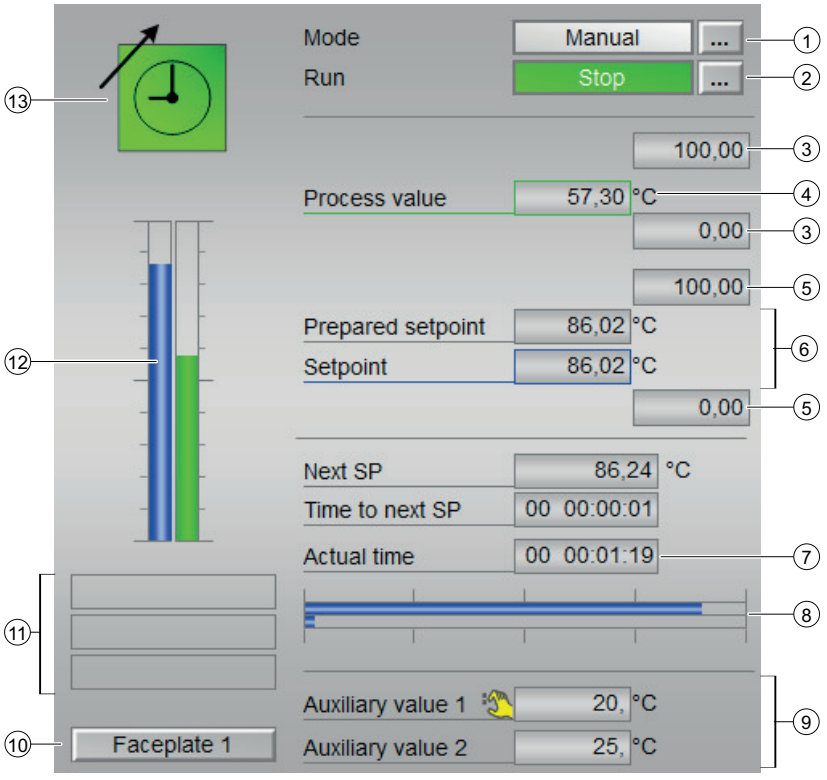
The SPCurve block provides the following views:

- SPCurve standard view (Page 883)
- Message view
- Trend view
- SPCurve parameter view (Page 886)
- SPCurve preview (Page 888)
- Memo view
- Batch view
- SPCurve block icons (Page 889)

You can find general information on the faceplate and block icon in the "Structure of the faceplate" and "Block icon structure" sections of the APL function manual.

16.4.8.2 SPCurve standard view

SPCurve standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(2) Display and switchover of the operating mode

This area shows the currently valid operating state. The following operating states can be displayed:

- "Run"
- "Stop"

You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(3) High and low scale range for the process value

These values provide information on the display range for the bar graph of the process value. The scale range is specified in the engineering system.

(4) Display of the process value including signal status

This area shows the current process value with the corresponding signal status.

(5) High and low scale range for the setpoint

This area is already set and cannot be changed.

(6) Display and change of the setpoint including signal status

This area shows the current setpoint with the corresponding signal status and trend setpoint.

If text is configured for this command, it is displayed as additional text and button label for command selection.

You can find more information on this in the section "Labeling of buttons and text" . You can change the text for the process value at the `PV_Out` parameter.

(7) Preview

This area shows the current run time, the remaining time until the next setpoint and the next setpoint.

(8) Bar graph display for timers

This area shows the following values in the form of a bar graph:

- Upper bar: Representation of the time to the next value.
- Lower bar: Representation of the elapsed time compared to total time.

The visible area in the bar graph depends on the configuration in the engineering system (ES).

(9) Display of auxiliary values

This display is only visible when the corresponding block input is interconnected.

You can use this area to display two auxiliary values that have been configured in the engineering system (ES). You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(10) Navigation button for switching to the standard view of any faceplate

This display is only visible when the corresponding block input is interconnected.

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(11) Display area for block states

This area provides additional information on the operating state of the block:

Field 1:

- Forced tracking
- Tracking
- Optimizing

Field 2:

- Data error
- Invalid signal
- Switchover error

You can find more information on this in the sections "Monitoring the feedbacks" and "Error handling" (subsections "Invalid input signals" as well as "Mode switchover error") in the manual of the APL

Field 3:

- Maintenance

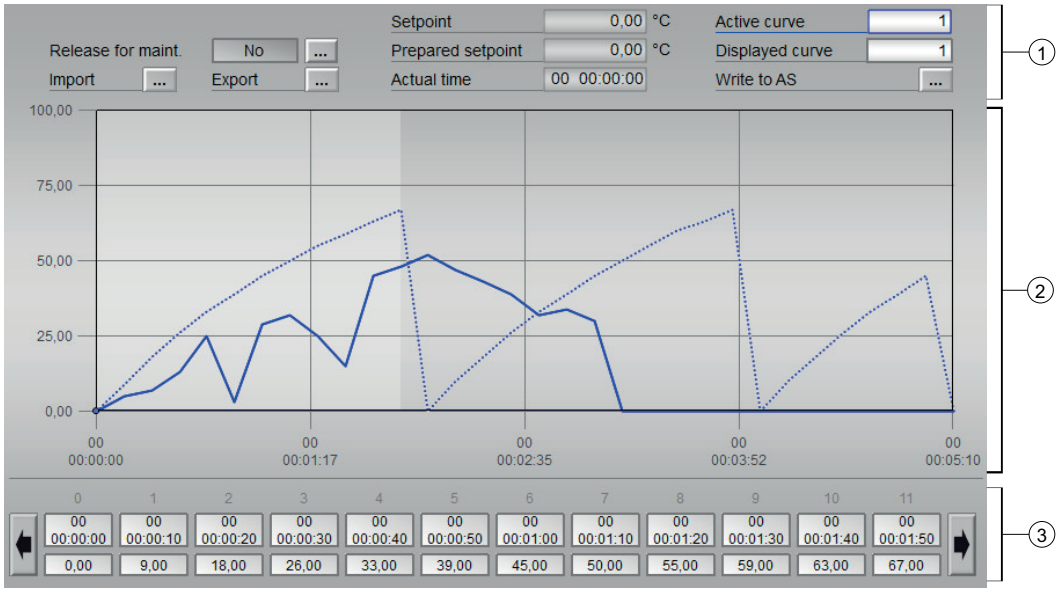
You can find additional information on this in the section "Release for maintenance" in the manual of the APL.

(12) bar graph for "Process value" and "Setpoint"

This area shows the current "Process value" and "Setpoint" in the form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

16.4.8.3 SPCurve parameter view

SPCurve parameter view



(1) Trend properties and settings

- Release for maintenance
  You can find information on this area in the "Release for maintenance" section in the APL documentation.
- Import trend:
  Use the 'Import trend' function to import the trend from a .csv file. The "IL File Dialog" is used for this purpose. You can find more detailed information on this in the documentation for the "IL File Dialog (Page 56)".
  The imported trend is then displayed in the area (2) as a dashed line and can be written to the AS block with the function "Write to AS".
- Export trend:
  Use the 'Export trend' function to export the current trend to a .csv file. If there is a current trend and a modified trend (dashed line), the current trend is exported.
- Setpoint:
  Setpoint currently output by the block
- Trend setpoint:
  Current setpoint according to trend
- Current time:
  Currently expired time / current AS time (depending on Feature.Bit3)
- Active trend:
  Selection of the trend that is executed in the AS

- Displaced trend:
  Selection of the trend displayed under 2
- Write to AS:
  You trigger loading of the modified trend to the AS with the "Write to AS" function.

(2) Display of the current trend

The trend progression of the "Displayed trend" (1) is displayed in this area and the changed trend progression may be displayed (dashed line). The changed progression is not applied until "Write to AS".

If the "Displayed trend" (1) corresponds to the "Active trend" (1), it is displayed in dark blue. If the "Displayed trend" (1) does not correspond to the "Active trend" (1), it is displayed in bright blue.

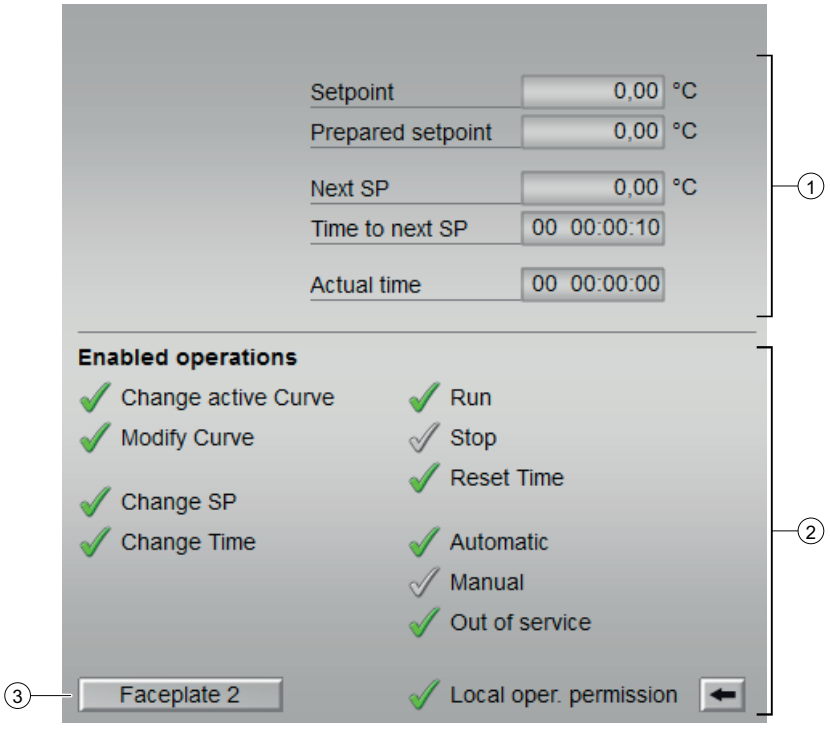
The trend is loaded when you open the standard view but it is not continuously updated. In other words, if the trend were to be changed from a different OS, for example, these changes would not be visible until you open or refresh the standard view again. A display that is not up-to-date is represented by the following symbol:

(3) Display and setting of curve points

Each field in the area represents a data point. You can reach all data points with the arrow keys left and right. If a data point is changed, the changed trend progression is displayed in the area (2) as a dashed line. However, this change only becomes effective when you write the trend to the AS.

16.4.8.4 SPCurve preview

SPCurve preview



(1) Preview area

This area shows you a preview for the following values:

- Setpoint
- Trend setpoint
- Next setpoint
- Remaining time
- Current time

(2) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- "Switch active trend": You can switch the active trend.
- "Change trend": You can change the values of the trend.
- "Change setpoint": You can change the setpoint.
- "Change time": You can change the time.
- "Run": You can set the block to "Run".
- "Stop": You can set the block to "Stop".
- "Reset time": You can reset the time.
- "Automatic": You can switch to "automatic mode".
- "Manual": You can switch to "manual mode".
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to change to the standard view of the OpStations/UserM block. You can find more information on this in the section "Operator control permissions" in the manual of the APL and in the section "Operation of multiple control rooms" of this manual.
  You can find more information on this in the APL documentation in the section "Operator control permissions".

(3) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

16.4.8.5 SPCurve block icons

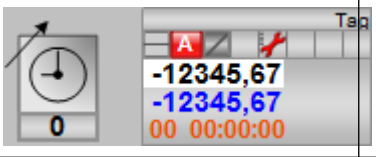
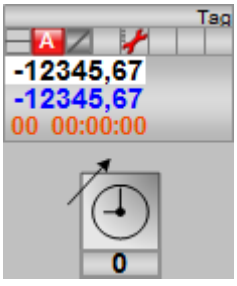
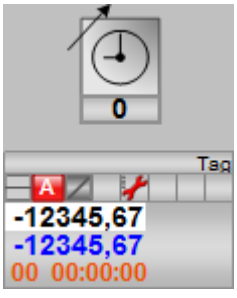
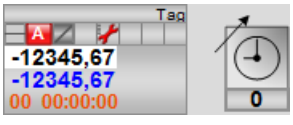
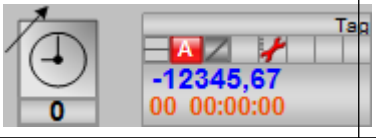
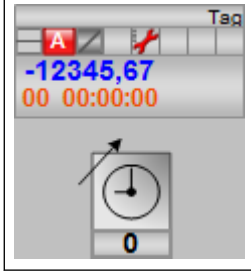
SPCurve block icons

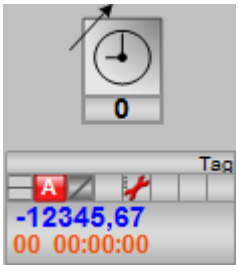
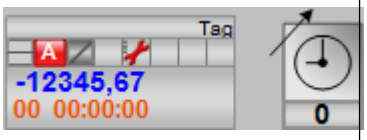
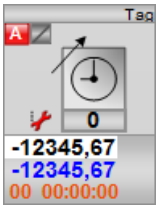
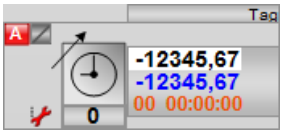
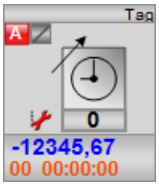
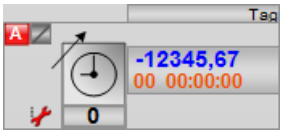
A variety of block icons are available with the following functions:

- Process tag type
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes

- Signal status, release for maintenance
- Memo display
- Process values
- Operating mode

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	Icon with process value
	2	Icon with process value
	3	Icon with process value
	4	Icon with process value
	5	Icon without process value
	6	Icon without process value

Icons	Selection of the block icon in CFC	Special features
	7	Icon without process value
	8	Icon without process value
	9	Icon with process value
	10	Icon with process value
	11	Icon without process value
	12	Icon without process value

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

16.5 TimeSwitch - Time switch with 8 switch points

16.5.1 TimeSwitch description

Object name (type + number) and family

Type + number: FB 1142

Family: Control

Number of process objects: 1 PO

Area of application

The TimeSwitch block is the software equivalent to a clock timer. Eight switch channels (pairs of "On" and "Off" switch points) can be preset. Each channel can be switched On or Off for every day of the week separately.

Configuration

Compilation settings

The TimeSwitch block requires different FCs in the number range 1-8. Therefore, the compiling settings for the CFC must be configured under Options\Settings\Compile\Download in such a way that the FC numbers 1-8 are in the reserved area.

Overlapping of block numbers

The TimeSwitch block requires FC60 (LOC_TIME). It overlaps with the block FC60 (CMP_R) frequently used in the past. This is not a problem in the APL environment because FC387 (CompAn2) is used here for comparison of analog values. If the CMR_R block is to be used in the project, it must be given a different FC number for each project. This is not possible for LOC_TIME because it is referenced with block number FC60 by the TimeSwitch.

Startup characteristics

All pulse timers are reset during startup. According to APL (Feature Bit0 'Startup'), messages and operator authorizations either retain their last value or they are reset.

Called blocks

FC1	AD_DT_TM
FC3	D_TOD_DT
FC7	DT_DAY
FC8	DT_TOD
FC60	LOC_TIME
SFB35	ALARM_8P

SFC1	READ_CLK
SFC6	RD_SINFO
SFC20	BLKMOV
SFC51	RDSYSST

Configuration

Install the block in a cyclic interrupt OB (such as OB32) in the CFC Editor. The block is also installed automatically in the startup OB (OB100).

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section TimeSwitch I/Os (Page 898).

Status bit	Parameter
0	Allocated
1	`BatchEn`
2	`On.Value`
3	`OosAct.Value`
4	`OosLi.Value`
5	`OnAct.Value`
22 - 31	Not used

Status word assignment for the parameter `Status2`

For a description of the individual parameters, refer to the section TimeSwitch I/Os (Page 898).

Status bit	Parameter
0	`MsgLock.Value`
1 - 30	Not used
31	`MS_RelOp`

Status word assignment for the parameter `Status3`

For a description of the individual parameters, refer to the section TimeSwitch I/Os (Page 898).

Status bit	Parameter
0...29	Not used
30	`UserAna1 interconnected`
31	`UserAna2 interconnected`

Status word assignment for the parameter `Status4`

For a description of the individual parameters, refer to the section TimeSwitch I/Os (Page 898).

Status bit	Parameter
0	Effective signal 1 of the message block interconnected via `EventTsIn`
1	Effective signal 2 of the message block interconnected via `EventTsIn`
2	Effective signal 3 of the message block interconnected via `EventTsIn`
3	Effective signal 4 of the message block interconnected via `EventTsIn`
4	Effective signal 5 of the message block interconnected via `EventTsIn`
5	Effective signal 6 of the message block interconnected via `EventTsIn`
6	Effective signal 7 of the message block interconnected via `EventTsIn`
7	Effective signal 8 of the message block interconnected via `EventTsIn`
8...31	Not used

16.5.2 TimeSwitch operating modes

The block can be operated using the following modes:

- On
- Out of service

If the block is in the "On" mode it functions as described. If the block is in the "Out of service" mode, all switching outputs are immediately set to "Off". When this happens, no "Off" pulses (`P_Offn`) are output.

General information on operating modes is available in the APL documentation.

16.5.3 TimeSwitch functions

The functions for this block are listed below.

Function

Every switch point can be set to a time between 00:00:00 and 23:59:59. The switch-on time must always be earlier than the switch-off time.

The output signal of a switch channel is set to "On" if the current time is later than or the same as the switch-on time and earlier or the same as the switch-off time. Timings are to the second.

Example:

Switch-on time: 08:00:00 ⇒	Output is set to "On" at 08:00:00
Switch-off point: 09:00:00 ⇒	Output is set to "Off" at 09:00:01

All 8 switch channels are logically ORed and the result determines the status of the block switching output. In other words, if a switch channel is 'On' then the switching output is 'On'. As a result, the switching output can be left "On" continuously overnight (00:00:00).

Example:

Switch-on time for channel 1: 22:00:00 (day 1) ⇒ Output is set to "On" at 22:00:00 (day 1)
Switch-off time for channel 1: 23:59:59 (day 1) ⇒
Switch-on time for channel 2: 00:00:00 (day 2) ⇒ Output is set to "Off" at 03:00:01 (day 2)
Switch-off time for channel 2: 03:00:00 (day 2) ⇒

For every switching output there are two pulsed outputs–one for the "On" pulse (`P_Onn`) and another one for the "Off" pulse (`P_Offn`). The pulse length can be set for all pulses at once in the parameter view of the faceplate.

In addition to performing the switching function, the TimeSwitch block indicates when each channel is next due to be switched "On" or "Off" based on the current program. The periods of time prior to the next switch-on points are indicated at the NxTiOn block outputs and the periods of time prior to the next switch-off points at the NxTiOffn block outputs. The amount of time is indicated in seconds (double integer).

At OS level, the information is displayed in the faceplate preview in "dd hh:mm:ss" format when you move the mouse pointer to a channel display and left-click.

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Reserved
1	1 = Operator can change switching points
2	1 = Operator can activate "On" mode
3	1 = Operator can activate "Out of service" mode
4	1 = Operator can change pulse time
5 - 11	Not used
12	1 = Operator can activate maintenance
3 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
2	1 = External time signal is used for switching
3	1 = Automatically switch over to internal time if TiExtQc <> 16#80 or 16#60
4	0 = Local time, 1 = AS internal time used for switching

Bit	Function
22	See APL function: "Update acknowledgment and error status of the message call"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"

Features

In addition to the general features (which correspond to the features of the APL), the block provides some specific features. The specific features allow you to specify which time is to be used by the block.

- **External time signal**
  If the "External time signal" feature is activated, the TimeSwitch block uses the time present at the TiExt, TiExtQc, and TiExtSt block inputs as the basis for switching.
- **Automatic switchover of time signal feature**
  This feature is only effective if the 'External time signal' feature is active at the same time. If this feature is activated and the quality code of the external time signal is bad (TiExtQc ≠ 16#80 and TiExtQc ≠ 16#60), the block uses the AS time or local AS time as the basis for switching.
- **AS time feature**
  If the AS time feature is activated, the AS time is used as the basis for switching. If the feature is not active, the local AS time is used.

Configurable reactions using the `Feature2` parameter

Bit	Function
0 - 31	Not used

16.5.4 TimeSwitch error handling

Since the block does not handle process values, it only recognizes two errors:

- `MsgErr`
  Generated by ALARM_8P and indicates that a message error has occurred.
- `ErrorNum`
  Displays the current error number.

Overview of error numbers

The `ErrorNum` I/O can be used to output various error numbers:

Error number	Meaning of the error number
00	No error

16.5.5 TimeSwitch messaging

Message behavior

The TimeSwitch block uses the ALARM_8P block for generating messages.

The messages can be suppressed centrally via the faceplate or via `MsgLock` (program).

The free alarm inputs can be interconnected via the parameters `ExtMsg1` to `ExtMsg3`.

The associated values (`ExtVaXXX`) of the message block can be assigned freely.

`MsgSup` is set if the `RunUpCyc` cycles have not finished running since the restart, `MsgLock = TRUE` or `MsgStat = 21`.

Process messages

Message instance	Message ID	Message class	Event
`MsgEvId`	SIG 1	AS control system fault	$$BlockComment$$ External error
	SIG 2	AS control system fault	$$BlockComment$$ External message 1
	SIG 3	AS control system fault	$$BlockComment$$ External message 2
	SIG 4	AS control system fault	$$BlockComment$$ External message 3
	SIG 5	-	-
	SIG 6	-	-
	SIG 7	-	-
	SIG 8	-	-

Associated values for the message instance MsgEvId

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVa104`
5	`ExtVa105`
6	`ExtVa106`
7	`ExtVa107`
8	`ExtVa108`

16.5.6 TimeSwitch I/Os

Input parameters

Parameter	Description	Data type	Default
MsgEvId	Message event ID	DWORD	16#000000D1
P_Time *	Time for on/off pulse	REAL	2.0
SwAct1 ... SwAct8 *	active days switching point 1 ... 8	BYTE	16#00
SwTiOff1 ... SwTiOff8 *	time switching point 1 ... 8 off	DINT	0
SwTiOn1 ... SwTiOn8 *	time switching point 1 ... 8 on	DINT	0
TiExt	External time	DATE_AND_TIME	2011-04-04-0:00:00
TiExtQc	Quality of external time signal	BYTE	16#80
TiExtSt	Daylight saving time	STRUCT • VALUE: BOOL • ST: BYTE	- • 0 • 16#80

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
DwAs	AS day of the week	INT	0
DwCurr	Currently used day of the week	INT	0
DwExt	External day of the week	INT	0
DwLoc	Local day of the week	INT	0
MsgAckn	ALARM_8P: ACK_STATE output	WORD	16#0000
MsgErr	1=Messaging error occurs	BOOL	0
MsgStat	ALARM_8P: Status output	WORD	16#0000
NxTiOff	Next off switch point	REAL	0.0
NxTiOff1 ... NxTiOff8	Next off switch point	REAL	0.0
NxTiOn	Next on switch point	REAL	0.0
NxTiOn1 ... NxTiOn8	Next on switch point	REAL	0.0
On	Output switch point general	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
On1 ... On8	Output switch point 1 ... 8	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Parameter	Description	Data type	Default
P_Off	Off pulse output switch point general	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
P_Off1 ... P_Off8	Off pulse output switch point 1 ... 8	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
P_On	On pulse output switch point general	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
P_On1 … P_On8	On pulse output switch point 1 ... 8	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
StCurr	Current daylight saving time	BOOL	0
StExt	External daylight saving time active	BOOL	0
StLoc	Local daylight saving time active	BOOL	0
TdAs	AS time of the day	DINT	0
TdCurr	Currently used time of the day	DINT	0
TdExt	External time of the day	DINT	0
TdLoc	Local time of the day	DINT	0

16.5.7 TimeSwitch block diagram

The block has no block diagram.

16.5.8 Operator control and monitoring

16.5.8.1 TimeSwitch views

TimeSwitch block views

The TimeSwitch block provides the following views:

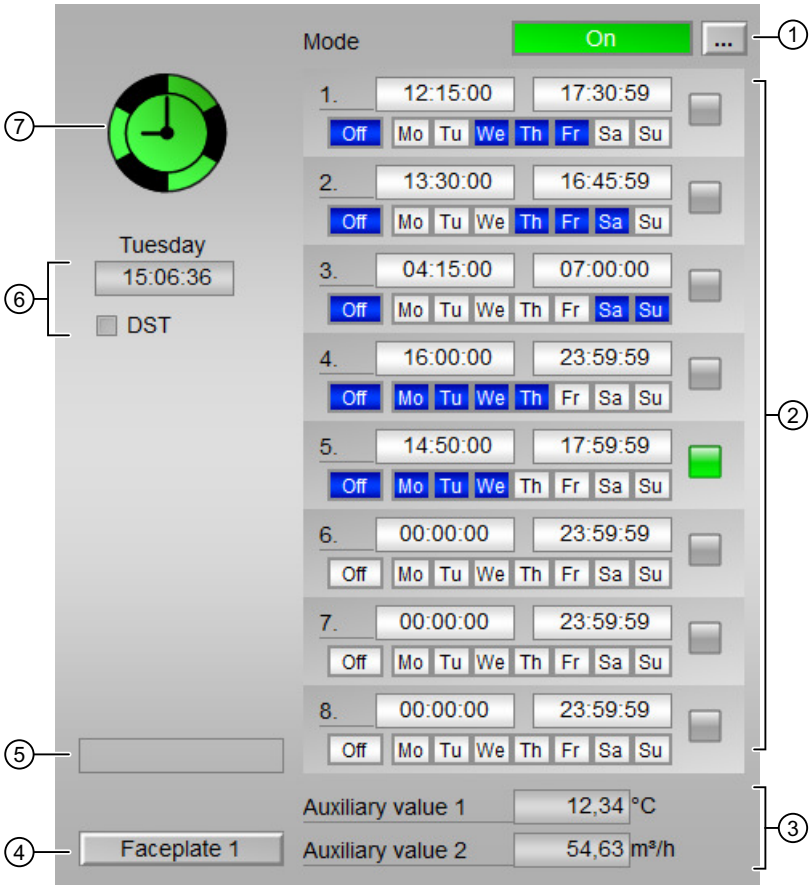
- TimeSwitch standard view (Page 900)
- Message view
- Trend view
- TimeSwitch parameter view (Page 902)
- TimeSwitch preview (Page 903)
- Memo view

- Batch view
- TimeSwitch block icons (Page 904)

You can find general information on the faceplate and block icon in the "Structure of the faceplate" and "Block icon structure" sections of the APL function manual.

16.5.8.2 TimeSwitch standard view

TimeSwitch standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

You can find more information on changing the operating mode in the "Switching operating states and operating modes" section of the APL manual.

(2) Display of the activation and deactivation times

This display shows the activation and deactivation times and displays the active/inactive weekdays. The 8 time windows can be activated or deactivated. If a time window is currently active, this is visualized by a green square.

(3) Display of auxiliary values

This display is only visible when the corresponding block input is interconnected.

You can use this area to display two auxiliary values that have been configured in the engineering system (ES). You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(4) Navigation button for switching to the standard view of any faceplate

This display is only visible when the corresponding block input is interconnected.

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(5) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

(6) Display of the current time

This area displays the current day and current time and whether daylight-saving time is currently active.

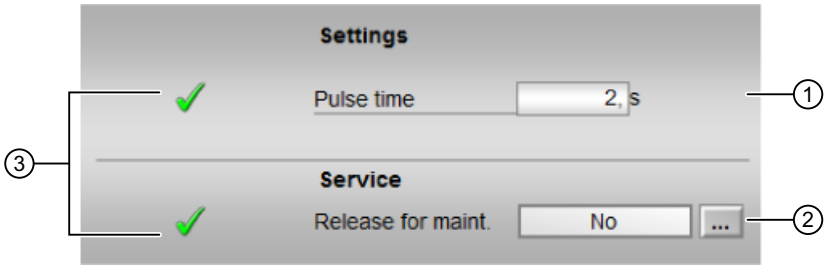
(7) Status display

The current status of the timer is shown.

You can find more information about this in the section TimeSwitch block icons (Page 904)

16.5.8.3 TimeSwitch parameter view

TimeSwitch parameter view



(1) Settings

You can change parameters in this area. Refer to the section "Changing values" in the APL manual for more information on the following parameters you can set:

- "Pulse time": Duration of the pulse at the pulsed output signal.

(2) Service

You can select the following functions in this area:

- "Release for maintenance"

Refer to the "Switching operating states and operating modes" section in the APL manual.

You can find information on this area in the section "Release for maintenance" in the manual of the APL.

(3) Enabled operations

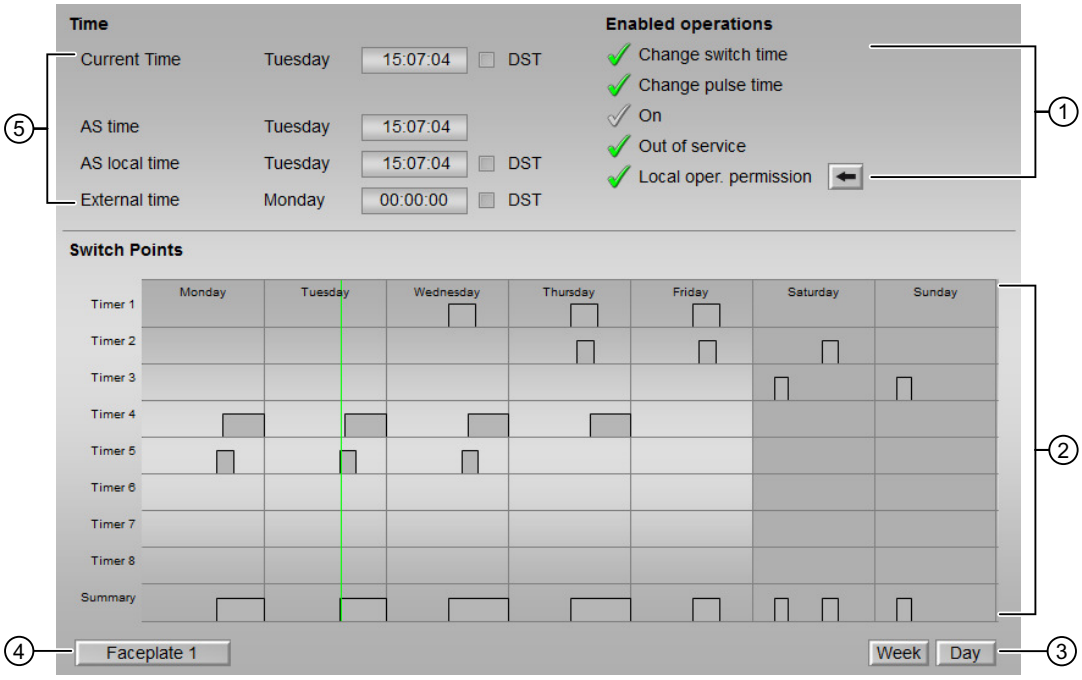
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

16.5.8.4 TimeSwitch preview

TimeSwitch preview



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- "Change switching time": You can adjust the switching times.
- "Change pulse time": You can change the duration of the pulse.
- "Run": You can switch to "On" operating mode.

- "Out of service": You can switch to "Out of service" operating mode.
- "Local operating permission": Use the ← button to change to the standard view of the OpStations/UserM block. You can find more information on this in the section "Operator control permissions" in the manual of the APL and in the section "Operation of multiple control rooms" of this manual.
  You can find more information on this in the APL documentation in the section "Operator control permissions".

(2) Preview area of the switching points

In this area, you obtain a weekly or daily graphical preview of the individual timers (timers 1-8). In addition, the result of the summed timers (sum) is displayed on the switching output of the timer.

(3) Switchover of the preview area of the switching point

With these buttons, you can toggle between a weekly and a daily display of the preview range of the switching points.

(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find more information on this in the section "Opening additional faceplates" in the APL documentation.

(5) Preview area of the timers

This area shows you a preview for the following values:

- Current time: Time currently used by the block.
- AS time: Actual time of the automation system.
- AS local time: AS time including the local time zone
- External time signal: Time made available to the block via an external time signal.

16.5.8.5 TimeSwitch block icons

TimeSwitch block icons

A variety of block icons are available with the following functions:

- Process tag type
- Control system errors and external errors
- Operating modes
- Signal status, release for maintenance

- Memo display
- Status display

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	
	3	
	4	
	5	
	6	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

Status display

The following states are shown here:

Icon	Meaning
	Output signal of the timer enabled
	Output signal of the timer disabled
	Timer for out of service

16.6 SplRng4 signal splitter

16.6.1 SplRng4 description

Object name (type + number) and family

Type + number: FB 1175

Family: Control

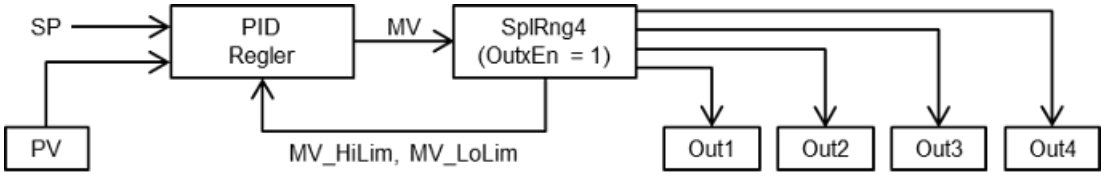
Area of application for SplRng4

The block is used for the following applications:

- Splitting the output signal of a PID controller to a max. of 4 final controlling elements.
- Adjusting the controller manipulated variable with deadbands for bypassing, if needed

Method of operation

The block is used to split signals output coming from a PID controller. You can use this controller to control up to four valves, for example.



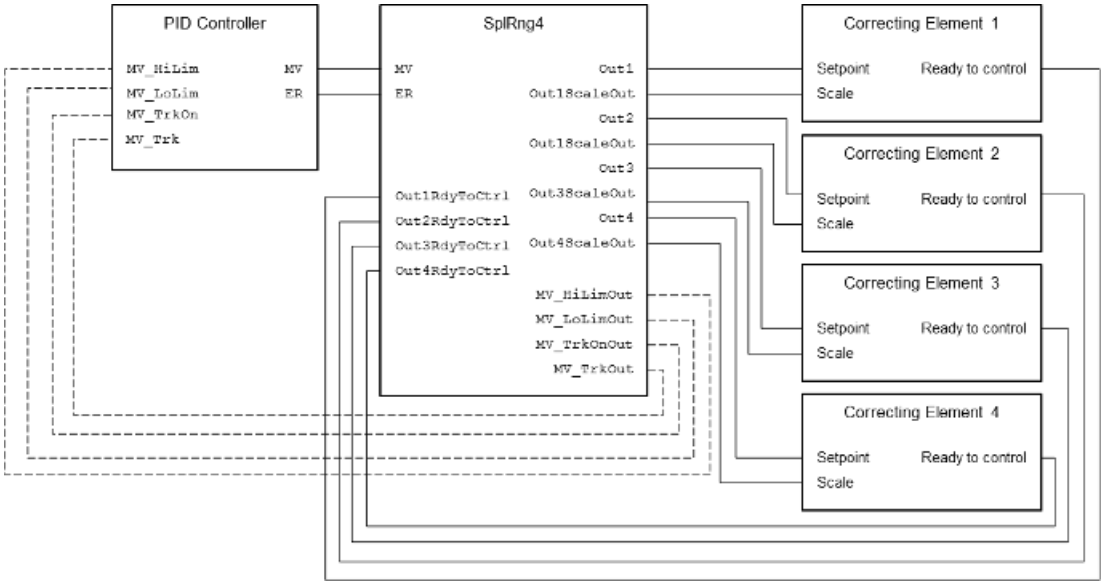
The outputs `Outx` are adapted to the physical variable by configuring the high/low limits of `OutxScale`.

The output parameters `OutxAct` show (= 1) that the corresponding output parameters `Outx` are active if the input value `MV` is within the setting range for `Outx`.

Configuration

Use CFC editor to install the block in the OB in which the controller block runs whose manipulated variable is being processed.

Interconnection in the CFC:



The I/Os for controlling the final controlling elements can vary depending on the final controlling element:

Blocks as final controlling element	VlvPosL	VlvAnL	MotSpdC
Setpoint	MV	MV-Ext	SP_Ext
Scale	MV_OpScale	-	-
Ready to control	Logic based on the user requirement. E.g.: Value should be specified only in automatic mode (AutAct + PosReached)	Logic based on the user requirement. E.g.: Value should be specified only in automatic mode and with active external value (AutAct + MV_ExtAct + PosReached)	Logic based on the user requirement. E.g.: Value should be specified only in automatic mode and with active external value (AutAct + SP_ExtAct)

Startup characteristics

This block has no startup characteristics.

Status word allocation for `Status` parameter

This block does not have the `Status` parameter.

16.6.2 SplRng4 operating modes

SplRng4 operating modes

This block does not have any operating modes.

16.6.3 SplRng4 functions

SplRng4 functions

The functions for this block are listed below.

Supplying the manipulated variables

When the startup of the split range block is finished and the controller enable is present, the manipulated variables are output for the final controlling elements.

Final controlling elements that are not available are assigned the configured lower limit as a manipulated variable.

Manipulated variables are calculated for final controlling elements that are available based on the "inverted" state, the slope, and the ranges of the final controlling elements and the manipulated variable of the controller.

If the manipulated variable of the controller is outside the limits determined by the block, the value is limited internally.

Adjusting the controller

If the active manipulated variable of the controller is in a standby area, the manipulated variable of the controller is adjusted in order to skip this idle range. This means that a jump to the higher or lower limit of the next final controlling element takes place. A deadband is taken into consideration to prevent cyclic switching on and off of the final controlling elements.

The controller is adjusted; if it was not, the next final controlling element would not be activated due to limitation of the final controlling elements. If the manipulated variable of the controller exceeds the limits determined by the block, it is adjusted to the limits.

Splitting of the output signal of a controller

The `MV` block input can be used to split the output signal of a controller.

The example in the diagram uses 2 inverted and 2 non-inverted final controlling elements. The manipulated variable of the final controlling elements is shown on the y axis (`Outx`: 0 – 100%).

The x axis shows the manipulated variable that is output by the controller block. The range of the manipulated variable of the controller is determined by the SplRng4 block and calculated from the individual ranges of the final controlling elements.

The ranges of the MVx_Init final controlling elements are determined from:

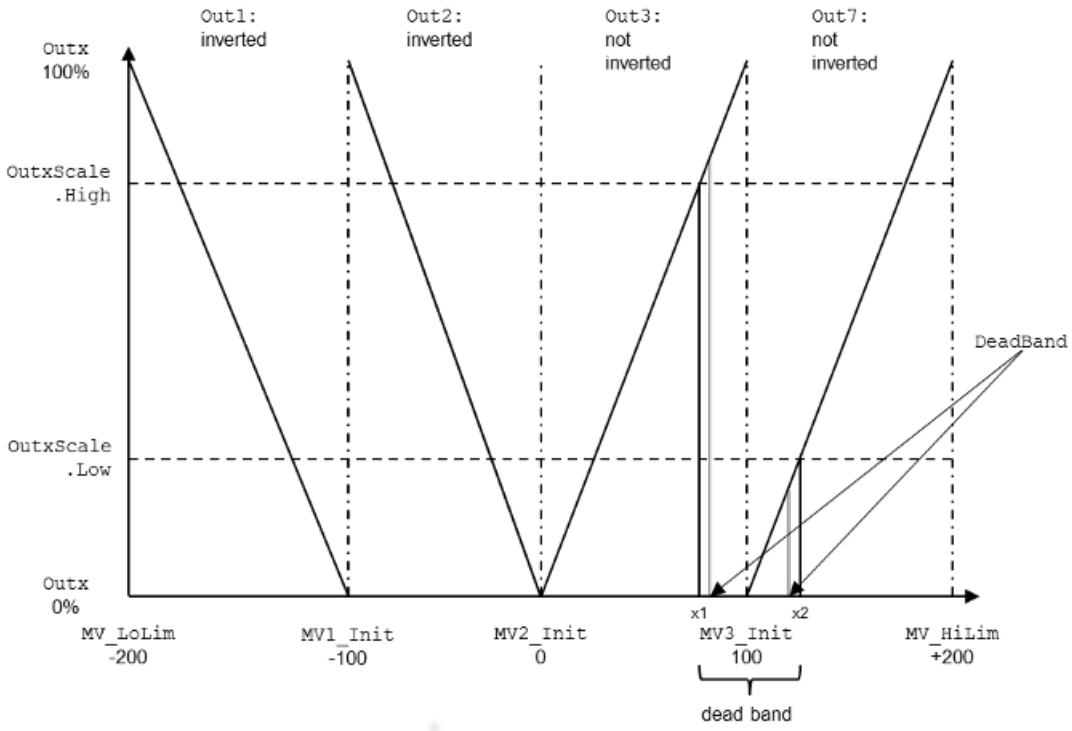
- MV[x]_Init = MV[x-1]_Init + (100 / `OutxSteepN`)

(`OutxSteepN` = slope of the final controlling element).

These limits are determined cyclically and stored internally. If a final controlling element is not enabled or not available (`OutxEn` = FALSE), the range = 0 and thus MV[x]_Init = MV[x+1]_Init.

The following is calculated for the controller.

- High limit `MV_HiLim` = Σ (100 / `OutxSteepN`) of the non-inverted final controlling elements
- Low limit `MV_LoLim` = Σ (100 / `OutxSteepN`) of the inverted final controlling elements



The response at the transition between the final controlling elements can be changed with the input parameters `DeadBand12`, `DeadBand23` or `DeadBand34`. If a negative value is configured, the final controlling elements overlap; in the case of a positive value, the final controlling elements are equalized.

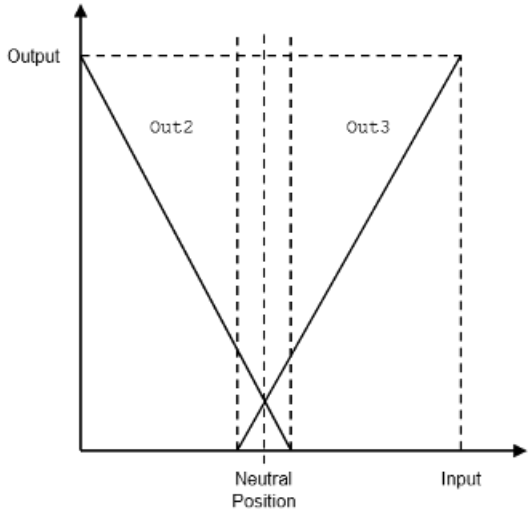


Figure 16-3 Negative value for the deadband

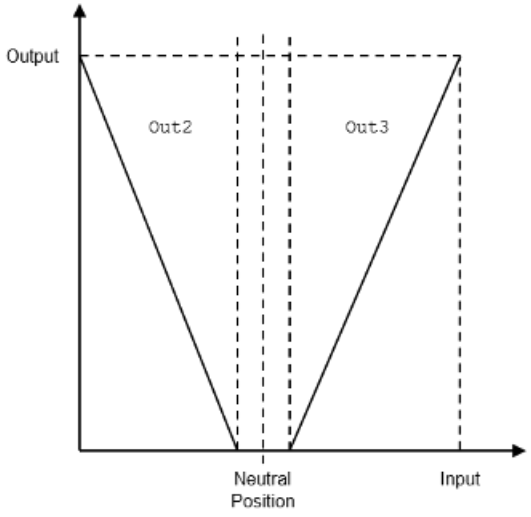


Figure 16-4 Positive value for the deadband

The final controlling elements are limited to their respective limits, `OutxScale.High` and `OutxScale.Low`. This would mean that a deadband would be created for a controller manipulated variable between x1 and x2 in which there would be no change in the final controlling elements. To prevent this from happening, waiting for a configured deadband (`DeadBand`) is incorporated after violations of `OutxScale.High` / `OutxScale.Low` and then a jump is made to the limit of the next final controlling element.

Example

`MV` > x1 + DeadBand and control deviation is positive

⇒ `MV` is adjusted x2 by setting outputs `MV_TrkOut` = x2 and `MV_TrkOnOut` = TRUE for `ExtT` seconds.

Startup characteristics for SplRng4

Use the `Feature`.Bit 0 to define the startup characteristics of this block.

`Feature`.Bit 0 = 0:

The `Outx.Value` output parameter is reset to `OutxScale.Low` on startup.

`Feature`.Bit 0 = 1:

The most recently saved value is output at the `Outx.Value` output parameter on startup.

Zero offset

Use the `Feature`.Bit 5 to define the zero point characteristics of this block. The zero point defines the transition between the inverted and non-inverted final controlling elements.

`Feature`.Bit5 = 0:

The zero point between inverted and non-inverted final controlling elements is zero, `MV_HiLim` is in the negative range.

`Feature`. Bit5 = 1:

The zero point between inverted and non-inverted final controlling elements is in the positive range, `MV_HiLim` is zero.

Four final controlling elements are configured in the example below. Two of these final controlling elements are inverted and two are not inverted.

Fea-ture.Bit5	Out1SteepN (Out1Inv = 1)	Out2SteepN (Out2Inv = 1)	Out3SteepN (Out3Inv = 0)	Out4SteepN (Out4Inv = 0)	MV_Lo-Lim	MV_Hi-Lim	Zero point
0	4	4	4	4	-50.0	50.0	0.0
1	4	4	4	4	0.0	100.0	50.0

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
5	Zero offset

16.6.4 SplRng4 error handling

SplRng4 error handling

For information about error handling for all blocks, see also 'Error handling' in the Basics section of the APL manual.

The following errors can be displayed for this block:

- Error numbers

Overview of error numbers

The `ErrorNum` I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
30	The value of `In` can no longer be displayed in the REAL number field.

Error number	Meaning of the error number
70	Invalid `OutxScale`: `OutxScale.High` < 0.0 `OutxScale.Low` < 0.0
71	Invalid `OutxInv`: Plausibility check for sequence of the final controlling elements. An non-inverted final controlling element must not be followed by an inverted final controlling element.

16.6.5 SplRng4 messaging

Message functionality

This block has no message behavior.

16.6.6 SplRng4 I/Os

SplRng4 I/Os

Input parameters

Parameter	Description	Type	Default
`MV`	Manipulated variable	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
`ER`	Control Difference	REAL	0.0
`CtrlEn`	PID enable	BOOL	1
`Out1En` .. `Out4En`	Out1 .. Out4: 1 = Enable	BOOL	0
`Out1RdyToCtrl` .. `Out4RdyToCtrl`	Out1 .. Out4:1 = Ready to control	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
`Out1Scale` .. `Out4Scale`	Out1 .. Out4 Limit values for scale	STRUCT • High: REAL • Low: REAL	- • 100.0 • 0.0
`Out1SteepN` .. `Out4SteepN`	Out1 .. Out4 Steepness of the actuator	REAL	1.0
`Out1Inv` .. `Out4Inv`	Out1 .. Out4: 1 = Inverted	BOOL	0
`ExtT`	Extension time for MV_TrkOn	REAL	2.0
`DeadBand12`	Dead Band Width Out1 – Out2	REAL	0.0

Parameter	Description	Type	Default
DeadBand23	Dead Band Width Out2 – Out3	REAL	0.0
DeadBand34	Dead Band Width Out3 – Out4	REAL	0.0

Output parameters

Parameter	Description	Type	Default
Out1 .. Out4	Out1 .. Out4: Output Value	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
Out1Act .. Out4Act	Out1 .. Out4: Output Value Active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Out1ScaleOut .. Out4ScaleOut	Out1 .. Out4: Limit value for scale	STRUCT • High: REAL • Low: REAL	- • 100.0 • 0.0
MV_HiLimOut	Limit (high) for Manipulated variable MV	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
MV_LoLimOut	Limit (low) for Manipulated variable MV	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
MV_TrkOnOut	1 = Tracking of Manipulated variable MV	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
MV_TrkOut	Tracking value for Manipulated variable MV	STRUCT • Value: BOOL • ST: BYTE	- • 0.0 • 16#80

16.6.7 SplRng4 block diagram

The block has no block diagram.

System blocks

17

17.1 ASTimeBCD - Output the CPU time in BCD format

17.1.1 ASTimeBCD description

Object name (type + number) and family

Type + number: FC 611

Family: System

Number of process objects: 0 PO

Called blocks

SFC1	READ_CLK

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

17.1.2 ASTimeBCD operating modes

The block does not have any operating modes.

17.1.3 ASTimeBCD functions

The block calls the SFC READ_CLK system function and transfers the read time stamp (date and time) to the output in BCD format.

17.1.4 ASTimeBCD error handling

The block does not have any error handling.

17.1.5 ASTimeBCD messaging

The block has no message behavior.

17.1.6 ASTimeBCD I/Os

Input parameters

None

Output parameters

Parameter	Description	Data type	Default
DAY	Output Day in Integer Format	BYTE	16#00
HOUR	Output Hour in Integer Format	BYTE	16#00
MINUTE	Output Minute in Integer Format	BYTE	16#00
MONTH	Output Month in Integer Format	BYTE	16#00
MSEC	Output Msec in Integer Format	WORD	16#0000
SECOND	Output Second in Integer Format	BYTE	16#00
YEAR	Output Year in Integer Format	BYTE	16#00

17.1.7 ASTimeBCD block diagram

The block has no block diagram.

17.1.8 Operator control and monitoring of ASTimeBCD

The block does not have any views.

Monitoring blocks 18

18.1 MonAn08 - Measured value monitoring with 8 limits

18.1.1 MonAn08 description

Object name (type + number) and family

Type + number: FB 1102

Family: Monitor

Number of process objects: 1 PO

Area of application of MonAn08

The block is used for the following areas of application:

- Monitoring of an analog process value to a maximum of 8 freely configurable limits.
- Monitoring of the gradient of an analog process value

The "Smooth" block from the Advanced Process Library (APL) can be used to smooth the input value.

If the limits are overshot/undershot, this is indicated at a corresponding output (`PV_LimxAct`) (see Message behavior (Page 927)).

Startup characteristics

Following startup, the messages are suppressed for the number of cycles assigned in the `RunUpCyc` parameter. A startup can be simulated via the `Restart` input.

Time response

The block must be called by means of a cyclic interrupt OB. The sampling time of the block is entered in the `SampleTime` parameter.

Called blocks

SFC6	RD_SINFO
SFB35	ALARM_8P

Calling OBs

In the same OB with and after the block whose measured value is to be monitored. Also in OB100.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section MonAn08 I/Os (Page 929).

Status bit	Parameter
0	Allocated
1	`BatchEn`
2	`SimAct`
3	`OosAct.Value`
4	`OosLi.Value`
5	Not used
6	`OnAct.Value`
7	`BypassAct.Value`
8	Display of `BypassAct.Value` in the faceplate (display and operator controls) and block icon
9	Not used
10	`SimLiOp`
11	`DelPV_Lim1 (PV_Lim1 delay active)`
12	`DelPV_Lim2 (PV_Lim2 delay active)`
13	`DelPV_Lim3 (PV_Lim3 delay active)`
14	`DelPV_Lim4 (PV_Lim4 delay active)`
15	`DelPV_Lim5 (PV_Lim5 delay active)`
16	`DelPV_Lim6 (PV_Lim6 delay active)`
17	`DelPV_Lim7 (PV_Lim7 delay active)`
18	`DelPV_Lim8 (PV_Lim8 delay active)`
19	`SumDelay (delay active)`
20	`PV_Lim1MsgEn`
21	`PV_Lim2MsgEn`
22	`PV_Lim3MsgEn`
23	`PV_Lim4MsgEn`
24	`PV_Lim5MsgEn`
25	`PV_Lim6MsgEn`
26	`PV_Lim7MsgEn`
27	`PV_Lim8MsgEn`
28 - 29	Not used
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word assignment for the parameter Status2

For a description of the individual parameters, refer to the section MonAn08 I/Os (Page 929).

Status bit	Parameter
0	MsgLock
1	PV_Lim1Act
2	PV_Lim2Act
3	PV_Lim3Act
4	PV_Lim4Act
5	PV_Lim5Act
6	PV_Lim6Act
7	PV_Lim7Act
8	PV_Lim8Act
9	PV_Lim1En
10	PV_Lim2En
11	PV_Lim3En
12	PV_Lim4En
13	PV_Lim5En
14	PV_Lim6En
15	PV_Lim7En
16	PV_Lim8En
17	PV_Lim1Op (1 = PV_Lim1.ST = 16#FF)
18	PV_Lim2Op (1 = PV_Lim2.ST = 16#FF)
19	PV_Lim3Op (1 = PV_Lim3.ST = 16#FF)
20	PV_Lim4Op (1 = PV_Lim4.ST = 16#FF)
21	PV_Lim5Op (1 = PV_Lim5.ST = 16#FF)
22	PV_Lim6Op (1 = PV_Lim6.ST = 16#FF)
23	PV_Lim7Op (1 = PV_Lim7.ST = 16#FF)
24	PV_Lim8Op (1 = PV_Lim8.ST = 16#FF)
25 - 30	Not used
31	MS_RelOp

Status word assignment for the parameter Status3

Status bit	Parameter
0 - 18	Not used
19	GradHUpAct.Value
20	GradHDnAct.Value
21	GradLAct.Value
22	GradHUpEn
23	GradHDnEn
24	GradLEn
25	GradHUpMsgEn

Status bit	Parameter
26	GradHDnMsgEn
27	GradLMsgEn
28	0 = Falling measured value 1 = Rising measured value
29 - 31	Not used

Status word assignment for the parameter Status4

Status bit	Parameter
0	Effective signal 1 of the message block connected via EventTsIn
1	Effective signal 2 of the message block connected via EventTsIn
2	Effective signal 3 of the message block connected via EventTsIn
3	Effective signal 4 of the message block connected via EventTsIn
4	Effective signal 5 of the message block connected via EventTsIn
5	Effective signal 6 of the message block connected via EventTsIn
6	Effective signal 7 of the message block connected via EventTsIn
7	Effective signal 8 of the message block connected via EventTsIn
8 - 23	Effective signal 9..16 of the message block connected via EventTsIn
24 - 31	Not used

Status word assignment for the parameter PV_LimStatus

For a description of the individual parameters, refer to the section MonAn08 I/Os (Page 929).

Status bit	Parameter
0	PV_Lim1Act.Value
1	PV_Lim2Act.Value
2	PV_Lim3Act.Value
3	PV_Lim4Act.Value
4	PV_Lim5Act.Value
5	PV_Lim6Act.Value
6	PV_Lim7Act.Value
7	PV_Lim8Act.Value

18.1.2 MonAn08 operating modes

The block does not have any operating modes.

18.1.3 MonAn08 functions

PV_Out output

The measured analog value appears at the PV_Out output. The analog value (PV) is specified by the process.

Measured value monitoring

The PV_Out output is monitored for compliance with up to 8 limits. The limits are programmable. Incoming messages can be delayed with the DelayTx inputs (x = 1 to 8); outgoing messages with the DelayTx_G inputs (x = 1 to 8).

Each limit is specified in a ModeX parameter in its behavior. The configured colors (ModeX.Bit1 – ModeX.Bit3) are only evaluated in the limit value view if the "User-configurable message classes" function is not active. If the function is used, the colors of the configured message classes are displayed in the limit view.

Quality code

The .ST structure elements of the parameters for the measured value PV plus the external message signals contain the quality code.

The following quality code data is evaluated:

- Quality Code = 16#60: Simulation on driver block active
- Quality Code = 16#80: Valid value
- Quality Code <> 16#60 oder <> 16#80: Invalid value

Deadband

To suppress values that fluctuate around the zero point, this block has the deadband standard function.

Low pass filter

The block smooths the input value (PV) using a 1st order delay. This delay time can be configured with the SmoothTi parameter. The input of SmoothTi is limited from 0 to 999999 s. The block writes back the limit values if the input value is outside the limits.

The block works according to the following formula:

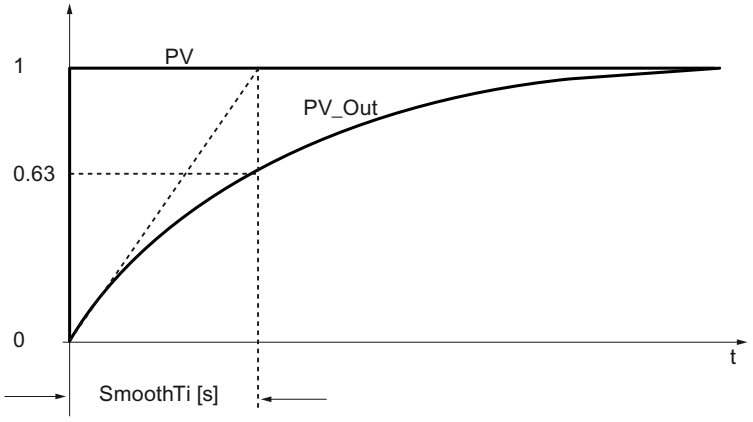
$$PV_Out_{(n)} = PV_{(n)} + (PV_Out_{(n-1)} - PV_{(n)}) \cdot e^{\left(\frac{-SampleTime}{SmoothTi}\right)}$$

Where:

- PV_Out = output value
- SmoothTi = delay time

- `SampleTime` = sampling time
- `PV` = input value

The formula only applies to `SmoothTi` > 0. If `SmoothTi` = 0, the input is passed directly to the output. If the input value is outside the REAL range limits, the calculation is stopped. If the input value is outside the range limits again, the calculation is resumed automatically.



Release for maintenance

This block features the standard function "Release for maintenance".

Simulating signals

This block features the standard function "Simulate signals".

Bypass function

This block has the standard function, bypass signals.

Interconnection of limits

The `PV_LimX` limits can be interconnected. If they are not interconnected (`PV_LimX.ST = 16#FF`), the operator can change the corresponding limit.

Gradient monitoring

The `PV_Grad` gradient is calculated with a delay function configured via `LagTime`. This setting smooths the jumps of the PV input value with gradient calculation.

The gradient peak values are output at the output parameters `PV_GradNP` (negative slope) and `PV_GradPP` (positive slope). They are reset as soon as the operator issues the reset command.

The slope of the PV_Grad gradient can be monitored for the following limits:

- Limit (high) for positive gradients (GradHUpLim)
- Limit (high) for negative gradients (GradHDnLim)
- Limit (low) for absolute gradients (GradLLim)

The individual monitoring functions are activated at the corresponding "Enable" parameters, e.g. GradHUpEn for activating the gradient high limit for positive gradients (GradHUpLim).

When the values you have defined are reached or exceeded, this is indicated at the corresponding "Active" output parameters, e.g. with GradHUpAct = 1 for the limit (high) for positive gradients.

The "EventTs" block of the APL is used in order to output messages for these alarms. Connect the EventTsOut output of the EventTs-block to the EventTsIn input of the MonAn08-Bausteins to link the blocks.

To generate messages, the "Active" output parameters and message enables of MonAn08 must be connected to the InX and MsgEnX inputs of the EventTs-block and the message texts and classes must be appropriately configured:

Output limit active	Enable message	Message class (APL)	Message text (APL)
GradHUpAct	GradHUpMsgEnOut	Alarm - high	$$BlockComment$$ limit (high) for positive gradient violation
GradHDnAct	GradHDnMsgEnOut	Alarm - high	$$BlockComment$$ limit (high) for negative gradient violation
GradLAct	GradLMsgEnOut	Alarm - low	$$BlockComment$$ limit (low) for absolute encoder violated

The messages can be enabled and disabled at the block MonAn08:

- Message for alarms (high) for positive gradients: GradHUpMsgEn
- Message for alarms (high) for negative gradients: GradHDnMsgEn
- Message for alarms (low) for absolute gradients: GradLMsgEn

Example of generating alarms for gradient monitoring

The following example shows how alarms for gradient monitoring are generated. (with appropriate configuration via EventTs)

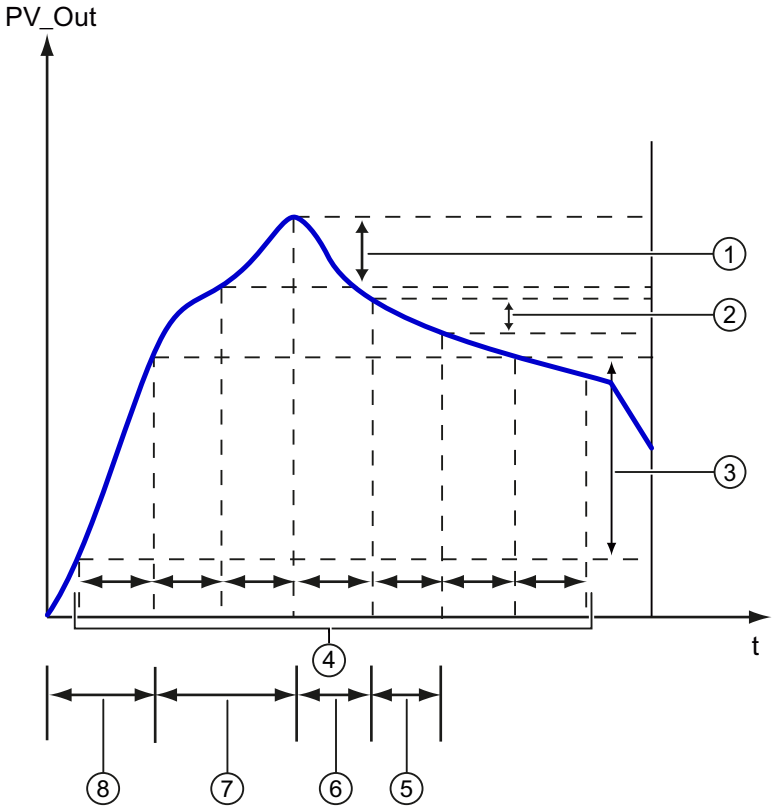


Figure 18-1 MonAnL - Gradient monitoring

Number	Description
1	Absolute gradient difference ≥ `GradHDnLim`
2	Absolute gradient difference ≤ `GradLLim`
3	Gradient difference ≥ `GradHUpLim`
4	Sampling time (`SampleTime`)
5	Alarm (low) for absolute gradients (`GradLAct = 1`)
6	Alarm (high) for negative gradients (`GradHDnAct` = 1)
7	No alarm
8	Alarm (high) for positive gradients (`GradHUpAct` = 1)

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode

Bit	Function
4	1 = Operator can enable the bypass functionality
5	1 = Operator can disable the bypass functionality
6	Not used
7	1 = The operator can reset the peak values
8	1 = Operator can activate/deactivate messages via `GradHUpMsgEn`
9	1 = Operator can activate/deactivate messages via `GradHDnMsgEn`
10	1 = Operator can activate/deactivate messages via `GradLMsgEn`
11	1 = Operator can activate the Simulation function
12	1 = Operator can activate the Release for maintenance function
13	1 = Operator can enable/disable limits and messages
14 - 15	Not used
16	1 = Operator can change the hysteresis
17 - 18	Not used
19	1 = The operator can change the value for the gradient high limit for positive slopes (`GradHUpLim`)
20	1 = The operator can change the value for the gradient high limit for negative slopes (`GradHDnLim`)
21	1 = The operator can change the value for the gradient low limit for negative slopes (`GradLLim`)
22	1 = Operator can change the "Deadband" parameter `DeadBand`
23 - 29	Not used
30	1 = Operator can enter `SmoothTi`
31	Not used

The block has the following permissions for the `OS1Perm` parameter:

Bit	Function
0	1 = Operator can change limit 1
1	1 = Operator can change limit 2
2	1 = Operator can change limit 3
3	1 = Operator can change limit 4
4	1 = Operator can change limit 5
5	1 = Operator can change limit 6
6	1 = Operator can change limit 7
7	1 = Operator can change limit 8
8	1 = Operator can activate / deactivate limit 1
9	1 = Operator can activate / deactivate limit 2
10	1 = Operator can activate / deactivate limit 3
11	1 = Operator can activate / deactivate limit 4
12	1 = Operator can activate / deactivate limit 5
13	1 = Operator can activate / deactivate limit 6
14	1 = Operator can activate / deactivate limit 7
15	1 = Operator can activate / deactivate limit 8
16	1 = Operator can activate / deactivate messages for limit 1

Bit	Function
17	1 = Operator can activate / deactivate messages for limit 2
18	1 = Operator can activate / deactivate messages for limit 3
19	1 = Operator can activate / deactivate messages for limit 4
20	1 = Operator can activate / deactivate messages for limit 5
21	1 = Operator can activate / deactivate messages for limit 6
22	1 = Operator can activate / deactivate messages for limit 7
23	1 = Operator can activate / deactivate messages for limit 8
24 - 31	Not used

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
9	See APL function: "The substitute value is enabled if the block is bypassed"
13	Use of separate operator authorizations for Limit view: 0: Operator authorization is activated and deactivated via OS_Perm.Bit12. 1: Operator authorization is activated and deactivated via OS1Perm.Bit0 – OS1Perm.Bit23.
19	See APL function: "Changes signal status of outputs in OOS"
22	See APL function: "Update acknowledgment and error status of the message call"
23	See APL function: "Limits disabled in faceplate"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"
26	See APL function: "Reaction of the switching points in the "Out of service" operating mode"
28	See APL function: "Deactivate switching points"
29	See APL function: "Signal limit violation"

18.1.4 MonAn08 error handling

The following errors can be displayed for this block:

- Error numbers
- Control system fault (`CSF`)

Overview of error numbers

The ErrorNum I/O can be used to output various error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
30	The value of PV can no longer be displayed in the REAL number field.

Control system fault (CSF)

An external signal can be activated via the CSF input. A control system fault is triggered if this signal = 1. You will find more detailed information in the APL documentation.

18.1.5 MonAn08 messaging

Message behavior

The MonAn08 block uses two ALARM8_P blocks for generating messages.

The following messages can be generated for this block:

- Control system fault
- Process messages
- Instance-specific messages

The messages can be delayed by the time configured in the parameters DelayTx and DelayTx_G.

The messages for limit value violations can be enabled individually via the appropriate PV_LimXMsgEn inputs.

The messages can be locked centrally via the faceplate or via MsgLock (program).

Control system fault

You can interconnect an external fault (signal) to the input parameter CSF. If this signal changes to CSF = 1, a control system fault is triggered ((MsgEvId2, SIG 2).).

Process messages

Message instance	Message ID	Message class	Event
MsgEvId 1	SIG 1	High warning limit	$$BlockComment$$ Limit 1
	SIG 2	High warning limit	$$BlockComment$$ Limit 2
	SIG 3	High warning limit	$$BlockComment$$ Limit 3
	SIG 4	High warning limit	$$BlockComment$$ Limit 4
	SIG 5	High warning limit	$$BlockComment$$ Limit 5
	SIG 6	High warning limit	$$BlockComment$$ Limit 6
	SIG 7	High warning limit	$$BlockComment$$ Limit 7
	SIG 8	High warning limit	$$BlockComment$$ Limit 8

Explanation:

$$BlockComment$$: content of the instance-specific comment

Instance-specific messages

You can use up to six instance-specific messages with this block.

Message instance	Message ID	Message class	Event
MsgEvId 2	SIG 1	AS process control message - fault	$$BlockComment$$ External error has occurred
	SIG 2	< No message >	$$BlockComment$$ Simulation active
	SIG 3	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 4	AS process control message - fault	$$BlockComment$$ External message 2
	SIG 5	AS process control message - fault	$$BlockComment$$ External message 3
	SIG 6	AS process control message - fault	$$BlockComment$$ External message 4
	SIG 7	AS process control message - fault	$$BlockComment$$ External message 5
	SIG 8	AS process control message - fault	$$BlockComment$$ External message 6

Explanation:

$$BlockComment$$: content of the instance-specific comment

Associated values for the message instance `MsgEvId1`

Associated value	Block parameters
1	BatchName
2	StepNo

Associated value	Block parameters
3	BatchID
4	ExtVa104
5	ExtVa105
6	ExtVa106
7	ExtVa107
8	ExtVa108
9	ExtVa109
10	ExtVa110

The associated values 4 ... 8 are allocated to the parameters ExtVa104 ... ExtVa110, which you are free to use. You can find additional information on this in the "Process Control System PCS 7 - Engineering System" manual.

Associated values for the message instance MsgEvId2

Associated value	Block parameters
1	BatchName
2	StepNo
3	BatchID
4	ExtVa204
5	ExtVa205
6	ExtVa206
7	ExtVa207
8	ExtVa208
9	ExtVa209
10	ExtVa210

The associated values 4 ... 8 are allocated to the parameters ExtVa204 ... ExtVa2010, which you are free to use. You can find additional information on this in the "Process Control System PCS 7 - Engineering System" manual.

18.1.6 MonAn08 I/Os

Input parameters

Parameter	Description	Data type	Default
DelayT1 … DelayT8	Delay time for coming for Limit 1 ... 8 monitoring [s]	REAL	0.0
DelayT1_G … DelayT8_G	Delay time for going for Limit 1 ... 8 monitoring [s]	REAL	0.0
Mode1 ... Mode8	Mode limit 1 ... 8	STRUCT	
Mode1.Bit0 ... Mode8.Bit0	0=Low limit, 1=High limit	BOOL	0

Parameter	Description	Data type	Default
`Mode1.Bit1 ... Mode8.Bit1`	1=Status limit (purple)	BOOL	0
`Mode1.Bit2 ... Mode8.Bit2`	1=Alarm limit (red)	BOOL	0
`Mode1.Bit3 ... Mode8.Bit3`	1=Warning limit (yellow)	BOOL	0
`Mode1.Bit4 ... Mode8.Bit4`	Reserved	BOOL	0
`Mode1.Bit5 ... Mode8.Bit5`	Reserved	BOOL	0
`Mode1.Bit6 ... Mode8.Bit6`	Reserved	BOOL	0
`Mode1.Bit7 ... Mode8.Bit7`	Reserved	BOOL	1
`Mode1.Bit8 ... Mode8.Bit8`	Reserved	BOOL	0
`PV_Lim1 ... PV_Lim8 *`	Limit 1 ... 8	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#FF
`PV_Lim1En ... PV_Lim8En *`	Enable limit 1 ... 8	BOOL	1
`PV_Lim1MsgEn ... PV_Lim8MsgEn *`	Enable message for limit 1 ... 8	BOOL	1
`Restart`	Manual restart	BOOL	0
`SmoothTi`	Delay time [s]	REAL	1.0

The configured colors are only evaluated in the limit value view if the "User-configurable message classes" function is not active.

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
`CSF_Out`	External error	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`GrErr`	Group error for limit monitoring, CSF and ExtMsgx The output parameter GrpErr summarizes the following signals of a block and provides them: • Limit value monitoring of the process value (MonAn08: Mode1.Bit1 <> 1)	BOOL	0
`GradHDnMsgEnOut`	1 = Gradient (high) message for positive changes activated	BOOL	1

Parameter	Description	Data type	Default
GradHUpMsgEnOut	1 = Gradient (high) message for negative changes activated	BOOL	1
GradLMsgEnOut	1 = Gradient (low) message activated	BOOL	1
Mode1_Out … Mode8_Out	Mode limit 1 ... 8	BYTE	16#00
MsgSup	Message suppressed	BOOL	0
PV_Lim1Act ... PV_Lim8Act	Limit 1 ... 8 active	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
PV_Lim1Out ... PV_Lim8Out	Limit 1 ... 8	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
PV_LimStatus	Status mask of outputs PV_LimxAct	BYTE	16#00
SimAct	1=Simulation active	BOOL	0

18.1.7 MonAn08 block diagram

The block has no block diagram.

18.1.8 Operator control and monitoring

18.1.8.1 MonAn08 views

MonAn08 block views

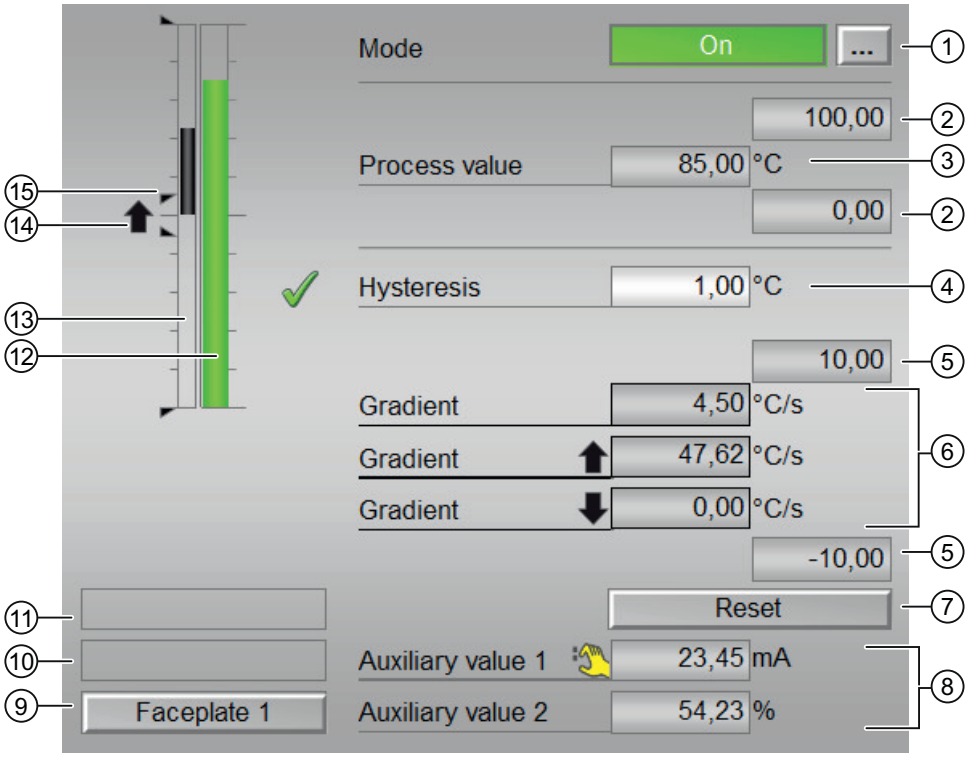
The MonAn08 block provides the following views:

- MonAn08 standard view (Page 932)
- Message view
- Limit view of MonAn08 (Page 935)
- Trend view
- MonAn08 parameter view (Page 938)
- MonAn08 preview (Page 939)
- Memo view
- Batch view
- MonAn08 block icon (Page 940)

For general information on the faceplate and block icon, refer to the "Structure of the faceplate" and "Block icon structure" in the APL online help.

18.1.8.2 MonAn08 standard view

MonAn08 standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

For more information on changing the operating mode, refer to the "Switching operating states and operating modes" section in the APL manual.

(2) High and low scale range for the process value

These values provide information on the display range for the bar graph of the process value. The scale range is specified in the engineering system.

(3) Display of the process value including signal status

This area shows the current process value with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the PV_Out parameter:

Changing the display of the text for "Process value"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the parameter (PV_Out) to what you want to later see in runtime.
- In the "Identifier" column, change the default setting for the input parameter (PV_Out) to what you later want to see in runtime.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding PV_Out input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (PV_Out), this text is displayed even then the "Identifier" text field of the input parameter (PV_Out) is not empty. If the "OS additional text" field of the (PV_Out) input parameter is empty, the "Identifier" text of the (PV_Out) input parameter is displayed.

(4) Hysteresis

In this area, you can enter the hysteresis for the limits of the process value.

Refer to the "Changing values" section in the APL manual.

The operator control permissions for the value are also displayed. They depend on the configuration in the engineering system (ES) that applies to this operator authorization:

- Green check mark: The OS operator can control this parameter
- Gray check mark: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- Red cross: The OS operator is generally prohibited from accessing this parameter due to the configured AS operator control permissions (OS_Perm or OS1Perm).

(5) High and low scale range for the gradient value

These values provide information on the display range for the bar graph of the gradient. The scale range corresponds to 10% of the scale range for the process value: For example, once you have specified a process value scale range of 0 to 100, the scale range of the gradient will be automatically set to a value between -10 and 10.

The current gradient value is displayed when one of the following monitoring functions is activated:

- Gradient monitoring for positive changes `(GradHUpEn = 1)`
- Gradient monitoring for negative changes `(GradHDnEn = 1)`
- Gradient monitoring `(GradLEn = 1)`

(6) Display of the gradient

This area shows the current, minimum and maximum gradient value and the rise and fall of the value. This display of the minimum and maximum gradient value functions like a min/max pointer.

The current gradient value is displayed when one of the following monitoring functions is activated:

- Gradient monitoring for positive changes `(GradHUpEn = 1)`
- Gradient monitoring for negative changes `(GradHDnEn = 1)`
- Gradient monitoring `(GradLEn = 1)`

The maximum peak gradient value is displayed when the gradient monitoring is activated for positive changes `(GradHUpEn = 1)`

The minimum peak gradient value is displayed when the gradient monitoring is activated for negative changes `(GradHDnEn = 1)`

(7) Reset the peak values of the gradient

You can use this button to reset the maximum or minimum peak value of the gradient (`PV_GradPP` and `PV_GradNP` output parameters).

The button is displayed when gradient monitoring is activated for positive `(GradHUpEn = 1)` or negative changes `(GradHDnEn = 1)`.

(8) Display of auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system. You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(9) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(10) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

(11) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

(12) Bar graph for the "Process value"

This area shows the current "Process value" in the form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

(13) Bar graph for the gradient

This area shows the current gradient value in the form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

The bar graph is displayed when gradient monitoring is activated for positive `(GradHUpEn = 1)` or negative changes `(GradHDnEn = 1)`.

(14) Display of the gradient

This display indicates the movement of the gradient up (↑) or down (↓).

Gradient monitoring is displayed when the gradient value PV_Grad ≠ 0 and one of the following monitoring functions is activated:

- Gradient monitoring for positive changes `(GradHUpEn = 1)`
- Gradient monitoring for negative changes `(GradHDnEn = 1)`
- Gradient monitoring `(GradLEn = 1)`

(15) Display of the limits in the bar graph

This area shows you the configured limits. For more on this, see section Limit view of MonAn08 (Page 935).

The display only appears when the bar for the gradients is also displayed.

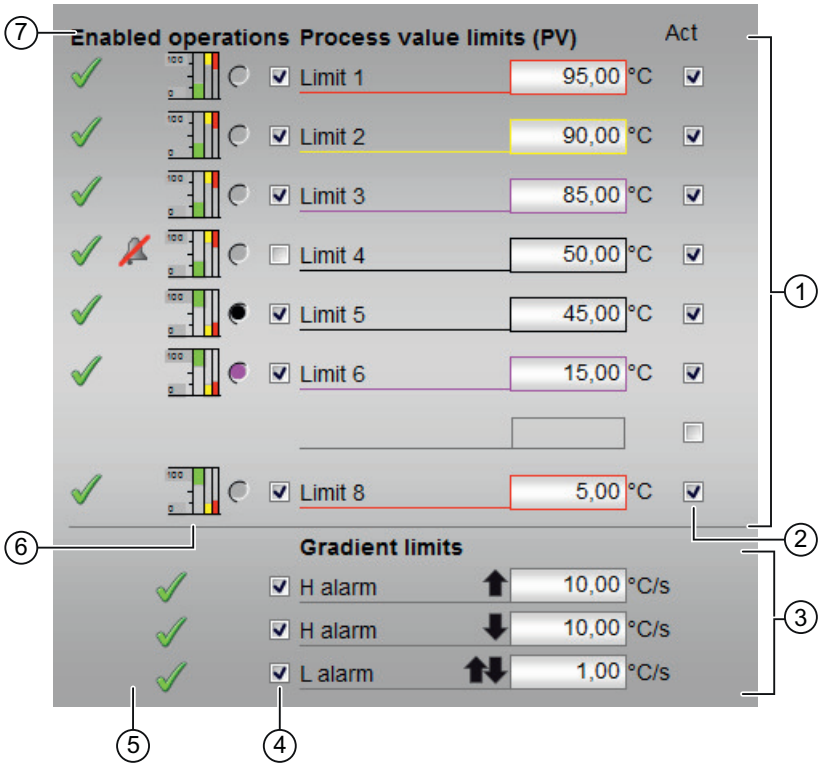
18.1.8.3 Limit view of MonAn08

Limit view of MonAn08L

Several values are set in this view by default:

- Process value limits
- Gradient limits

The toolbar of the faceplate and the block icon indicate when the limits are reached or violated.



(1) Process value limits

In this area, you can enter the limits for the process value. Refer to the "Changing values" section in the APL manual.

You can change the following limits:

- Limit 1 - 8: The text in the limits can be adapted in the ES in the ID of the `PV_LimX` I/O.

The display, display icons and colors of the limits depend on the configuration in the ES (`ModeX`).

Changing the display of the text for "Process value"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the parameter (`PV_LimX`) to what you want to later see in runtime.
- In the "Identifier" column, change the default setting for the input parameter (`PV_LimX`) to what you later want to see in runtime.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding `PV_LimX` input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (`PV_LimX`), this text is displayed even then the "Identifier" text field of the input parameter (`PV_LimX`) is not empty. If the "OS additional text" field of the (`PV_LimX`) input parameter is empty, the "Identifier" text of the (`PV_LimX`) input parameter is displayed.

(2) Enable messages

You can enable/disable a limit by setting the check mark.

(3) Gradient limits

You can enter the gradient limits in this area. Refer to the "Changing values" section in the APL manual.

You can change the following limits:

- "H alarm ⇑": Gradient for the high slope for positive changes
- "H alarm ⇓": Gradient for the high slope for negative changes
- "L alarm ⇑ ⇓": Gradient for the low slope (absolute)

(4) Suppress messages

You can enable/disable messages by setting the check mark.

(5) Message suppression/delay

Message suppression indicates whether the suppression of the associated message in the AS block is activated with the `xx_MsgEn` parameter. The output of messages is not suppressed when the block is installed (all `xx_MsgEn` parameters are preset to 1). Messages can only be output if limit monitoring of the additional analog value has been enabled.

Alarm delays are also displayed in this position. For more on this, see section "Application area for the alarm delays" in the APL manual.

(6) Limit display

The display, display icons and colors of the limits depend on the configuration in the ES (ModeX).

(7) "Enable operations"

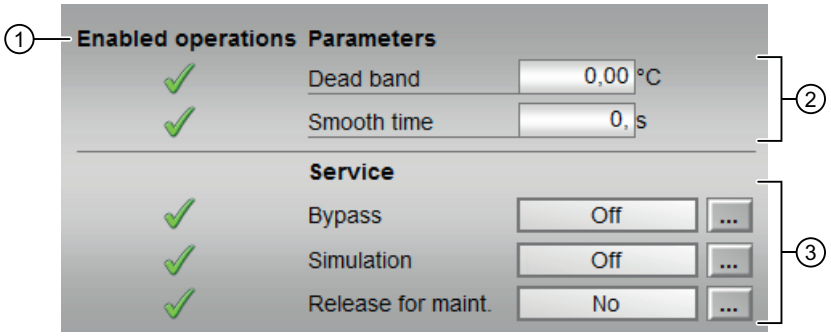
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

18.1.8.4 MonAn08 parameter view

MonAn08 parameter view



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

(2) Parameter

You can change the following parameters in this area:

- "Deadband"
- "Smoothing time"

You can find additional information on this in the section "Changing values" in the manual of the APL.

(3) Service

You can select the following functions in this area:

- "Bypass"
- "Simulation"
- "Release for maintenance"

Refer to the "Switching operating states and operating modes" section in the APL manual.

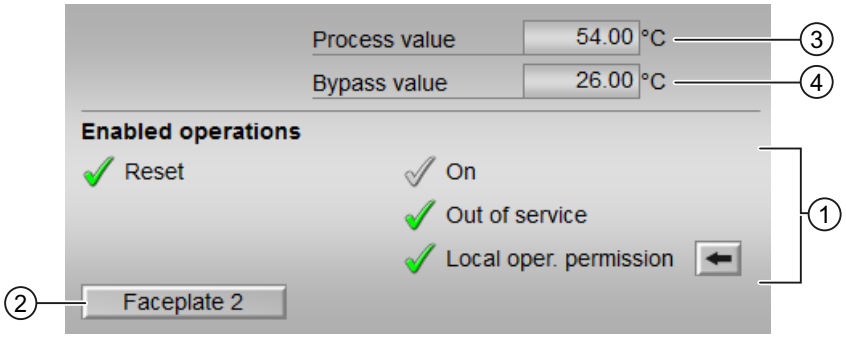
You can find information on this area in the section:

- "Bypass signals"
- "Simulate signals"
- "Release for maintenance"

in the APL manual.

18.1.8.5 MonAn08 preview

MonAn08 preview



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Symbols for enabled operations:

- **Green check mark**: The OS operator may access this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- Reset: You can reset the peak value of the gradient.
- "On": You can switch to "On" operating mode.

- "Out of service": You can switch to "Out of service" operating mode.
- "Local operator authorization": Use the ← button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(2) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(3) Process value

This area shows the real process value (`PV`).

(4) Bypass value

The bypass value (`BypPV`) is displayed in this area.

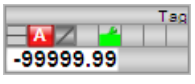
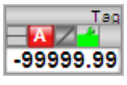
18.1.8.6 MonAn08 block icon

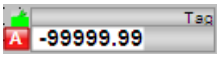
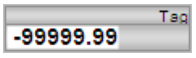
MonAn08 block icons

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes
- Signal status, release for maintenance
- Memo display
- Process value

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	

Icons	Selection of the block icon in CFC	Special features
	3	
	4	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

18.2 MonAnDi - Analog measured value monitoring with binary limits

18.2.1 MonAnDi description

Object name (type + number) and family

Type + number: FB 1104

Family: Monitor

Number of process objects: 1 PO

Area of application of MonAnDi

The block is used for the following areas of application:

- Monitoring of an analog process value for warning limits (high/low) as analog and binary value.
- Monitoring of an analog process value for alarm limits (high / low) as analog and binary value.
- Monitoring of the gradient of an analog process value.

The block is used to monitor a measured value (analog signal) in terms of the following limit value pairs:

Startup characteristics

Following startup, the messages are suppressed for the number of cycles assigned in the `RunUpCyc` parameter. A startup can be simulated via the `Restart` input.

Time response

The block must be called by means of a cyclic interrupt OB. The sampling time of the block is entered in the SampleTime parameter.

Called blocks

FC369	SelST16
SFC6	RD_SINFO
SFB35	ALARM_8P

Calling OBs

In the same OB with and after the block whose measured value is to be monitored. Also in OB100.

Status word assignment for the parameter Status1

For a description of the individual parameters, refer to the section MonAnDi I/Os (Page 951).

Status bit	Parameter
0	Allocated
1	BatchEn
2	SimAct
3	OosAct.Value
4	OosLi.Value
5	Not used
6	OnAct.Value
7	BypassAct.Value
8	Display of BypassAct.Value in the faceplate (display and operator controls) and block icon
9	Not used
10	SimLiOp
11	DelPV_AH (PV_AH delay active)
12	DelPV_WH (PV_WH delay active)
13	DelPV_WL (PV_WL delay active)
14	DelPV_AL (PV_AL delay active)
15 - 16	Not used
17	SumDelay (delay active)
18 - 28	Not used
29	Not used
30	Auxiliary value 1 visible
31	Auxiliary value 2 visible

Status word assignment for the parameter Status2

Status bit	Parameter
0	MsgLock
1	PV_AH_Act
2	PV_WH_Act
3	PV_WL_Act
4	PV_AL_Act
5	PV_AH_MsgEn
6	PV_WH_MsgEn
7	PV_WL_MsgEn
8	PV_AL_MsgEn
9	= Feature.Bit29
10	PV_AH_En
11	PV_WH_En
12	PV_WL_En
13	PV_AL_En

Status bit	Parameter
14 - 16	Not used
17	PV_HystOp (1 = PV_Hyst.ST = 16#FF)
18	PV_AH_LimOp (1 = PV_AH_Lim.ST = 16#FF)
19	PV_WH_LimOp (1 = PV_WH_Lim.ST = 16#FF)
20	PV_WL_LimOp (1 = PV_WL_Lim.ST = 16#FF)
21	PV_AL_LimOp (1 = PV_AL_Lim.ST = 16#FF)
22 - 30	Not used
31	MS_RelOp

Status word assignment for the parameter Status3

Status bit	Parameter
0 - 18	Not used
19	GradHUpAct.Value
20	GradHDnAct.Value
21	GradLAct.Value
22	GradHUpEn
23	GradHDnEn
24	GradLEn
25	GradHUpMsgEn
26	GradHDnMsgEn
27	GradLMsgEn
28	0 = Falling measured value 1 = Rising measured value
29 - 31	Not used

Status word assignment for the parameter Status4

Status bit	Parameter
0	Effective signal 1 of the message block connected via EventTsIn
1	Effective signal 2 of the message block connected via EventTsIn
2	Effective signal 3 of the message block connected via EventTsIn
3	Effective signal 4 of the message block connected via EventTsIn
4	Effective signal 5 of the message block connected via EventTsIn
5	Effective signal 6 of the message block connected via EventTsIn
6	Effective signal 7 of the message block connected via EventTsIn
7	Effective signal 8 of the message block connected via EventTsIn
8 - 23	Effective signal 9..16 of the message block connected via EventTsIn
24 - 31	Not used

18.2.2 MonAnDi operating modes

The block does not have any operating modes.

18.2.3 MonAnDi functions

PV_Out output

The measured analog value appears at the PV_Out output. The analog value (PV) is specified by the process.

Measured value monitoring

The block monitors the measured value connected to the PV input. The respective upward or downward violation of a limit or triggering of a limit input is indicated at a corresponding output and signaled, if applicable (see Message behavior (Page 949)). Monitoring of individual limits can be activated with the inputs PV_AH_En, PV_WH_En, PV_WL_En and PV_AL_En.

Quality code

The .ST structure elements of the parameters for the measured value PV plus the external message signals contain the quality code.

The following quality code data is evaluated:

- Quality Code = 16#60: Simulation on driver block active
- Quality Code = 16#80: Valid value
- Quality Code <> 16#60 or <> 16#80: Invalid value

Deadband

To suppress values that fluctuate around the zero point, this block has the deadband standard function.

Low pass filter

The block smooths the input variable (I/O) after the delay 1. order time delay. This delay time can be configured (SmoothTi connection). The block works according to the following formula:

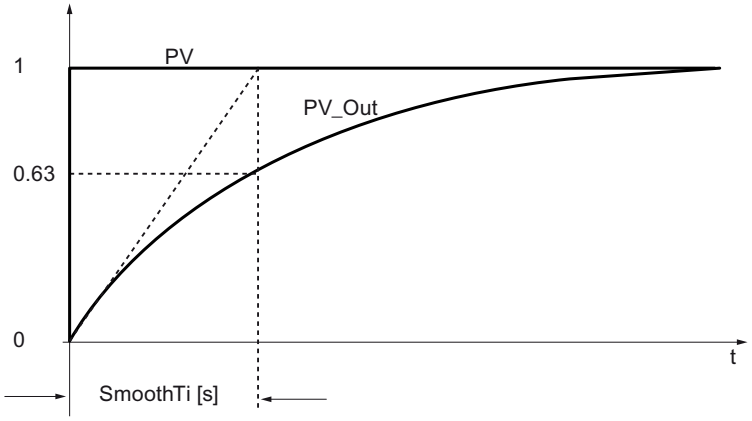
Where:

$$PV_Out_{(n)} = PV_{(n)} + (PV_Out_{(n-1)} - PV_{(n)}) \cdot e^{\left(\frac{-SampleTime}{SmoothTi}\right)}$$

- PV_Out = output value
- SmoothTi = delay time

- `SampleTime` = sampling time
- `PV` = input value

The formula only applies to `SmoothTi` > 0. If `SmoothTi` = 0, the input is passed directly to the output. If the input value is outside the REAL range limits, the calculation is stopped. If the input value is outside the range limits again, the calculation is resumed automatically.



Release for maintenance

This block features the standard function "Release for maintenance".

Simulating signals

This block features the standard function "Simulate signals".

Bypass function

This block has the standard function, bypass signals.

Interconnection of limits

The `PV_Hyst`, `PV_AH_Lim`, `PV_WH_Lim`, `PV_WL_Lim` and `PV_AL_Lim` limits can be interconnected. If they are not interconnected (`PV_ .ST` = 16#FF), the operator can change the corresponding limit.

Gradient monitoring

The `PV_Grad` gradient is calculated with a delay function configured via `LagTime`. This setting smooths the jumps of the PV input value with gradient calculation.

The gradient peak values are output at the output parameters `PV_GradNP` (negative slope) and `PV_GradPP` (positive slope). They are reset as soon as the operator issues the reset command.

The slope of the PV_Grad gradient can be monitored for the following limits:

- Limit (high) for positive gradients (GradHUpLim)
- Limit (high) for negative gradients (GradHDnLim)
- Limit (low) for absolute gradients (GradLLim)

The individual monitoring functions are activated at the corresponding "Enable" parameters, e.g. GradHUpEn for activating the gradient high limit for positive gradients (GradHUpLim).

When the values you have defined are reached or exceeded, this is indicated at the corresponding "Active" output parameters, e.g. with GradHUpAct = 1 for the limit (high) for positive gradients.

The "EventTs" block of the APL is used in order to output messages for these alarms. Connect the EventTsOut output of the EventTs-block to the EventTsIn input of the MonAnDi-Bausteins to link the blocks.

To generate messages, the "Active" output parameters and message enables of MonAnDi must be connected to the InX and MsgEnX inputs of the EventTs-block and the message texts and classes must be appropriately configured:

Output limit active	Enable message	Message class (APL)	Message text (APL)
GradHUpAct	GradHUpMsgEnOut	Alarm - high	$$BlockComment$$ limit (high) for positive gradient violation
GradHDnAct	GradHDnMsgEnOut	Alarm - high	$$BlockComment$$ limit (high) for negative gradient violation
GradLAct	GradLMsgEnOut	Alarm - low	$$BlockComment$$ limit (low) for absolute encoder violated

The messages can be enabled and disabled at the block MonAnDi:

- Message for alarms (high) for positive gradients: GradHUpMsgEn
- Message for alarms (high) for negative gradients: GradHDnMsgEn
- Message for alarms (low) for absolute gradients: GradLMsgEn

Operator authorizations

This block provides the standard function, operator authorizations. Information is available in the section *Operator control permissions* in the *APL Function manual*.

The block has the following permissions for the OS_Perm parameter:

Bit	Function
0	Not used
1	1 = Operator can switch to "On" mode
2	Not used
3	1 = Operator can switch to "Out of service" mode
4	1 = Operator can enable the bypass functionality
5	1 = Operator can disable the bypass functionality
6	Not used
7	1 = The operator can reset the highest values

Bit	Function
8	1 = Operator can activate/deactivate messages via `GradHUpMsgEn`
9	1 = Operator can activate/deactivate messages via `GradHDnMsgEn`
10	1 = Operator can activate/deactivate messages via `GradLMsgEn`
11	1 = Operator can enable the Simulation function
12	1 = Operator can activate the Release for maintenance function
13	1 = Operator can change the limit (`PV`) for the high alarm
14	1 = Operator can change the limit (`PV`) for the high warning
15	Not used
16	1 = Operator can change the hysteresis
17	1 = Operator can change the limit ((`PV`) for the low alarm
18	1 = Operator can change the limit (`PV`) for the low warning
19	1 = The operator can change the value for the gradient high limit for positive slopes (`GradHUpLim`)
20	1 = The operator can change the value for the gradient high limit for negative slopes (`GradHDnLim`)
21	1 = The operator can change the value for the gradient low limit for negative slopes (`GradLLim`)
22	1 = Operator can change the "Dead band" parameter `DeadBand`
23	1 = Operator can activate / deactivate the high alarm messages with `PV_AH_MsgEn`
24	1 = Operator can activate / deactivate the high warning messages with `PV_WH_MsgEn`
25	1 = Operator can activate / deactivate the low alarm messages with `PV_AL_MsgEn`
26	1 = Operator can enable/disable the low warning messages with `PV_WH_MsgEn`
27 - 29	Not used
30	1 = Operator can enter `SmoothTi`
31	Not used

The block has the following permissions for the `OS1Perm` parameter:

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
1	See APL function: "Reaction to the out of service operating mode"
8	See APL function: "Separate delay times for each alarm"
9	See APL function: "The substitute value is enabled if the block is bypassed"
19	See APL function: "Changes signal status of outputs in OOS"
22	See APL function: "Update acknowledgment and error status of the message call"
23	See APL function: "Limits disabled in faceplate"
24	See APL function: "Enable local operating permission"
25	See APL function: "Suppression of all messages"

Bit	Function
26	See APL function: "Reaction of the switching points in the "Out of service" operating mode"
28	See APL function: "Deactivate switching points"
29	See APL function: "Signal limit violation"

18.2.4 MonAnDi error handling

The following errors can be displayed for this block:

- Error numbers
- Control system fault (CSF)

Control system fault (CSF)

An external signal can be activated via the CSF input. If this signal = 1, a control system fault is triggered. You can find more information on this in the APL documentation.

Overview of error numbers

The ErrorNum I/O can be used to output various error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
30	The value of PV can no longer be displayed in the REAL number field.

18.2.5 MonAnDi messaging

Message behavior

The MonAnDi block uses the ALARM8_P block for generating messages.

The following messages can be generated for this block:

- Control system fault
- Process messages
- Instance-specific messages

The signaling of the messages can be delayed depending on Feature.Bit8 by the time configured in the DelayT, DelayT_G, PV_AH_DC, PV_AH_DG, PV_WH_DC, PV_WH_DG, PV_WL_DC, PV_WL_DG, PV_AL_DC, PV_AL_DG parameters.

If Feature.Bit8 = 0, any incoming alarm and warning is delayed by DelayT and each outgoing alarm and warning is delayed by DelayT_G.

If `Feature.Bit8 = 1`, the separate delay time (`PV_AH_DC`, `PV_AH_DG`, `PV_WH_DC`, `PV_WH_DG`, `PV_WL_DC`, `PV_WL_DG`, `PV_AL_DC`, `PV_AL_DG`) is used for each incoming and outgoing alarm and warning.

Control system fault

You can interconnect an external fault (signal) to the input parameter `CSF`. If this signal changes to `CSF` = 1, a control system fault is triggered (MsgEvId1, SIG 5).

Messages

Message instance	Message ID	Message class	Event
`MsgEvId`	SIG 1	High alarm	$$BlockComment$$ PV - High alarm limit violated
	SIG 2	High warning limit	$$BlockComment$$ PV - High warning limit violated
	SIG 3	Warning - low	$$BlockComment$$ PV - Low warning limit violated
	SIG 4	Alarm - low	$$BlockComment$$ PV - Low alarm limit violated
	SIG 5	AS process control message - fault	$$BlockComment$$ External error has occurred
	SIG 6	< No message >	$$BlockComment$$ Simulation active
	SIG 7	AS process control message - fault	$$BlockComment$$ External message 1
	SIG 8	AS process control message - fault	$$BlockComment$$ External message 2

Explanation:

$$BlockComment$$: content of the instance-specific comment

Associated values for the message instance `MsgEvId`

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`ExtVa104`
5	`ExtVa105`
6	`ExtVa106`
7	`ExtVa107`
8	`ExtVa108`

Associated value	Block parameters
9	ExtVa109
10	ExtVa110

The associated values 4 ... 8 are allocated to the parameters ExtVa104 ... ExtVa110, which you are free to use. You can find additional information on this in the "Process Control System PCS 7 - Engineering System" manual.

18.2.6 MonAnDi I/Os

Input parameters

Parameter	Description	Data type	Default
DelayT	Delay time for Limit Monitoring Coming [s]	REAL	0.0
DelayT_G	Delay time for Limit Monitoring Going [s]	REAL	0.0
MsgEvId	Message event ID	DWORD	16#00000014
PV_AH	1=Limit AH	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
PV_AH_DC	PV-Alarm Delay Time Coming High [s]	REAL	0.0
PV_AH_DG	PV-Alarm Delay Time Going High [s]	REAL	0.0
PV_AH_En	PV - High Alarm Enable	BOOL	1
PV_AH_Lim *	Limit value AH	STRUCT • Value:REAL • ST: BYTE	- • 10 • 16#FF
PV_AL	1=Limit AL	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
PV_AL_DC	PV-Alarm Delay Time Coming Low [s]	REAL	0.0
PV_AL_DG	PV-Alarm Delay Time Going Low [s]	REAL	0.0
PV_AL_En	PV - Low Alarm Enable	BOOL	1
PV_AL_Lim *	Limit value AL	STRUCT • Value:REAL • ST: BYTE	- • 10 • 16#FF
PV_WH	1=Limit WH	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
PV_WH_DC	PV-Warning Delay Time Coming High [s]	REAL	0.0
PV_WH_DG	PV-Warning Delay Time Going High [s]	REAL	0.0
PV_WH_En	PV - High Warning Enable	BOOL	1

Parameter	Description	Data type	Default
PV_WH_Lim *	Limit value WH	STRUCT • Value:REAL • ST: BYTE	- • 10 • 16#FF
PV_WL	1=Limit WL	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
PV_WL_DC	PV-Warning Delay Time Coming Low [s]	REAL	0.0
PV_WL_DG	PV-Warning Delay Time Going Low [s]	REAL	0.0
PV_WL_En	PV - Low Warning Enable	BOOL	1
PV_WL_Lim *	Limit value WL	STRUCT • Value:REAL • ST: BYTE	- • 10 • 16#FF
SmoothTi	Delay time [s]	REAL	1.0

Output parameters

Parameter	Description	Data type	Default
CSF_Out	External error	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
ExtMsg1Out	Message input 1 - output	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
ExtMsg2Out	Message input 2 - output	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
GradHDnMsgEnOut	1 = Gradient (high) message for positive changes activated	BOOL	1
GradHUpMsgEnOut	1 = Gradient (high) message for negative changes activated	BOOL	1
GradLMsgEnOut	1 = Gradient (low) message activated	BOOL	1
GrErr	Group error for limit monitoring, CSF and ExtMsgx The output parameter GrpErr summarizes the following signals of a block and provides them: • Limit value monitoring of the process value • External process control error • Freely selectable messages ExtMsg1..2	BOOL	0
MsgAckn	Message acknowledge	WORD	16#0000

Parameter	Description	Data type	Default
MsgErr	Message error	BOOL	0
MsgStat	Message status	WORD	16#0000
MsgSup	Message suppressed	BOOL	0
PV_AH_Out	Limit value AH	STRUCT • Value:REAL • ST: BYTE	- • 10 • 16#FF
PV_AL_Out	Limit value WL	STRUCT • Value:REAL • ST: BYTE	- • 10 • 16#FF
PV_WH_Out	Limit value WH	STRUCT • Value:REAL • ST: BYTE	- • 10 • 16#FF
PV_WL_Out	Limit value WL	STRUCT • Value:REAL • ST: BYTE	- • 10 • 16#FF
SimAct	1=Simulation active	BOOL	0

* Values can be written back to these inputs during processing of the block by the block algorithm.

18.2.7 MonAnDi block diagram

The block has no block diagram.

18.2.8 Operator control and monitoring

18.2.8.1 MonAnDi views

MonAnDi block views

The MonAnDi block provides the following views:

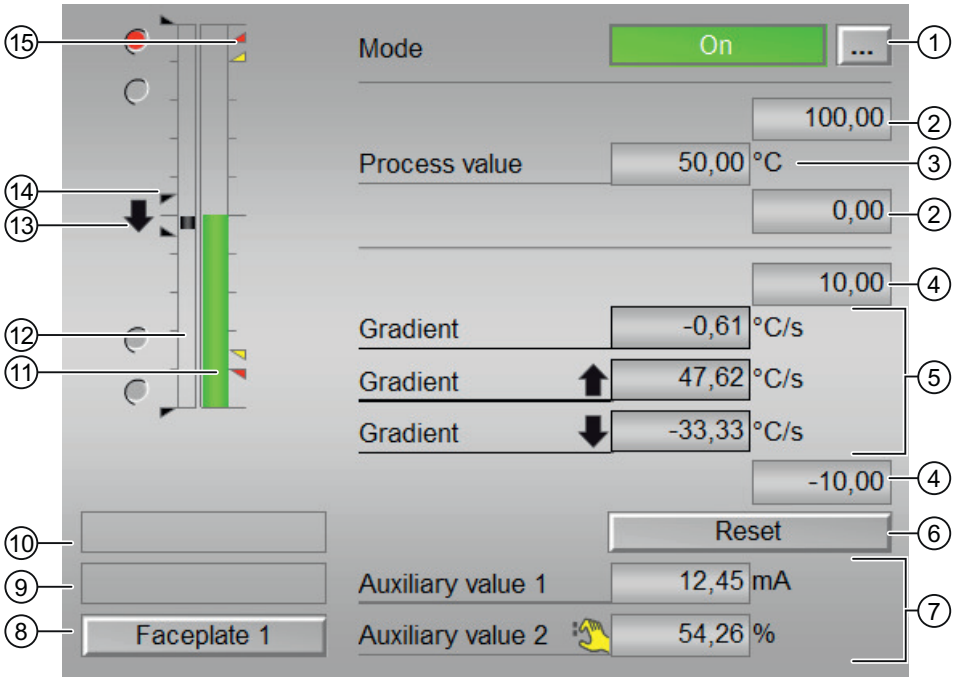
- MonAnDi standard view (Page 954)
- Message view
- Limit view of MonAnDi (Page 958)
- Trend view
- MonAnDi parameter view (Page 959)
- MonAnDi preview (Page 961)
- Memo view

- Batch view
- MonAnDi block icon (Page 962)

For general information on the faceplate and block icon, refer to the "Structure of the faceplate" and "Block icon structure" in the APL online help.

18.2.8.2 MonAnDi standard view

MonAnDi standard view



(1) Display and switchover of the operating mode

This area provides information on the currently valid operating mode. The following operating modes can be shown here:

- On
- Out of service

For more information on changing the operating mode, refer to the "Switching operating states and operating modes" section in the APL manual.

(2) High and low scale range for the process value

These values provide information on the display range for the bar graph of the process value. The scale range is specified in the engineering system.

(3) Display of the process value including signal status

This area shows the current process value with the corresponding signal status.

If text is configured for this command, it is displayed as additional text and button label for command selection. For additional information, refer to the "Labeling of buttons and texts" section in the APL manual.

You can change the text for the process value at the `PV_Out` parameter:

Changing the display of the text for "Process value"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the parameter (`PV_Out`) to what you want to later see in runtime.
- In the "Identifier" column, change the default setting for the input parameter (`PV_Out`) to what you later want to see in runtime.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding `PV_Out` input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (`PV_Out`), this text is displayed even then the "Identifier" text field of the input parameter (`PV_Out`) is not empty. If the "OS additional text" field of the (`PV_Out`) input parameter is empty, the "Identifier" text of the (`PV_Out`) input parameter is displayed.

(4) High and low scale range for the gradient value

These values provide information on the display range for the bar graph of the gradient. The scale range corresponds to 10% of the scale range for the process value: For example, once you have specified a process value scale range of 0 to 100, the scale range of the gradient will be automatically set to a value between -10 and 10.

The current gradient value is displayed when one of the following monitoring functions is activated:

- Gradient monitoring for positive changes `(GradHUpEn = 1)`
- Gradient monitoring for negative changes `(GradHDnEn = 1)`
- Gradient monitoring `(GradLEn = 1)`

(5) Display of the gradients

This area shows the current, minimum and maximum gradient value and the rise and fall of the value. This display of the minimum and maximum gradient value functions like a min/max pointer.

The current gradient value is displayed when one of the following monitoring functions is activated:

- Gradient monitoring for positive changes `(GradHUpEn = 1)`
- Gradient monitoring for negative changes `(GradHDnEn = 1)`
- Gradient monitoring `(GradLEn = 1)`

The maximum peak gradient value is displayed when the gradient monitoring is activated for positive changes `(GradHUpEn = 1)`

The minimum peak gradient value is displayed when the gradient monitoring is activated for negative changes `(GradHDnEn = 1)`

(6) Reset the peak values of the gradient

You can use this button to reset the maximum or minimum peak value of the gradient (`PV_GradPP` and `PV_GradNP` output parameters).

The button is displayed when gradient monitoring is activated for positive `(GradHUpEn = 1)` or negative changes `(GradHDnEn = 1)`.

(7) Display of auxiliary values

You can use this area to display two auxiliary values that have been configured in the engineering system. You can find additional information on this in the section "Displaying auxiliary values" in the manual of the APL.

(8) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(9) Display area for block states

This area provides additional information on the operating state of the block:

- "Maintenance"

(10) Display area for block states

This area provides additional information on the operating state of the block:

- "Simulation"

(11) bar graph for the "Process value"

This area shows the current "Process value" in the form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

(12) bar graph for the gradients

This area shows the current gradient value in the form of a bar graph. The visible area in the bar graph depends on the configuration in the engineering system (ES).

The bar graph is displayed when gradient monitoring is activated for positive `(GradHUpEn = 1)` or negative changes `(GradHDnEn = 1)`.

(13) Display of the gradients

This display indicates the movement of the gradient up (↑) or down (↓).

Gradient monitoring is displayed when the gradient value PV_Grad ≠ 0 and one of the following monitoring functions is activated:

- Gradient monitoring for positive changes `(GradHUpEn = 1)`
- Gradient monitoring for negative changes `(GradHDnEn = 1)`
- Gradient monitoring `(GradLEn = 1)`

(14) Display of the limits in the bar graph

This area shows you the configured limits. For more on this, see section Limit view of MonAnDi (Page 958).

The small colored triangles indicate the limits in the bar graph. The status of the four binary limits is also displayed as circles next to the bar graph.

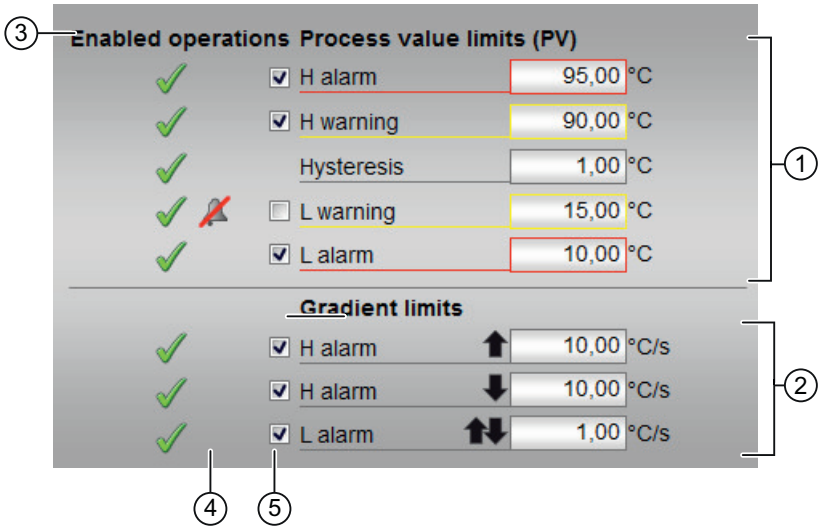
The display only appears when the bar for the gradients is also displayed.

(15) Limit display

These colored triangles show you the configured limits in the respective bar graph.

18.2.8.3 Limit view of MonAnDi

Limit view of MonAnDi



(1) Process value limits

In this area, you can enter the limits for the process value. Refer to the "Changing values" section in the APL manual.

You can change the following limits:

- "H alarm": Alarm high
- "H warning": Warning high
- "Hysteresis"
- "L warning": Low warning
- "L alarm": Low alarm

(2) Gradient limits

You can enter the gradient limits in this area. Refer to the "Changing values" section in the APL manual.

You can change the following limits:

- "H alarm ⇑": Gradient for the high slope for positive changes
- "H alarm ⇓": Gradient for the high slope for negative changes
- "L alarm ⇑ ⇓": Gradient for the low slope (absolute)

(3) "Enable operations"

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

(4) Message suppression/delay

Message suppression indicates whether the suppression of the associated message in the AS block is activated with the `xx_MsgEn` parameter. The output of messages is not suppressed when the block is installed (all `xx_MsgEn` parameters are preset to 1). Messages can only be output if limit monitoring of the additional analog value has been enabled.

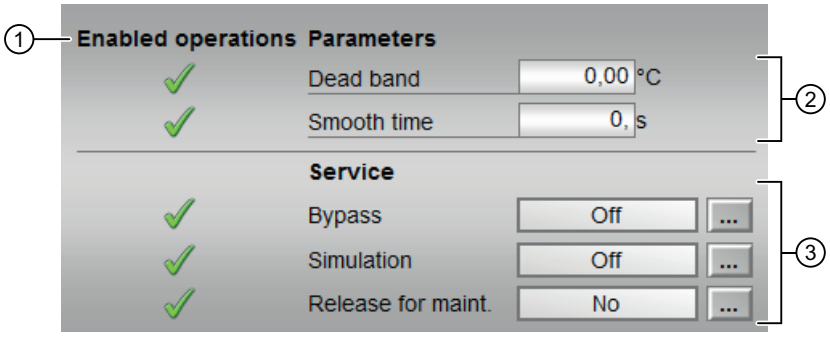
Alarm delays are also displayed in this position. For more on this, see section "Area of application of the alarm delays" in the APL manual.

(5) Suppress messages

You can enable/disable messages by setting the check mark.

18.2.8.4 MonAnDi parameter view

MonAnDi parameter view



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).

(2) Parameter

You can change the following parameters in this area:

- "Deadband"
- "Smoothing time"

You can find additional information on this in the section "Changing values" in the manual of the APL.

(3) Service

You can select the following functions in this area:

- "Bypass"
- "Simulation"
- "Release for maintenance"

Refer to the "Switching operating states and operating modes" section in the APL manual.

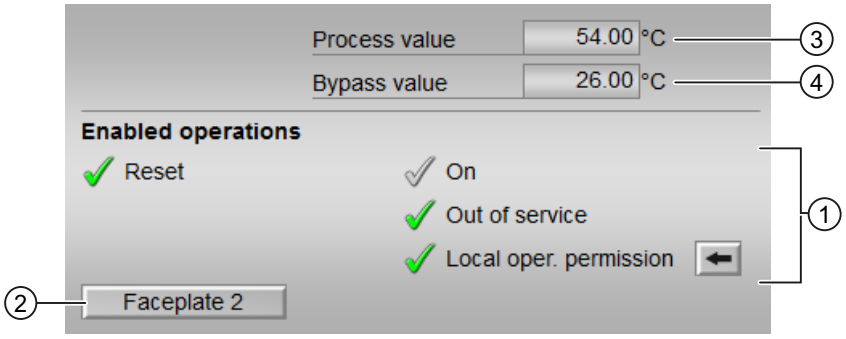
You can find information on this area in the section:

- "Bypass signals"
- "Simulate signals"
- "Release for maintenance"

in the APL manual.

18.2.8.5 MonAnDi preview

MonAnDi preview



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Symbols for enabled operations:

- **Green check mark**: The OS operator may access this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- Reset: You can reset the peak value of the gradient.
- "On": You can switch to "On" operating mode.
- "Out of service": You can switch to "Out of service" operating mode.
- "Local operator authorization": Use the ← button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(2) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

(3) Process value

This area shows the real process value (`PV`).

(4) Bypass value

The bypass value (BypPV) is displayed in this area.

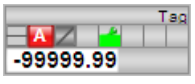
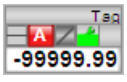
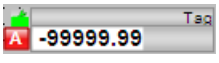
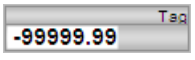
18.2.8.6 MonAnDi block icon

MonAnDi block icons

A variety of block icons are available with the following functions:

- Process tag type
- Limits (high/low)
- Violation of alarm, warning, and tolerance limits as well as control system faults
- Operating modes
- Signal status, release for maintenance
- Memo display
- Process value

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Special features
	1	
	2	
	3	
	4	

Additional information on the block icon and the control options in the block icon is available in the following sections:

- "Configure the block icons"
- "Block icon structure"
- "Operation via the block icon"

in the APL manual.

18.3 MonDin – Monitoring of a digital input

18.3.1 Description of 'MonDin'

Area of application and operating principle

This function block is used to monitor a digital process tag. The digital value to be monitored is interconnected to the `In` input parameter. The signal transitions (1 → 0 or 0 → 1) can be delayed with one configurable or interconnectable time each. In addition this block allows the values to be simulated from the operating level.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).

For the MonDin block, the demo project of the PCS 7 Industry Library contains a template for process tag types as an example with an application scenario for this block.

Startup characteristics

Use the `Feature` bit 'Set startup characteristics' to define the startup characteristics of this block.

The messages are suppressed after startup for the number of cycles set at `RunUpCyc` .

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'MonDin' (Page 967)".

Status bit	Parameter
0	`BatchOcc`
1	`BatchEn`
2	`SimEnOut`
3	`SimOnOut`
4	`SimOnOp`
5	`SimOut`
6	`InNorm.Value`
7	`xFeatHighActive`

18.3.2 Operating modes of 'MonDin'

The block can be operated using the following operating modes:

- •"On"
- •"Out of service"

"On"

General information on the "On" mode is available in the "On" section of the APL documentation.

"Out of service"

You will find general information about the out of service mode in the APL documentation, section "Out of service".

18.3.3 Functions of 'MonDin'

Delaying on and off switching functions

In normal operation the output Out is set in accordance with the input In and can be delayed to a positive or negative edge. The active signal level can be set by means of a "Feature Bit" (1=High level, 0=Low level active). The delay times are displayed and can be parameterized in the OS faceplate for the user with corresponding user rights.

You set delay times for setting the output using the input parameter `Out_DC` or Out_DG :

- `Out_DC`: delay time [s] for rising edges (0 - 1 edge)
- `Out_DG`: delay time [s] for falling edge (1 - 0 edge)

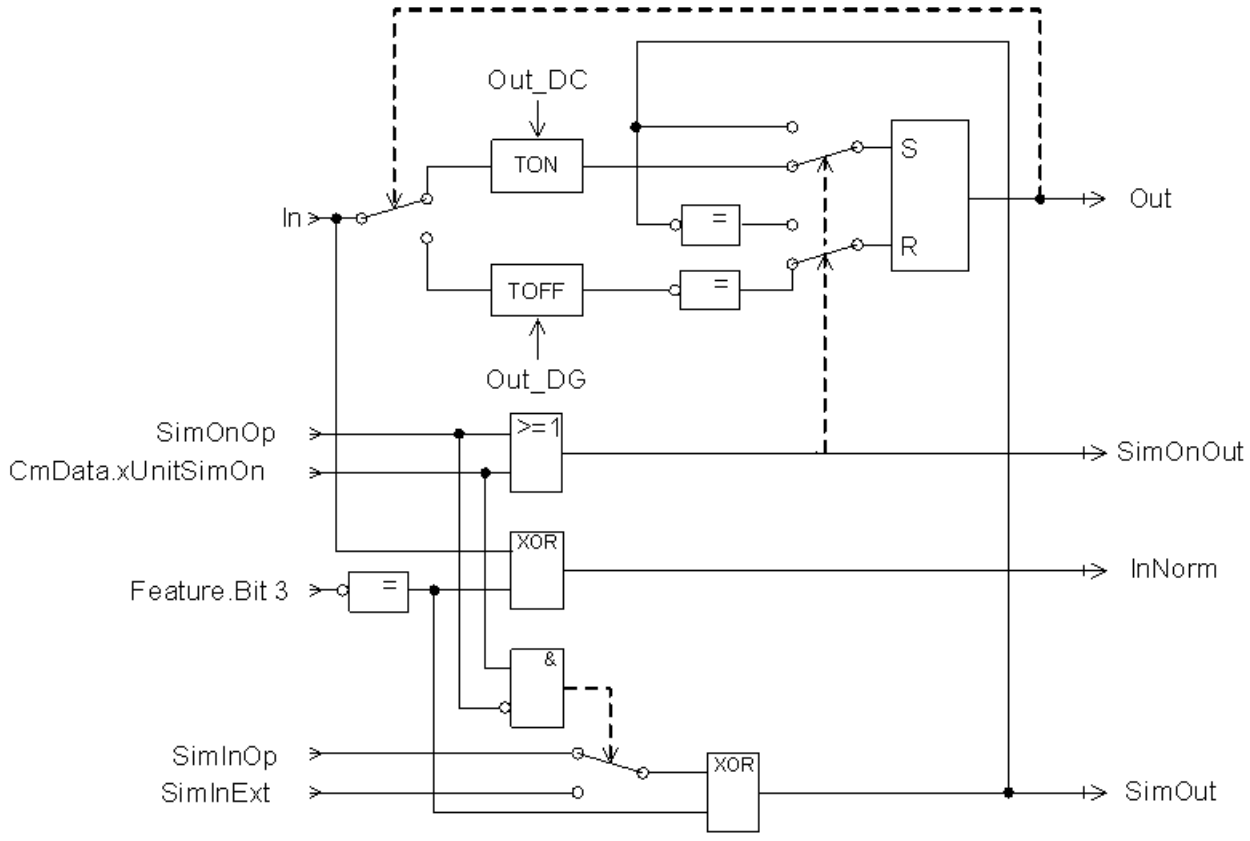
The `Out` output parameter is set after expiration of the delay time.

You disable this function if you set the value of the respective parameter to 0 seconds.

The output

`InNorm` indicates the input value in a standardized form, meaning always 1-active.

Block diagram



Further functions of Din

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwUnitID`
- `OvwBatchInfo`
- `OvwSimInfo`

You will find more information on this subject in the section AUTOHOTSPOT

Call additional faceplates

This block includes the APL standard function "Call additional faceplates".

Forming the signal status for blocks

This block includes the APL standard function "Form and output signal status for blocks".

The worst signal status `ST_Worst` for the block is generated from the following parameters: `In.ST`

Release for maintenance

This block includes the APL standard function "Release for maintenance".

Simulating signals

This block includes the APL standard function "Simulating signals". In addition a higher-level simulation operation is available that can be specified by the unit or the EQM respectively.

- The higher-level enable (`CmData.xSimEnCM`) takes precedence over the local enable (`OS_Perm / Bit 2`) with regard to the simulation enable.
- The local simulation (`SimOnOp`) takes precedence over the higher-level one (`CmData.xSimOnParent`) with regard to the simulation activation.
- The process value `In` is simulated in simulation operation. This is specified either by the operator / input parameter `SinInOp` or in the case of higher-level simulation by the input `SimInExt` .
- Activation of the higher-level simulation can be carried out via the faceplates of the unit (`UnitIf`) or the equipment module (`EqmIf`).
- If the overwrite input `OvwSimInfo=1` is set, the simulation parameters are derived from the inputs `SimEnExt, SimOnExt` instead of the structure `CmData` . The simulation value is also determined here by the input `SimInExt` .
- The output `SimOut` indicates the simulation value in a standardized form, meaning always 1-active.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the Feature parameter is available in the "APL documentation" in the section "Configurable functions with the Feature I/O". The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	Start-up without call in OB100 (default = 0)
1	Enable switchover to "Out of service" via `OosLi` (default = 0)
2	Active signal level of the input `In` (1=High level, 0=Low level active)
3	Reserve
...	...

Operator permissions

This block includes the APL standard function "Operator permissions".

The block has the following permissions for the OS_Perm parameter:

Status bit	Parameter
0	1= Operator can switch to "On" mode
1	1 = Operator can switch to "Out of service" mode
2	1 = Operator can enable the "Simulation" function
3	1 = Operator can change the ON-delay time OUT_DC
4	1 = Operator can change the OFF-delay time OUT_DG
5	Reserve
...	...

18.3.4 Connections of 'MonDin'

Input parameters

Parameter	Description	Type	Default
In	Digital input value	DigVal	
Out_DC	Out delay time for coming signal [s]	AnaVal	
Out_DG	Out delay time for going signal [s]	AnaVal	
CMData	Control and status information	BlUdtCmData	
OvwUnitID	1=Overwrite CmData unit ID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
OvwSimInfo	1=Overwrite CmData Simulation presets	BOOL	
SimEnExt	Enable simulation (overwritable CmData element)	BOOL	
SimOnExt	Turn on/off unit simulation (overwritable CmData element)	BOOL	
SimInExt	Simulation value during unit simulation	BOOL	
UnitID	Unit ID (overwritable CmData element)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIdLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
BatchStepNoLi	Batch step number (overwritable CmData element)	DWORD	
MS_RelOp	Operator input for MS release, 1: MS release requirement	BOOL	
RunUpCyc	Number of cycles for which this block keeps in start up mode	INT	3
SampleTime	Sampling time [s]	REAL	1.0
UserStatus	User status bits	BYTE	
CSF	Control System Fault message - external error	BOOL	
OnOp	1=On mode: On mode by operator	BOOL	
OosOp	1=Oos mode: Oos mode by operator	BOOL	

Parameter	Description	Type	Default
OosLi	1=Oos mode: Oos mode switch via link	DigVal	
SimOnOp	Simulation on/off by operator	BOOL	
SimInOp	Operator simulation value	BOOL	
SelFp1/2	Select Faceplate 1/2	BOOL	
OS_PERM	Operator Permissions	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

Output parameters

Parameter	Description	Type	Default
MS_Release	1= Maintenance status release	DigVal	
InNorm	Normalized input signal	DigVal	
UnitIDOut	Unit ID	DWORD	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING[32]	
StepNo	Batch step number	DWORD	
SimEnOut	Simulation is enabled	BOOL	
SimOnOut	Simulation is active	BOOL	
SimOut	Simulation value	BOOL	
P_Rst	Reset signal (pulse)	BOOL	
EffCSF	Effective CSF (zero during simulation)	BOOL	
OutAct_DC	On delay time in seconds	REAL	
OutAct_DG	Off delay time in seconds	REAL	
OnAct	On mode is active	DigVal	
OosAct	Out of service is active	DigVal	
OS_PermOut	Operator permission (OS info)	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission (OS info)	DWORD	16#FFFFFFFF
OSPermLnk	Operator permission (OS info)	DWORD	
OpSt_Out	Enabled operator stations	DWORD	
Status1	Status1 data	DWORD	
ST_Worst	Worst signal status	BYTE	16#80

18.3.5 Operator control and monitoring

18.3.5.1 Symbol



Tag name
Status bar

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	Empty	
2	Empty	
3	Operating mode	
		= "On",
		= "Out of service"
4	Empty	
5		= Enable maintenance operation
6		= Memo is available
7	Empty	

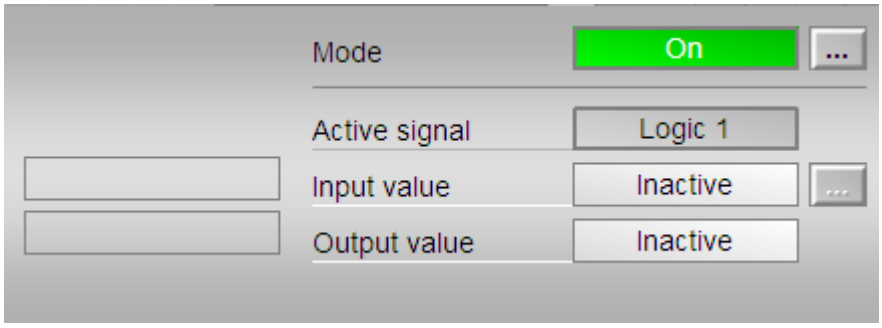
18.3.5.2 Faceplate

Views of the MonDin block

The block MonDin provides the following views:

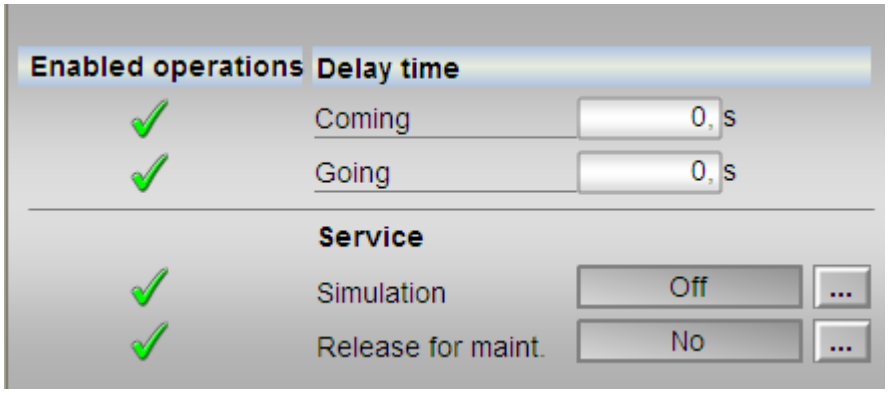
Symbol	View	Information
	Standard view	See below.
	Parameter view	See below.
	Trend view	Standard trend view of all the blocks of the Advanced Process Library
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

Standard view of the MonDin



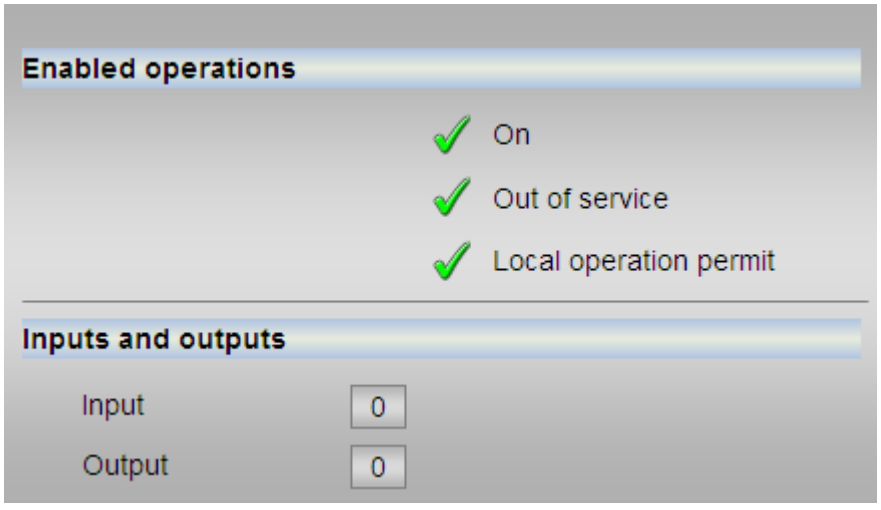
Explanation of the screen shot	
(1)	**Displaying and switching the operating mode**
	This area provides information on the currently valid operating mode. The following operating modes can be shown and executed here: • On • Out of service
(2)	**Display input value (process value) and output value**
	This section shows you the state of the input and output value as well as the active signal value (in accordance with Feature Bit 2). If the block is in simulation, you can enable or disable the process value. To do so, click the button to open the operating area. If texts are configured for these commands, it is displayed as additional text and as button labels for command selection. For further information please refer to the APL section "Labeling of buttons and texts".
(3)	**Navigation button to the standard view of any faceplate**
	Use this navigation button to reach the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

Parameter view of the MonDin



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration in the engineering system (ES) that applies to this block. Symbols for operator control enable: • Green check mark: The OS operator can control this parameter • Gray check mark: The OS operator cannot control this parameter at this time due to the process • Red cross: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).
(2)	**Delay time**
	Enter here the delay time by which the output should be set. Enter delay times here for positive ("coming", 0-to-1 edge) and negative ("going", 1-to-0 edge) edges.
(3)	**Service**
	You can activate the following functions in this area: • "Simulation" • "Release for maintenance" You can find information on this section in the APL sections "Simulation of signals" and "Release for maintenance".

Preview of MonDin



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration in the engineering system (ES) that applies to this block (parameter `OS_PERM`). Symbols for operator control enable: • Green check mark: The OS operator can control this parameter • Gray check mark: The OS operator cannot control this parameter at this time due to the process • Red cross: The OS operator cannot control this parameter due to the configured operator permissions The following operator control enables are shown here: • "On": You can switch to "On" operating mode. • "Out of service": You can switch to "Out of service" operating mode. • "Local operator permission": You can switch to the "Local operation" operating mode.
(2)	**Inputs and outputs**
	The status and state of the signal input and output are displayed here.

18.4 MonMof – Combined monitoring function

18.4.1 Description of 'MonMof'

Area of application and operating principle

This function block serves to monitor digital and / or analog inputs with the option of an alarm and acknowledgement function. The input values can be operated or interconnected, with the switchover being carried automatically with the interconnection in CFC.

The logic operation at the output `Out` can also be simulated from the operating level or from external signals (input signal or `CmData` structure of a unit).

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).

For the BlMof block, the demo project of the PCS 7 Industry Library contains a template for process tag types as an example with an application scenario for this block.

Startup characteristics

Use the Feature Bit 'Set startup characteristics' to define the startup characteristics of this block.

The messages are suppressed after startup for the number of cycles set at `RunUpCyc` .

Status word allocation ‚Status1'

For a description of the individual parameters, see the section "Connections of 'MonMof' (Page 982)".

Status bit	Parameter
0	`BatchOcc`
1	`BatchEn`
2	`SimEnOut`
3	`SimOnOut`
4	`SimOnOp`
5	`SimOut`
6	`Reserve`
7	`Reserve`
8	`OosAct.Value`
9	`OnAct.Value`
10	`OutSel.Value`
11	`Out.Value`
12	`OutRestart.Value`

Status bit	Parameter
13	OutAck.Value
14	OutRel.Value
15	OosLi.Value
16	Reserve
...	…
31	MS_RelOp

18.4.2 Operating modes of 'MonMof'

The block can be operated using the following standard operating modes:

- "On"
- "Out of service"

"On"

General information on the "On" mode is available in the "On" section of the APL documentation.

"Out of service"

You will find general information about the out of service mode in the APL documentation, section "Out of service".

18.4.3 Functions of Mon'Mof'

The block monitors a binary input (InDig) and up to 4 analog inputs (InAna1 … InAna4) and outputs the logic operation result of the monitoring to a binary output Out .

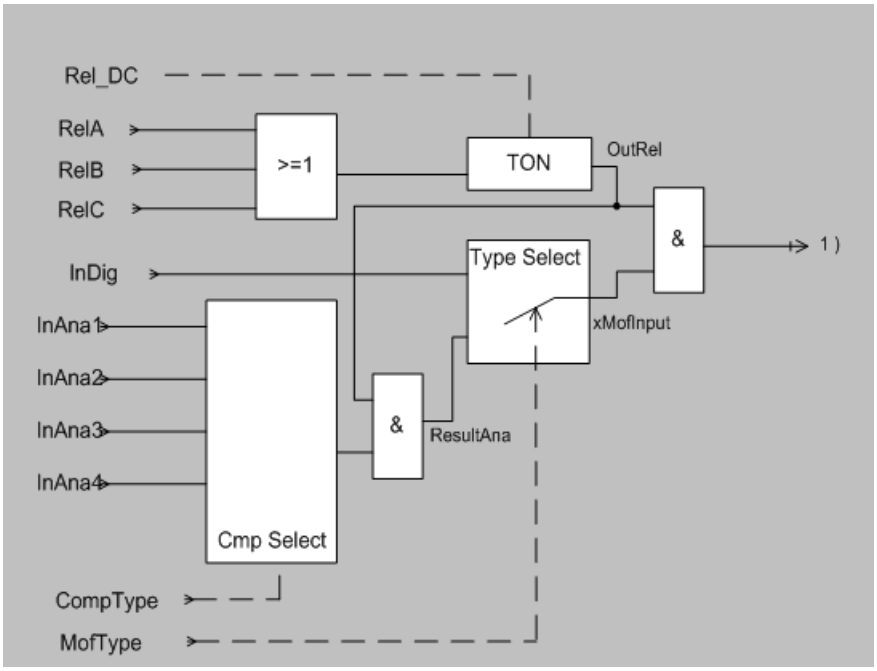
The rising and the falling edge of the logic operation can be delayed at the output Out with the times Out_DC and Out_DG .

The logic operation itself is formed depending on:

- The set MonMof function at the input MofType
- The set comparison function at the input CompType

The MonMof functions as well as the configurable comparison functions are shown below:

Block diagram (1)



Output OutRel:

The output parameter **OutRel** becomes active if at least one enable input (`RELA` or `RELB` or `RELC`) is set and the configured delay time has REL_DC has expired.

OutRel = TON (RELA or RELB or RELC)

Monitoring type:

The parameter `MofType` can be used to select the combination of digital input `InDig` and analog intermediate result `ResultAna` (block-internally).

Input '**MofType**'	Logic operation of 'TypeSelect' (internal parameter **xMofInput**)
`DiOnly`	`InDig`
`AnOnly`	`ResultAna`
`DiAndAn`	`InDig .AND. ResultAna`
`DiOrAn`	`InDig .OR. ResultAna`

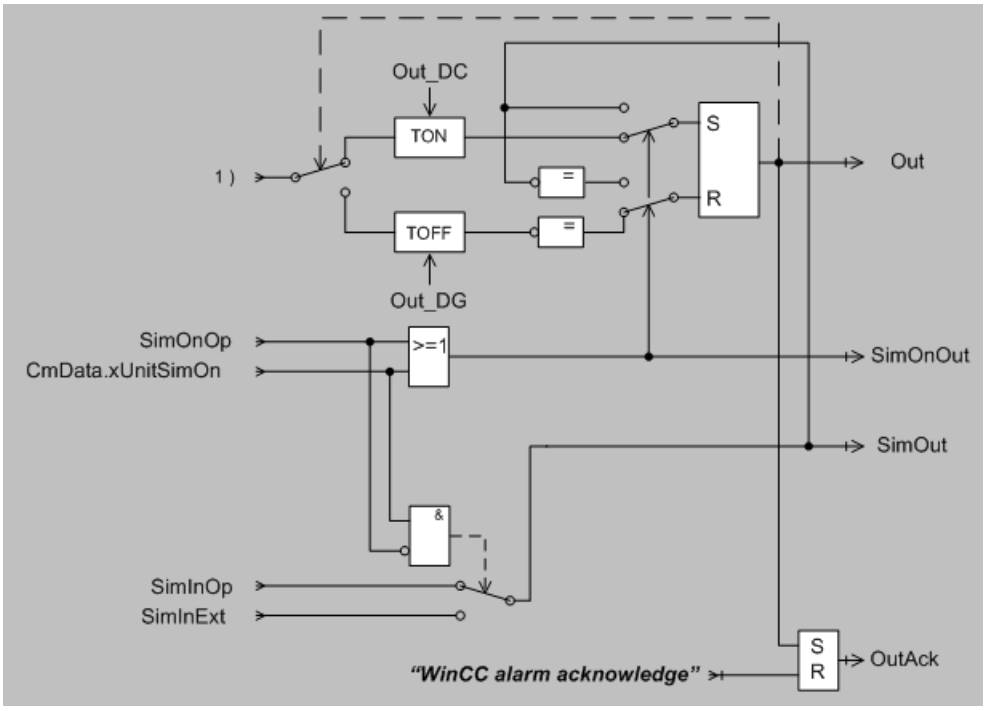
Comparison type

At the input `CompType` you can select the comparison operation between the analog inputs `InAna1 ... InAna4` for the analog intermediate result `ResultAna` in accordance with the following table.

Value CompType	Analog result of logic operation at the output '**ResultAna**'
0	`ResultAna = False`
1	• `SET = InAna1 > InAna2` • `RESET = InAna1 < InAna3` Set takes precedence

Value CompType	Analog result of logic operation at the output ‚ResultAna'
2	• SET = InAna1 > (InAna2 + InAna3) • RESET = InAna1 < (InAna2 - InAna3) Set takes precedence
3	• SET = InAna1 > (InAna2+(InAna2*InAna3)) • RESET = InAna1 < (InAna2-(InAna2*InAna3)) Set takes precedence
4	(InAna1 > InAna2) OR (InAna1 < InAna3)
5	(InAna1 > InAna2) AND (InAna3 > InAna4)
6	• SET = (InAna1 < InAna2) • RESET = (InAna1 > InAna3) Set takes precedence
7	(InAna1 < InAna2) AND (InAna1 < InAna3)
8	(InAna1 > InAna2) AND (InAna1 > InAna3)
9	• SET = InAna1 > (InAna2 + InAna3) • RESET = InAna1 < (InAna2 - InAna4) Set takes precedence
10	InAna1 > InAna2
11	InAna1 >= InAna2
12	InAna1 = InAna2
13	InAna1 <= InAna2
14	InAna1 < InAna2
15	• SET = InAna1 < (InAna2 - InAna3) • RESET = InAna1 > (InAna2 - InAna4) Set takes precedence
0 > CompType > 15	ResultAna = False

Block diagram (2)



Output Out:

In standard operation, the output `Out` is calculated according to the set comparison function in connection with the input values `InDig/InAna1...4`. It can be delayed on a positive edge with the time `Out_DC` or on a negative edge with `Out_DG` . In addition, enabling of the comparison logic can be delayed by using the input parameter `RelA, RelB, RelC`

The delay times are displayed and can be parameterized in the OS faceplate for the user with corresponding user rights.

You set delay times for setting the output using the input parameters:

- `Out_DC`:
  : delay time [s] for rising edges (edge of 0 `-->` 1)
- `Out_DG`:
  : delay time [s] for falling edge (edge of 1 `-->` 0)
- `Rel_DC`:
  delay time [s] for enable / rising edge (edge of 0 `-->` 1)

You disable this function if you set the value of the respective parameter to 0 seconds.

Output OutAck:

`OutAck` serves as additional output with signal memory.

`OutAck` is set at the same time as the output `Out` and remains active until the associated message has been acknowledged at the OS. If no message is enabled or if no acknowledgment is enabled for the configured message class (parameter `MsgClass`, see Messages of 'MonMof' (Page 980)), then `Out` and `OutAck` are identical.

The acknowledgment is enabled with the input parameter `MsgMask` of the type `Word`. The six possible message classes can be activated (Bit = 1) or disabled (Bit = 0) with one bit each (Bit 2^0-2^5). The default value of `MsgMask` is `110011=16#33H`.

In the initial state, this results in the following constellation for the acknowledgement:

- 2^0 High alarm (1) - active
- 2^1 High warning (1) - active
- 2^2 High tolerance (0) - inactive
- 2^3 Low tolerance (0) - inactive
- 2^4 Low warning (1) - active
- 2^5 Low alarm (1) - active

Output OutAckEx

Depending on the activated/deactivated message, `OutAckEx` is derived either from the normal output `Out` or the latching output `OutAck`:

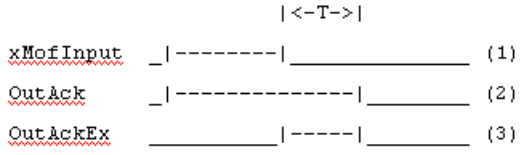
- Without message:
  `MsgClass = 0 --> OutAckEx = Out`
- With message:
  `MsgClass <> 0 --> OutAckEx = OutAck AND NOT xMofInput`

This means that `OutAckEx` is only active if the internal comparison condition `xMofInput` has already "gone" again and the latching output `OutAck` is still `True`, that is, the message has not been acknowledged yet.

Note

During the simulation `OutAckEx` corresponds to the output `OutAck`.

Diagram for output `OutAckEx` when messages are activated:



(1) Comparison condition fulfilled
(2) Result output latching
(3) Comparison condition outgoing and not confirmed

Output OutSel:

Output `OutSel` is set in accordance with the parameters `OutSelSwitch` and `MsgClass` :

- `OutSelSwitch = 0 --> OutSel = Out` (direct value of the monitoring function)
- `OutSelSwitch = 1 --> OutSel = OutAck` (stored value with operator acknowledgement)
- `OutSelSwitch = 2 --> OutSel = OutAckEx` (combined output ... see above)

Further functions of MonMof

In this block the following overwrite functions are implemented with regard to the input selections:

- OvwUnitID
- OvwBatchInfo
- OvwSimInfo

You will find more information on this subject in the section AUTOHOTSPOT

Opening additional faceplates

This block includes the APL standard function "Call additional faceplates".

Forming the signal status for blocks

This block includes the APL standard function "Form and output signal status for blocks".

The worst signal status ST_Worst for the block is formed from the following parameters:

- InDig
- InAna1
- InAna2
- InAna3
- InAna4
- RelA
- RelB
- RelC

Release for maintenance

This block includes the APL standard function "Release for maintenance".

Simulating signals

This block includes the APL standard function "Simulating signals". In addition a higher-level simulation operation is available that can be specified by the unit or the EQM respectively.

- The higher-level enable (CmData.xSimEnCM) takes precedence over the local enable (OS_Perm / Bit 2) with regard to the simulation enable.
- The local simulation (SimOnOp) takes precedence over the higher-level one (CmData.xSimOnParent) with regard to the simulation activation.
- The output value Out is simulated in the local as well as the higher-level simulation operation. This is specified either by the operator / input parameter SinInOp or in the case of higher-level simulation by the input SimInExt.
- Activation of the higher-level simulation can be carried out via the faceplates of the unit (UnitIf) or the equipment module (EqmIf).
- If the overwrite input OvwSimInfo=1 is set, the higher-level simulation is derived from the inputs SimEnExt, SimOnExt instead of the structure CmData . The simulation value is also determined here by the input SimInExt.

Configurable reactions using the Feature parameter

An overview of all the reactions that are provided by the Feature parameter is available in the "APL documentation" in the section "Configurable functions with the Feature I/O". The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	Start-up without call in OB 100 (default = 0)
1	Enable switchover to "Out of service" via `OosLi` (default = 0)
...	Reserve
22	See APL function: "Update acknowledgment and error status of the message call"
...	...

Operator permissions

This block includes the APL standard function "Operator permissions".

The block has the following permissions for the `OS_Perm` parameter:

Status bit	Parameter
0	1= Operator can switch to "On" mode
1	1 = Operator can switch to "Out of service" mode
2	1 = Operator can enable the "Simulation" function
3	1 = Operator can enable the "Release for maintenance" function
4	1 = Operator can change the delay time `OUT_DC`
5	1 = Operator can change the delay time `OUT_DG`
6	1 = Operator can change the delay time `REL_DC`
7	1 = Operator can change the input value ,`InDig`'
8	1 = Operator can change the input value ,`InAna1`'
9	1 = Operator can change the input value ,`InAna2`'
10	1 = Operator can change the input value ,`InAna3`'
11	1 = Operator can change the input value ,`InAna4`'
12	Reserve
13	...

18.4.4 Messages of 'MonMof'

Suppressing messages using the MsgLock parameter

This block includes the APL standard function "Suppressing messages using the parameter MsgLock".

The following messages can be generated for this block:

- Process control fault
- Process messages

Process control fault

You can interconnect an external fault (signal) to the input parameter CSF. If it changes to `CSF = 1`, a process control fault is triggered (`MsgEvId, SIG 8`).

Process messages

When process messages are enabled (`MsgLock=0`) , a message can be generated with the binary output signal Out. One of six default message classes (`SIG1 ... SIG6`) can be generated with this message instance (Alarm 8P block), depending on the application. The message class to be used is configured with the enumeration 'BMS_EventType' at the block parameter `MsgClass` .

Message instance	Message identifier	Message class	Event
MsgEvId	SIG1	Alarm high	$$BlockComment$$ Alarm high
	SIG2	Warning high	$$BlockComment$$ Warning high
	SIG3	Tolerance high	$$BlockComment$$ Tolerance high
	SIG4	Tolerance low	$$BlockComment$$ Tolerance low
	SIG5	Warning low	$$BlockComment$$ Warning low
	SIG6	Alarm low	$$BlockComment$$ Alarm low
	SIG7	<False>	
	SIG8	AS process control message - fault	$$BlockComment$$ External error

Explanation:

$$BlockComment$$: Content of the instance-specific comment

Associated values for reporting instance MsgEvId

Associated value	Block parameters
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`OutSel`
5	`ExtVa105`
6	`ExtVa106`
7	`ExtVa107`
8	`UnitIDOut`

The associated values 5 ... 7 are allocated to the parameters `ExtVa105 ... ExtVa107` and can be used.

18.4.5 Connections of 'MonMof'

Input parameters

Parameter	Description	Type	Default
Out_DC	Out delay time for coming signal [s]	AnaValFF	
Out_DG	Out delay time for going signal [s]	AnaValFF	
Rel_DC	Release delay time [s]	AnaValFF	
RelA	Release monitoring function (ORed with RelB, RelC)	DigValFF	
RelB	Release monitoring function (ORed with RelA, RelC)	DigValFF	
RelC	Release monitoring function (ORed with RelA, RelB)	DigValFF	
MofType	MOF type: • 0= digital • 1= analog • 2= digital AND analog • 3= digital OR analog	INT	
CompType	Comparison type (see table)	INT	
InDig	Digital input value	DigValFF	
InAna1	Analog input value	AnaValFF	
InAna2	Analog input value	AnaValFF	
InAna3	Analog input value	AnaValFF	
InAna4	Analog input value	AnaValFF	
OutSelSwitch	Output selection switch: • 0=Out • 1=OutAck • 2=OutRst	INT	
CMData	Control and status information	BlUdtCmData	
OvwUnitID	1=Overwrite CmData unit ID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
OvwSimInfo	1=Overwrite CmData Simulation presets	BOOL	
SimEnExt	Enable simulation (overwritable CmData element)	BOOL	
SimOnExt	Turn on/off unit simulation (overwritable CmData element)	BOOL	
SimInExt	Simulation value during unit simulation	BOOL	
UnitID	Unit ID (overwritable CmData element)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIdLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
BatchStepNoLi	Batch step number (overwritable CmData element)	DWORD	
MS_RelOp	Operator input for MS release, 1: MS release requirement	BOOL	

Parameter	Description	Type	Default
RunUpCyc	Number of cycles for which this block keeps in start up mode	INT	3
SampleTime	Sampling time [s]	REAL	1.0
UserStatus	User status bits	BYTE	
CSF	Control System Fault message - external error	BOOL	
OnOp	1=On mode: On mode by operator	BOOL	
OosOp	1=Oos mode: Oos mode by operator	BOOL	
OosLi	1=Oos mode: Oos mode switch via link	DigVal	
SimOnOp	Simulation on/off by operator	BOOL	
SimInOp	Operator simulation value	BOOL	
MsgClass	Message class of signal	INT	
MsgLock	Message lock	DigVal	
MsgEvId	Message event ID 01	DWORD	16#FFFFFFFF
MsgMask	Acknowledgement mask (matched for OutAck)	WORD	16#33
ExtVa105	External value 5	ANY	
ExtVa106	External value 6	ANY	
ExtVa107	External value 7	ANY	
SelFp1/2	Select Faceplate 1/2	BOOL	
OS_PERM	Operator Permissions	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

Output parameters

Parameter	Description	Type	Default
MS_Release	1= Maintenance status release	BOOL	
GrpErr	1=Group error is active	DigVal	
Out	Output value monitorfunction	DigVAl	
OutSel	Output value based on selection (OutSelSwitch)	DigVal	
OutAckEx	OutAck combined with input signal	DigVal	
OutAck	Output value combined with message acknowledgement	DigVal	
OutRel	At least one release signal is active	DigVal	
ResultAna	Result Analog value Comparision	DigVal	
SimEnOut	Simulation is enabled	BOOL	
SimOnOut	Simulation is active	BOOL	
SimOut	Simulation value	BOOL	
P_Rst	Reset signal (pulse)	BOOL	
OutAct_DC	On delay time [s]	REAL	
OutAct_DG	Off delay time [s]	REAL	
RelAct_DC	Release delay time [s]	REAL	
OnAct	On mode is active	DigVal	

Parameter	Description	Type	Default
OosAct	Out of service is active	DigVal	
UnitIDOut	Unit ID	DWORD	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING[32]	
StepNo	Batch step number	DWORD	
MsgErr	ALARM_8P: Messaging error	BOOL	
MsgStat	ALARM_8P: Status output	WORD	
MsgAckn	ALARM_8P: ACK_STATE output	WORD	
OS_PermOut	Operator permission (OS info)	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission (OS info)	DWORD	16#FFFFFFFF
OS_PermLnk	Operator permission (OS info)	DWORD	
OpSt_Out	Enabled operator stations	DWORD	
Status1	Status1 data	DWORD	
ST_Worst	Worst signal status	BYTE	16#80

18.4.6 Operator control and monitoring

18.4.6.1 Symbol



Tag name
Status bar

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	Message status	
		AH
		AL
		WH
		WL
		TH
		TL
2	Message status *)	

Position	Content	
3	Operating mode	
		= "On",
		= "Out of service"
4	Empty	
5		Enable maintenance operation
6		Memo is available
7	Empty	

*) For more information on "Message status", see "Functions of the block icons > Block icon structure" chapter of the APL manual.

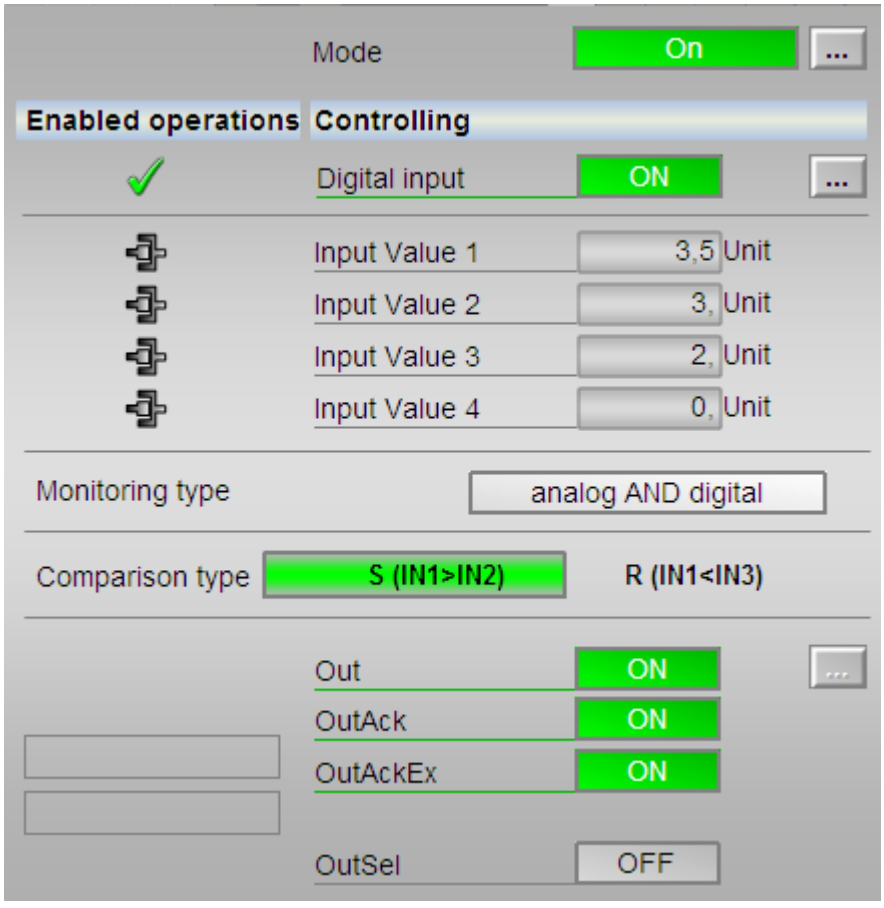
18.4.6.2 Faceplate

Views of the BlMof block

The block BlMof provides the following views:

Sym-bol	View	Information
	Standard view	See below.
	Message view	Standard message view of all the blocks of the Advanced Process Library
	Trend view	Standard trend view of all the blocks of the Advanced Process Library
	Parameter view	See below.
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

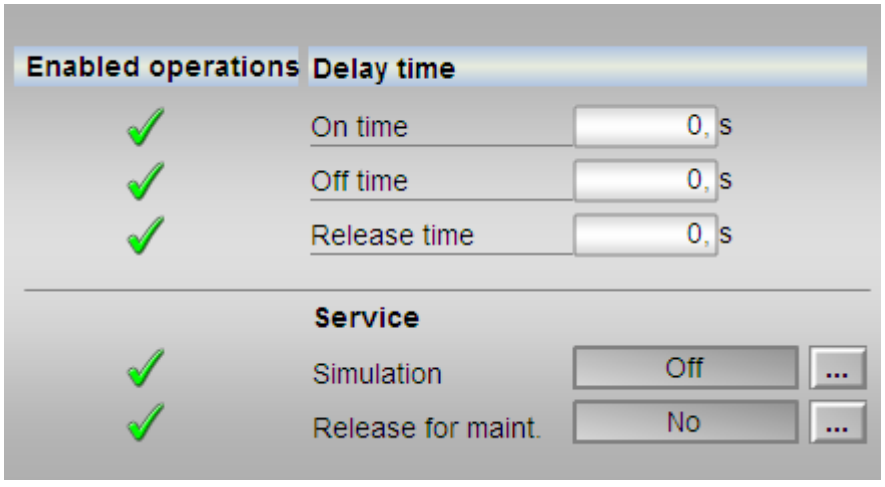
Standard view of the BlMof.



Explanation of the screen shot		
(1)	**Displaying and switching the operating mode**	
	This sections shows you the status of the individual connected process value inputs. The names for the connected parameters can be set using the `s7_string` attribute at the corresponding input parameter. The default text is displayed if nothing is specified.	
	• One of the following states is displayed with the symbol on the left:	
		Parameter is interconnected / cannot be operated
		Parameter is not interconnected / can be operated
	If the parameter is not interconnected, you can change the process value. To do so, click the display to open the control field.	
(2)	**Display of process values**	
	Use this navigation button to reach the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).	
(3)	**Navigation button to the standard view of any faceplate**	
	Use this navigation button to reach the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).	

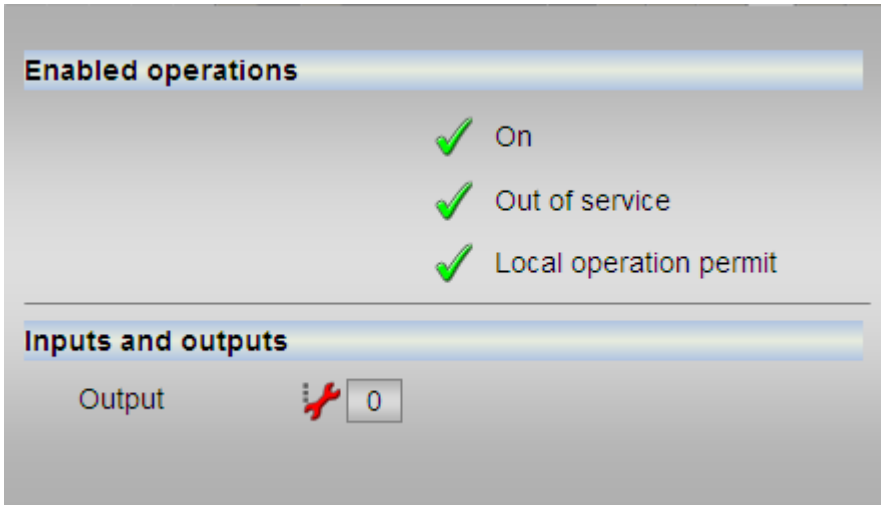
Explanation of the screen shot		
(4)	**Display of the current monitoring type**	
	Here the set monitoring type is displayed in text form referenced to the input values (see table in the section "Functions"). The logic operation status is displayed in text form	
(5)	**Display of the comparison type**	
	Here the comparison function is displayed in text form referenced to the input values (see table in the section "Functions"). The status per condition is displayed as follows:	
	Green	Condition is fulfilled
	Gray	Condition is unfulfilled
(6)	**Display of the result outputs**	
	The various output parameters of the block are displayed in this section (see section "Functions"): • **Output Out** Result of the comparison and logic operation • **OutAck** Result with signal memory • **OutAckEx** Extended result output • **OutSel** Position indication of the output selectable through the input `OutSelSwitch` The simulated output value can be specified with the button in "Local simulation operation". At "Higher-level simulation" the simulation symbol is displayed at the `Out` output. The simulation value is derived from the input parameter `SimInExt` .	

Parameter view of the BlMof



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration in the engineering system (ES) that applies to this block. • Symbols for operator control enable: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`).
(2)	**Delay time**
	Enter here the delay times after which the respective output should be set. • ON time for positive 0 - 1 edge of the logic operation • OFF time for negative 1 - 0 edge of the logic operation • Enable delay time for positive 0 - 1 edge of the enable sum signal
(3)	**Service**
	You can activate the following functions in this area: • "Simulation" (Parameter SimOnOp) • "Release for maintenance" (parameter) You can find information on this section in the APL sections "Simulation of signals" and "Release for maintenance".

Preview of BlMof



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration in the engineering system (ES) that applies to this block (parameter `OS_PERM`). • Symbols for operator control enable: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions • The following operator control enables are shown here: – "On": You can switch to "On" operating mode. – "Out of service": You can switch to "Out of service" operating mode. – "Local operator permission": You can switch to the "Local operation" operating mode.
(2)	**Delay time**
	The status and state of the signal output are displayed here.

Interlock blocks 19

19.1 ILck4x3L - Interlock with 8 input signals and 4x3 logic

19.1.1 ILck4x3L description

Object name (type + number) and family

Type + number: FB 1133

Family: Interlck

Number of process objects: 1 PO

Application area for ILck4x3L

The block is used for the following applications:

- Standardized interlock with display

How it works

The block is used to calculate a standardized interlock that can be displayed on the OS. Maximum 8 input signals can be supplied to the block. These input signals are linked in groups with up to 3 levels with a selectable binary logic. Each group within the logic can be linked to AND, OR or XOR. The groups of levels 1 and 2 can be switched on or off. If incorrect logic is entered, corresponding group will be switched off. If incorrect logic is entered in level 3, Logic XOR will be set. The existing signals are then directly attached to the next group. The signal status of the output signal is also determined. You can assign an analog value with the signal status and unit to each input value for display in the faceplate.

The current state is shown at the `Out` output parameter:

- `Out` = 0: interlock
- `Out` = 1: Good state

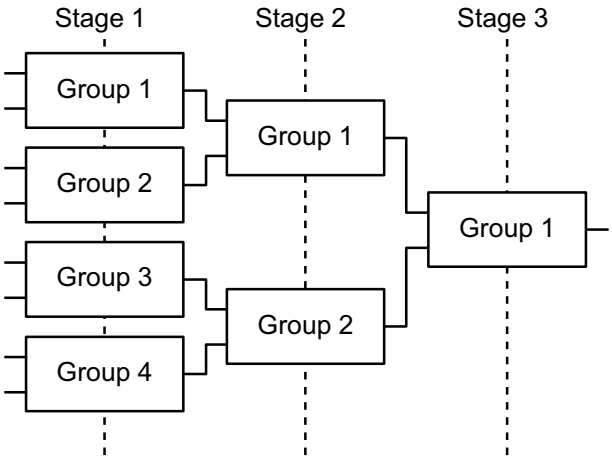


Figure 19-1 How it works

Configuration

Install the block in a cyclic interrupt OB (OB30 to OB38) in the CFC Editor.

Startup characteristics

Use the startup characteristics Feature bit 0 "Set startup characteristics" to define the startup characteristics for this block.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section ILck4x3L I/Os (Page 1004).

Status bit	Parameter
0	1 = `Logic4` = OR
1	1 = `Logic4` = AND
2	1 = `Logic4` = XOR
3	Result of logic operation `Out.Value`
4	1 = At least one input value is excluded. See the section Excluding input values in ILck4x3L functions (Page 997)
5	1 = All input values are excluded
6	1 = No input value is interconnected or `NotUsed.Value`
7	Status display for simulation
8	Status display for not interlocked
9	Status display for interlocked
10	`Feature.Bit5`
11	`Feature.Bit2`
12	`Feature.Bit6`
13	Not used
14	`Feature.Bit15`
15 - 16	Not used

Status bit	Parameter
17	Delay time / Temp. temporary bypass
18	Delay time `Out.Value`
19	Temp. temporary bypass active
20	`BypPreWarnAct.Value`
21	`OS_PermLog.Bit10`
22 - 24	Not used
25	`Out_Byp_Out.Value`
26	1 = Output value is excluded
27 - 31	Not used

Status word assignment for the parameter `Status2`

Status bit	Parameter
0	`In01.Value`
1	`In02.Value`
2	`In03.Value`
3	`In04.Value`
4	`In05.Value`
5	`In06.Value`
6	`In07.Value`
7	`In08.Value`
8 - 31	Not used

Status word assignment for the parameter `Status3`

Status bit	Parameter
0	`InvIn01.Value`
1	`InvIn02.Value`
2	`InvIn03.Value`
3	`InvIn04.Value`
4	`InvIn05.Value`
5	`InvIn06.Value`
6	`InvIn07.Value`
7	`InvIn08.Value`
8 - 31	Not used

Status word assignment for the parameter `Status4`

Status bit	Parameter
0	`In01.Value` with inversion
1	`In02.Value` with inversion
2	`In03.Value` with inversion

Status bit	Parameter
3	`In04.Value` with inversion
4	`In05.Value` with inversion
5	`In06.Value` with inversion
6	`In07.Value` with inversion
7	`In08.Value` with inversion
8 - 15	Not used
16	`BypLi01.Value`
17	`BypLi02.Value`
18	`BypLi03.Value`
19	`BypLi04.Value`
20	`BypLi05.Value`
21	`BypLi06.Value`
22	`BypLi07.Value`
23	`BypLi08.Value`
24 - 31	Not used

Status word assignment for the parameter `Status5`

Status bit	Parameter
0	`BypIn01.Value` or temporary bypass `In01`
1	`BypIn02.Value` or temporary bypass `In02`
2	`BypIn03.Value` or temporary bypass `In03`
3	`BypIn04.Value` or temporary bypass `In04`
4	`BypIn05.Value` or temporary bypass `In05`
5	`BypIn06.Value` or temporary bypass `In06`
6	`BypIn07.Value` or temporary bypass `In07`
7	`BypIn08.Value` or temporary bypass `In08`
8 - 15	Not used
16 - 23	`In01`..`In08` hidden bypass information
24 - 31	Not used

Status word assignment for the parameter `Status6`

Status bit	Parameter
0	`In01` not connected
1	`In02` not connected
2	`In03` not connected
3	`In04` not connected
4	`In05` not connected
5	`In06` not connected
6	`In07` not connected

Status bit	Parameter
7	In08 not connected
8 - 31	Not used

Status word assignment for the parameter Status7

Status bit	Parameter
0	AV01 not connected
1	AV02 not connected
2	AV03 not connected
3	Av04 not connected
4	AV05 not connected
5	AV06 not connected
6	AV07 not connected
7	AV08 not connected
8 - 15	Not used
16	BypIn01
17	BypIn02
18	BypIn03
19	BypIn04
20	BypIn05
21	BypIn06
22	BypIn07
23	BypIn08
24 - 31	Not used

Status word assignment for the parameter Status8

Identical to FirstIn.

Status word assignment for the parameter Status9

Status bit	Parameter
0	1 = Logic1_1 = OR
1	1 = Logic1_1 = AND
2	1 = Logic1_1 = XOR
3	Result of logic operation (Level1/Group1)
4	1 = At least one input value is excluded. (Level1/Group1)
5	1 = All input values are excluded. (Level1/Group1)
6	1 = No input value is interconnected. (Level1/Group1)
7	Status display for simulation (Level1/Group1)
8	Status display for not interlocked (Level1/Group1)
9	Status display for interlocked (Level1/Group1)

Status bit	Parameter
10 - 15	Not used
16	1 = `Logic1_2` = OR
17	1 = `Logic1_2` = AND
18	1 = `Logic1_2` = XOR
19	Result of logic operation (Level1/Group2)
20	1 = At least one input value is excluded. (Level1/Group2)
21	1 = All input values are excluded. (Level1/Group2)
22	1 = No input value is interconnected. (Level1/Group2)
23	Status display for simulation (Level1/Group2)
24	Status display for not interlocked (Level1/Group2)
25	Status display for interlocked (Level1/Group2)
25 - 31	Not used

Status word assignment for the parameter `Status10`

Status bit	Parameter
0	1 = `Logic1_3` = OR
1	1 = `Logic1_3` = AND
2	1 = `Logic1_3` = XOR
3	Result of logic operation (Level1/Group3)
4	1 = At least one input value is excluded. (Level1/Group3)
5	1 = All input values are excluded. (Level1/Group3)
6	1 = No input value is interconnected. (Level1/Group3)
7	Status display for simulation (Level1/Group3)
8	Status display for not interlocked (Level1/Group3)
9	Status display for interlocked (Level1/Group3)
10 - 15	Not used
16	1 = `Logic1_4` = OR
17	1 = `Logic1_4` = AND
18	1 = `Logic1_4` = XOR
19	Result of logic operation (Level1/Group4)
20	1 = At least one input value is excluded. (Level1/Group4)
21	1 = All input values are excluded. (Level1/Group4)
22	1 = No input value is interconnected. (Level1/Group4)
23	Status display for simulation (Level1/Group4)
24	Status display for not interlocked (Level1/Group4)
25	Status display for interlocked (Level1/Group4)
25 - 31	Not used

Status word assignment for the parameter Status13

Status bit	Parameter
0	1 = Logic2_1 = OR
1	1 = Logic2_1 = AND
2	1 = Logic2_1 = XOR
3	Result of logic operation (Level2/Group1)
4	1 = At least one input value is excluded. (Level2/Group1)
5	1 = All input values are excluded. (Level2/Group1)
6	1 = No input value is interconnected. (Level2/Group1)
7	Status display for simulation (Level2/Group1)
8	Status display for not interlocked (Level2/Group1)
9	Status display for interlocked (Level2/Group1)
10 - 15	Not used
16	1 = Logic2_2 = OR
17	1 = Logic2_2 = AND
18	1 = Logic2_2 = XOR
19	Result of logic operation (Level2/Group2)
20	1 = At least one input value is excluded. (Level2/Group2)
21	1 = All input values are excluded. (Level2/Group2)
22	1 = No input value is interconnected. (Level2/Group2)
23	Status display for simulation (Level2/Group2)
24	Status display for not interlocked (Level2/Group2)
25	Status display for interlocked (Level2/Group2)
25 - 31	Not used

19.1.2 ILck4x3L operating modes

ILck4x3L operating modes

This block does not have any operating modes.

19.1.3 ILck4x3L functions

ILck4x3L functions

The functions for this block are listed below.

Logic operators

Use the LogicX_X parameter to specify the logic operator that the block should employ when determining the interlock state.

The name of the `Logic` I/O is always formed so that the first number in the name of the parameter indicates the level and the second number the group for which you specify the logic with this I/O.

Set the following:

- `Logic` = 0: OR
- `Logic` = 1: AND
- `Logic` = 2: XOR
- `Logic` = 3: Do not use group (not for `Logic3`)

The result of the respective groups is displayed at the `OutX_X` output of the block. The name of the `OutX_X` I/O is formed similarly to the `Logic` I/Os: Here too, the first number in the name of the parameter is the level and the second number is the group. The final result of `Logic3` is sent to the `Out` output.

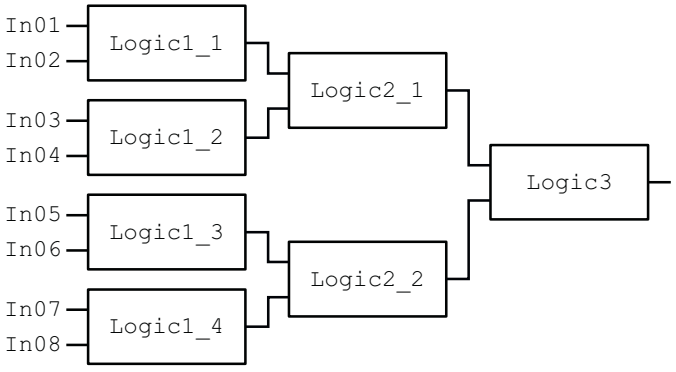


Figure 19-2 Function of the logic I/O of ILck4x3L

If a logic group is selected as logical operator = 3, all existing signals are linked to the logic group of the next level. The logic group in level 3 is always present.

Example:

Setting the input parameters `Logic1_2` = 3 and `Logic2_1` = 3 results in the logic shown below. All other groups are evaluated according to the configured logical operator.

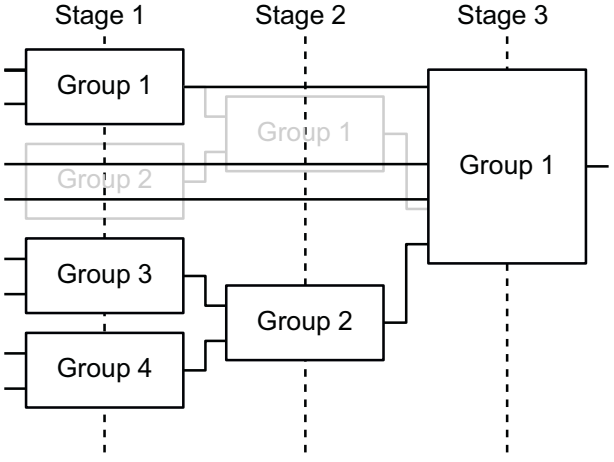


Figure 19-3 Example of a logical operator configured as a "non-existent group"

Automatic reset of the output value

You can use the "Automatic reset of the output value" function (`Feature`.Bit 15) to realize a store function for the output value.

`Feature`**.Bit =0**

The output value is not automatically reset. When the conditions for the interlock are met, the output value is set. The output value is first reset when the condition is no longer fulfilled and the operator has executed a reset.

`Feature`**.Bit =1**

The output value is automatically reset. When the conditions for the interlock are met, the output value is set. The output value is reset as soon as the condition is no longer fulfilled.

Inversion of logic signals

You can invert the input signals by setting the input parameter `InvInx` for the respective input to `Inx` = 1, e.g. for input `In01` set the I/O `InvIn01`.

The inversion is displayed in the faceplate. If you invert signals using any other method, this is not shown in the faceplate.

Bypass

Note

In the ILck4x3L block, exclusion "bypassing) the interlock means that the interlock signal (input signal) is excluded from the logic of the interlock block, in other words, this signal is ignored in the logic operations.

This function can only be executed in the faceplate with "high-level operating permission".

You can disable input signals that are temporarily not to be used for the calculation in the block by setting the corresponding I/O.

- Regardless of whether or not `BypLix` is connected, the operator can exclude the bypass signal:
  `BypInx` = 1 (by the operator via faceplate)
- `BypLix.ST` <> 16#FF (input is connected) and a change to the `BypLix` value is detected:
  `BypLix` = 1 (of I/O on CFC)

The I/O is shown in the faceplate by the following symbol:



Special case: If all input parameters are excluded, the output value is defined using the `DefaultOut` parameter.

Depending on the configuration of the Feature bit "Separate evaluation of excluded and simulated interlock signals" (Feature.Bit 2), the "Bypass" output is formed and the hidden bypass signal is set at the Out output.

Feature.Bit =0:

If one of the interlock inputs is disabled (BypIn01...BypIn16 or BypLi01...BypLi16) the BypAct output is set. The hidden bypass signal at the Out output is reset.

- BypAct.Value =
  BypIn01 OR BypLi01 (when BypLi01 is connected and the value has changed)
  OR BypInx OR BypLix (when BypLix is connected and the value has changed)
- Out.Bit1 = 0

A disabled interlock input that is switch-relevant after the bypass sets the status of Out to Simulation.

Feature.Bit =1:

If one of the interlock values is disabled (BypIn01...BypIn0x or BypLi01...BypLix) or a hidden bypass signal is set at an interlock input (In01...Inx), the BypAct output is set. The bypass signal at the In01...Inx input can be read from the upstream block at the BypAct output. The hidden bypass signal at the Out output is set to the value of the BypAct.

- BypAct.Value =
  BypIn01 OR BypLi01 (when BypLi01 is connected and the value has changed)
  OR BypInx OR BypLix (when BypLix is connected and the value has changed)
  OR In01.Bit1 OR Inx.Bit1 (hidden lines)
- Out.Bit1 = BypAct.Value

A disabled interlock input has no influence on the status of Out.

Note

Do **not** connect the BypLix input to the BypAct output of an interlock block, because BypAct is calculated from hidden bits.

Temporary bypass by operator

The operator can activate an exclusion for a specified time. This time is set in the CFC at the input parameter BypTime and applies to all signals. The input parameter BypPreWarnTime determines how long prior to the expiration of BypTime the BypPreWarnAct output should be set, for example, to generate a message for the operator.

A temporary bypass is shown in the faceplate by the following symbol:



Temporary and permanent bypass cannot be activated at the same time because a permanent bypass overwrites a temporary bypass.

Restart the temporary bypass time

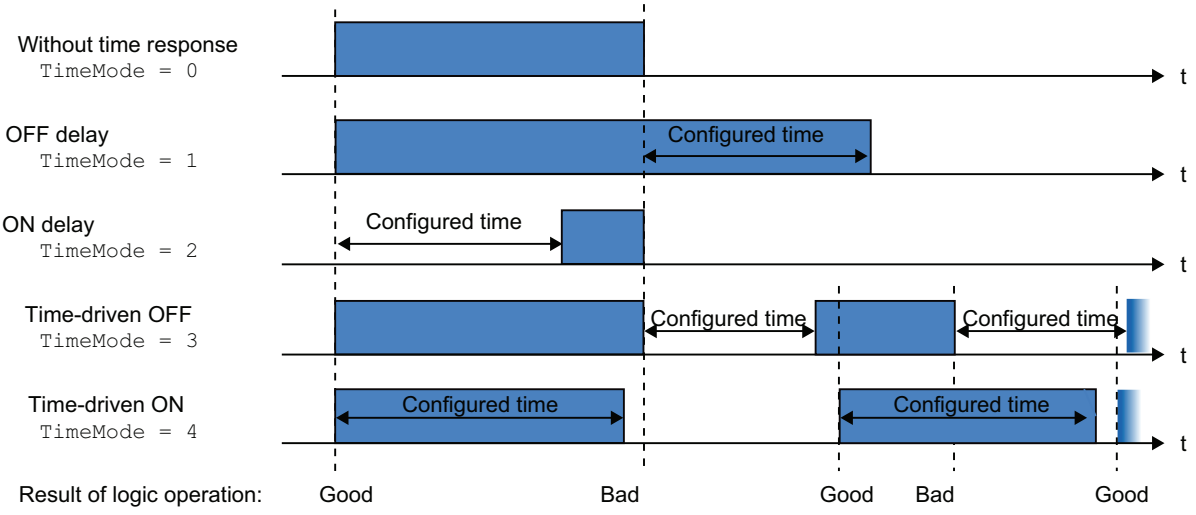
You can use the "Allow restart of the temporary bypass time" function (`Feature`.Bit 30) to enable an additional button in runtime with which the operator can restart the time for the temporary bypass.

Time behavior

A time response for the output value can be configured for the block. The `TimeMode` I/O can be used to configure the following time responses:

- `TimeMode` =0: Without time response (default)
- `TimeMode` =1: OFF delay (changing the result of logic operation from good to bad is delayed by `ILckTime`)
- `TimeMode` =2: ON delay (changing the result of logic operation from bad to good is delayed by `ILckTime`)
- `TimeMode` =3: Time-driven OFF (the result of logic operation is bad for the time `ILckTime`)
- `TimeMode` =4: Time-driven ON (the result of logic operation is good for the time `ILckTime`)

The time for the response is specified at the `ILckTime` I/O.



WARNING

Changes to the `TimeMode` connection

If the time response at the `TimeMode` connection is changed during runtime, this may have an effect on the signal state of the block output. The time function has an effect on the output once after the change regardless of the input signals.

Handling non-connected inputs

Inputs which are not interconnected are not evaluated. They are not displayed in the faceplate either.

Special case: If no inputs are connected, the output value is defined using the `DefaultOut` parameter. The signal status is set to 16#FF.

Opening additional faceplates

This block provides the standard APL function calling additional faceplates. Only one additional faceplate can be called up using `SelFp1` and `SelFp2`.

Select unit of measure

The block provides the standard APL function for selecting a unit of measure.

First-in detection for interlock blocks

This block provides the standard APL function for first-in signal detection. Note that a signal status change only has an effect on the first-in signal detection when the `Feature`.Bit 23 evaluation signal status and the `FirstInEn` input is set.

The response to deactivation via the `FirstInEn` input can be influenced by `Feature`.Bit 21 first-in signal detection response to deactivation.

Note

This function can only be executed in the faceplate with "process control" operating permission.

Forming the signal status for blocks

This block provides the standard APL function for forming and outputting signal status for interlock blocks.

The worst signal status `ST_Worst` for the block is formed from the following parameter:

- `OUT.ST`

Access control

This block provides the standard APL function for operator permissions.

The block has the following permissions for the `OS_Perm` parameter:

Bit	Function
0	1 = Operator can exclude all values
1	1 = Operator can reset the exclusion for all values
2	1 = Operator can reset the first-in signal detection
3	1 = Operator can set or reset temporary exclusion for all values
4-8	Not used
9	1 = Operator can reset the output signal
10	1 = Operator can start the time for temporary exclusion.
11	1 = Operator can change the time for temporary exclusion.

Bit	Function
12	1 = Operator can change the prewarning time for temporary exclusion (BypPreWarnTime)
13	1 = Operator can change the time for the time response (`ILckTime`)
14	Not used
15	1 = Operator can set or reset the exclusion of output value Out
16	1 = Operator can set or reset the exclusion of input value In01
17	1 = Operator can set or reset the exclusion of input value In02
18	1 = Operator can set or reset the exclusion of input value In03
19	1 = Operator can set or reset the exclusion of input value In04
20	1 = Operator can set or reset the exclusion of input value In05
21	1 = Operator can set or reset the exclusion of input value In06
22	1 = Operator can set or reset the exclusion of input value In07
23	1 = Operator can set or reset the exclusion of input value In08
24 - 31	Not used

Note

If you interconnect a parameter that is also listed in `OS_Perm` as a parameter, you have to reset the corresponding `OS_Perm` bit.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
2	See APL function: "Separate evaluation of excluded and simulated interlock signals"
5	Activate OS_Perm bits 0 = OS_Perm bits 16...23 inactive (evaluation only in faceplate) 1 = OS_Perm bits 16...23 active (evaluation only in faceplate)
6	Activate OS_Perm bits 1 = OS_Perm Bit 15 active
15	See APL function: "First-in detection response to deactivation"
21	See APL function: "Update acknowledgment and error status of the message call"
23	Evaluation of signal status
24	See APL function: "Enable local operating permission"
30	1 = Allow restart of the temporary bypass
31	See APL function: "Enable first-in detection"

19.1.4 ILck4x3L error handling

ILck4x3L error handling

For information about troubleshooting for all blocks, see 'Troubleshooting ' in the Basics section of the APL manual.

The following errors can be displayed for this block:

- Error numbers

Overview of error numbers

The `ErrorNum` I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
13	The parameter • `Logic3` is <0 or >2. • `LogicX_X` is <0 or >3.

19.1.5 ILck4x3L messaging

Messages

This block has no message behavior.

19.1.6 ILck4x3L I/Os

ILck4x3L I/Os

Input parameters

Parameter	Description	Type	Initial value
`AV01 .. AV08`	Analog value of In01 .. In08	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
`AV01_Unit .. AV08_Unit`	Unit of measure for AV01 .. AV08	INT	0
`BypIn01* .. ByIn08*`	1 = Input In01 .. In08 is not used	BOOL	0

Parameter	Description	Type	Initial value
BypLi01...BypLi08	Bypass In01...In08 (via interconnection)	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
DefaultOut	Output value for cases where all inputs are excluded or not interconnected. For more on this, see section ILck4x3L functions (Page 997) of the block.	BOOL	1
Feature	I/O for additional functions (Page 997)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 0 • .. • 0
FirstInEn	First-in detection	STRUCT • Value: BOOL • ST: BYTE	- • 1 • 16#80
In01 .. In08	Input In01 .. In08	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
InvIn01 .. InvIn08	Input value In01 .. Invert In08	BOOL	0
Logic1_1 ..Logic3	Logical linking of the respective level: 0 = Logical OR 1 = Logical AND 2 = Logical XOR 3 = Not used	INT	0
NotUsed	1 = Block is not used (only for display in the faceplate). You can find additional information on this in the section Standard view of the interlock blocks / color of the field.	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
OS_Perm	I/O for operator permissions (Page 997)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 1 • .. • 1
Out_Byp	Enable bypass (operator)	BOOL	0
BypOp*	Temporary bypass by operator	STRUCT	0
BypPreWarnTime*	Prewarning time for temporary bypass [s]	REAL	10.0
BypRST*	Time for temporary bypass and restart	BOOL	0
BypTime*	Time for temporary bypass [s]	REAL	30.0
RstBypOp*	Disable all bypasses (operator)	BOOL	0
RstFirstOp*	First-in via operator reset	BOOL	0
RstLi	First-in via interconnection reset	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
RstOp*	Reset by the operator	BOOL	0

Parameter	Description	Type	Initial value
TimeMode	Dynamic response: 0: No time response 1: On delay 2: Off delay 3: Time-controlled OFF 4: Time-driven ON	INT	0
ILckTime	Time of ON/OFF delay or evaluation	REAL	0.0

* Values can be written back to these inputs during execution of the block by the block algorithm.

Output parameters

Parameter	Description	Type	Initial value
ErrorNum	Output of pending error number. For the error numbers that can be output by this block, see ILck4x3L error handling (Page 1004)	INT	-1
FirstIn	Bit-coded number of the first signal that has resulted in a change at the output	DWORD	16#00000000
OS_PermOut	Display of OS_Perm	DWORD	16#FFFFFFFF
OS_PermLog	Display of OS_Prem with settings changed by the block algorithm	DWORD	16#FFFFFFFF
Out	Output	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Out1_1 ...Out2_2	Output: Result of the respective group on the respective level	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypTmpAct	Temporary bypass active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Out_Byp_Out	Bypass via evaluation active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypPreWarnAct	1 = Prewarning time of the temporary bypass has expired.	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypTime_Out	Remaining time for temporary bypass	REAL	0.0
ILckTime_Out	Remaining time for time response	REAL	0.0
ST_Worst	Worst signal status	BYTE	16#80
Status1	Status word 1 (Page 991)	DWORD	16#00000000
Status2	Status word 2 (Page 991)	DWORD	16#00000000
Status3	Status word 3 (Page 991)	DWORD	16#00000000
Status4	Status word 4 (Page 991)	DWORD	16#00000000

Parameter	Description	Type	Initial value
Status5	Status word 5 (Page 991)	DWORD	16#00000000
Status6	Status word 6 (Page 991)	DWORD	16#00000000
Status7	Status word 7 (Page 991)	DWORD	16#00000000
Status8	Status word 8 (Page 991)	DWORD	16#00000000
Status9	Status word 9 (Page 991)	DWORD	16#00000000
Status10	Status word 10 (Page 991)	DWORD	16#00000000
Status13	Status word 13 (Page 991)	DWORD	16#00000000

19.1.7 ILck4x3L block diagram

ILck4x3L block diagram

A block diagram is not provided for this block.

19.1.8 Operator control and monitoring

19.1.8.1 ILck4x3L views

ILck8x4L block views

The block provides the following views:

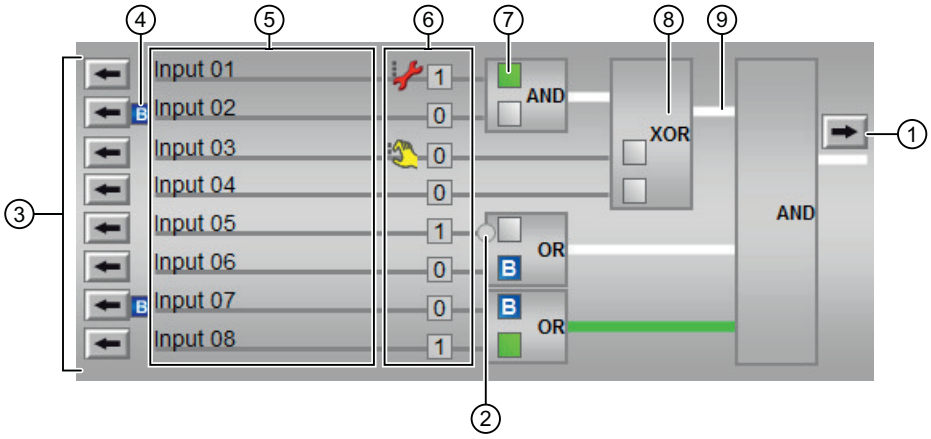
- ILck4x3L standard view (Page 1008)
- ILck4x3L extended standard view (Page 1010)
- ILck4x3L parameter view (Page 1015)
- ILck4x3L preview (Page 1016)
- Memo view
- ILck4x3L block icon (Page 1017)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

19.1.8.2 ILck4x3L standard view

ILck4x3L standard view

The standard view provides an overview of the current state of ILck4x3L. More detailed information is given to the operator in the extended standard view.



(1) Open faceplate of the output value

If you press the button ( ), you can open the faceplate for the output value. The function of this button depends on the configuration in the engineering system (ES). Refer the Opening additional faceplates section of the APL documentation.

(2) Inversion of an input signal

If an inversion is configured for an I/O, this is displayed as a circle at the input signal.

(3) Call the faceplate of the input value

When you press the button ( ), you can open the faceplate associated with each input value. The function of this button depends on the configuration in the engineering system (ES). Refer the Opening additional faceplates section of the APL documentation.

(4) Bypass information of the input value

The symbol indicates whether the input value has been externally bypassed via an interconnection.

(5) Label of the input signals

Changing the display

You can change the displays in the CFC in the object properties of the Interlock block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the input parameter (`INxx`) to what you want to see during runtime later.
- In the "Identifier" column, change the default setting for the input parameter (`AVxx`) to what you want to see during runtime later.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding `AVxxx` input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (`INxx`), this text is displayed even then the "Identifier" text field of the input parameter (`AVxx`) is not empty. If the "OS additional text" field of the (`INxx`) input parameter is empty, the "Identifier" text of the (`AVxx`) input parameter is displayed.

(6) Display of input values (BOOL) with signal status (in front of the field)

These fields show the interlock information associated with the value (13) with a signal status:

- 1 = "Good" state
- 0 = "Locked"

(7) Display the status for further processing

The symbol shows the status for further processing of the input values:

Icon	Further processing
	The input value is processed further with value 1.
	The input value is further processed with the value 0.
	The input value is excluded from further processing.

(8) Logic of the interlock

The area (6) shows you the function of the logical operator.

- AND
- OR
- XOR

You can configure the logic in the engineering system (ES).

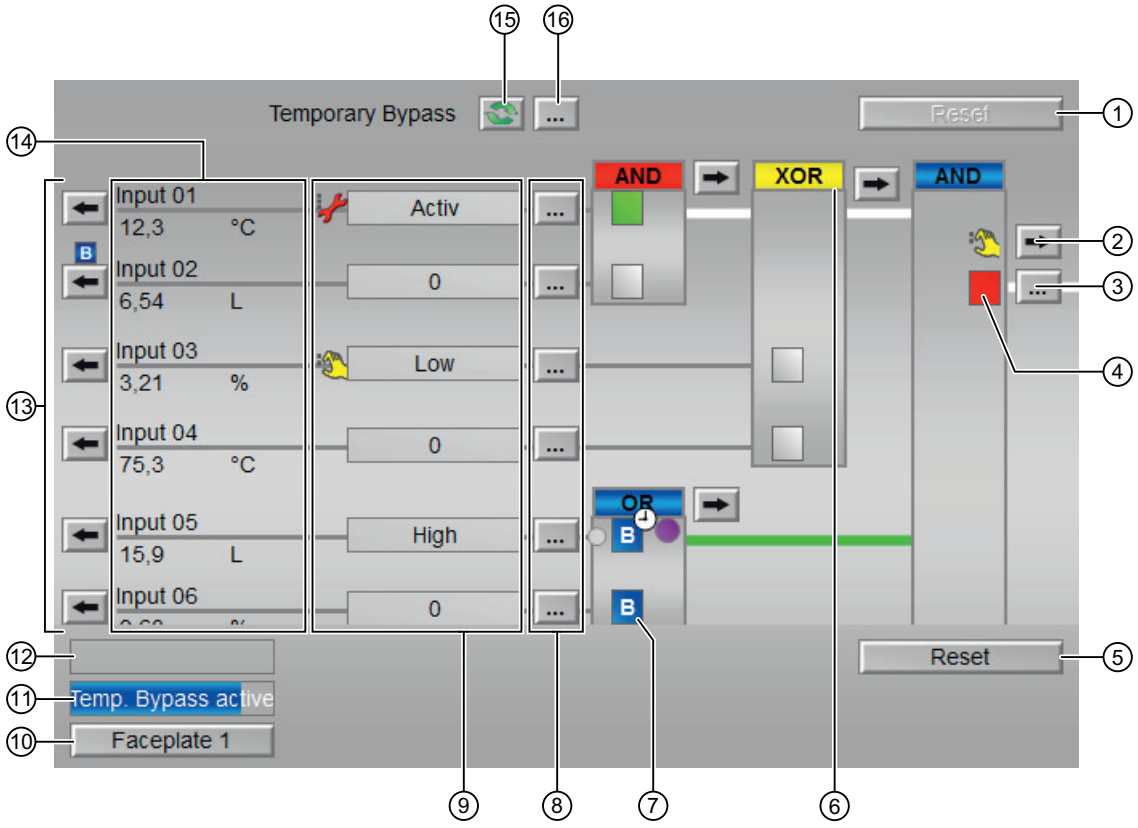
(9) Result of logic operation

The line indicates the status of the result of operation with its color:

Color of the line	Result of logic operation
Green	Output is enabled
White	Output is disabled

19.1.8.3 ILck4x3L extended standard view

ILck4x3L extended standard view



(1) Reset the output value

You can reset the output value with the button. The button is only operable if a reset is required.

The display of this button depends on the configuration in the engineering system (ES).

(2) Open the faceplate of the output value

If you press the button (➡), you can open the faceplate for the output value. The function of this button depends on the configuration in the engineering system (ES). Refer the Opening additional faceplates section of the APL documentation.

(3 / 8) Exclude input values / output value

You can use the button (3) to exclude input or output values from processing. Depending on the previous settings, you can "Set" or "Reset" this property.

If the input or output value has been excluded, the following symbol appears in the field (4 / 7):



You can find additional information on operation in the section "Switching operating states and operating modes" of the APL documentation.

Note

This function can only be executed in the faceplate with "high-level operating permission".

Note

The operator permissions via OS_Perm do not depend on the setting of bit 5. You can find additional information on this in the section "Enabling OS_Perm bits" of the APL documentation.

(4) Status of the block output

The field shows the status of the block output with the display:

Color of the line	Output status
	Output is not locked
	Output is interlocked
	Output bypassed
	Output is temporarily bridged

The line color indicates the status of the block output:

Color of the line	Output status
Green	Output is enabled
White	Output is disabled

(5) "Reset" the settings for further processing

When you press the button (10), you can "Reset" all input values:

- "Reset exclusions": the exclusions of the input values are reset.
- "Reset FirstIn": First-in detection / status display (4) is reset.

You can find additional information on this in the section Recording the first signal for interlock blocks of the APL documentation.

(6) Logic and status of the interlock

The area (6) shows you the function and status of the logical operator (priority from high to low). You can configure the logic in the engineering system (ES).

	Logic		
Color of the field	**AND**	**OR**	**XOR**
Blue	Excluded (bypass)		
Yellow	Simulated		
Red	Interlocked		
Green	Not interlocked		

(7) Display the status for further processing

The symbol shows the status for further processing of the input values:

Icon	Further processing
	The input value is processed further with value 1.
	The input value is further processed with the value 0.
	The input value is excluded from further processing.

(9) Display of input values (BOOL) with signal status (in front of the field)

These fields show the interlock information associated with the value (13) with a signal status:

- 1 = "Good" state
- 0 = "Locked"

Changing the display

You can change the displays for 0 and 1 in the CFC in the object properties of the Interlock block:

- Navigate to the I/Os (object properties).
- In the Text 0 and Text 1 columns, change the default for the input parameter (`Inxx`) to what you later want to see in runtime.

(10) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" of the APL documentation.

(11) Display area for block states

This area provides additional information on the operating state of the block:

- "Temporary bypass active"
  The color bar of the display area changes depending on the state. It shows the currently elapsed time of the set bypass time in bar graph and is initially blue.
- "Prewarning time active"
  If the temporary prewarning time for the temporary bypass is reached, the text and the display changes from blue to yellow.

(12) Display area for block states

This area provides additional information on the operating state of the block:

- "Delay active"
  The color bar of the display area shows the elapsed time of the configured time response in bar graph form.

(13) Open faceplate of the input value

When you press the button ( ), you can open the faceplate associated with each input value. The function of this button depends on the configuration in the engineering system (ES). Refer the Opening additional faceplates section of the APL documentation.

(14) Display of analog input values

The interconnected analog input values (`AVxx`) are displayed in this area. You set a unit of measure (`AVxx_Unit`) for each input in the engineering system (ES).

Changing the display

You can change the displays in the CFC in the object properties of the Interlock block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the input parameter (`INxx`) to what you want to see during runtime later.
- In the "Identifier" column, change the default setting for the input parameter (`AVxx`) to what you want to see during runtime later.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding`AVxxx` input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (`INxx`), this text is displayed even then the "Identifier" text field of the input parameter (`AVxx`) is not empty. If the "OS additional text" field of the (`INxx`) input parameter is empty, the "Identifier" text of the (`AVxx`) input parameter is displayed.

(15) Reset temporary bypass bridging time

With this button, you can reset the running bypass time to the start value if you have the appropriate operator permission. The display of this button depends on the configuration in the engineering system (ES).

(16) Temporary bypass

Use this button, you can open the operating window for activating the temporary bypass if you have the appropriate operator permission.

Within the user interface, you can select the inputs and the output for which the temporary bypass is to be activated. While the operating window is open, the names of the signals are displayed in the extended standard view.

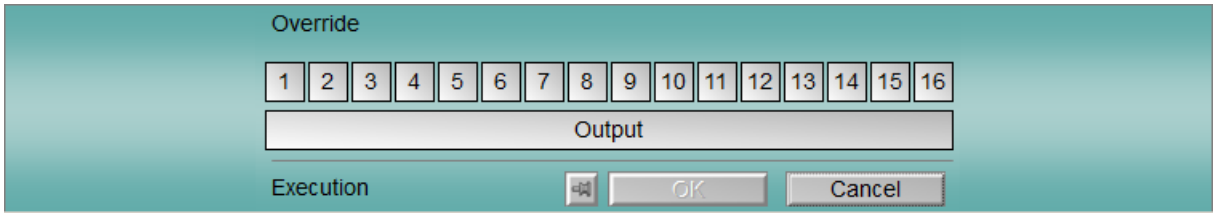
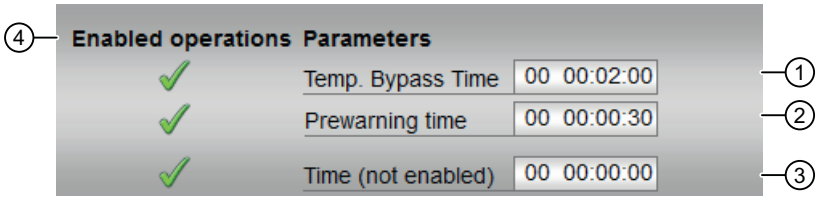


Figure 19-4 Operating window for setting a temporary bypass

19.1.8.4 ILck4x3L parameter view

ILck4x3L parameter view

In this area, you change the times and thus influence the behavior of the block.



(1) Time for temporary bypass

This time indicates how long a signal is excluded when a temporary bypass is activated.

(2) Prewarning time:

The prewarning time defines the time before the end of the temporary bypass at which a corresponding message is to be generated.

(3) Time response

The time for the time response configured at the block can be set here.

The display depends on the configuration of the time response.

You can influence the following parameters:

- Time (not enabled): A delay is not configured for the output; parameter cannot be manipulated.
- On-delay: On-delay is configured for the output. This can be controlled with corresponding operator authorization.
- Off-delay: Off-delay is configured for the output. This can be controlled with corresponding operator authorization.
- Duration (On/Off): The output is enabled/disabled time-controlled. The minimum time for which the output is active/inactive can be controlled with the appropriate operator permission.

(4) Enabled operations

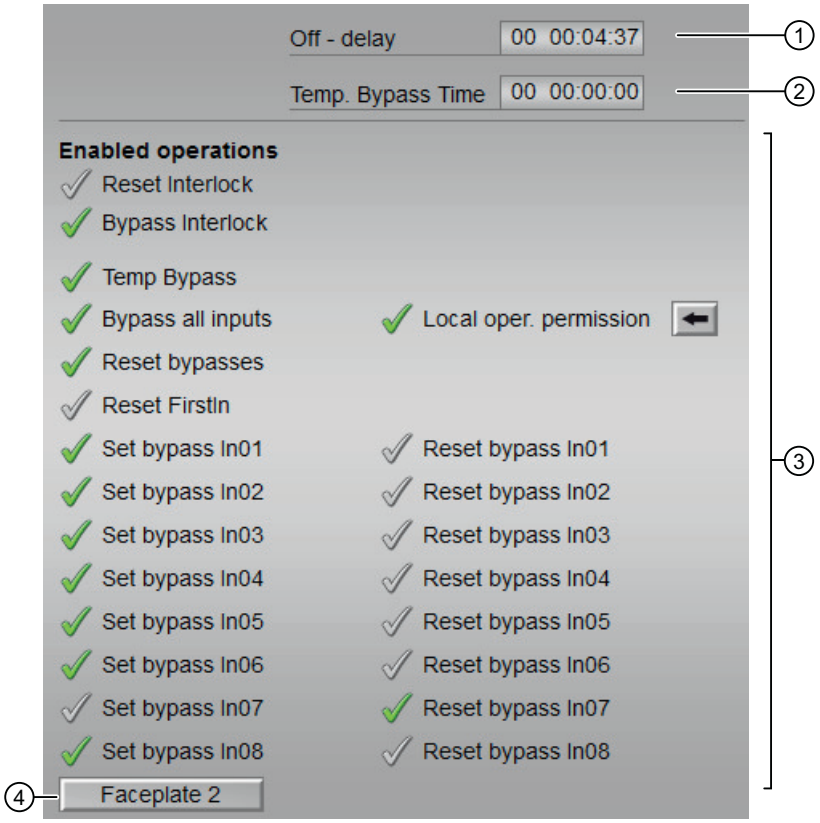
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured operator authorizations.

19.1.8.5 ILck4x3L preview

ILck4x3L preview



(1) Time response

The current time for the time response configured for the block is displayed here.

The display depends on the configuration.

You can display the following times here depending on the configuration:

- Time (not enabled): No delay is configured for the output.
- On-delay: On-delay is configured for the output.

- Off-delay: Off-delay is configured for the output.
- Duration: The output is active time controlled.

(2) Time for temporary bypass

The current time of the temporary bypass is displayed here.

The display depends on the configuration of the block.

(3) Operator control enable

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: the OS operator cannot control this parameter due to the configured AS operator permissions (OS_Perm or OS1Perm).

The following enabled operations are shown here:

- "Local operating permission": Use the <-- button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

19.1.8.6 ILck4x3L block icon

Properties of the ILck4x3L block icon

A variety of block icons are available with the following functions:

- Signal status
- Memo display
- Output signal

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Block icon for
	1	ILck4x3L
	2	ILck4x3L

Block icons (mini) without display of instance-specific names

These icons only show the output signal. They take the form of a small rectangle.

Display of the output signal

The display can show the following states for the output signal (priority from high to low):

- Gray: No inputs interconnected at the interlock block, the block is not used
- Blue: The output signal is 1, at least one input signal is bypassed. There is no gray state.
- Yellow: The signal status is 16#60; the output signal is simulated. There are no gray and blue states.
- Red: The output signal is 0; there is an interlock. There are no gray, blue, and yellow states.
- Green: The output signal is 1; the block is in the good state. There are no gray, blue, yellow, and red states.

You can find additional information on the block symbol in the following sections of the APL manual:

- Configuring the block icons
- Block icon structure
- Operation via the block icon

19.2 ILck8x4L - Interlock with 16 input signals and 8x4 logic

19.2.1 ILck8x4L description

Object name (type + number) and family

Type + number: FB 1134

Family: Interlck

Number of process objects: 1 PO

Application area for ILck8x4L

The block is used for the following applications:

- Standardized interlock with display

How it works

The block is used to calculate a standardized interlock that can be displayed on the OS. Maximum 16 input signals can be supplied to the block. These input signals are linked in groups with up to 4 levels with a selectable binary logic. Each group within the logic can be linked to AND, OR or XOR. The groups of levels 1 to 3 can be switched on or off. If incorrect logic is entered, corresponding group will be switched off. If incorrect logic is entered in level 4, Logic XOR will be set. The existing signals are then directly attached to the next group. The signal status of the output signal is also determined. You can assign an analog value with the signal status and unit to each input value for display in the faceplate.

The current state is shown at the `Out` output parameter:

- `Out` = 0: interlock
- `Out` = 1: Good state

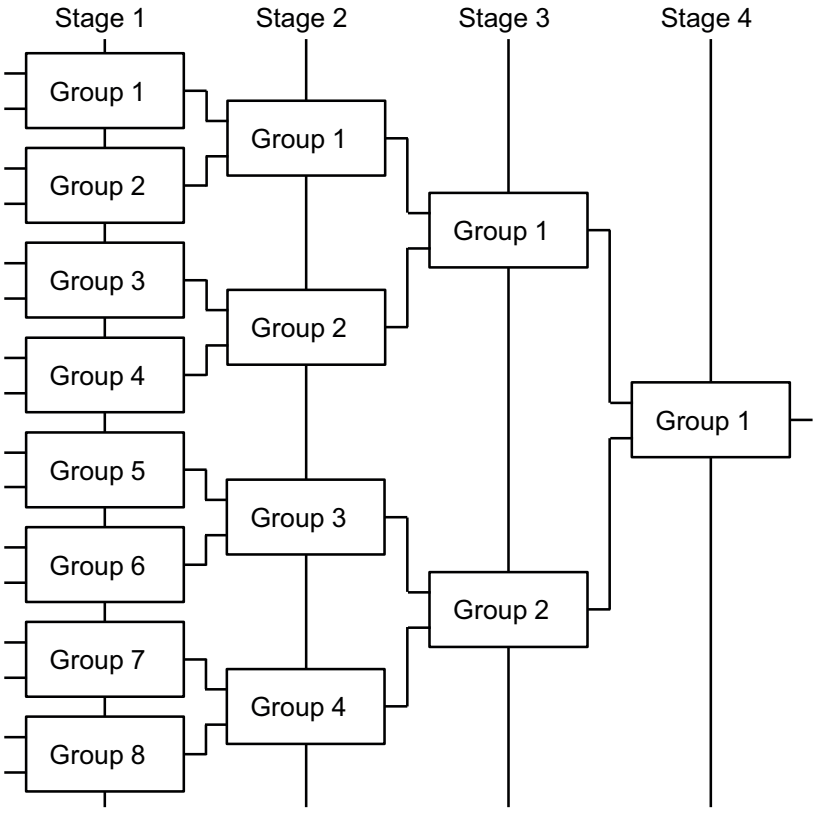


Figure 19-5 How it works

Configuration

Install the block in a cyclic interrupt OB (OB30 to OB38) in the CFC Editor.

Startup characteristics

Use the startup characteristics Feature bit 0 "Set startup characteristics" to define the startup characteristics for this block.

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section ILck8x4L functions (Page 1029).

Status bit	Parameter
0	1 = `Logic4` = OR
1	1 = `Logic4` = AND
2	1 = `Logic4` = XOR
3	Result of logic operation `Out.Value`
4	1 = At least one input value is excluded. For more on this, see section Exclusion of input signals in Functions of ILck8x4L (Page 1029)
5	1 = All input values are excluded

Status bit	Parameter
6	1 = No input value is interconnected or `NotUsed.Value`
7	Status display for simulation
8	Status display for not interlocked
9	Status display for interlocked
10	`Feature.Bit5`
11	`Feature.Bit2`
12	`Feature.Bit6`
13	Not used
14	`Feature.Bit15`
15 - 16	Not used
17	Delay time / Temp. Override active
18	Delay time `Out.Value`
19	Temp. override active
20	`BypPreWarnAct.Value`
21	`OS_PermLog.Bit10`
22 - 24	Not used
25	`Out_Byp_Out.Value`
26	1 = Output value is excluded
27 - 31	Not used

Status word assignment for the parameter `Status2`

Status bit	Parameter
0	`In01.Value`
1	`In02.Value`
2	`In03.Value`
3	`In04.Value`
4	`In05.Value`
5	`In06.Value`
6	`In07.Value`
7	`In08.Value`
8	`In09.Value`
9	`In10.Value`
10	`In11.Value`
11	`In12.Value`
12	`In13.Value`
13	`In14.Value`
14	`In15.Value`
15	`In16.Value`
16 - 31	Not used

Status word assignment for the parameter `Status3`

Status bit	Parameter
0	`InvIn01.Value`
1	`InvIn02.Value`
2	`InvIn03.Value`
3	`InvIn04.Value`
4	`InvIn05.Value`
5	`InvIn06.Value`
6	`InvIn07.Value`
7	`InvIn08.Value`
8	`InvIn09.Value`
9	`InvIn10.Value`
10	`InvIn11.Value`
11	`InvIn12.Value`
12	`InvIn13.Value`
13	`InvIn14.Value`
14	`InvIn15.Value`
15	`InvIn16.Value`
16 - 31	Not used

Status word assignment for the parameter `Status4`

Status bit	Parameter
0	`In01.Value` with inversion
1	`In02.Value` with inversion
2	`In03.Value` with inversion
3	`In04.Value` with inversion
4	`In05.Value` with inversion
5	`In06.Value` with inversion
6	`In07.Value` with inversion
7	`In08.Value` with inversion
8	`In09.Value` with inversion
9	`In10.Value` with inversion
10	`In11.Value` with inversion
11	`In12.Value` with inversion
12	`In13.Value` with inversion
13	`In14.Value` with inversion
14	`In15.Value` with inversion
15	`In16.Value` with inversion
16	`BypLi01.Value`
17	`BypLi02.Value`
18	`BypLi03.Value`
19	`BypLi04.Value`

Status bit	Parameter
20	BypLi05.Value
21	BypLi06.Value
22	BypLi07.Value
23	BypLi08.Value
24	BypLi09.Value
25	BypLi10.Value
26	BypLi11.Value
27	BypLi12.Value
28	BypLi13.Value
29	BypLi14.Value
30	BypLi15.Value
31	BypLi16.Value

Status word assignment for the parameter Status5

Status bit	Parameter
0	BypIn01.Value or temporary bypass In01
1	BypIn02.Value or temporary bypass In02
2	BypIn03.Value or temporary bypass In03
3	BypIn04.Value or temporary bypass In04
4	BypIn05.Value or temporary bypass In05
5	BypIn06.Value or temporary bypass In06
6	BypIn07.Value or temporary bypass In07
7	BypIn08.Value or temporary bypass In08
8	BypIn09.Value or temporary bypass In09
9	BypIn10.Value or temporary bypass In10
10	BypIn11.Value or temporary bypass In11
11	BypIn12.Value or temporary bypass In12
12	BypIn13.Value or temporary bypass In13
13	BypIn14.Value or temporary bypass In14
14	BypIn15.Value or temporary bypass In15
15	BypIn16.Value or temporary bypass In16
16 - 31	In01..In16 hidden bypass information

Status word assignment for the parameter Status6

Status bit	Parameter
0	In01 not connected
1	In02 not connected
2	In03 not connected
3	In04 not connected
4	In05 not connected

Status bit	Parameter
5	`In06` not connected
6	`In07` not connected
7	`In08` not connected
8	`In09` not connected
9	`In10` not connected
10	`In11` not connected
11	`In12` not connected
12	`In13` not connected
13	`In14` not connected
14	`In15` not connected
15	`In16` not connected
16 - 31	Not used

Status word assignment for the parameter `Status7`

Status bit	Parameter
0	`AV01` not connected
1	`AV02` not connected
2	`AV03` not connected
3	`Av04` not connected
4	`AV05` not connected
5	`AV06` not connected
6	`AV07` not connected
7	`AV08` not connected
8	`AV09` not connected
9	`AV10` not connected
10	`AV11` not connected
11	`AV12` not connected
12	`AV13` not connected
13	`AV14` not connected
14	`AV15` not connected
15	`AV16` not connected
16	`BypIn01`
17	`BypIn02`
18	`BypIn03`
19	`BypIn04`
20	`BypIn05`
21	`BypIn06`
22	`BypIn07`
23	`BypIn08`
24	`BypIn09`
25	`BypIn10`

Status bit	Parameter
26	BypIn11
27	BypIn12
28	BypIn13
29	BypIn14
30	BypIn15
31	BypIn16

Status word assignment for the parameter Status8

Identical to FirstIn.

Status word assignment for the parameter Status9

Status bit	Parameter
0	1 = Logic1_1 = OR
1	1 = Logic1_1 = AND
2	1 = Logic1_1 = XOR
3	Result of logic operation (Level1/Group1)
4	1 = At least one input value is excluded. (Level1/Group1)
5	1 = All input values are excluded. (Level1/Group1)
6	1 = No input value is interconnected. (Level1/Group1)
7	Status display for simulation (Level1/Group1)
8	Status display for not interlocked (Level1/Group1)
9	Status display for interlocked (Level1/Group1)
10 - 15	Not used
16	1 = Logic1_2 = OR
17	1 = Logic1_2 = AND
18	1 = Logic1_2 = XOR
19	Result of logic operation (Level1/Group2)
20	1 = At least one input value is excluded. (Level1/Group2)
21	1 = All input values are excluded. (Level1/Group2)
22	1 = No input value is interconnected. (Level1/Group2)
23	Status display for simulation (Level1/Group2)
24	Status display for not interlocked (Level1/Group2)
25	Status display for interlocked (Level1/Group2)
25 - 31	Not used

Status word assignment for the parameter Status10

Status bit	Parameter
0	1 = Logic1_3 = OR
1	1 = Logic1_3 = AND

Status bit	Parameter
2	1 = `Logic1_3` = XOR
3	Result of logic operation (Level1/Group3)
4	1 = At least one input value is excluded. (Level1/Group3)
5	1 = All input values are excluded. (Level1/Group3)
6	1 = No input value is interconnected. (Level1/Group3)
7	Status display for simulation (Level1/Group3)
8	Status display for not interlocked (Level1/Group3)
9	Status display for interlocked (Level1/Group3)
10 - 15	Not used
16	1 = `Logic1_4` = OR
17	1 = `Logic1_4` = AND
18	1 = `Logic1_4` = XOR
19	Result of logic operation (Level1/Group4)
20	1 = At least one input value is excluded. (Level1/Group4)
21	1 = All input values are excluded. (Level1/Group4)
22	1 = No input value is interconnected. (Level1/Group4)
23	Status display for simulation (Level1/Group4)
24	Status display for not interlocked (Level1/Group4)
25	Status display for interlocked (Level1/Group4)
25 - 31	Not used

Status word assignment for the parameter `Status11`

Status bit	Parameter
0	1 = `Logic1_5` = OR
1	1 = `Logic1_5` = AND
2	1 = `Logic1_5` = XOR
3	Result of logic operation (Level1/Group5)
4	1 = At least one input value is excluded. (Level1/Group5)
5	1 = All input values are excluded. (Level1/Group5)
6	1 = No input value is interconnected. (Level1/Group5)
7	Status display for simulation (Level1/Group5)
8	Status display for not interlocked (Level1/Group5)
9	Status display for interlocked (Level1/Group5)
10 - 15	Not used
16	1 = `Logic1_6` = OR
17	1 = `Logic1_6` = AND
18	1 = `Logic1_6` = XOR
19	Result of logic operation (Level1/Group6)
20	1 = At least one input value is excluded. (Level1/Group6)
21	1 = All input values are excluded. (Level1/Group6)
22	1 = No input value is interconnected. (Level1/Group6)

Status bit	Parameter
23	Status display for simulation (Level1/Group6)
24	Status display for not interlocked (Level1/Group6)
25	Status display for interlocked (Level1/Group6)
25 - 31	Not used

Status word assignment for the parameter `Status12`

Status bit	Parameter
0	1 = `Logic1_7` = OR
1	1 = `Logic1_7` = AND
2	1 = `Logic1_7` = XOR
3	Result of logic operation (Level1/Group7)
4	1 = At least one input value is excluded. (Level1/Group7)
5	1 = All input values are excluded. (Level1/Group7)
6	1 = No input value is interconnected. (Level1/Group7)
7	Status display for simulation (Level1/Group7)
8	Status display for not interlocked (Level1/Group7)
9	Status display for interlocked (Level1/Group7)
10 - 15	Not used
16	1 = `Logic1_8` = OR
17	1 = `Logic1_8` = AND
18	1 = `Logic1_8` = XOR
19	Result of logic operation (Level1/Group8)
20	1 = At least one input value is excluded. (Level1/Group8)
21	1 = All input values are excluded. (Level1/Group8)
22	1 = No input value is interconnected. (Level1/Group8)
23	Status display for simulation (Level1/Group8)
24	Status display for not interlocked (Level1/Group8)
25	Status display for interlocked (Level1/Group8)
25 - 31	Not used

Status word assignment for the parameter `Status13`

Status bit	Parameter
0	1 = `Logic2_1` = OR
1	1 = `Logic2_1` = AND
2	1 = `Logic2_1` = XOR
3	Result of logic operation (Level2/Group1)
4	1 = At least one input value is excluded. (Level2/Group1)
5	1 = All input values are excluded. (Level2/Group1)
6	1 = No input value is interconnected. (Level2/Group1)
7	Status display for simulation (Level2/Group1)

Status bit	Parameter
8	Status display for not interlocked (Level2/Group1)
9	Status display for interlocked (Level2/Group1)
10 - 15	Not used
16	1 = `Logic2_2` = OR
17	1 = `Logic2_2` = AND
18	1 = `Logic2_2` = XOR
19	Result of logic operation (Level2/Group2)
20	1 = At least one input value is excluded. (Level2/Group2)
21	1 = All input values are excluded. (Level2/Group2)
22	1 = No input value is interconnected. (Level2/Group2)
23	Status display for simulation (Level2/Group2)
24	Status display for not interlocked (Level2/Group2)
25	Status display for interlocked (Level2/Group2)
25 - 31	Not used

Status word assignment for the parameter `Status14`

Status bit	Parameter
0	1 = `Logic2_3` = OR
1	1 = `Logic2_3` = AND
2	1 = `Logic2_3` = XOR
3	Result of logic operation (Level2/Group3)
4	1 = At least one input value is excluded. (Level2/Group3)
5	1 = All input values are excluded. (Level2/Group3)
6	1 = No input value is interconnected. (Level2/Group3)
7	Status display for simulation (Level2/Group3)
8	Status display for not interlocked (Level2/Group3)
9	Status display for interlocked (Level2/Group3)
10 - 15	Not used
16	1 = `Logic2_4` = OR
17	1 = `Logic2_4` = AND
18	1 = `Logic2_4` = XOR
19	Result of logic operation (Level2/Group4)
20	1 = At least one input value is excluded. (Level2/Group4)
21	1 = All input values are excluded. (Level2/Group4)
22	1 = No input value is interconnected. (Level2/Group4)
23	Status display for simulation (Level2/Group4)
24	Status display for not interlocked (Level2/Group4)
25	Status display for interlocked (Level2/Group4)
25 - 31	Not used

Status word assignment for the parameter Status15

Status bit	Parameter
0	1 = Logic3_1 = OR
1	1 = Logic3_1 = AND
2	1 = Logic3_1 = XOR
3	Result of logic operation (Level3/Group1)
4	1 = At least one input value is excluded. (Level3/Group1)
5	1 = All input values are excluded. (Level3/Group1)
6	1 = No input value is interconnected. (Level3/Group1)
7	Status display for simulation (Level3/Group1)
8	Status display for not interlocked (Level3/Group1)
9	Status display for interlocked (Level3/Group1)
10 - 15	Not used
16	1 = Logic3_2 = OR
17	1 = Logic3_2 = AND
18	1 = Logic3_2 = XOR
19	Result of logic operation (Level3/Group2)
20	1 = At least one input value is excluded. (Level3/Group2)
21	1 = All input values are excluded. (Level3/Group2)
22	1 = No input value is interconnected. (Level3/Group2)
23	Status display for simulation (Level3/Group2)
24	Status display for not interlocked (Level3/Group2)
25	Status display for interlocked (Level3/Group2)
25 - 31	Not used

See also

ILck8x4L I/Os (Page 1037)

19.2.2 ILck8x4L operating modes

ILck8x4L operating modes

This block does not have any operating modes.

19.2.3 ILck8x4L functions

ILck8x4L functions

The functions for this block are listed below.

Logic operators

Use the `LogicX_X` parameter to specify the logic operator that the block should employ when determining the interlock state.

The name of the `Logic` I/O is always formed so that the first number in the name of the parameter indicates the level and the second number the group for which you specify the logic with this I/O.

Set the following:

- `Logic` = 0: OR
- `Logic` = 1: AND
- `Logic` = 2: XOR
- `Logic` = 3: Do not use group (not for `Logic4`)

The result of the respective groups is displayed at the `OutX_X` output of the block. The name of the `OutX_X` I/O is formed similarly to the `Logic` I/Os: Here too, the first number in the name of the parameter is the level and the second number is the group. The final result of `Logic4` is sent to the `Out` output.

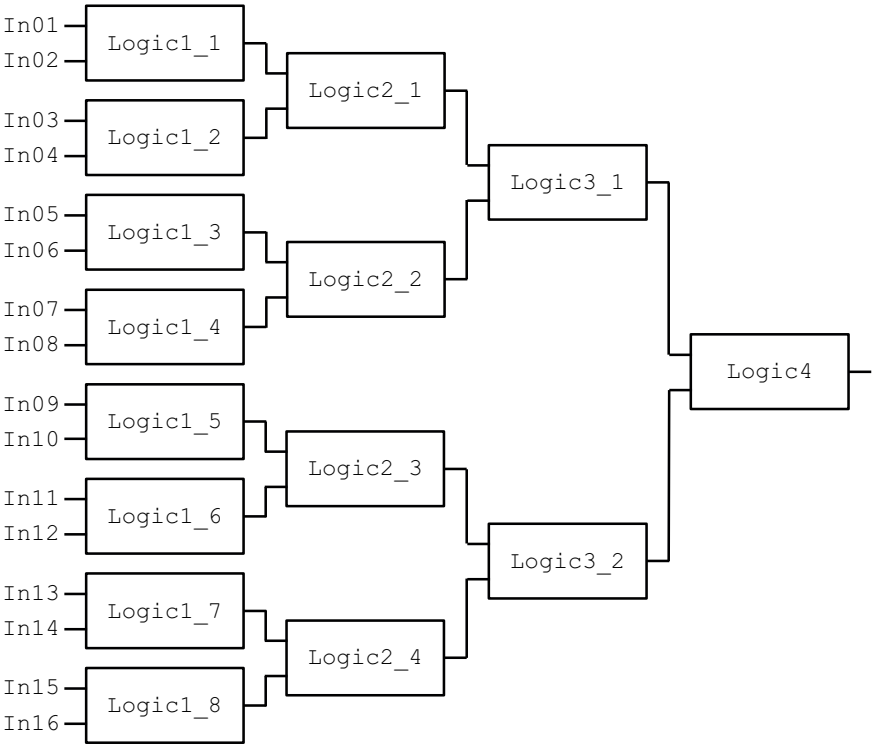


Figure 19-6 Function of the logic I/O of ILck8x4L

If a logic group is selected as logical operator = 3, all existing signals are linked to the logic group of the next level. The logic group in level 4 is always present.

Example:

Setting the input parameters `Logic1_2` = 3, `Logic1_6` = 3 and `Logic2_3` = 3 results in the logic shown below. All other groups are evaluated according to the configured logical operator.

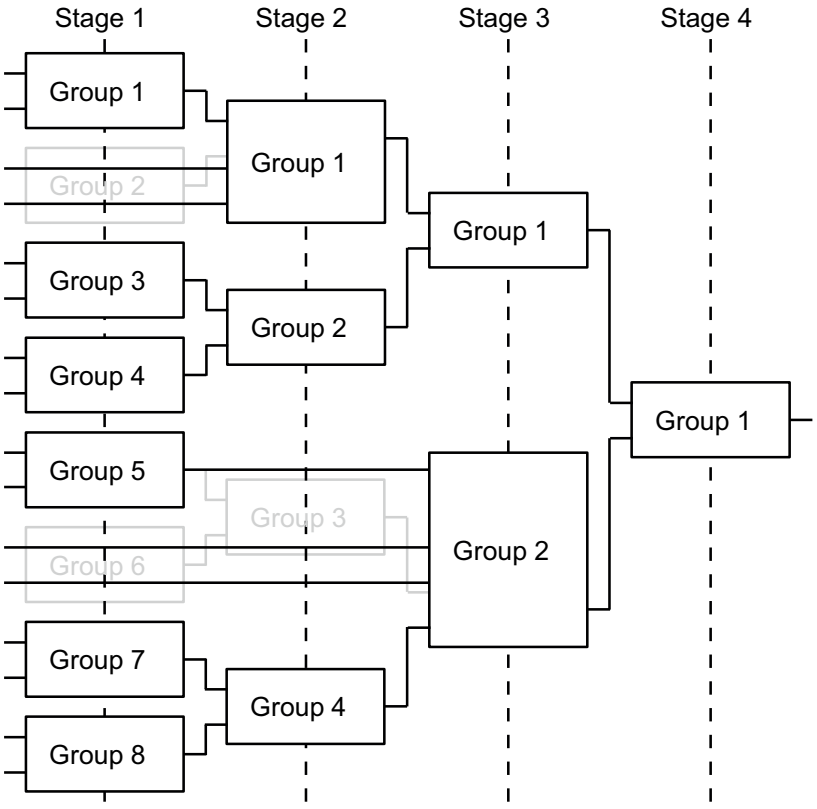


Figure 19-7 Example of a logical operator configured as a "non-existent group"

Automatic reset of the output value

You can use the "Automatic reset of the output value" function (`Feature`.Bit 15) to realize a store function for the output value.

`Feature`**.Bit =0**

The output value is not automatically reset. When the conditions for the interlock are met, the output value is set. The output value is first reset when the condition is no longer fulfilled and the operator has executed a reset.

`Feature`**.Bit =1**

The output value is automatically reset. When the conditions for the interlock are met, the output value is set. The output value is reset as soon as the condition is no longer fulfilled.

Inversion of logic signals

You can invert the input signals by setting the input parameter `InvInx` for the respective input to `Inx` = 1, e.g. for input `In01` set the I/O `InvIn01`.

The inversion is displayed in the faceplate. If you invert signals using any other method, this is not shown in the faceplate.

Bypass

Note

In the ILck8x4L block, exclusion "bypassing) the interlock means that the interlock signal (input signal) is excluded from the logic of the interlock block, in other words, this signal is ignored in the logic operations.

This function can only be executed in the faceplate with "high-level operating permission".

You can disable input signals that are temporarily not to be used for the calculation in the block by setting the corresponding I/O.

- Regardless of whether or not `BypLix` is connected, the operator can exclude the bypass signal:
  `BypInx` = 1 (by the operator via faceplate)
- `BypLix.ST` <> 16#FF (input is connected) and a change to the `BypLix` value is detected:
  `BypLix` = 1 (of I/O on CFC)

The I/O is shown in the faceplate by the following symbol:



Special case: If all input parameters are excluded, the output value is defined using the `DefaultOut` parameter.

Depending on the configuration of the Feature bit "Separate evaluation of excluded and simulated interlock signals" (`Feature`.Bit 2), the "Bypass" output is formed and the hidden bypass signal is set at the `Out` output.

`Feature`.Bit =0:

If one of the interlock inputs is disabled (`BypIn01`...`BypIn16` or `BypLi01`...`BypLi16`) the `BypAct` output is set. The hidden bypass signal at the `Out` output is reset.

- `BypAct.Value` =
  `BypIn01` OR `BypLi01` (when `BypLi01` is connected and the value has changed)
  OR `BypInx` OR `BypLix` (when `BypLix` is connected and the value has changed)
- `Out.Bit1` = 0

A disabled interlock input that is switch-relevant after the bypass sets the status of `Out` to Simulation.

`Feature`.Bit =1:

If one of the interlock values is disabled (`BypIn01`...`BypIn0x` or `BypLi01`...`BypLix`) or a hidden bypass signal is set at an interlock input (`In01`...`Inx`), the `BypAct` output is set. The bypass signal at the `In01`...`Inx` input can be read from the upstream block at the `BypAct` output. The hidden bypass signal at the `Out` output is set to the value of the `BypAct`.

- `BypAct.Value` =
  `BypIn01` OR `BypLi01` (when `BypLi01` is connected and the value has changed)
  OR `BypInx` OR `BypLix` (when `BypLix` is connected and the value has changed)
  OR `In01.Bit1` OR `Inx.Bit1` (hidden lines)
- `Out.Bit1` = `BypAct.Value`

A disabled interlock input has no influence on the status of `Out`.

Note

Do **not** connect the `BypLix` input to the `BypAct` output of an interlock block, because `BypAct` is calculated from hidden bits.

Temporary bypass by operator

The operator can activate an exclusion for a specified time. This time is set in the CFC at the input parameter `BypTime` and applies to all signals. The input parameter `BypPreWarnTime` determines how long prior to the expiration of `BypTime` the `BypPreWarnAct` output should be set, for example, to generate a message for the operator.

A temporary bypass is shown in the faceplate by the following symbol:



Temporary and permanent bypass cannot be activated at the same time because a permanent bypass overwrites a temporary bypass.

Restart the temporary bypass time

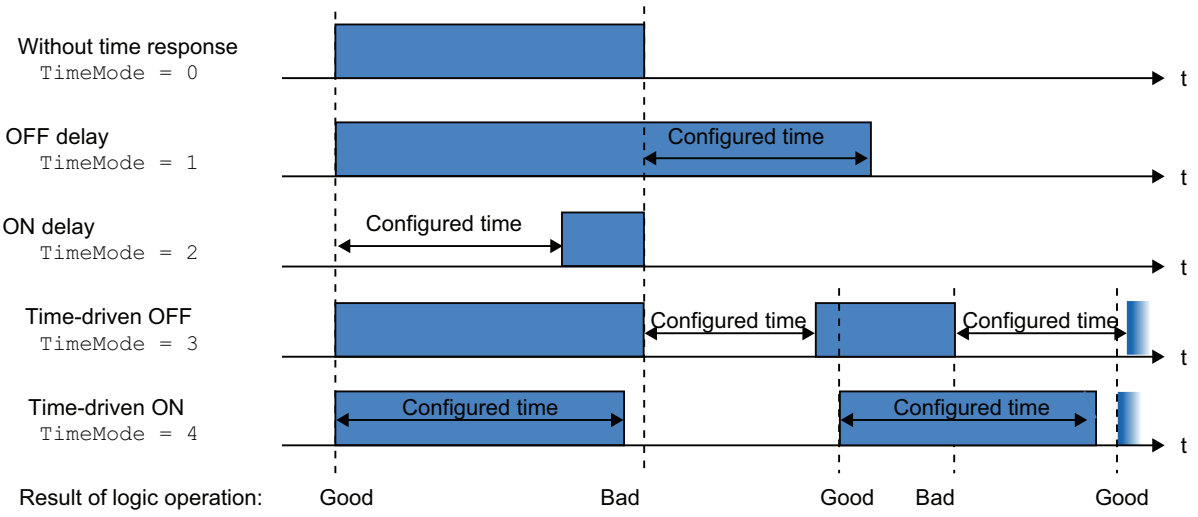
You can use the "Allow restart of the temporary bypass time" function (`Feature`.Bit 30) to enable an additional button in runtime with which the operator can restart the time for the temporary bypass.

Time behavior

A time response for the output value can be configured for the block. The `TimeMode` I/O can be used to configure the following time responses:

- `TimeMode` =0: Without time response (default)
- `TimeMode` =1: OFF delay (changing the result of logic operation from good to bad is delayed by `ILckTime`)
- `TimeMode` =2: ON delay (changing the result of logic operation from bad to good is delayed by `ILckTime`)
- `TimeMode` =3: Time-driven OFF (the result of logic operation is bad for the time `ILckTime`)
- `TimeMode` =4: Time-driven ON (the result of logic operation is good for the time `ILckTime`)

The time for the response is specified at the `ILckTime` I/O.



WARNING

Changes to the `TimeMode` connection

If the time response at the `TimeMode` connection is changed during runtime, this may have an effect on the signal state of the block output. The time function has an effect on the output once after the change regardless of the input signals.

Handling non-connected inputs

Inputs which are not interconnected are not evaluated. They are not displayed in the faceplate either.

Special case: If no inputs are connected, the output value is defined using the `DefaultOut` parameter. The signal status is set to 16#FF.

Opening additional faceplates

This block provides the standard APL function calling additional faceplates. Only one additional faceplate can be called up using `SelFp1` and `SelFp2`.

Select unit of measure

The block provides the standard APL function for selecting a unit of measure.

First-in detection for interlock blocks

This block provides the standard APL function for first-in signal detection. Note that a signal status change only has an effect on the first-in signal detection when the `Feature`.Bit 23 evaluation signal status and the `FirstInEn` input is set.

The response to deactivation via the FirstInEn input can be influenced by Feature.Bit 21 first-in signal detection response to deactivation.

Note

This function can only be executed in the faceplate with "process control" operating permission.

Forming the signal status for blocks

This block provides the standard APL function for forming and outputting signal status for interlock blocks.

The worst signal status ST_Worst for the block is formed from the following parameter:

- OUT.ST

Access control

This block provides the standard APL function for operator permissions.

The block has the following permissions for the OS_Perm parameter:

Bit	Function
0	1 = Operator can exclude all values
1	1 = Operator can reset the exclusion for all values
2	1 = Operator can reset the first-in signal detection
3	1 = Operator can set or reset temporary exclusion for all values
4-8	Not used
9	1 = Operator can reset the output signal
10	1 = Operator can start the time for temporary exclusion.
11	1 = Operator can change the time for temporary exclusion.
12	1 = Operator can change the prewarning time for temporary exclusion (BypPreWarnTime)
13	1 = Operator can change the time for the time response (ILckTime)
14	Not used
15	1 = Operator can set or reset the exclusion of output value Out
16	1 = Operator can set or reset the exclusion of input value In01
17	1 = Operator can set or reset the exclusion of input value In02
18	1 = Operator can set or reset the exclusion of input value In03
19	1 = Operator can set or reset the exclusion of input value In04
20	1 = Operator can set or reset the exclusion of input value In05
21	1 = Operator can set or reset the exclusion of input value In06
22	1 = Operator can set or reset the exclusion of input value In07
23	1 = Operator can set or reset the exclusion of input value In08
24	1 = Operator can set or reset the exclusion of input value In09
25	1 = Operator can set or reset the exclusion of input value In10
26	1 = Operator can set or reset the exclusion of input value In11
27	1 = Operator can set or reset the exclusion of input value In12

Bit	Function
28	1 = Operator can set or reset the exclusion of input value In13
29	1 = Operator can set or reset the exclusion of input value In14
30	1 = Operator can set or reset the exclusion of input value In15
31	1 = Operator can set or reset the exclusion of input value In16

Note

If you interconnect a parameter that is also listed in `OS_Perm` as a parameter, you have to reset the corresponding `OS_Perm` bit.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in section "Configurable functions using the Feature I/O" in the *Function Manual of the APL*. The following function is available for this block with the corresponding bits:

Bit	Function
0	See APL function: "Set startup characteristics"
2	See APL function: "Separate evaluation of excluded and simulated interlock signals"
5	Activate OS_Perm bits 0 = OS_Perm bits 16...31 inactive (evaluation only in faceplate) 1 = OS_Perm bits 16...31 active (evaluation only in faceplate)
6	Activate OS_Perm bits 1 = OS_Perm Bit 15 active
15	See APL function: "First-in detection response to deactivation"
21	See APL function: "Update acknowledgment and error status of the message call"
23	Evaluation of signal status
24	See APL function: "Enable local operating permission"
30	1 = Allow restart of the temporary bypass
31	See APL function: "Enable first-in detection"

19.2.4 ILck8x4L error handling

ILck8x4L error handling

For information about troubleshooting for all blocks, see 'Troubleshooting ' in the Basics section of the APL manual.

The following errors can be displayed for this block:

- Error numbers

Overview of error numbers

The `ErrorNum` I/O can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Predefined value when the block is inserted; the block is not executed.
0	There is no error.
13	The parameter • `Logic4` is <0 or >2. • `LogicX_X` is <0 or >3.

19.2.5 ILck8x4L messaging

Messages

This block has no message behavior.

19.2.6 ILck8x4L I/Os

ILck8x4L I/Os

Input parameters

Parameter	Description	Type	Initial value
`AV01 .. AV16`	Analog value of In01 .. In16	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#FF
`AV01_Unit .. AV16_Unit`	Unit of measure for AV01 .. AV16	INT	0
`BypIn01* .. ByIn16*`	1 = Input In01 .. In16 is not used	BOOL	0
`BypLi01...BypLi16`	Bypass In01...In16 (via interconnection)	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
`DefaultOut`	Output value for cases where all inputs are excluded or not interconnected. For more on this, see section ILck8x4L functions (Page 1029) of the block.	BOOL	1
`Feature`	I/O for additional functions (Page 1029)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 0 • .. • 0

Parameter	Description	Type	Initial value
FirstInEn	First-in detection	STRUCT • Value: BOOL • ST: BYTE	- • 1 • 16#80
In01 .. In16	Input In01 .. In16	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
InvIn01 .. InvIn16	Input value In01 .. Invert In16	BOOL	0
Logic1_1 ..Logic4	Logical linking of the respective level: 0 = Logical OR 1 = Logical AND 2 = Logical XOR 3 = Not used	INT	0
NotUsed	1 = Block is not used (only for display in the faceplate). You can find additional information on this in the section Standard view of the interlock blocks / color of the field.	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
OS_Perm	I/O for operator permissions (Page 1029)	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 1 • .. • 1
Out_Byp	Enable bypass (operator)	BOOL	0
BypOp*	Temporary bypass by operator	STRUCT	0
BypPreWarnTime*	Prewarning time for temporary bypass [s]	REAL	10.0
BypRST*	Time for temporary bypass and restart	BOOL	0
BypTime*	Time for temporary bypass [s]	REAL	30.0
RstBypOp*	Disable all bypasses (operator)	BOOL	0
RstFirstOp*	First-in via operator reset	BOOL	0
RstLi	First-in via interconnection reset	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#FF
RstOp*	Reset by the operator	BOOL	0
TimeMode	Dynamic response: 0: No time response 1: On delay 2: Off delay 3: Time-controlled OFF 4: Time-driven ON	INT	0
ILckTime	Time of ON/OFF delay or evaluation	REAL	0.0

* Values can be written back to these inputs during execution of the block by the block algorithm.

Output parameters

Parameter	Description	Type	Initial value
ErrorNum	Output of pending error number. For the error numbers that can be output by this block, see ILck8x4L error handling (Page 1036)	INT	-1
FirstIn	Bit-coded number of the first signal that has resulted in a change at the output	DWORD	16#00000000
OS_PermOut	Display of OS_Perm	DWORD	16#FFFFFFFF
OS_PermLog	Display of OS_Prem with settings changed by the block algorithm	DWORD	16#FFFFFFFF
Out	Output	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Out1_1 ...Out3_2	Output: Result of the respective group on the respective level	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypTmpAct	Temporary bypass active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
Out_Byp_Out	Bypass via evaluation active	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypPreWarnAct	1 = Prewarning time of the temporary bypass has expired.	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypTime_Out	Remaining time for temporary bypass	REAL	0.0
ILckTime_Out	Remaining time for time response	REAL	0.0
ST_Worst	Worst signal status	BYTE	16#80
Status1	Status word 1 (Page 1019)	DWORD	16#00000000
Status2	Status word 2 (Page 1019)	DWORD	16#00000000
Status3	Status word 3 (Page 1019)	DWORD	16#00000000
Status4	Status word 4 (Page 1019)	DWORD	16#00000000
Status5	Status word 5 (Page 1019)	DWORD	16#00000000
Status6	Status word 6 (Page 1019)	DWORD	16#00000000
Status7	Status word 7 (Page 1019)	DWORD	16#00000000
Status8	Status word 8 (Page 1019)	DWORD	16#00000000
Status9	Status word 9 (Page 1019)	DWORD	16#00000000
Status10	Status word 10 (Page 1019)	DWORD	16#00000000
Status11	Status word 11 (Page 1019)	DWORD	16#00000000
Status12	Status word 12 (Page 1019)	DWORD	16#00000000
Status13	Status word 13 (Page 1019)	DWORD	16#00000000

Parameter	Description	Type	Initial value
Status14	Status word 14 (Page 1019)	DWORD	16#00000000
Status15	Status word 15 (Page 1019)	DWORD	16#00000000

19.2.7 ILck8x4L block diagram

ILck8x4L block diagram

A block diagram is not provided for this block.

19.2.8 Operator control and monitoring

19.2.8.1 ILck8x4L views

ILck8x4L block views

The block provides the following views:

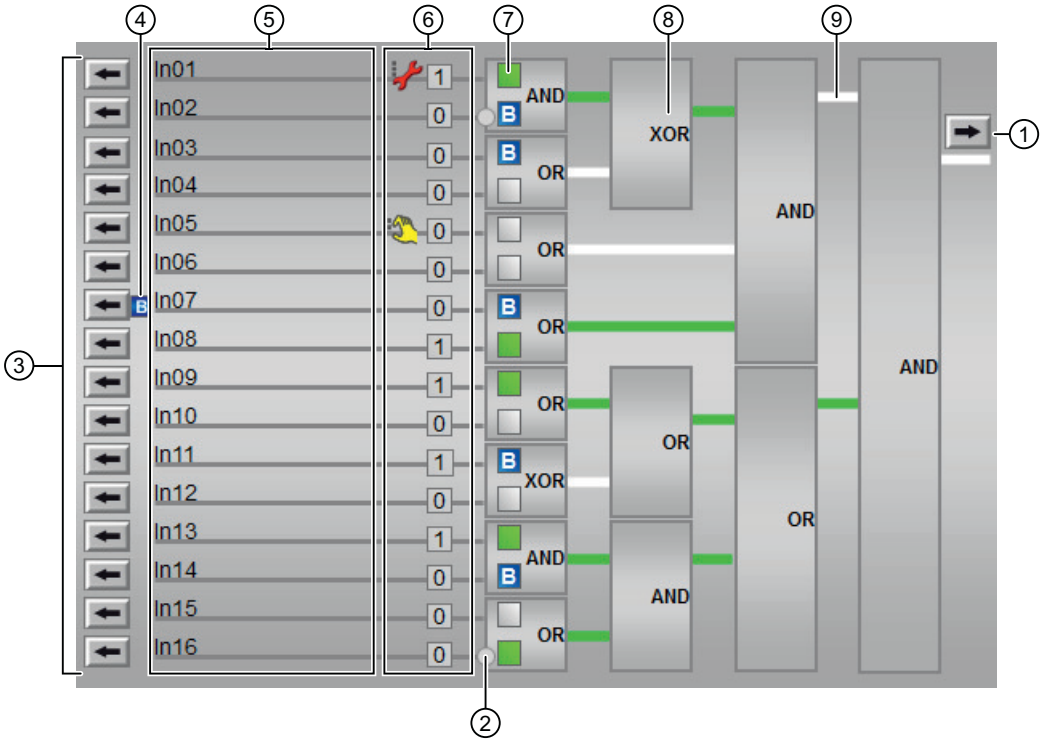
- Intlk8x4 standard view (Page 1041)
- ILck8x4L extended standard view (Page 1044)
- ILck8x4L parameter view (Page 1048)
- ILck8x4L preview (Page 1050)
- Memo view
- ILck8x4L block icon (Page 1051)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

19.2.8.2 Intlk8x4 standard view

ILck8x4L standard view

The standard view provides an overview of the current state of ILck8x4L. More detailed information is given to the operator in the extended standard view.



(1) Open faceplate of the output value

If you press the button ( ), you can open the faceplate for the output value. The function of this button depends on the configuration in the engineering system (ES). Refer the Opening additional faceplates section of the APL documentation.

(2) Inversion of an input signal

If an inversion is configured for an I/O, this is displayed as a circle at the input signal.

(3) Call the faceplate of the input value

When you press the button ( ), you can open the faceplate associated with each input value. The function of this button depends on the configuration in the engineering system (ES). Refer the Opening additional faceplates section of the APL documentation.

(4) Bypass information of the input value

The symbol indicates whether the input value has been externally bypassed via an interconnection.

(5) Label of the input signals

Changing the display

You can change the displays in the CFC in the object properties of the Interlock block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the input parameter (`INxx`) to what you want to see during runtime later.
- In the "Identifier" column, change the default setting for the input parameter (`AVxx`) to what you want to see during runtime later.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding `AVxxx` input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (`INxx`), this text is displayed even then the "Identifier" text field of the input parameter (`AVxx`) is not empty. If the "OS additional text" field of the (`INxx`) input parameter is empty, the "Identifier" text of the (`AVxx`) input parameter is displayed.

(6) Display of input values (BOOL) with signal status (in front of the field)

These fields show the interlock information associated with the value (13) with a signal status:

- 1 = "Good" state
- 0 = "Locked"

(7) Display the status for further processing

The symbol shows the status for further processing of the input values:

Icon	Further processing
	The input value is processed further with value 1.
	The input value is further processed with the value 0.
	The input value is excluded from further processing.

(8) Logic of the interlock

The area (6) shows you the function of the logical operator.

- AND
- OR
- XOR

You can configure the logic in the engineering system (ES).

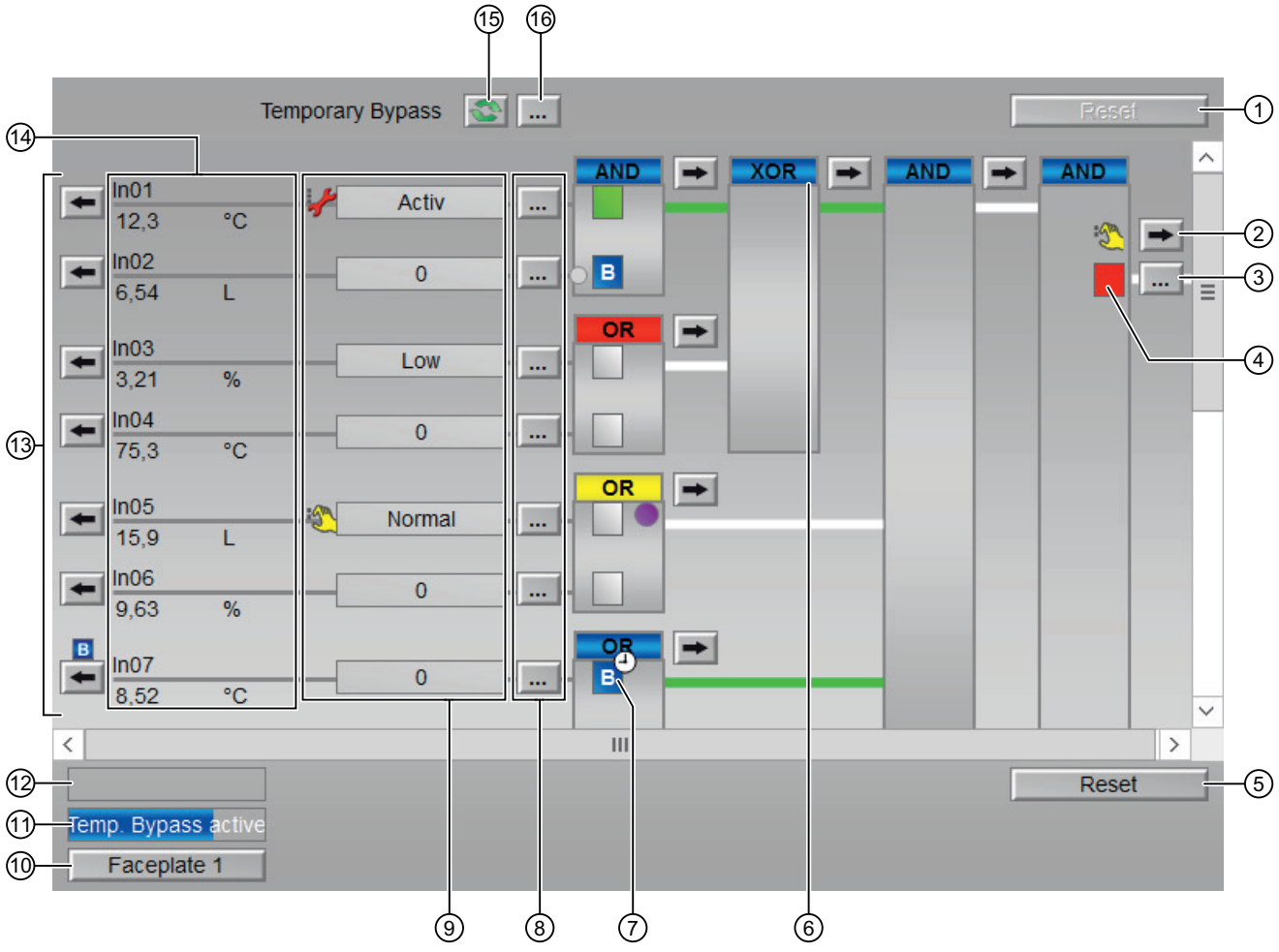
(9) Result of logic operation

The line indicates the status of the result of operation with its color:

Color of the line	Result of logic operation
Green	Output is enabled
White	Output is disabled

19.2.8.3 ILck8x4L extended standard view

ILck8x4L extended standard view



(1) Reset the output value

You can reset the output value with the button. The button is only operable if a reset is required.

The display of this button depends on the configuration in the engineering system (ES).

(2) Open the faceplate of the output value

If you press the button (➡), you can open the faceplate for the output value. The function of this button depends on the configuration in the engineering system (ES). Refer the Opening additional faceplates section of the APL documentation.

(3 / 8) Exclude input values / output value

You can use the button (3) to exclude input or output values from processing. Depending on the previous settings, you can "Set" or "Reset" this property.

If the input or output value has been excluded, the following symbol appears in the field (4 / 7):



You can find additional information on operation in the section "Switching operating states and operating modes" of the APL documentation.

Note

This function can only be executed in the faceplate with "high-level operating permission".

Note

The operator permissions via OS_Perm do not depend on the setting of bit 5. You can find additional information on this in the section "Enabling OS_Perm bits" of the APL documentation.

(4) Status of the block output

The field shows the status of the block output with the display:

Color of the line	Output status
	Output is not locked
	Output is interlocked
	Output bypassed
	Output is temporarily bridged

The line color indicates the status of the block output:

Color of the line	Output status
Green	Output is enabled
White	Output is disabled

(5) "Reset" the settings for further processing

When you press the button (10), you can "Reset" all input values:

- "Reset exclusions": the exclusions of the input values are reset.
- "Reset FirstIn": First-in detection / status display (4) is reset.

You can find additional information on this in the section Recording the first signal for interlock blocks of the APL documentation.

(6) Logic and status of the interlock

The area (6) shows you the function and status of the logical operator (priority from high to low). You can configure the logic in the engineering system (ES).

	Logic		
Color of the field	**AND**	**OR**	**XOR**
Blue	Excluded (bypass)		
Yellow	Simulated		
Red	Interlocked		
Green	Not interlocked		

(7) Display the status for further processing

The symbol shows the status for further processing of the input values:

Icon	Further processing
	The input value is processed further with value 1.
	The input value is further processed with the value 0.
	The input value is excluded from further processing.

(9) Display of input values (BOOL) with signal status (in front of the field)

These fields show the interlock information associated with the value (13) with a signal status:

- 1 = "Good" state
- 0 = "Locked"

Changing the display

You can change the displays for 0 and 1 in the CFC in the object properties of the Interlock block:

- Navigate to the I/Os (object properties).
- In the Text 0 and Text 1 columns, change the default for the input parameter (`Inxx`) to what you later want to see in runtime.

(10) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" of the APL documentation.

(11) Display area for block states

This area provides additional information on the operating state of the block:

- "Temporary bypass active"
  The color bar of the display area changes depending on the state. It shows the currently elapsed time of the set bypass time in bar graph and is initially blue.
- "Prewarning time active"
  If the temporary prewarning time for the temporary bypass is reached, the text and the display changes from blue to yellow.

(12) Display area for block states

This area provides additional information on the operating state of the block:

- "Delay active"
  The color bar of the display area shows the elapsed time of the configured time response in bar graph form.

(13) Open faceplate of the input value

When you press the button (), you can open the faceplate associated with each input value. The function of this button depends on the configuration in the engineering system (ES). Refer the Opening additional faceplates section of the APL documentation.

(14) Display of analog input values

The interconnected analog input values (`AVxx`) are displayed in this area. You set a unit of measure (`AVxx_Unit`) for each input in the engineering system (ES).

Changing the display

You can change the displays in the CFC in the object properties of the Interlock block:

- Navigate to the I/Os (object properties).
- In the "OS additional text" column, change the default setting for the input parameter (`INxx`) to what you want to see during runtime later.
- In the "Identifier" column, change the default setting for the input parameter (`AVxx`) to what you want to see during runtime later.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding`AVxxx` input.
- The font size is reduced during runtime if the length of the input text exceeds the width of the label.

Note

If a text is added to the "OS additional text" field of the input parameter (`INxx`), this text is displayed even then the "Identifier" text field of the input parameter (`AVxx`) is not empty. If the "OS additional text" field of the (`INxx`) input parameter is empty, the "Identifier" text of the (`AVxx`) input parameter is displayed.

(15) Reset temporary bypass bridging time

With this button, you can reset the running bypass time to the start value if you have the appropriate operator permission. The display of this button depends on the configuration in the engineering system (ES).

(16) Temporary bypass

Use this button, you can open the operating window for activating the temporary bypass if you have the appropriate operator permission.

Within the user interface, you can select the inputs and the output for which the temporary bypass is to be activated. While the operating window is open, the names of the signals are displayed in the extended standard view.

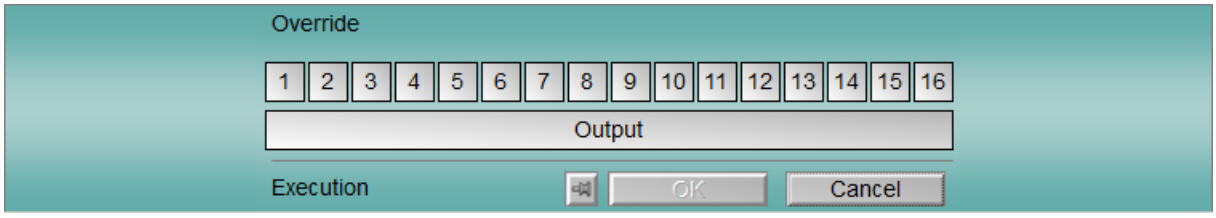
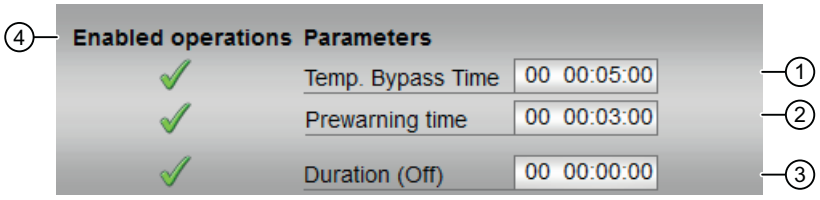


Figure 19-8 Operating window for setting a temporary bypass

19.2.8.4 ILck8x4L parameter view

ILck8x4L parameter view

In this area, you change the times and thus influence the behavior of the block.



(1) Time for temporary bypass

This time indicates how long a signal is excluded when a temporary bypass is activated.

(2) Prewarning time:

The prewarning time defines the time before the end of the temporary bypass at which a corresponding message is to be generated.

(3) Time response

The time for the time response configured at the block can be set here.

The display depends on the configuration of the time response.

You can influence the following parameters:

- Time (not enabled): A delay is not configured for the output; parameter cannot be manipulated.
- On-delay: On-delay is configured for the output. This can be controlled with corresponding operator authorization.
- Off-delay: Off-delay is configured for the output. This can be controlled with corresponding operator authorization.
- Duration (On/Off): The output is enabled/disabled time-controlled. The minimum time for which the output is active/inactive can be controlled with the appropriate operator permission.

(4) Enabled operations

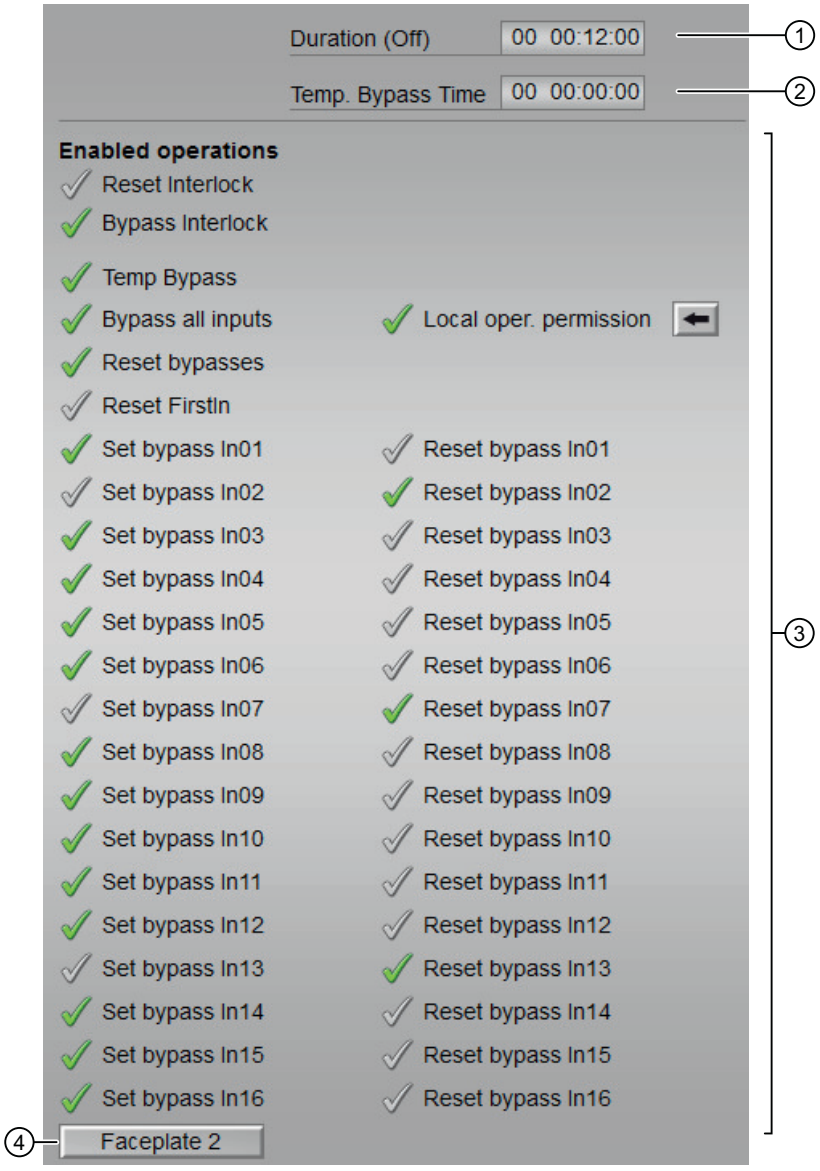
This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured operator authorizations.

19.2.8.5 ILck8x4L preview

ILck8x4L preview



(1) Time response

The current time for the time response configured for the block is displayed here.

The display depends on the configuration.

You can display the following times here depending on the configuration:

- Time (not enabled): No delay is configured for the output.
- On-delay: On-delay is configured for the output.

- Off-delay: Off-delay is configured for the output.
- Duration: The output is active time controlled.

(2) Time for temporary bypass

The current time of the temporary bypass is displayed here.

The display depends on the configuration of the block.

(3) Operator control enable

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: the OS operator cannot control this parameter due to the configured AS operator permissions (OS_Perm or OS1Perm).

The following enabled operations are shown here:

- "Local operating permission": Use the <-- button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(4) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system (ES). The visibility of this navigation button depends on the configuration in the engineering system (ES).

19.2.8.6 ILck8x4L block icon

Properties of the ILck8x4L block icon

A variety of block icons are available with the following functions:

- Signal status
- Memo display
- Output signal

The block icons from the template picture @TemplateIL_PCS_7.PDL:

Icons	Selection of the block icon in CFC	Block icon for
	1	ILck8x4L
	2	ILck8x4L

Block icons (mini) without display of instance-specific names

These icons only show the output signal. They take the form of a small rectangle.

Display of the output signal

The display can show the following states for the output signal (priority from high to low):

- Gray: No inputs interconnected at the interlock block, the block is not used
- Blue: The output signal is 1, at least one input signal is bypassed. There is no gray state.
- Yellow: The signal status is 16#60; the output signal is simulated. There are no gray and blue states.
- Red: The output signal is 0; there is an interlock. There are no gray, blue, and yellow states.
- Green: The output signal is 1; the block is in the good state. There are no gray, blue, yellow, and red states.

You can find additional information on the block symbol in the following sections of the APL manual:

- Configuring the block icons
- Block icon structure
- Operation via the block icon

Maintenance blocks 20

20.1 SimAn - Simulation block for analog values

20.1.1 SimAn description

Object name (type + number) and family

Type + number: FB 1100

Family: Maint

Number of process objects: 0 PO

Area of application

The block constitutes the interface between an analog channel driver block from the Advanced Process Library (APL) and the operator system (OS). In terms of the run sequence, the block is inserted upstream of the driver block.

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

Status word assignment for the parameter `Status`

For a description of the individual parameters, refer to the section SimAn I/Os (Page 1055).

Status bit	Parameter
0	`Sim1ActOut`
1	`Sim2ActOut`
2	`Sim3ActOut`
3	`Sim4ActOut`
4	`Sim5ActOut`
5	`Sim6ActOut`
6	`Sim7ActOut`
7	`Sim8ActOut`
8	`OpEnSiOff`

Status bit	Parameter
9	OpEnSiOn
10	OpEnSiValue
11 - 31	Not used

Status word assignment for the parameter Status1

For a description of the individual parameters, refer to the section SimAn I/Os (Page 1055).

Status bit	Parameter
0	Occupied
1	BatchEn
2 - 30	Not used
31	Not used

20.1.2 SimAn operating modes

The block does not have any operating modes.

20.1.3 SimAn functions

Configurable reactions using the Feature parameter

An overview of all the reactions provided by the Feature parameter can be found in the section titled *Configurable response using the Feature I/O* in the *APL Function manual*.

The following modes of behavior are available for this block at the relevant bits:

Bit	Function
0	1=Operator enable, simulation value 1
1	1=Operator enable, simulation value 2
2	1=Operator enable, simulation value 3
3	1=Operator enable, simulation value 4
4	1=Operator enable, simulation value 5
5	1=Operator enable, simulation value 6
6	1=Operator enable, simulation value 7
7	1=Operator enable, simulation value 8
8 - 23	Not used
24	See APL function: "Enable local operating permission"
25 - 31	Not used

20.1.4 SimAn error handling

The block does not have any error handling.

20.1.5 SimAn messaging

The block has no message behavior.

20.1.6 SimAn I/Os

Input parameters

Parameter	Description	Data type	Default
`OpEnSiOff`	Enable 1=Operator may switch simulation off	BOOL	1
`OpEnSiOn`	Enable 1=Operator may switch simulation on	BOOL	1
`OpEnSiValue`	Enable 1=Operator may input simulation value	BOOL	1
`Scale1 ... Scale8 *`	Range of simulation value 1 ... 8	STRUCT • HIGH:REAL • LOW: REAL	- • 100.0 • 0.0
`Sim1ActOp ... Sim8ActOp *`	Operator input 1 ... 8 simulation on/off	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`Sim1ValueOp ... Sim8ValueOp *`	Operator input 1 ... 8 simulation value	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80

* Values can be written back to these inputs during processing of the block by the block algorithm.

Output parameters

Parameter	Description	Data type	Default
`Sim1ActOut ... Sim8ActOut`	Input 1 ... 8 simulation on/off	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
`Sim1ValueOut ... Sim8ValueOut`	Output 1 ... 8 simulation value	STRUCT • Value:REAL • ST: BYTE	- • 0 • 16#80
`Status`	Status	DWORD	16#00000000

20.1.7 SimAn block diagram

The block has no block diagram.

20.1.8 Operator control and monitoring

20.1.8.1 SimAn views

SimAn block views

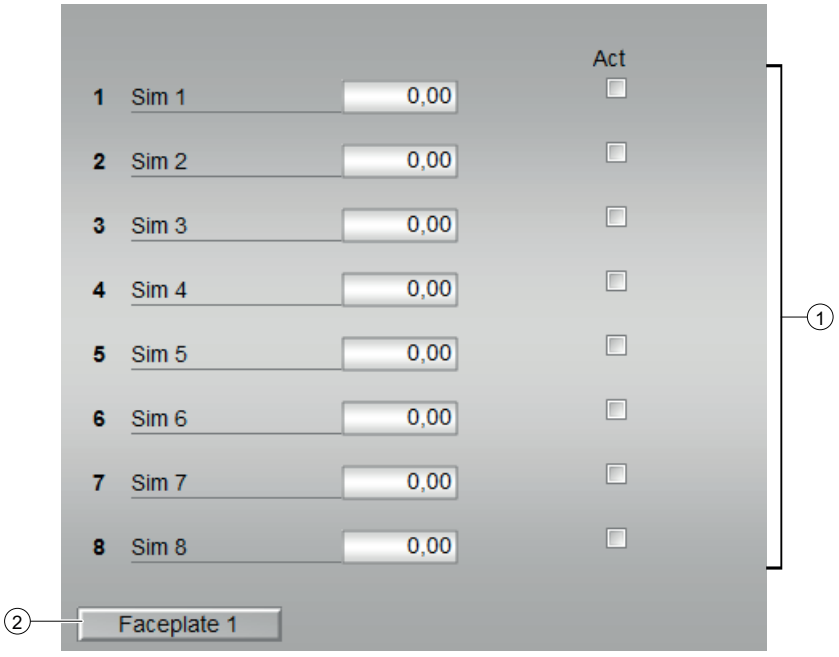
The SimAn block provides the following views:

- SimAn standard view (Page 1057)
- SimAn preview (Page 1058)
- Memo view
- Batch view
- SimAn block icons (Page 1058)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

20.1.8.2 SimAn standard view

SimAn standard view



(1) Change of the value 1 to 8

This area allows you to enter values. You can activate and deactivate the respective simulation value via the "Act." field.

Refer to the section "Changing values" in the APL manual for information about changing the simulation value.

You can freely rename the displayed texts as described in the section "Labeling of buttons and texts" in the APL manual.

Change the text for the simulation values at the `SimXValueOp` parameter.

Changing the display of the text for "Simulation"

You can change the displays in the CFC in the object properties of the monitoring block:

- Navigate to the I/Os (object properties).
- In the "Identifier" column, change the default setting for the input parameter (`Sim1ValueOp` .. `Sim8ValueOp`) to what you later want to see in runtime.
- The text is used as a label and is therefore always displayed, in other words, it is independent of the signal status of the corresponding Simulation input.

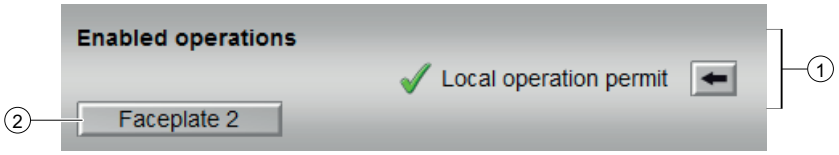
(2) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

20.1.8.3 SimAn preview

SimAn preview



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- "Local operating permission": Use the ← button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(2) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

20.1.8.4 SimAn block icons

There is no block icon for this block.

20.2 SimDi - Simulation block for digital values

20.2.1 SimDi description

Object name (type + number) and family

Type + number: FB 1101

Family: Maint

Number of process objects: 0 PO

Area of application

The block constitutes the interface between a digital channel driver block from the Advanced Process Library (APL) and the operator system (OS). In terms of the run sequence, the block is inserted upstream of the driver block.

Called blocks

The block calls no further blocks.

Calling OBs

The cyclic interrupt OB in which you install the block (e.g. OB32).

Status word assignment for the parameter `Status`

For a description of the individual parameters, refer to the section SimDi I/Os (Page 1061).

Status bit	Parameter
0	Sim1ActOut
1	Sim2ActOut
2	Sim3ActOut
3	Sim4ActOut
4	Sim5ActOut
5	Sim6ActOut
6	Sim7ActOut
7	Sim8ActOut
8	OpEnSiOff
9	OpEnSiOn
10	OpEnSiValue
11 - 31	Not used

Status word assignment for the parameter `Status1`

For a description of the individual parameters, refer to the section SimDi I/Os (Page 1061).

Status bit	Parameter
0	Occupied
1	BatchEn
2 - 31	Not used

20.2.2 SimDi operating modes

The block does not have any operating modes.

20.2.3 SimDi functions

Configurable reactions using the `Feature` parameter

An overview of all the reactions provided by the `Feature` parameter can be found in the section titled *Configurable response using the Feature I/O* in the *APL Function manual*.

The following modes of behavior are available for this block at the relevant bits:

Bit	Function
0	1=Operator enable, simulation value 1
1	1=Operator enable, simulation value 2
2	1=Operator enable, simulation value 3
3	1=Operator enable, simulation value 4
4	1=Operator enable, simulation value 5
5	1=Operator enable, simulation value 6
6	1=Operator enable, simulation value 7
7	1=Operator enable, simulation value 8
8 - 23	Not used
24	See APL function: "Enable local operating permission"
25 - 31	Not used

20.2.4 SimDi error handling

The block does not have any error handling.

20.2.5 SimDi messaging

The block has no message behavior.

20.2.6 SimDi I/Os

Input parameters

Parameter	Description	Data type	Default
OpEnSiOff	Enable 1=Operator may switch simulation off	BOOL	1
OpEnSiOn	Enable 1=Operator may switch simulation on	BOOL	1
OpEnSiValue	Enable 1=Operator may input simulation value	BOOL	1
Sim1ActOp ... Sim8ActOp	Operator input 1 ... 8 simulation on/off	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Sim1ValueOp ... Sim6ValueOp	Operator input 1 ... 8 simulation value	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80

Output parameters

Parameter	Description	Data type	Default
Sim1ActOut ... Sim8ActOut	Input 1 ... 8 simulation on/off	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Sim1ValueOut ... Sim8ValueOut	Output 1 ... 8 simulation value	STRUCT • Value:BOOL • ST: BYTE	- • 0 • 16#80
Status	Status	DWORD	16#00000000
Status1	Status word 1	DWORD	16#00000000

20.2.7 SimDi block diagram

The block has no block diagram.

20.2.8 Operator control and monitoring

20.2.8.1 SimDi views

SimDi block views

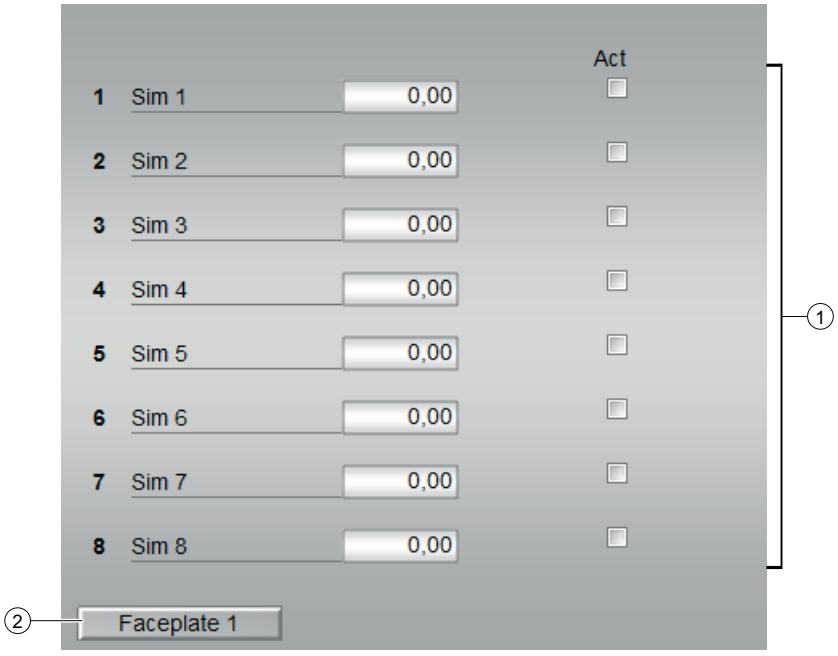
The SimDi block provides the following views:

- SimDi standard view (Page 1062)
- SimDi preview (Page 1063)
- Memo view
- Batch view
- SimDi block icons (Page 1064)

You can find general information about the faceplate and the block icon, in the sections "Structure of the faceplate" and "Block icon structure" in the manual of the APL.

20.2.8.2 SimDi standard view

SimDi standard view



(1) Display and switchover of the simulation value

This area shows the current selection. You can output a continuous signal as follows:

- "On": Continuous signal is output
- "Off"

You can find additional information on this in the section "Changing values" in the manual of the APL.

You can freely rename the displayed texts as described in the section "Labeling of buttons and texts" in the APL manual.

Do this with the following parameters:

- Text for "Sim 1" .. "Sim8": Parameter `SimXValueOp.Value#string_1`

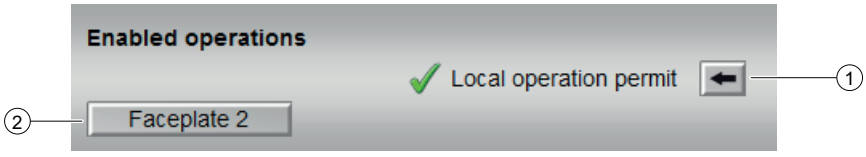
(2) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

20.2.8.3 SimDi preview

SimDi preview



(1) Enabled operations

This area shows all operations for which special operator authorizations are assigned. They depend on the configuration in the engineering system (ES) that applies to this block.

Icons for enabled operations:

- **Green check mark**: The OS operator can control this parameter
- **Gray check mark**: Access to this parameter is temporarily disabled for the OS operator due to process conditions
- **Red cross**: The OS operator cannot control this parameter due to the configured AS operator permissions (`OS_Perm` or `OS1Perm`)

The following enabled operations are shown here:

- "Local operating permission": Use the ← button to switch to the standard view of the OpStations block. You can find additional information on this in the section "Operator control permissions" in the manual of the APL.

(2) Navigation button for switching to the standard view of any faceplate

Use this navigation button to jump to the standard view of a block configured in the engineering system. The visibility of this navigation button depends on the configuration in the engineering system (ES).

You can find additional information on this in the section "Opening additional faceplates" in the manual of the APL.

20.2.8.4 SimDi block icons

There is no block icon for this block.

S88 structure interfaces and batch interface 21

21.1 UnitIf – Unit interface block

21.1.1 Description of 'UnitIf'

Area of application and operating principle

This unit interface block communicates with the SIMATIC BATCH UNIT_PLC block and the lower-level interface blocks EqmIf of the equipment modules. The block allows the operator to monitor and to control the unit (UNIT) with its lower-level equipment modules (EQMs) and control modules (CMs) by using the OS faceplate.

- An instance of the block UnitIf has to be set in the UNIT-CFC for each unit.
- Up to twelve EQM instances of the block EqmIf can be connected to a UnitIf interface block. The information and command transfer to the lower-level blocks is made via a 1:n data structure connection.
- Every EqmIf instance returns a feedback data structure (1:1 connection) to the UnitIf block.
- Sending of hold / resume commands to SIMATIC BATCH is implemented through the inputs `RqHold01 ... RqHold16` and `RqResu01 ... RqResu08` . Correspondingly, executing equipment functions belonging to the unit can be held / resumed by SIMATIC BATCH (further information is available in SIMATIC BATCH description).

  Note

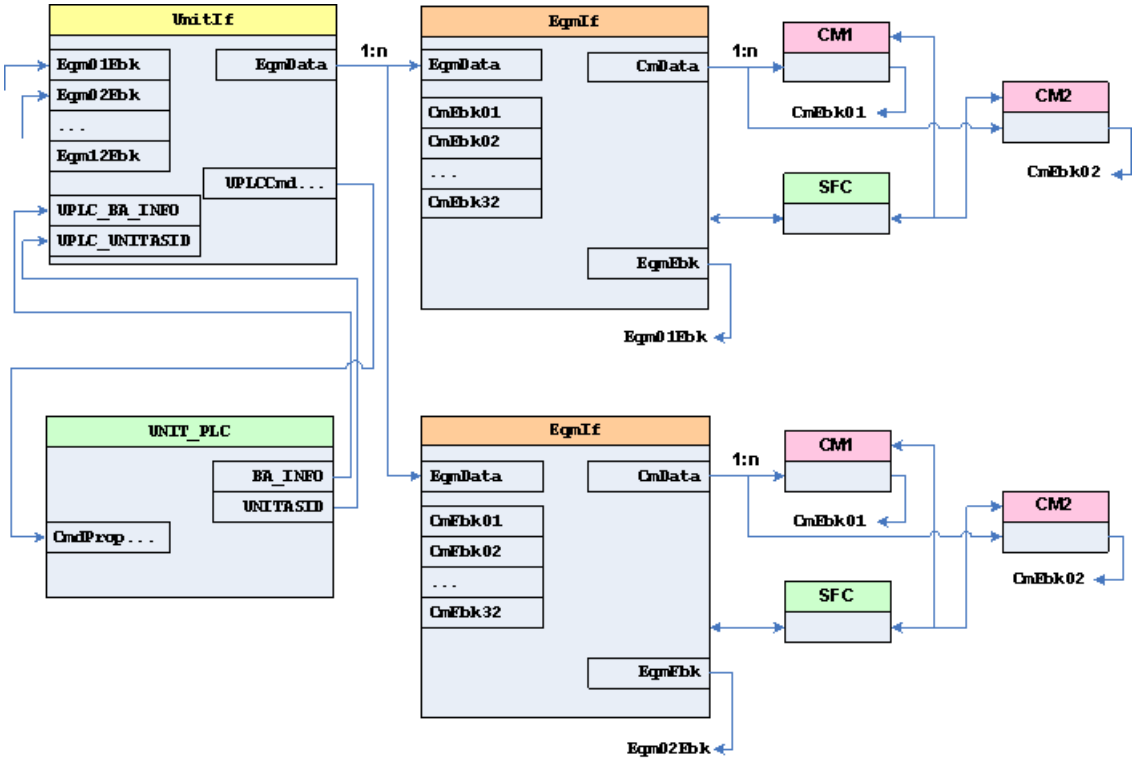
  The following functionality is only supported in AS-based mode.

- The faceplates of the respective equipment modules (block 'EqmIf') can be called from the faceplate of the UnitIf .

Configuration

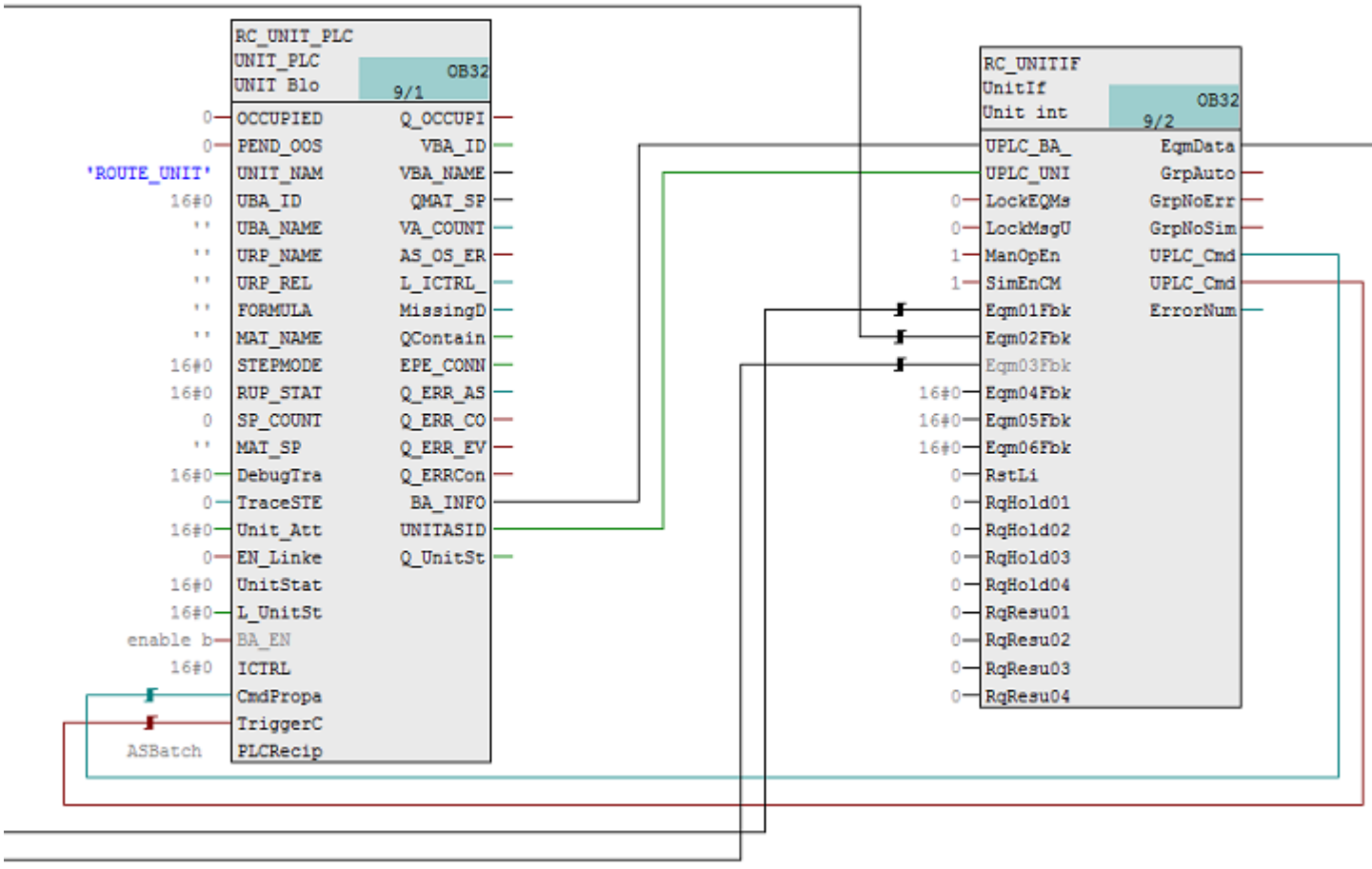
Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The EqmIf block is required for each equipment module and is instantiated together with the SFC type in the EQM-specific CFC.

The following figure shows the required interconnection schematically:


UnitIf
Egm01Ebk
Egm02Ebk
...
Egm12Ebk
EgmData
UPLCCmd...
UPLC_BA_INFO
UPLC_UNITASID
1:n
EgmIf
EgmData
CmData
CmEbk01
CmEbk02
...
CmEbk32
EgmEbk
Egm01Ebk
1:n
CM1
CmEbk01
CM2
CmEbk02
SFC
UNIT_PLC
BA_INFO
UNITASID
CmdProp...
EgmIf
EgmData
CmData
CmEbk01
CmEbk02
...
CmEbk32
EgmEbk
Egm02Ebk
1:n
CM1
CmEbk01
CM2
CmEbk02
SFC

Interconnection UNIT_PLC - UnitIf

The UnitIf block is instantiated together with the SIMATIC BATCH block UNIT_PLC in the UNIT-specific CFC.



Signal transmission from UNIT_PLC to UnitIf

- Data structure connection `BA_INFO → UPLC_BA_INFO` (`UNIT_NAME, VBA_NAME, VBA_ID, STEP_NO, Q_STATUS`)
  The unit PLC status information `Q_STATUS` is passed on to the output STATUS2. The bits BA_EN and OCCUPIED are decoded and passed on to the corresponding outputs.
- Connection `UNITASID → UPLC_UNITASID`
  The project-wide unique IDs are assigned by SIMATIC BATCH and passed on through the data structure output `EqmData` to the lower-level modules. There they serve to identify the messages (message associated value 4), allowing unit-specific filtering of the messages in the faceplate.

Signal transmission from UnitIf to UNIT_PLC

- `UPLC_CmdProp` → `CmdPropagation` (command transfer)
  The output has a corresponding command identifier (for example 4 = HOLD or 6 = RESUME) assigned to it and sent through the connection to the UNIT_PLC block of SIMATIC BATCH.
- `UPLC_CmdTrig` → `TriggerCmdPropagation` (trigger command transfer)
  The output is set when an S88 status change is to be passed on to the UNIT_PLC.

Interconnection UnitIf - EqmIf

- A 1:n data structure connection `EqmData` with the following elements:

Structure parameters

Structure element	Description	Type
`EqmData`	Cumulated EQM – info data structure	STRUCT
`dwUnitID`	Unit ID	DWORD
`xUnitOcc`	Unit is allocated	BOOL
`xUnitOccPulse`	Unit uses pulse	BOOL
`xEqmLock`	Lock equipment module	BOOL
`xOnOccAuto`	Switch to automatic mode when unit is allocated	BOOL
`xManOPEn`	Enable manual operation for operator	BOOL
`xCMSimEn`	Simulation release	BOOL
`xUnitSimOn`	Unit simulation is active	BOOL
`xResetPulse`	Reset pulse	BOOL
`xSwitchToManual`	Command manual operation	BOOL
`xSwitchToAuto`	Command automatic mode	BOOL
`xBatchEnabled`	Batch is enabled	BOOL
`xOccupyEqm`	Equipment module occupies request	BOOL
`xMsgLock`	Lock messages for all lower-level modules	BOOL
`xRes3`	(Reserve)	BOOL
`xRes4`	(Reserve)	BOOL
`xRes5`	(Reserve)	BOOL
`byRes6`	(Reserve)	BYTE
`byFbkDelayRatio`	Unit-wide delay ratio for the simulation of CM feedback signals	BYTE
`dwBatchID`	Batch ID	DWORD
`sBatchName`	Batch name	STRING[32]

Feedback signal transmission EqmIf → UnitIf

- One 1:1 data structure connection `EqmFbk` to `Eqm01Fbk ... Eqm12Fbk` per EQM (up to 12 EQMs can be interconnected). The data structure contains the feedback information of the respective equipment module that is used for the status display in the UnitIf faceplate. (The internal bitwise encoding is not relevant for the user.)

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'UnitIf' (Page 1073)".

Status bit	Parameter
0	BatchOcc (=1, when batch is allocated to unit)
1	BatchEn (=1, when unit is enabled for batch allocation)
2	= Input parameters ReqHold
3	= Input parameter ReqResume
4	= Input parameter ManOpEn
5	= Input parameter SimEnCM
6	= Input parameter LockEQMs
7	Reserve
8	xAutModOne (=1, if one of the EQM-SFCs or CMs in AUTO)
9	xManModOne (=1, if one of the EQM-SFCs or CMs in AUTO)
10	xHRQInitiator (=1, if one of the EQMs signals HOLD request)
11	xSimOnOne (=1, if one of the EQMs or CMs in simulation mode or Unit is on simulation)
12	GrpError (=1, if one of the EQM-SFCs or CMs signals error)
13	xUnitSim (=1, simulation is enabled for CMs & Unit is on simulation)
14	xRtError (=1, if hold and resume requests are active simultaneously)
15	= Input parameter LockMsgUnit (unit message lock active)

Status word allocation 'Status2'

The status word contains a copy of the status word UPLC_BA_INFO.Q_STATUS of the type DWORD. Then input parameter UPLC_BA_INFO is interconnected with the output BA_INFO of the SIMATIC BATCH block UNIT_PLC . For a description of the individual status bits, see the SIMATIC BATCH documentation.

21.1.2 Functions of 'UnitIf'

The block makes the control and switchover functions that are described below available on the unit level. These are distributed through the described signals in the data structure connection `EqmData` to all the lower-level elements (EQMs and CMs). In order for them to become effective at the control modules, the corresponding interconnections between the adapter block and the APL block have to exist. Additional details are available in the documentation of the adapter blocks. In addition the type-specific process tag templates in the supplied demo project can be used for reference purposes.

- Enabling of the simulation operation
- Activation of the simulation operation
- Enable for the switchover to manual operation
- Mode switch AUTO / MANUAL
- Locking signal
- Reset command

Additional functions of the block:

- The interconnectable input `RstLi` reacts edge-triggered to a positive (0 → 1) signal change.
- In case of an occupation of a unit, all the lower-level motors / valves are switched to AUTO mode. In detail the CMs are set to the following commands to this purpose:
  - AUTOMATIC
  - SIMULATION OFF
  - RESET
    These functions can be activated through separate Feature Bits (see below).
- "Request HOLD" function and hold diagnostics:
  - External hold requests can be fed through the input parameters `RqHold01` … `RqHold16` . If the unit is in the "Running" state, active hold requests are displayed in the "Hold diagnostics" view in the OS plate.
  - If holding was due to block-internal hold requests, the cause is displayed in the "Hold diagnostics" view. If a CM error was responsible, it is also displayed here (only EqmIf).
  - The hold history (signal incoming / outgoing) remains until the "Resume" command (also: "Reset"). It is cleared at the latest when the rest position "Idle" has been reached.
  - The hold causes (texts) are configured in the signal properties of the respective input (call via DK for structure element 'Value').
  - The SIMATIC BATCH block UNIT_PLC is informed through the output parameter `UPLC_CmdProp` / `UPLC_CmdTrig` that a switch to HOLD is to be carried out. The EQM SFCs are then also set to HOLD.

- "Request RESUME" function:
  - External resume requests can be fed through the input parameters `RqResu01` ... `RqResu08` .
  - The SIMATIC BATCH block UNIT_PLC is informed through the output parameter `UPLC_CmdProp` / `UPLC_CmdTrig` that a switch to RESUME is to be carried out. The EQM SFCs are then also set to RESUME.
  - In the "RESUMING" operating state the commands AUTO, SIMULATION OFF and RESET are also sent to the CMs. This can be activated via the Feature Bit.
- Function mode: MANUAL / AUTO
  - Changing over in the faceplate is always free, irrespective of whether UNIT is allocated or not.
  - In case of MANUAL the EQMs and CMs are switched via the data structure connections to MANUAL.
  - If an assigned EQM switches by itself to MANUAL, this fact is communicated through the feedback connections and displayed in the UnitIf faceplate.
  - If the UNIT is not running, the switchover nevertheless becomes effective at the inputs `ManModOp`, `AutModOp` .
  - `GrpAuto` is always tracked.
- "Reset" function
  - Input `RstOp` or `RstLi` is passed on to the output `ResetOut` as well as in the structure bit `EqmData.xResetPulse` .
  - `P_Rst` is also set at all the assigned EQMs (using `EqmData`).
  - In the case of CMs with an active fault (watchdog) the RESET function is carried out.
  - The operator input `RstOp` is reset automatically in the next cycle (=0).

- Simulation function: ON / OFF
  - The possibility of simulation of the control modules is enabled with `SimEnCM` .
  - Input `SimEnCM` is passed on to the data structure `EqmData.xSimEnCM` . In the faceplate the selection dialog is enabled or locked.
  - With the operation signal `SimOnOp` the simulation is passed on through the data structure `EqmData.xUnitSimOn` to the lower level EQMs and switched on. The `UnitSimOut` output is set (= 1).
  - At a set Feature Bit1 (= 1) the simulation is deactivated, if BATCH is allocated to the unit (`EqmData.xUnitOcc = 1`) and `UnitSimOut` is reset (=0).
- Passing on unit-specific commands:
  - Input `LockEQMs` is passed on to the structure element `EqmData.xEqmLock` and output `LockEQMsOut` . This is also distributed to the CMs using the block EqmIf , meaning that all connected CMs are locked.
  - Input `ManOpEn` is passed on to the structure element `EqmData.xManOpEn` and output `ManOpEnOut` . This is also distributed to the CMs using the block EqmIf , meaning that the switchover option is locked at all the connected CMs.
  - Input `LockMsgUnit` is passed on to the structure element `EqmData.xMsgLock` . This is also distributed to the CMs using the block EqmIf , meaning that the messages are suppressed at all the connected CMs.

Call additional faceplates

This block includes the function "Call additional faceplates". To this purpose the input parameter `Eqm01Fbk ... Eqm12Fbk` has the attribute 'Blk_Jump'. If CFC interconnections are available at these inputs, the respectively interconnected EqmIf instances can be opened in the OS faceplate.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the Feature parameter is available in the "APL documentation" in the section "Configurable functions with the Feature I/O". The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	1 = At unit allocation: Switch all the lower-level elements (CMs, SFCs) to AUTO mode
1	1 = At unit allocation: Deactivate the unit simulation
2	1 = At unit allocation: Reset all the lower-level CMs
3	1 = At Resume: Switch all the lower-level elements (CMs, SFCs) to AUTO mode

Bit	Comment
4	1 = At Resume: Deactivate the unit simulation
5	1 = At Resume: Reset all the lower-level CMs
...	...
24	See APL function: "Enabling local operator permission" 1 = "Local" operator permission is active
...	...

Operator permissions

This block includes the APL standard function "Operator permissions". The block has the following permissions for the `OS_Perm` parameter:

Status bit	Parameter
0	1= Operator can switch to "Auto" mode
1	1= Operator can switch to "Manual" mode
2	1= Operator can activate the unit simulation
3	1= Operator can deactivate the unit simulation
4	1= Operator can trigger a RESET command
5	Reserve
...	...

Overview of error numbers

The ErrorNum connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.
41	If hold and resume requests are set simultaneously

21.1.3 Connections of 'UnitIf'

Input parameters

Parameter	Description	Type	Default
`UPLC_BA_INFO`	Connect to UNIT_PLC's BA_INFO output structure	`BlUdtBatchInfo`	
`UPLC_UNITASID`	Connect to UNIT_PLC's UNITASID output	`DWORD`	
`ManModOp`	Switch equipment modules to manual mode	`BOOL`	

Parameter	Description	Type	Default
AutModOp	Switch equipment modules to automatic mode	BOOL	
SimOnOp	Activate simulation	BOOL	
LockEQMs	Lock EQMs	BOOL	
LockMsgUnit	Lock messages for all connected subordinate modules	BOOL	
ManOpEn	Enable CMs' manual mode switch for operator	BOOL	True
SimEnCM	Enable CMs' local simulation	BOOL	True
Eqm01Fbk ... Eqm12Fbk	Equipment module feedback 1 ... 12	BlUdtEqmFbk	
RstLi	Reset equipment modules (process)	DigVal	
RstOp	Reset equipment modules (operator)	BOOL	
RqHold01 ... RqHold16	External HOLD request 1 ... 16	DigValFF	
RqResu01 ... RqResu08	External RESUME request 1 ... 8	DigValFF	
OS_Perm	Operator Permissions	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	
FbkDelayRatio	Ratio between CM monitoring time and feedback delay during unit simulation	INT	50

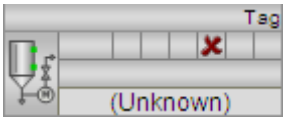
Output parameters

Parameter	Description	Type	Default
EqmData	Equipment data	BlUdtEqmData	
ReqHold	A hold request to UNIT_PLC is active	BOOL	
ReqResume	A resume request to UNIT_PLC is active	BOOL	
GrpAuto	Collected state: 1=all CMs are in automatic mode (or On)	BOOL	
GrpNoError	Collected state: 1=no CM has an error	BOOL	
GrpNoSim	Collected state: 1=no CM is simulating	BOOL	
LockEQMsOut	LockEqms copy	BOOL	
ResetOut	Reset pulse	BOOL	
ManOpEnOut	ManOpEn copy	BOOL	
UnitSimOut	Unit simulation is active	BOOL	
UnitASID	AS and unit ID	DWORD	
UnitName	Unit name	STRING[24]	
BatchID	Batch ident number	DWORD	
BatchName	Batch name	STRING[32]	

Parameter	Description	Type	Default
StepNo	Step number from EPH	DWORD	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Batch unit is occupied	BOOL	
OS_PermOut	Operator permission: output for OS	DWORD	
OS_PermLog	Operator permission: output for OS	DWORD	
OpSt_Out	Enabled operator stations	DWORD	
ST_Worst	Worst signal status	BYTE	16#80
Status1	Status 1 data	DWORD	
Status2	Status 2 data	DWORD	
HoldReqExtAct	Internal	WORD	
HoldReqExtPin	Internal	WORD	
HoldReqExtUse	Internal	WORD	
UsedEQMs	Connected signals	WORD	
UPLC_CmdProp	Command value for UNIT_PLC block	INT	4
UPLC_CmdTrig	Command trigger for UNIT_PLC command	BOOL	
ErrorNum	Error number	INT	-1

21.1.4 Operator control and monitoring

21.1.4.1 Symbol

	Tag name Status bar Unit name Unit state (S88 state from SIMATIC BATCH)

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	Common operating mode of lower-level elements	
		Automatic
		Manual
		Auto / Manual mixed
2		At least one external hold request active
3		At least one lower-level element has errors

Position	Content	
4		At least one lower-level element is in simulation
5		Unit not released
		Unit released
6		Unit allocated
7		Notice is available

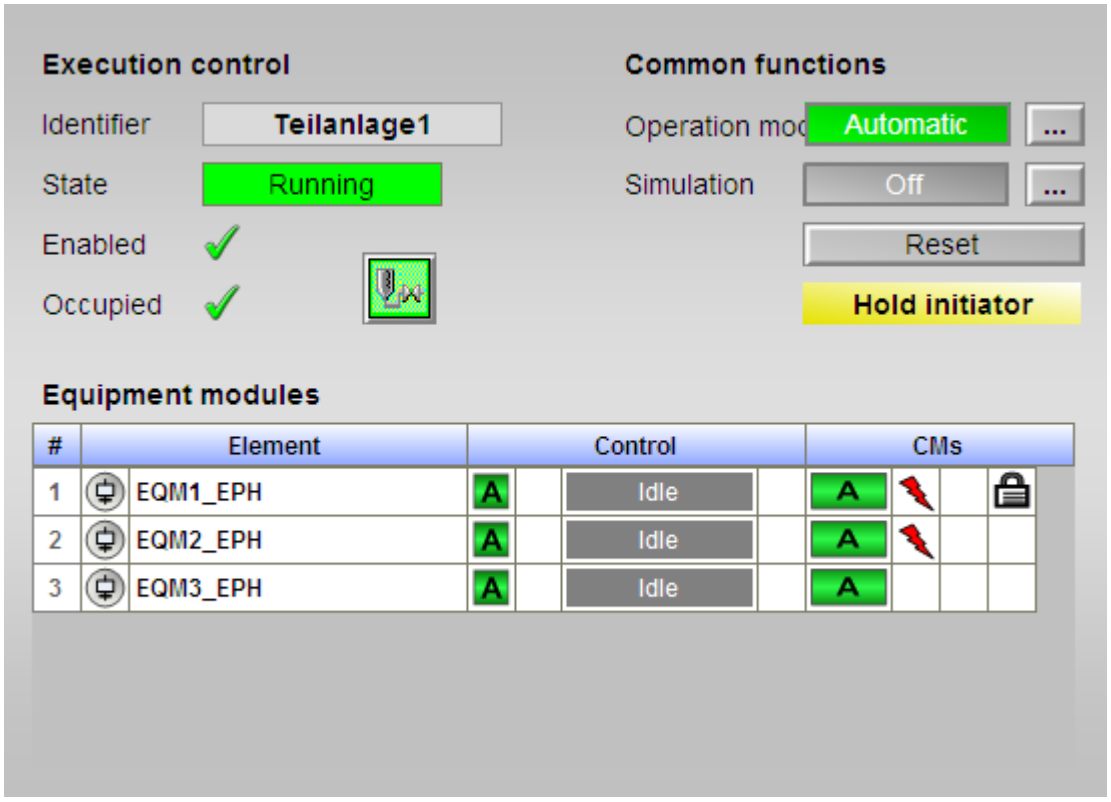
21.1.4.2 Faceplate

Views of the Unitlf block

The block Unitlf provides the following views:

	Standard view	See below.
	Alarm view	See below.
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library
	Hold diagnostics view	See below.

Standard view of the Unitlf

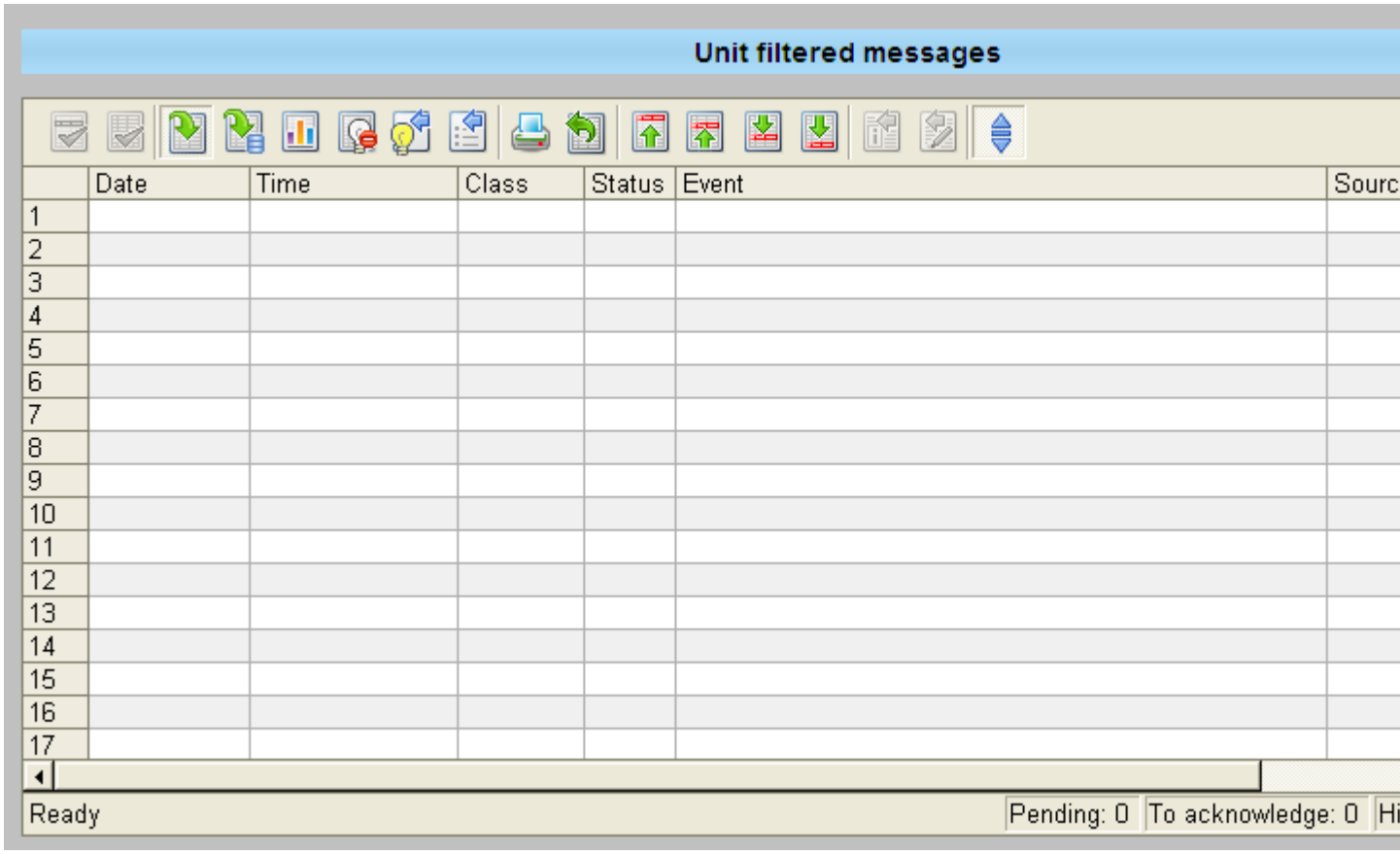


Explanation of the screen shot	
(1)	**Displaying and switching the operating mode**
	This section shows the currently valid operating mode for the UNIT as well as the lower-level equipment modules and control modules. The following operating modes can be shown here: • Automatic mode → lower-level EQMs and CMs are in automatic mode • Manual mode → lower-level EQMs and CMs are in manual mode • (Mixed) → lower-level EQMs or CMs have different operating modes The button can be used to switch the UNIT incl. the lower-level EQMs and CMs to the operating mode 'Automatic' or 'Manual operation'.
(2)	**Displaying and switching the unit simulation**
	This section shows the state of the unit simulation. The following states can be shown and executed here: • On • Off
(3)	**Resetting the modules**
	In case of a corresponding permission this button can be used to send a reset request to all the lower-level modules through an additional operating dialog.
(4)	**Hold initiator**
	The flashing text box is displayed when a hold request from external signals or lower-level EQMs is active. Further information is available in the following sections.

Explanation of the screen shot			
(5)	**Unit**		
	The unit section displays the current S88 state as well as whether the UNIT is enabled for SIMATIC BATCH and a batch is allocated to it.		
		This button can be used to call the faceplate of the SIMATIC BATCH block UNIT_PLC.	
(6)	**Equipment modules**		
	This section contains a table with all the connected equipment modules (EQMs) with their states. The Element column shows the names of the `,EqmIf'` block instances of the equipment modules of this unit.		
	• The 'Control' column contains the following information:		
			Automatic
			Manual operation of the EQM SFCs
			Fault of the EQM SFCs
		Text - S88 state of the EQM SFCs	
			Hold request is active
	• The CM column shows a common criterion for the lower-level control modules of the EQM. CM states are:		
		Operating mode	 All in automatic mode
			 All in manual mode
			 Partially in manual / automatic mode
			Monitoring error
			Simulation
			Interlock
	• Clicking on an EQM line calls the corresponding EQM faceplate.		

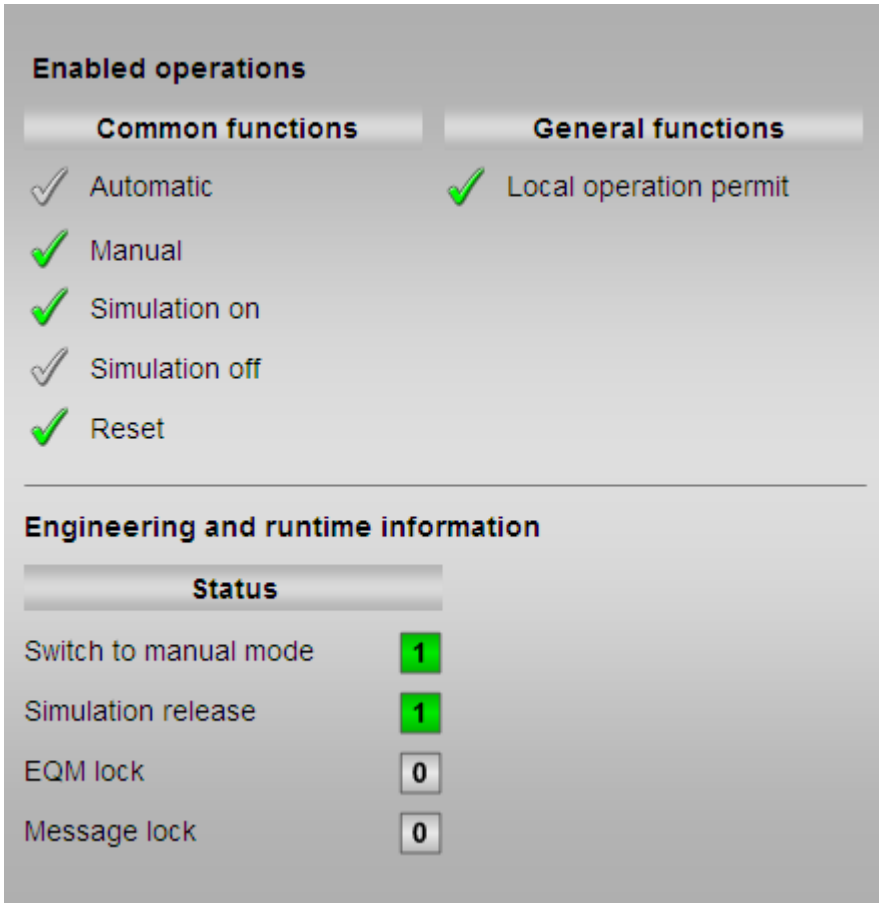
Alarm view of UnitIf

This is the standard alarm view of all the blocks of the Advanced Process Library. All the messages of the lower-level elements of this unit are displayed here.



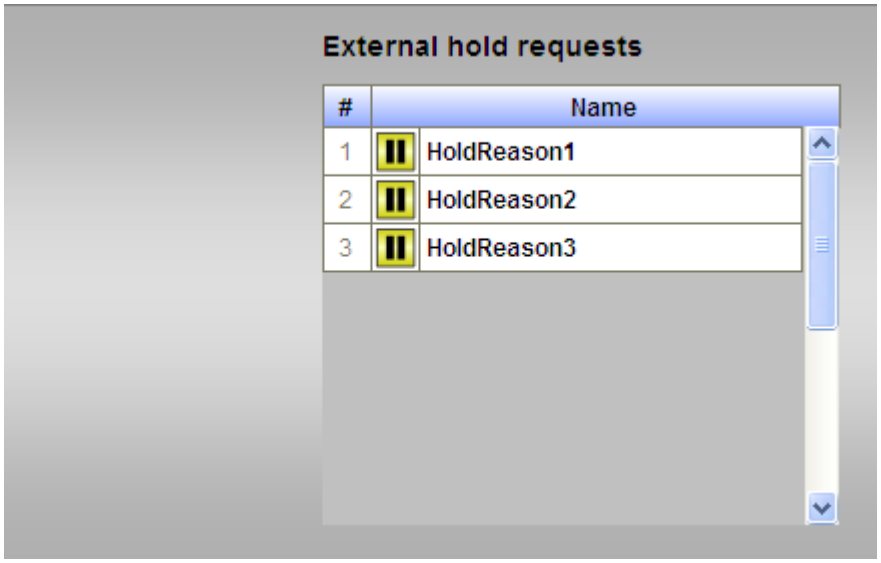
The "Source" column displays the triggering element.

Preview of Unitlf



Explanation of the screen shot		
(1)	**Operator control enables**	
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the `OS_Perm` parameter of this block. The operator control enables shown in the screenshot are possible here. • Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions	
(2)	**Engineering and runtime information**	
	This section contains information from the status word `Status1`. The 'States' column shows the following state bits (0 / 1):	
	• Enable manual mode	Input parameter `ManOpEn`
	• Simulation release	Input parameter `SimEnCM`
	• EQM lock	Input parameter `LockEQMs`
	• Message lock	Input parameter `LockMsgUnit`

Hold diagnostics view of the Unitlf



Explanation of the screen shot	
(1)	**States of the external input parameters**
	This view displays the states of the external input parameters `RqHold01 ... RqHold16` . The display window is not filled until the moment from which the UNIT is running and a hold request becomes active. The initial triggers (hold requests active in the same AS cycle) are highlighted in color, signals arriving later are only displayed as long as they are active. • The following information is displayed in the 'Causes' column:
	Hold request active
	Hold request outgoing
	Hold request initial value / active
	Hold course = 'Text 1" attribute of the external input parameter. Texts of the initial trigger signals are highlighted in color. This marking is retained even when the signal has gone again.
	The hold history remains until the "Resume" command (also: "Reset"). It is cleared at the latest when the rest position "Idle" has been reached.

21.2 EqmIf – Equipment module interface block

21.2.1 Description of 'EqmIf'

Area of application and operating principle

This function block communicates with the upstream BlUnitIf interface block as well as with the lower-level control modules (CMs) of the corresponding equipment module (EQM). The block allows the operator to monitor and to control the equipment module with its lower-level device modules and control modules (CMs) by using the OS faceplate.

The EqmIf block is required for each equipment module and is instantiated together with the SFC type in the EQM-specific CFC.

Up to 32 control modules can be connected to a EqmIf interface block using type-specific adapter blocks.

Each control module or the associated adapter block returns a feedback data word to the EqmIf block.

Hold commands at SIMATIC BATCH can be requested by using the inputs `RqHold01 ... RqHold16`. Correspondingly, executing equipment functions belonging to the unit can be held by SIMATIC BATCH (further information is available in SIMATIC BATCH description).

Note

The following functionality is only supported in AS-based mode.

The online views of the associated control modules can be called from the faceplate of the EqmIf .

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38).

Interconnection UnitIf - EqmIf

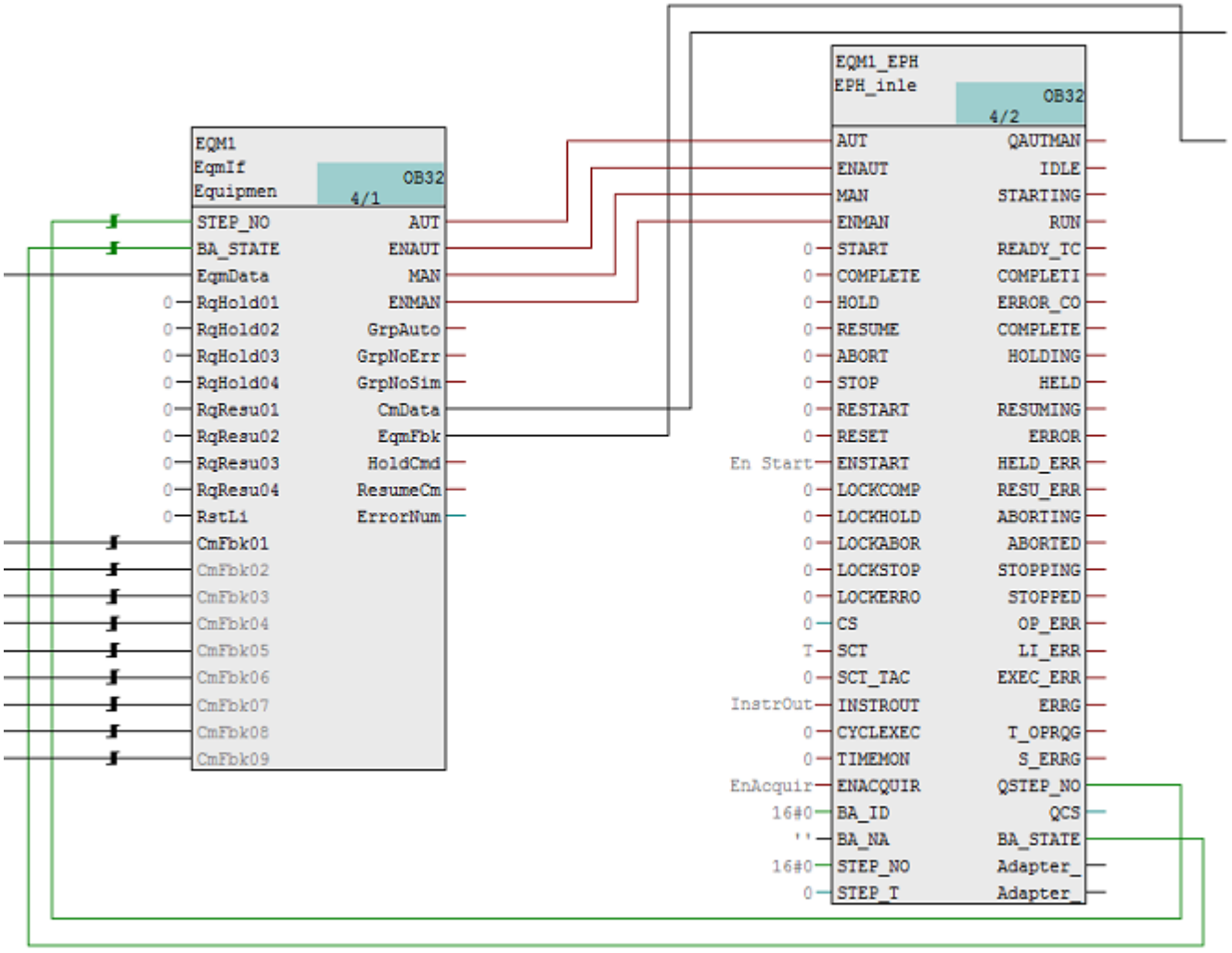
The interconnection between the blocks UnitIf and EqmIf is described in the section "UnitIf – Unit interface block (Page 1065) .

Interconnection EqmIf – CM adapter/CM

The interconnection of the control signals of the lower-level control modules or of their adapter blocks with the EQM interface block is made on the basis of a 1:n relationship, meaning that from the EqmIf a connection of the output `CmData` to each assigned CM is required.

For further details refer to the section "Higher-level controlling of the unit / EQM level (Page 70)"

CFC example:



Status word allocation 'Status1'

For a description of the individual parameters see the section "Connections of 'UnitIf' (Page 1073)".

Status bit	Parameter
0	xSfcOccupied (= 1 if SFC allocated by SIMATIC BATCH)
1	xSfcBatchEn (= 1 if SFC signals batch enable)
2	xSfcModeAuto (=1, if SFC in AUTO)
3	SimOnCM (=1, when simulation operation is active)

Status bit	Parameter
4	xAutModOne (=1, if one of the CMs in AUTO)
5	xManModOne (=1, if one of the CMs in MANUAL)
6	xErrorOne (=1, if one of the CMs reports error)
7	xHoldTrigger (=1, if a hold request is active at this EQM; external or from CM error)
8 ... 15	= BA_STATE of the connected EQM SFC (S88 state of the SFC type)
16	xGrpError (=1, if one of the CMs reports error)
17	xRtError (=1, if a hold request AND resume request are active simultaneously at this EQM)
18	xUnitMsgLock (=1, if message lock of unit is set)
...	Reserve

21.2.2 Functions of 'EqmIf'

The following control and switchover functions are available on the EQM level. These are signaled via the described data structure connections to the lower-level module:

- Enable for the switchover to manual operation for lower-level modules
- Mode switch for the EQM SFC as well as all lower-level CMs
- Automatic mode
- Manual mode
- Interlock signal for lower-level modules
- Reset command for lower-level modules
- Enable and activation of the unit simulation

Additional functions of the block:

- In the case of an allocation of a unit simulation operation is deactivated at all the lower-level motors / valves. This can be activated via the Feature Bit (see below).

"Request HOLD" function and hold diagnostics:

- External hold requests can be fed through the input parameters `RqHold01 ... RqHold16` . If the unit is in the "Running" state, active hold requests are displayed in the "Hold diagnostics" view in the OS plate.
- If holding was due to block-internal hold requests, the cause is displayed in the "Hold diagnostics" view.
- Through the CM enable inputs `Cm01EnHold ... Cm16EnHold` a hold request can be generated at CM group errors. Display is made in the "Hold diagnostics" view.

- The hold history (signal incoming / outgoing / initial value) remains until the "Resume" command (also: "Reset"). It is cleared at the latest when the rest position "Idle" has been reached.
- The hold causes (texts) are configured in the signal properties of the respective input (call via DK for structure element 'Value').
- The hold request is reported to the block BlUnitIf through the data structure connection `EqmFbk` and from there passed on to the SIMATIC BATCH block UNIT_PLC.

"Request RESUME" function:

- External resume requests can be fed through the input parameters `RqResu01` ... `RqResu08` .
- This is reported to the block BlUnitIf through the data structure connection `EqmFbk` and from there passed on to the SIMATIC BATCH block UNIT_PLC. The other EQM-SFCs are then also set to RESUMING.

Function mode: MANUAL / AUTO

- Changing over in the faceplate is always possible, whether the UNIT is running or not.
- If UNIT is set to MANUAL, the EQM SFC is also switched to MANUAL through the block (output parameter `ENMAN`, `MAN`) and the CMs are switched to MANUAL through the data structure connections.
- If the EQM SFC itself is switched to MANUAL (recognition via input `BA_STATE`), the operating mode is also written back to the BlUnitIf/UNIT_PLC through the data structure connection `EqmFbk` .
- If a lower-level CM is switched to MANUAL, this fact is communicated through the feedback connections and displayed in the faceplate.
- If the UNIT is not running, the switchover by the operator nevertheless becomes effective at the inputs `ManModOp`, `AutModOp` .
- `GrpAuto` is always tracked.

If AUTO:

- The output: `AutModLi` is set for one cycle.
- `AUT/ENAUT/ModLiOp` is set for the duration of one cycle at a mode change (MANUAL to AUTO) or during forced automatic mode during the allocation.
- The structure bits `xAutModLi` and `xModLiOp` are set for one cycle for the CMs in `CmData` . This means that the CMs switch to automatic mode.
- `xModLiOp` is tracked via the adapter block to the switchover signal input `ModLiOp` of the CM block and there decides which control inputs are used for the mode switch: `AutModLi/ ManModLi` or `AutModOp/ManModOp`

If MANUAL:

- The output: `ManModLi` is set for one cycle.
- The outputs: `MAN/ENMAN/ModLiOp` are set for one cycle.
- The structure bits `xManModLi` and `xModLiOp` are set for one cycle for the CMs in `CmData` . This means that the CMs switch to manual mode (see above).

"Reset" function

- At an active input `RstOp` or a positive edge at the input parameter `RstLi` a reset pulse is generated at the output `P_Rst` as well as in the structure bit `CmData.xResetPls` . This carries out the RESET function at CMs with an active fault (watchdog).
- The operator input `RstOp` is reset automatically in the next cycle (=0).

Simulation function: ON / OFF

- The simulation enable is received by the unit through the data structure `EqmData.xSimEnCM` and passed on through the output structure `CmData.xSimEnCM` to the lower-level CMs. In the CM faceplates the selection dialog is enabled or locked.
- With the faceplate operation signal `SimOnOp` or at a unit simulation (data structure `EqmData.xUnitSimOn`) the simulation is passed on through the data structure `CmData.xSimOnParent` to the lower-level CMs and switched on.
- At a set Feature Bit 2 = 1 the simulation is deactivated, if the unit is allocated by BATCH `CmData.xBatchOcc` = 1) and `SimOnCM` is reset (=0).

Function "Enable operator input manual mode":

- The structure bit command `EqmData.xManOpEn` is passed on to the output `ManOpEn` as well as structure bit `CmData.xManOpEn` .
- The `CmData` structure is passed on to the CM adapter blocks. All the lower-level CMs are then enabled for the operator switchover "MANUAL/AUTO".

Function General locking

- If the input `LockEQMs` is set at BlUnitIf , the output `LockEQMsOut` is set.
- The CMs are set through the structure bit `CmData.xEqmLock` to "EQM general lock".

Call additional faceplates

This block includes the function "Call additional faceplates". To this purpose the input parameter `Cm01Fbk ... Cm32Fbk` has the attribute 'Blk_Jump'. If CFC interconnections are available at these inputs, the respectively interconnected CM instances can be opened in the OS faceplate.

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the Feature parameter is available in the "*APL documentation" in the section "Configurable functions with the Feature I/O"*. The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	Reserve
1	Reserve
2	1 = At unit allocation: Deactivate the unit simulation
...	Reserve
24	See APL function: "Enable local operator permission" 1 = "Local" operator permission is active
...	...

Operator permissions

This block includes the APL standard function "Operator permissions". The block has the following permissions for the `OS_Perm` parameter:

Status bit	Parameter
0	1= Operator can switch to "Auto" mode
1	1 = Operator can switch to "Manual" mode
2	1 = Operator can activate the unit simulation
3	1 = Operator can deactivate the unit simulation
4	1 = Operator can trigger a RESET command
5	Reserve
...	...

Overview of error numbers

The `ErrorNum` connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.
41	If hold and resume requests are set simultaneously

21.2.3 Connections of 'EqmIf'

Input parameters

Parameter	Description	Type	Default
STEP_NO	Step number from EPH (SFC)	DWORD	
BA_STATE	Batch Status DWORD	DWORD	
ManModOp	Switch equipment modules to manual mode	BOOL	
AutModOp	Switch equipment modules to automatic mode	BOOL	
SimOnOp	Activate simulation	BOOL	
EqmData	Equipment info structure from BlUnitIf	BlUdtEqmData	
RqHold01 ... RqHold16	External HOLD request 1 ... 16	DigValFF	
RqResu01 ... RqResu08	External RESUME request 1 ...8	DigValFF	
RstLi	Reset equipment module (process)	DigVal	
RstOp	Reset equipment module (operator)	BOOL	
Cm01EnHold ... Cm16EnHold	Enable hold on CM01 group error ... Enable hold on CM16 group error	BOOL	True
CmFbk01 ... CmFbk32	CM feedback	BlUdtCmFbk	
OS_Perm	Operator Permissions	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

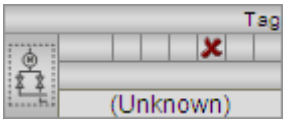
Output parameters

Parameter	Description	Type	Default
AUT	SFC command: Set automatic mode	BOOL	
ENAUT	SFC command: Enable automatic mode	BOOL	
MAN	SFC command: Set manual mode	BOOL	
ENMAN	SFC command: Enable manual mode	BOOL	
GrpAuto	Collected state: 1=all CMs are in automatic mode (or On)	BOOL	
GrpNoError	Collected state: 1=no CM has an error	BOOL	
GrpNoSim	Collected state: 1=no CM is simulating	BOOL	
CmData	Data structure for CMs	BlUdtCmData	
UnitASID	AS and Unit ID	DWORD	
BatchID	Batch ident number	DWORD	
BatchName	Batch name	STRING[32]	
StepNo	Step number from EPH	DWORD	

Parameter	Description	Type	Default
BatchEn	Batch is enabled	BOOL	
BatchOcc	Batch unit is occupied	BOOL	
SimEnCM	Enable control modules' local simulation	BOOL	
SimOnCM	Simulate control modules' feedback	BOOL	
EqmLock	Routed signal from unit, can be used in own interlocks	BOOL	
MsgLock	Lock messages	BOOL	
ManOpEn	Enable operator input manual mode	BOOL	
P_Rst	Reset pulse	BOOL	
ModLiOp	Mode switch selector	DigVal	
AutModLi	Switch to AUTO mode	DigVal	
ManModLi	Switch to MANUAL mode	DigVal	
EqmFbk	This module's feedback for BlUnitIf	BlUdtEqmFbk	
HoldCmd	1 = A HOLD command is active	BOOL	
ResumeCmd	1 = A RESUME command is active	BOOL	
HoldReqExtAct	Internal	WORD	
HoldReqExtPin	Internal	WORD	
HoldReqExtUse	Internal	WORD	
HoldReqCMsAct	Internal	DWORD	
HoldReqCMsPin	Internal	DWORD	
UsedCMs	Internal	DWORD	
OS_PermOut	Operator permission: output 1 for OS	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
OpSt_Out	Enabled operator stations	DWORD	
ST_Worst	Worst signal status	BYTE	16#80
Status1	Status data	DWORD	
ErrorNum	Error number	INT	-1

21.2.4 Operator control and monitoring

21.2.4.1 Symbol



Tag name
Status bar
EQM SFC name
SFC state (S88 state from SIMATIC BATCH)

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	Common operating mode of lower-level elements	
		Automatic
		Manual
		Auto / Manual mixed
2		An external or lower-level hold request is active
3		At least one lower-level element has errors
4		At least one lower-level element is in simulation
5		Unit not released
		Unit released
6		Unit allocated
7		Notice is available

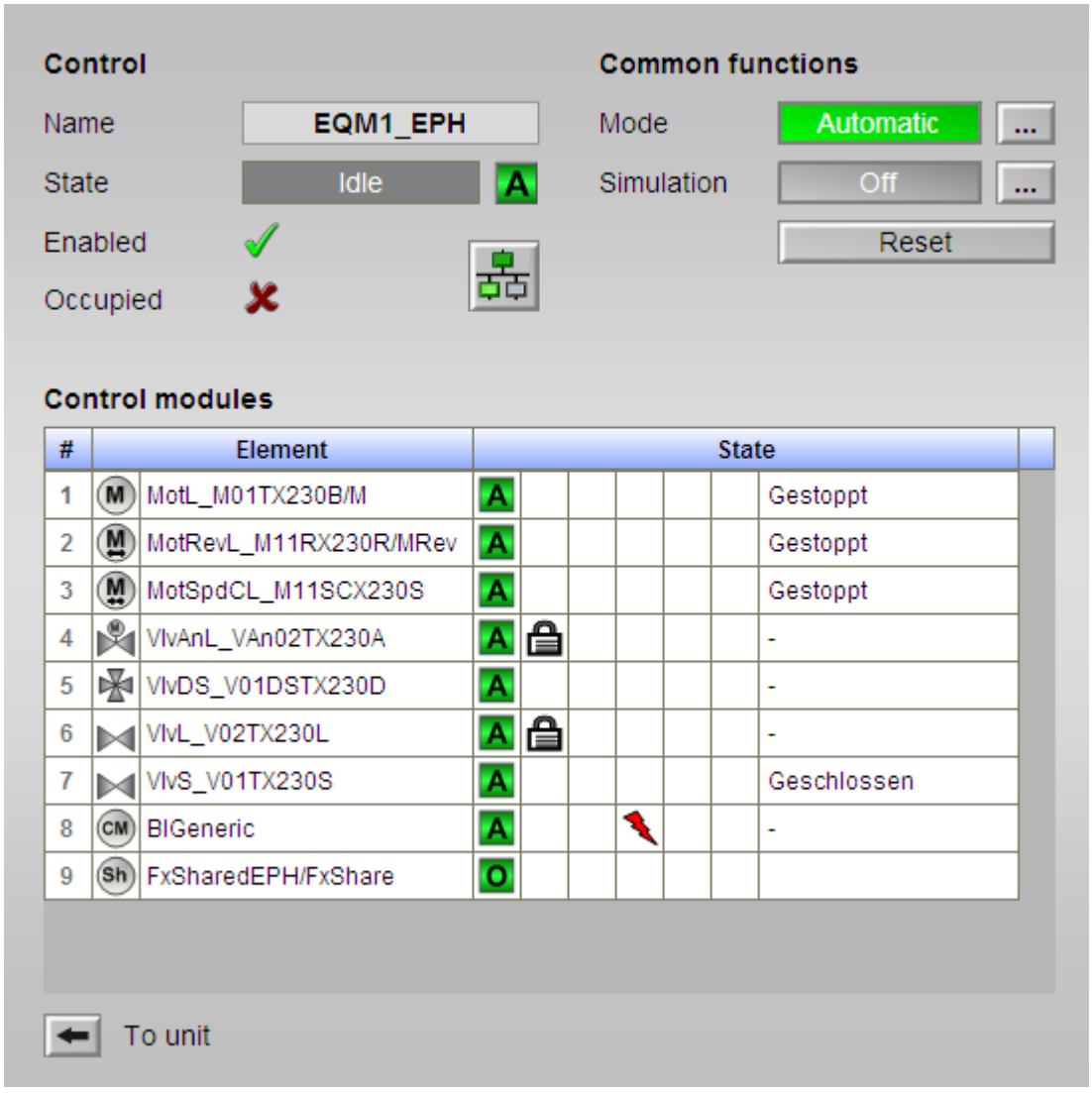
21.2.4.2 Faceplate

Views of the EqmIf block

The block EqmIf provides the following views:

	Standard view	See below.
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library
	Hold diagnostics view	See below.

Standard view of the EqmIf



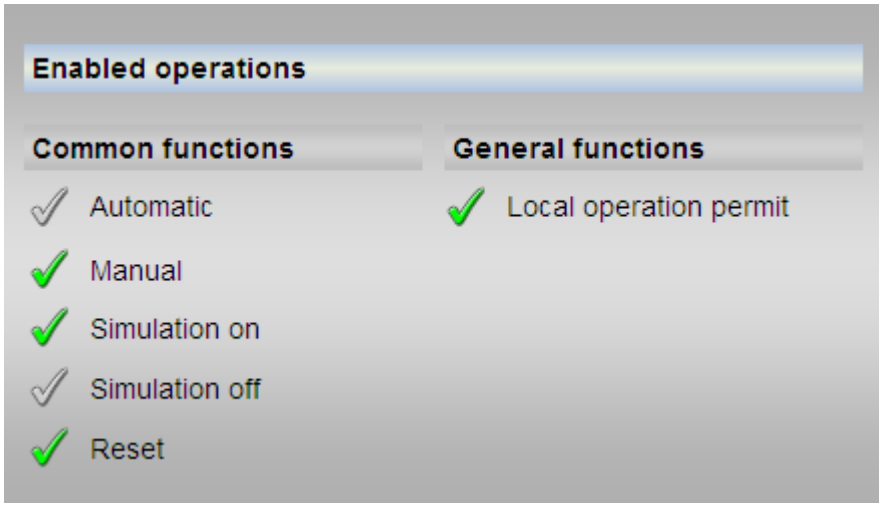
Explanation of the screen shot	
(1)	**Displaying and switching the operating mode**
	This section shows the currently valid operating mode for the EQM as well as the lower-level CMs. The following operating modes can be shown here: • Automatic mode → lower-level CMs are in automatic mode • Manual mode → lower-level CMs are in manual mode • (Mixed) → lower-level CMs have different operating modes The button can be used to switch the EQM incl. the lower-level CMs to the operating mode 'Automatic' or 'Manual operation'.

Explanation of the screen shot	
(2)	**Switching the EQM simulation on and off**
	This section shows the state of the EQM simulation. The following states can be shown and executed here: • On • Off
(3)	**Resetting the modules**
	In case of a corresponding permission this button can be used to send a reset request to all the lower-level modules through an additional operating dialog.
(4)	**Sequential control system**
	This section displays the current state of the EQM SFC as well as whether it is enabled for SIMATIC BATCH and a batch is allocated to it.
	SFC Visualization  This button can be used to call the associated SFC Visualization
(5)	**Control modules**
	This section contains a table with all the connected control modules with their states. • The Element column displays optionally the CFC names of the CM process tags, those of the CM block instances or both together (selection by means of Feature Bits at the respective CM adapter block). • The CM state column contains the following information:

			Manual
			Automatic
			Local
			On
			Out of service
		Interlock	
		Simulation	
		Monitoring error	
		Forced operation (APL basic function "Forcing of operating states")	
		Out of service	
	Clicking a CM line calls the corresponding faceplate.		

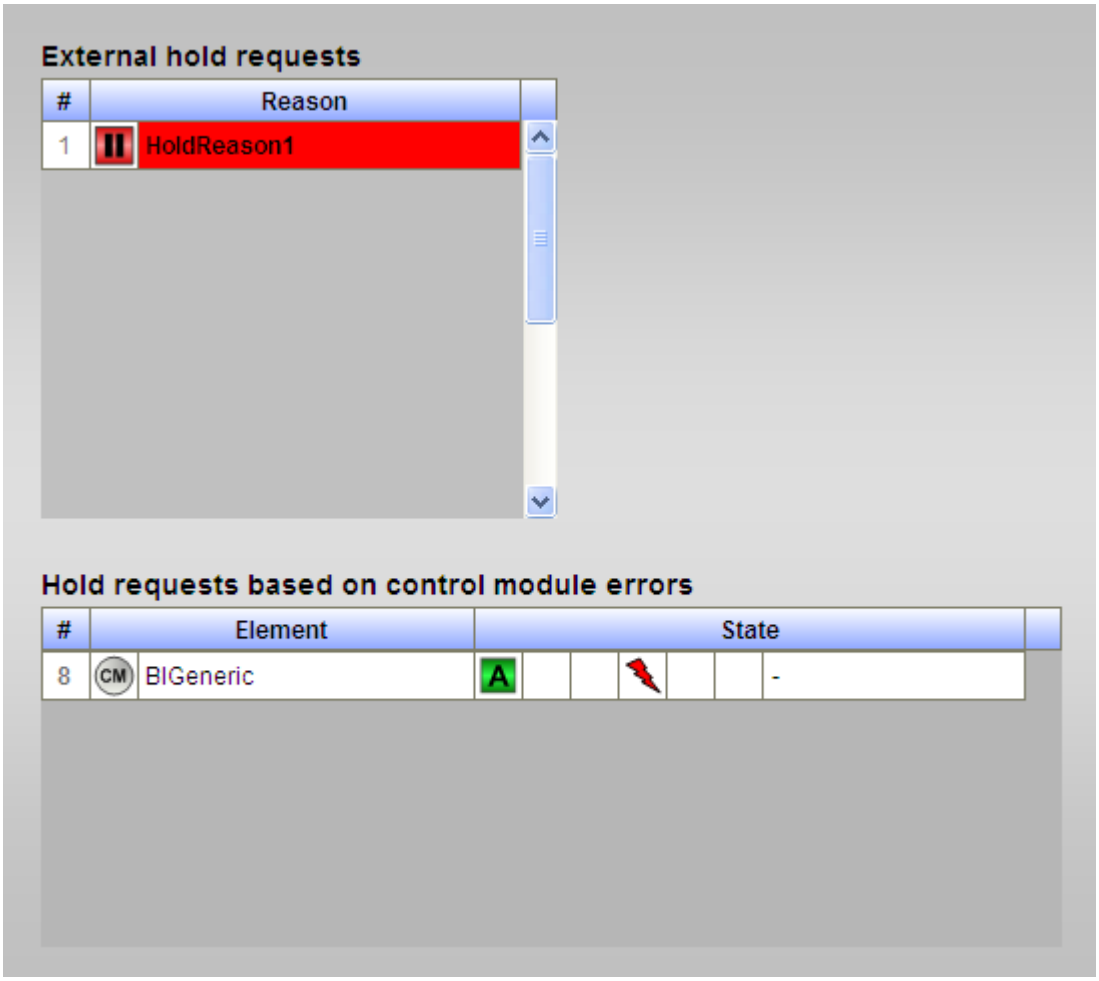
(6)	**Calling the unit faceplate** This button is used to call the higher-level 'BlUnitIf' faceplate.

Preview of BIEqmIf



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the `OS_Perm` parameter of this block. The operator control enables shown in the screenshot are possible here. Meaning of the symbols: • Green check mark: The OS operator can control this parameter • Gray check mark: The OS operator cannot control this parameter at this time due to the process • Red cross: The OS operator cannot control this parameter due to the configured operator permissions

Hold diagnostics view of the EqmIf



Explanation of the screen shot		
(1)	**External hold requests**	
	This view displays the states of the external input parameters `RqHold01 ... RqHold16` as well as the CM-related hold requests. The display window is not filled until the moment from which the UNIT is running and a hold request becomes active. The initial triggers (hold requests active in the same AS cycle) are highlighted in color, signals arriving later are only displayed as long as they are active. The following information is displayed in the 'Causes' column:	
		Hold request active
		Hold request outgoing
		Hold request initial value / active
	Hold cause = 'Text 1" attribute of the external input parameter. Texts of the initial trigger signals are highlighted in color. This marking is retained even when the signal has gone again.	

Explanation of the screen shot	
(2)	**Hold requests based on control module errors**
	Here the hold requests based on error messages from the control modules are displayed. The meaning of the columns and fields corresponds to that of the standard view. The hold history remains until the "Resume" command (also: "Reset"). It is cleared at the latest when the rest position "Idle" has been reached.

21.3 BatchIf – SIMATIC BATCH interface block

21.3.1 Description of 'BatchIf'

Area of application and operating principle

This block is used to generate batches from the PLC level using the WinCC global action script "BL_MainCycle.pas".

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (preferably OB32). The block is also installed automatically in the restart OB (OB 100).

The BatchIf block is used at units that require the functionality of the batch start from the application program. The block is instantiated preferably in the UNIT-specific CFC together with the blocks UNIT_PLC (SIMATIC BATCH) and UnitIf .

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'UnitIf' (Page 1073)".

Status bit	Parameter
0	BatchOcc (=1, when batch is allocated to unit)
1	BatchEn (=1, when unit is enabled for batch allocation)
2	1= Internal operation (faceplate operation is active)
3	`OosAct.Value`
4	`OosLi.Value`
5	`OnAct.Value`
6	1= When `ErrorNum` shows an error
7	`MsgLock.Value`
8	`OvwUnitID`
9	`OvwBatchInfo`
...	Reserve

Status word allocation 'Status2'

The status word serves internal purposes and is not relevant for the user.

21.3.2 Operating modes of 'BatchIf'

The block can be operated using the following operating modes:

- "On"
- "Out of service"

General information on these operating modes is available in the "Operating modes of the blocks" section of the APL documentation.

21.3.3 Functions of 'BatchIf'

The block supports a batch start using interconnectable inputs / outputs in CFCs. A second command input for OS operation (not interconnectable) is available. The global action script installed automatically by the setup recognizes the change in the command output, processes the block parameters and sends a "SIMATIC BATCH" command via API Wrapper / SIMATC Batch API to the Batch Control Server. The feedback of the "Batch API" call is displayed at the "QOS_RetVal" output. In as far as errors occur during command processing corresponding messages are output.

Proper operation has the following prerequisites:

- Installation:
  The component "SIMATIC BATCH Base" (contains Batch API) has to be installed on the PCS 7 OS Server and Client stations!
- ES configuration:
  A Batch client has to be configured for these PCS 7 OS stations in the hardware configuration and be loaded into the stations.
- Runtime:
  PCS 7 OS Runtime and the relevant Batch Server have to run

Operating mode external / internal selection

The operating parameter `IntOp` specifies whether the command / parameter specification for SIMATIC BATCH is carried out by the operator in the faceplate or through the automatic program of the EQM SFC (external block inputs). The standard mode is "External". A switchover to "Internal" by the operator is only valid until the end of the command now following. Afterwards the block switches back to external mode.

- `IntOp = 0` "External operating mode"
  In external mode (standard) the command / parameter specifications are derived from the interconnectable inputs. These are displayed in faceplate, but cannot however be changed.
- `IntOp = 1`"Internal operating mode"
  In internal mode the command / parameter specifications are specified by the operator in the faceplate.

Supported parameters

The following parameters have to be provided for batch generation:

- Name of the PCELL `PCell`
- Name of the order category `OrderCat`
- Name of the master recipe or the formula name `RecipeName`
- Name of the recipe version `RecipeVersion`
- Name of order `OrderName`, if empty: Default order name is automatically generated in accordance with the following scheme:
- <Year>/CW<CW>, e.g. '2009/CW01'
- Name of the batch `BatchNameNew`, if empty: Default batch name is automatically generated in accordance with the following scheme:
- <Year><Month><Day>_<Hour><Minute><Second>_<Recipe name>
  Example: '20090129_124542_BeerPale'
- Batch quantity `BatchSize`

Up to 16 batch parameters can be transferred for the batch with the following input parameters each

Parameter	Function
Pxx_Name	Parameter name
Pxx_ValInt	Parameter value type INT
Pxx_ValDint	Parameter value type DINT
Pxx_ValReal	Parameter value type REAL
Pxx_ValStr	Parameter value type STRING
Pxx_ValSel	Parameter value type selection: 0 := 'not used'; 1 := INT; 2 := DINT; 3 := REAL; 4 := STRING

Material specifications

Up to 2 materials can be transferred for the batch with the following input parameters:

Parameter	Function
Mxx_Name	Material name
Mxx_Amount	Material amount of the type REAL
Mxx_ID	Material ID type DINT
Mxx_Code	Material code type STRING
Mxx_IdSel	Material ID/code selection: • 0 = 'not used' • 1 := MatID/DINT • 2 := MatCode/STRING

Rules for the parameters:

In the OS command processing all the recipe parameters / material definitions are browsed for matching names, beginning with P01_NAME/M01_NAME until the first empty name (empty character string) is discovered. In the case of matching the corresponding parameter value, or at material the amount, is set.

Rules for unit assignment:

- The parameter UnitName1 allows a start unit that is to be used for the batch to be set.
- The OS command processing searches for the corresponding unit names in all the recipe unit procedures (RUPs) of the generated batch.
- Allocation is the carried out for the first RUP that has this unit as a candidate.
- Optionally the parameters `UnitName1/RupName1, UnitName2/RupName2` can be used to carry out direct unit allocation for up to 2 RUPs.
- Alternatively the unit can also be specified using a "normal" parameter of the String type if the allocation strategy of the recipe unit procedure is set to "Process parameters" in the recipe properties.

Command processing

The generation of batches can be controlled through the following SIMATIC BATCH commands:

Values for `CmdLi/ CmdOp`	Command	Description
0	Empty command	No function
1	Create batch	Generates the batch and sets the parameters
2	Create and enable batch	Generates the batch, sets the parameters and enables the batch
3	Create, enable and start batch	Generates the batch, sets the parameters, enables the batch and starts it.
11	Start next batch	Starts the next batch from the list of SIMATIC BATCH batches with the specified order category. Criteria: • Batches that are not enabled are skipped
12	Start next batch with interruption	Starts the next batch from the list of SIMATIC BATCH batches with the specified order category. Criteria: • If the command encounters a batch that is not enabled, the system aborts with an error return value

A new command is recognized by a value change at the interconnectable input `CmdLi` when `CmdBusy = 0`

The user program has to set the input `CmdLi = 0` for at least 1 cycle after a command has been activated - as preparation for the next command

As long as a command is executed in the Batch Control Server, `CmdBusy = 1` and `OS_RetValOut = 0`

After completion of the command in SIMATIC BATCH , i.e. with the negative edge of the output `CmdBusy` the result is displayed at the output `OS_RetValOut` .

`OS_RetValOut > 1` (not OK) → binary output `CmdError = 1`

The following table shows the possible return values of the `OS_RetValOut` output:

Return value	Meaning	Reason
-1	Wait	Order running -> no message
0	Busy	
1	OK	Order accepted by SIMATIC BATCH → no message
2	API is not connected	SIMATIC BATC API not active / connection problems
3	Invalid order category	Parameter `OrderCat` missing or cannot be assigned
4	Unable to create order	
5	Invalid Recipe	Parameter `RecipeName` cannot be assigned (recipe missing or not enabled)
6	Unable to create batch	
7	Invalid parameter list	
8	Invalid parameter	
9	Unable to set parameter(s)	Parameter `Pxx_Name` does not exist in the recipe
10	Invalid unit list	
11	Invalid unit	Parameter `UnitName1/UnitName2` cannot be allocated (unit does not exist)
12	Unable to allocate unit	
13	Unable to release batch	Parameter Pxx_Val<xxx> lies outside the limits
14	Unable to start batch	
15	Unable to abort batch	
16	Unable to hold batch	
17	Unable to pause batch	
18	Unable to resume batch	
19	Invalid batch name	Attempt to start several batches with the same name
20	Invalid order name	
21	Unknown command	
22	Internal error	
23	Incomplete command execution	Network fault or OS redundancy switchover
24	SIMATIC BATCH database blocked	Other application (e.g. BCC) blocks SB database
25	No suitable batch found	No batch with suitable properties (e.g. status) found

Return value	Meaning	Reason
26	Aborted by user script	At return value "ZERO" in user script
27 ...31	Reserve	

Command execution times:

The typical response time for the command execution is made up of:

- Cycle times of the `BatchIf` block / OS Action Script "BL_MainCycle.pas" → < 10 sec
- SIMATIC BATCH command execution time → < 60..100 sec generate order, generate batch, change parameters, enable batch,
  ... For comparison values see "SB-readme.mht", Section 5 – Performance specifications

Overwrite functions:

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwUnitID`
- `OvwBatchInfo`

You will find more information on this subject in the section "Higher-level controlling of the unit / EQM level (Page 70)".

Call additional faceplates

This block includes the APL standard function "Call additional faceplates".

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in the "APL documentation" in the section "Configurable functions with the Feature I/O". The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	See APL function: "Setting the startup response" 1 = Startup without call in OB100 0 = Startup with call in OB100 (default): During the CPU start-up the messages are suppressed for a number of cycles configured in `RunUpCyc` (default is 3). During the CPU start-up the outputs "`QCmd`" and "`QOS_RetVal`" are initialized.
1	See APL function: "Behavior for 'Out of service' mode" Release for switchover to "Out of service" via `OosLi` (default = 0)
2	1 = In the case of a successful batch start a message is generated (default = 0)

Bit	Comment
3	1 = Switch automatically to externally (linked) operation after completion of a operator command (default = 1)
...	Reserve
22	See APL function: "Update acknowledgment and error status of the message call"
...	Reserve
24	See APL function: "Enable local operator permission" 1 = "Local" operator permission is active
...	Reserve

Operator permissions

This block includes the APL standard function "Operator permissions". The block has the following permissions for the `OS_Perm` parameter:

Status bit	Parameter
0	1= Operator can switch to "On" mode
1	1 = Operator can switch to "Out of service" mode
2	1 = Operator can enter commands
3	1 = Operator can switch between external / internal operating mode
4	1 = Operator can change the attribute "Process Cell"
5	1 = Operator can change the attribute "Order Category"
6	1 = Operator can change the attribute "Recipe Name"
7	1 = Operator can change the attribute "Recipe Version"
8	1 = Operator can change the attribute "Order Name"
9	1 = Operator can change the attribute "Batch Name"
10	1 = Operator can change the attribute "Batch Size"
11	1 = Operator can change the attribute "Unit name 1"
12	2 = Operator can change the attribute "Unit name 1"
13	1 = Operator can change the attribute "RUP name 1"
14	2 = Operator can change the attribute "RUP name 1"
15	1 = Operator can change the attribute "Material names"
16	1 = Operator can change the attribute "Material amount"
17	1 = Operator can change the attribute "Material code"
18	1 = Operator can change the attribute "Parameter names"
19	1 = Operator can change the attribute "Parameter values"
...	Reserve

Overview of error numbers

The `ErrorNum` connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.

Error number	Meaning of the error number
30	Invalid parameter data type `Pxx_ValSel <> 0…4`
31	Invalid material data type `Mxx_IdSel <> 0…2`

21.3.4 Messaging of 'BatchIf'

The block uses an instance of the ALARM_8P block (MsgEvId). The following messages can be generated:

Message instance	Message identifier	Message class	Event
`MsgEvId`	SIG 1	AS process control message - fault	$$BlockComment$$ @4W%t#IL-BatchIfErrors@
	SIG 2	AS process control message - fault	$$BlockComment$$ @5W%t#IL-BatchIfErrors@
	SIG 3	Status message OS	$$BlockComment$$ Batch was created successfully
	SIG 4	AS process control message - fault	$$BlockComment$$ External error has occurred
	SIG 5	Alarm high	
	SIG 6	Alarm high	
	SIG 7	Alarm high	
	SIG 8	Alarm high	

Explanation

- $$BlockComment$$
  Content of the instance-specific comment
- @...ILBatchIfErrors@
  Generates the error text from the text library 'ILBatchIfErrors'. The block generates an information message under the following conditions:
  - `OS_RetValOut > 1` (i.e. when the OS-Action-Script returns the value NOT OK)
  - `ErrorNum > 0`
- You can interconnect an external fault to the input parameter CSF. If this signal changes to `CSF = 1`, a control system error is triggered (`MsgEvId`, SIG 4).
- In addition external messages SIG 5 ... SIG 8 can be generated through the input parameter `ExtMsg1 … ExtMsg4` .
- Message generation is prevented with the input signal `MsgLock` or `CmData.xMsgLock = 1` .

Associated values for message instance `MsgEvId`

Associated value	Block parameter
1	BatchName
2	StepNo
3	BatchID
4	siMsgRetVal
5	siMsgErrorNum
6	ExtVa106
7	ExtVa107
8	UnitIDOut

21.3.5 Connections of 'BatchIf'

Input parameters

Parameter	Description	Type	Default
CmdLi	Batch command (linkable)	INT	
CmdOp	Batch command (operator)	INT	
PCell	Process cell	STRING[32]	
OrderCat	Order category	STRING[32]	
RecipeName	Recipe name	STRING[32]	
RecipeVersion	Recipe version	STRING[5]	
OrderName	Order name	STRING[32]	
BatchNameNew	Batch name	STRING[32]	
BatchSize	Batch size	STRING[10]	
UnitName1	Unit name 1	STRING[24]	
UnitName2	Unit name 2	STRING[24]	
RupName1	RUP name 1	STRING[24]	
RupName2	RUP name 2	STRING[24]	
M01_Name	Material name	STRING	
M01_Amount	Material amount	REAL	
M01_ID	Material ID	DINT	
M01_Code	Material code	STRING	
M01_IdSel	Material identifier selector	INT	
M02_Name	Material name	STRING	
M02_Amount	Material amount	REAL	
M02_ID	Material ID	DINT	
M02_Code	Material code	STRING	
M02_IdSel	Material identifier selector	INT	

Parameter	Description	Type	Default
P01_Name	Parameter name	STRING	
P01_ValInt	Parameter value as an INT	INT	
P01_ValDint	Parameter value as a DINT	DINT	
P01_ValReal	Parameter value as a REAL	REAL	
P01_ValStr	Parameter value as a STRING	STRING	
P01_ValSel	Parameter value type selector	INT	
...			
P16_Name	Parameter name	STRING	
P16_ValInt	Parameter value as an INT	INT	
P16_ValDint	Parameter value as a DINT	DINT	
P16_ValReal	Parameter value as a REAL	REAL	
P16_ValStr	Parameter value as a STRING	STRING	
P16_ValSel	Parameter value type selector	INT	
OS_RetVal	OS command return value	INT	
OS_BatchName	Batch name from OS	STRING[32]	
CmData	Control and status information for this CM	BlUdtCmData	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
OvwUnitID	1=Overwrite CmData unit ID with input UnitID	BOOL	
UnitID	Unit ID (usually from BlUnitIf block)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIdLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
RunUpCyc	Default startup cycles during witch messages are disabled	INT	3
SampleTime	Sample time	REAL	1.0
MsgLock	Inhibit process messages	DigVal	
UserStatus	User status bits	BYTE	
CSF	Control System Fault message - external error	DigVal	
MsgEvId	Message event ID 01	DWORD	16#FFFFFFFF
ExtMsg1 ... 4	External message signal 1 ... 4	DigVal	
ExtVa106	External value 6	ANY	
ExtVa107	External value 7	ANY	
OnOp	1=On mode: On mode by operator	BOOL	
OosOp	1=Oos mode: Oos mode by operator	BOOL	
IntOp	1=Internal mode for overall parameter manipulation	BOOL	
OosLi	1=Oos mode: Oos mode switch via link	DigVal	
SelFp1	Open faceplate of connected block (type)	BOOL	
SelFp2	Open faceplate of connected block (type)	BOOL	
OS_Perm	Operator permissions	STRUCT	

Parameter	Description	Type	Default
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

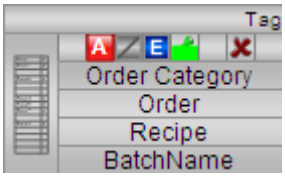
Output parameters

Parameter	Description	Type	Default
OnAct	On mode is active	DigVal	
OosAct	Out of service is active	DigVal	
CmdOut	Command	INT	
CmdBusy	Batch is busy	BOOL	False
CmdError	OS signaled an error	BOOL	False
OS_RetValOut	Internal	INT	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING[32]	
StepNo	Batch step number	DWORD	
UnitIDOut	Unit ID	DWORD	
PCellOut	Process cell	STRING[32]	
OrderCatOut	Order category	STRING[32]	
RecipeNameOut	Recipe name	STRING[32]	
RecipeVersionOut	Recipe version	STRING[5]	
OrderNameOut	Order name	STRING[32]	
BatchNameOut	Batch name	STRING[32]	
BatchSizeOut	Batch size	STRING[10]	
UnitName1Out	Unit name 1	STRING[32]	
UnitName2Out	Unit name 2	STRING[32]	
RupName1Out	RUP name 1	STRING[32]	
RupName2Out	RUP name 2	STRING[32]	
M01_NameOut	Material name	STRING	
M02_NameOut	Material name	STRING	
M01_AmountOut	Material amount as a STRING	STRING	
M02_AmountOut	Material amount as a STRING	STRING	
M01_CodeOut	Material code as a STRING	STRING	
M02_CodeOut	Material code as a STRING	STRING	
P01_NameOut ... P16_NameOut	Parameter name	STRING	
P01_ValStrOut ... P16_ValStrOut	Parameter value as a STRING	STRING	
ErrorNum	Error number	INT	

Parameter	Description	Type	Default
ErrAddInfo	Additional error information	INT	
MsgErr	ALARM_8P: Messaging error	BOOL	
MsgStat	ALARM_8P: Status output	WORD	
MsgAckn	ALARM_8P: ACK_STATE output	WORD	
OS_PermOut	Operator permissions (OS info)	DWORD	16#FFFFFFFF
OS_PermLog	Operator permissions (OS info)	DWORD	16#FFFFFFFF
OpSt_Out	Enabled operator stations	DWORD	
Status1	Status1 data	DWORD	
Status2	Status2 data	DWORD	
ST_Worst	Worst signal status	BYTE	16#80

21.3.6 Operator control and monitoring

21.3.6.1 Symbol

	Tag name Status bar Order category Order name Recipe name Batch name

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	Empty	
2	Message status *)	
3	Operating mode	
		On
		Out of service
4	Mode	
		External
		Internal
5		Enable maintenance operation
6	Empty	
7		Unit not released
		Unit released
8		Memo is available

*) Additional details about the "Message status" are available in the "Block icon structure" section of the APL documentation.

21.3.6.2 Faceplate

Views of the BatchIf block

Note

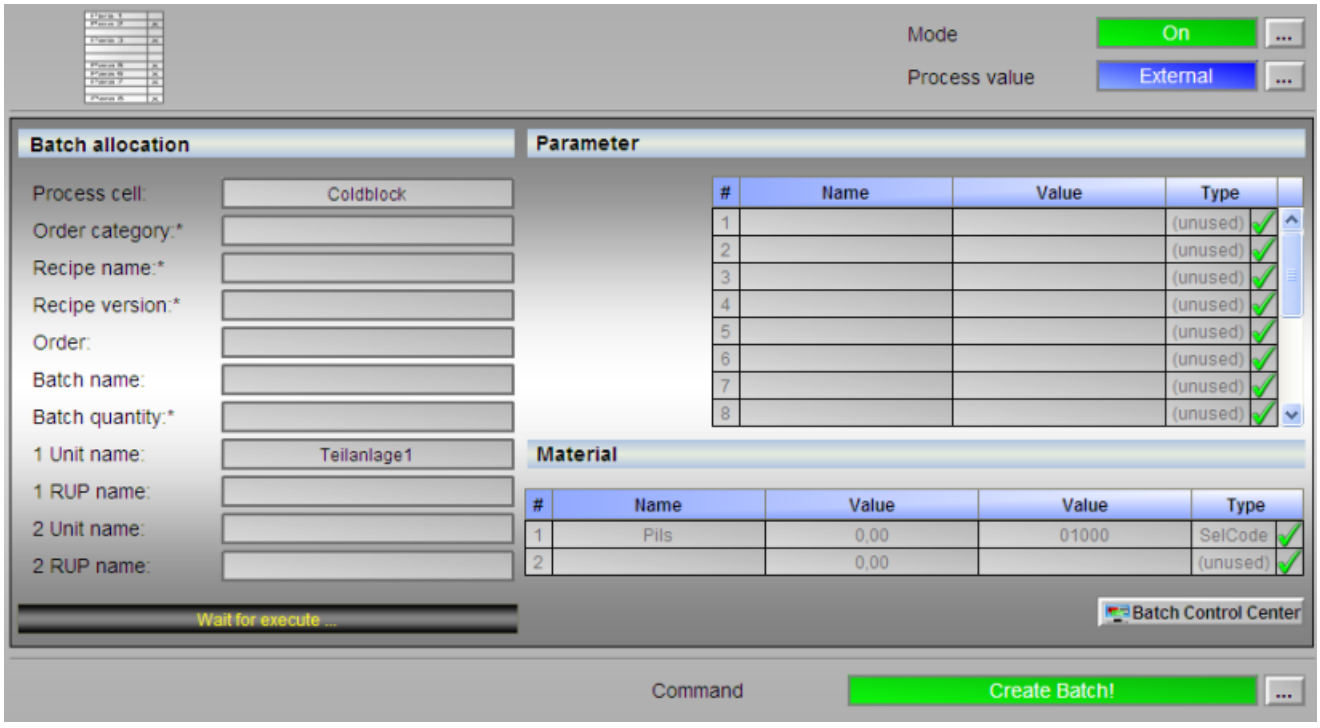
Purpose and field of application

The faceplate is used primarily to visualize the interconnected parameters (diagnostics function for command state, error feedbacks, parameters, etc.). If the corresponding operator permission is available ("higher-order operator process controls"), the block can be switched to internal operation for test and diagnostics purposes. This makes all the fields for batch data and parameter / material data editable and a SIMATIC BATCH command can be selected and issued.

The block BatchIf provides the following views:

	Standard view	See below.
	Alarm view	Standard message view of all the blocks of the Advanced Process Library
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

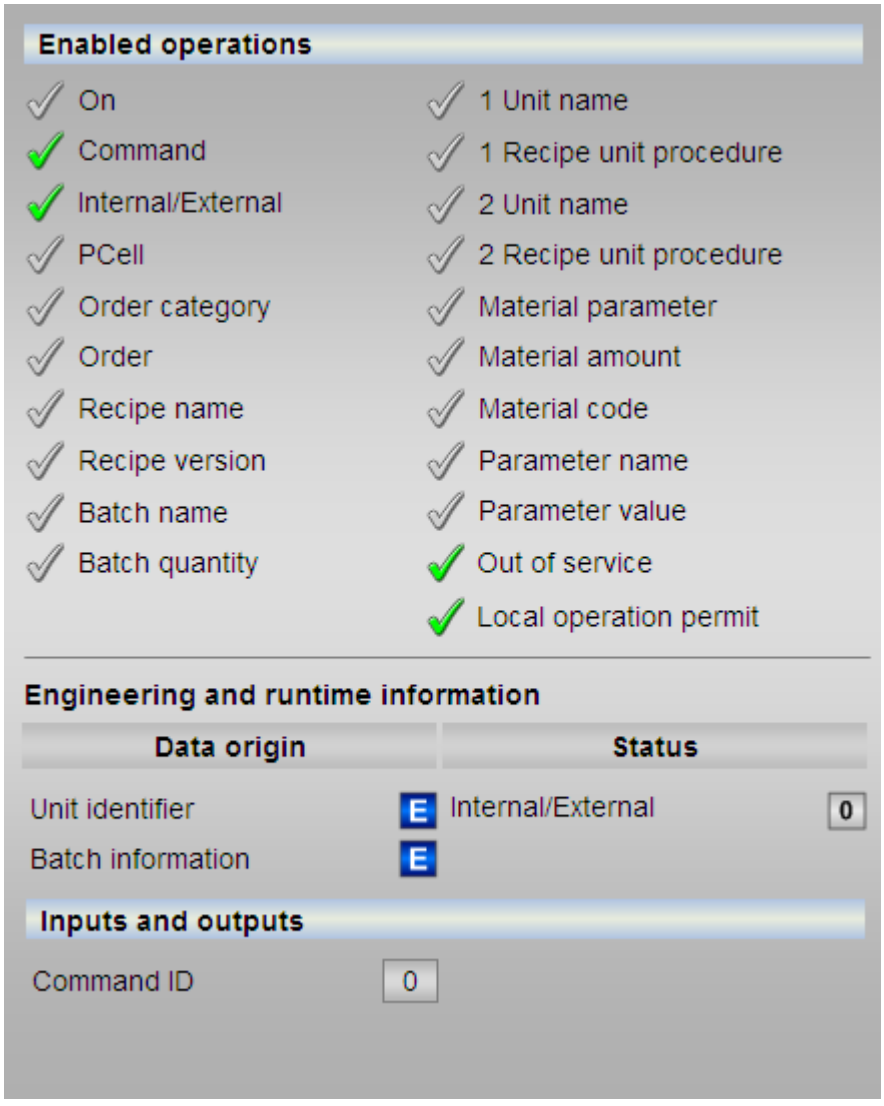
Standard view of the block BatchIf



Explanation of the screen shot	
(1)	**Displaying and switching the operating mode**
	This area provides information on the currently valid operating mode. The following operating modes can be shown and executed here: • On • Out of service
(2)	**Displaying and switching the process parameters**
	• External → Command / parameter specification is made via interconnection. • Internal → Command / parameter specification is made directly in the faceplate.
(3)	**Displaying and changing the recipe parameters**
	The current recipe parameters are displayed here. These can also be changed in "Internal" operation.
(4)	**Displaying and changing the material parameters**
	The current material parameters are displayed here. These can also be changed in "Internal" operation.
(5)	**Command buttons**
	"Batch Control Center" → Opens the SIMATIC BATCH Control Center → Only available at SIMATIC BATCH-Client stations
(6)	**Displaying and changing the batch data**
	The current batch parameters are displayed here. These can also be changed in "Internal" operation.
(7)	**Command feedback status**
	The state of the input parameter `OS_RetVal` (status of the SIMATIC BATCH command) is displayed here. The possible feedback texts are listed in the table "Return values of the parameter `OS_RetValOut` in the section Functions of 'BatchIf' (Page 1097).

Explanation of the screen shot	
(8)	**Navigation button to the standard view of any faceplate**
	This navigation button is used to reach the standard view of a block connected at the block input `SelFpl` in the CFC.
(9)	**Command (SIMATIC BATCH command)**
	Here a command can be sent to SIMATIC BATCH in internal operation if corresponding permission is available. The possible commands are listed in the table "Command processing through the parameter `CmdLi/CmdOp`" in the section Functions of 'BatchIf' (Page 1097).

Preview of BatchIf



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the `OS_Perm` parameter of this block. The operator control enables shown in the screenshot are possible here. • Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions
(2)	**Engineering and runtime information**
	This section contains information from the status word `Status1`
	• The "Data origin" column shows the setting of the overwrite signals. – Unit identifier `OvwUnitID` – Batch information `OvwBatchInfo`
	The symbols have the following meaning
	 Signal origin = Unit / `CmData` structure
	 Signal origin = Block input `<xxx>`
	• 'States' column displays the state of the input parameter IntOp (0 / 1) : – `IntOp = 1` Command processing is carried out by the operator (internal). – `IntOp = 0` Command processing is carried out via interconnection (external).
(3)	**Inputs and outputs**
	The value of the output parameter `CmdOut` (active SIMATIC BATCH command) is displayed here.
(4)	**Navigation button to the standard view of any faceplate**
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp2` in the CFC.

Technological blocks

22

22.1 Qing - Queue Transfer control block

Area of application and operating principle

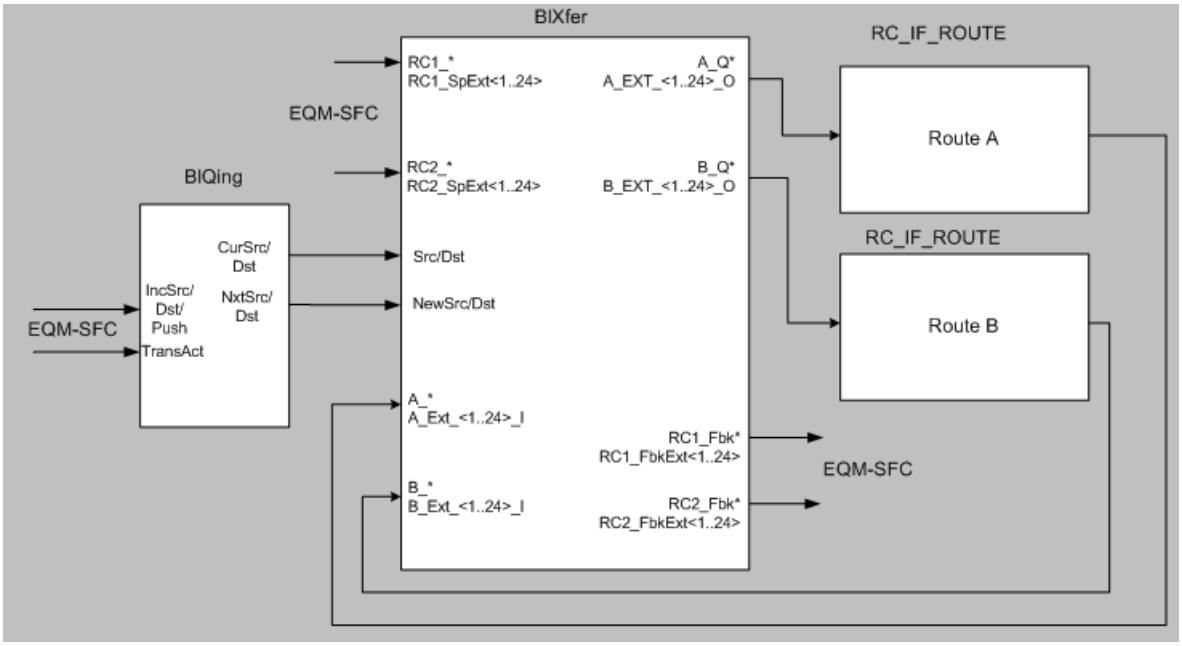
This function block manages a list of material transfers. The list positions can be specified through interconnection ('External' operating mode) or can be selected by the operator by selecting from a list of the tanks overall available (filtered) ('Internal' operating mode). Execution of the transfer list is carried preferably by using the Xfer block in combination with SIMATIC Route Control. The block supports the following transfer constellations, whereby the selection is specified when configuring the block instance:

1 source → 8 destinations	8 sources → 1 destination
	
Parameter `LocXType = ,Source'`	Parameter `LocXType = ,Destination'`

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). In addition a block 'Xfer' for "bumpless switchover" between master and standby route as well as two RCS interface blocks for these two routes are required.

Interconnection principle



Note on the installation sequence

Note that the specified blocks of this sequence including EQM-SFC have to run in the same OB to ensure that the modes and controls requested by the EQM SFC are not lost before the RCS interface block is reached. The installation sequence is:

→ EQM-SFC → Qing → Xfer → RC_IF_ROUTE (A) → RC_IF_ROUTE (B)

The required CFC connections between the Qing / Xfer block as well as to the RCS–IF blocks are manifold and are specified in the supplied CFC template.

In order to display the transfer lists as well as to select the filter criteria, the faceplate of the block requires the names for specific parameters of the assigned tanks that are represented by instances of the Tank block. To ensure correct name resolution the user therefore has to configure the enumerations named below and has to carry out parameter assignment at the block parameters of both blocks Qing and Tank.

Description / Purpose	Enumeration (preset)	Block parameter (Qing)
Definition tank name and assignment to the Location ID	BlTankName	`LocXNames, LocNames`
Definition of the tank types	BlTankType	`TankType`
Definition of the tank groups	BlTankGroup	`TankGrp`
Definition of the tank states	BlTankStatus	`TankStatus`
Definition of the material classes	BlMatClass	`MatClass`

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'Qing' (Page 1121)".

Status bit	Parameter
0	BatchOcc
1	BatchEn
2	xLocLiOp
3	OosAct.value
4	OosLi.value
5	OnAct.value
6	Xfer Status1.Bit06 SwitchedOut=0/1 Route A/Route B is Master
7	Xfer Status1.Bit07 • 1=Route A Automatic • 0=Manual mode
8	Xfer Status1.Bit08 1=Route A request
9	Xfer Status1.Bit09 1=Route A running
10	Xfer Status1.Bit10 1=Route A request/stopped
11	BlXfer Status1.Bit11 1=Route A error
12	Xfer Status1.Bit09 1=Route A error /=1 Ignore error
13	Reserve
14	BlXfer Status1.Bit14 • 1=Route B Automatic • 0=Manual mode
15	Xfer Status1.Bit15 1=Route B request
16	Xfer Status1.Bit16 1=Route B running
17	Xfer Status1.Bit17 1=Route B request/stopped
18	Xfer Status1.Bit18 1=Route B error
19	Xfer Status1.Bit19 1=Route B error /=1 Ignore error
20	TransAct.Value
21	PushOutAct.value
22	LocXType
23	OvwUnitID
24	OvwBatchInfo

Status bit	Parameter
25	`Xfer Status1.Bit03` =1 Out of service
26	`Xfer Status1.Bit03` =1 On mode
27	`TransIndex`
28	`PushOutIndex`
...	Reserve

22.1.1 Operating modes of 'Qing'

The block can be operated using the following operating modes:

- "On"
- "Out of service"

General information on these operating modes is available in the "Operating modes of the blocks" section of the APL documentation.

22.1.2 Functions of 'Qing'

Operating mode external / internal selection

The input parameter `LocLiOp`specifies whether the selection of transfer sources and destinations is carried out by the operator in the faceplate or through the automatic program of the EQM-SFC (external block inputs).

`LocLiOp = 1` "External operating mode"

In external mode the transfer parameters (Location IDs, setpoints / quantities) are derived from interconnectable inputs. These are displayed in faceplate, but cannot however be changed. Enabling for executing the transfer / output list as well as the transition of the unit points takes place via interconnectable input signals.

`LocLiOp = 0` "Internal operating mode"

In internal mode the parameters are specified by the operator in the faceplate. Enabling for execution of the transfer / output list also takes place in the faceplate.

The following table describes the most important block functions as well as the parameters valid for both active operating modes:

Block function	'Internal' operating mode Operator (faceplate)	'External' operating mode input parameter (interconnected)
Definition of the transfer unit points	`LocIdXOp, LocId<n>Op`	`LocIdXLi, LocId<n>Li`
Definition of the transfer setpoints (quantity)	`LocSpXOp, LocSP<n>Op`	`LocSpXLi, LocSp<n>Li`

Block function	'Internal' operating mode Operator (faceplate)	'External' operating mode input parameter (interconnected)
Transfer list • Switch on / off • Next unit point	`TransOnOp, CmdOffOp` `IncTransOp`	`TransOnLi, CmdOffLi` `IncTransLi`
Output list • Switch on • Next unit point	`PushOnOp,` `IncPushOp`	`PushOnLi` `IncPushLi`
Step enable to the next list position	`ReqSwitchOp`	`ReqSwitchLi`
Reset, On, Out of service	`RstOp, OnOp,` `OosOp`	`RstLi,` `OosLi`

Explanation

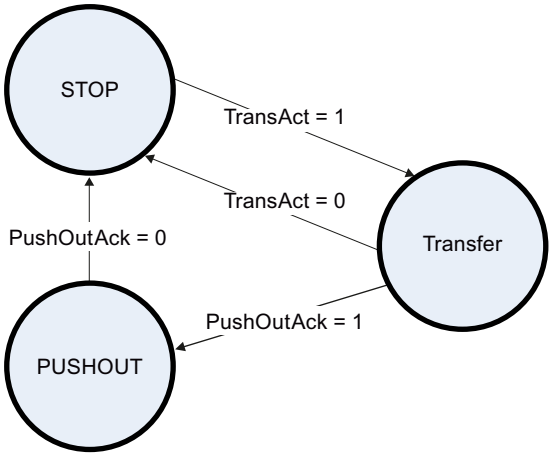
The abbreviation `<n>` stands for a unit point or a list position with n = 1...8

Operating states 'Idle state', 'Transfer operation' and 'Output operation':

The block is in one of these operating states. The output list contains the 'history' of the complete material transfers since the last transition from idle state to transfer operation. The following actions take place in the states:

- STOP
  No transfer routes are requested
- TRANSFER
  The transfer list is executed
- PUSHOUT
  The output list is executed

The following diagram describes the states and the respective block functions as well as the state changes:



Explanation of the screen shot	
STOP	• Input of locations + setpoints (quantity) possible without restrictions • * RstOp 0 -> 1 deletes the transfer list in 'Internal * RstLi 0 -> 1 Deletes the transfer list in 'External' • 'Output list' is deleted, 'Transfer list' stands at 1. Location • No evaluation of the RCS feedbacks in the faceplate (white marking)
TRANSFER	• For 'Internal' operation entry of locations possible except if a location is active • Input of the quantity is always possible • For 'External' the interconnected parameters are frozen • Evaluation of the RCS feedback messages in the faceplate (marking green / yellow) for transfer list
PUSHOUT	• Input of locations + quantity not possible • Evaluation of the RCS feedback messages in the faceplate (marking green / yellow) for output list • If `IncPushOp/IncPushLi 0 -> 1` • the output list is increased

Additional block functions:

- In the TRANSFER or PUSHOUT operating mode the current source / destination parameters are output per list position and the subsequent list position is output at the `NxtSource/ NxtDest` at the outputs `CurSource/CurDest` These are preferably interconnected with the transfer block '`Xfer`'. The end position in the list is recognized by the entry '`NoLocation`' in the source / destination. When the last position is reached, the binary output `EndTrans` or `EndPushOut` is set to TRUE.
- The input `ChkMode` can be used to check the tank selection while entering the source and target transfer lists. If the check produces an error, a yellow flashing bar with the text "Invalid location order" is displayed at the faceplate. `ChkMode` can be set to the following values:

1	Location IDs have to be increasing
2	Location IDs have to be decreasing
0	No check active

- The output list can be sorted as follows by using the input `SortMode`:

1	Increasing sorting of the Location IDs
2	Increasing sorting of the Location IDs
0	No sorting.
Sorting is carried out when the output list is increased or when the sorting instruction has been changed.	

- The status of the two routes is transferred through the CFC connection of the input `Status1Xfer` with the block `Xfer` . If a route request is invalid, the current or next unit point respectively is marked in the OS faceplate.

- The setpoints of the current material transfer are active at the outputs `ActSpSrc/ ActSpDst`. These are marked in the faceplate in green The outputs `PvSpSrc/PvSpDst` always contain the last setpoint after a change in the source or the destination.
- The output `NxtLocValid` (type BYTE with enumeration "BlValidData") shows the result of a check of the next unit point:
  - Result = "NoLocation".
    No additional unit point selected
  - Result = "EqualLocation"
    Next unit point = Predecessor
  - Result = "NewLocation"
    Next unit point <> Predecessor

Overwrite functions

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwUnitID`
- `OvwBatchInfo`

You can find more information on this subject in the section Higher-level controlling of the unit / EQM level (Page 70)

Calling additional faceplates

This block includes the APL standard function "Call additional faceplates".

Forming the signal status for blocks

This block includes the APL standard function "Form and output signal status for blocks".

The worst signal status for the block is formed from the following parameters:

- `LocSp1Li`
- `LocSp2Li`
- `LocSp3Li`
- `LocSp4Li`
- `LocSp5Li`
- `LocSp6Li`
- `LocSp7Li`
- `LocSp8Li`

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in the APL documentation in the section "Configurable functions with the Feature I/O". The following functionality is available for this block at the relevant bits:

- **Feature bits**

Bit	Comment
0	Enable switchover to "Out of service" via `OosLi` (default = 0)
...	Reserve
24	1 = "Local" operator permission is active
...	Reserve

- **Operator authorizations**
  This block includes the APL standard function "Operator permissions". The block has the following authorizations for the `OS_Perm` parameter:

Status bit	Parameter
0	1= Operator can switch to "On" mode
1	1= Operator can switch to "Out of service" mode
2	1= Operator can start the transfer
3	1= Operator can start the output transfer
4	1= Operator can terminate the transfer / output transfer
5	1= Operator can step enable the transfer index
6	1= Operator can step enable the output index
7	1= Operator can reset the list
8	Reserve
9	Reserve
10	Reserve
11	Reserve
12	1= Operator can change the static unit point
13	1= Operator can change the setpoint for the static unit point
14	1= Operator can change the unit point 1
15	1= Operator can change the setpoint for the unit point 1
16	1= Operator can change the unit point 2
17	1= Operator can change the setpoint for the unit point 2
18	1= Operator can change the unit point 3
19	1= Operator can change the setpoint for the unit point 3
20	1= Operator can change the unit point 4
21	1= Operator can change the setpoint for the unit point 4
22	1= Operator can change the unit point 5
23	1= Operator can change the setpoint for the unit point 5
24	1= Operator can change the unit point 6
25	1= Operator can change the setpoint for the unit point 6
26	1= Operator can change the unit point 7

Status bit	Parameter
27	1= Operator can change the setpoint for the unit point 7
28	1= Operator can change the unit point 8
29	1= Operator can change the setpoint for the unit point 8
30	1= Operator can change the filter settings
31	Reserve

22.1.3 Error correction for 'Qing'

For information about troubleshooting all the blocks please also refer to the basic information in the APL section "Troubleshooting". The following error messages can be displayed for this block:

- **Overview of error numbers**
  The `ErrorNum` connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.
51	When start and stop signals are set simultaneously for the transfer or pushout list
56	Unit point for source has the incorrect type
57	Unit point for destination has the incorrect type
58	Unit point not defined
59	If sequence of the unit points does not correspond to the specified sorting sequence (transfer or output list)
60	If a setpoint < 0.0

22.1.4 Connections of 'Qing'

Input parameters

Parameter	Description	Type	Default
`LocLiOp`	1=Extern, 0=Intern: Input to extern/intern locations and setpoints	`DigVal`	
`TransOnLi`	1=Transfer: Start transfer command in link mode	`DigVal`	
`TransOnOp`	1=Transfer: Start transfer command in operator mode	`BOOL`	`False`
`PushOnLi`	1=PushOut: Start pushout command in link mode	`DigVal`	
`PushOnOp`	1=PushOut: Start pushout command in operator mode	`BOOL`	`False`
`CmdOffLi`	1=Stop: Stop command in link mode	`DigVal`	
`CmdOffOp`	1=Stop: Stop command in operator mode	`BOOL`	`True`

Parameter	Description	Type	Default
IncTransLi	Linkable increment signal location list	DigValFF	
IncTransOp	Operator increment signal location list	BOOL	
IncPushLi	Linkable increment signal push out list	DigValFF	
IncPushOp	Operator increment signal push out list	BOOL	
RstLi	Linked reset signal	DigValFF	
RstOp	Operator reset signal	BOOL	
OnOp	1=On mode: On mode by operator	BOOL	
OosLi	1=Oos mode: Oos mode switch via link	DigVal	
OosOp	1=Oos mode: Oos mode by operator	BOOL	
ReqSwitchLi	Linkable Request switch route	DigValFF	
ReqSwitchOp	Operator Request switch route	BOOL	
LocXType	Fix Location Type	BOOL	
LocXNames	Enumeration of fix Location ID	DINT	0
LocSpXUnit	Engineering units of fix Location SP	INT	0
LocSpXHiLim	Fix Location SP High Limit	REAL	5000.0
LocSpXLoLim	Fix Location SP Low Limit	REAL	0.0
LocIdXOp	Operator fix Location ID	DINT	0
LocSpXOp	Operator fix Location SP	REAL	0.0
LocIdXLi	Linkable fix Location ID	DINT	0
LocSpXLi	Linkable fix Location SP	AnaValFF	
LocCnt	Number of locations	INT	8
LocNames	Enumeration of operator Location ID's	DINT	0
LocSpUnit	Engineering units of Location SP	INT	0
LocSpHiLim	Location SP High Limit	REAL	5000.0
LocSpLoLim	Location SP Low Limit	REAL	0.0
LocId1Op ... LocId8Op	Operator Location ID 1 ... 8	DINT	0
LocSp1Op ... LocSp8Op	Operator Location SP 1 ... 8	REAL	0.0
LocId1Li ... LocId8Li	Linkable Location ID 1 ... 8	DINT	0
LocSp1Li ... LocSp8Li	Linkable Location SP 1 ... 8	AnaValFF	
ActPvSrc	Actual process value for source	AnaValFF	
ActPvDst	Actual process value for destination	AnaValFF	
TankType	Enumeration for the tank type	INT	
Filter1Def	Filter definitions for the tank type	STRING[128]	
TankGrp	Enumeration for the tank group	INT	
Filter2Def	Filter definitions for the tank group	STRING[128]	

Parameter	Description	Type	Default
TankStatus	Enumeration for the tank status	INT	
Filter3Def	Filter definitions for the tank status	STRING[128]	
Filter4Def	Filter definitions for the material class	STRING[128]	
Filter5Def	Filter definitions for the material	STRING[128]	
DynamicDef	Dynamic filter definition	STRING[128]	
CmData	Control and status information for this CM	BlUdtCmData	
OvwUnitID	1=Overwrite CmData.UnitID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
UnitID	Unit ID (usually from BlUnitIf block)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIdLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
FuncIdXfer	Connection to Xfer (Transport ID 301...)	INT	0
A_RouteIDXfer	Connection to Xfer (Number of actually used route 1..300)	INT	0
A_ServIPXfer	Connection to Xfer (IP-Adress of actual RC-Server)	DWORD	0
A_MaterialXfer	Connection to Xfer (Actual material used in Route A)	DINT	0
B_RouteIDXfer	Connection to Xfer (Number of actually used route 1..300)	INT	0
B_ServIPXfer	Connection to Xfer (IP-Adress of actual RC-Server)	DWORD	0
B_MaterialXfer	Connection to Xfer (Actual material used in Route B)	DINT	0
Status1Xfer	Connection to Xfer (Info from Route A/B)	DWORD	0
SortMode	Sorting mode of PUSH OUT list. 0 - no sort, 1 - sort upwards, 2 - sort downwards	INT	0
CheckMode	Check condition of source/destination list. 0 - no check, 1 - upward check, 2 - downward check	INT	0
SelFp1	Open faceplate of connected block (type)	BOOL	
SelFp2	Open faceplate of connected block (type)	BOOL	
OS_Perm	Operator permissions	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

Output parameters

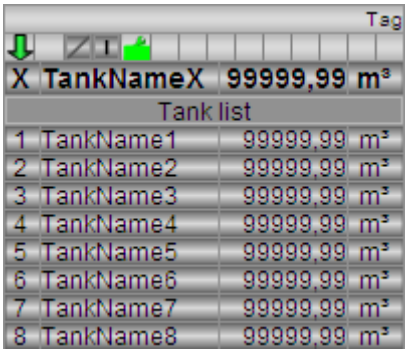
Parameter	Description	Type	Default
OnAct	1=On mode is active	DigVal	
TransAct	Transaction: 0 - not working, 1 – working	DigVal	
TransIndex	Location index value	INT	1
PushOutAct	Push Out: 0 - not working, 1 – working	DigVal	

Parameter	Description	Type	Default
PushOutIndex	Push out index value	INT	1
OosAct	1=Out of service is active	DigVal	
ReqSwitchOut	Feedback request switch route	DigVal	
CurSource	Current source	DINT	
CurDest	Current destination	DINT	
NxtSource	Next source	DINT	
NxtDest	Next destination	DINT	
NxtLocValid	Next locations are valid	BYTE	
Nxt2Loc	Next second Location ID	DINT	
Nxt3Loc	Next third Location ID	DINT	
LocIdX	Act. fix Location ID	DINT	0
LocId1 ... LocId8	Act. Location ID 1 ... 8	DINT	0
LocSpX	Act. fix Location SP	AnaValFF	0
LocSp1 ... LocSp8	Act. Location SP 1 ... 8	AnaValFF	0
PvLocIdX	Previous fix location ID	DINT	0
PvLocId1 ... PvLocId8	Previous 1 ... 8 location ID	DINT	0
ActSpSrc	Actual setpoint for source	AnaValFF	
ActSpDst	Actual setpoint for destination	AnaValFF	
PvSpSrc	Previous setpoint for source	AnaValFF	
PvSpDst	Previous setpoint for destination	AnaValFF	
EndTrans	End position of location list	BOOL	
EndPushOut	End position of push out list	BOOL	
RC1_NoValIndex	No valid route control location index	INT	
RC1_NoValReq	No valid route control request RC1	BOOL	
RC2_NoValIndex	No valid route control location index	INT	
RC2_NoValReq	No valid route control request RC2	BOOL	
UnitIDOut	Unit ID	DWORD	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING32	
StepNo	Batch step number	DWORD	
ErrorNum	Error number	INT	-1
OS_PermOut	Operator permission: output 1 for OS	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
OS_PermLnk	Operator permission (OS info)	DWORD	16#0
OpSt_Out	Enabled operator stations	DWORD	

Parameter	Description	Type	Default
ST_Worst	Worst signal status	DWORD	16#80
Status1	Status1 data	DWORD	

22.1.5 Operator control and monitoring

22.1.5.1 Symbol



Status bar
Current source / destination | Quantity | Unit
Transfer list
1st Source / Destination | Quantity: | Unit:
...
...
...
...
...
...
8. Source / Destination | Quantity: | Unit.

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	Transfer type	
		1 source → 1 to 8 destinations
		1 to 8 sources → 1 destination
2	Empty	
3	Operating mode	
		On
		Out of service
4	Mode	
		External
		Internal
5		Enable maintenance operation
6 ... 13	Empty	
14		Memo is available

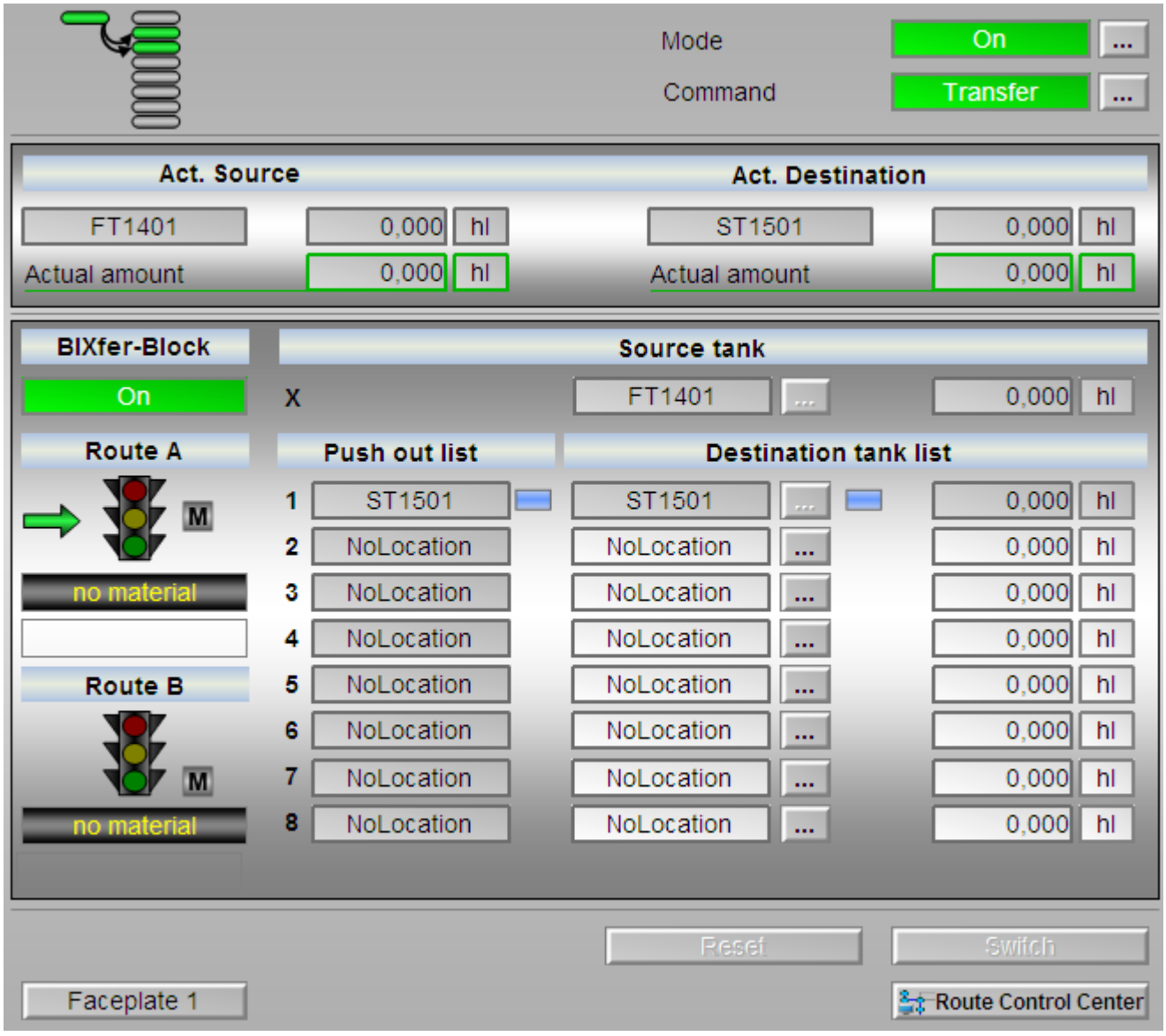
22.1.5.2 Faceplate

Views of the Qing block

The block Qing provides the following views:

	Standard view	See below.
	"Filter settings" view	See below.
	Trend view	Standard trend view of all the blocks of the Advanced Process Library. The following measured values are displayed here: • Tank fill level of source tank block parameter `ActPvSrc, ActSpSrc` • Tank fill level of destination tank block parameter `ActPvDst, ActSpDst`
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

Standard view of the block Qing

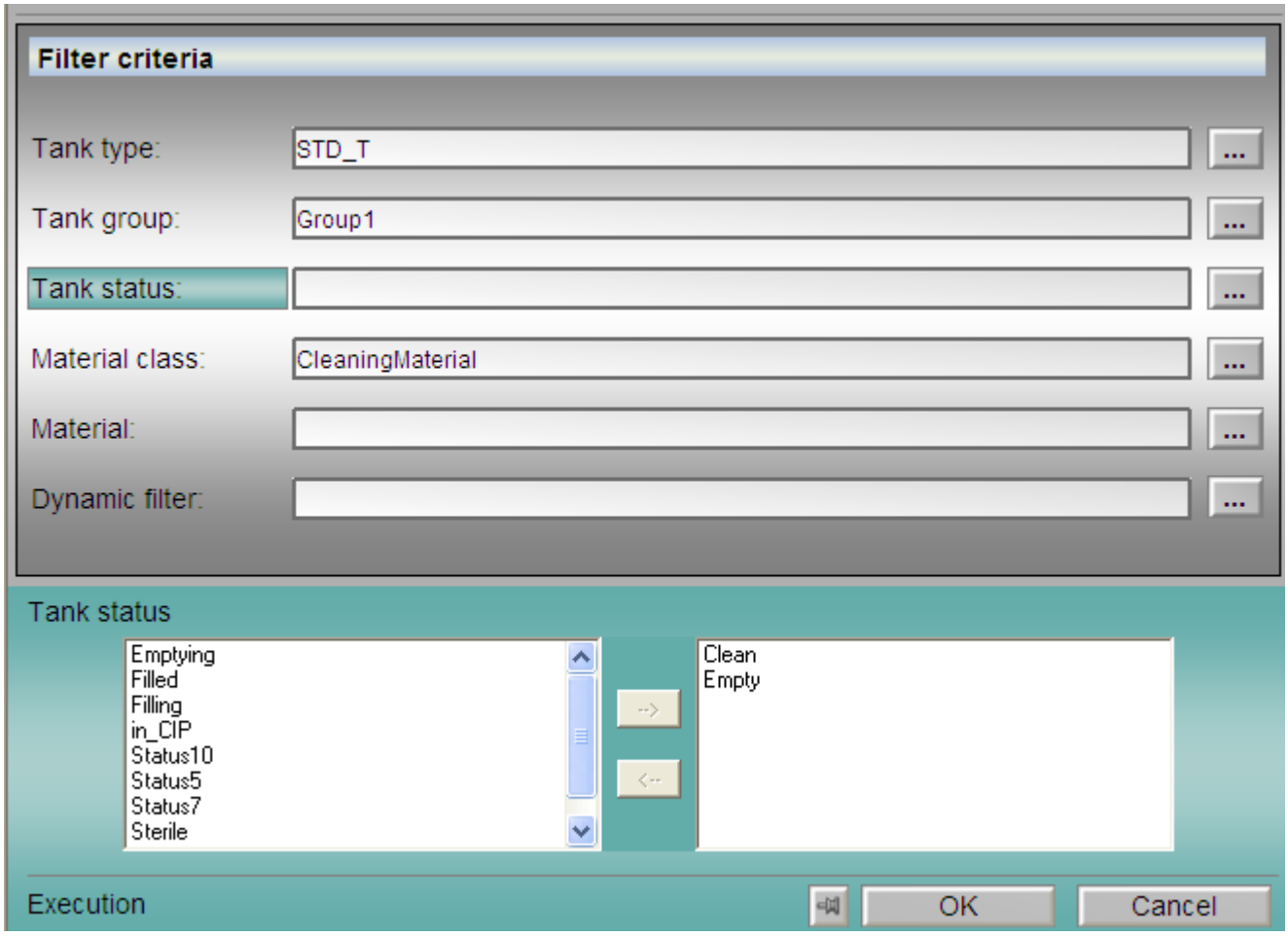


Explanation of the screen shot	
(1)	**Displaying and switching the operating mode**
	This area provides information on the currently valid operating mode. The following operating modes can be shown and executed here: • On • Out of service

Explanation of the screen shot	
(2)	**Displaying and switching the command**
	The following transfer mode can be shown and executed here: • **Stop** Transfer list is inactive. Unrestricted processing is possible in manual mode • **Transfer** Transfer list is active. Unit points that are not active can be processed in manual mode. • **Pushout** Output list is active. No unit points can be processed in manual mode. The change to be pushout mode is only possible during running transfer mode.
(3)	**Display of the source / destination data active during transfer list**
	This section shows you the current source / destination parameters while the transfer list is running. • Current source tank, name, setpoint quantity / unit, actual value quantity / unit • Current destination tank, name, setpoint quantity / unit, actual value quantity / unit
(4)	**1 x source parameter at type 1:n transfer or destination parameter at type n:1 transfer**
	The respective parameters "Unit point / tank" as well as the setpoint (quantity including unit) are displayed here. These can also be changed in Internal operation.
(5)	**8 x destination parameter at type 1:n transfer or source parameter at type n:1 transfer**
	The respective parameters "Unit point / tank" as well as the setpoint (quantity including unit) are displayed here. These can also be changed in Internal operation. • Input of transfer parameters: Clicking the input buttons opens a list box for selecting the tank. The tanks offered there can be filtered from the totality of configured tanks. The filter criteria are defined in a separate faceplate view (see next section).
(6)	**Display of the transfer constellation of the block instance 1:n or n:1**
	The symbol represents the transfer constellation (input parameter `LocXType`):
	 1:n operation `(LocXType = 'Source')`
	 n:1 operation `(LocXType = ,Destination')`
(7)	**Transfer block state**
	The field displays the operating mode of the connected transfer control block Xfer • `Xfer` is in the operating mode "On" • `Xfer` is in the "Out of service" operating mode

Explanation of the screen shot		
(8 / 9)	**State of the transfer routes – Blocks (Route A, Route B)**	
	The operating mode AUTO or MANUAL as well as the status of both transfer routes "Route A" and "Route B" are displayed:	
		AUTO operating mode
		MANUAL operating mode
		Route is inactive (IDLE)
		Route has an error (ERROR)
		Route request is running or route is ready to start (STARTING) or is held (HELD)
		Route is running and elements are being controlled (RUN)
	Clicking on a traffic light symbols starts the RCS faceplate with the associated route. Status display in the upper text field: • Material name • 'No material' → Feedback MAT-ID <= 0 from Xfer block • 'Unknown material' → If the MAT-ID (Xfer block) cannot be converted into a name	
	Status display in the lower text field: • Request OK → Feedback request OK from RCS • Feedback OK → Feedback route OK from RCS • Hold signal → Feedback "Route has been stopped" from RCS • Error → Feedback "Group error" from RCS • Error / Ignore → Feedback error AND error ignore from RCS	
(10)	**Command buttons**	
	• Reset → Only possible at inactive transfer or pushout list. The block is set to the idle state. Both transfer lists are deleted (Internal mode) • Switchover → If transfer or output list is active, system switches over to the next tank • Route Control Center → Opens the Route Control Center with further details about this route (incl. all elements and their status). **Note:** Only available at RCS Client stations!	
(11)	**Navigation button to the standard view of any faceplate**	
	This navigation button is used to reach the standard view of a block connected at the block input **SelFp1** in the CFC.	

'Filter settings' view



Filter criteria for the tank selection list boxes of the standard view (previous section) can be defined in this view. When a tank selection list box is opened, only those tanks are listed whose current state (type, group, status, material class, material) fulfill these filter criteria.

The following attributes are available in the filter criteria section:

Explanation of the screen shot

Tank type

The tank type (for example GT_T, CIP_T) is configured at every instance of the ‚Tank' block. The possible tank types are specified in an enumeration (default = ‚lTankType').

Tank group

The tanks can be combined in the project to individual project groups. The tank group is configured at every instance of the ‚lTank' block. The possible tank groups are specified in an enumeration (default = ‚lTankGroup').

Tank status

The ‚lTank' block has status machines that provide information about the current state of the tank (default = 'undefined' / 'Empty' / 'Cleaned', etc.). The possible tank states are specified in an enumeration (default = ‚lTankStatus').

Material class

The available materials can be combined into individual material classes. The derivation of the material classes from the material codes is carried out on the basis of a fixed divisor factor (default = 1000). This requires that the material codes be structured correspondingly during their definition in SIMATIC BATCH.

Material

In the list box the available materials are read from SIMATIC BATCH and displayed, if appropriate filtered in accordance with the specifications in the material class field. When the tank selection box is opened later in the standard view, only those tanks are displayed that currently contain this material.

Dynamic filter attributes

Here you define whether, after the first tank has been selected, additional filtering of the further tanks is to take place in the destination (1:n transfer) or source list (n:1 transfer). If, for example the first destination tank has been selected for a 1:n transfer sequence, it is possible to exclude for example tanks with other materials or tank groups from the further destination tanks, although the static filter would allow these. This makes it possible to define a 'less restrictive' static filter for all the tanks and a more restrictive dynamic filter additionally for the further tanks of the 1:n transfer list.

All static attributes (material, material group, tank group, tank type, tank status) can be selected as the dynamic criterion.

The following rules apply to the filter settings:

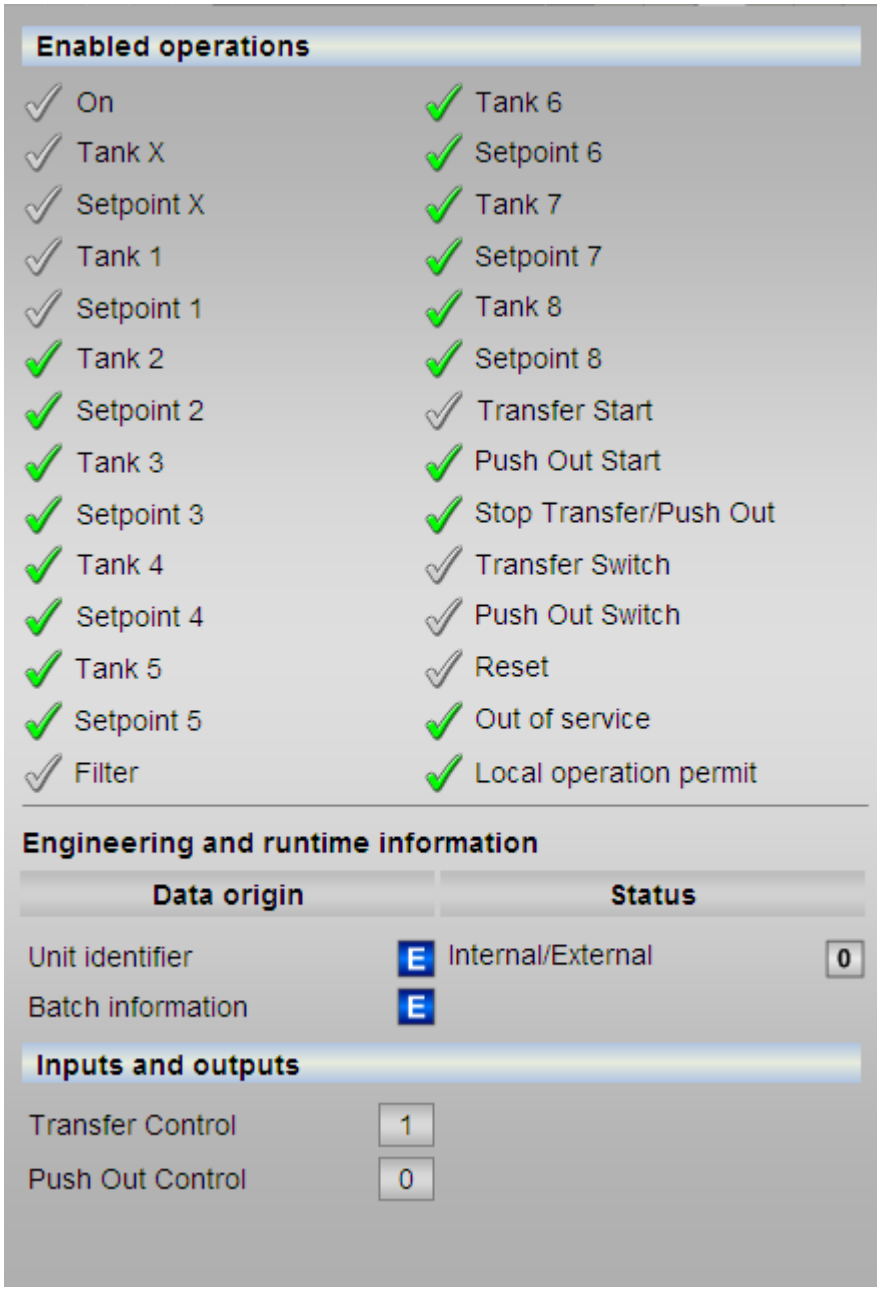
- The static filter attributes have an AND logic operation.
- The filter settings are stored at string input parameters. These parameters can also be edited in the process object view of the SIMATIC Manager or be copied to other Qing instances.
- An empty filter criterion has the effect that this attribute is ignored or that no filtering is carried out for it.
- Up to 16 specification are possible within a filter attribute. These are separated in the text field by semicolons. During the evaluation when the tank selection list box is opened an OR logic operation is applied to them. Order evaluation does not take place within an attribute.
- The operator authorization level 6 = higher process controlling is required to change the filter criteria.

Example:

The following logic operations can be specified for the filter setting shown in the above screenshot:

- (Tank type = 'GT_T') AND (Tank group = ‚Group1') AND (Tank status = 'Emptied') AND (Material class = ‚BrightBeer') AND (Dynamic = 'Material')

Preview of Qing



Explanation of the screen shot		
(1)	**Operator control enables**	
	This section shows all the operations for which there are special operator authorizations. They depend on the configuration of the `OS_Perm` parameter of this block. The operator control enables shown in the screenshot are possible here. • Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions	
(2)	**Engineering and runtime information**	
	This section contains information from the status word `Status1`	
	• The "Data origin" column shows the setting of the overwrite signals.	
	• Unit identifier	`OvwUnitID`
	• Batch information	`OvwBatchInfo`
	The symbols have the following meaning	
		Signal origin = Unit / `CmData` structure
		Signal origin = Block input `Ovw<xxx>`
	• 'States' column displays the state of the input parameter LocLiOp (0 / 1) :	
	`LocLiOp = 0`	Unit points / locations from operator (internal)
	`LocLiOp = 1`	Unit points / locations from interconnection / SFC parameters (external)
(3)	**Inputs and outputs**	
	The status of the following outputs is displayed in this section.	
	Transfer control	`TransAct` (0 = inactive, 1 = active)
	Pushout control	`PushAct` (0 = inactive, 1 = active)
(4)	**Navigation button to the standard view of any faceplate**	
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp2` in the CFC.	

22.2 Xfer – Transfer routes – Control block

22.2.1 Description of 'Xfer'

Area of application and operating principle

The block provides a mechanism for smooth activation and switching over of routes. It is possible to run 2 routes consecutively by switching them over or also to control 2 routes parallel independently of each other. Both routes use the same function definitions.

- If, during a transfer from Source A to Destination B, either Source A becomes empty or Destination B becomes full (or any other condition for switching over is fulfilled), the current route with the Combination A-B is switched over to a new combination A'-B or A-B' (A' and B' are the sources or destinations and come from the user application). After the switchover a new source-destination combination can be prepared.
- It is possible to switch over smoothly, meaning that all the valves, pumps, etc. that are controlled by both routes may not be switched off in the first route and then actuated again the new route. This is achieved by starting the new route in "Standby mode".
- The block makes it possible to operate two routes in parallel. A switchover to the standby route is possible.

Advantages for engineering:

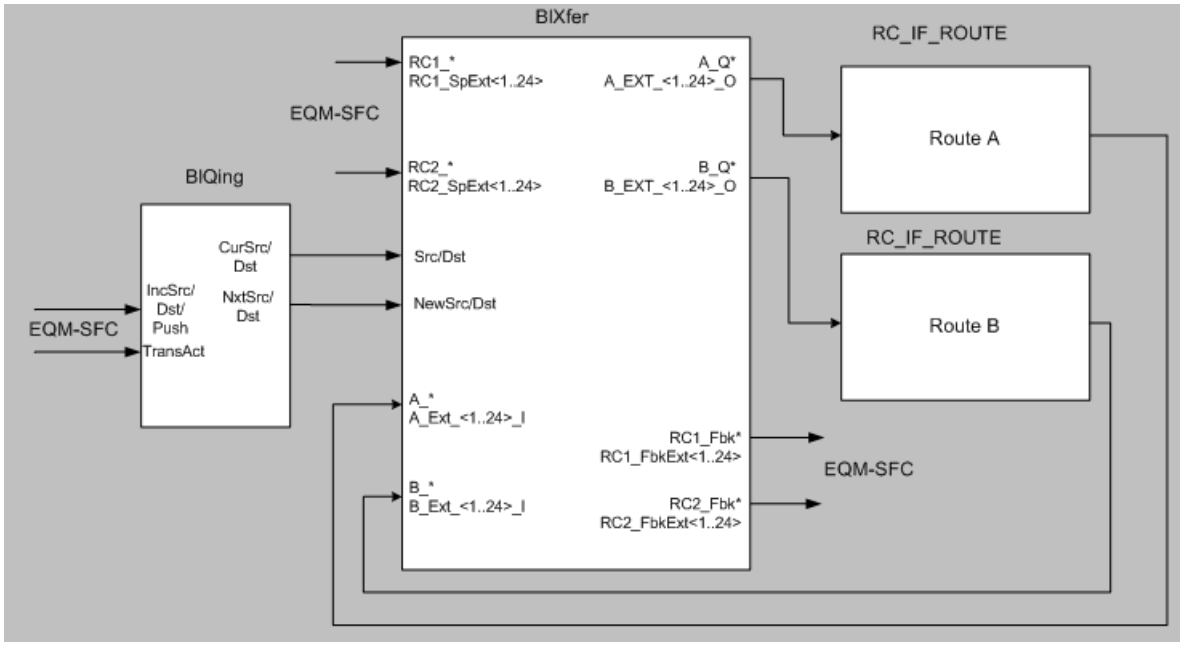
- The engineering of the associated SFC in combination with the 'Xfer' block is far more transparent and user-friendly than the solution that controls the two RC_IF_ROUTE blocks individually. In addition, coordination of the routes that run parallel is carried out by the 'Xfer' block.
- In combination with the 'Qing' block a number of transfers can be executed without deactivating the routes.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). Two RCS interface blocks RC_IF_ROUTE are furthermore required for the master and standby route.

The CFC connections between the EQM SFC and Xfer block as well as to the RCS IF blocks are manifold and are specified in the supplied CFC template.

Interconnection principle



The two routes are controlled through the Route Control interface blocks RC_IF_ROUTE. Activation of these blocks is carried out through the 'Xfer' block. The first RC_IF_ROUTE block controls the active route, the second route block is started in standby mode and is only still waiting for its activation. After the switchover (`ReqSwitchOp, ReqSwitchLi`) the two RC_IF_ROUTE blocks swap their roles. The second block becomes the master, the first one becomes the standby and can now have new values for source / destination assigned to it. The 'Xfer' block handles this mechanism transparently for the application program (SFC logic). The SFC thus operates only on one fixed interface instead of on the two blocks with the switchover.

Note on the installation sequence:

Note that the specified blocks of this sequence including EQM-SFC have to run in the same OB to ensure that the modes and controls requested by the EQM SFC are not lost before the RCS interface block is reached. The installation sequence is:

- → EQM-SFC → Xfer → RC_IF_ROUTE (A) → RC_IF_ROUTE (B)

Description / Purpose	Enumeration (preset)	Block parameter (Xfer)
Enumeration for the function table for RCS	BlModeTable	`ModeNames`
Enumeration for the active control steps	BlXferSteps	`StepNames`
Enumeration for the switch off masks	BlXferMasks	`MaskNames`
Enumeration for source or destination	BlTankName	`SrcNames, DestNames`

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'Xfer' (Page 1141)".

Status bit	Parameter
0	BatchOcc
1	BatchEn
2	xLocLiOp
3	OosAct.value
4	OosLi.value
5	nAct.value
6	SwitchedOut 0= Route A/1= Route B is Master
7	1 = Route A Automatic; 0 = Manual
8	1 = Route A request OK
9	1 = Route A running
10	1 = Route A request; 0 = stopped
11	1 = Route A error
12	1 = Route A error; 0 = Ignore error
13	1 = Route A on
14	1 = Route B Automatic; 0 = Manual
15	1 = Route B request OK
16	1 = Route B running
17	1 = Route B request; 0 = stopped
18	1 = Route B error
19	1 = Route B error; 0 = Ignore error
20	1 = Route B on
21	RC1_FbkReqOk
22	RC2_FbkReqOk
23	OvwUnitID
24	OvwBatchInfo
	Reserve

22.2.2 Operating modes of 'Xfer'

The block can be operated using the following operating modes:

- "On"
- "Out of service"

General information on these operating modes is available in the "Operating modes of the blocks" section of the APL documentation.

22.2.3 Functions of 'Xfer'

Operating mode external / internal selection

The input parameter `LocLiOp` specifies whether the selection of transfer sources and destinations is carried out by the operator in the faceplate or through the automatic program of the EQM-SFC (external block inputs).

LocLiOp = 1 "External operating mode"

In external mode the transfer parameters (Location IDs, setpoints / quantities) are derived from interconnectable inputs, (EQM-SFC or ‚Qing' block). These are displayed in faceplate, but cannot however be changed. The master / standby route switchover is also carried out through an interconnectable input signal.

LocLiOp = 0 "Internal operating mode"

In internal mode the parameters are specified by the operator in the faceplate. The master / standby route switchover is also carried out in the faceplate. This operating mode is intended primarily for tests and diagnostics during the commissioning phase.

The following table describes the most important block functions as well as the associated block parameters valid for both operating modes.

Block function	'Internal' operating mode Operator (faceplate)	'External' operating mode input parameter (interconnected)
Specification of the source / destination unit points	`SourceOp, NewSourceOp` `DestOp, NewDestOp`	`SourceLi, NewSourceLi` `DestLi, NewDestLi`

Specification of the RCS commands and modes	`RC1_AutOp, RC2_AutOp`
	`RC1_OnOp, RC2_OnOp`
	`RC1_OffOp, RC2_OffOp`
	`RC1_StepNoOp,`
	`RC2_StepNoOp`
	`RC1_MaskNoOp,`
	`RC2_MaskNoOp`
		`RC1_On, RC2_On`
		`RC1_Off, RC2_Off`
		`RC1_Hold, RC2_Hold`
		`RC1_Stop, RC2_Stop`
		`RC1_Ack, RC2_Ack`
		`RC1_StepNo, RC2_StepNo`
		`RC1_MaskNo,`
		`RC2_MaskNoRC1_SpExt(x),`
		`RC2_SpExt(x)`

Master / Standby route switchover	`ReqSwitchOp`	`ReqSwitchLi`
Commands for resetting, out of service	`RstOp, OosOp`	`RstLi, OosLi`

Note

The abbreviation '(x)' stands for the up to 24 interconnectable RCS external parameters x = 1 to 24

Additional block functions:

- The outputs NewSourceAvl and NewDestAvl are set when new source / destination values are selected for the standby route. If for example, input NewSourceLi/Op has a location that is different from the current value of the parameter SourceLi/Op , NewSourceAvl = 1 is set. If both source and destination change, both outputs are set. Further changes to the source or destination are ignored when the standby route has been started. As soon as the standby route is deactivate, the outputs are reset. They are also reset after a trailing edge of the input ReqSwitchLi. The outputs are only set if RC1_On is set (Feature Bit 0 = 1 → 'Normal behavior' mode) or RC1_On and RC2_On are set (Feature Bit 0 = 0 → 'Concurrent routes' mode)

  Note

  The block takes over the location IDs at the inputs SourceLi/Op, DestLi/Op or NewSourceLi/Op, NewDestLi/Op only as long as RC<x>_On=0. As soon as RC<x>_On=1 is set, the IDs are no longer evaluated. Consequently the IDs from the previous cycle are stored internally and passed on to the SIMATIC Route Control in the cycle in which RC<x>_On=1 is set (= 1 cycle delay).

- The block can administer up to 64 function definitions (steps) to control the route. Each definition can have a bit pattern assigned to it to set the 32 mode bits. The corresponding function definition is passed on through the outputs A_QMODE / B_QMODE to the RC_IF_ROUTE blocks in accordance with the Step number 1 to 64 at the inputs RC1_StepNo/ RC2_StepNo (set as a rule by SFC). The function definitions are configured in a separate ES configuration dialog (see the section "Configuration of the function and mask definitions (Page 1148)") and stored at the STRING parameters StepDef1 ... StepDef4 . The names of the modes are configured at the input parameter ModeNames (enumeration default = 'BlModeTable'). The names of the steps are configured at the input parameter StepNames (enumeration default = 'BlXferSteps').
- In order to handle exceptional situations in the user process, a pump can, for example, be deactivated before the route is deactivated. Four mask definitions are available to this purpose with which individual mode bits are deactivated (masked) without deactivating the complete route. The mask definitions are configured using the ES configuration dialog and are stored at the STRING parametersMaskDef1. The names of the mask definitions are configured at the input parameter MaskNames (enumeration default = 'BlXferMasks'). To handle exceptions the SFC sets the input RC1_MaskNo/RC2_MaskNo to a value 1 to 4. Now an AND logic operation is performed bitwise on the made mask of the current step Step<xx> with the inverted mask definition and passed on to the RC_IF_ROUTE block. If the value = 0, masking is deactivated.
  Example:
  Let us assume the mode mask of Step05 = 00011111 and the mask definition of Mask01 = 00000100. At an activated RC1_StepNo = 05 and RC1_MaskNo = 01 the mode mask A_QMODE = 00011011 is passed on at the RCS interface block for Route 1 (Bit 2 has been masked).

- `UnBlockingMask` enable input for the 32 mode bits:
  The interconnectable enable input allows the user to deactivate mode bits by setting individual bits = 0 (temporarily from the program) .
  Example:
  The pump has to be switched on during a transfer → Step programming = Pump on (buffer tank is filled). If the buffer tank is full (for example because the decrease in the tank is too low), the pump has to be halted "temporarily". When the level is slightly lower again, the pump pumps again. This is not an error scenario!
  There are other examples. For example: to activate the mode "reset" at a particular moment.
  The enable mask is displayed in the faceplate.
- The outputs `RC<x>_FbkStepNo` display the status of the aggregates currently being controlled. If the feedbacks agree with the controls, the value of `RC<x>_StepNo` is output (1 to 64). If not, 1 is output.
- The outputs `RC<x>_FbkMaskNo` display the status of the aggregates currently to be deactivated. If the feedbacks agree with the controls, the value of `RC<x>_MaskNo` is output (1 to 4). If not, 1 is output.
- Different input signals are used for the feedbacks in accordance with the input parameter `MaskFbkRestPos`.
  Instance 1 - `MaskFbkRestPos` = 1 (default): Input parameters `x_RESTPOS` are evaluated (to be interconnected with RC_IF/ QRESTPOS)
  Instance 2 - `MaskFbkRestPos = 0`: Input parameters `x_MODE_AC` are evaluated (to be interconnected with RC_IF/QMODE_AC)
- The input `ReqSwitchLi/Op` (positive edge) allows switching over between `SourceLi/Op`, `DestLi/Op` and `NewSourceLi/Op`, `NewDestLi/Op`. Prerequisite is that both routes have already been requested through `RC1_On / RC2_On` . The desired step number that addresses the associated mode definitions is set at the input `RC1_StepNo / RC2_StepNo` . In the case of a positive edge of `ReqSwitchLi/Op` the mode is switched over under `A_QMODE / B_QMODE` . In the case of a negative edge of `ReqSwitchLi/Op` the active route is deactivated and `SourceLi/Op`, `DestLi/Op` are reset (overwritten with 0).
- The tank names are configured at the input parameters `SrcNames / DestNames` through enumeration (default = 'BlTankNames') These are displayed in the faceplate for the current transfer.
- Further engineering details are available in the enclosed example project with a description as well as the standard documentation of the RC_IF_ROUTE block.

Overwrite functions

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwUnitID`
- `OvwBatchInfo`

You can find more information on this subject in the section Structure data connections EQM <-> CM (Page 70)

Call additional faceplates

This block includes the APL standard function "Call additional faceplates".

Configurable reactions using the Feature parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in the "APL documentation" in the section "*Configurable functions with the Feature I/O*". The following functionality is available for this block at the relevant bits:

Feature bits

Bit	Comment
0	Enable switchover to "Out of service" via `OosLi` (default = 0)
...	Reserve
24	1 = "Local" operator permission is active
...	Reserve

Operator permissions

Status bit	Parameter
0	1= Operator can switch to "On" mode
1	1 = Operator can switch to "Out of service" mode
2	1 = Operator can start the transfer
3	1 = Operator can start the output transfer
4	1 = Operator can terminate the transfer / output transfer
5	1 = Operator can step enable the transfer index
6	1 = Operator can step enable the output index
7	1 = Operator can reset the list
8	Reserve
9	Reserve
10	Reserve
11	Reserve
12	1 = Operator can change the static unit point
13	1 = Operator can change the setpoint for the static unit point
14	1 = Operator can change the unit point 1
15	1 = Operator can change the setpoint for the unit point 1
16	1 = Operator can change the unit point 2
17	1 = Operator can change the setpoint for the unit point 2
18	1 = Operator can change the unit point 3
19	1 = Operator can change the setpoint for the unit point 3
20	1 = Operator can change the unit point 4
21	1 = Operator can change the setpoint for the unit point 4
22	1 = Operator can change the unit point 5

Status bit	Parameter
23	1 = Operator can change the setpoint for the unit point 5
24	1 = Operator can change the unit point 6
25	1 = Operator can change the setpoint for the unit point 6
26	1 = Operator can change the unit point 7
27	1 = Operator can change the setpoint for the unit point 7
28	1 = Operator can change the unit point 8
29	1 = Operator can change the setpoint for the unit point 8
30	1 = Operator can change the filter settings
31	Reserve

22.2.4 Error correction for 'Xfer'

For information about troubleshooting all the blocks please also refer to the basic information in the APL section "*Troubleshooting*".

The following errors can be displayed for this block:

- Error numbers

Overview of error numbers

The following error numbers can be output using the `ErrorNum` connection:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.
52	1 = Invalid command (RCx_On and RCx_Off simultaneously)

22.2.5 Connections of 'Xfer'

Input parameters

Parameter	Description	Type	Default
`LocLiOp`	1=Link/Auto, 0=Manual: Input to auto/manual commands	`DigVal`	
`RstLi`	Linked reset signal	`DigValFF`	
`RstOp`	Operator reset signal	`BOOL`	
`OnOp`	1=On mode: On mode by operator	`BOOL`	
`OosLi`	1=Oos mode: Oos mode switch via link	`DigVal`	
`OosOp`	1=Oos mode: Oos mode by operator	`BOOL`	
`ReqSwitchLi`	Linked Request switch route	`DigValFF`	
`ReqSwitchOp`	Operator Request switch route	`BOOL`	

Parameter	Description	Type	Default
FuncId	Transport ID 301..	INT	-1
ModeNames	Enumeration for the modes	INT	
RC1_ModeTbl	Mode_Table of Route from EPH	DINT	
RC1_Material	Material of Route from EPH	DINT	
RC1_SetMat	Store Material in Route Elements from EPH	BOOL	
RC1_On	Switch route on from EPH	BOOL	
RC1_Off	Switch route off from EPH	BOOL	
RC1_Hold	Hold route from EPH	BOOL	
RC1_Stop	Stop route from EPH	BOOL	
RC1_Ack	Resume route from EPH	BOOL	
RC1_StepNo	Step Number to Ctrl Route from EPH	INT	
RC1_MaskNo	Mask Number to be used for interlocks e.g.switching off	INT	
RC1_AutOp	1=Auto mode: auto mode by operator	BOOL	
RC1_OnOp	1=Switch route 1 on by operator	BOOL	
RC1_OffOp	1=Switch route 1 off by operator	BOOL	
RC1_StepNoOp	Operator Step Number to Ctrl Route	INT	
RC1_MaskNoOp	Operator Mask Number to Ctrl Route	INT	
RC2_ModeTbl	Mode_Table of Route from EPH	DINT	
RC2_Material	Material of Route from EPH	DINT	
RC2_SetMat	Material of Route from EPH	BOOL	
RC2_On	Switch Route ON from EPH	BOOL	
RC2_Off	Switch Route OFF from EPH	BOOL	
RC2_Hold	Hold Route from EPH	BOOL	
RC2_Stop	Stop Route from EPH	BOOL	
RC2_Ack	Resume Route from EPH	BOOL	
RC2_StepNo	Step Number to Ctrl Route from EPH	INT	
RC2_MaskNo	Mask Number to be used for interlocks e.g.switching off	INT	
RC2_AutOp	1=Auto mode: auto mode by operator	BOOL	
RC2_OnOp	1=Switch route 2 on by operator	BOOL	
RC2_OffOp	1=Switch route 2 off by operator	BOOL	
RC2_StepNoOp	Operator Step Number to Ctrl Route	INT	
RC2_MaskNoOp	Operator Mask Number to Ctrl Route	INT	
IgnErr	Ignore error (IGN_ERR)	BOOL	
RouteOffErr	Turn off route in case of error and IGN_ERR	BOOL	
A_ID	Connection to QID of route	INT	0
A_RET_VAL	Return value from route	WORD	0
A_REQ_STA	Request state from route	WORD	0
A_ACTIVE	Route is active from route	BOOL	0
A_AUTO	Route in automatic mode	BOOL	0
A_ON	Route is on from route	BOOL	0
A_REQ	Route is requested from route	BOOL	0

Parameter	Description	Type	Default
A_HOLD	Route is held from route	BOOL	0
A_STOP	Route is stopped from route	BOOL	0
A_ERR	Route has error from route	BOOL	0
A_MAT_ERR	Error of the following material from route	BOOL	0
A_MAT_OK	Following material from route is OK	BOOL	0
A_VALID	Locations in route are valid	BOOL	0
A_MODE_AC	Mode feedback from route	DWORD	0
A_RESTPOS	Restposition: active CEs are in rest position	DWORD	0
A_RPOSERR	Restposition: error not all active CEs in rest position	DWORD	0
A_GRP_ERR	Mode error (mon,flt,restpos) reached from route	DWORD	0
A_SERV_IP	IP-Adress of actual RC-Server	DWORD	0
A_SP_MODE	Actual Setpoint Value of Mode (Input or Server)	DWORD	0
A_STATE	State from route	INT	0
B_ID	Connection to QID of route	INT	0
B_RET_VAL	Return value from route	WORD	0
B_REQ_STA	Request state from route	WORD	0
B_ACTIVE	Route is active from route	BOOL	0
B_AUTO	Route in automatic mode	BOOL	0
B_ON	Route is on from route	BOOL	0
B_REQ	Route is requested from route	BOOL	0
B_HOLD	Route is held from route	BOOL	0
B_STOP	Route is stopped from route	BOOL	0
B_ERR	Route has error from route	BOOL	0
B_MAT_ERR	Error of the following material from route	BOOL	0
B_MAT_OK	Following material from route is OK	BOOL	0
B_VALID	Locations in route are valid	BOOL	0
B_MODE_AC	Mode feedback from route	DWORD	0
B_RESTPOS	Restposition: active CEs are in rest position	DWORD	0
B_RPOSERR	Restposition: error not all active CEs in rest position	DWORD	0
B_GRP_ERR	Mode error (mon,flt,restpos) reached from route	DWORD	0
B_SERV_IP	IP-Adress of actual RC-Server	DWORD	0
B_SP_MODE	Actual Setpoint Value of Mode (Input or Server)	DWORD	0
B_STATE	State from route	INT	0
RC1_SpExt1 ... RC1_SpExt24	actual setpoint of external parameter-element	REAL	
A_EXT_1_I ... A_EXT24_I	actual feedback of external parameter-element	REAL	
RC2_SpExt1 ... RC2_SpExt24	actual setpoint of external parameter-element	REAL	

Parameter	Description	Type	Default
B_EXT_1_I ... B_EXT24_I	actual feedback of external parameter-element	REAL	
RC1_ReqDelay	delay start new request	INT	16#F
RC1_ReqCount	Request count until feedback error=1	INT	16#3
RC2_ReqDelay	delay start new request	INT	16#F
RC2_ReqCount	Request count until feedback error=1	INT	16#3
ESTransAct	Internal	INT	0
ESTransTm	Internal	INT	16#5
StepNames	Enumeration for the activate steps	INT	
StepDef1	Parameter definitions for the active modes (step 1 to 16)	STRING[128]	
StepDef2	Parameter definitions for the active modes (step 17 to 32)	STRING[128]	
StepDef3	Parameter definitions for the active modes (step 33 to 48)	STRING[128]	
StepDef4	Parameter definitions for the active modes (step 49 to 64)	STRING[128]	
ActFctAOp	Operator activates Functions Route A	DWORD	
ActFctBOp	Operator activates Functions Route B	DWORD	
MaskFbkRestPos	Parameter switch MODE_AC or RESTPOS	BOOL	1
MaskNames	Enumeration for the switch off masks	INT	
MaskDef1	Parameter definitions modes switch off (mask 1 to 4)	STRING[32]	
UnBlockingMask	Unblocking mask for the active modes	DWORD	16#FFFFFFFF
SampleTime	Sampling time [s]	REAL	1.0
CmData	Control and status information for this CM	BlUdtCmData	
OvwUnitID	1=Overwrite CmData.UnitID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
UnitID	Unit ID (usually from BlUnitIf block)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIdLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
SrcNames	Enumeration for source	DINT	0
DestNames	Enumeration for destination	DINT	0
SourceLi	Linked Source of Route	DINT	
SourceOp	Operator Source of Route	DINT	
Via1 ... Via10	VIA1 ... Via10 of Route	DINT	
DestLi	Linked Destination of Route	DINT	
DestOp	Operator Destination of Route	DINT	
NewSourceLi	New Source of Route	DINT	

Parameter	Description	Type	Default
NewSourceOp	New Operator Source of Route	DINT	
NewVia1 ... NewVia10	New VIA1 ... Via10 of Route	DINT	
NewDestLi	New Destination of Route	DINT	
NewDestOp	New Operator Destination of Route	DINT	
SelFp1	Open faceplate of connected block (type)	ANY	
SelFp2	Open faceplate of connected block (type)	ANY	
OS_Perm	Operator Permission	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

Output parameters

Parameter	Description	Type	Default
OnAct	1=On mode is active	DigVal	
OosAct	1=Out of service is active	DigVal	
ReqSwitchOut	Feedback request switch route	DigVal	
SwitchedOut	Route switched	BOOL	
Reset	Reset signal	DigVal	
NewSourceAv1	new source route available	BOOL	
NewDestAv1	new destination route available	BOOL	
RC1_FbkRetVal	diagnosis value	WORD	
RC1_FbkStepNo	feedback modes at actual step number	INT	
RC1_FbkOn	feedback route is on	BOOL	
RC1_FbkHold	feedback route is held	BOOL	
RC1_FbkStop	feedback route is stopped	BOOL	
RC1_FbkErr	feedback route has error	BOOL	
RC1_FbkReqOk	feedback route request is done	BOOL	
RC1_FbkMan	feedback route in manual mode	BOOL	
RC1_FbkRouteOk	feedback route is on and step OK	BOOL	
RC1_FbkMatErr	feedback route error of the following material	BOOL	
RC1_FbkMatOk	feedback route following material is OK	BOOL	
RC1_FbkMaskNo	feedback modes at actual mask number	INT	
RC2_FbkRetVal	diagnosis value	WORD	
RC2_FbkStepNo	feedback modes at actual step number	INT	
RC2_FbkOn	feedback route is on	BOOL	
RC2_FbkHold	feedback route is held	BOOL	
RC2_FbkStop	feedback route is stopped	BOOL	
RC2_FbkErr	feedback route has error	BOOL	
RC2_FbkReqOk	feedback route request is done	BOOL	

Parameter	Description	Type	Default
RC2_FbkMan	feedback route in manual mode	BOOL	
RC2_FbkRouteOk	feedback route is on and step OK	BOOL	
RC2_FbkMatErr	feedback route error of the following material	BOOL	
RC2_FbkMatOk	feedback route following material is OK	BOOL	
RC2_FbkMaskNo	feedback modes at actual mask number	INT	
A_EXT_1_O ... A_EXT24_O	actual setpoint of external parameter-element	REAL	
RC1_SpExt1 ... RC1_SpExt24	actual feedback of external parameter-element	REAL	
B_EXT_1_O ... B_EXT24_O	actual setpoint of external parameter-element	REAL	
RC2_SpExt1 ... RC2_SpExt24	actual feedback of external parameter-element	REAL	
IgnErrOut	ignore error	BOOL	
FuncIdOut	to route: transport id	INT	
A_QRES_MAN	Switch Route to Automatik	BOOL	
A_QMODE	Set modes active	DWORD	
A_QSOURCE	Source to route	DINT	
A_QVIA_1 ... A_QVIA_10	VIA1 ... Vial10 to Route	DINT	
A_QDEST	Destination to route	DINT	
A_QMODE_TBL	Mode_Table of Route from EPH	DINT	
A_QMATERIAL	Material of Route from EPH	DINT	
A_QSET_MAT	Set material of Route from EPH	BOOL	
A_QREQ_AU	Request route	BOOL	
A_QON_AU	Set route on	BOOL	
A_QHOLD_AU	Set route to hold	BOOL	
A_QACK_AU	Resume route	BOOL	
A_QSTOP_AU	Stop route	BOOL	
A_QMASK	Mask to switch off modes	DWORD	
B_QRES_MAN	Switch Route to Automatik	BOOL	
B_QMODE	Set modes active	DWORD	
B_QSOURCE	Source to route	DINT	
B_QVIA_1 ... B_QVIA_10	VIA1 ... Vial10 to Route	DINT	
B_QDEST	Destination to route	DINT	
B_QMODE_TBL	Mode_Table of Route from EPH	DINT	
B_QMATERIAL	Material of Route from EPH	DINT	

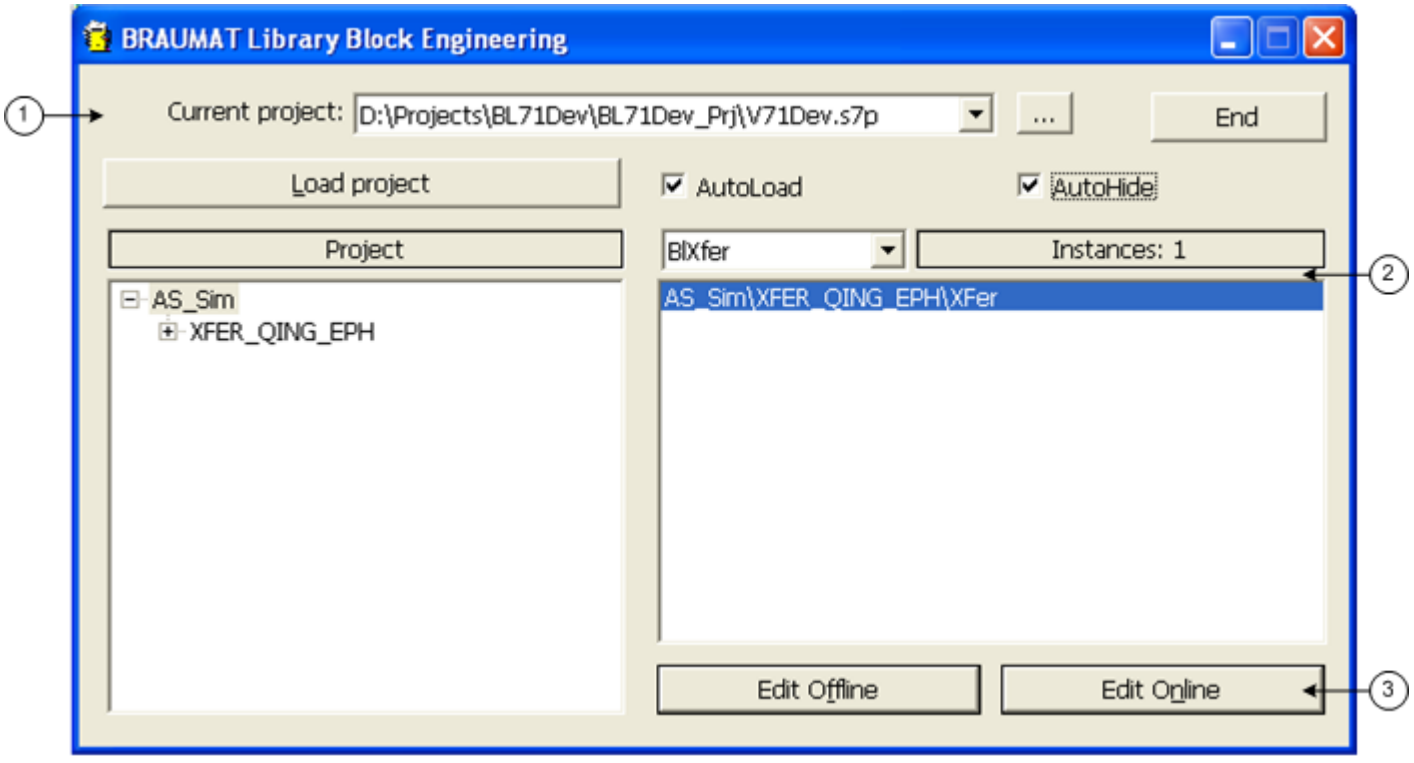
Parameter	Description	Type	Default
B_QSET_MAT	Set material of Route from EPH	BOOL	
B_QREQ_AU	Request route	BOOL	
B_QON_AU	Set route on	BOOL	
B_QHOLD_AU	Set route to hold	BOOL	
B_QACK_AU	Resume route	BOOL	
B_QSTOP_AU	Stop route	BOOL	
B_QMASK	Mask to switch off modes	DWORD	
A_QIDAdrStat	adress state of A_QID	STRING[16]	'?'
A_QReqTime	actual time after request failed	INT	
A_QReqCount	actual error counts for request	INT	
A_QStepNo	Step number to Route	INT	
A_QMaskNo	Mask number to Route	INT	
A_QRouteID	Number of actually used route 1..300	INT	
A_QServIP	IP-Adress of actual RC-Server 1..300	DWORD	
B_QIDAdrStat	adress state of B_QID	STRING[16]	'?'
B_QReqTime	actual time after request failed	INT	
B_QReqCount	actual error counts for request	INT	
B_QStepNo	Step number to Route	INT	
B_QMaskNo	Mask number to Route	INT	
B_QRouteID	Number of actually used route 1..300	INT	
B_QServIP	IP-Adress of actual RC-Server 1..300	DWORD	
UnitIDOut	Unit ID	DWORD	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING32	
StepNo	Batch step number	DWORD	
ErrorNum	Error number	INT	-1
OS_PermOut	Operator permission: output 1 for OS	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission: output 2 for OS	DWORD	16#FFFFFFFF
OS_PermLnk	Operator permission (OS info)	DWORD	16#0
OpSt_Out	Enabled operator stations	DWORD	
ST_Worst	Worst signal status	BYTE	16#80
Status1	Status1 data	DWORD	

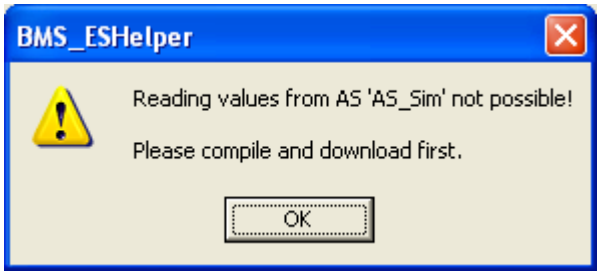
22.2.6 Configuration of the function and mask definitions

Configuration is carried out roughly in the following steps:

1. Instantiating and interconnecting of the above-mentioned blocks and adapters in the CFC Editor.
2. Creation of and parameter assignment to the enumerations in accordance with the table in the section "Description of 'Xfer' (Page 1134)".
3. Calling up the ES configuration dialog using the SIMATIC Manager menu "Options → Industry Library → Block engineering".
   The dialog for selecting the block type (in this case 'Xfer') as well as the desired block instance is displayed first.

22.2.6.1 Configuration dialog "Project overview screen"



Explanation of the screen shot	
(1)	**Selecting and downloading the ES project**
	• **Current project"** Select path for the ES project • **"Load project" button** to read the project data The project has to be loaded again after configuration changes in the `Xfer` instances and enumerations (CFC Editor or SIMATIC Manager). • **"Auto Load" check box** – The project last selected is also loaded whenever the application is started – After a new project has been selected, it is loaded immediately • **Selection box "Auto Hide"** After the editing window (Edit Offline, Edit Online) has been opened, the selection window is minimized automatically. • **"Exit" button** Closes the application.
(2)	**Select block types and instances**
	• List box for selecting the block type (here `Xfer`) Only instances of the respective type are displayed. • The CFC folders and charts with clock instances are displayed in tree view under "Project". • The charts with instances for the selected folder are listed in tree view under "Instance(s) x/y".
(3)	**Start the editor dialog**
	The editing window for the selected instance can be opened either in Offline or Online mode: • Offline: Can always be opened in the case of existing and prepared block instances. • Online: Can only be opened under the following preconditions: – Block instances have been compiled and loaded to AS. – Connection to AS is functioning and CFC test mode is possible. Note: Due to CFC restrictions *Compile and download AS changes* first has to be carried out in the CFC editor after offline changes before the editor can be started in Online mode. A message box is displayed if the preconditions for Online mode are unfulfilled.
	

22.2.6.2 Configuration dialog "Editing window"

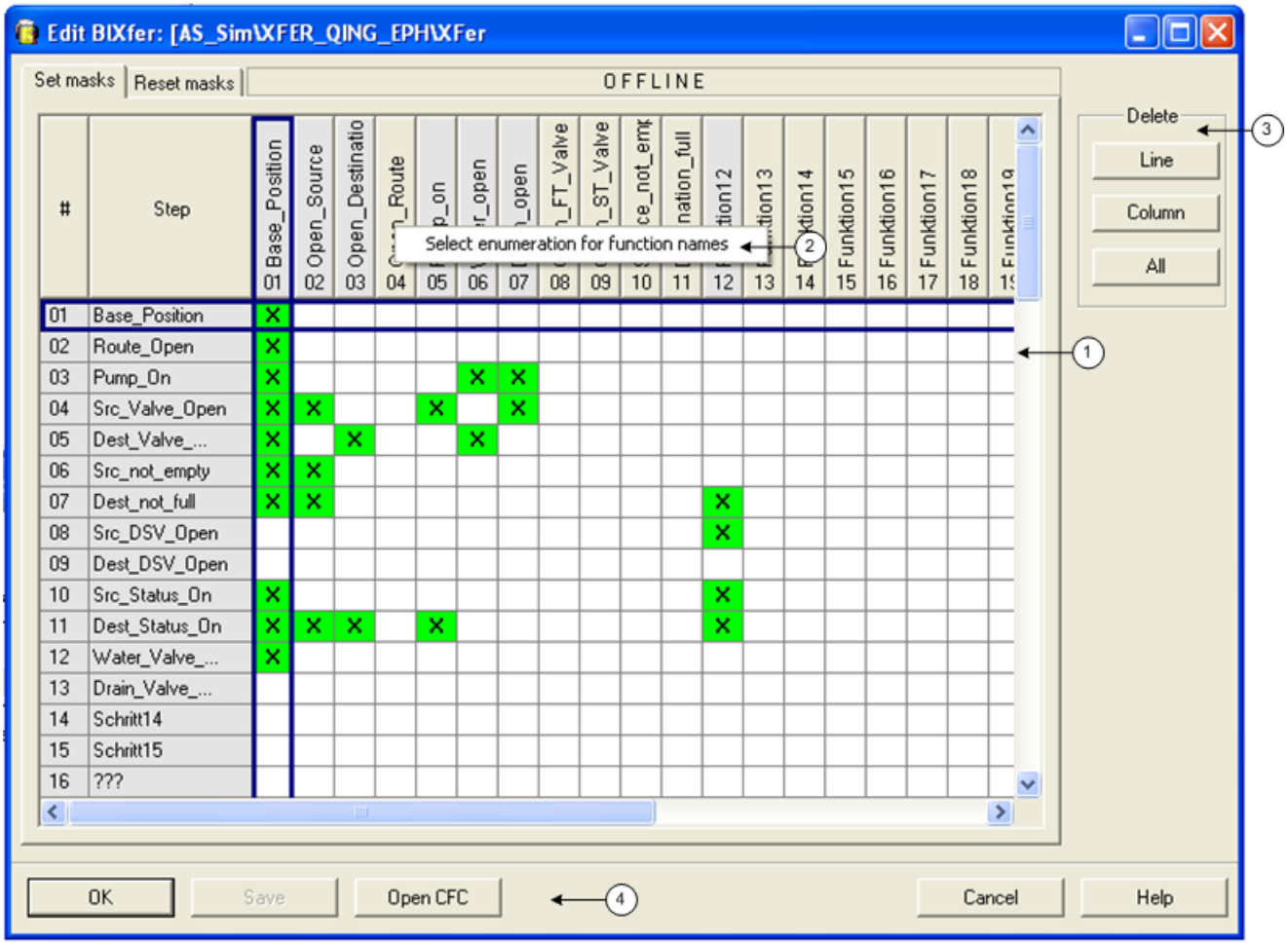
Topic

Configuration of the block instance takes place in the editing window described below. Tabs are available respectively for the configuration of the

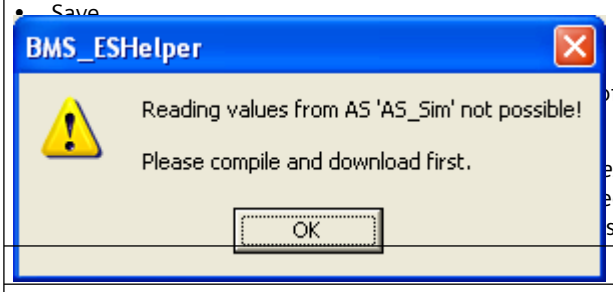
- Function definitions (Set Masks tab)
- Deactivation masks (Reset Masks tab)

NOTICE
Changes in online mode In the case of configuration changes in online mode and subsequent saving, the current block instance is written directly to the AS after a repeated confirmation prompt. Before saving you have to ensure that the configured modes cannot induce damage in the system.

The function / mask definitions are entered using the following configuration dialog:

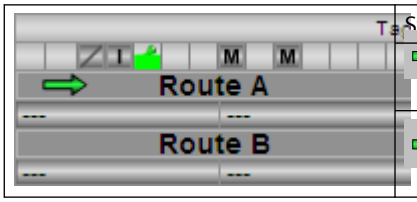


Explanation of the screen shot	
(1)	**Table with steps and function definitions (mode bits)**
	• The 64 step definitions are displayed in the lines from top to bottom. • 32 function definitions are displayed in the columns from left to right. The step / function names are derived from the correspondingly configured enumerations (see above). • Configuration of the function definition / mode bits is carried out by setting / resetting the cell currently selected in the matrix.
(2)	**Selection of the enumerations for function and step designation**
	• The enumeration for the function name can be reassigned in the shortcut menu (right-hand mouse click) of the function cell. The enumeration selected in the list is written back to the block parameter `MaskNames`. • The enumeration for the step name can be reassigned in the shortcut menu (right-hand mouse click) of the step column. The enumeration selected in the list is written back to the block parameter `StepNames`.

Explanation of the screen shot	
(3)	**Clear function**
	• Line All the mode bits are reset for the selected step • Column The selected function definition is reset in all the steps • All All the function definitions of this block instance are reset
(4)	**Command buttons**
	• OK → Editor is closed Storing of changes is carried out implicitly in Offline mode. In Online mode a query may be displayed whether changes are to be rejected. • Save  ffline es first has to be carried out in the CFC editor after offline ect view or a second instance of the configuration dialog) s displayed if the preconditions for storing are unfulfilled.
	• Open CFC The CFC editor can be opened by using the button. Compiling / Downloading can, for example, be carried out here. • Abort Exit the editor If changes in the configuration have been carried out, a prompt is displayed whether these are to be rejected. • Help • Opens the documentation window (CHM file) with the subsection of the configuration dialog

22.2.7 Operator control and monitoring

22.2.7.1 Symbol

 	Status bar
	Route A active Route A source | destination
	Route B active Route B source | destination

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	Empty	
2	Empty	
3	Operating mode	
		On
		"Out of service"
4	Mode	
		External
		Internal
5		Enable maintenance operation
6	Empty	
7	Empty	
8	Operating mode Route A	
		Automatic
		Manual
9	Empty	
10	Operating mode Route B	
		Automatic
		Manual
11 ... 13	Empty	
14		Memo is available

22.2.7.2 Faceplate

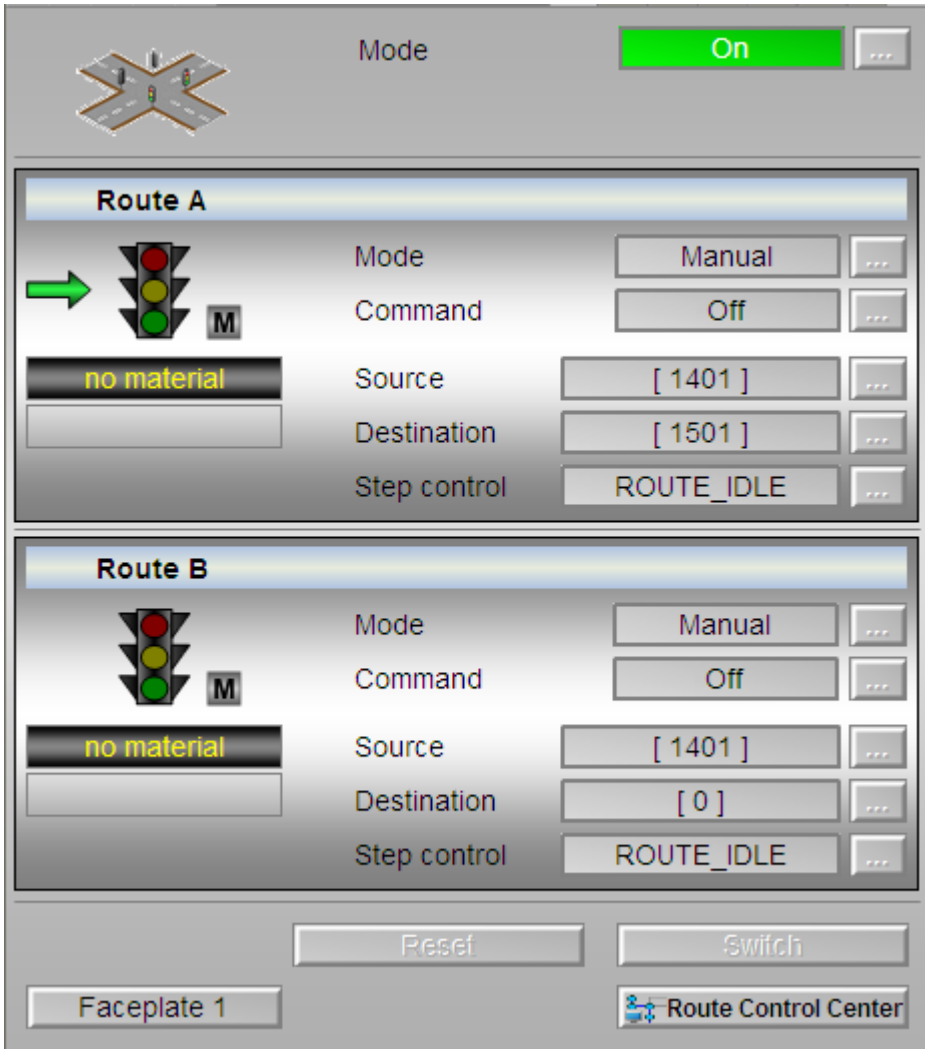
Views of the Xfer block

The block Xfer provides the following views:

	Standard view	See below.
	Parameter view	See below.
	Function definitions	See below.

	Mask definitions	See below.
	Preview	See below.
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

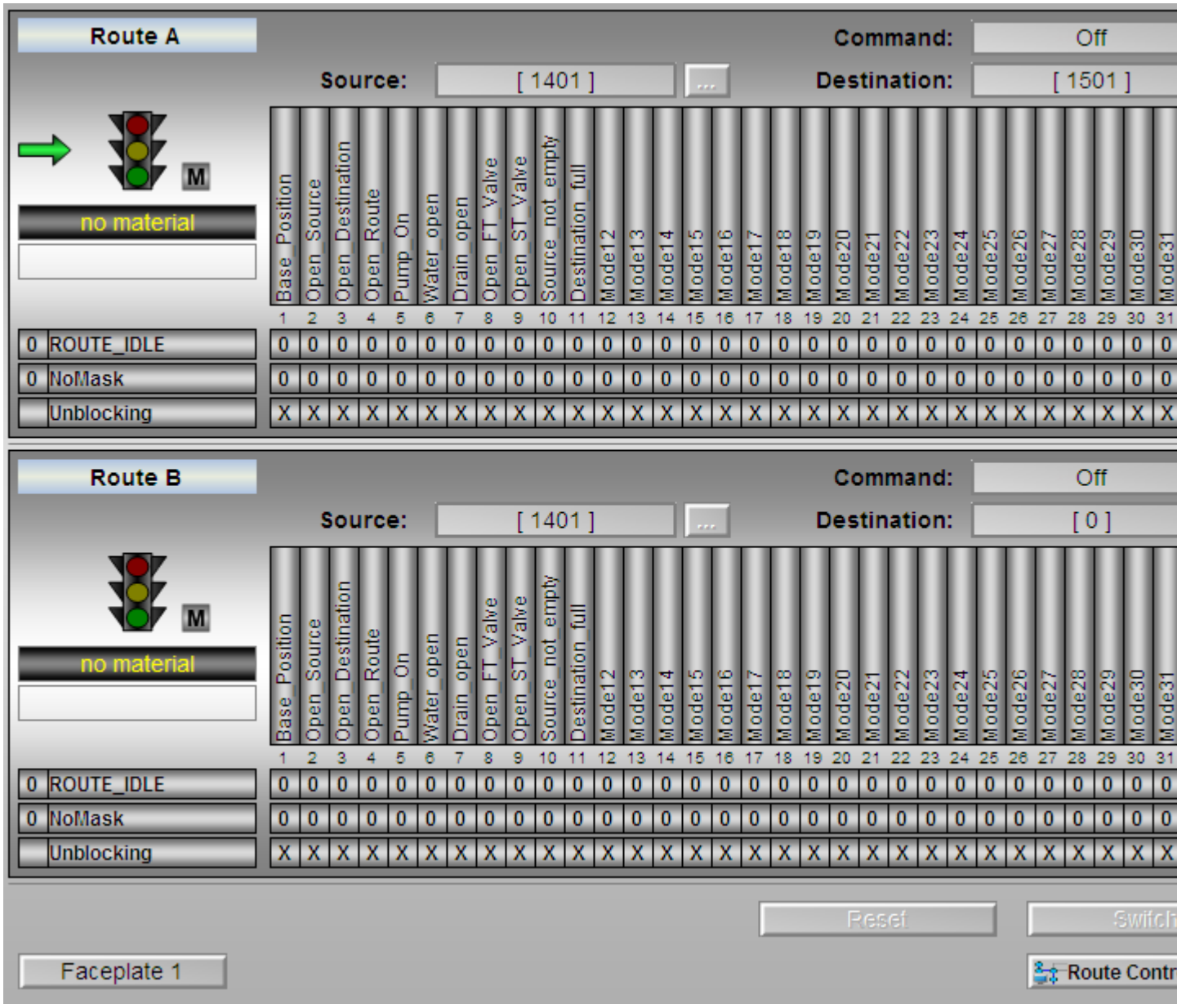
Standard view of the block Xfer



Explanation of the screen shot		
(1)	**Displaying and switching the operating mode**	
	This area provides information on the currently valid operating mode. The following operating modes can be shown and executed here: • On • Out of service	
(2/3)	**Displaying and switching over the parameters Route A / Route B**	
	This section displays the operating mode, command and parameters of the two connected route blocks. These can also be changed in Internal operation.	
	• Operating mode	Manual / Auto
	• Command	Route On / Off
	• Source	Name of the source unit point
	• Destination	Name of the destination unit point
	• Step selection	Currently controlled step. The resulting RCS function controls are displayed in the 'Parameters' view (next section). If enabled, the step can also be changed manually
(4)	**Command buttons**	
	• Reset	Currently no function
	• Switchover	If the transfer is active, the function controls are swapped (master<-> standby route swap)
	• Route Control Center	Opens the Route Control Center with further details about this route (incl. all elements and their status). **Note:** Only available at RCS Client stations!

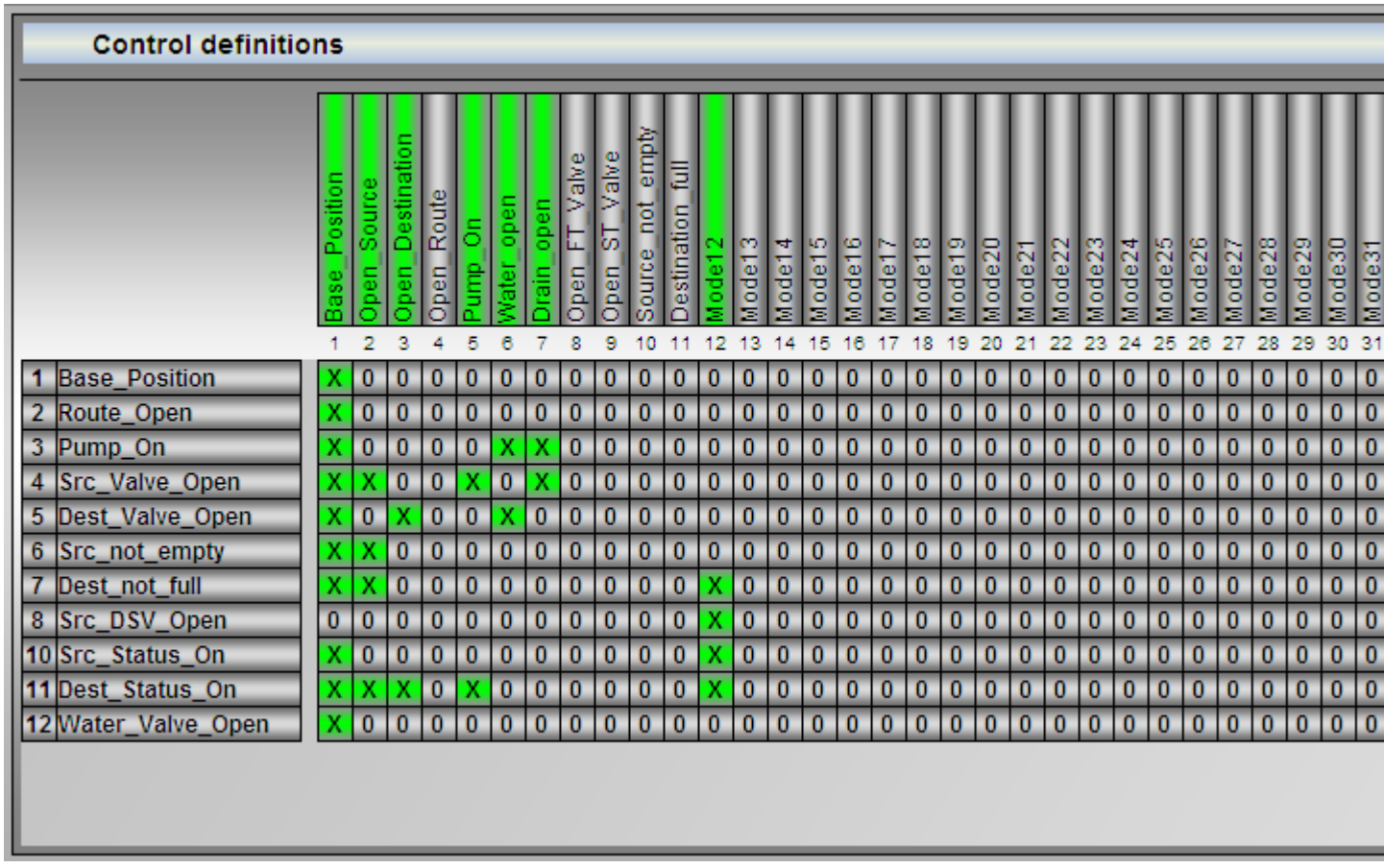
Explanation of the screen shot			
(5/6)	**State of the transfer routes – Blocks (Route A, Route B)**		
	• The operating mode AUTO or MANUAL as well as the status of both transfer routes "Route A" and "Route B" are displayed:		
			AUTO
			MANUAL
			Route is inactive (IDLE)
			Route has an error (ERROR)
			Route request is running or route is ready to start (STARTING) or is held (HELD)
			Route is running and elements are being controlled (RUN)
	• Clicking on a traffic light symbols starts the RCS faceplate with the associated route. Status display in the upper text field:		
		• Material name	
		• 'No material'	Feedback MAT-ID <= 0 from `Xfer` block
		• 'Unknown material'	If the MAT-ID (BIXfer block) cannot be converted into a name
	• Status display in the lower text field:		
		• Request OK	Feedback request OK from RCS
		• Feedback OK	Feedback route OK from RCS
		• Hold signal	Feedback "Route has been stopped" from RCS
		• Error	Feedback "Group error" from RCS
		• Error / Ignore	Feedback error AND error ignore from RCS
(7)	**Navigation button to the standard view of any faceplate**		
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp1` in the CFC.		

Parameter view of Xfer



Explanation of the screen shot	
(1/2)	**Display RCS parameters of both routes**
	This sections shows the current source / destination locations as well as the function catalogs for the two connected route blocks. Source and destination location can also be changed in Internal operation. The mode table shows the modes in the header bar on the basis of the enumeration at the block parameter `ModeNames`. The following information is displayed: • First row: Controlled mode bits of the active step – X/green = Mode is requested – 0/gray = Mode is deactivated • Second row: Mask bits of the active mask • Third row: Enable bits on the basis of the block parameter `UnblockingMask` – X = Mode is enabled (`UnblockingMask` Bit<x> = 1) – 0 = Mode is deactivated (`UnblockingMask` Bit<x> = 0) During internal operation a mode that is not controlled can be requested manually by clicking the mode name (bar color gray → yellow → blue) if a corresponding permission exists. Clicking once more deactivates the mode again. When a new step is activated, manually requested modes are enabled again. Manual mode selection is only possible if the corresponding enable bit is set (X display).
(4/5)	**State of the transfer routes – Blocks (Route A, Route B)**
	... See standard view
(3)	**Command buttons**
	... See standard view
(6)	**Navigation button to the standard view of any faceplate**
	... See standard view

'Function definitions' view



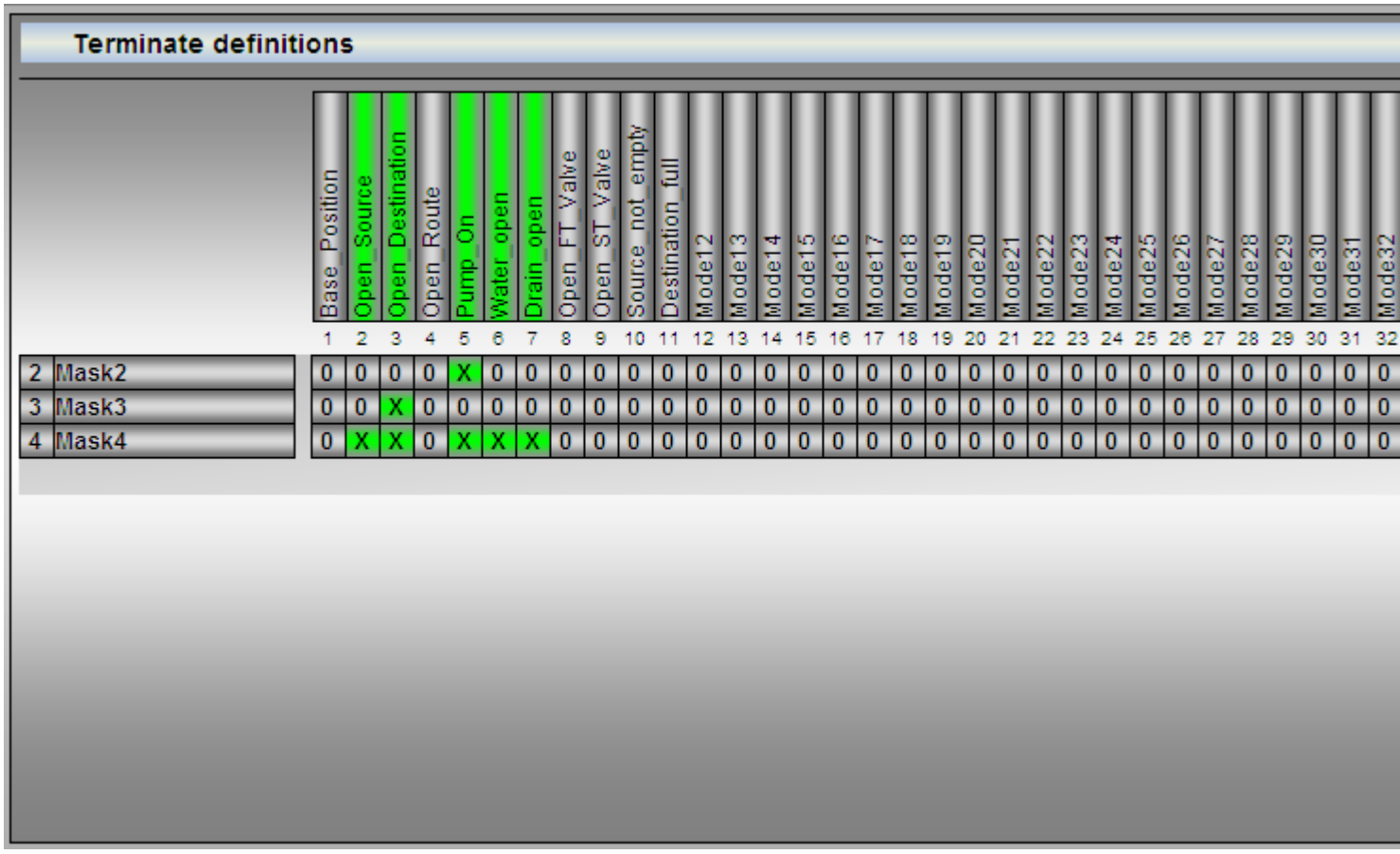
The configured function definitions of the block instance are displayed in this section. The step display for this mode can be filtered by clicking the respective column header:

- Green → active, meaning that steps with this mode are displayed
- Gray → inactive, meaning that steps with this mode are not displayed

Note

A change in the function definitions is only possible in the ES project by using the configuration dialog (see the section "Configuration of the function and mask definitions (Page 1148)"

'Mask definitions' view



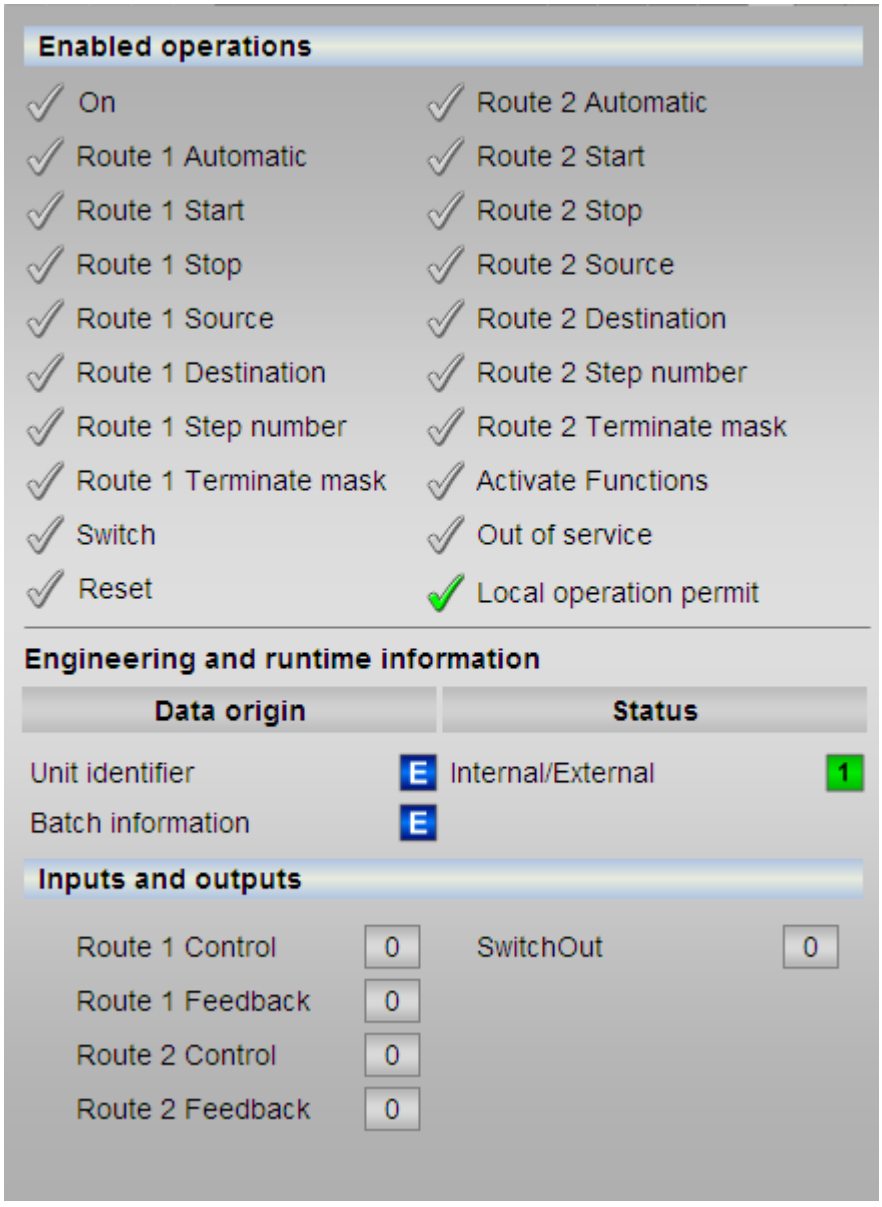
The configured deactivation definitions of the block instance are displayed in this section. The mask display for this mode can be filtered by clicking the respective column header:

- Green → active, meaning that masks with this mode are displayed
- Gray → inactive, meaning that masks with this mode are not displayed

Note

A change in the mask definitions is only possible in the ES project by using the configuration dialog (see the section Configuration of the function and mask definitions (Page 1148)

Preview of Xfer



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the `OS_Perm` parameter of this block. The operator control enables shown in the screenshot are possible here. • Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions

Explanation of the screen shot			
(2)	**Engineering and runtime information**		
	This section contains information from the status word Status1		
	• The "Data origin" column shows the setting of the overwrite signals.		
		Unit identifier	`OvwUnitID`
		Batch information	`OvwBatchInfo`
		• The symbols have the following meaning	
			Signal origin = Unit / `CmData` structure
			Signal origin = Block input `Ovw<xxx>`
	• 'States' column displays the state of the input parameter `LocLiOp` (0 / 1):		
		`LocLiOp = 0`	Unit points / locations from operator (internal)
		`LocLiOp = 1`	Unit points / locations from interconnection / SFC parameters (external)
(3)	**Inputs and outputs**		
	The status of the following outputs is displayed in this section. • Route 1 control → Output parameter `A_RC1_ON` • Route 1 feedback → Output parameter `A_RC1_FB_ON` • Route 2 control → Output parameter `B_RC2_ON` • Route 2 feedback → Output parameter `B_RC2_FB_ON` • Switchover → Output parameter `SwitchedOut`		
(4)	**Navigation button to the standard view of any faceplate**		
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp2` in the CFC.		

22.3 Tank – Tank status block

22.3.1 Description of 'Tank'

Area of application and operating principle

This function block manages tank and states and makes a runtime monitoring function with interfaces and messages for the process and quality status as well as the recording of the material allocations, including an operator control and monitoring user interface available to this purpose. Thanks to its general approach the block can be used for a wide variety of vessel types such as silos, buffer tanks, etc.

- The material management offers a FIFO with up to 10 materials, including the corresponding batch IDs. The material management can be influenced through the OS faceplate and / or control inputs.
- The process status allows the handling of any number of tank states (10 at delivery) and can be operated using the OS faceplate and / or control inputs.
- The quality status allows the handling of any number of quality states (4 at delivery) and can be operated using the OS faceplate and / or control inputs.
- 3 separate timers are available for monitoring the remaining times for material quality, tank cleaning and sterility.
- Material selection in the OS faceplate as well as the assignment of the material code to the material name is made on the basis of the SIMATIC BATCH material model.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The required CFC connections between the BlTank block and the SFC to be controlled are specified in the supplied CFC template.

The parameters of the assigned tanks are defined through the following enumerations and are configured at the corresponding block parameters of the Tank block:

Description / Purpose	Enumeration (preset)	Block parameter (Tank)
Definition tank name and assignment to the Location ID	BlTankName	`TankLocId`
Definition of the tank type	BlTankType	`TankType`
Definition of the tank group	BlTankGroup	`TankGrp`
Definition of the tank status	BlTankStatus	`TankStLi, TankStOp, Ext1TankSt … Ext4TankSt, TankMon1, TankMon2, CleanSt, UnCleanSt, SterileSt, UnSterileSt, PvTankStatus`
Definition of the quality status	BlQualityState	`Ext1QualSt … Ext4QualSt, UnQualSt, QualStOp, PvQualState`

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'Tank' (Page 1173)".

Status bit	Parameter
0	BatchOcc
1	BatchEn
2	Reserve
3	OosAct.value
4	AutStAct.value
5	ManStAct.value
6	AutMatAct.Value
7	ManMatAct.Value
8	Reserve
9	QualMonStarted
10	QualMonExpired
11	CleanMonStarted
12	CleanMonExpired
13	SterMonStarted
14	SterMonExpired
15	OvwManOpEn
16	OvwModSel
17	OvwUnitID
18	OvwBatchInfo
19	xManOpEn
20	OosLi.Value
...	Reserve
30	xMsgLock
31	MS_RelOp

22.3.2 Operating modes of 'Tank'

- The block can be operated using the following operating modes:
- "Automatic"
- "Manual"
- "Out of service"

General information on these operating modes is available in the "Operating modes of the blocks" section of the APL documentation.

22.3.3 Functions of 'Tank'

General functions

The following general input / output signals are available:

- `StLiOp`→ Linked / Operator selection for tank status operating mode specification
  - 1 = Auto / Manual operating mode specification is made via the input parameter `AutStLi/ManStLi`
  - 0 = Auto / Manual operating mode specification is made in the OS-Faceplate (inputs `AutStOp/ManStOp`)
- `MatLiOp` → Linked / Operator selection for material operating mode specification
  - 1 = Auto / Manual operating mode specification is made via the input parameter `AutMatLi/ManMatLi`
  - 0 = Auto / Manual operating mode specification is made in the OS-Faceplate (inputs `AutMatOp/ManMatOp`)
- `OvmModSel` → Determines which switchover signals are to be used:
  - 1 = Inputs `StLiOp/MatLiOp` are used
  - 0 (default) = The selection of the operating mode specification for the status and material is derived from `CmData.xModLiOp`
- In order to supply batch-related information to the faceplate system status messages, the block includes individual inputs as well as of the structure input `CmData`. The switchover signal `OvwBatchInfo` is used for the selection:
  - 1 = `BatchEnLi, BatchOccLi, BatchIdLi, BatchNameLi` and `StepNoLi` are used
  - 0 (default) = The batch information is derived from `CmData`
- The switchover signal `OvwUnitId` is used to select the unit ID
  - 1 = Input `UnitId` is used
  - 0 (default) = The unit ID is derived from `CmData.dwUnitID`
- All the free ALARM_8P signals are available through the individual block interfaces.

Process status, output `PvTankStatus`

The Tank block provides a tank process status management with different states (default = 11). The status definitions are contained in the enumeration "BlTankStatus" that is located in the "Shared Declarations" folder of the project. The following states are defined and preset here:

Value	Object name
0	Undefined
1	Cleaned
2	Sterile
3	Fill
4	Filled

Value	Object name
5	Status5
6	Empty
7	Status7
8	Emptied
9	Cleaning
10	Status10

The process status can be influenced both by the OS faceplate (manual operation) and the block inputs (automatic operating mode). The preferred operation can be selected through the input `StLiOp` .

- `StLiOp = 0`
  Auto / Manual switchover is carried out through the OS faceplate (inputs `AutStOp`/ `ManStOp`)
- `StLiOp = 1`
  Auto / Manual switchover is carried out through interconnectable inputs `AutStLi`/ `ManStLi`)

AUTO mode and `TankStLi > 0`:

The tank status of input `TankStLi` is imported into the process status `PvTankStatus` .

AUTO mode and `TankStLi = 0`:

If `TankStLi = 0` (undefined), the output `PvTankStatus` retains the last valid value and the process status can be influenced through the signal inputs `HiLimSig, LoLimSig, InSig3` and `InSig4`. The four inputs can be interconnected freely, but the following preferred allocation should be observed:

- `HiLimSig` = Tank HighLevel probe
- `LoLimSig` = Tank LowLevel probe
- `InSig3, InSig4` = Free interconnectable

For the evaluation up to 4 masks of the type WORD `PstMask1-4` can be applied to these bits (AND logical operation) in accordance with the definition:

Mask `PstMask1`	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
...	-	-	-	-	-	-	-	-	-	-	-	-	1/0	1/0	1/0	1/0
`PstMask4`																
Input signal													`InSig4`	`InSig3`	`LoLimSig`	`HiLimSig`

(The mask bits 4 to 15 are not evaluated)

If the state of the 4 input signals corresponds to one of the bit combinations in the masks `PstMask1-4`, the status configured at the assigned input `Ext1TankSt ... Ext4TankSt` is imported from the enumeration "BlTankStatus" as the process status `PvTankStatus`. If more

than one event is true, the status with the highest priority at the output PvTankStatus is imported:

- PstMask4 and Ext4TankSt = Highest priority
- ...
- PstMask1 and Ext1TankSt = Lowest priority

To ensure smooth transition of the Auto / manual switchover, the PvTankStatus output is tracked to the current status in automatic mode.

Remaining time monitoring for the process status "Cleaned" and "Sterile"

Two monitoring timers (cleanliness and sterility timer) are available for influencing the process status. The following conditions apply for the handling of the two timers:

- The timer setpoints CleanTOp and SterileTOp can be specified in the OS faceplate in the format (Number of days, hr, min) and the MonCleanOp and MonSterileOp enables can be set.
- The timer remaining times CleanRmTAct and SterRmTAct can be followed at the OS faceplate.
- Starting condition 'Cleanliness timer':
  If PvTankStatus corresponds to the configured value at CleanSt (default = 1 = "Cleaned") and the Cleanliness timer has been enabled (MonCleanOp =1), it is started. After the configured time has expired (displayed at the output CleanMonAct) the binary output CleanMonExpired is reset. The timer is reset automatically when the starting condition is no longer fulfilled.
- Starting condition 'Sterile timer':
  If PvTankStatus corresponds to the configured value at SterileSt (default = 2 = "Sterile") and the Sterile timer has been enabled (MonSterileOp =1), it is started. After the configured time has expired (displayed at the output SterMonAct) the binary output SterMonExpired is reset. The timer is reset automatically when the starting condition is no longer fulfilled.
- After the configured time has expired, the respective status is set to the value that is configured at the input parameters UnCleanSt or UnSterileSt. In both cases an alarm message is generated (see the section "Messages of 'Tank' (Page 1172)").

Quality status, output PvQualState

The Tank block provides a quality status management with different states (default = 5). The status definitions are contained in the enumeration "BlQualityState" that is located in the "Shared Declarations" folder of the project. The following states are defined and preset here:

Value	Object name
0	Undefined
1	Not_tested
2	Approved
3	Not_approved
4	Irrelevant

The quality status can be influenced both by the OS faceplate and the block inputs (no preferred operating mode). The operator can change the quality status in manual operation by selecting a value from the list in the OS faceplate. If the edge of the control input `In1QualSt ... In4QualSt` is positive at the block in the AS, the correspondingly configured parameter value of the assigned input `Ext1QualSt ... Ext4QualSt` is imported. In both cases the corresponding status is displayed at the output `PvQualState` .

Quality status, operator prompt

If the current process status at the output `PvTankStatus` matches one of the parameter inputs `TankMon1` or `TankMon2` , an operator prompt message "Take sample" is initiated and the quality status at the output `PvQualState` is set to "1" (in this case: "Not_tested"). The operator prompt is displayed at the output `OpRequest=1` and is reset when the quality status "1" is exited (see the section "Messages of 'Tank' (Page 1172)" for further information).

Quality status, monitoring timer

- The timer setpoint `QualityTOp` can be specified in the OS faceplate in the format (Number of days, hr, min) and the `MonQualityOp` enable can be set.
- The timer remaining time `QualRmTAct` can be followed at the OS faceplate.
- If the quality status "2" (in this case: "Approved") is selected and the configured value at the input `TankMon1` or `TankMon2` matches the current process status, the quality timer is started.
- After the configured time has expired, the quality status at the output `PvQualState` is set to the value `UnQualSt` (default = 0 = "Undefined").
- The timer is reset automatically when one of the status conditions for starting the timer is no longer fulfilled.
- An alarm message is generated when the timer expires (see the section "Messages of 'Tank' (Page 1172)").

Material management:

The tank control block can store a sequence of up to 10 materials, including the batch IDs and quantity.

- The preferred mode can be selected with the input `MatLiOp` .
  `MatLiOp = 0`
  The auto/manual changeover takes place in the OS faceplate.
  `MatLiOp = 1`
  The auto/manual changeover takes place via the block inputs `AutMatLi/ManMatLi`.
- Automatic mode (`AutMatAct = 1`):
  If the input `SetMAtLi` is set (positive signal edge), the values `MatID1, BatchID1, PvVol1` are imported into the FIFO list at the first position. The additional lines are moved downwards by one (outputs `MatID2, BatchID2, PvVol2 ... MatID10, BatchID10, PvVol10`). The last line is lost in the process (FIFO principle).

- Manual operation (`AutMatAct = 0`):
  Material selection is carried out using a list box in the OS faceplate and the corresponding quantity is entered. After confirmation with the Save button, the material record is imported into the FIFO list.
- When the operating mode is switched over from manual to automatic, the current interconnection values `BatchIdLi` and `MatIDLi` are imported into the first line of the list.
- `RstLi` input
  With the positive edge, the entire FIFO list is deleted and the parameter `MatIDRst` is preassigned. "0" is then preassigned to the batch ID for the materials.

  Note

  Note:
  - A material ID in the material archive should be defined as a standard for an empty tank, for example . MatIDRst =1 "No product" or "CO2". This definition is important, because otherwise a check of the material sequence is not possible.
  - After the reset function the current interconnection values `BatchIdLi` and `MatIDLi` are imported into the first line of the list.

- Material model of SIMATIC BATCH:
  This is used in order to select the material names and assign the material IDs in the OS faceplate. The faceplate contains scripts that can read the information from the Batch Control Server (BCS). To this purpose it is necessary that the BCS is running and that a BATCH project that contains the material definitions is loaded
- The following conditions and limitation for material definitions in SIMATIC BATCH have to be observed if the Tank block is to use these in combination with SIMATC BATCH :
  - The material ID inputs / outputs always have the data type DINT, meaning that the SIMATIC BATCH material code may only contain numbers (0 <= Value < 2,147,483,647).
  - The BATCH material quality is not processed by the Tank block and is thus not considered. It can therefore be used normally.
  - In order to use the material IDs together with SIMATIC Route Control the SIMATIC BATCH material code has to be limited to the range 0 to 1024. A material code that exceed 1024, can be processed by the Tank block, but cannot be used together with SIMATIC Route Control .
  - In view of the limited width available at the list field, the material names should be shorter than 20 characters to ensure that the material name is displayed completely.
- The following conversion function blocks are provided by the library to convert the material code into the material ID and vice versa (connection of the SFC PI/PO parameter with Tank).
  - MidStr: Material conversion DINT to string value (see F1 help for MidStr)
    Interconnect Tank output `MatID<x>` with MidStr input `MatID` for this purpose.
  - StrMid: Material conversion string value to DINT (see F1 help for StrMid)
    Interconnect StrMid output `MatID` with BlTank input `MatIDLi` for this purpose.

Overwrite functions:

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwManOpEn`
- `OvwModSel`
- `OvwUnitID`
- `OvwBatchInfo`

You will find more information on this subject in the section Higher-level controlling of the unit / EQM level (Page 70)

Specifying the display area for process and setpoint values as well as operations

This block features the APL standard function "Display and operator input area for process values and setpoints". The following input parameters are available for this step:

- `Temp_OpScale` Limit values for temperature trend display in faceplate
- `Press_OpScale` Limit values for pressure trend display in faceplate
- `Level_OpScale` Limit values for fill level trend display in faceplate

Calling additional faceplates

This block includes the APL standard function "Call additional faceplates".

Forming the signal status for blocks

This block includes the APL standard function "Form and output signal status for blocks".

The worst signal status for the block is formed from the following parameters:

- `SpVolLi`

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in the "APL documentation" in the section "Configurable functions with the Feature I/O". The following reactions are available for this block at the relevant bits

Feature bits

Bit	Comment
0	See APL function: "Set startup response" 1 = Start-up without call in OB100 0 = Start-up with call in OB100 (default): • During the CPU start-up the messages are suppressed for a number of cycles configured in `RunUpCyc` (default is 3). • The material data `MatID1 ... MatID10` have the parameter value `MatIDRst` pre-assigned to them and the values `BatchID1 ... BatchID10` and `PvVol1 ... PvVol10` have "0" pre-assigned.
1	See APL function: "Reaction to the Out of service mode" Enable switchover to "Out of service" via `OosLi` (default = 0)
...	Reserve
22	See APL function: "Update acknowledgment and error status of the message call"
...	Reserve
24	See APL function: "Activate local operator permission" 1 = "Local" operator permission is active
...	Reserve

Operator permissions

This block includes the APL standard function "Operator permissions". The block has the following permissions for the `OS_Perm` parameter.

Status bit	Parameter
0	1= Operator can switch to "AUTO" mode for tank status
1	1= Operator can switch to "MANUAL" mode for tank status
2	1 = Operator can switch to "Out of service" mode
3	1= Operator can switch to "AUTO" mode for material management
4	1= Operator can switch to "MANUAL" mode for material management
5	1 = Operator can activate the release for maintenance function
6	1 = Operator can reset the tank values
7	1= Operator can change the monitoring time for quality status
8	1= Operator can change the monitoring time for cleaning status
9	1= Operator can change the monitoring time for sterility status
10	1 = Operator can activate the function monitoring (Bits 6 & 7 & 8)
...	Reserve

Overview of error numbers

The `ErrorNum` connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	No active fault

Error number	Meaning of the error number
51	If AUTO and MANUAL requests for tank status or material management are set simultaneously by the operator
55	If AUTO and MANUAL requests for tank status or material management are set simultaneously at the interconnection inputs or the `CmData` structure.

22.3.4 Messages of 'Tank'

Messages of 'Tank'

The block uses an instance of the ALARM_8P block (MsgEvId). The following messages can be generated:

Message instance	Message identifier	Message class	Event
MsgEvId	SIG 1	Alarm high	$$BlockComment$$ Quality time expired
	SIG 2	Alarm high	$$BlockComment$$ Cleaning time expired
	SIG 3	Alarm high	$$BlockComment$$ Sterile time expired
	SIG 4	Operating message – without acknowledgment	$$BlockComment$$ Tank status change from '@4i%t#ILTankStatus@' to '@5i%t#ILTankStatus@'
	SIG 5	Operating message – without acknowledgment	$$BlockComment$$ Quality change from '@6i%t#ILQualityState@' to '@7i%t#ILQualityState@'
	SIG 6	Operator prompt – general	$$BlockComment$$ Take sample
	SIG 7	AS process control message - fault	$$BlockComment$$ External message 7
	SIG 8	AS process control message - fault	$$BlockComment$$ External error 8

Explanation:

- $$BlockComment$$ → Content of the instance-specific comment
- @...ILTankStatus@ → Generates the old / new tank status identifier from the text library

- @...ILQualityState@ → Generates the old / new quality status identifier from the text library

Note

The message texts for the tank or quality status are determined from the message associated values 4 and 6 from the supplied text libraries ('ILTankStatus' and 'ILQualityState'). In the case of user changes to the corresponding enumerations these text libraries also have to be adapted (per language de, en, sp)

- You can interconnect an external fault to the input parameter CSF. If this signal changes to `CSF = 1`, a fault is triggered (MsgEvId, SIG 8).
- In addition, an external message SIG 7 can be generated through the input parameter `ExtMsg7`.
- Message generation is prevented with the input signal `MsgLock` or `CmData.xMsgLock = 1`.

Associated values for reporting instance MsgEvId

Associated value	Block parameter
1	`BatchName`
2	`StepNo`
3	`BatchID`
4	`snOldTankState` (previous cycle)
5	`PvTankStatus`
6	`snOldQualityState` (previous cycle)
7	`PvQualState`
8	`UnitIDOut`

22.3.5 Connections of 'Tank'

Input parameters

Parameter	Description	Type	Default
`TankLocId`	Tank and location ID	`DINT`	`0`
`TankGrp`	Tank group	`INT`	`0`
`TankType`	Tank type	`INT`	`0`
`OosLi`	1=Oos mode: Oos mode switch via link	`DigVal`	
`OosOp`	1=Oos mode: Oos mode by operator	`BOOL`	
`RstLi`	Linked reset signal	`DigVal`	
`RstOp`	Operator reset signal	`BOOL`	
`TankStLi`	Setpoint for tank state in automatic mode	`INT`	
`TankStOp`	Setpoint for tank state in manual mode	`INT`	

Parameter	Description	Type	Default
HiLimSig	High level signal (mask value = 1)	DigVal	
LoLimSig	Low level signal (mask value = 2)	DigVal	
InSig3	External Bit 3 (mask value = 4)	DigVal	
InSig4	External Bit 4 (mask value = 8)	DigVal	
PStMask1	Bitmask for Ext1TankSt take-over	WORD	1
PStMask2	Bitmask for Ext2TankSt take-over	WORD	2
PStMask3	Bitmask for Ext2TankSt take-over	WORD	0
PStMask4	Bitmask for Ext2TankSt take-over	WORD	0
Ext1TankSt	Status to set when PStMask1 matches	INT	4
Ext2TankSt	Status to set when PStMask2 matches	INT	8
Ext3TankSt	Status to set when PStMask3 matches	INT	0
Ext4TankSt	Status to set when PStMask4 matches	INT	0
TankMon1	Process status to check for (signals operator request or starts quality timer)	INT	4
TankMon2	Process status to check for (signals operator request or starts quality timer)	INT	5
In1QualSt	External signal for quality status change to Ext1QualSt	DigVal	
In2QualSt	External signal for quality status change to Ext2QualSt	DigVal	
In3QualSt	External signal for quality status change to Ext3QualSt	DigVal	
In4QualSt	External signal for quality status change to Ext4QualSt	DigVal	
Ext1QualSt	Status for signal InQualS1	INT	1
Ext2QualSt	Status for signal InQualS2	INT	2
Ext3QualSt	Status for signal InQualS3	INT	3
Ext4QualSt	Status for signal InQualS4	INT	4
UnQualSt	Status after Quality Monitoring Time is up	INT	0
CleanSt	Status number for Clean Monitoring	INT	1
UnCleanSt	Status after Clean Monitoring Time is up	INT	0
SterileSt	Status number for Sterile Monitoring	INT	2
UnSterileSt	Status after Sterile Monitoring Time is up	INT	0
TempUnit	Engineering units of temperature value	INT	0
Temperature	Temperature for Trend	REAL	0.0
Temp_OpScale	Limit values for scale in Temperatur bar display of faceplate	STRUCT • High REALLow REAL	100.00.0
PressUnit	Engineering units of pressure value	INT	0
Pressure	Pressure for Trend	REAL	0.0

Parameter	Description	Type	Default
Press_OpScale	Limit values for scale in Pressure bar display of faceplate	STRUCT • High REALLow REAL	100.00.0
LevelUnit	Engineering units of level value	INT	0
Level	Level for Trend	REAL	0.0
Level_OpScale	Limit values for scale in Level bar display of faceplate	STRUCT • High REALLow REAL	100.00.0
QualityTOp	Quality State monitoring time setpoint	DINT	3600
CleanTOp	Clean State monitoring time setpoint	DINT	3600
SterileTOp	Steril State monitoring time setpoint	DINT	3600
MonQualityOp	Quality: 1=Monitoring ON, 0=Monitoring OFF	BOOL	False
MonCleanOp	Clean: 1=Monitoring ON, 0=Monitoring OFF	BOOL	False
MonSterileOp	Sterile: 1=Monitoring ON, 0=Monitoring OFF	BOOL	False
StLiOp	1=Link, 0=Manual: Input to auto/manual commands for tank status	BOOL	False
AutStOp	1=Auto mode: auto mode switch by operator for tank status	BOOL	
ManStOp	1=Manual mode: manual mode switch by operator for tank status	BOOL	True
AutStLi	1=Auto mode: auto mode switch via link for tank status	DigVal	
ManStLi	1=Manual mode: manual mode switch via link for tank status	DigVal	
QualStOp	Quality State operator input	INT	0
MatLiOp	1=Link, 0=Manual: Input to auto/manual commands for material	BOOL	False
AutMatOp	1=Auto mode: auto mode switch by operator for material	BOOL	
ManMatOp	1=Manual mode: manual mode switch by operator for material	BOOL	True
AutMatLi	1=Auto mode: auto mode switch via link for material	DigVal	
ManMatLi	1=Manual mode: manual mode switch via link for material	DigVal	
MatIDRst	Material ID to reset to on startup or manual reset	DINT	0
BatchIdOp	Operator Batch ID	DWORD	0
MatIDOp	Operator value for material ID	DINT	0
MatIDLi	Linked value for material ID	DINT	0
SpUnit	Engineering units of setpoint value	INT	0
SpVolOp	Operator amount setpoint value	REAL	0.0
SpVolLi	Linked amount setpoint value	AnaValFF	
SetMatOp	Operator set material values	BOOL	

Parameter	Description	Type	Default
RstMatOp	Operator reset material values	BOOL	
SetMatLi	Linked input to set material values	DigVal	
CmData	Data-Struct	BlUdtCmData	
ManOpEn	Enable switchover to manual mode	BOOL	
OvwManOpEn	1=Overwrite CmData ManOpEn with input ManOpEn	BOOL	
OvwModSel	1=Overwrite CmData mode switches	BOOL	
OvwUnitID	1=Overwrite CmData.UnitID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
UnitID	Unit ID (overwritable CmData element)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIdLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
MS_RelOp	Operator input for MS release, 1: MS release requirement	BOOL	
RunUpCyc	Number of cycles for which this block keeps in start up mode	INT	3
SampleTime	Sampling time [s]	REAL	1.0
UserStatus	User status bits	BYTE	
CSF	Control System Fault message - external error	BOOL	
MsgLock	Message lock	DigVal	
MsgEvId	Message event ID 01	DWORD	16#FFFFFFFF
MsgMask	Acknowledgement mask (matched for OutAck)	WORD	16#33
ExtMsg7	External message 7	BOOL	
SelFp1	Open faceplate of connected block (type)	BOOL	
SelFp2	Open faceplate of connected block (type)	BOOL	
OS_Perm	Operator Permission	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

Output parameters

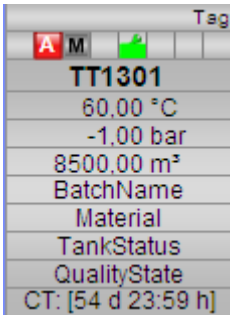
Parameter	Description	Type	Default
MS_Release	1: Maintenance status release	DigVal	
PvTankStatus	Output tank status	INT	0
PvQualState	Output quality state	INT	0
QualMonAct	Quality State monitoring time actual value	DINT	0
CleanMonAct	Clean State monitoring time actual value	DINT	0
SterMonAct	Steril State monitoring time actual value	DINT	0
QualMonActR	Quality State monitoring time actual value real	REAL	0.0

Parameter	Description	Type	Default
CleanMonActR	Clean State monitoring time actual value real	REAL	0.0
SterMonActR	Steril State monitoring time actual value real	REAL	0.0
QualRmTAct	Quality State monitoring time remaining value	DINT	0
CleanRmTAct	Clean State monitoring time remaining value	DINT	0
SterRmTAct	Steril State monitoring time remaining value	DINT	0
QualMonStarted	Quality State monitoring time started	BOOL	False
CleanMonStarted	Clean State monitoring time started	BOOL	False
SterMonStarted	Steril State monitoring time started	BOOL	False
QualMonExpired	Quality State monitoring time expired	BOOL	False
CleanMonExpired	Clean State monitoring time expired	BOOL	False
SterMonExpired	Steril State monitoring time expired	BOOL	False
MatInfo	Material info collection	BlUdtMaterialInfo	
QualSt0…QualSt3	Quality state flag 0 ... 3	BOOL	
OpRequest	1=Operator Request	BOOL	False
OpError	1=Operator Error	BOOL	False
StLiOpOut	1=Link, 0=Manual: Input to auto/manual commands for tank status	BOOL	False
AutStAct	1=Automatic mode for tank status is active	DigVal	
ManStAct	1=Manual mode for tank status is active	DigValTrue	
MatLiOpOut	1=Link, 0=Manual: Input to auto/manual commands for material	BOOL	False
AutMatAct	1=Automatic mode for material is active	DigVal	
ManMatAct	1=Manual mode for material is active	DigValTrue	
OosAct	Out of service is active	DigVal	
HiLimSigOut	High level output signal (mask value = 1)	DigVal	
LoLimSigOut	Low level output signal (mask value = 2)	DigVal	
InSig3Out	External Bit 3 output signal (mask value = 4)	DigVal	
InSig4Out	External Bit 4 output signal (mask value = 8)	DigVal	
OpEnStOut	1=Enable input tank status	BOOL	False
OpEnMatOut	1=Enable input material	BOOL	False
MatIDAct	Actual Material ID	DINT	0
PvVolAct	Actual Volume	AnaValFF	
BatchID1 ... BatchID10	Batch1 ID ... Batch10 ID	DWORD	0
MatID1 ... MatID10	Material1 ID ... Material10 ID	DINT	0
PvVol1 ... PvVol10	Volume 1 ... Volume 10	REAL	0.0
TotalVolAct	Total material amounts	AnaValFF	

Parameter	Description	Type	Default
MatMovements	Material movements	INT	0
UnitIDOut	Unit ID	DWORD	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING32	
StepNo	Batch step number	DWORD	
MsgErr	ALARM_8P: Messaging error	BOOL	
MsgStat	ALARM_8P: Status output	WORD	
MsgAckn	ALARM_8P: ACK_STATE output	WORD	
ErrorNum	Error number	INT	-1
OS_PermOut	Operator permission (OS info)	DWORD	16#FFFFFFFF
OS_PermLog	Operator permission (OS info)	DWORD	16#FFFFFFFF
OS_PermLnk	Operator permission (OS info)	DWORD	
OpSt_Out	Enabled operator stations	DWORD	
Status1	Status1 data	DWORD	
ST_Worst	Worst signal status	BYTE	16#80

22.3.6 Operator control and monitoring

22.3.6.1 Symbol



Block instance name
Status bar
Tank name (from enumeration)
Tank temperature | Unit
Tank pressure | Unit
Tank level | Unit
Batch name
Material
Tank process status (from enumeration)
Tank quality status (from enumeration)
Remaining time of 'CleanTimer' or 'SterileTimer' depending on which timer is running

The status bar contains the following symbol information sequentially from left to right:

Position	Content	
1	Symbol dynamic / static	
	D	Dynamic interconnection active

Position	Content	
2	Message status *)	
3	Operating mode	
		Automatic
		Manual
		"Out of service"
4	Empty	
5		Enable maintenance operation
6	Empty	
7	Empty	
8		Memo is available
*) Additional details about the "Message status" are available in the "Block icon structure" section of the APL documentation.		

Applications of the block icon

The icon of the block 'Tank' can be used as follows:

- **Standard**
  Display of a defined tank instance.
  The connection with the corresponding block instance is carried out automatically through the AS-OS transfer.
- **Dynamic**
  In addition, the icon can be installed, for example, manually into a transfer process picture and parameterized in order to visualize the other tank(s) involved in a transfer batch (for example 1x source tank, 8x destination tank). The following attributes have to be parameterized in the WinCC Graphics Designer to this purpose:
  - Configurations\EnableDynamicTankID = YES
  - Configurations\DynamicTankID
    Interconnection using dynamic dialog to a block parameter of the type Location ID which specifies the ID number of the tank to be displayed during the runtime.
    (Example: Parameter `NxtSrc / NxtDest` of the block Qing when the transfer control is taken from it.)

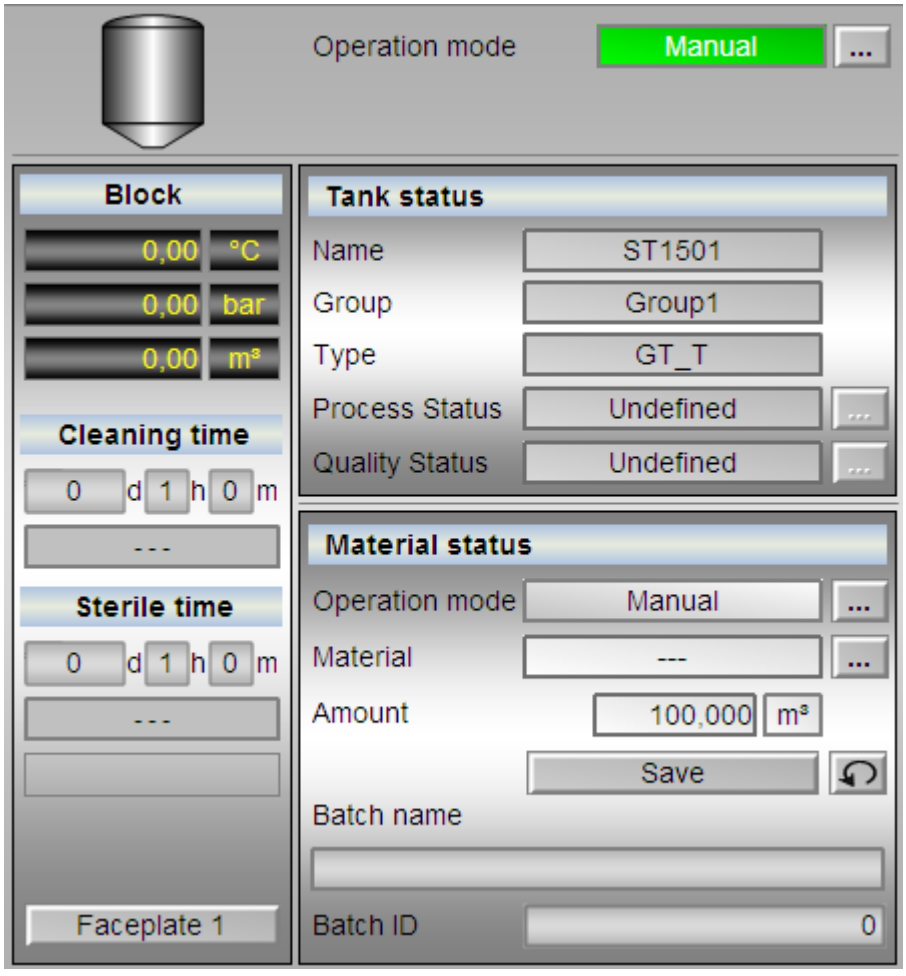
22.3.6.2 Faceplate

Views of the Tank block

The block provides the following views.

	Standard view	See below.
	Message view	Standard message view of the Advanced Process Library
	Batch history	See below.
	Monitoring times	See below.
	Trend view	Standard trend view of all the blocks of the Advanced Process Library. The following measured values are displayed here: • Tank temperature from block parameter `Temperature` • Pressure from block parameter `Pressure` • Level from block parameter `Level`
	Preview	See below.
	Memo view	Standard memo view of the Advanced Process Library
	Batch view	Standard batch view of the Advanced Process Library

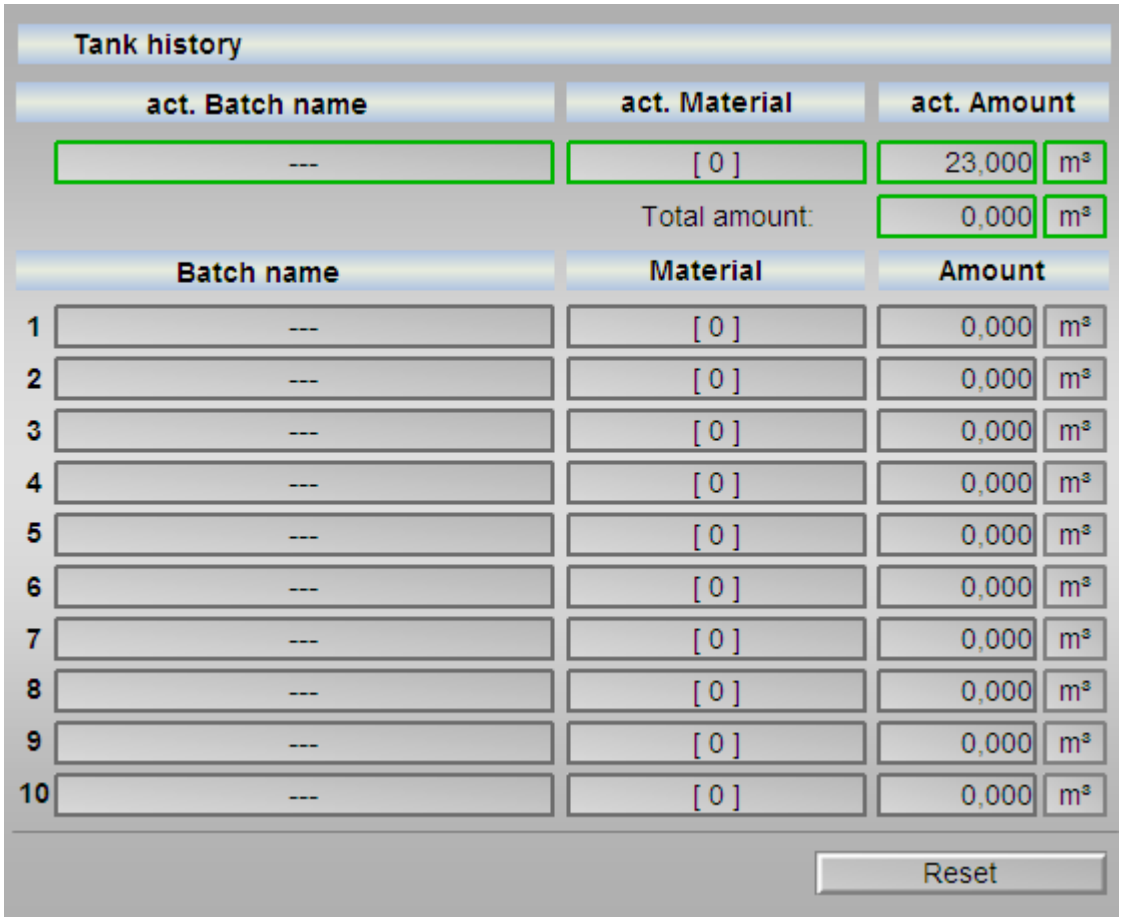
Standard view of the 'Tank'



Explanation of the screen shot	
(1)	**Displaying and switching the operating mode**
	This area provides information on the currently valid operating mode. The following operating modes can be shown and executed here: • Automatic • Manual • Out of service

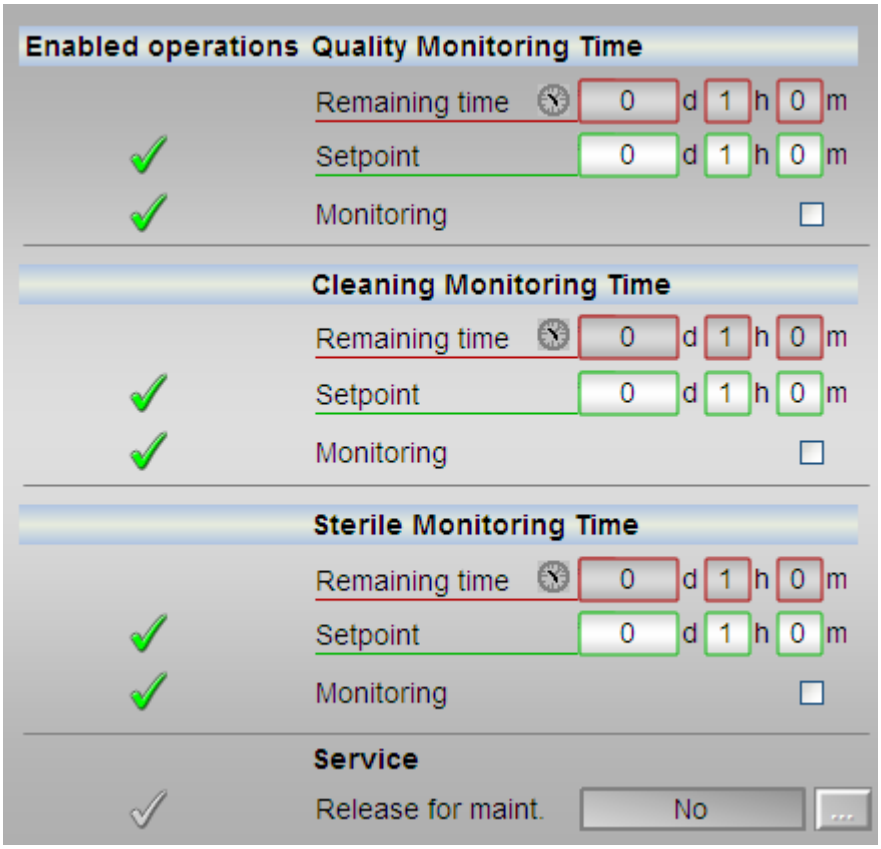
Explanation of the screen shot			
(2)	**Displaying and switching the tank status parameters**		
	• The following tank parameters can be shown here:		
		Name	Tank name of block parameter `TankLocId`
		Group	Tank group from block parameter `TankGrp`
		Type	Tank type from block parameter `TankType`
	• The following tank status mode can be shown and executed here in manual mode:		
		Process status	Current tank process status from block parameter `PvTankStatus`
		Material status	Current tank quality status from block parameter `PvQualState`
	• In Manual operating mode the new value can be selected from a list through the additional dialog		
		Process status	List from enumeration `BlTankStatus` from block parameter `PvTankStatus`
		Material status	List from enumeration `BlQualityState`from block parameter `PvQualState`
(3)	**Displaying and switching the material status parameters**		
	• Operating mode	Display or specification of the material tracking operating mode	
	• Material	Display or specification (Manual mode) of the last material transfer in the tank. The list of the available materials is determined from SIMATIC BATCH .	
	• Quantity	Setpoint of the active material transfer	
	• Batch name	Name of an active transfer batch	
	• Batch ID	ID number of an active transfer batch	
	Note: Material selection is carried out using a list box in the OS faceplate and the corresponding quantity is entered. After confirmation with the Save button, the material record is imported into the FIFO list. The value of the input `BatchIdLi` is pre-assigned to the batch ID.		
(4)	Displaying further tank parameters		
	• Tank temperature from block parameter `Temperature` • Pressure from block parameter `Pressure` • Level from block parameter `Level`		
(5)	Remaining time "Cleaned"		
	During activated monitoring this field shown the remaining time for the process status "Cleaned" (see the section Functions of 'Tank' (Page 1165)). The field below it contains the start time of the process status "Cleaned". After the monitoring time has expired, the field for the remaining time is colored red.		
(6)	Remaining time "Sterility"		
	This field shown the remaining time for the process status "Sterile" (see the section Functions of 'Tank' (Page 1165)). The field below it contains the start time of the process status "Sterile". After the monitoring time has expired, the field for the remaining time is colored red.		
(7)	Navigation button to the standard view of any faceplate		
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp1` in the CFC.		

Batch history of Tank



Explanation of the screen shot	
(1)	**Parameters of the current batch and total quantity**
	The following values are displayed here from left to right: • Name of the running batch • Material of the running batch • Setpoint quantity of the running batch Total field for the quantity. The total of the material transfers since the last rest is displayed here. This can deviate from the last 10 transfer quantities displayed below.
(2)	**History of the up to 10 previous transfer batches**
	The parameters of the last 10 transfer batches are displayed here: • Name of the batch • Material of the batch • Setpoint quantity of the batch

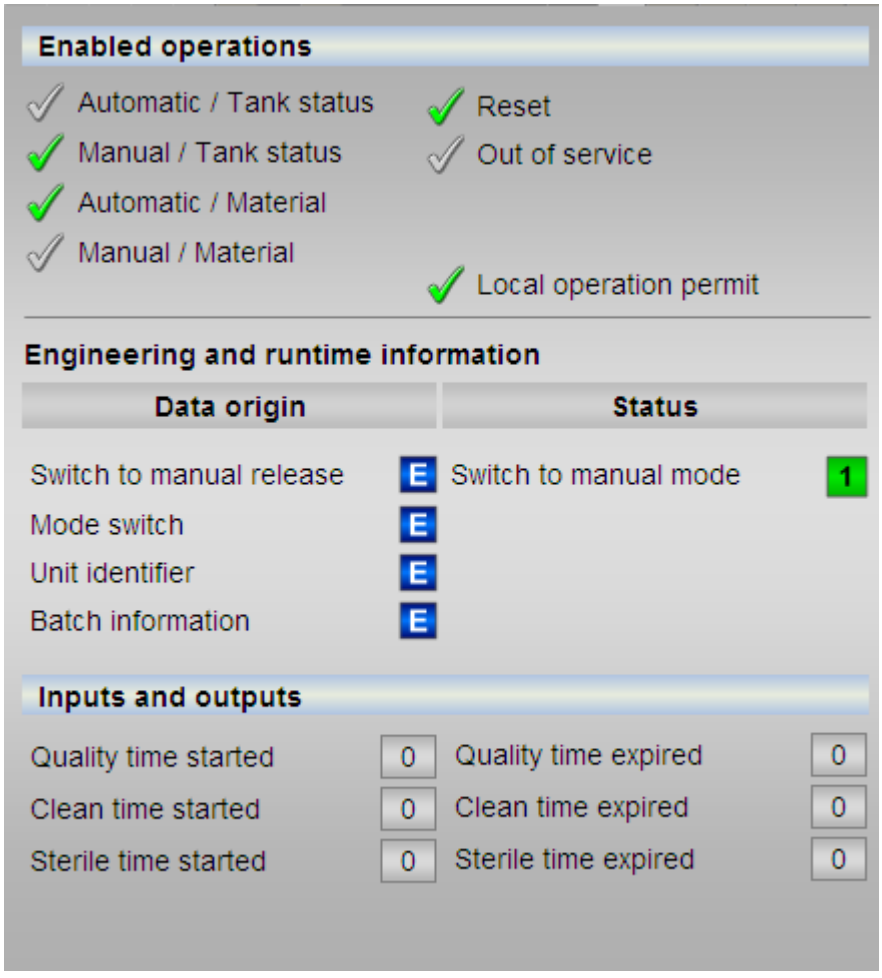
Monitoring times of Tank



The display and activation of the monitoring times for the "Quality status" as well as for the process status "Cleaned" and "Sterile" are made in this view. The display fields are interlinked with the corresponding block parameters:

Explanation of the screen shot	
(1)	**Quality time monitoring**
	• Remaining time → `QualRmTAct` • Target time → `QualityTOp` • Monitoring (enable) → `MonQualityOp`
(2)	**Cleaning monitoring time**
	• Remaining time → `CleanRmTAct` • Target time → `CleanTOp` • Monitoring (enable) → `MonCleanOp`
(3)	**Sterile monitoring time**
	• Remaining time → `SterRmTAct` • Target time → `SterileTOp` • Monitoring (enable) → `MonSterileOp`
(4)	**Service**
	In this section the release for maintenance function can be activated / deactivated if the corresponding operator control enable exists.

Preview of Tank



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator permissions. They depend on the configuration of the `OS_Perm` parameter of this block. • The operator control enables shown in the screenshot are possible here. Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions
(2)	**Engineering and runtime information**

Explanation of the screen shot			
	This section contains information from the status word `Status1` • The "Data origin" column shows the setting of the overwrite signals.		
	Enable manual mode	`OvwManOpEn`	
	Operating mode switchover	`OvwModSel`	
	Unit identifier	`OvwUnitID`	
	Batch information	`OvwBatchInfo`	
	• The symbols have the following meaning		
		Signal origin = Unit / `CmData` structure	
		Signal origin = Block input `Ovw<xxx>`	
	• 'States' column displays the state of the input parameter `ManOpEn` (0 / 1).		
(3)	**Inputs and outputs**		
	The status of the following output bits is displayed in this section.		
	Identifier	Parameter "started"	Parameter "expired"
	Quality time	`QualMonStarted`	`QualMonExpired`
	Cleaning time	`CleanMonStarted`	`CleanMonExpired`
	Sterile time	`SterMonStarted`	`SterMonExpired`
(4)	**Navigation button to the standard view of any faceplate**		
	This navigation button is used to reach the standard view of a block connected at the block input `SelFp2` in the CFC.		

22.4 TcCtrl – Cooling zones control block

22.4.1 Description of 'TcCtrl'

Area of application and operating principle

This function block controls the cooling zones (solenoid valves) of a tank in cooperation, for example, with an SFC dependent on the level and the preselected temperature zones. The program parameters (target temperature, duration, ramp mode) as well as the temperature sensors and cooling zone solenoid valves (each with a maximum of 8 units) including their assignment can be specified by means of interconnection ('Automatic' operation mode) or by the operator ('Manual' operating mode). Messages can be generated for various error states and upper or lower limit value violations.

Configuration

Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). In addition the configuration block 'TcCfg' (see section 'TcCfg – Cooling zones configuration block (Page 1220)") that allows the configuration of the up to 8 temperature sensors and cooling zone solenoid valves is required.

Status word allocation 'Status1'

For a description of the individual parameters, see the section "Connections of 'TcCtrl' (Page 1198)".

Status bit	Parameter
0	`BatchOcc`
1	`BatchEn`
2	`OosAct.value`
3	`OosLi.value`
4	`AutAct.value`
5	`ManAct.value`
6	1 = Cooling program has been started (automatic or manual)
7	Done 1 = Cooling program has terminated, no error
8	WaitTs 1 = Waiting until temperature sensor is available
9	WaitCz 1 = Waiting until cooling zone is available
10	ErrCfg 1 = Error in the configuration data (at the block 'TcCfg')
11	ErrPsTs 1 = Error in the preselection of the temperature sensors
12	ErrPsCz 1 = Error in the preselection of the cooling zones

Status bit	Parameter
13	ErrProg 1 = Error: Invalid cooling program
14	GrpErr 1 = Group error
15	NOT GrpErr 1 = No error
16	ErrHiT 1 = Error temperature too high
17	ErrLoT 1 = Error temperature too low
18	1 = Cooling start command in automatic operating mode
19	1 = Cooling stop command in automatic operating mode
...	Reserve
30	1 = Message lock
31	1 = Release for maintenance

Status word allocation 'Status2'

Status bit	Parameter
0	`CoolEn` 1 = Cooling program is enabled
1	`CoolProg = COOLPROG_INT_TOPDOWN` 1 = Cooling program internal from top downward
2	`CoolProg = COOLPROG_INT_BTMUP` 1 = Cooling program internal from bottom upward
3	`CoolProg = COOLPROG_INT_USEALL` 1 = Cooling program internal with all cooling zones
4	`CoolProg = COOLPROG_EXT_AVGT` 1 = Cooling program external with all cooling zones until sensor mean value reached
5	`CoolProg = COOLPROG_EXT_MULTIT` 1 = Cooling program external with sensor-specific cooling zones and temperature setpoint per sensor
6	`CoolFuncOut` 1 = Cooling / Heating; 0 = Keep temperature
7	`NOT CoolFuncOut` 1 = Keep temperature; 0 = Cooling / Heating
8	`CoolState = TCCS_IDLE` 1 = Waiting state
9	`CoolState = TCCS_STARTING` 1 = Starting state
10	`CoolState = TCCS_RUNNING` 1 = Running state
11	Inverted Status bit 08
...	Reserve
16	Solenoid valve V08 opened

Status bit	Parameter
17	Solenoid valve V07 opened
18	Solenoid valve V06 opened
19	Solenoid valve V05 opened
20	Solenoid valve V04 opened
21	Solenoid valve V03 opened
22	Solenoid valve V02 opened
23	Solenoid valve V01 opened
24	Solenoid valve V08 locked
25	Solenoid valve V07 locked
26	Solenoid valve V06 locked
27	Solenoid valve V05 locked
28	Solenoid valve V04 locked
29	Solenoid valve V03 locked
30	Solenoid valve V02 locked
31	Solenoid valve V01 locked

22.4.2 Operating modes of 'TcCtrl'

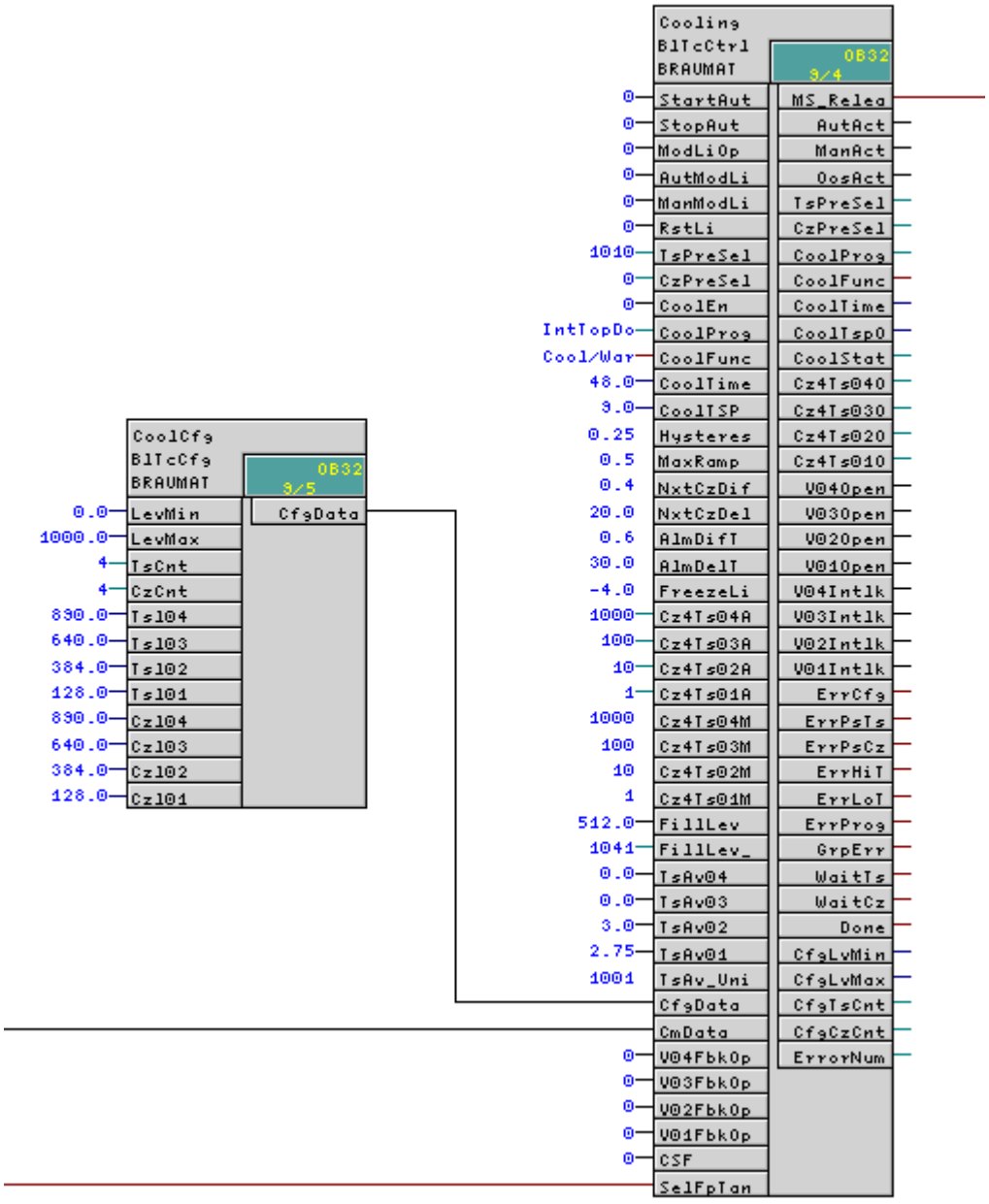
The block can be operated using the following operating modes:

- "Automatic" → The cooling program operates on the basis of the interconnected program parameters
- "Manual" → The cooling program operates on the basis of the operable program parameters
- "Out of service" → The cooling program is inactive, the cooling zone valves are deactivated.

General information on these operating modes is available in the "Operating modes of the blocks" section of the APL documentation.

22.4.3 Functions of 'TcCtrl'

Interconnection principle



The example shows the parameter block 'TcCfg' and the tank cooling block 'TcCtrl'. The inputs and outputs each for 4 sensors are connected visibly.

Basic parameters

TcCtrl controls the cooling zones of a tank on the basis of the following program parameters that can be specified as required through interconnection or by the operator (Operating mode = "Manual" and State = "Idle").

Block function	'Automatic' operating mode Parameter interconnection	'Manual' operating mode Parameter operation
Binary enable signal for the cooling program	CoolEn	<Not relevant>
Target temperature of the complete tank	CoolTSP	CoolTSpMan
Duration of the cooling program	CoolTime	CoolTimeMan
Specification of the cooling course: • Keep temperature • Let cool or heat up	CoolFunc	CfKeepManCfCoolMan
Specification of the cooling program: • Internal: From top downward (fermentation) • Internal: From bottom upward (maturation) • Internal: Use all cooling zones (fast cooling) • External: Cooling zone specification by user	CoolProg (enum 'BlTcPrograms')	CpIntTopDownMan CpIntBtmUpMan CpIntUseAllMan CpExtAvrgTMan CpExtMultiTMan
Specification of the temperature sensors to be used	TsPreSel	TsPreSelMan
Specification of the cooling zones to be used (only relevant for external cooling program)	CzPreSel	CzPreSelMan
Assignments between temperature sensors and cooling zones ("Cooling zones for temperature sensor")	Cz4Ts08Aut … Cz4Ts01Aut	Cz4Ts08Man … Cz4Ts01Man
Starting and stopping the cooling program.	StartAutStopAut	StartManStopMan
Commands for resetting, out of service	RstLi, OosLi	RstOp, OosOp

Control parameters

The following control parameters are available solely in the faceplate (change possible at any time).

Block function	Parameter
Switching hysteresis for the control valves	Hysteresis
Maximum permitted rate of rise of the cooling ramp	MaxRamp
Temperature difference between actual value and setpoint that results in the request for a further cooling zone.	NxtCzDifT
Delay time for the activation of the next cooling zone	NxtCzDelT

Block function	Parameter
Temperature difference between actual value and setpoint that results in an error message.	AlmDifT
Delay time for the generation of an error message	AlmDelT
Monitoring of the cooling course ramp	MonRamp
Monitoring of the temperature differences	MonDifT
Freeze monitoring	FreezeProt

Operating state / Output parameter ‚CoolState':

The cooling program is always in one of the following states:

Block function	Description
CoolState = 'Idle'	The cooling program is inactive, the cooling zone valves are deactivated. The program parameters can be changed. Switchover to 'Starting' through the start command
CoolState = 'Starting'	The program parameters are imported. If there are no errors, the system changes to 'Running'. Switchover to 'Idle' through the stop command
CoolState = 'Running'	The cooling program is active. Switchover to 'Idle' through the stop command

Cooling program / Output parameter ‚CoolProgOut':

A distinction is made between internal and external cooling programs:

- Internal programs:
  - In internal cooling programs use of the cooling zones is carried out autonomously in the block. The average temperature across the specified temperature sensors is used as the setpoint. The programs differ in the sequence of the cooling zone activation.
  - Preselection of the cooling zones is carried out in the faceplate (manual mode) or through interconnection at the input parameter CzPreSel.
  - Preselection of the temperature sensors is carried out in the faceplate (manual mode) or through interconnection at the input parameter TsPreSel (automatic).
  - Assignment of the cooling zones is carried out in the faceplate (manual mode) or through interconnection at the input parameters Cz4Ts08Aut … Cz4Ts01Aut (automatic).

Program Enum 'BlTcPrograms'	Temperature specification	Behavior	Application (example)
CoolProgOut = 1 ,IntTopDown'	CoolTSP	The cooling zones are regulated to the specified temperature from top to bottom.	Fermentation
CoolProgOut = 2 ,IntBtmUp'	CoolTSP	The cooling zones are regulated to the specified temperature from bottom to top.	Maturation
CoolProgOut = 3 ,IntUseAll'	CoolTSP	All the available cooling zones are regulated to the specified temperature at the same time	Fast cooling

- External programs:

Program Enum 'BlTcPrograms'	Temperature specification	Behavior
CoolProgOut = 4 ,ExtAvrgT'	CoolTSP	The preselected cooling zones are always activated simultaneously until the mean value of the selected sensors corresponds to the temperature setpoint value. Preselection of the cooling zones is carried out in the faceplate (manual mode) or through interconnection at the input parameter CzPreSel.
CoolProgOut = 5 ,ExtMultiT'	TsSp08Man … TsSp01Man TsSp08Aut … TsSp01Aut	A separate temperature setpoint is specified for each temperature sensor. The assigned cooling zones are regulated individually to the specified temperature. Assignment of the cooling zones is carried out in the faceplate (manual mode) or through interconnection at the input parameters Cz4Ts08Aut … Cz4Ts01Aut.

Cooling course output parameter 'CoolFuncOut'

The course of the cooling function can be specified as follows using input parameters:

Cooling course	Parameter automatic	Parameter manual
Cooling down with time target (gradient) then hold temperature	CoolFunc = ,Cool' CoolTime, CoolTSP	CfCoolMan = 1 CoolTimeMan, CoolTSpMan
Cooling down with max. gradient then hold temperature	CoolFunc = ,Keep' CoolTSP	CfKeepMan = 1 CoolTSpMan

Freeze monitoring:

If the temperature measured at a sensor falls below the configured lower limit for freeze protection, the cooling zones assigned to the sensor are deactivated and their locking outputs activated.

Freeze monitoring is active unless the block is in the "Out of service" state. The method of operation of the monitoring depends on the control state:

- Freeze monitoring in "Idle" state
  Because there is no reliable assignment between the temperature sensors and the cooling zone valves, all the valves are interlocked as soon as any sensor indicates violation of the low limit.
- Freeze monitoring in "Running" state
  Only those valves whose assigned temperature sensor indicates that the low limit has been violated are interlocked.

Cooling zone locking:

In addition to the control outputs for the cooling zone valves `V08Open ... V01Open`, the block makes the locking outputs `V08Intlk ... V01Intlk` available with positive logic (1 = Good case). Interlock takes place when

- The freeze monitoring indicates this (depending on the operating state, see above) or
- The cooling zone is not available, i.e. the threshold for the level has not been reached and the enable signal `TsRel08 ... TsRel01` (wet sensor) is missing.
  This interlock takes place independent of the operating state of the block.

Overwrite functions:

In this block the following overwrite functions are implemented with regard to the input selections:

- `OvwUnitID`
- `OvwBatchInfo`
- `OvwManOpEn`

You will find more information on this subject in the section Higher-level controlling of the unit / EQM level (Page 70).

Specifying the display area for process and setpoint values as well as operations

This block features the APL standard function "Display and operator input area for process values and setpoints". The following input parameter is available for this step:

- `TsAv_OpScale` Limit values for temperature trend display in faceplate

Calling additional faceplates

This block includes the APL standard function "Call additional faceplates" (input parameters `SelFp1, SelFp2`). In addition the input parameter `SelFpTank` is available. This can be interconnected to any output signal of the type BOOL of the block '`Tank`'.

Forming the signal status for blocks

This block includes the APL standard function "Form and output signal status for blocks". The worst signal status for the block is formed from the following parameters:

- `TsAv08 … TsAv01`
- `TsRel08 … TsRel08`
- `FillLev`
- `CzlRel08 … CzlRel08`

Configurable reactions using the `Feature` parameter

An overview of all the reactions that are provided by the `Feature` parameter is available in the "APL documentation" in the section "Configurable functions with the Feature I/O". The following functionality is available for this block at the relevant bits:

Bit	Comment
0	See APL function: "Setting the startup response" 1 = Startup without call in OB100 0 = Startup with call in OB100 (default): • During the CPU start-up the messages are suppressed for a number of cycles configured in `RunUpCyc` (default is 3). • Manual mode is activated • The operating state `CoolState` is set to "Idle"
1	Enable switchover to "Out of service" via `OosLi` (default = 0)
...	Reserve
4	Automatic/Manual switchover 0 = Pushbutton operation; 1 = Switch operation
...	Reserve
22	See APL function: "Update acknowledgment and error status of the message call"
...	Reserve
24	1 = "Local" operator permission is active
...	Reserve

Operator permissions

This block includes the APL standard function "Operator permissions". The block has the following permissions for the `OS_Perm` parameter:

Status bit	Parameter
0	1= Operator can switch to "Automatic" mode
1	1= Operator can switch to "Manual" mode
2	Reserve
3	1 = Operator can switch to "Out of service" mode
4	1 = Operator can start the cooling program
5	1 = Operator can end the cooling program
6	1 = Operator can start the cooling program `IntTopDown`

Status bit	Parameter
7	1 = Operator can start the cooling program `IntBtmUp`
8	1 = Operator can start the cooling program `IntUseAll`
9	1 = Operator can start the cooling program `ExtAvrgT`
10	1 = Operator can start the cooling program `ExtMultiT`
11	1= Operator can start the cooling course (cooling / heating)
12	1 = Operator can start the cooling course `keep`
13	1 = Operator can change the parameter "Cooling time"
14	1 = Operator can change the parameter "Setpoint temperature"
15	1= Operator can reset errors
16	1 = Operator can change limit for temperature deviation
17	1 = Operator can change the delay time for errors
18	1 = Operator can change limit for freeze monitoring
19	1 = Operator can change the hysteresis value
20	1= Operator can change the max. cooling gradient
21	1= Operator can change the temperature deviation for the next cooling zone
22	1= Operator can change the delay time for the next cooling zone
23	1= Operator can change the function "Cooling enable follows unit allocation"
24	1 = Operator can activate the release for maintenance
25	1= Operator can change the preselection of the temperature sensors
26	1= Operator can change the preselection of the cooling zones
27	1= Operator can change the assignment of cooling zones to temperature sensors
	Reserve

22.4.4 Error correction for 'TcCtrl'

Quintessence

For information about troubleshooting all the blocks please also refer to the basic information in the APL section "Troubleshooting". The following error numbers can be displayed for this block:

Overview of error numbers

The `ErrorNum` connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.
40	ERR_CFG_MISSING Data of configuration block TcCfg faulty
41	ERR_PRESEL_TS Error in the preselection of the temperature sensors

Error number	Meaning of the error number
42	ERR_PRESEL_CZ Error in the preselection of the cooling zones
43	ERR_PROGSEL Invalid cooling program (1 > `CoolProgOut` > 5)

22.4.5 Messages of 'TcCtrl'

The block uses an instance of the ALARM_8P block (MsgEvId). The following messages can be generated:

Message instance	Message identifier	Message class	Event
`MsgEvId`	SIG 1	AS process control message - fault	$$BlockComment$$ Configuration data missing
	SIG 2	AS process control message - fault	$$BlockComment$$ Error preselection temperature sensors
	SIG 3	AS process control message - fault	$$BlockComment$$ Error preselection cooling zones
	SIG 4	AS process control message - fault	$$BlockComment$$ Error program selection
	SIG 5	Alarm high	$$BlockComment$$ Temperature too high
	SIG 6	Alarm low	$$BlockComment$$ Temperature too low
	SIG 7	AS process control message - fault	External error
	SIG 8		Free for user

Explanation:

- $$BlockComment$$ → Content of the instance-specific comment
- You can interconnect an external fault to the input parameter CSF. If this signal changes to `CSF = 1`, a fault is triggered (MsgEvId, SIG 7).
- In addition, an external message SIG 8 can be generated through the input parameter `ExtMsg1` .
- Message generation is prevented with the input signal `MsgLock`or `CmData.xMsgLock = 1` .

Associated values for message instance `MsgEvId`

Associated value	Block parameter
1	BatchName
2	StepNo
3	BatchID
4	ExtVa104
5	ExtVa105
6	ExtVa106
7	ExtVa107
8	UnitIdOut

22.4.6 Connections of 'TcCtrl'

Input parameters

Parameter	Description	Type	Default
StartAut	Start command in automatic mode	DigVal	
StopAut	Stop command in automatic mode	DigVal	
StartMan	1=Start: Start command in manual mode	BOOL	
StopMan	1=Stop: Stop command in manual mode	BOOL	
ModLiOp	1=Link/Auto,0=Manual: Input to auto/manual commands	DigVal	
AutModOp	Switch to automatic mode	BOOL	
ManModOp	Switch to manual mode	BOOL	
AutModLi	1=Auto mode: auto mode switch via link	DigVal	
ManModLi	1=Manual mode: manual mode switch via link	DigVal	
RstLi	Reset (from process)	DigVal	
RstOp	Reset (operator)	BOOL	
TsPreSel	Temperature sensor preselection (decimal flag field)	DINT	
CzPreSel	Cooling zone preselection (decimal flag field)	DINT	
CoolEn	Enable cooling program	DigVal	
CoolProg	Cooling program	INT	1
CoolFunc	Cooling function: 1=Cool/Let grow warm, 0=Keep temperature	BOOL	True
CoolTime	Cooling time [h]	REAL	48.0
CoolTSP	Setpoint for target temperature	REAL	9.0

Parameter	Description	Type	Default
TsSp08Aut ... TsSp01Aut	Temperature setpoint for sensor 1 ... 8 (program "ExtMultiT")	REAL	
TsSp08Man ... TsSp01Man	Temperature setpoint for sensor 1 ... 8 (program "ExtMultiT")	REAL	
TsPreSelMan	Temperature sensor preselection	DINT	
CzPreSelMan	Cooling zone preselection (decimal flag field)	DINT	
CpIntTopDownMan	Cooling program "Top down"	BOOL	
CpIntBtmUpMan	Cooling program "Bottom up"	BOOL	
CpIntUseAllMan	Cooling program "Use all zones"	BOOL	
CpExtAvrgTMan	Cooling program "Average temperature"	BOOL	
CpExtMultiTMan	Cooling program "Multipe temperatures"	BOOL	
CfCoolMan	Cooling direction: 1=Cool/Let grow warm	BOOL	
CfKeepMan	Cooling direction: 1=Keep temperature	BOOL	
CoolTimeMan	Cooling time [s]	REAL	
CoolTSpMan	Temperature setpoint	REAL	
Hysteresis	Temperature hysteresis	REAL	0.25
MaxRamp	Cool ramp maximum [°C/h]	REAL	0.50
NxtCzDifT	Temperature difference before next cooling zone will (should) be activated	REAL	0.4
NxtCzDelT	Delay time before activating next cooling zone [s]	REAL	20.0
AlmDifT	Temperature difference before raising an error	REAL	0.6
AlmDelT	Delay time before an error is raised [s]	REAL	30.0
FreezeLim	Minimum temperature to allow before locking valves	REAL	-4.0
RelOnOcc	Program release follows batch occupation	BOOL	
Cz4Ts08Aut	Cooling zones for temperatur sensor 8	DINT	10000000
Cz4Ts07Aut	Cooling zones for temperatur sensor 7	DINT	1000000
Cz4Ts06Aut	Cooling zones for temperatur sensor 6	DINT	100000
Cz4Ts05Aut	Cooling zones for temperatur sensor 5	DINT	10000
Cz4Ts04Aut	Cooling zones for temperatur sensor 4	DINT	1000
Cz4Ts03Aut	Cooling zones for temperatur sensor 3	DINT	100
Cz4Ts02Aut	Cooling zones for temperatur sensor 2	DINT	10
Cz4Ts01Aut	Cooling zones for temperatur sensor 1	DINT	1
Cz4Ts08Man	Cooling zones for temperatur sensor 8	DINT	10000000
Cz4Ts07Man	Cooling zones for temperatur sensor 7	DINT	1000000
Cz4Ts06Man	Cooling zones for temperatur sensor 6	DINT	100000
Cz4Ts05Man	Cooling zones for temperatur sensor 5	DINT	10000
Cz4Ts04Man	Cooling zones for temperatur sensor 4	DINT	1000
Cz4Ts03Man	Cooling zones for temperatur sensor 3	DINT	100
Cz4Ts02Man	Cooling zones for temperatur sensor 2	DINT	10
Cz4Ts01Man	Cooling zones for temperatur sensor 1	DINT	1

Parameter	Description	Type	Default
FillLev	Tank fill level	AnaVal	
FillLev_Unit	Unit of fill level	INT	1041
TsAv08 … TsAv01	Sensor 1 ... 8 actual value	AnaVal	
TsAv_Unit	Engineering units of temperature value	INT	1001
TsAv_Opscale	Limit values for scale in Temperature display of face-plate	STRUCT • High: REAL • Low: REAL	100.0 -10.0
TsRel08 … TsRel01	Sensor 1 ... 8 is covered	DigValFF	
CzlRel08 … CzlRel01	Cooling zone 1 ... 8 is covered	DigValFF	
CfgData	Configuration data	STRUCT • High: REAL • Low: REAL	
CmData	Control and status information	BlUdtCmData	
OvwUnitID	1=Overwrite CmData.UnitID with input UnitID	BOOL	
OvwBatchInfo	1=Overwrite CmData batch data with BatchX input data	BOOL	
OvwManOpEn	1=Overwrite CmData ManOpEn with input ManOpEn	BOOL	
UnitID	Unit ID (overwritable CmData element)	DWORD	
BatchEnLi	Batch is enabled (overwritable CmData element)	BOOL	
BatchOccLi	Occupied by batch (overwritable CmData element)	BOOL	
BatchIdLi	Batch ID (overwritable CmData element)	DWORD	
BatchNameLi	Batch name (overwritable CmData element)	STRING[32]	
StepNoLi	Batch step number (overwritable CmData element)	DWORD	
ManOpEn	Enable switchover to manual mode	BOOL	
V08FbkOpen ... V01FbkOpen	1=Feedback valve is open	DigVal	
MsgLock	Message lock	DigVal	
MsgEvId	Message event ID 01	DWORD	16#FFFFFFFF
ExtMsg1	External message 1	BOOL	
ExtVa104 ... ExtVa107	External value 4 ... 7	ANY	
MS_RelOp	Operator input for MS Release, 1: MS release requirement	BOOL	
RunUpCyc	Number of cycles for which this block keeps in start up mode	INT	3
SampleTime	Sampling time [s]	REAL	1.0

Parameter	Description	Type	Default
UserStatus	User status bits	BYTE	
CSF	Control System Fault message - external error	DigVal	
OosLi	1=Oos mode: Oos mode switch via link	DigVal	
OosOp	1=Oos mode: Oos mode by operator	BOOL	
SelFp1	Open faceplate of connected block	BOOL	
SelFp2	Open faceplate of connected block	BOOL	
SelFpTank	Open faceplate of connected Tank block	BOOL	
OS_Perm	Operator Permissions	STRUCT	
OS_Perm1	Operator Permissions 1	STRUCT	
OpSt_In	Enabled operator stations	DWORD	
Feature	Feature collection	STRUCT	

Output parameters

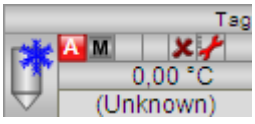
Parameter	Description	Type	Default
MS_Release	1: Maintenance status release	BOOL	
AutAct	1=Auto: Automatic mode indicator	DigVal	
ManAct	1=Manual: Manual mode indicator	DigVal	
OosAct	Out of service is active	DigVal	
TsPreSelOut	Temperature sensor preselection	DINT	
CzPreSelOut	Cooling zone preselection (decimal flag field)	DINT	
CoolProgOut	Cooling program	INT	1
CoolFuncOut	Cooling function: 1=Cool/Let grow warm, 0=Keep temperature	BOOL	True
CoolTimeOut	Cooling time (in hours)	REAL	
CoolTspOut	Temperature setpoint	REAL	
CoolState	Cooling state	INT	1
Cz4Ts08Out	Cooling zones for temperatur sensor 8	DINT	10000000
Cz4Ts07Out	Cooling zones for temperatur sensor 7	DINT	1000000
Cz4Ts06Out	Cooling zones for temperatur sensor 6	DINT	100000
Cz4Ts05Out	Cooling zones for temperatur sensor 5	DINT	10000
Cz4Ts04Out	Cooling zones for temperatur sensor 4	DINT	1000
Cz4Ts03Out	Cooling zones for temperatur sensor 3	DINT	100
Cz4Ts02Out	Cooling zones for temperatur sensor 2	DINT	10
Cz4Ts01Out	Cooling zones for temperatur sensor 1	DINT	1
TsSp08Out	Temperature setpoint for sensor 8 (program "ExtMultiT")	REAL	
TsSp07Out	Temperature setpoint for sensor 7 (program "ExtMultiT")	REAL	
TsSp06Out	Temperature setpoint for sensor 6 (program "ExtMultiT")	REAL	

Parameter	Description	Type	Default
TsSp05Out	Temperature setpoint for sensor 5 (program "ExtMultiT")	REAL	
TsSp04Out	Temperature setpoint for sensor 4 (program "ExtMultiT")	REAL	
TsSp03Out	Temperature setpoint for sensor 3 (program "ExtMultiT")	REAL	
TsSp02Out	Temperature setpoint for sensor 2 (program "ExtMultiT")	REAL	
TsSp01Out	Temperature setpoint for sensor 1 (program "ExtMultiT")	REAL	
V08Open ... V01Open	Valve control: 1=Open, 0=Close	DigVal	
V08Intlk ... V01Intlk	0=Zone not available or freezing	DigVal	
ErrCfg	Error: invalid configuration data	BOOL	
ErrPsTs	Error: temperature sensor preselection invalid	BOOL	
ErrPsCz	Error: temperature sensor preselection invalid	BOOL	
ErrHiT	Error: temperature too high	BOOL	
ErrLoT	Error: temperature too low	BOOL	
ErrProg	Error: invalid cooling program	BOOL	
GrpErr	Group error	BOOL	
WaitTs	Waiting for a usable temperature value	BOOL	
WaitCz	Waiting for a usable cooling zone	BOOL	
Done	Program is done (time's up, no error)	BOOL	
TmElPrg	Elapsed program time [s]	REAL	
TmRmPrg	Remaining program time [s]	REAL	
TmRmNcz	Remaining time before next cooling zone is requested [s]	REAL	
Tsd08 ... Tsd01	Internal	BlUdtTcT_Data	
CfgLvMin	Internal	REAL	
CfgLvmax	Internal	REAL	
CfgTsCnt	Internal	INT	
CfgCzCnt	Internal	INT	
CfgTsl08 ... CfgTsl01	Internal	REAL	
CfgCzl08 ... CfgCzl01	Internal	REAL	
OSCzStatus	Internal	DWORD	
OSTsCz	Internal	DWORD	

Parameter	Description	Type	Default
OSAvgT	Internal	AnaVal	
UnitIDOut	Unit ID	DWORD	
BatchEn	Batch is enabled	BOOL	
BatchOcc	Occupied by batch	BOOL	
BatchID	Batch ID	DWORD	
BatchName	Batch name	STRING[32]	
StepNo	Batch step number	DWORD	
OS_PermOut	Operator permissions (OS info)	DWORD	16#FFFFFFFF
OS_PermLog	Operator permissions (OS info)	DWORD	16#FFFFFFFF
OpSt_Out	Enabled operator stations	DWORD	
ST_Worst	Worst signal status	BYTE	16#80
Status1	Status1 data	DWORD	
Status2	Status2 data	DWORD	
ErrorNum	Error number	INT	-1
MsgErr	ALARM_8P: Messaging error	BOOL	
MsgStat	ALARM_8P: Status output	WORD	
MsgAckn	ALARM_8P: ACK_STATE output	WORD	

22.4.7 Operator control and monitoring

22.4.7.1 Symbol



Block instance name
Status bar
Tank temperature | Unit
Status cooling program:

- 0=(unknown);
- 1=Idle
- 2=Starting
- 3=Running

The status bar contains the following symbol information sequentially from left to right:

Position	Content		
1	Message status *)		
2	Operating mode	A	Automatic
		M	Manual
			"Out of service"

Position	Content		
3	State		Cooling program is running
			Cooling program terminated
			Wait until temperature sensor available
4	Group error		
5	Empty		
6	Enable maintenance operation		
7	Memo is available		
*) Additional details about the "Message status" are available in the "Block icon structure" section of the APL documentation.			

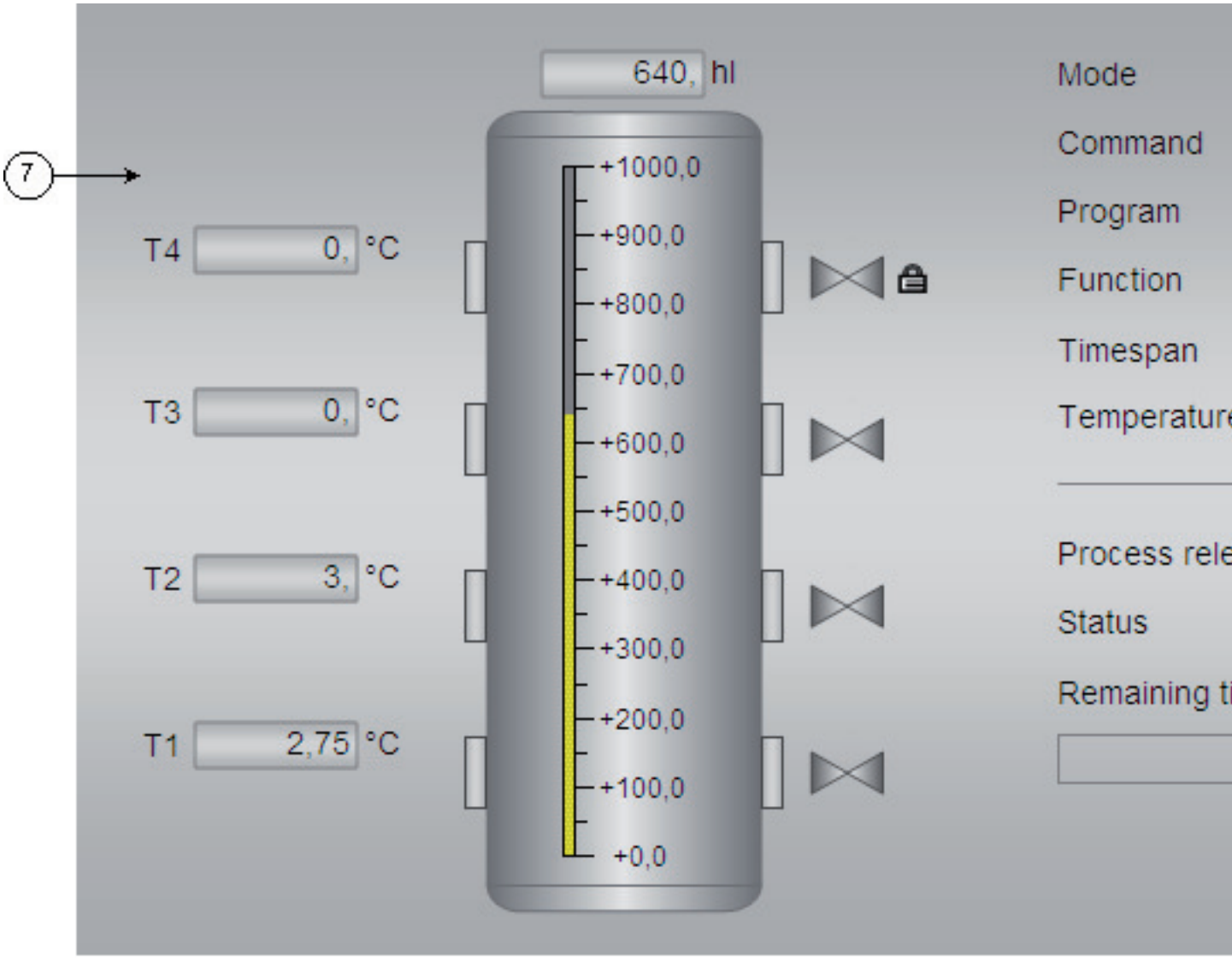
22.4.7.2 Faceplate

Views of the ‚TcCtrl' block

The block provides the following views:

Symbol	Views	
	Standard view	See below.
	Message view	Standard message view of the Advanced Process Library
	Parameter view 1	See below.
	Parameter view 2	See below.
	Preview	See below.
	Trend view	Standard trend view of all the blocks of the Advanced Process Library. • The changes in temperature of the temperature sensors defined at the configuration block TcCfg are displayed here
	Memo view	Standard memo view of all the blocks of the Advanced Process Library
	Batch view	Standard batch view of all the blocks of the Advanced Process Library

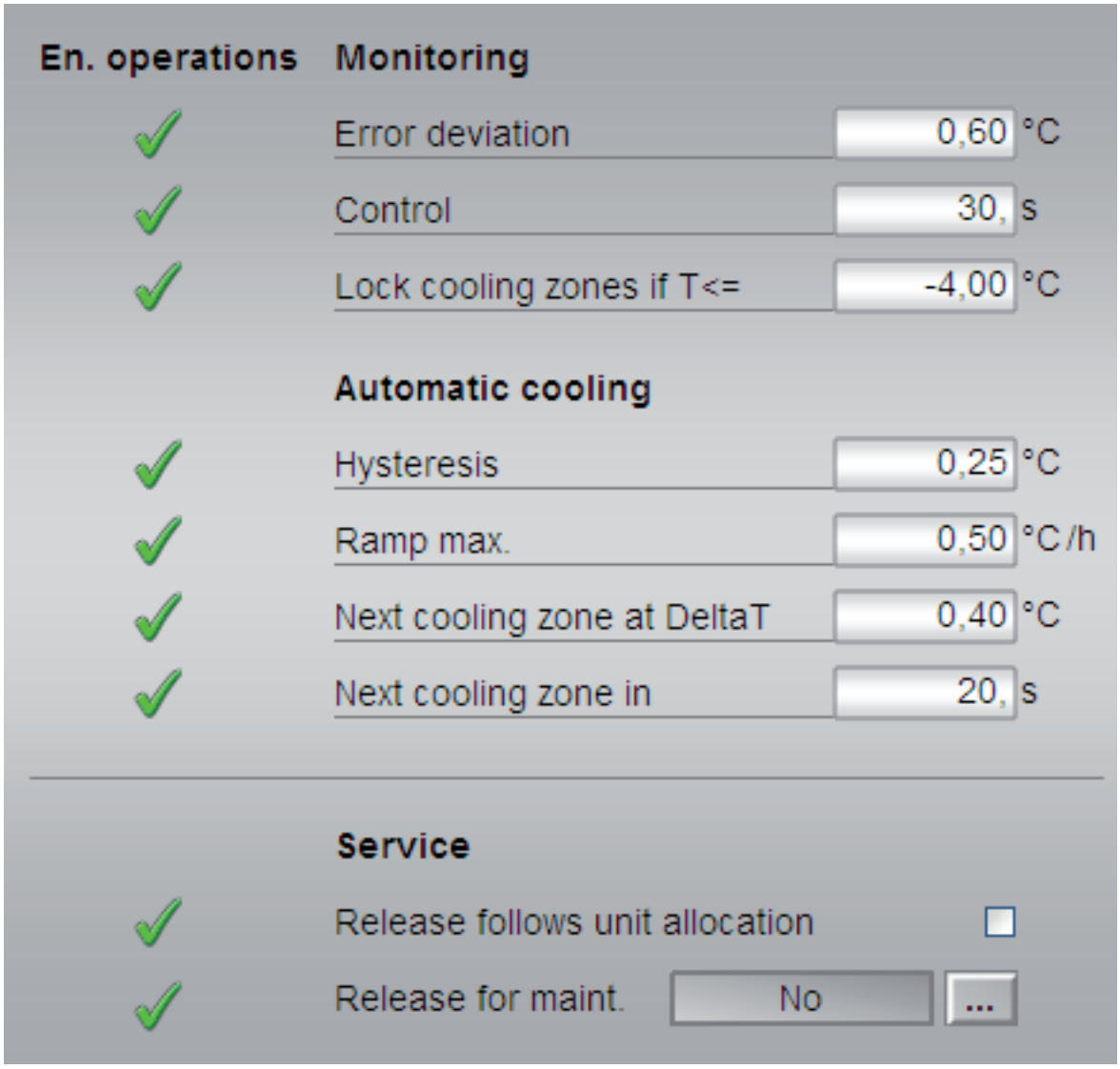
Standard view of the ‚TcCtrl'.



Explanation of the screen shot	
(1)	**Operating mode**
	This section provides information on the currently valid operating mode for tank cooling. The following operating modes can be shown and executed here: • Automatic • Manual • Out of service

Explanation of the screen shot		
(2)	**Command**	
	This section shows the current command for the cooling program. The following commands can be shown and executed here: • Start The preselected cooling program is started. The cooling program state changes to 'Starting' and then to 'Running' • Stop The preselected cooling program ('Starting' or 'Idle' state) is terminated. The cooling program state changes to the 'Idle' state	
(3)	**Displaying and switching the cooling program**	
	This section displays the currently valid cooling program. The following cooling programs can be executed in the 'Idle' state: • Fermentation • Maturation • Fast cooling • Extern AvgT • Extern MultT	
(4)	**Displaying and switching the change in temperature**	
	This section displays the currently valid change in temperature. The following changes in temperature can be executed in the 'Idle' state: • Let cool / heat up • Hold	
(5)	**Specification of duration and target temperature**	
	The following data for the running cooling program are displayed here. These can be changed in manual mode and the state 'Idle': • Duration of the program (days, hours, minutes) • Setpoint temperature of the tank	
(6)	**Current status of the cooling program**	
	This section shows the following data for the current cooling program:	
	• Process release	Status of the enable signal / input parameter `CoolEn`
	• State	Cooling program state 'Idle' or 'Starting' or 'Running'
	• Remaining time	Remaining time of the current cooling program (in the 'Running' state)
	• Message status	Display of the current / last message
(7)	**Current status of the tank**	
	This section shows the state of the temperature sensors and cooling zone valves as well as the tank level. • Current temperature of the available sensors Ts01 ... Ts08 • Current status of the available cooling zone valves V01 ... V08 (the locking state is displayed next to the valve symbol)	
(8)	**Reset command**	
	The button can be used to reset the block error messages 'Temperature too high' or 'Temperature too low'.	
(9)	**Navigation button to the standard view of the tank status block**	
	This navigation button is used to access the standard view of the block ‚Tank'.	

Parameter view 1 of ‚TcCtrl'

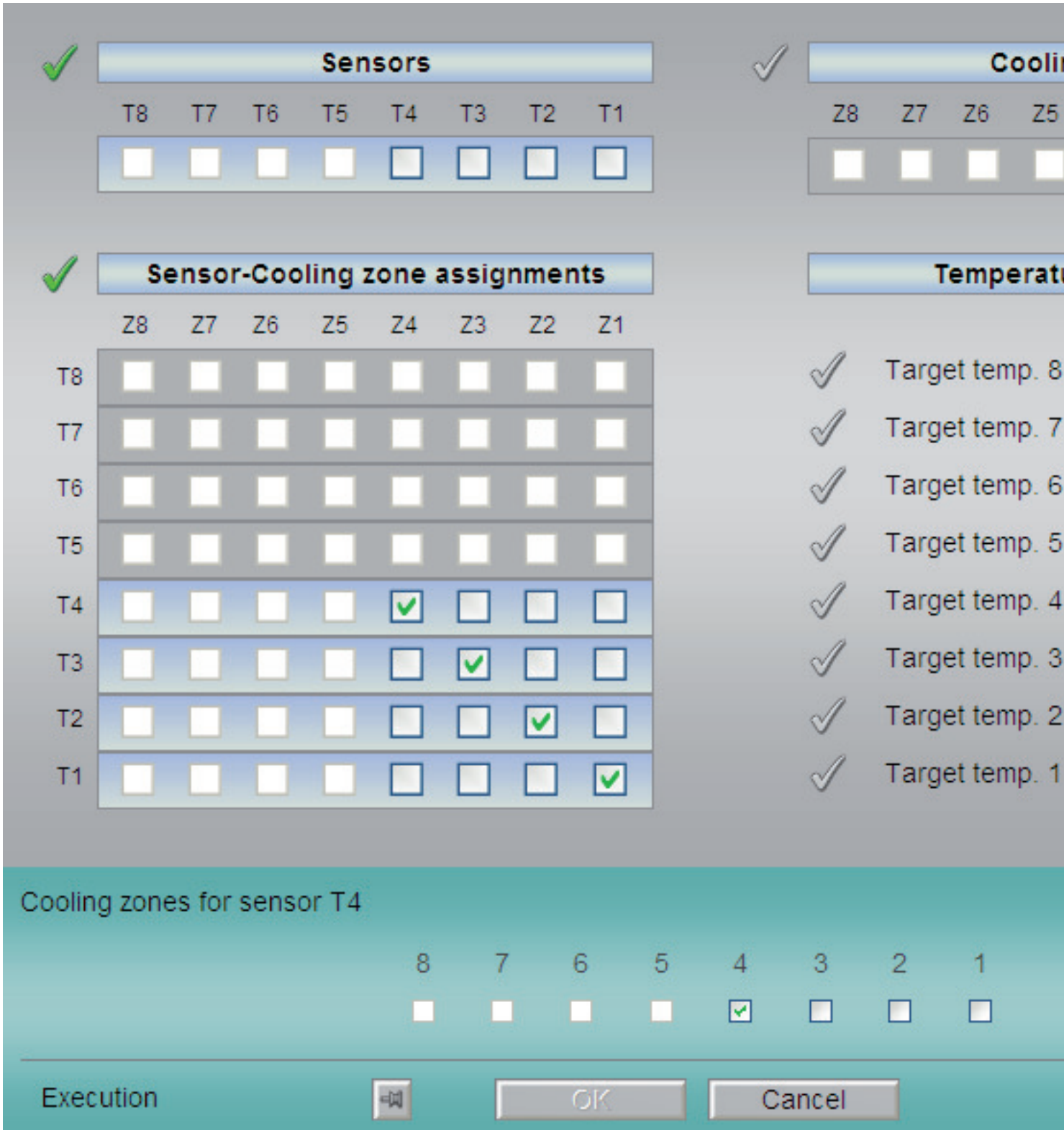


Explanation of the screen shot	
(1)	**Monitoring parameters**
	The display and change of the monitoring parameters is carried out in this section. The display fields are interlinked with the corresponding block parameters: • Error deviation – Temperature difference between actual value and setpoint that results in an error message / input parameter `AlmDifT` • Message time delay – Delay time for the generation of an error message / input parameter `AlmDelT` • Interlock cooling zones at T <= → Min. temperature for the interlocking of the cooling valves (freeze monitoring) / input parameter `FreezeLim`

Explanation of the screen shot	
(2)	**Automatic cooling**
	The display and change of the cooling program parameters is carried out in this section. The display fields are interlinked with the corresponding block parameters: • Hysteresis – Switching hysteresis for the control valves / input parameter `Hysteresis` • Ramp max. – Maximum permitted rate of rise of the cooling ramp / input parameter `MaxRamp` • Next cooling zone at DeltaT – Temperature difference between actual value and setpoint that results in the request for a further cooling zone. Input parameter `NxtCzDifT` • Next cooling zone in – Delay time for the activation of the next cooling zone / input parameter `NxtCzDelT`
(3)	**Service**
	In this section the release for maintenance function as well as the preselection 'Release after unit allocation' (input parameter `RelOnOcc`) can be activated / deactivated if the corresponding operator control enable exists.

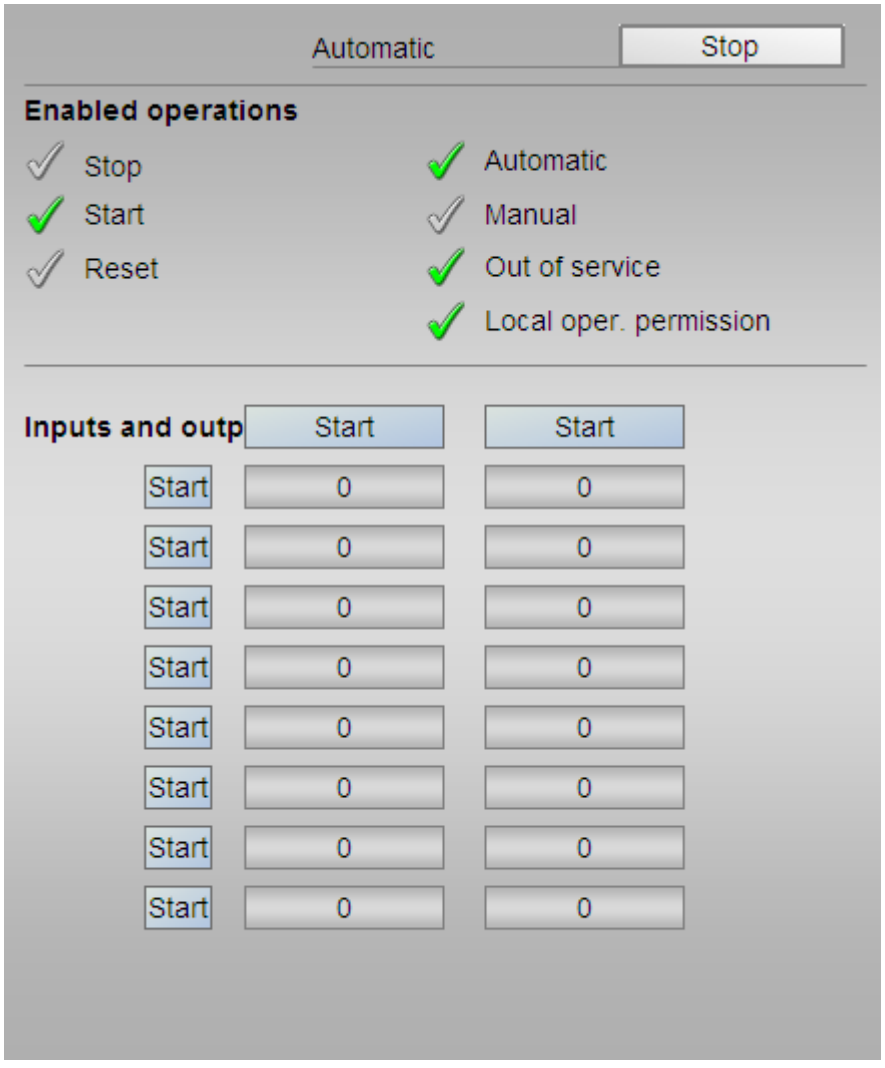
Parameter view 2 of 'TcCtrl'

The preselection of the temperature sensors and cooling zones to be used by the cooling program is carried out in manual operating mode in this view.


Sensors
T8 T7 T6 T5 T4 T3 T2 T1
Cooli
Z8 Z7 Z6 Z5
Sensor-Cooling zone assignments
Z8 Z7 Z6 Z5 Z4 Z3 Z2 Z1
T8
T7
T6
T5
T4
T3
T2
T1
Temperat
Target temp. 8
Target temp. 7
Target temp. 6
Target temp. 5
Target temp. 4
Target temp. 3
Target temp. 2
Target temp. 1
Cooling zones for sensor T4
8 7 6 5 4 3 2 1
Execution
OK
Cancel

Explanation of the screen shot	
(1)	**Guide sensors**
	Preselection of the temperature sensors that can be activated by the cooling program
(2)	**Sensor-Cooling zone assignments**
	Preselection / Assignment of the active cooling zones per temperature sensor
(3)	**Cooling zones**
	Preselection of the cooling zones that can be activated (only enabled for the program 'Extern AvgT')
(4)	**Temperature setpoints**
	Specification of the temperature setpoints (only enabled for the program 'Extern MultT')
(5)	**Selection box (cooling zone assignment for ...)**
	Selection of the temperature sensors / cooling zones (is displayed after a click on a line in the parameter window)

Preview of 'TcCtrl'



Explanation of the screen shot	
(1)	**Operator control enables**
	This section shows all the operations for which there are special operator authorizations. They depend on the configuration of the `OS_Perm` parameter of this block. The operator control enables shown in the screenshot are possible here. • Meaning of the symbols: – Green check mark: The OS operator can control this parameter – Gray check mark: The OS operator cannot control this parameter at this time due to the process – Red cross: The OS operator cannot control this parameter due to the configured operator permissions
(2)	**Outputs**
	The status of the following output bits is displayed per cooling zone valve V08 ... V01 in this section. • Control → Output parameter `V08Open … V01Open`: • Enable → Output parameter `V08Intlk … V01Intlk`

22.5 TcTsSel – Temperature sensor selection block

22.5.1 Description of 'TcTsSel'

Area of application and operating principle

Fermentation and storage tanks dispose of temperature sensors mounted at different heights. Different temperature zones that are queried in an automatic program are of interest for the introduction and monitoring of the fermentation process. Since tanks do not always have to be filled completely (so that some may be dry) it is tedious for the automatic program to obtain reliable temperature feedbacks for a desired zone.

This function block determines a clear temperature feedback of the available, enabled or selected temperature sensors from the configured geometric sensor parameters and the current tank level.

Configuration

This block is called in the cooling zone control block 'TcCtrl' as a multi-instance FB. Therefore the installation in the CFC is not required when it is used. However, the block can be installed and interconnected freely in the application CFCs to achieve an own user-specific cooling zone control system without the use of 'TcCtrl'.

The block 'TcTsSel' must always exist in the block container of the S7 program.

The number of available temperature sensors (1 to 8) as well as their level thresholds are specified through an own parameter block 'TcCfg' that is connected through a data structure connection.

Status word allocation 'Status1'

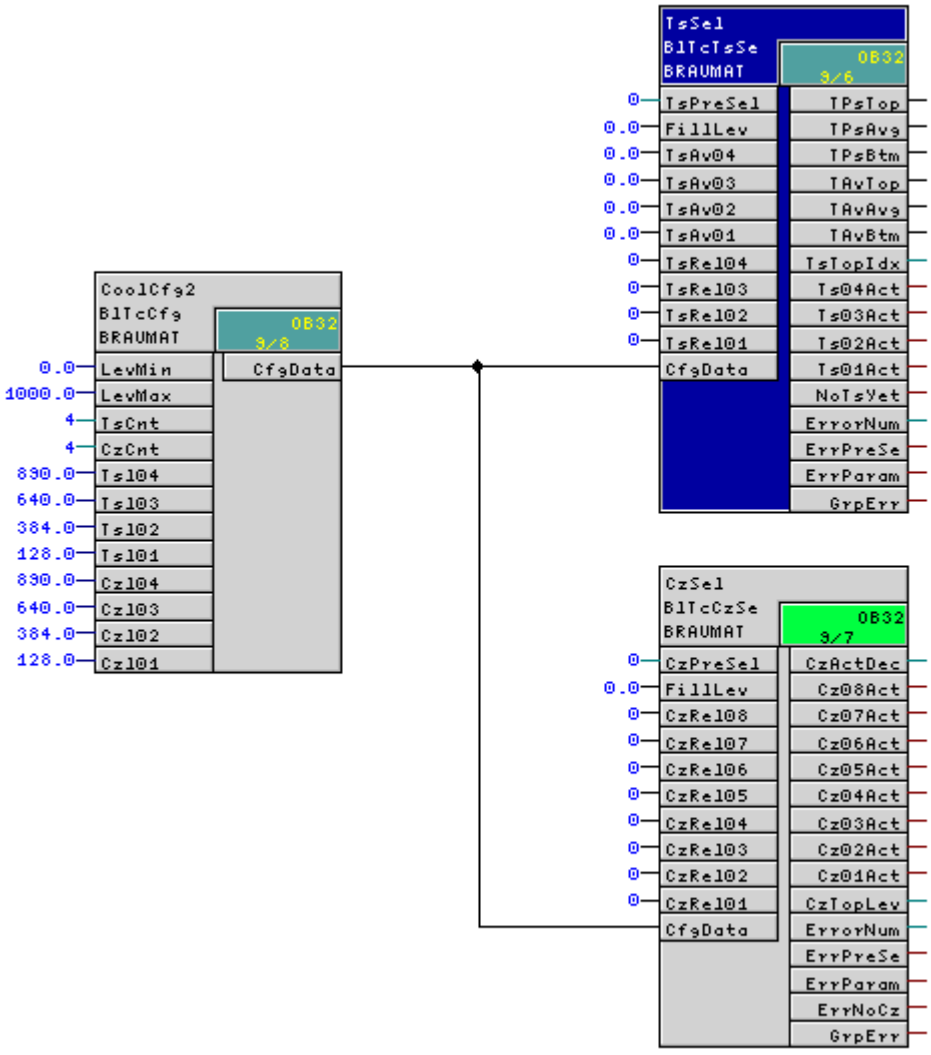
For a description of the individual parameters, see the section "Connections of 'TcTsSel' (Page 1216)".

Status bit	Parameter
0	`Reserve`
1	`Reserve`
2	`Reserve`
3	`Reserve`
4	`NoTsYet` 1 = No valid temperature sensors available
5	`ErrPreSel` 1 = Sensor preselection lies outside the limits
6	`ErrParam` 1 = Parameter assignment error
7	`GrpErr` 1 = Group error

Status bit	Parameter
8 ... 15	`Ts01Act ... Ts08Act` 1 = Temperature sensor is active
	`Reserve`

22.5.2 Functions of 'TcTsSel'

Interconnection example



The example shows the parameter block 'TcCfg' and the selection blocks 'TcTsSel' (temperature) and 'TcCzSel' (cooling zones -> TcCzSel – Cooling zone selection block (Page 1217)). The inputs and outputs each for 4 sensors are connected visibly.

Description of functions:

- Up to eight temperature sensors can be processed.
- The desired temperature sensors are specified through the input parameter `TsPreSel` of the type DINT that represents the sensor indices (bit index 1 to 8) in decimal (!) form (decimal 11.000 then corresponds to Sensors 5 and 4).
- Each temperature sensor receives the following at the input side:
  - An analog value `TsAv01 ... TsAv08` (REAL) with quality status
  - A binary enable signal `TsRel01 ... TsRel08` (possibly from a wetness sensor)
  - An analog level threshold value (block `TcCfg`, connections `Tsl01 ... Tsl08`) that the signals the usability (intended alternatively to the enable signal if no wetness sensors are available.)
- The block makes the following values / signals available at the output side:
  - `TPsTop` → Temperature of the highest preselected sensor
  - `TPsAvg` → Mean temperature of the preselected sensors
  - `TPsTop` → Temperature of the lowest preselected sensor
  - `TAvTop` → Temperature of the highest available sensor
  - `TAvAvg` → Mean temperature of the available sensors
  - `TAvTop` → Temperature of the lowest available sensor
  - `Ts01Act ... Ts08Act` → Bit signals: 'Sensor is active'
- The actual level of the tank is supplied by the analog value input `FillLev` .
- If a desired temperature sensor is not available, the next possible one(s) downward is/are searched for and used. If this search does not produce a result, a error output is set (see the section Connections of 'TcTsSel' (Page 1216)).

Overview of error numbers

The `ErrorNum` connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.
1	If a parameterization data structure (block 'TcCfg') is faulty or does not exist
2	If the sensor preselection `TsPreSel` is faulty

22.5.3 Connections of 'TcTsSel'

Input parameters

Parameter	Description	Type	Default
TsPreSel	Temperature sensor preselection (decimal flag field, e.g. 1100)	DINT	
FillLev	Tank fill level	AnaVal	
TsAv08 ... TsAv01	Sensor 1 ...8 actual value	AnaValFF	
TsRel08 ... TsRel01	Tank level covers sensor 1 ... 8	DigValFF	
CfgData	Configuration data	STRUCT • wCfgId : WORD • iCfgSz : INT	

Output parameters

Parameter	Description	Type	Default
TPsTop	Temperature from highest preselected sensor	AnaVal	
TPsAvg	Best match temperature for selected level sensor(s) (averaged if applicable)	AnaVal	
TPsBtm	Temperature from lowest preselected sensor	AnaVal	
TAvTop	Temperature from highest available sensor	AnaVal	
TAvAvg	Averaged temperature based on available sensors	AnaVal	
TAvBtm	Temperature from lowest available sensor	AnaVal	
TsTopIdx	Index of highest available sensor (1 - 8)	INT	
TsActBit	Active sensors bit coded	BYTE	
TsAvlBit	Available sensors regardless of preselection, bit coded	BYTE	
PreSelBit	Original preselection bit coded	BYTE	
Ts08Act ... Ts01Act	Temperature sensor at level 1 ... 8 is active	BOOL	
NoTsYet	No matching sensors available yet	BOOL	
ErrorNum	Error number	INT	-1
ErrPreSel	Error: preselection out of bounds	BOOL	
ErrParam	Parameterization error	BOOL	
GrpErr	Group error	BOOL	
ST_Worst	Worst signal status	BYTE	16#80
Status1	Status1	DWORD	

22.6 TcCzSel – Cooling zone selection block

22.6.1 Description of 'TcCzSel'

Area of application and operating principle

Fermentation and storage tanks dispose of cooling zones mounted at different heights. Since the tanks do not always have to be filled completely, only the 'wet' cooling zones may be controlled by the automatic program.

This function block determines a clear activation of the available, enabled or selected cooling zones from the configured geometric cooling zone parameters and the current tank level.

Configuration

This block is called in the cooling zone control block 'TcCtrl' as a multi-instance FB. Therefore installation in the CFC is not required in this case. However, the block can be installed and interconnected freely in the application CFCs to achieve an own user-specific cooling zone control system without the use of 'TcCtrl'.

The block 'TcCzSel' must always exist in the block container of the S7 program.

The number of available cooling zones (1 to 8) as well as their level thresholds are specified through an own parameter block 'TcCfg'. This is connected through a data structure connection

22.6.2 Functions of 'TcCzSel'

Interconnection example

See the section Functions of 'TcTsSel' (Page 1214)

Description of functions:

- Up to eight cooling zones can be processed.
- The desired cooling zones are specified through the input parameter `CzPreSel` of the type DINT that represents the sensor indices (bit index 1 to 8) in decimal (!) form (decimal 11.000 then corresponds to Cooling zones 5 and 4).
- Each cooling zone receives the following at the input side:
  - A binary enable signal `CzRel01 ... CzRel08`
  - An analog level threshold value (block `TcCfg`, connections `Czl01 ... Czl08`) from which the cooling zone enable is derived (intended alternatively to the enable signal if no wetness sensors are available.)

- The block makes the following values / signals available at the output side:
  - `CzActDec` → Active cooling zones ('decimal bit encoded' of the type DINT)
  - `Cz01Act` … `Cz08Act` → Bit signals: 'Cooling zone is active'
- The actual level of the tank is supplied by the analog value input `FillLev` .
- If a desired cooling zone is not available, the next possible one(s) downward is/are searched for and used. If this search does not produce a result, a error output is set (see the section Connections of 'TcCzSel' (Page 1218)).

Overview of error numbers

The `ErrorNum` connection can be used to output the following error numbers:

Error number	Meaning of the error number
-1	Default value when implementing the block; block will not be processed
0	There is no error.
1	If a parameterization data structure (block 'TcCfg') is faulty or does not exist
2	If the cooling zone preselection `CzPreSel` is faulty

22.6.3 Connections of 'TcCzSel'

Input parameters

Parameter	Description	Type	Default
`CzPreSel`	Cooling zone preselection (decimal flag field, e.g. 1100)	`DINT`	
`FillLev`	Tank fill level	`AnaVal`	
`CzRel08` … `CzRel01`	Tank level covers cooling zone 1 ... 8	`DigValFF`	
`CfgData`	Configuration data	`STRUCT` • `WORD` • `INT`	

Output parameters

Parameter	Description	Type	Default
`CzActDec`	Active selection (decimal flag field, e.g. 1100)	`DINT`	
`Cz08Act` … `Cz01Act`	Cooling zone 1 ... 8 is active	`BOOL`	
`CzTopLev`	Index of highest available cooling zone (1 - 8)	`INT`	
`CzActBit`	Usable zones bit coded	`BYTE`	

Parameter	Description	Type	Default
CzAvlBit	Available zones regardless of preselection, bit coded	BYTE	
TpDnMasks	Available zones as decimal masks from top to bottom	STRUCT • MSK01 … MSK08 : DINT	
BtmUpMasks	Available zones as decimal masks from bottom to top	STRUCT • MSK01 … MSK08 : DINT	
ErrorNum	Error number	INT	-1
ErrPreSel	Error: preselection out of bounds	BOOL	
ErrParam	Parameterization error	BOOL	
ErrNoCz	Error: no matching cooling zone available	BOOL	
GrpErr	Group error	BOOL	
ST_Worst	Worst signal status	BYTE	16#80

22.7 TcCfg – Cooling zones configuration block

22.7.1 Operating principle of 'TcCfg'

Operating principle of 'TcCfg'

The number of temperature sensors and cooling zones actually available (1 to 8 each) as well as their level thresholds can be specified through this parameter block'. The configured parameters are made available though a data structure output. Its use as well as the interconnection principle are described in the preceding sections.

Configuration and interconnection example

See the section Functions of 'TcTsSel' (Page 1214)

22.7.2 Connections of 'TcCfg'

Input parameters

Parameter	Description	Type	Default
`LevMin`	Minimum fill level	`REAL`	
`LevMax`	Maximum fill level	`REAL`	
`TsCnt`	Number of temperature sensors	`INT`	
`CzCnt`	Number of cooling zones	`INT`	

`Tsl08`
...
`Tsl01`
	Fill level for usage of sensor 1 ... 8 (0.0 if not in use)	`REAL`	

`Czl08`
...
`Czl01`
	Fill level for usage of cooling zone 1 ... 8 (0.0 if not in use)	`REAL`	

Output parameters

Parameter	Description	Type	Default
`CfgData`	Configuration data	`STRUCT` • `wCfgId : WORD` • `iCfgSz : INT`	

Timers

23

23.1 TimerXP – Signal delay / incremental encoder

23.1.1 Description of 'TimerXP'

Area of application and operating principle

This function block is used to condition a digital input signal. Different types of signal delay or signal extension can be preselected through six different operating modes (can be selected by an interconnectable input). The delay / extension time can be prespecified in different formats.

The block does not have an OS faceplate for operation.

Configuration

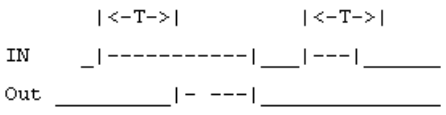
Use the CFC editor to install the block in a cyclic interrupt OB (OB30 to OB38). The block is also installed automatically in the startup OB (OB 100).

23.1.2 Functions of 'TimerXP'

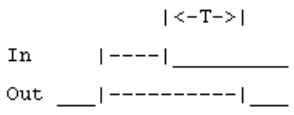
- The timer operating modes described below are supported. These can be selected with the input `TiMode`.
- The input time value is specified at the inputs `TiHours, TiMinutes, TiSeconds`.
- The output time value 'T' (Elapsed Time) is available at the `outputs ElHours, ElMinutes, ElSeconds`.
- The following output time formats (that can be selected using the input OutFormat) are available:
  - **OutFormat = 0** "discrete" meaning that the outputs display the elapsed time as follows:
    `ElHours = Elapsed Time / 3600` (= Elapsed overall time in hours)
    `ElMinutes = Elapsed Time / 60` (= Elapsed overall time in minutes)
    `ElSeconds = Elapsed Time` (= Elapsed overall time in seconds)
  - **OutFormat = 1**
    "normal" time format hh:mm:ss
    The outputs `ElHours, ElMinutes, ElSeconds` show the overall time in hours, minutes and seconds.

Operating modes of the timer block:

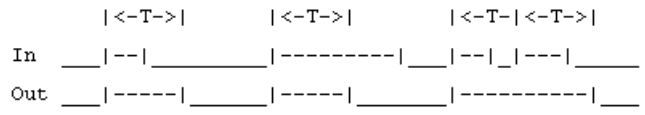
- **TiMode = 0**
  delays ON with a timer decrement of the time value from T to 0
  - Start the timer with a positive edge of the input `In`
  - Hold timer when zero → set output `Out`
  - Reset output `Out` with negative edge of the input `In`



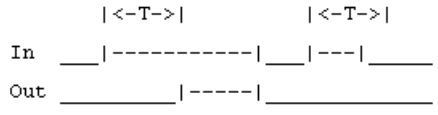
- **TiMode = 1**
  delays OFF with a timer decrement of the time value from T to 0
  - Set output `Out = 0` with a positive edge of the input `In`
  - Start the timer with a negative edge of the input `In`
  - Hold timer when zero → reset output `Out`



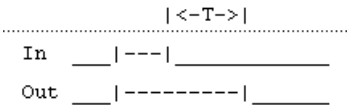
- **TiMode = 2**
  Extended PULSE with timer decrement from T to 0
  - Start the timer with a positive edge of the input `In` → set output `Out = 1`
  - Hold timer when zero → reset output `Out`



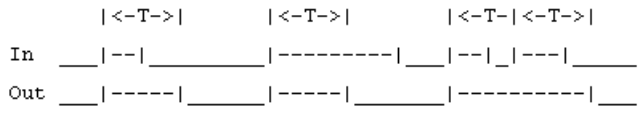
- **TiMode = 3**
  delayed ON with timer increment from 0 to T
  - Start the timer from zero with a positive edge of the input `In`
  - Hold timer when > T and set output `Out = 1`
  - Reset output `Out` with negative edge of the input `In`



- **TiMode = 4**
  delayed OFF with timer increment from 0 to T
  - Set output `Out = 1` with a positive edge of the input `In`
  - Start the timer from zero with a negative edge of the input `In`
  - Hold timer when > T → reset output `Out`



- **TiMode = 5**
  Extended PULSE with timer increment from 0 to T
  - Start the timer from zero with a positive edge of the input `In`
  - Set output `Out = 1`
  - Hold timer when > T → reset output `Out`



Note

Function input `EnableLi` (Release Timer)

- If enable input `EnableLi = 0` and the timer is started, the time stops.
- If enable input `EnableLi = 1` and the timer is started, the time continues to be counted.

Function input SampleTime

- The value at the input `SampleTime` determines the time value that is added to / subtracted from the current output time value T per cycle at a started timer.
- For example the timer is installed into and called up in the 1-sec-OB32.
  The value at the input `SampleTime = 1.0` meaning that the output time is increased / decreased every second by 1.0 at a started timer.

23.1.3 Connections of 'TimerXP'

Input parameters

Parameter	Description	Type	Default
`In`	Value input	`DigVal`	
`RstLi`	1=Reset	`DigVal`	
`TiHours`	Time [h]	`REAL`	

Parameter	Description	Type	Default
TiMinutes	Time [m]	REAL	
TiSeconds	Time [s]	REAL	5.0
TiMode	Timer mode	INT	3
OutFormat	Output format	BOOL	
EnableLi	Process release	STRUCT	16#80
SampleTime	Sampling time [s]	REAL	1.0

Output parameters

Parameter	Description	Type	Default
Out	Output	BOOL	
ElHours	Elapsed hours	REAL	
ElMinutes	Elapsed minutes	REAL	
ElSeconds	Elapsed seconds	REAL	

Appendix

24

24.1 General block I/Os

Inputs

Parameter	Description	Data type	Default
AdvCoAct *	1 = Advanced control ("Program" mode) active	STRUCT	
AdvCoAct.ST	Signal status	BYTE	16#80
AdvCoAct.Value *	Value	BOOL	0
AdvCoModSP *	Type of program mode: 1 = Set-point target, 0 = Manipulated variable target	BOOL	0
AdvCoRdy *	1 = Controller ready for advanced control (program mode)	STRUCT	
AdvCoRdy.ST	Signal status	BYTE	16#80
AdvCoRdy.Value *	Value	BOOL	0
AutModLi	1=Auto Mode: Auto Mode by Linked or SFC	STRUCT	
AutModLi.ST	Signal status	BYTE	16#80
AutModLi.Value	Value	BOOL	0
AutModOp *	1=Auto Mode: Auto Mode by Operator	BOOL	0
BatchEn	Enable remote operation of controller by batch recipe	BOOL	0
BatchID *	Current batch ID (number)	DWORD	16#00000000
BatchName *	Current batch name	STRING[32]	''
BypAct	1 = Bypass is enabled in this block	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypLiOp	1 = Bypass commands via interconnection	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypLock	1 = Bypass activation or deactivation is blocked for operator	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypPV	Substitute value if the block is bypassed	REAL	0.0

Parameter	Description	Data type	Default
BypPVLi	1 = Enable bypass PV (via interconnection)	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
BypPVOp	1 = Enable bypass PV (via operator)	BOOL	0
BypProt *	Bypass protection in simulation/ local modes	BOOL	0
CSF	Control system fault message - External error	STRUCT	
CSF.ST	Signal status	BYTE	16#80
CSF.Value	Value	BOOL	0
DeadBand	Deadband for accumulating	REAL	1.0e-2
EN		BOOL	1
EventTsIn	For interconnecting data between a technology block and the message blocks EventTs, Event16Ts. The EventTsIn input parameter serves to interconnect the EventTsOut output parameter of the EventTs, Event16Ts block. When this interconnection is configured, the messages of the EventTs, Event16Ts block are displayed on the OS in the alarm view of the technology block and can also be acknowledged there.	ANY	
ExtMsg1	External message 1	STRUCT	
ExtMsg1.ST	Signal status	BYTE	16#80
ExtMsg1.Value	Value	BOOL	0
ExtMsg2	External message 2	STRUCT	
ExtMsg2.ST	Signal status	BYTE	16#80
ExtMsg2.Value	Value	BOOL	0
ExtMsg3	External message 3	STRUCT	
ExtMsg3.ST	Signal status	BYTE	16#80
ExtMsg3.Value	Value	BOOL	0
ExtMsg4	External message 4	STRUCT	
ExtMsg4.ST	Signal status	BYTE	16#80
ExtMsg4.Value	Value	BOOL	0
ExtVa104	External value 4	ANY	
ExtVa105	External value 5	ANY	
ExtVa106	External value 6	ANY	
ExtVa107	External value 7	ANY	
ExtVa108	External value 8	ANY	

Parameter	Description	Data type	Default
ExtVa204	External value 4	ANY	
ExtVa205	External value 5	ANY	
ExtVa206	External value 6	ANY	
ExtVa207	External value 7	ANY	
ExtVa208	External value 8	ANY	
ExtVa209	External value 9	ANY	
ExtVa210	External value 10	ANY	
FaultExt	External Error	STRUCT	
FaultExt.ST		BYTE	16#80
FaultExt.Value		BOOL	0
Feature	I/O for additional functions. The description of the functions can be found in the "... functions" section for the respective block.	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 0 • .. • 0
Feature2	I/O for additional functions. The description of the functions can be found in the "... functions" section for the respective block.	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 0 • .. • 0
GradHUpLim	Gradient monitoring: Limit (high) for positive gradients	REAL	10.0
GradHDnLim	Gradient monitoring: Limit (high) for negative gradients	REAL	10.0
GradLLim	Gradient monitoring: Limit (low) for absolute gradients	REAL	1.0
GradHUpEn	1 = Activate gradient monitoring (high) for positive changes	BOOL	1
GradHDnEn	1 = Activate gradient monitoring (high) for negative changes	BOOL	1
GradLEn	1 = Activate gradient monitoring (low)	BOOL	0
GradHUpMsgEn	1 = Activate gradient (high) message for positive changes	BOOL	1
GradHDnMsgEn	1 = Activate gradient (high) message for negative changes	BOOL	1
GradLMsgEn	1 = Activate gradient (low) message	BOOL	1
Intl_En	1=Interlock enabled, 0=Interlock disabled	BOOL	1
Intlock *	1=Valve is not interlocked ,0=Valve is interlocked (no reset is required)	STRUCT	
Intlock.ST	Signal status	BYTE	16#FF
Intlock.Value *	Value	BOOL	1
LagTime	Delay time [s]	REAL	1.0

Parameter	Description	Data type	Default
LocalLi *	1=Local Mode: Local operation by field signal	STRUCT	
LocalLi.ST		BYTE	16#80
LocalLi.Value *		BOOL	0
LocalOp *	1=Local Mode: Local operation by operator	BOOL	0
LocalSetting	Local mode behavior	INT	0
ManModLi	1=Manual Mode: Manual Mode by Linked or SFC	STRUCT	
ManModLi.ST	Signal status	BYTE	16#80
ManModLi.Value	Value	BOOL	0
ManModOp *	1=Manual Mode: Manual Mode by Operator	BOOL	1
ModLiOp	1 = Link/Auto, 0 = Manual: Input to Mode Commands	STRUCT	
ModLiOp.ST	Signal status	BYTE	16#80
ModLiOp.Value	Value	BOOL	0
Monitor *	1=Monitor ON, 0=Monitor OFF: Feedback monitor	BOOL	1
MonTiDynamic *	Monitoring time for expected change of feedback FbkState [s]	REAL	3.0
MonTiStatic *	Monitoring time for unexpected change in any feedback [s]	REAL	3.0
MS_RelOp *	Operator input for MS Release, 1:MS release requirement	BOOL	0
MsgEvId1	Message event ID	DWORD	16#000000D6
MsgEvId2	Message event ID	DWORD	16#000000D7
MsgLock	Inhibit process message	STRUCT	
MsgLock.ST	Signal status	BYTE	16#80
MsgLock.Value	Value	BOOL	0
MV_HiAct *	MV high limit is actively restricting manipulated variable	STRUCT	
MV_HiAct.ST	Signal status	BYTE	16#80
MV_HiAct.Value *	Value	BOOL	0
MV_HiLim *	Limit (high) for manipulated variable MV	STRUCT	
MV_HiLim.ST	Signal status	BYTE	16#80
MV_HiLim.Value *	Value	REAL	0.0
MV_LoAct *	MV low limit is actively restricting manipulated variable	STRUCT	
MV_LoAct.ST	Signal status	BYTE	16#80
MV_LoAct.Value *	Value	BOOL	0

Parameter	Description	Data type	Default
MV_LoLim *	Limit (low) for manipulated variable MV	STRUCT	
MV_LoLim.ST	Signal status	BYTE	16#80
MV_LoLim.Value *	Value	REAL	0.0
MV_OpScale *	MV - display limits for OS	STRUCT	
MV_OpScale.High *	High value	REAL	100.0
MV_OpScale.Low *	Low value	REAL	0.0
MV_Unit *	Unit of MV	INT	1342
Occupied	Occupied by batch	BOOL	0
OnOp *	1 = On mode: On mode by operator	BOOL	0
OosLi	1 = Oos mode: Oos mode by field signal	STRUCT	
OosLi.ST	Signal status	BYTE	16#80
OosLi.Value	Value	BOOL	0
OosOp *	1 = Oos mode: Oos mode by operator	BOOL	0
OpSt_In	Enabled operator stations	DWORD	
OS_Perm	I/O for operator permissions The description of the operator permissions can be found in the section "... functions" of the relevant block.	STRUCT • Bit1: BOOL • .. • Bit31: BOOL	- • 1 • .. • 1
Perm_En	1=Permit enabled, 0=Permit disabled	BOOL	1
Permit *	1=Permit is OK, 0=Not allowed to activate the valve	STRUCT	
Permit.ST	Signal status	BYTE	16#FF
Permit.Value *	Value	BOOL	1
Prot_En	1=Protection enabled, 0=Protection disabled	BOOL	1
Protect *	0=Protection is active, 1= Protection is not active	STRUCT	
Protect.ST	Signal status	BYTE	16#FF
Protect.Value *	Value	BOOL	1
PulseWidth	Control output pulse width [s]	REAL	3.0
PV	Process value	STRUCT	
PV.ST	Signal status	BYTE	16#80
PV.Value	Value	REAL	0.0
PV_AH_Lim *	Limit value AH	STRUCT	
PV_AH_MsgEn	PV - High Alarm Enable	BOOL	1
PV_AL_Lim *	Limit value AL	STRUCT	

Parameter	Description	Data type	Default
PV_AL_MsgEn	PV - Low Alarm Enable	BOOL	1
PV_Hyst *	Hysteresis	REAL	1.0
PV_OpScale *	PV- bar display limits for OS	STRUCT	
PV_OpScale.High *	High value	REAL	100.0
PV_OpScale.Low *	Low value	REAL	0.0
PV_Unit *	Engineering units of feedback input	INT	1001
PV_WH_Lim *	Limit value WH	STRUCT	
PV_WH_MsgEn	PV - High Warning Enable	BOOL	1
PV_WL_Lim *	Limit value WL	STRUCT	
PV_WL_MsgEn	PV - Low Warning Enable	BOOL	1
RstBypLi	1 = Reset bypass PV (via interconnection)	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
RstBypOp	1 = Reset bypass PV (via operator)	BOOL	0
RstLi	Linked reset signal	STRUCT	
RstLi.ST		BYTE	16#80
RstLi.Value		BOOL	0
RstOp *	Operator reset signal	BOOL	0
RunUpCyc	Number of cycles for which all messages are suppressed	INT	3
SampleTime	Sample Time [s]	REAL	0.1
SelFp1	Select faceplate 1	ANY	
SelFp2	Select faceplate 2	ANY	
SimLiOp	Simulation on/off via 0 = Operator, 1 = Interconnection or SFC	STRUCT	
SimLiOp.ST	Signal status	BYTE	16#80
SimLiOp.Value	Value	BOOL	0
SimOn *	Simulation on/off	BOOL	0
SimOnLi	1 = Simulation activated by interconnection or SFC (controlled by SimLiOp = 1)	STRUCT	
SimOnLi.ST	Signal status	BYTE	16#80
SimOnLi.Value	Value	BOOL	0
SimPV *	Simulation Value	REAL	0.0
SimPV_Li	Linkable simulation value PV	STRUCT	
SimPV_Li.ST	Signal status	BYTE	16#80
SimPV_Li.Value	Value	REAL	0.0
SP_DnRaLim *	Downward rate limit for setpoint changes or ramping	REAL	100.0
SP_Ext *	External Analog setpoint	STRUCT	

Parameter	Description	Data type	Default
SP_Ext.ST	Signal status	BYTE	16#80
SP_Ext.Value *	Value	REAL	0.0
SP_ExtLi	Linkable input for selecting external setpoint	STRUCT	
SP_ExtLi.ST	Signal status	BYTE	16#80
SP_ExtLi.Value	Value	BOOL	0
SP_ExtOp *	Operator input for selecting external setpoint	BOOL	0
SP_HiLim *	Input high limit	REAL	100.0
SP_Int *	Internal Analog setpoint	REAL	0.0
SP_IntLi	Linkable input to select internal setpoint	STRUCT	
SP_IntLi.ST	Signal status	BYTE	16#80
SP_IntLi.Value	Value	BOOL	0
SP_IntOp *	Operator input for selecting internal setpoint	BOOL	1
SP_LiOp	Select setpoint source via 1 = interconnection, 0 = operator	STRUCT	
SP_LiOp.ST	Signal status	BYTE	16#80
SP_LiOp.Value	Value	BOOL	0
SP_LoLim *	Input low limit	REAL	0.0
SP_RateOn *	Setpoint changes are rate limited by SP_UpRaLim and SP_DnRaLim	BOOL	0
SP_RmpModTime *	1: SP ramped over SP_RmpTime, 0: SP ramped by rates SP_UpRaLim, SP_DnRaLim	BOOL	0
SP_RmpOn *	Setpoint is ramped to SP_RmpTarget	BOOL	0
SP_RmpTarget *	Setpoint ramp target	REAL	0.0
SP_RmpTime *	Setpoint ramp time [s]	REAL	0.0
SP_TrkExt *	1: SP_Int tracks SP_Ext in External	BOOL	0
SP_UpRaLim *	Upward rate limit for setpoint changes or ramping	REAL	100.0
StepNo *	Batch step number	DWORD	16#00000000
StopAut	1=Stop: Stop Command in Auto Mode	STRUCT	
StopAut.ST		BYTE	16#80
StopAut.Value		BOOL	0
StopMan *	1=Stop: Stop Command in Manual Mode	BOOL	0
TimeFactor *	Time Conversion Factor (0- Sec, 1- Min, 2- Hour)	INT	0
UA1unit *	Unit of UserAna1	INT	0
UA2unit *	Unit of UserAna2	INT	0

Parameter	Description	Data type	Default
UserAna1 *	User analog input 1	STRUCT	
UserAna1.ST	Signal status	BYTE	16#FF
UserAna1.Value *	Value	REAL	0.0
UserAna2 *	User analog input 2	STRUCT	
UserAna2.ST	Signal status	BYTE	16#FF
UserAna2.Value *	Value	REAL	0.0
UserStatus *	User status bits	BYTE	16#00
WarnTiAut	Warning time prior to valve motion in automatic mode [s]	REAL	0.0
WarnTiMan	Warning time prior to valve motion in manual mode [s]	REAL	0.0

* Values can be written back to these inputs during processing of the block by the block algorithm.

Outputs

Parameters	Description	Data type	Default
AdvCoEn *	1 = Enable program mode	STRUCT	
AdvCoEn.ST	Signal status	BYTE	16#80
AdvCoEn.Value *	Value	BOOL	0
AdvCoMstrOn *	Activate or deactivate program	STRUCT	
AdvCoMstrOn.ST	Signal status	BYTE	16#80
AdvCoMstrOn.Value *	Value	BOOL	0
AdvCoMV *	Target value from the external program	STRUCT	
AdvCoRdy	1 = Ready for advanced control (signal for advanced controller)	STRUCT	
AdvCoRdy.ST	Signal status	BYTE	16#80
AdvCoRdy.Value	Value	BOOL	0
AutAct	1 = Auto: Automatic Mode Indicator	STRUCT	
AutAct.ST	Signal status	BYTE	16#80
AutAct.Value	Value	BOOL	0
AutoST	1 = Worst signal status of automatic commands	BYTE	16#80
ENO		BOOL	0
ErrorNum	Error Number	INT	-1

Parameters	Description	Data type	Default
GradHUpAct	1 = Gradient alarm (high) for positive changes. You can change the response for this parameter by setting Feature bit 28 (section "Disabling operating points") and Feature bit 29 (section "Signaling limit violation") in the APL manual.	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
GradHDnAct	1 = Gradient alarm (high) for negative changes. You can change the response for this parameter by setting Feature bit 28 (section "Disabling operating points") and Feature bit 29 (section "Signaling limit violation") in the APL manual.	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
GradLAct	1 = Low gradient alarm. You can change the response for this parameter by setting Feature bit 28 (section "Disabling operating points") and Feature bit 29 (section "Signaling limit violation") in the APL manual.	STRUCT • Value: BOOL • ST: BYTE	- • 0 • 16#80
GrpErr	1 = Group error is active	STRUCT	
GrpErr.ST		BYTE	16#80
GrpErr.Value		BOOL	0
LocalAct	1=Local operation mode is active	STRUCT	
LocalAct.ST		BYTE	16#80
LocalAct.Value		BOOL	0
LockAct	1 = Interlock (Permit, Interlock or Protect) is active	STRUCT	
LockAct.ST		BYTE	16#80
LockAct.Value		BOOL	0
ManAct	Manual value is active	STRUCT	
ManAct.ST	Signal status	BYTE	16#80
ManAct.Value	Value	BOOL	1
MonDynErr	Feedback Error on change in control output	STRUCT	
MonDynErr.ST		BYTE	16#80
MonDynErr.Value		BOOL	0
MonStaErr	Feedback Error on unexpected feedback change	STRUCT	
MonStaErr.ST		BYTE	16#80
MonStaErr.Value		BOOL	0

Parameters	Description	Data type	Default
MS_Release	1: MS release	STRUCT	
MS_Release.ST	Signal status	BYTE	16#80
MS_Release.Value	Value	BOOL	0
MsgAckn1	ALARM_8P: ACK_STATE output	WORD	16#0000
MsgAckn2	ALARM_8P: ACK_STATE output	WORD	16#0000
MsgErr1	1=Messaging error occurs	BOOL	0
MsgErr2	1=Messaging error occurs	BOOL	0
MsgStat1	ALARM_8P: Status output	WORD	16#0000
MsgStat2	ALARM_8P: Status output	WORD	16#0000
OnAct	On mode is active	STRUCT	
OnAct.ST	Signal status	BYTE	16#80
OnAct.Value	Value	BOOL	1
OosAct *	Out of service is active	STRUCT	
OosAct.ST	Signal status	BYTE	16#80
OosAct.Value *	Value	BOOL	0
OpSt_Out *	Enabled operator stations	DWORD	16#00000000
OS_PermLog *	Operator permissions: Output for OS	DWORD	16#00000000
OS_PermOut *	Operator permissions: Output for OS	DWORD	16#00000000
P_Rst	Reset Impulse	STRUCT	
P_Rst.ST		BYTE	16#80
P_Rst.Value		BOOL	0
PV_AH_Act	Limit AH active	STRUCT	
PV_AH_Act.ST	Signal status	BYTE	16#80
PV_AH_Act.Value	Value	BOOL	0
PV_AL_Act	Limit AL active	STRUCT	
PV_AL_Act.ST	Signal status	BYTE	16#80
PV_AL_Act.Value	Value	BOOL	0
PV_Grad	Gradient value.	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
PV_GradPP	Gradient maximum peak value	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
PV_GradNP	Gradient minimum peak value	STRUCT • Value: REAL • ST: BYTE	- • 0.0 • 16#80
PV_Out *	Measured Value	STRUCT	
PV_Out.ST	Signal status	BYTE	16#80
PV_Out.Value *	Value	REAL	0.0

Parameters	Description	Data type	Default
PV_WH_Act	Limit WH active	STRUCT	
PV_WH_Act.ST	Signal status	BYTE	16#80
PV_WH_Act.Value	Value	BOOL	0
PV_WL_Act	Limit WL active	STRUCT	
PV_WL_Act.ST	Signal status	BYTE	16#80
PV_WL_Act.Value	Value	BOOL	0
RdyToReset	1 = ready to reset via RstLi or automatic commands	STRUCT	
RdyToReset.ST		BYTE	16#80
RdyToReset.Value		BOOL	0
RdyToStart	1 = ready to start	STRUCT	
RdyToStart.ST		BYTE	16#80
RdyToStart.Value		BOOL	0
Reserved	Reserved	STRUCT	
Reserved.ST		BYTE	16#80
Reserved.Value		BOOL	0
SP *	Setpoint	STRUCT	
SP.ST		BYTE	16#80
SP.Value *		REAL	0.0
SP_DnRaAct	Setpoint negative rate limit is active	STRUCT	
SP_DnRaAct.ST	Signal status	BYTE	16#80
SP_DnRaAct.Value	Value	BOOL	0
SP_ExtAct	1: External SP is active, 0: Internal SP is active	STRUCT	
SP_ExtAct.ST	Signal status	BYTE	16#80
SP_ExtAct.Value	Value	BOOL	0
SP_HiAct	Input High Limit Active	STRUCT	
SP_HiAct.ST	Signal status	BYTE	16#80
SP_HiAct.Value	Value	BOOL	0
SP_LoAct	Input Low Limit Active	STRUCT	
SP_LoAct.ST	Signal status	BYTE	16#80
SP_LoAct.Value	Value	BOOL	0
SP_Out *	Analog Output Value	STRUCT	
SP_Out.ST	Signal status	BYTE	16#80
SP_Out.Value *	Value	REAL	0.0
SP_RateTarget	Target for limitation of setpoint gradients	REAL	0.0
SP_UpRaAct	Setpoint positive rate limit is active	STRUCT	

Parameters	Description	Data type	Default
SP_UpRaAct.ST	Signal status	BYTE	16#80
SP_UpRaAct.Value	Value	BOOL	0
ST_Worst *	Worst signal status	BYTE	16#80
Status1 *	Status word 1	DWORD	16#00000000
Status2 *	Status2 Word	DWORD	16#00000000
Status3 *	Status3 Word	DWORD	16#00000000
Status4 *	Status4 Word	DWORD	16#00000000
SumMsgAct	Summary message Active	STRUCT	
SumMsgAct.ST	Signal status	BYTE	16#80
SumMsgAct.Value	Value	BOOL	0
WarnAct	1 = Warning prior to motion valve in auto/manual mode	STRUCT	
WarnAct.ST		BYTE	16#80
WarnAct.Value		BOOL	0

* Values can be written back to these inputs during processing of the block by the block algorithm.

Index

A

Access control
  ILck4x3L, 1002
  ILck8x4L, 1035
AccuS
  Area of application, 671
  Block diagram, 676
  Block icon, 679
  Called blocks, 671
  Calling OBs, 671
  Error handling, 674
  Feature, 674
  Input parameters, 674
  Integrate, 672
  Message behavior, 674
  Object name, 671
  Operating modes, 672
  Operator authorizations, 673
  Output parameters, 675
  Parameter view, 678
  Preview, 679
  Standard view, 677
  Startup characteristics, 671
  Status word assignment, 672
  Time response, 671
Adapter blocks
  General input parameters, 64
  General output parameters, 66
Advanced Controller
  APC_MV, 170
Aggr08
  Parameter view, 197
  Preview, 199
  Standard view, 194
Aggr08/ Aggr16
  I/Os, 190
Aggr16
  Parameter view, 198
  Preview, 199
  Standard view, 194
Aggr16 / Aggr08
  Area of application, 181
  Called blocks, 182
  Configuration, 181
  Functional principles, 181
  Startup characteristics, 182
  Status word assignment, 182
Aggr16/ Aggr08
  Associated values, 189
  Block diagram, 193
  Delay of activation and deactivation, 186
  Error handling, 187
  External/internal setpoint specification, 184
  Feature, 186
  Instance-specific messages, 189
  Message behavior, 188
  Number of units, 184
  Object name, 181
  Opening additional faceplates, 187
  Operating modes, 183
  Operator authorizations, 186
  Process messages, 188
  Switchover criterion, 184
APC
  Configuring instructions, 125
  Inputs, 127
  Loop, 126
  Process tag list view, 130
  Run sequence, 127
APC_MpList
  Block diagram, 172
  Error handling, 171
  Functions, 171
  Message behavior, 171
  Object name, 171
  Operating modes, 171
  Operator control and monitoring, 172
APC_MV
  Advanced Controller, 170
  Block diagram, 170
  Called blocks, 166
  Error handling, 167
  Feature, 167
  Function, 166
  Input parameters, 168
  Message behavior, 167
  Object name, 166
  Operating modes, 166
  Operator control and monitoring, 170
  Output parameters, 169
APC_OpSP
  Associated values, 159
  Block diagram, 162
  Block icon, 164
  Error handling, 158
  Feature, 158

Function, 157
Object name, 156
Operating modes, 157
Operator authorizations, 157
Process messages, 159
Standard view, 163
Status word assignment, 156
APC_Supervisor
Associated values, 140
Block icon, 154
Called blocks, 133
Calling OBs, 133
Error handling, 138
Feature, 138
Function, 136
List view, 148
Message behavior, 139
Object name, 133
Operating modes, 135
Operator authorizations, 137
Overview of error numbers, 138
Parameter view, 151
Preview for redundant higher-level controller, 152
Preview for standalone higher-level controller, 153
Process messages, 139
Process tag list view, 150
Standard view for redundant higher-level controller, 145
Standard view for standalone higher-level controller, 147
Startup characteristics, 133
Status word assignment, 133
ARcvH
Called blocks, 559
Calling OBs, 559
Functional principles, 559
Object name, 558
Startup characteristics, 559
Time response, 559
Area of application
AccuS, 671
Aggr16 / Aggr08, 181
CalcPwrL, 303
CalcTHX, 473
CalcWatP, 423
HxFct, 439
ILck4x3L, 991
ILck8x4L, 1019
LdMgmt8, 272
MonAn08, 917
MonAnDi, 942
PeakMon, 333
PolygonExt, 841
PulseCon, 361
RcvH_AnaVal, 536
RcvH_DigVal, 532
S7RcvAna, 520
S7RcvDig, 517
S7RcvHAna, 549
S7RcvHDig, 545
S7SndAna, 514
S7SndDig, 511
S7SndHAna, 542
S7SndHDig, 539
SelFp, 242
SimAn, 1053
SimDi, 1059
SndH_AnaVal, 528
SndH_DigVal, 524
SplRng4, 907
UsrM, 246
VlvDiv, 681
VoteAnL, 600
VoteDiL, 638
ASRcvH
Associated values, 561
Block diagram, 563
Error handling, 560
Input parameters, 561
Messages, 560
Operating modes, 560
Operator control and monitoring, 563
Output parameters, 562
Receive process, 560
ASSendH
Associated values, 556
Block diagram, 558
Called blocks, 554
Calling OBs, 554
Error handling, 555
Functional principles, 554
Input parameters, 557
Messages, 556
Monitoring, 555
Object name, 553
Operating modes, 555
Operator control and monitoring, 558
Output parameters, 557
Startup characteristics, 554
Time response, 554
Associated values
Aggr16/ Aggr08, 189
APC_OpSP, 159

APC_Supervisor, 140
ASRcvH, 561
ASSendH, 556
CalcPwrL, 314
CalcTHX, 486
CalcWatP, 428
HxFct, 443
LdMgmt8, 287
MonAnDi, 950
MonAnL, 928
PeakMon, 343
PolyCurve, 861
PolygonExt, 845
TimeSwitch, 897
UsrM, 248
VlvDiv, 696
VoteAnL, 619
VoteDiL, 654
ASTimeBCD
Block diagram, 144, 916
Called blocks, 915
Calling OBs, 915
Error handling, 915
Function, 915
Message behavior, 916
Object name, 915
Operating modes, 915
Operator control and monitoring, 916
Automatic mode, 715
CalcPwrL, 305
LdMgmt8, 275
SPCurve, 871
VlvDiv, 686
Auxiliary values
CalcPwrL, 310
LdMgmt8, 276
PeakMon, 340
VlvDiv, 693

B

BatchIF
Input parameters, 1104
Output parameters, 1106
BitInt'
Parameter, 174
BlCmDec"
Parameter, 175
BlCmEnc'
Parameter, 176
Block calls
Called blocks, 182
Block diagram
AccuS, 676
Aggr16/ Aggr08, 193
APC_MpList, 172
APC_MV, 170
APC_OpSP, 162
ASRcvH, 563
ASSendH, 558
ASTimeBCD, 144, 916
CalcTHX, 491
CalcWatP, 431
ConvAbRe, 472
ConvCF, 469
HxFct, 446
ILck4x3L, 1007
ILck8x4L, 1040
OptiOT, 460
ParaCtrl/ParaMem, 234
PCalcWatP, 743
PeakMon, 318, 348
PHxFct, 747
PMonAn08, 780
PMonAnDi, 786
PMonAnL, 776
PMonDi08, 796
PMonDiL, 791
PMotL, 753
PMotSpdCL, 764
PMotSpdL, 770
PolyCurve, 862
PolygonExt, 847
POpAnL, 801
POpD, 805
PulseCon, 364
PUsrM, 836
PVlvL, 818
PVlvMotL, 824
PVlvPosL, 830
SelB, 587, 759, 812
SelByt, 591
SelC, 599
SelDI, 597
SelDW, 595
SelFp, 244, 881, 931, 953
SelI, 589
SelR, 585
SelStr, 582
SelW, 593
SimAn, 1056
SimDi, 1061
SplRng4, 914
TimeSwitch, 899

UsrM, 250
VlvDiv, 291, 698
VlvDsL, 727
VoteAnL, 624
VoteDiL, 658
Block icon
Panel block, 54
BlOpEnum
Input parameters, 212
Output parameters, 212
BlUnitIf'
Input parameters, 1073
Output parameters, 1074
Bumpless switchover, 356
Buttons
VlvDiv, 693
Bypass
ILck4x3L, 999
ILck8x4L, 1032
VoteAnL, 607
VoteDiL, 645
Bypass function
MonAn08, 922
MonAnDi, 946

C

CalcPwrL
Area of application, 303
Associated values, 314
Automatic mode, 305
Auxiliary values, 310
Block icon, 331
Configuration, 303
Control system fault, 313
Feature, 311
Flutter suppression, 309, 312
Functions, 305
Histogram view, 327
Limit value view, 323
Message functionality, 313
Messages, 311, 313, 314
Object name, 303
Operating Modes, 305
Operator authorizations, 309
Out of service, 305
Parameter view, 325
Preview, 330
Release for maintenance, 312
Simulate, 311
Standard view, 320
Startup characteristics, 303
Status word allocation, 303, 304, 305
Subentry, 303
Time base, 308
Time window, 305
CalcTHX
Area of application, 473
Associated values, 486
Block diagram, 491
Block icon, 501, 852
Called blocks, 475
Chart view, 496
Cooling and humidifying, 480
ERC circuit, 477
Error handling, 485
External ERC optimization, 476
Feature, 484
Function, 476
Functional principles, 474
Heating and humidifying, 479
Heating/cooling with humidifying, 482
Heating/cooling without humidifying, 481
Input parameters, 487
Limit value view, 495
Message behavior, 485
Messages, 486
Object name, 473
Opening additional faceplates, 483
Operating modes, 476
Operator authorizations, 483
Output parameters, 489
Parameter view, 497
Preview, 499
Quality code, 483
Release for maintenance, 484
Simulating signals, 485
Standard view, 493
Startup characteristics, 475
Status word assignment, 475
Time response, 475
CalcWatP
Area of application, 423
Associated values, 428
Block diagram, 431
Block icon, 437
Called blocks, 424
Calling OBs, 424
Error handling, 427
Feature, 426
Input parameters, 428
Interconnecting, 426
Message behavior, 427
Object name, 423

Operating modes, 425
Operator authorizations, 426
Output parameters, 430
Parameter view, 435
Preview, 436
Process messages, 427
Quality code, 425
Release for maintenance, 425
Simulate, 425
Standard view, 432
Startup characteristics, 424
Status word assignment, 424
Time response, 424
Called blocks
AccuS, 671
APC_MV, 166
APC_Supervisor, 133
ARcvH, 559
ASSendH, 554
ASTimeBCD, 915
CalcTHX, 475
CalcWatP, 424
ConvAbRe, 470
ConvCF, 467
HxFct, 439
MonAn08, 917
MonAnDi, 942
OptiOT, 452
ParaCtrl/ParaMem, 223
PCalcWatP, 739
PHxFct, 744
PMonAn08, 777
PMonAnDi, 782
PMonAnL, 772
PMonDi08, 793
PMonDiL, 788
PMotL, 749
PMotRevL, 755
PMotSpdCL, 760
PMotSpdL, 766
PolygonExt, 842
POpAnL, 798
POpD, 803
PPIDL, 809
PUsrM, 832
PVlvL, 814
PVlvMotL, 820
PVlvPosL, 826
SelB, 586
SelByt, 590
SelC, 598
SelDI, 596
SelDW, 594
SelFp, 243
SelI, 588
SelR, 584
SelStr, 581
SelW, 592
SimAn, 1053
SimDi, 1059
SPCurve, 869
TimeSwitch, 892
UsrM, 246
VlvDsL, 712
VoteAnL, 601
VoteDiL, 639
Calling OBs, 745
AccuS, 671
APC_Supervisor, 133
ARcvH, 559
ASSendH, 554
ASTimeBCD, 915
CalcWatP, 424
ConvAbRe, 470
ConvCF, 467
HxFct, 440
MonAn08, 918
MonAnDi, 942
ParaCtrl/ParaMem, 223
PCalcWatP, 740
PHxFct, 745
PMonAn08, 778
PMonAnDi, 783
PMonAnL, 773
PMonDi08, 793
PMonDiL, 789
PMotL, 750
PMotRevL, 756
PMotSpdCL, 761
PolygonExt, 842
POpAnL, 798
POpD, 803
PPIDL, 809
PulseCon, 361
PUsrM, 833
PVlvL, 815
PVlvMotL, 821
PVlvPosL, 827
SelB, 586
SelByt, 590
SelC, 598
SelDI, 596
SelDW, 594
SelFp, 243

Sell, 588
SelR, 584
SelStr, 581
SelW, 592
SimAn, 1053
SimDi, 1059
Subentry, 452
UsrM, 246
VlvDsL, 713
VoteAnL, 601
VoteDiL, 639
CM
Input parameters, 74
Output parameters, 74, 75
CmGen
Input parameters, 111
Output parameters, 112
Command and feedback processing
Input parameters, 73
Configurable reactions using the Feature parameter
SPCurve, 876
Configuration
Aggr16 / Aggr08, 181
CalcPwrL, 303
ILck4x3L, 992
ILck8x4L, 1020
LdMgmt8, 272
PCalcWatP, 739
PeakMon, 333
PHxFct, 744
PMonAn08, 777
PMonAnDi, 782
PMonAnL, 772
PMonDi08, 793
PMotL, 749
PMotRevL, 755
PMotSpdCL, 760
PMotSpdL, 766
PolyCurve, 853
POpAnL, 798
POpD, 803
PPIDL, 808
PUsrM, 832
PVlvL, 814
PVlvMotL, 820
PVlvPosL, 826
RcvH_AnaVal, 537
RcvH_DigVal, 533
S7RcvAna, 521
S7RcvDig, 518
S7RcvHAna, 550
S7RcvHDig, 546
S7SndAna, 515
S7SndDig, 512
S7SndHAna, 543
S7SndHDig, 540
SelFp, 242
SndH_AnaVal, 529
SndH_DigVal, 525
SPCurve, 868
SplRng4, 907
TimeSwitch, 892
VlvDiv, 681
Configuring instructions
APC, 125
PMonDiL, 788
PMotSpdL, 767
Consider bad quality of the automatic commands or the external values
VlvDiv, 689
VlvDsL, 718
Control system fault
CalcPwrL, 313
LdMgmt8, 286
MonAnDi, 950
MonAnL, 927
PeakMon, 343
VlvDiv, 696
VlvDsL, 723
Control system fault (CSF), 927, 949
ConvAbRe
Block diagram, 472
Called blocks, 470
Calling OBs, 470
Error handling, 471
Functions, 470
Input parameters, 471
Object name, 470
Operating modes, 470
Operator control and monitoring, 472
Output parameters, 472
Quality code, 471
Startup characteristics, 470
Time response, 470
ConvCF
Block diagram, 469
Called blocks, 467
Calling OBs, 467
Error handling, 468
Functions, 467
Input parameters, 468
Object name, 467
Operating modes, 467
Operator control and monitoring, 469

Output parameters, 468
Quality code, 468
Startup characteristics, 467
Time response, 467

D

Deadband
MonAn08, 921
MonAnDi, 945
Delay of activation and deactivation
Aggr16/ Aggr08, 186
DIN
Input parameters, 967
Output parameters, 968

E

EQM
Input parameters, 74
Output parameters, 74, 75
EqmIF
Input parameters, 1088
Output parameters, 1088
ERC circuit
CalcTHX, 477
ERC optimization
CalcTHX, 476
Error handling
AccuS, 674
Aggr16/ Aggr08, 187
APC_MpList, 171
APC_MV, 167
APC_OpSP, 158
APC_Supervisor, 138
ASRcvH, 560
ASSendH, 555
ASTimeBCD, 915
CalcTHX, 485
CalcWatP, 427
ConvAbRe, 471
ConvCF, 468
HxFct, 442
MonAn08, 926, 949
OptiOT, 458
ParaCtrl/ParaMem, 230
PCalcWatP, 741
PHxFct, 746
PMonAn08, 779
PMonAnDi, 784
PMonAnL, 774
PMonDi08, 794
PMonDiL, 790
PMotL, 751
PMotRevL, 757
PMotSpdCL, 762
PMotSpdL, 768
PolyCurve, 860
POpAnL, 799
POpD, 804
PPIDL, 810
PUsrM, 834
PVlvL, 816
PVlvMotL, 822
PVlvPosL, 828
SelB, 586
SelByt, 590
SelC, 598
SelDI, 596
SelDW, 594
SelI, 588
SelR, 584
SelStr, 582
SelW, 592
SimAn, 1055
SimDi, 1060
SplRng4, 912
TimeSwitch, 896
UsrM, 247
Error numbers
VlvDsL, 721
External error
PeakMon, 313, 342
VlvDiv, 695
External/internal setpoint specification
Aggr16/ Aggr08, 184

F

Feature
AccuS, 674
Aggr16/ Aggr08, 186
APC_MV, 167
APC_OpSP, 158
APC_Supervisor, 138
CalcPwrL, 311
CalcTHX, 484
CalcWatP, 426
HxFct, 442
LdMgmt8, 284
MonAn08, 926
MonAnDi, 948
OptiOT, 457

ParaCtrl, 229
ParaMem, 229
PeakMon, 341
PolyCurve, 859
PolygonExt, 843, 844
SimAn, 1054
SimDi, 1060
SplRng4, 912
TimeSwitch, 895, 896
VlvDiv, 692
VoteAnL, 616
VoteDiL, 651
Feature2
PolyCurve, 859
VoteAnL, 617
VoteDiL, 652
First-in detection
ILck4x3L, 1002
ILck8x4L, 1034
VoteAnL, 609
VoteDiL, 647
Flutter suppression
CalcPwrL, 309, 312
PeakMon, 341
Forming the signal status for blocks
ILck4x3L, 1002
ILck8x4L, 1035
VoteAnL, 610
VoteDiL, 647
Function
APC_MV, 166
APC_OpSP, 157
APC_Supervisor, 136
ASTimeBCD, 915
CalcTHX, 476
PMonAnL, 774
PMotL, 757
SelByt, 590
SelC, 598
SelDI, 596
SelI, 588
SelW, 592
TimeSwitch, 894
Functional principles
Aggr16 / Aggr08, 181
ARcvH, 559
ASSendH, 554
CalcTHX, 474
SPCurve, 868
VlvDsL, 712
Functions
APC_MpList, 171
CalcPwrL, 305
ConvAbRe, 470
ConvCF, 467
ILck4x3L, 997
ILck8x4L, 1029
LdMgmt8, 275
OptiOT, 456
PCalcWatP, 741
PeakMon, 336
PHxFct, 745
PMonAn08, 778
PMonAnDi, 784
PMonDi08, 794
PMonDiL, 789
PMotL, 751
PMotSpdCL, 762
PMotSpdL, 768
PolyCurve, 856
PolygonExt, 843
POpAnL, 799
POpD, 804
PPIDL, 810
PUsrM, 834
PVlvL, 816
PVlvMotL, 822
PVlvPosL, 828
RcvH_AnaVal, 537
RcvH_DigVal, 533
S7RcvAna, 522
S7RcvDig, 518
S7RcvHAna, 550
S7RcvHDig, 547
S7SndAna, 515
S7SndDig, 512
S7SndHAna, 543
S7SndHDig, 540
SelB, 586
SelDW, 594
SelR, 584
SelStr, 581
SndH_AnaVal, 530
SndH_DigVal, 526
SplRng4, 909
TimeSwitch, 894
UsrM, 246
VlvDiv, 687
FxCtrl
Input parameters, 382
Output parameters, 384
FxDef16
Input parameters, 411
Output parameters, 411

FxDef32
Input parameters, 411
Output parameters, 411
FxDef64
Input parameters, 411
Output parameters, 411
FxShare
Input parameters, 417
Output parameters, 418
FxSigBin
Input parameters, 403
Output parameters, 404
FxSigInt
Input parameters, 407
Output parameters, 408

G

Gradient monitoring
MonAnL, 922, 946
Group error
VlvDiv, 695
Group status
VlvDiv, 688

H

Handling non-connected inputs
ILck4x3L, 1001
ILck8x4L, 1034
VoteAnL, 606
VoteDiL, 644
How it works
ILck4x3L, 991
ILck8x4L, 1019
RcvH_AnaVal, 536
RcvH_DigVal, 532
S7RcvAna, 520
S7RcvDig, 517
S7RcvHAna, 549
S7RcvHDig, 545
S7SndAna, 514
S7SndDig, 511
S7SndHAna, 542
S7SndHDig, 539
SndH_AnaVal, 528
SndH_DigVal, 524
HxFct, 441
Area of application, 439
Associated values, 443
Block diagram, 446
Block icon, 450
Called blocks, 439
Calling OBs, 440
Error handling, 442
Feature, 442
Input parameters, 444
Object name, 439
Operating modes, 440
Operator authorizations, 441
Output parameters, 445
Parameter view, 448
Preview, 449
Process messages, 443
Release for maintenance, 441
Simulating signals, 441
Standard view, 447
Startup characteristics, 439
Status word assignment, 440
Time response, 439

I

I/Os
Aggr08/ Aggr16, 190
ILck4x3L, 1004
ILck8x4L, 1037
Idle state
VlvDiv, 691
ILck4x3L
Access control, 1002
Area of application, 991
Block diagram, 1007
Block icon, 1017
Bypass, 999
Configuration, 992
Extended standard view, 1010
First-in detection, 1002
Forming the signal status for blocks, 1002
Functions, 997
Handling non-connected inputs, 1001
How it works, 991
I/Os, 1004
Inversion of logic signals, 999
Logic operators, 997
Messages, 1004
Object name, 991
Opening additional faceplates, 1002
Operating modes, 997
Overview of error numbers, 1004
Parameter view, 1015
Preview, 1016

Responses that can be configured with the "Feature" parameter, 1003
Standard view, 1008
Startup characteristics, 992
Status word assignment, 992
Troubleshooting, 1004
ILck8x4L
Access control, 1035
Area of application, 1019
Block diagram, 1040
Block icon, 1051
Bypass, 1032
Configuration, 1020
Extended standard view, 1044
First-in detection, 1034
Forming the signal status for blocks, 1035
Functions, 1029
Handling non-connected inputs, 1034
How it works, 1019
I/Os, 1037
Inversion of logic signals, 1031
Logic operators, 1030
Messages, 1037
Object name, 1019
Opening additional faceplates, 1034
Operating modes, 1029
Overview of error numbers, 1037
Parameter view, 1048
Preview, 1050
Responses that can be configured with the "Feature" parameter, 1036
Standard view, 1041
Startup characteristics, 1020
Status word assignment, 1020
Troubleshooting, 1036
Input parameters
AccuS, 674
APC_MV, 168
ASRcvH, 561
ASSendH, 557
CalcTHX, 487
CalcWatP, 428
ConvAbRe, 471
ConvCF, 468
HxFct, 444
LdMgmt8, 287
MonAn08, 929
MonAnDi, 951
OptiOT, 458
ParaCtrl, 231
ParaMem, 232
PCalcWatP, 741
PeakMon, 314, 344
PHxFct, 746
PMonAn08, 779
PMonAnDi, 784
PMonAnL, 774
PMonDi08, 795
PMonDiL, 790
PMotL, 751
PMotRevL, 757
PMotSpdCL, 762
PMotSpdL, 768
PolyCurve, 861
PolygonExt, 846
POpAnL, 800
POpD, 805
PPIDL, 811
PUsrM, 834
PVlvL, 816
PVlvMotL, 822
PVlvPosL, 829
SelB, 587
SelByt, 591
SelC, 599
SelDI, 597
SelDW, 595
SelFp, 244
SelI, 589
SelR, 585
SelStr, 582
SelW, 593
SimAn, 1055
SimDi, 1061
SPCurve, 878
TimeSwitch, 898
UsrM, 249
VlvDiv, 697
VlvDsL, 724
Inputs
APC, 127
Inputs/outputs
SplRng4, 913
Instance-specific messages
Aggr16/ Aggr08, 189
MonAnL, 928
VoteAnL, 618
INTBIT
Parameter, 173
Integrate
AccuS, 672
Interconnecting
CalcWatP, 426

MonAn08, 922
MonAnDi, 946
Interlocks
VlvDiv, 688
Inversion of logic signals
ILck4x3L, 999
ILck8x4L, 1031

L

LdMdmt8
Parameter view, 297
Preview, 299
Standard view, 292
Time view, 297
LdMgmt8
Area of application, 272
Associated values, 287
Automatic mode, 275
Auxiliary values, 276
Block icon, 301
Configuration, 272
Control system fault, 286
Energy data view, 295
Feature, 284
Functions, 275
Input parameters, 287
Manual mode, 275
Messages, 286
Messaging, 286
Object name, 272
Operating modes, 275
Operator authorizations, 283
Out of service, 275
Output parameters, 289
Release for maintenance, 275
Startup characteristics, 272
Status word allocation, 272, 273, 274
Subentry, 272
Local mode
VlvDiv, 685
Local operator authorization, 41
Logic operators
ILck4x3L, 997
ILck8x4L, 1030
Loop
APC, 126
Low pass filter
MonAn08, 309, 921
MonAnDi, 945

M

Main entry
CalcPwrL, 303
LdMgmt8, 272
PeakMon, 333
PulseCon, 361
Subentry, 846
VlvDiv, 681
VlvDsL, 715
Maintenance data
Input parameters, 77
Output parameters, 77
Manual mode
LdMgmt8, 275
SPCurve, 872
VlvDiv, 686
VlvDsL, 716
Measured value monitoring
MonAn08, 921
MonAnDi, 945
Message behavior
AccuS, 674
Aggr16/ Aggr08, 188
APC_MpList, 171
APC_MV, 167
APC_Supervisor, 139
ASTimeBCD, 916
CalcTHX, 485
CalcWatP, 427
MonAnDi, 949
MonAnL, 927
ParaCtrl/ParaMem, 231
PCalcWatP, 741
PHxFct, 746
PMonAn08, 779
PMonAnDi, 784
PMonAnL, 774
PMonDi08, 795
PMonDiL, 790
PMotL, 751
PMotRevL, 757
PMotSpdCL, 762
PMotSpdL, 768
PolyCurve, 860
POpAnL, 800
POpD, 804
PPIDL, 810
PUsrM, 834
PVlvL, 816
PVlvMotL, 822

SelB, 586
SelByt, 590
SelC, 598
SelDI, 596
SelDW, 594
SelFp, 243
SelI, 588
SelR, 584
SelStr, 582
SelW, 592
SimAn, 1055
SimDi, 1061
UsrM, 247
VlvDiv, 695
Message functionality
CalcPwrL, 313
SplRng4, 913
Message response
VoteAnL, 618
VoteDiL, 653
Message view
Panel block, 55
Messages
ASRcvH, 560
ASSendH, 556
CalcPwrL, 311, 313, 314
CalcTHX, 486
ILck4x3L, 1004
ILck8x4L, 1037
LdMgmt8, 286
MonAnDi, 950
PeakMon, 287, 341, 343
RcvH_AnaVal, 537
RcvH_DigVal, 534
S7RcvAna, 522
S7RcvDig, 519
S7RcvHAna, 551
S7RcvHDig, 547
S7SndAna, 516
S7SndDig, 513
S7SndHAna, 544
S7SndHDig, 541
SndH_AnaVal, 530
SndH_DigVal, 526
VlvDiv, 692, 696
VlvDsL, 723
VoteAnL, 617
VoteDiL, 652, 653
Messaging
LdMgmt8, 286
PeakMon, 342
PulseCon, 363
PVlvPosL, 828
Method of operation
SplRng4, 907
MidStr'
Parameter, 177
Mode switch
Detailed representation, 76
Input parameters, 76
Output parameters, 76
Mof
Input parameters, 982
Output parameters, 983
MonAn08, (Called blocks)
Area of application, 917
Block icon, 940
Bypass function, 922
Calling OBs, 918
Deadband, 921
Error handling, 926, 949
Feature, 926
Input parameters, 929
Interconnecting, 922
Low pass filter, 309, 921
Measured value monitoring, 921
Object name, 917
Operating modes, 920, 945
Operator authorizations, 924
Output parameters, 930
Overview of error numbers, 927, 949
Parameter view, 938
Preview, 939
Quality code, 921
Release for maintenance, 922
Simulate, 922
Standard view, 932
Startup characteristics, 917
Status word assignment, 918, 920
Time response, 917
Views, 781
MonAn08L
Limit value view, 935
MonAnDi
Area of application, 942
Associated values, 950
Block icon, 962
Bypass function, 946
Called blocks, 942
Calling OBs, 942
Control system fault, 950
Deadband, 945
Feature, 948

Input parameters, 951
Interconnecting, 946
Limit value view, 958
Low pass filter, 945
Measured value monitoring, 945
Message behavior, 949
Messages, 950
Object name, 942
Operator authorizations, 947
Output, 945
Output parameters, 952
Parameter view, 959
Preview, 961
Quality code, 945
Release for maintenance, 946
Simulate, 946
Standard view, 954
Startup characteristics, 942
Status word assignment, 943
Time response, 942
MonAnL
Associated values, 928
Control system fault, 927
Gradient monitoring, 922, 946
Instance-specific messages, 928
Message behavior, 927
Process messages, 928
Monitoring
ASSendH, 555
VlvDiv, 689
MotL
Input parameters, 84
Output parameters, 85
MotRevL
Input parameters, 89
Output parameters, 89
MotSpdCL
Input parameters, 93
Output parameters, 94
Multiple control room concept
Function, 41
Operator authorization, 41

N

Number of units
Aggr16/ Aggr08, 184

O

Object name
AccuS, 671
Aggr16/ Aggr08, 181
APC_MpList, 171
APC_MV, 166
APC_OpSP, 156
APC_Supervisor, 133
ARcvH, 558
ASSendH, 553
ASTimeBCD, 915
CalcPwrL, 303
CalcTHX, 473
CalcWatP, 423
ConvAbRe, 470
ConvCF, 467
HxFct, 439
ILck4x3L, 991
ILck8x4L, 1019
LdMgmt8, 272
MonAn08, 917
MonAnDi, 942
OptiOT, 452
ParaCtrl/ParaMem, 216
PCalcWatP, 739
PeakMon, 333
PHxFct, 744
PMonAn08, 777
PMonAnDi, 782
PMonAnL, 772
PMonDi08, 793
PMonDiL, 788
PMotL, 749
PMotRevL, 755
PMotSpdCL, 760
PMotSpdL, 766
PolygonExt, 841
POpAnL, 798
POpD, 803
PPIDL, 808
PulseCon, 361
PUsrM, 832
PVlvL, 814
PVlvMotL, 820
PVlvPosL, 826
RcvH_AnaVal, 535
RcvH_DigVal, 532
S7RcvAna, 520
S7RcvDig, 517
S7RcvHAna, 549

S7RcvHDig, 545
S7SndAna, 514
S7SndDig, 511
S7SndHAna, 542
S7SndHDig, 539
SelB, 586
SelByt, 590
SelC, 598
SelDI, 596
SelDW, 594
SelFp, 242
SelI, 588
SelR, 584
SelStr, 581
SelW, 592
SimAn, 1053
SimDi, 1059
SndH_AnaVal, 528
SndH_DigVal, 524
SPCurve, 868
SplRng4, 907
TimeSwitch, 892
UsrM, 246
VlvDiv, 681
VlvDsL, 712
VoteAnL, 600
VoteDiL, 638
On and Off delays
Input parameters, 69, 70
OpA16
Input parameters, 205
Output parameters, 206
OpA16Dec'
Parameter, 179
Opening additional faceplates
Aggr16/ Aggr08, 187
CalcTHX, 483
ILck4x3L, 1002
ILck8x4L, 1034
ParaCtrl, 228
VlvDsL, 716
Operating mode
RcvH_AnaVal, 537
RcvH_DigVal, 533
S7RcvAna, 521
S7RcvDig, 518
S7RcvHAna, 550
S7RcvHDig, 546
S7SndAna, 515
S7SndDig, 512
S7SndHAna, 543
S7SndHDig, 540
SndH_AnaVal, 530
SndH_DigVal, 526
Operating modes
AccuS, 672
Aggr16/ Aggr08, 183
APC_MpList, 171
APC_MV, 166
APC_OpSP, 157
APC_Supervisor, 135
ASRcvH, 560
ASSendH, 555
ASTimeBCD, 915
CalcTHX, 476
CalcWatP, 425
ConvAbRe, 470
ConvCF, 467
HxFct, 440
ILck4x3L, 997
ILck8x4L, 1029
LdMgmt8, 275
MonAn08, 920, 945
ParaCtrl/ParaMem, 225
PCalcWatP, 740
PeakMon, 335
PHxFct, 745
PMonAn08, 778
PMonAnDi, 783
PMonAnL, 773
PMonDi08, 794
PMonDiL, 789
PMotL, 751
PMotSpdCL, 762
PMotSpdL, 768
PolyCurve, 856
PolygonExt, 842
POpAnL, 799
POpD, 804
PPIDL, 810
PulseCon, 361
PUsrM, 834
PVlvL, 816
PVlvMotL, 822
SelB, 586, 757
SelByt, 590
SelC, 598
SelDI, 596
SelDW, 594
SelFp, 243
SelI, 588
SelR, 584
SelStr, 581
SelW, 592

SimAn, 1054
SimDi, 1060
SPCurve, 871
TimeSwitch, 894
UsrM, 246
VlvDiv, 685
VlvDsL, 715
VoteAnL, 606
VoteDiL, 644
Operating Modes
CalcPwrL, 305
PVlvPosL, 828
SplRng4, 908
Operator authorizations
AccuS, 673
Aggr16/ Aggr08, 186
APC_OpSP, 157
APC_Supervisor, 137
CalcPwrL, 309
CalcTHX, 483
CalcWatP, 426
HxFct, 441
LdMgmt8, 283
MonAn08, 924
MonAnDi, 947
OptiOT, 457
ParaCtrl, 228
PeakMon, 339
PolyCurve, 859
PolygonExt, 843
SPCurve, 876
TimeSwitch, 895
VlvDiv, 687
VlvDsL, 717
VoteAnL, 614
VoteDiL, 650
Operator control and monitoring
ASRcvH, 563
ASSendH, 558
ASTimeBCD, 916
SelB, 587
SelByt, 591
SelC, 599
SelDI, 597
SelDW, 595
SelI, 589
SelR, 585
SelStr, 583
SelW, 593
OptiOT
Block diagram, 460
Block icon, 466
Called blocks, 452
Calling OBs, 452
Error handling, 458
Feature, 457
Functions, 456
Input parameters, 458
Object name, 452
Operator authorizations, 457
Output parameters, 460
Parameter view, 463
Preview, 464
Quality code, 456
Simulate, 456
Standard view, 461
Startup characteristics, 452
Status word assignment, 452
OSStat
Status word assignment, 740, 745, 750, 756, 761, 767, 773, 778, 783, 789, 794, 799, 809, 821, 833
OSStatAl
Status word assignment, 740, 745, 750, 756, 761, 767, 773, 778, 783, 789, 794, 799, 821, 833
Out of service
CalcPwrL, 305
LdMgmt8, 275
VlvDiv, 686
Output
MonAnDi, 945
Output parameters
AccuS, 675
APC_MV, 169
ASRcvH, 562
ASSendH, 557
CalcTHX, 489
CalcWatP, 430
ConvAbRe, 472
ConvCF, 468
HxFct, 445
LdMgmt8, 289
MonAn08, 930
MonAnDi, 952
OptiOT, 460
ParaCtrl, 232
ParaMem, 233
PCalcWatP, 742
PeakMon, 317, 345
PHxFct, 747
PMonAn08, 780
PMonAnDi, 785
PMonAnL, 775
PMonDi08, 796

PMonDiL, 791
PMotL, 752
PMotRevL, 758
PMotSpdCL, 763
PMotSpdL, 769
PolyCurve, 862
POpAnL, 801
POpD, 805
PPIDL, 812
PUsrM, 835
PVlvL, 817
PVlvMotL, 823
PVlvPosL, 830
SelB, 587
SelByt, 591
SelC, 599
SelDI, 597
SelDW, 595
SelFp, 244
SelI, 589
SelR, 585
SelStr, 582
SelW, 593
SimAn, 1056
SimDi, 1061
SPCurve, 880
TimeSwitch, 898
UsrM, 249
VlvDiv, 698
VlvDsL, 727
Overview of error numbers
APC_Supervisor, 138
ILck4x3L, 1004
ILck8x4L, 1037
MonAn08, 927, 949
ParaCtrl/ParaMem, 230
PolyCurve, 860
SPCurve, 877
SplRng4, 912
TimeSwitch, 896
UsrM, 247
VoteAnL, 617
VoteDiL, 653

P

Panel block
Block icon, 54
Standard view, 55
Views, 54
Panel blocks
Views, 50
Panel integration
Core functionality, 44
Function, 46
Instance data block, (IDB)
Message behavior, 45
Operating philosophy, 44
ParaCtrl
Feature, 229
Input parameters, 231
Opening additional faceplates, 228
Operator authorizations, 228
Output parameters, 232
Parameter view, 238
Preview, 240
Standard view, 235
ParaCtrl/ParaMem
Block diagram, 234
Called blocks, 223
Calling OBs, 223
Error handling, 230
Message behavior, 231
Object name, 216
Operating modes, 225
Overview of error numbers, 230
Saving data records, 225
Setting parameters, 220
Startup characteristics, 223
ParaMem
Feature, 229
Input parameters, 232
Output parameters, 233
Selecting a unit of measure, 228
PCalcWatP, 739
Block diagram, 743
Called blocks, 739
Calling OBs, 740
Configuration, 739
Error handling, 741
Functions, 741
Input parameters, 741
Message behavior, 741
Object name, 739
Operating modes, 740
Output parameters, 742
Views, 743
PeakMon
Area of application, 333
Associated values, 343
Auxiliary values, 340
Block diagram, 318, 348
Block icon, 359
Configuration, 333

Control system fault, 343
External error, 313, 342
Feature, 341
Flutter suppression, 341
Functions, 336
Histogram view, 357
Input parameters, 314, 344
Limit value view, 354
Messages, 287, 341, 343
Messaging, 342
Object name, 333
Operating modes, 335
Operator authorizations, 339
Output parameters, 317, 345
Parameter view, 355
Preview, 358
Release for maintenance, 341
Setpoint specification, 341
Simulate, 341
Standard view, 349
Startup characteristics, 333
Status word allocation, 333, 334, 335
Subentry, 333
Time Basis, 339
Time window, 336
PHxFct
Block diagram, 747
Called blocks, 744
Configuration, 744
Error handling, 746
Functions, 745
Input parameters, 746
Message behavior, 746
Object name, 744
Operating modes, 745
Output parameters, 747
Startup characteristics, 744
Subentry, 745
Views, 748
PID controller, 907
PIDConL
Input parameters, 119
Output parameters, 120
PMonAn08
Block diagram, 780
Called blocks, 777
Calling OBs, 778
Configuration, 777
Error handling, 779
Functions, 778
Input parameters, 779
Message behavior, 779
Object name, 777
Operating modes, 778
Output parameters, 780
Startup characteristics, 777
PMonAnDi
Block diagram, 786
Called blocks, 782
Calling OBs, 783
Configuration, 782
Error handling, 784
Functions, 784
Input parameters, 784
Message behavior, 784
Object name, 782
Operating modes, 783
Output parameters, 785
Startup characteristics, 782
Views, 786
PMonAnDi08
Functions, 794
PMonAnL
Block diagram, 776
Called blocks, 772
Calling OBs, 773
Configuration, 772
Error handling, 774
Function, 774
Input parameters, 774
Message behavior, 774
Object name, 772
Operating modes, 773
Output parameters, 775
Startup characteristics, 772
Views, 776
PMonDi08
Block diagram, 796
Called blocks, 793
Calling OBs, 793
Configuration, 793
Error handling, 794
Input parameters, 795
Message behavior, 795
Object name, 793
Operating modes, 794
Output parameters, 796
Views, 797
PMonDiL
Block diagram, 791
Called blocks, 788
Calling OBs, 789
Configuring instructions, 788
Error handling, 790

Functions, 789
Input parameters, 790
Message behavior, 790
Object name, 788
Operating modes, 789
Output parameters, 791
Startup characteristics, 788
PMotL
Block diagram, 753
Called blocks, 749
Calling OBs, 750
Configuration, 749
Error handling, 751
Function, 757
Functions, 751
Input parameters, 751
Message behavior, 751
Object name, 749
Operating modes, 751
Output parameters, 752
Startup characteristics, 749
Views, 753
PMotRevL
Called blocks, 755
Calling OBs, 756
Configuration, 755
Error handling, 757
Input parameters, 757
Message behavior, 757
Object name, 755
Output parameters, 758
Startup characteristics, 755
Views, 759
PMotSpdCL
Block diagram, 764
Called blocks, 760
Calling OBs, 761
Configuration, 760
Error handling, 762
Functions, 762
Input parameters, 762
Message behavior, 762
Object name, 760
Operating modes, 762
Output parameters, 763
Startup characteristics, 760
Views, 764
PMotSpdL
Block diagram, 770
Called blocks, 766
Configuration, 766
Configuring instructions, 767
Error handling, 768
Functions, 768
Input parameters, 768
Message behavior, 768
Object name, 766
Operating modes, 768
Output parameters, 769
Startup characteristics, 766
Views, 770
PolyCurve
Associated values, 861
Block diagram, 862
Block icon, 867
Configuration, 853
Error handling, 860
Feature, 859
Feature2, 859
Functions, 856
Input parameters, 861
Message behavior, 860
Object name, 853
Operating modes, 856
Operator authorizations, 859
Output parameters, 862
Overview of error numbers, 860
Parameter view, 865
Preview, 866
Process messages, 860
Standard view, 863
Startup characteristics, 854
Status word assignment, 855
PolygonExt
Area of application, 841
Associated values, 845
Block diagram, 847
Called blocks, 842
Calling OBs, 842
Feature, 843, 844
Functions, 843
Input parameters, 846
Object name, 841
Operating modes, 842
Operator authorizations, 843
Parameter view, 850
Preview, 851
Process messages, 845
Standard view, 848
Startup characteristics, 841
Status word assignment, 842
POpAnL
Block diagram, 801
Called blocks, 798

Calling OBs, 798
Configuration, 798
Error handling, 799
Functions, 799
Input parameters, 800
Message behavior, 800
Object name, 798
Operating modes, 799
Output parameters, 801
Startup characteristics, 798
Views, 802
POpD
Block diagram, 805
Called blocks, 803
Calling OBs, 803
Configuration, 803
Error handling, 804
Functions, 804
Input parameters, 805
Message behavior, 804
Object name, 803
Operating modes, 804
Output parameters, 805
Startup characteristics, 803
Status word assignment, 804
Views, 806
PPIDL
Called blocks, 809
Calling OBs, 809
Configuration, 808
Error handling, 810
Functions, 810
Input parameters, 811
Message behavior, 810
Object name, 808
Operating modes, 810
Output parameters, 812
Startup characteristics, 808
Status word assignment, 809
Views, 813
Process messages
Aggr16/ Aggr08, 188
APC_OpSP, 159
APC_Supervisor, 139
CalcWatP, 427
HxFct, 443
MonAnL, 928
PolyCurve, 860
PolygonExt, 845
TimeSwitch, 897
UsrM, 248
VoteAnL, 618
Process tag list view
APC, 130
PulseCon
Area of application, 361
Block diagram, 364
Calling OBs, 361
Messaging, 363
Object name, 361
Operating modes, 361
Operator control and monitoring, 364
Subentry, 361
PuPa
Input parameters, 259
Output parameters, 261
PUsrM
Block diagram, 836
Called blocks, 832
Calling OBs, 833
Configuration, 832
Error handling, 834
Functions, 834
Input parameters, 834
Message behavior, 834
Object name, 832
Operating modes, 834
Output parameters, 835
Startup characteristics, 832
Views, 836
PVlvL
Block diagram, 818
Called blocks, 814
Calling OBs, 815
Configuration, 814
Error handling, 816
Functions, 816
Input parameters, 816
Message behavior, 816
Object name, 814
Operating modes, 816
Output parameters, 817
Startup characteristics, 814
Status word assignment, 815
Views, 818
PVlvMotL
Block diagram, 824
Called blocks, 820
Calling OBs, 821
Configuration, 820
Error handling, 822
Functions, 822
Input parameters, 822
Message behavior, 822

Object name, 820
Operating modes, 822
Output parameters, 823
Startup characteristics, 820
Views, 824
PVlvPosL
Block diagram, 830
Called blocks, 826
Calling OBs, 827
Configuration, 826
Error handling, 828
Functions, 828
Input parameters, 829
Messaging, 828
Object name, 826
Operating Modes, 828
Output parameters, 830
Startup characteristics, 826
Status word allocation, 827
Views, 831

Q

Qing
Input parameters, 1121
Output parameters, 1123
Quality code, 441
CalcTHX, 483
CalcWatP, 425
ConvAbRe, 471
ConvCF, 468
HxFct, 441
MonAn08, 921
MonAnDi, 945
OptiOT, 456

R

RcvH_AnaVal
Area of application, 536
Configuration, 537
Functions, 537
How it works, 536
Messages, 537
Object name, 535
Operating mode, 537
Startup characteristics, 537
Troubleshooting, 537
RcvH_AnaVal block diagram, 539
RcvH_AnaVal I/Os, 538
RcvH_DigVal
Area of application, 532
Configuration, 533
Functions, 533
How it works, 532
Messages, 534
Object name, 532
Operating mode, 533
Startup characteristics, 533
Time behavior, 533
Troubleshooting, 534
RcvH_DigVal block diagram, 535
Receive process
ASRcvH, 560
Release for maintenance
CalcPwrL, 312
CalcTHX, 484
CalcWatP, 425
HxFct, 441
LdMgmt8, 275
MonAn08, 922
MonAnDi, 946
PeakMon, 341
VlvDiv, 691
VoteAnL, 610
VoteDiL, 647
Responses that can be configured with the "Feature" parameter
ILck4x3L, 1003
ILck8x4L, 1036
Run sequence
APC, 127

S

S7RcvAna
Area of application, 520
Configuration, 521
Functions, 522
How it works, 520
Messages, 522
Object name, 520
Operating mode, 521
Startup characteristics, 521
Troubleshooting, 522
S7RcvAna block diagram, 523
S7RcvAna I/Os, 522
S7RcvDig
Area of application, 517
Configuration, 518
Functions, 518
How it works, 517

Messages, 519
Object name, 517
Operating mode, 518
Startup characteristics, 518
Time behavior, 518
Troubleshooting, 519
S7RcvDig block diagram, 520
S7RcvDig I/Os, 519
S7RcvHAna
Area of application, 549
Configuration, 550
Functions, 550
How it works, 549
Messages, 551
Object name, 549
Operating mode, 550
Startup characteristics, 550
Troubleshooting, 551
S7RcvHAna block diagram, 552
S7RcvHAna I/Os, 551
S7RcvHDig
Area of application, 545
Configuration, 546
Functions, 547
How it works, 545
Messages, 547
Object name, 545
Operating mode, 546
Startup characteristics, 546
Time behavior, 546
Troubleshooting, 547
S7RcvHDig block diagram, 548
S7RcvHDig I/Os, 534, 547
S7SndAna
Area of application, 514
Configuration, 515
Functions, 515
How it works, 514
Messages, 516
Object name, 514
Operating mode, 515
Startup characteristics, 515
Time behavior, 515
Troubleshooting, 515
S7SndAna block diagram, 516
S7SndAna I/Os, 516
S7SndDig
Area of application, 511
Configuration, 512
Functions, 512
How it works, 511
Messages, 513
Object name, 511
Operating mode, 512
Startup characteristics, 512
Time behavior, 512
Troubleshooting, 512
S7SndDig block diagram, 513
S7SndDig I/Os, 513
S7SndHAna
Area of application, 542
Configuration, 543
Functions, 543
How it works, 542
Messages, 544
Object name, 542
Operating mode, 543
Startup characteristics, 543
Time behavior, 543
Troubleshooting, 544
S7SndHAna block diagram, 545
S7SndHAna I/Os, 544
S7SndHDig
Area of application, 539
Configuration, 540
Functions, 540
How it works, 539
Messages, 541
Object name, 539
Operating mode, 540
Startup characteristics, 540
Time behavior, 540
Troubleshooting, 540
S7SndHDig block diagram, 542
S7SndHDig I/Os, 541
Saving data records
ParaCtrl/ParaMem, 225
SelB
Block diagram, 587, 759, 812
Called blocks, 586
Calling OBs, 586
Error handling, 586
Functions, 586
Input parameters, 587
Message behavior, 586
Object name, 586
Operating modes, 586, 757
Operator control and monitoring, 587
Output parameters, 587
SelByt
Block diagram, 591
Called blocks, 590
Calling OBs, 590
Error handling, 590

Function, 590
Input parameters, 591
Message behavior, 590
Object name, 590
Operating modes, 590
Operator control and monitoring, 591
Output parameters, 591
SelC
Block diagram, 599
Called blocks, 598
Calling OBs, 598
Error handling, 598
Function, 598
Input parameters, 599
Message behavior, 598
Object name, 598
Operating modes, 598
Operator control and monitoring, 599
Output parameters, 599
SelDI
Block diagram, 597
Calling OBs, 596
Error handling, 596
Function, 596
Input parameters, 597
Message behavior, 596
Object name, 596
Operating modes, 596
Operator control and monitoring, 597
Output parameters, 597
Subentry, 596
SelDW
Block diagram, 595
Called blocks, 594
Calling OBs, 594
Error handling, 594
Functions, 594
Input parameters, 595
Message behavior, 594
Object name, 594
Operating modes, 594
Operator control and monitoring, 595
Output parameters, 595
Select unit of measure
VoteAnL, 614
VoteDiL, 650
Selecting a unit of measure
ParaMem, 228
SelFp
Area of application, 242
Block diagram, 244, 881, 931, 953
Called blocks, 243
Calling OBs, 243
Configuration, 242
Input parameters, 244
Message behavior, 243
Object name, 242
Operating modes, 243
Output parameters, 244
Standard view, 245
Startup characteristics, 243
SelI
Input parameters, 589
Output parameters, 589
SelI
Block diagram, 589
Called blocks, 588
Calling OBs, 588
Error handling, 588
Function, 588
Message behavior, 588
Object name, 588
Operating modes, 588
Operator control and monitoring, 589
SelR
Block diagram, 585
Called blocks, 584
Calling OBs, 584
Error handling, 584
Functions, 584
Input parameters, 585
Message behavior, 584
Object name, 584
Operating modes, 584
Operator control and monitoring, 585
Output parameters, 585
SelStr
Block diagram, 582
Called blocks, 581
Calling OBs, 581
Error handling, 582
Functions, 581
Input parameters, 582
Message behavior, 582
Object name, 581
Operating modes, 581
Operator control and monitoring, 583
Output parameters, 582
SelW
Block diagram, 593
Called blocks, 592
Calling OBs, 592
Error handling, 592
Function, 592

Input parameters, 593
Message behavior, 592
Object name, 592
Operating modes, 592
Operator control and monitoring, 593
Output parameters, 593
Setpoint specification
PeakMon, 341
Setting parameters
ParaCtrl/ParaMem, 220
Signal status
VlvDiv, 688
SimAn
Area of application, 1053
Block diagram, 1056
Called blocks, 1053
Calling OBs, 1053
Error handling, 1055
Feature, 1054
Input parameters, 1055
Message behavior, 1055
Object name, 1053
Operating modes, 1054
Output parameters, 1056
Preview, 1058
Standard view, 1057
Status word assignment, 1053, 1054
SimDi
Area of application, 1059
Block diagram, 1061
Called blocks, 1059
Calling OBs, 1059
Error handling, 1060
Feature, 1060
Input parameters, 1061
Message behavior, 1061
Object name, 1059
Operating modes, 1060
Output parameters, 1061
Preview, 1063
Standard view, 1062
Status word assignment, 1059
Simulate
CalcPwrL, 311
CalcWatP, 425
MonAn08, 922
MonAnDi, 946
OptiOT, 456
PeakMon, 341
VlvDiv, 691
VoteAnL, 610
Simulating signals
CalcTHX, 485
HxFct, 441
Snd_DigVal
Time behavior, 521, 537, 550
SndH_AnaVal
Area of application, 528
Configuration, 529
Functions, 530
How it works, 528
Messages, 530
Object name, 528
Operating mode, 530
Startup characteristics, 529
Time behavior, 530
Troubleshooting, 530
SndH_AnaVal block diagram, 532
SndH_AnaVal I/Os, 530
SndH_DigVal
Area of application, 524
Configuration, 525
Functions, 526
How it works, 524
Messages, 526
Object name, 524
Operating mode, 526
Startup characteristics, 525
Time behavior, 525
Troubleshooting, 526
SndH_DigVal block diagram, 527
SndH_DigVal I/Os, 526
SPCurve
Automatic mode, 871
Block icon, 889
Called blocks, 869
Configurable reactions using the Feature parameter, 876
Configuration, 868
Functional principles, 868
Functions, 872
Input parameters, 878
Manual mode, 872
Object name, 868
Operating modes, 871
Operator authorizations, 876
Output parameters, 880
Overview of error numbers, 877
Parameter view, 886
Preview, 888
Standard view, 883
Startup characteristics, 869
Status word assignment, 870

Splitting of the output signal of a controller
SplRng4, 909
SplRng4
Area of application, 907
Configuration, 907
Error handling, 912
Feature, 912
Functions, 909
Inputs/outputs, 913
Message functionality, 913
Method of operation, 907
Object name, 907
Operating Modes, 908
Overview of error numbers, 912
Splitting of the output signal of a controller, 909
Startup characteristics, 908
Status word allocation, 908
Standard view
Panel block, 55
Startup characteristics, 739
AccuS, 671
Aggr16 / Aggr08, 182
APC_Supervisor, 133
ARcvH, 559
ASSendH, 554
CalcPwrL, 303
CalcTHX, 475
CalcWatP, 424
ConvAbRe, 470
ConvCF, 467
HxFct, 439
ILck4x3L, 992
ILck8x4L, 1020
LdMgmt8, 272
MonAn08, 917
MonAnDi, 942
OptiOT, 452
ParaCtrl/ParaMem, 223
PCalcWatP, 739
PeakMon, 333
PHxFct, 744
PMonAn08, 777
PMonAnDi, 782
PMonAnL, 772
PMonDiL, 788
PMotL, 749
PMotRevL, 755
PMotSpdCL, 760
PMotSpdL, 766
PolyCurve, 854
PolygonExt, 841
POpAnL, 798
POpD, 803
PPIDL, 808
PUsrM, 832
PVlvL, 814
PVlvMotL, 820
PVlvPosL, 826
RcvH_AnaVal, 537
RcvH_DigVal, 533
S7RcvAna, 521
S7RcvDig, 518
S7RcvHAna, 550
S7RcvHDig, 546
S7SndAna, 515
S7SndDig, 512
S7SndHAna, 543
S7SndHDig, 540
SelFp, 243
SndH_AnaVal, 529
SndH_DigVal, 525
SPCurve, 869
SplRng4, 908
TimeSwitch, 892
UsrM, 246
VlvDiv, 682
VlvDsL, 712
VoteAnL, 601
VoteDiL, 639
States
Functions, 872
Status word allocation
CalcPwrL, 303, 304, 305
LdMgmt8, 272, 273, 274
PeakMon, 333, 334, 335
PVlvPosL, 827
SplRng4, 908
Status word assignment
AccuS, 672
Aggr16 / Aggr08, 182
APC_OpSP, 156
APC_Supervisor, 133
CalcTHX, 475
CalcWatP, 424
HxFct, 440
ILck4x3L, 992
ILck8x4L, 1020
MonAn08, 918, 920
MonAnDi, 943
OptiOT, 452
OSStat, 740, 745, 750, 756, 761, 767, 773, 778, 783, 789, 794, 799, 809, 821, 833
OSStatAl, 740, 745, 750, 756, 761, 767, 773, 778, 783, 789, 794, 799, 821, 833

PolyCurve, 855
PolygonExt, 842
POpD, 804
PPIDL, 809
PVlvL, 815
SimAn, 1053, 1054
SimDi, 1059
SPCurve, 870
TimeSwitch, 893
VlvDiv, 682
VlvDsL, 713
VoteAnL, 601
VoteDiL, 639
StrMid
Parameter, 178
Subentry
Subentry, 846
Switchover criterion
Aggr16/ Aggr08, 184

T

Tank
Input parameters, 1173
Output parameters, 1176
TcCfg
Input parameters, 1220
Output parameters, 1220
TcCtrl
Input parameters, 1198
Output parameters, 1201
TcCzSel
Input parameters, 1218
Output parameters, 1218
TcTsSel
Input parameters, 1216
Output parameters, 1216
Time base
CalcPwrL, 308
Time Basis
PeakMon, 339
Time behavior
RcvH_DigVal, 533
S7RcvDig, 518
S7RcvHDig, 546
S7SndAna, 515
S7SndDig, 512
S7SndHAna, 543
S7SndHDig, 540
Snd_DigVal, 521, 537, 550
SndH_AnaVal, 530
SndH_DigVal, 525
Time response
AccuS, 671
ARcvH, 559
ASSendH, 554
CalcTHX, 475
CalcWatP, 424
ConvAbRe, 470
ConvCF, 467
HxFct, 439
MonAn08, 917
MonAnDi, 942
VoteAnL, 601
VoteDiL, 639
Time stamp
VoteAnL, 617
VoteDiL, 652
Time stamping
VlvDiv, 693
Time window
CalcPwrL, 305
PeakMon, 336
Timer
Input parameters, 1223
Output parameters, 1224
TimeSwitch
Associated values, 897
Block diagram, 899
Block icon, 904
Called blocks, 892
Configuration, 892
Error handling, 896
Feature, 895, 896
Function, 894
Functions, 894
Input parameters, 898
Object name, 892
Operating modes, 894
Operator authorizations, 895
Output parameters, 898
Overview of error numbers, 896
Parameter view, 902
Preview, 903
Process messages, 897
SplRng4, 914
Standard view, 900
Startup characteristics, 892
Status word assignment, 893
Troubleshooting
ILck4x3L, 1004
ILck8x4L, 1036
RcvH_AnaVal, 537
RcvH_DigVal, 534

S7RcvAna, 522
S7RcvDig, 519
S7RcvHAna, 551
S7RcvHDig, 547
S7SndAna, 515
S7SndDig, 512
S7SndHAna, 544
S7SndHDig, 540
SndH_AnaVal, 530
SndH_DigVal, 526
VoteAnL, 617
VoteDiL, 653

U

UsrM
Area of application, 246
Associated values, 248
Block diagram, 250
Block icon, 251
Called blocks, 246
Calling OBs, 246
Error handling, 247
Functions, 246
Input parameters, 249
Message behavior, 247
Object name, 246
Operating modes, 246
Output parameters, 249
Overview of error numbers, 247
Process messages, 248
Standard view, 250
Startup characteristics, 246

V

Views
MonAn08, 781
Panel block, 54
PCalcWatP, 743
PHxFct, 748
PMonAnDi, 786
PMonAnL, 776
PMonDi08, 797
PMotL, 753
PMotRevL, 759
PMotSpdCL, 764
PMotSpdL, 770
POpAnL, 802
POpD, 806
PPIDL, 813
PUsrM, 836
PVlvL, 818
PVlvMotL, 824
PVlvPosL, 831
VlvAnL
Input parameters, 102
Output parameters, 103
VlvDiv
Area of application, 681
Associated values, 696
Automatic mode, 686
Auxiliary values, 693
Block diagram, 291, 698
Block icon, 708
Buttons, 693
Configuration, 681
Consider bad quality of the automatic commands or the external values, 689
Control system fault, 696
External error, 695
Feature, 692
Functions, 687
Group error, 695
Group status, 688
Idle state, 691
Input parameters, 697
Interlocks, 688
Local mode, 685
Manual mode, 686
Message behavior, 695
Messages, 692, 696
Monitoring, 689
Object name, 681
Operating modes, 685
Operator authorizations, 687
Out of service, 686
Output parameters, 698
Parameter view, 702
Preview, 704
Release for maintenance, 691
Signal status, 688
Simulate, 691
Standard view, 699
Startup characteristics, 682
Status word assignment, 682
Subentry, 681
Time stamping, 693
Warning times, 691
VlvDS
Input parameters, 107
Output parameters, 107

VlvDsL
Automatic mode, 715
Block diagram, 727
Block icon, 736
Called blocks, 712
Calling OBs, 713
Consider bad quality of the automatic commands or the external values, 718
Control system fault, 723
Error numbers, 721
Functional principles, 712
Input parameters, 724
Manual mode, 716
Messages, 723
Object name, 712
Opening additional faceplates, 716
Operating modes, 715
Operator authorizations, 717
Output parameters, 727
Preview, 734
Seat lifting view, 732
Standard view, 729
Startup characteristics, 712
Status word assignment, 713
VlvL
Input parameters, 98
Output parameters, 98
VoteAnL, (Called blocks)
Area of application, 600
Associated values, 619
Block diagram, 624
Block icon, 636
Bypass, 607
Calling OBs, 601
Feature, 616
Feature2, 617
First-in detection, 609
Forming the signal status for blocks, 610
Handling non-connected inputs, 606
Instance-specific messages, 618
Limit value view, 632
Message response, 618
Messages, 617
Object name, 600
Operating modes, 606
Operator authorizations, 614
Overview of error numbers, 617
Parameter view, 633
Preview, 635
Process messages, 618
Release for maintenance, 610
Select unit of measure, 614
Simulate, 610
Standard view, 625
Startup characteristics, 601
Status word assignment, 601
Time response, 601
Time stamp, 617
Troubleshooting, 617
VoteDiL, (Called blocks)
Area of application, 638
Associated values, 654
Block diagram, 658
Block icon, 668
Bypass, 645
Calling OBs, 639
Feature, 651
Feature2, 652
First-in detection, 647
Forming the signal status for blocks, 647
Handling non-connected inputs, 644
Message response, 653
Messages, 652, 653
Object name, 638
Operating modes, 644
Operator authorizations, 650
Overview of error numbers, 653
Parameter view, 665
Preview, 667
Release for maintenance, 647
Select unit of measure, 650
Standard view, 660
Startup characteristics, 639
Status word assignment, 639
Time response, 639
Time stamp, 652
Troubleshooting, 653

W

Warning times
VlvDiv, 691

X

Xfer
Input parameters, 1141
Output parameters, 1145